



Panduan Developer

Amazon CloudFront



Amazon CloudFront: Panduan Developer

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan antara para pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan hak milik masing-masing pemiliknya, yang mungkin atau mungkin tidak terafiliasi, terkait dengan, atau disponsori oleh Amazon.

Table of Contents

Apa itu Amazon CloudFront?	1
Bagaimana Anda mengatur CloudFront untuk mengirimkan konten	2
Pilih antara distribusi standar atau distribusi multi-tenant	4
Harga	5
Cara menggunakan CloudFront	5
Percepat pengiriman konten situs web statis	5
Sajikan video sesuai permintaan atau video streaming langsung	6
Mengkripsi bidang tertentu di seluruh pemrosesan sistem	6
Sesuaikan di edge	6
Sajikan konten pribadi dengan menggunakan kustomisasi Lambda@Edge	7
Bagaimana CloudFront memberikan konten	8
Cara CloudFront mengirimkan konten ke pengguna Anda	8
Cara CloudFront bekerja dengan cache tepi regional	9
CloudFront server tepi	11
Gunakan daftar awalan CloudFront terkelola	12
Bekerja dengan AWS SDKs	13
CloudFront sumber daya teknis	14
Memulai	15
Siapkan Akun AWS	15
Mendaftar untuk Akun AWS	15
Pilih cara mengakses CloudFront	16
Memulai dengan distribusi standar	16
Prasyarat	17
Membuat bucket	17
Unggah konten	18
Buat distribusi	18
Akses konten	19
Bersihkan	20
Tingkatkan distribusi dasar Anda	20
Memulai (AWS CLI)	21
Prasyarat	21
Buat bucket Amazon S3.	22
Unggah konten ke ember	22
Membuat Origin Access Control (OAC)	23

Buat distribusi standar	23
Perbarui kebijakan bucket S3 Anda	25
Konfirmasikan penyebaran distribusi	25
Akses konten Anda melalui CloudFront	26
Bersihkan	26
Memulai dengan situs web statis yang aman	28
Ikhtisar solusi	29
Terapkan solusinya	29
CloudFront paket harga flat-rate	35
Manfaat paket harga CloudFront flat-rate	36
Fitur berdasarkan tingkat paket harga	36
Fitur paket harga	37
Tunjangan penggunaan bulanan	77
Apa yang diperhitungkan dalam tunjangan penggunaan Anda	78
Apa yang terjadi ketika penggunaan melebihi tunjangan	78
Tunjangan penggunaan yang dapat dikonfigurasi pada paket Premium	79
Kelayakan berdasarkan penggunaan historis	80
Biaya yang ditanggung oleh paket Anda	81
Rute 53 manajemen DNS dan rencana Anda	81
Kurangi secara keseluruhan AWS biaya dengan paket harga	83
Kelola paket harga flat-rate Anda	83
Berlangganan distribusi baru ke paket harga	83
Berlangganan distribusi yang ada ke paket harga	84
Tingkatkan paket harga	84
Ubah tingkat penggunaan paket Premium	85
Turunkan paket harga	86
Batalkan paket harga	86
Membatalkan perubahan paket yang tertunda	87
Menghapus distribusi dengan paket harga	87
Izin	88
Flat-rate kuota rencana harga	88
Fitur yang tidak didukung	88
Fitur yang tidak didukung	89
Asosiasi yang tidak didukung	91
Account-level kendala	91
Resource-level kendala	91

Fitur tambahan yang dapat memengaruhi paket harga Anda	91
Paket harga vs. harga pay-as-you-go	93
Konfigurasi distribusi	94
Memahami cara kerja distribusi multi-tenant	96
Cara kerjanya	97
Istilah	99
Fitur yang tidak didukung	101
Kustomisasi penyewa distribusi	102
Minta sertifikat (penyewa distribusi)	106
Buat grup koneksi khusus (opsional)	113
Migrasi ke distribusi multi-tenant	115
Buat distribusi	116
Buat CloudFront distribusi di konsol	118
Nilai yang ditampilkan	123
Tautan tambahan	125
Tambahkan domain ke distribusi CloudFront standar Anda	125
Pengaturan distribusi yang telah dikonfigurasi sebelumnya	128
Asal Amazon S3	128
Asal API Gateway	129
Asal kustom dan instans EC2	131
Asal Elastic Load Balancing	132
MediaPackage asal v1	134
MediaPackage v2 asal	135
MediaTailor asal	136
Semua pengaturan distribusi	137
Pengaturan asal	138
Pengaturan perilaku cache	149
Pengaturan distribusi	164
Halaman kesalahan kustom dan caching kesalahan	173
Pembatasan geografis	175
Uji distribusi	175
Buat tautan ke objek Anda	175
Perbarui distribusi	176
Perbarui distribusi di konsol	176
Menandai CloudFront sumber daya	179
Pembatasan tag	180

Menandai sumber daya Anda untuk penagihan	181
Menambahkan, mengedit, dan menghapus tag	181
Penandaan terprogram	182
Menghapus sebuah distribusi	182
Gunakan berbagai asal	185
Gunakan bucket Amazon S3	186
Gunakan MediaStore wadah atau MediaPackage saluran	198
Menggunakan Application Load Balancer	199
Menggunakan Network Load Balancer	199
Gunakan URL fungsi Lambda	199
Gunakan Amazon EC2 (atau asal kustom lainnya)	201
Gunakan grup CloudFront asal	202
Gunakan Amazon API Gateway	203
Aktifkan IPv6	203
Permintaan penampil IPv6	203
Permintaan asal IPv6	205
Gunakan penerapan berkelanjutan untuk menguji perubahan dengan aman	205
CloudFront alur kerja penerapan berkelanjutan	207
Bekerja dengan distribusi pementasan dan kebijakan penyebaran berkelanjutan	208
Pantau distribusi pementasan	218
Pelajari cara kerja penerapan berkelanjutan	219
Kuota dan pertimbangan lain untuk penyebaran berkelanjutan	221
Gunakan URL khusus	222
Persyaratan untuk menggunakan nama domain alternatif	223
Pembatasan penggunaan nama domain alternatif	224
Tambahkan nama domain alternatif	227
Memindahkan nama domain alternatif	231
Hapus nama domain alternatif	243
Gunakan wildcard dalam nama domain alternatif	245
Gunakan WebSockets	245
Bagaimana WebSocket protokol bekerja	246
WebSocket persyaratan	246
WebSocket Header yang direkomendasikan	247
Minta IP statis Anycast untuk digunakan untuk daftar yang diizinkan	247
Prasyarat	248
Minta daftar IP statis Anycast	248

Buat daftar IP statis Anycast	248
Kaitkan daftar IP statis Anycast dengan distribusi yang ada	249
Kaitkan daftar IP statis Anycast dengan distribusi baru	250
Kaitkan daftar IP statis Anycast dengan grup koneksi	251
Perbarui daftar IP statis Anycast	251
Bawa IP Anda sendiri untuk CloudFront menggunakan IPAM	252
Menggunakan gRPC	257
Bagaimana gRPC bekerja di CloudFront	257
Bekerja dengan sumber daya bersama di CloudFront	260
Prasyarat untuk berbagi sumber daya	260
Berbagi asal VPC	261
Menggunakan asal VPC bersama	264
Mengidentifikasi asal VPC bersama	264
Membatalkan berbagi asal VPC bersama	265
Tanggung jawab dan izin untuk asal VPC bersama	266
Izin untuk pemilik	266
Izin untuk konsumen	266
AWSRAMDefaultPermissionCloudFront	266
Tagihan dan pengukuran	266
Kuota sumber daya bersama	267
Caching dan ketersediaan	268
Tingkatkan rasio hit cache Anda	269
Tentukan berapa lama CloudFront cache objek Anda	269
Gunakan Origin Shield	269
Caching berdasarkan parameter string kueri	270
Memisahkan berdasarkan nilai cookie	270
Menyimpan berdasarkan header permintaan	271
Hapus Accept-Encoding header saat kompresi tidak diperlukan	272
Sajikan konten media melalui HTTP	273
Gunakan Origin Shield	273
Gunakan kasus untuk Tameng Asal	274
Pilih AWS Wilayah untuk Origin Shield	280
Aktifkan Origin Shield	282
Perkirakan biaya Origin Shield	284
Ketersediaan tinggi Origin Shield	285
Bagaimana Origin Shield berinteraksi dengan fitur lain CloudFront	285

Tingkatkan ketersediaan dengan failover asal	286
Buat grup asal	288
Kontrol batas waktu dan upaya asal	290
Gunakan failover asal dengan fungsi Lambda@Edge	291
Gunakan halaman kesalahan kustom dengan failover asal	292
Kelola kedaluwarsa cache	293
Gunakan header untuk mengontrol durasi cache untuk masing-masing objek	294
Sajikan konten basi (kedaluwarsa)	295
Tentukan jumlah waktu yang menyimpan objek dalam CloudFront cache	298
Tambahkan header ke objek Anda dengan menggunakan konsol Amazon S3	304
Caching dan parameter string kueri	304
Pengaturan konsol dan API untuk penerusan string dan caching kueri	306
Optimalkan caching	307
Parameter string kueri dan log CloudFront standar (log akses)	308
Konten cache berdasarkan cookie	309
Konten cache berdasarkan header permintaan	312
Header dan distribusi – ikhtisar	313
Pilih header untuk mendasarkan caching	314
Konfigurasikan CloudFront untuk menghormati pengaturan CORS	315
Konfigurasikan caching berdasarkan jenis perangkat	316
Konfigurasikan caching berdasarkan bahasa pemirsa	316
Konfigurasikan caching berdasarkan lokasi penampil	317
Konfigurasikan caching berdasarkan protokol permintaan	317
Konfigurasikan caching untuk file terkompresi	317
Bagaimana caching berdasarkan header memengaruhi kinerja	317
Bagaimana kasus nilai header dan header memengaruhi caching	317
Header yang CloudFront kembali ke penampil	318
Mengontrol kunci cache dengan kebijakan	319
Memahami kebijakan cache	320
Informasi kebijakan	320
Waktu ke pengaturan langsung (TTL)	320
Pengaturan tombol Cache	321
Buat kebijakan cache	327
Gunakan kebijakan cache terkelola	331
Amplify	332
CachingDisabled	333

CachingOptimized	334
CachingOptimizedForUncompressedObjects	335
Elemen- MediaPackage	336
UseOriginCacheControlHeaders	337
UseOriginCacheControlHeaders-QueryStrings	338
Memahami kunci cache	339
Kunci cache default	339
Sesuaikan kunci cache	341
Kontrol permintaan asal dengan kebijakan	343
Memahami kebijakan permintaan asal	344
Informasi kebijakan	344
Pengaturan permintaan asal	344
Buat kebijakan permintaan asal	347
Gunakan kebijakan permintaan asal terkelola	351
AllViewer	352
AllViewerAndCloudFrontHeaders-2022-06	352
AllViewerExceptHostHeader	353
CORS- CustomOrigin	354
CORS-S3asal	355
Elemental- - MediaTailor PersonalizedManifests	355
HostHeaderOnly	356
UserAgentRefererHeaders	356
Tambahkan header CloudFront permintaan	357
Header jenis perangkat	358
Header lokasi penampil	359
Header untuk menentukan struktur header pemirsa	360
Header terkait TLS	360
CloudFront Header lainnya	362
Memahami bagaimana kebijakan permintaan asal dan kebijakan cache bekerja sama	362
Menambah atau menghapus header respons dengan kebijakan	366
Memahami kebijakan header respons	367
Rincian kebijakan (metadata)	367
Header CORS	368
Header keamanan	371
Header kustom	374
Hapus header	374

Header Pengaturan Waktu Server	376
Buat kebijakan header respons	381
Menggunakan kebijakan header respons terkelola	388
CORS-dan- SecurityHeadersPolicy	388
CORS-dengan-preflight	389
CORS- - with-preflight-and SecurityHeadersPolicy	390
SecurityHeadersPolicy	391
SimpleCORS	392
Perilaku permintaan dan respons	394
Bagaimana CloudFront memproses permintaan HTTP dan HTTPS	394
Perilaku permintaan dan respons untuk asal Amazon S3	395
Cara CloudFront memproses dan meneruskan permintaan ke asal Amazon S3 Anda	395
Bagaimana CloudFront memproses tanggapan dari asal Amazon S3 Anda	402
Perilaku permintaan dan respons untuk asal kustom	404
Cara CloudFront memproses dan meneruskan permintaan ke asal kustom Anda	404
Bagaimana CloudFront memproses tanggapan dari asal kustom Anda	422
Perilaku permintaan dan respons untuk grup asal	427
Tambahkan header khusus ke permintaan asal	427
Kasus penggunaan	428
Konfigurasi CloudFront untuk menambahkan header khusus ke permintaan asal	429
Header khusus yang tidak CloudFront dapat ditambahkan ke permintaan asal	429
CloudFront Konfigurasi untuk meneruskan Authorization header	430
Bagaimana rentang CloudFront proses GETs	431
Gunakan permintaan rentang untuk menyimpan objek besar	433
Bagaimana CloudFront memproses kode status HTTP 3xx dari asal Anda	433
Bagaimana CloudFront memproses kode status HTTP 4xx dan 5xx dari asal Anda	434
Bagaimana CloudFront proses kesalahan ketika Anda telah mengkonfigurasi halaman kesalahan kustom	435
Cara CloudFront memproses kesalahan jika Anda belum mengonfigurasi halaman kesalahan kustom	438
Kode status HTTP 4xx dan 5xx yang di-cache CloudFront	440
Hasilkan respons kesalahan khusus	441
Konfigurasi perilaku respons kesalahan	442
Buat halaman kesalahan khusus untuk kode status HTTP tertentu	444
Menyimpan objek dan halaman kesalahan kustom di lokasi yang berbeda	446
Ubah kode respons yang dikembalikan oleh CloudFront	446

Kontrol berapa lama CloudFront kesalahan cache	447
Menambahkan, menghapus, atau mengganti konten	450
Menambahkan dan mengakses konten	450
Gunakan versi file untuk memperbarui atau menghapus konten yang ada	451
Perbarui file yang ada menggunakan nama file berversi	451
Hapus konten sehingga tidak CloudFront akan mendistribusikannya	452
Kustomisasi URL file	452
Gunakan nama domain Anda sendiri (example.com)	453
Gunakan garis miring (/) di URL	453
Buat URL yang ditandatangani untuk konten terbatas	454
Tentukan objek root default	454
Cara menentukan objek root default	454
Cara kerja objek root default	456
Bagaimana cara CloudFront kerja jika Anda tidak mendefinisikan objek root	458
Membatalkan file untuk menghapus konten	458
Pilih antara membatalkan file dan menggunakan nama file berversi	459
Menentukan jalur mana yang akan dihapus validasinya	459
Yang perlu Anda ketahui saat menghapus validasi jalur	460
Membatalkan konten dengan tag cache	464
Apa yang perlu Anda ketahui saat membatalkan tag	468
Membatalkan file	470
Membayar untuk penghapusan validasi file	474
Sajikan file terkompresi	474
Konfigurasi CloudFront untuk mengompres objek	475
Bagaimana CloudFront kompresi bekerja	476
Kondisi untuk kompresi	477
Jenis file yang CloudFront mengompres	479
ETagkonversi header	480
Gunakan AWS WAF perlindungan	482
Aktifkan AWS WAF untuk distribusi	483
AWS WAF Aktifkan distribusi baru	483
Gunakan ACL web yang ada	484
Aktifkan kontrol bot	485
Konfigurasi perlindungan berdasarkan kategori bot	485
Mengelola perlindungan AWS WAF keamanan untuk CloudFront	486
Prasyarat	487

Aktifkan AWS WAF log	487
Mengatur pembatasan tarif	488
Nonaktifkan perlindungan AWS WAF keamanan	489
Konfigurasi akses aman dan batasi akses ke konten	491
Gunakan HTTPS dengan CloudFront	492
Memerlukan HTTPS antara pemirsa dan CloudFront	493
Memerlukan HTTPS ke custom origin	495
Memerlukan HTTPS ke asal Amazon S3	498
Protokol dan sandi yang didukung antara pemirsa dan CloudFront	500
Protokol dan cipher yang didukung antara dan asal CloudFront	509
Gunakan nama domain alternatif dan HTTPS	511
Pilih cara CloudFront melayani permintaan HTTPS	512
Persyaratan untuk menggunakan SSL/TLS sertifikat dengan CloudFront	516
Kuota tentang penggunaan SSL/TLS sertifikat dengan CloudFront (HTTPS antara pemirsa dan CloudFront hanya)	521
Konfigurasi nama domain alternatif dan HTTPS	523
Tentukan ukuran kunci publik dalam sertifikat SSL/TLS RSA	527
Tingkatkan kuota untuk sertifikat SSL/TLS	527
Putar SSL/TLS sertifikat	528
Kembalikan dari SSL/TLS sertifikat kustom ke sertifikat default CloudFront	530
Beralih dari SSL/TLS sertifikat khusus dengan alamat IP khusus ke SNI	531
Otentikasi TLS timbal balik dengan CloudFront (Viewer mTLS)	532
Cara kerjanya	532
Kasus penggunaan	533
Toko kepercayaan dan manajemen sertifikat	534
Aktifkan TLS timbal balik untuk distribusi CloudFront	540
Kaitkan Fungsi CloudFront Koneksi	544
Mengkonfigurasi pengaturan tambahan	549
Header mTLS penampil untuk kebijakan cache dan diteruskan ke asal	551
Pencabutan sertifikat	555
Metode pembantu untuk TLS timbal balik	564
Mode validasi tambahan	564
Observabilitas menggunakan log koneksi	568
Asal TLS timbal balik dengan CloudFront	574
Penampil mTL vs mTL Asal	574
Cara kerjanya	574

Kasus penggunaan	575
Penting: Persyaratan Server Asal	575
Memulai	575
Manajemen sertifikat dengan AWS Certificate Manager	576
Aktifkan TLS timbal balik asal untuk distribusi CloudFront	580
Menggunakan CloudFront Fungsi dengan TLS timbal balik asal	583
Batasi konten dengan URL yang ditandatangani dan cookie yang ditandatangani	586
Cara menyajikan konten pribadi	586
Batasi akses ke file	588
Tentukan penandatanganan tepercaya	590
Memutuskan untuk menggunakan URL yang ditandatangani atau cookie yang ditandatangani	600
Gunakan URL yang ditandatangani	601
Gunakan cookie yang ditandatangani	625
Perintah Linux dan OpenSSL untuk pengkodean dan enkripsi base64	657
Contoh kode untuk URL yang ditandatangani	658
Membatasi akses ke AWS asal	689
Batasi akses ke asal AWS Elemental MediaPackage v2	689
Membatasi akses ke AWS Elemental MediaStore asal	697
Batasi akses ke asal URL AWS Lambda fungsi	704
Batasi akses ke asal Amazon S3	715
Batasi akses dengan asal VPC	730
Batasi akses ke Application Load Balancers	738
Konfigurasi CloudFront untuk menambahkan header HTTP kustom ke permintaan	739
Konfigurasi Application Load Balancer untuk hanya meneruskan permintaan yang berisi header tertentu	741
(Opsional) tingkatkan keamanan solusi ini.	742
(Opsional) Batasi akses ke asal dengan menggunakan daftar awalan AWS-managed untuk CloudFront	744
Pembatasan geografis	744
Gunakan CloudFront batasan geografis	745
Gunakan layanan geolokasi pihak ketiga	747
Gunakan enkripsi tingkat lapangan untuk membantu melindungi data sensitif	749
Ikhtisar enkripsi tingkat lapangan	751
Siapkan enkripsi tingkat lapangan	752
Dekripsi bidang data di tempat asal Anda	758

Video sesuai permintaan dan video streaming langsung	762
Tentang streaming video	762
Kirimkan video sesuai permintaan	763
Konfigurasikan video sesuai permintaan untuk Microsoft Smooth Streaming	764
Mengirimkan streaming video	766
Sajikan video dengan menggunakan AWS Elemental MediaStore sebagai asal	767
Sajikan video langsung yang diformat dengan AWS Elemental MediaPackage	768
Sajikan video-on-demand konten dengan AWS Elemental MediaPackage	774
Ketahanan sadar kualitas media	779
Bidang log MQAR	781
Gunakan fungsi untuk menyesuaikan di tepi	783
Perbedaan antara CloudFront Fungsi dan Lambda @Edge	784
Sesuaikan dengan CloudFront Fungsi	786
Tutorial: Buat CloudFront fungsi sederhana	787
Tutorial: Buat CloudFront fungsi yang menggunakan nilai kunci	790
Tulis kode fungsi	793
Buat fungsi	894
Fungsi uji	898
Perbarui fungsi	903
Publikasikan fungsi	906
Mengaitkan fungsi dengan distribusi	907
CloudFront KeyValueCollection	911
Sesuaikan dengan Fungsi CloudFront Koneksi	933
Ikhtisar dan alur kerja	933
Konfigurasi dan batasan	935
Buat Fungsi CloudFront Koneksi untuk validasi TLS (penampil) bersama	936
Tulis kode Fungsi CloudFront Koneksi untuk validasi TLS (penampil) timbal balik	940
Uji Fungsi CloudFront Koneksi sebelum penerapan	949
Mengaitkan Fungsi Koneksi dengan distribusi	950
Menerapkan pencabutan sertifikat untuk TLS bersama (penampil) dengan Fungsi dan CloudFront KeyValueCollection	952
Sesuaikan dengan Lambda @Edge	958
Bagaimana Lambda @Edge bekerja dengan permintaan dan tanggapan	959
Cara menggunakan Lambda @Edge	960
Memulai dengan Lambda @Edge	961
Mengatur izin dan peran IAM	969

Tulis dan buat fungsi Lambda @Edge	976
Tambahkan pemacu untuk fungsi Lambda @Edge	981
Uji dan debug	988
Hapus fungsi dan replika	996
Struktur peristiwa	997
Bekerja dengan permintaan dan tanggapan	1014
Contoh fungsi	1020
Pembatasan pada fungsi edge	1059
Pembatasan pada semua fungsi edge	1059
Pembatasan CloudFront Fungsi	1065
Pembatasan Lambda@Edge	1068
Laporan, metrik, dan log	1073
AWS laporan penagihan dan penggunaan untuk CloudFront	1073
Lihat laporan AWS penagihan untuk CloudFront	1074
Lihat laporan AWS penggunaan untuk CloudFront	1075
Menafsirkan laporan AWS tagihan dan penggunaan Anda untuk CloudFront	1077
Lihat laporan CloudFront konsol	1083
Lihat laporan statistik CloudFront cache	1084
Lihat laporan objek CloudFront populer	1090
Lihat laporan perujuk CloudFront teratas	1096
Lihat laporan CloudFront penggunaan	1100
Lihat laporan CloudFront pemirsa	1108
Pantau CloudFront metrik dengan Amazon CloudWatch	1119
Tampilan CloudFront dan metrik fungsi tepi	1120
Buat alarm	1128
Unduh data metrik	1129
CloudFront metrik	1132
CloudFront dan logging fungsi tepi	1138
Mencatat permintaan	1138
Fungsi tepi logging	1140
Aktivitas mencatat layanan	1140
Akses log (log standar)	1140
Gunakan log akses waktu nyata	1187
Log fungsi tepi	1209
AWS CloudTrail log	1213
Lacak perubahan konfigurasi dengan AWS Config	1226

Mengatur AWS Config dengan CloudFront	1227
Lihat riwayat CloudFront konfigurasi	1228
CloudFront Evaluasi konfigurasi dengan Aturan AWS Config	1229
Keamanan	1230
Perlindungan data	1231
Enkripsi saat bergerak	1232
Enkripsi saat diam	1233
Batasi Akses ke Konten	1233
Identity and Access Management	1234
Audiens	1235
Mengautentikasi dengan identitas	1235
Mengelola akses menggunakan kebijakan	1237
Bagaimana Amazon CloudFront bekerja dengan IAM	1238
Contoh kebijakan berbasis identitas	1244
AWS kebijakan terkelola	1255
Gunakan peran tertaut layanan	1264
Memecahkan masalah CloudFront identitas dan akses	1269
Pencatatan log dan pemantauan	1271
Validasi kepatuhan	1272
CloudFront praktik terbaik kepatuhan	1273
Ketahanan	1274
CloudFront failover asal	1275
Keamanan infrastruktur	1275
Pemecahan Masalah	1277
Memecahkan masalah distribusi	1277
CloudFront mengembalikan Access Denied kesalahan	1277
CloudFront mengembalikan InvalidViewerCertificate kesalahan ketika saya mencoba menambahkan nama domain alternatif	1280
CloudFront mengembalikan kesalahan catatan DNS yang tidak dikonfigurasi dengan benar ketika saya mencoba menambahkan CNAME baru	1281
Saya tidak dapat melihat file dalam distribusi saya	1282
Pesan galat: Sertifikat: <certificate-id>sedang digunakan oleh CloudFront	1283
Memecahkan masalah kode status respons kesalahan	1284
Kode status HTTP 400 (Permintaan Buruk)	1284
Kode status HTTP 401 (Tidak Sah)	1286
Kode status HTTP 403 (Metode tidak valid)	1287

Kode status HTTP 403 (Izin Ditolak)	1287
Kode status HTTP 404 (Tidak Ditemukan)	1290
Kode status HTTP 412 (Prasyarat Gagal)	1290
Kode status HTTP 500 (Kesalahan Server Internal)	1291
Kode status HTTP 502 (Gerbang Buruk)	1291
Kode status HTTP 503 (Layanan Tidak Tersedia)	1296
Kode status HTTP 504 (Gateway Timeout)	1299
Pengujian beban CloudFront	1305
Kuota	1306
Kuota umum	1307
Kuota umum di distribusi	1307
Kuotas Umum tentang Kebijakan	1310
Kuota pada mTL dan toko kepercayaan	1312
Kuota pada Fungsi CloudFront	1313
Kuota pada Fungsi Koneksi	1313
Kuota pada toko nilai utama	1314
Kuotas di Lambda@Edge	1315
Kuota pada sertifikat SSL	1317
Kuotas pada invalidasi	1317
Kuotas pada kelompok utama	1318
Kuota pada koneksi WebSocket	1319
Kuotas pada enkripsi tingkat lapangan	1319
Kuota pada cookie (pengaturan cache warisan)	1320
Kuota pada string kueri (pengaturan cache warisan)	1320
Kuota pada header	1321
Kuota pada distribusi multi-penyewa	1322
Informasi Terkait	1323
Contoh kode	1324
Hal-hal mendasar	1325
Tindakan	1326
Skenario	1395
Buat penyewa distribusi dan distribusi multi-tenant	1396
Hapus sumber penandatanganan	1407
Memulai dengan CloudFront distribusi dasar	1409
Memulai dengan WAF	1420
Tanda tangan URLs dan cookie	1432

CloudFront Contoh fungsi	1436
Tambahkan header keamanan HTTP	1436
Tambahkan header CORS	1437
Tambahkan header kontrol cache	1438
Tambahkan header IP klien yang benar	1439
Tambahkan header asal	1440
Tambahkan index.html untuk meminta URLs	1441
Menormalkan parameter string kueri	1442
Redirect ke URL baru	1443
Menulis ulang URI permintaan	1444
Pilih asal yang lebih dekat ke penampil	1447
Gunakan pasangan kunci-nilai	1448
Validasi token sederhana	1449
Riwayat dokumen	1454
.....	mcdlxxxv

Apa itu Amazon CloudFront?

Amazon CloudFront adalah layanan web yang mempercepat distribusi konten web statis dan dinamis Anda, seperti.html, .css, .js, dan file gambar, kepada pengguna Anda. CloudFront mengirimkan konten Anda melalui jaringan pusat data di seluruh dunia yang disebut lokasi tepi. Saat pengguna meminta konten yang Anda sajikan CloudFront, permintaan akan diarahkan ke lokasi tepi yang memberikan latensi terendah (penundaan waktu), sehingga konten dikirimkan dengan performa terbaik.

- Jika konten sudah berada di lokasi tepi dengan latensi terendah, segera CloudFront kirimkan.
- Jika konten tidak berada di lokasi tepi tersebut, CloudFront ambil dari asal yang telah Anda tetapkan—seperti bucket Amazon S3, MediaPackage saluran, atau server HTTP (misalnya, server web) yang telah Anda identifikasi sebagai sumber untuk versi definitif konten Anda.

Sebagai contoh, misalkan Anda menyajikan gambar dari server web tradisional, bukan dari CloudFront. Misalnya, Anda dapat menyajikan citra, sunsetphoto.png, menggunakan URL `https://example.com/sunsetphoto.png`.

Pengguna Anda dapat dengan mudah menavigasi ke URL ini dan melihat citra. Namun mereka mungkin tidak tahu bahwa permintaan mereka dirutekan dari satu jaringan ke jaringan lain—melalui koleksi kompleks dari jaringan yang saling terhubung yang terdiri dari internet—hingga citra itu ditemukan.

CloudFront mempercepat distribusi konten Anda dengan merutekan setiap permintaan pengguna melalui jaringan AWS backbone ke lokasi tepi yang dapat melayani konten Anda dengan sebaik-baiknya. Biasanya, ini adalah server CloudFront tepi yang menyediakan pengiriman tercepat ke pemirsa. Menggunakan AWS jaringan secara dramatis mengurangi jumlah jaringan yang harus dilewati permintaan pengguna Anda, yang meningkatkan kinerja. Pengguna mendapatkan latensi lebih rendah—waktu yang diperlukan untuk memuat byte pertama file—dan tingkat transfer data yang lebih tinggi.

Anda juga mendapatkan peningkatan keandalan dan ketersediaan karena salinan file Anda (juga dikenal sebagai objek) sekarang disimpan (atau disimpan) di beberapa lokasi edge di seluruh dunia.

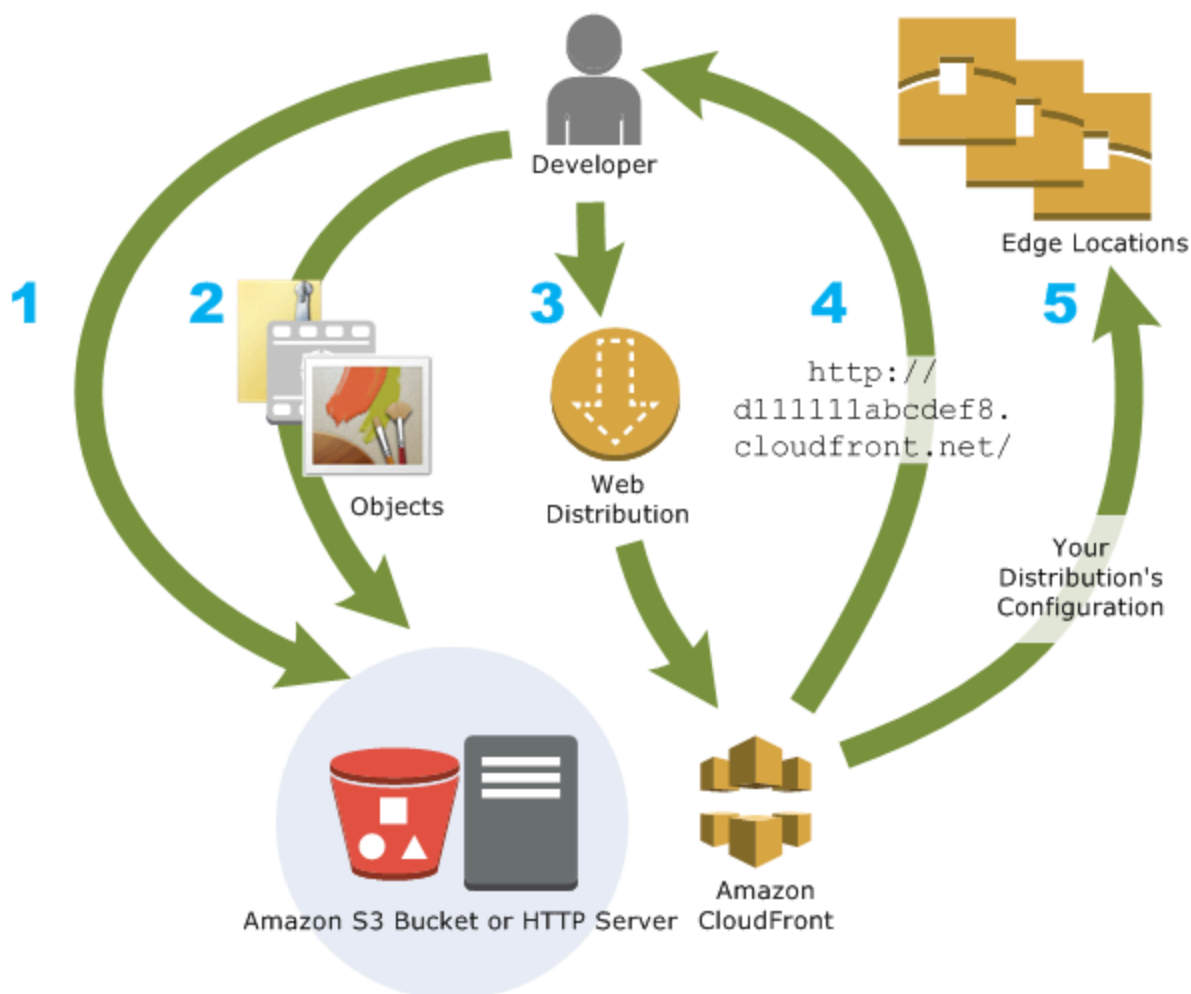
Topik

- [Bagaimana Anda mengatur CloudFront untuk mengirimkan konten](#)
- [Pilih antara distribusi standar atau distribusi multi-tenant](#)

- [Harga](#)
- [Cara menggunakan CloudFront](#)
- [Bagaimana CloudFront memberikan konten](#)
- [Lokasi dan rentang alamat IP server CloudFront edge](#)
- [Menggunakan CloudFront dengan AWS SDK](#)
- [CloudFront sumber daya teknis](#)

Bagaimana Anda mengatur CloudFront untuk mengirimkan konten

Anda membuat CloudFront distribusi untuk memberi tahu dari CloudFront mana Anda ingin konten dikirim, dan detail tentang cara melacak dan mengelola pengiriman konten. Kemudian CloudFront gunakan komputer — server tepi — yang dekat dengan pemirsa Anda untuk mengirimkan konten itu dengan cepat ketika seseorang ingin melihatnya atau menggunakannya.



Bagaimana Anda mengonfigurasi CloudFront untuk mengirimkan konten Anda

1. Anda menentukan server asal, seperti bucket Amazon S3 atau server HTTP Anda sendiri, dari mana CloudFront mendapatkan file Anda yang kemudian akan didistribusikan dari lokasi CloudFront tepi di seluruh dunia.

Server asal menyimpan versi asli dan definitif dari objek Anda. Jika Anda melayani konten melalui HTTP, server asal Anda adalah bucket Amazon S3 atau server HTTP, seperti server web. Server HTTP Anda dapat berjalan di Amazon Elastic Compute Cloud (pengecualian Amazon EC2) atau di server yang Anda kelola; server ini juga dikenal sebagai yang dibuat sesuai pesanan.

2. Anda mengunggah file ke server asal Anda. File Anda, juga dikenal sebagai objek, biasanya mencakup halaman web, citra, dan file media, tetapi dapat berupa apa pun yang dapat dilayani melalui HTTP.

Jika Anda menggunakan bucket Amazon S3 sebagai server asal, Anda dapat membuat objek di bucket dapat dibaca publik, sehingga siapa pun yang mengetahui objek Anda dapat CloudFront URLs mengaksesnya. Anda juga memiliki opsi untuk menjaga objek tetap privat dan mengendalikan siapa yang mengaksesnya. Lihat [Sajikan konten pribadi dengan URL yang ditandatangani dan cookie yang ditandatangani](#).

3. Anda membuat CloudFront distribusi, yang memberi tahu server asal CloudFront mana yang akan mendapatkan file Anda saat pengguna meminta file melalui situs web atau aplikasi Anda. Pada saat yang sama, Anda menentukan detail seperti apakah Anda CloudFront ingin mencatat semua permintaan dan apakah Anda ingin distribusi diaktifkan segera setelah dibuat.
4. CloudFront menetapkan nama domain ke distribusi baru yang dapat Anda lihat di CloudFront konsol, atau yang dikembalikan sebagai respons terhadap permintaan terprogram, misalnya, permintaan API. Jika Anda mau, Anda dapat menambahkan nama domain alternatif untuk digunakan.
5. CloudFront mengirimkan konfigurasi distribusi Anda (tetapi bukan konten Anda) ke semua lokasi tepi atau titik keberadaan (POPs) — kumpulan server di pusat data yang tersebar secara geografis tempat CloudFront menyimpan salinan file Anda.

Ketika Anda mengembangkan situs web atau aplikasi Anda, Anda menggunakan nama domain yang CloudFront menyediakan untuk Anda URLs. Misalnya, jika CloudFront kembali `d111111abcdef8.cloudfront.net` sebagai nama domain untuk distribusi Anda, URL untuk

logo.jpg di bucket Amazon S3 Anda (atau di direktori root pada server HTTP) adalah. `https://d1111111abcdef8.cloudfront.net/logo.jpg`

Atau Anda dapat mengatur CloudFront untuk menggunakan nama domain Anda sendiri dengan distribusi Anda. Dalam hal ini, URL mungkin `https://www.example.com/logo.jpg`.

Secara opsional, Anda dapat mengonfigurasi server asal Anda untuk menambahkan header ke file, untuk menunjukkan berapa lama Anda ingin file tetap berada di cache di lokasi CloudFront tepi. Secara default, setiap file tetap berada di lokasi edge selama 24 jam sebelum kedaluwarsa. Waktu kedaluwarsa minimum adalah 0 detik; tidak ada waktu kedaluwarsa maksimum. Untuk informasi selengkapnya, lihat [Mengelola berapa lama konten tetap dalam cache \(kedaluwarsa\)](#).

Pilih antara distribusi standar atau distribusi multi-tenant

CloudFront menawarkan opsi distribusi untuk situs web atau aplikasi tunggal, dan untuk skenario multi-penyewa.

Distribusi standar

Dirancang untuk konfigurasi unik per situs web atau aplikasi. Pilih ini dalam kasus penggunaan berikut:

- Anda membutuhkan distribusi mandiri CloudFront
- Setiap situs atau aplikasi memerlukan pengaturan kustom sendiri

Kebanyakan orang memulai dengan distribusi standar.

Penyewa distribusi dan distribusi multi-penyewa (Manajer SaaS CloudFront)

Dirancang khusus untuk penyedia SaaS dan skenario multi-penyewa. Pilih ini dalam kasus penggunaan berikut:

- Anda sedang membangun platform SaaS untuk melayani beberapa situs web atau aplikasi pelanggan
- Anda perlu mengelola beberapa distribusi serupa secara efisien
- Anda ingin kontrol terpusat atas konfigurasi bersama

Untuk informasi selengkapnya, lihat [Memahami cara kerja distribusi multi-tenant](#).

Harga

CloudFront biaya untuk transfer data keluar dari lokasi edge, bersama dengan permintaan HTTP atau HTTPS. Harga bervariasi menurut jenis penggunaan, wilayah geografis, dan pemilihan fitur.

Transfer data dari asal Anda ke CloudFront selalu gratis saat menggunakan AWS origin seperti Amazon Simple Storage Service (Amazon S3), Elastic Load Balancing, atau Amazon API Gateway. Anda hanya ditagih untuk transfer data keluar dari penampil CloudFront ke penampil saat menggunakan AWS asal.

Untuk informasi selengkapnya, lihat [CloudFront harga](#) dan Paket [FAQs](#) Penagihan dan Tabungan.

Cara menggunakan CloudFront

Menggunakan CloudFront dapat membantu Anda mencapai berbagai tujuan. Bagian ini hanya mencantumkan beberapa, bersama dengan tautan ke informasi lebih lanjut, untuk memberi Anda ide tentang kemungkinan.

Topik

- [Percepat pengiriman konten situs web statis](#)
- [Sajikan video sesuai permintaan atau video streaming langsung](#)
- [Mengkripsi bidang tertentu di seluruh pemrosesan sistem](#)
- [Sesuaikan di edge](#)
- [Sajikan konten pribadi dengan menggunakan kustomisasi Lambda@Edge](#)

Percepat pengiriman konten situs web statis

CloudFront dapat mempercepat pengiriman konten statis Anda (misalnya, gambar, style sheet JavaScript, dan sebagainya) ke pemirsa di seluruh dunia. Dengan menggunakan CloudFront, Anda dapat memanfaatkan jaringan AWS backbone dan server CloudFront edge untuk memberi pemirsa Anda pengalaman yang cepat, aman, dan andal ketika mereka mengunjungi situs web Anda.

Pendekatan sederhana untuk menyimpan dan mengirimkan konten statis adalah menggunakan bucket Amazon S3. Menggunakan S3 bersama-sama dengan CloudFront memiliki sejumlah keunggulan, termasuk opsi untuk menggunakan [kontrol akses asal untuk dengan mudah membatasi akses](#) ke konten Amazon S3 Anda.

Untuk informasi selengkapnya tentang penggunaan Amazon S3 bersama CloudFront, termasuk CloudFormation templat untuk membantu Anda memulai dengan cepat, lihat [Memulai dengan situs web statis yang aman](#)

Sajikan video sesuai permintaan atau video streaming langsung

CloudFront menawarkan beberapa opsi untuk streaming media Anda ke pemirsa global—baik file yang direkam sebelumnya maupun acara langsung.

- Untuk streaming video on demand (VOD), Anda dapat menggunakan CloudFront streaming dalam format umum seperti MPEG DASH, Apple HLS, Microsoft Smooth Streaming, dan CMAF, ke perangkat apa pun.
- Untuk menyiarkan streaming langsung, Anda dapat menyimpan fragmen media di tepi, sehingga beberapa permintaan file manifest yang mengirimkan fragmen dalam urutan yang tepat dapat digabungkan, untuk mengurangi beban di server asal Anda.

Untuk informasi selengkapnya tentang cara mengirimkan konten streaming CloudFront, lihat [Video sesuai permintaan dan video streaming langsung dengan CloudFront](#).

Menkripsi bidang tertentu di seluruh pemrosesan sistem

Ketika Anda mengkonfigurasi HTTPS dengan CloudFront, Anda sudah memiliki end-to-end koneksi aman ke server asal. Saat Anda menambahkan enkripsi tingkat lapangan, Anda dapat melindungi data spesifik selama pemrosesan sistem di samping keamanan HTTPS, sehingga hanya aplikasi tertentu yang berasal dari Anda yang dapat melihat data tersebut.

Untuk mengatur enkripsi tingkat bidang, Anda menambahkan kunci publik ke CloudFront, lalu tentukan kumpulan bidang yang ingin dienkripsi dengan kunci tersebut. Untuk informasi selengkapnya, lihat [Gunakan enkripsi tingkat lapangan untuk membantu melindungi data sensitif](#).

Sesuaikan di edge

Menjalankan kode nirserver di edge membuka sejumlah kemungkinan untuk menyesuaikan konten dan pengalaman bagi penonton, dengan latensi lebih rendah. Misalnya, Anda dapat mengembalikan pesan kesalahan kustom ketika server asal Anda tidak dalam pemeliharaan, sehingga penampil tidak mendapatkan pesan kesalahan HTTP generik. Atau Anda dapat menggunakan fungsi untuk membantu mengotorisasi pengguna dan mengontrol akses ke konten Anda, sebelum CloudFront meneruskan permintaan ke asal Anda.

Menggunakan Lambda @Edge dengan CloudFront memungkinkan berbagai cara untuk menyesuaikan konten yang CloudFront dikirimkan. Untuk mempelajari lebih lanjut tentang Lambda @Edge dan cara membuat dan menerapkan fungsi dengan CloudFront, lihat [Sesuaikan di tepi dengan Lambda @Edge](#) Untuk melihat sejumlah sampel kode yang dapat disesuaikan untuk solusi Anda sendiri, lihat [Lambda @Edge contoh fungsi](#).

Sajikan konten pribadi dengan menggunakan kustomisasi Lambda@Edge

Menggunakan Lambda @Edge dapat membantu Anda mengonfigurasi CloudFront distribusi Anda untuk menyajikan konten pribadi dari asal kustom Anda sendiri, selain menggunakan cookie yang ditandatangani URLs atau ditandatangani.

Untuk menyajikan konten pribadi menggunakan CloudFront, Anda melakukan hal berikut:

- Mengharuskan pengguna Anda (pemirsa) mengakses konten menggunakan [cookie yang ditandatangani URLs atau ditandatangani](#).
- Batasi akses ke asal Anda sehingga hanya tersedia dari CloudFront server yang menghadap ke asal. Untuk melakukan ini, Anda dapat melakukan salah satu dari yang berikut:
 - Untuk asal Amazon S3, Anda dapat [menggunakan kontrol akses asal \(OAC\)](#).
 - Untuk custom origin, Anda dapat melakukan hal berikut:
 - Jika asal kustom dilindungi oleh grup keamanan VPC Amazon atau AWS Firewall Manager, Anda dapat [menggunakan daftar awalan CloudFront terkelola](#) untuk mengizinkan lalu lintas masuk ke asal Anda hanya CloudFront dari alamat IP yang menghadap asal.
 - Gunakan header HTTP kustom untuk membatasi akses ke hanya permintaan dari CloudFront. Untuk informasi selengkapnya, lihat [the section called “Batasi akses ke file pada asal kustom”](#) dan [the section called “Tambahkan header khusus ke permintaan asal”](#). Untuk contoh yang menggunakan header kustom untuk membatasi akses ke asal Application Load Balancer, lihat [the section called “Batasi akses ke Application Load Balancers”](#)
 - Jika custom origin memerlukan logika kontrol akses kustom, Anda dapat menggunakan Lambda @Edge untuk menerapkan logika tersebut, seperti yang dijelaskan dalam posting blog ini: [Melayani Konten Pribadi Menggunakan Amazon & CloudFront Lambda @Edge](#).

Bagaimana CloudFront memberikan konten

Setelah beberapa pengaturan awal, CloudFront bekerja sama dengan situs web atau aplikasi Anda dan mempercepat pengiriman konten Anda. Bagian ini menjelaskan cara CloudFront menayangkan konten Anda saat pemirsa memintanya.

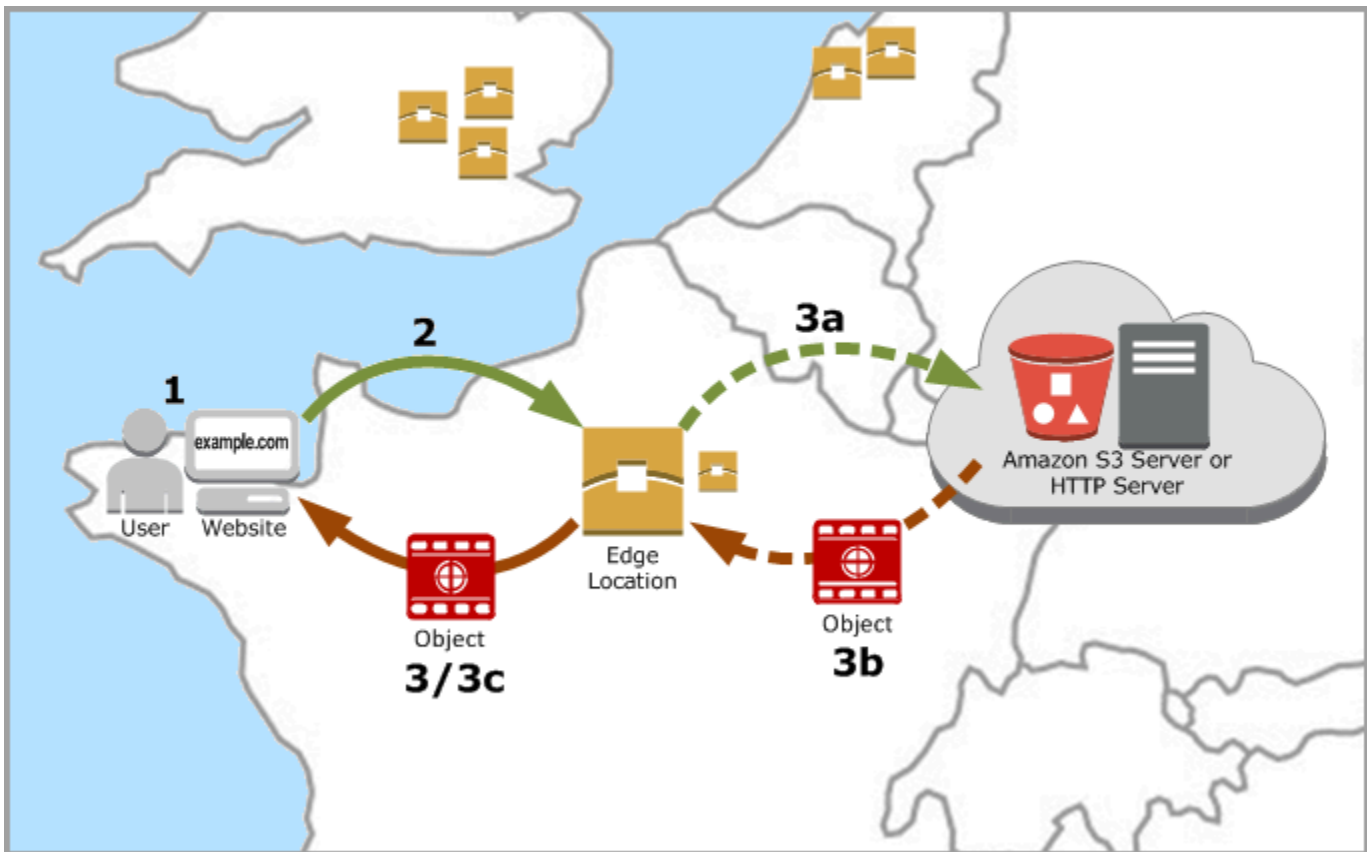
Topik

- [Cara CloudFront mengirimkan konten ke pengguna Anda](#)
- [Cara CloudFront bekerja dengan cache tepi regional](#)

Cara CloudFront mengirimkan konten ke pengguna Anda

Setelah Anda mengonfigurasi CloudFront untuk mengirimkan konten Anda, inilah yang terjadi ketika pengguna meminta objek Anda:

1. Pengguna mengakses situs web atau aplikasi Anda dan mengirimkan permintaan untuk objek, seperti file gambar atau file HTML.
2. DNS merutekan permintaan ke CloudFront POP (lokasi tepi) yang dapat melayani permintaan dengan baik, biasanya CloudFront POP terdekat dalam hal latensi.
3. CloudFront memeriksa cache untuk objek yang diminta. Jika objek dalam cache, CloudFront mengembalikannya ke pengguna. Jika objek tidak dalam cache, CloudFront lakukan hal berikut:
 - a. CloudFront membandingkan permintaan dengan spesifikasi dalam distribusi Anda dan meneruskan permintaan ke server asal Anda untuk objek yang sesuai—misalnya, ke bucket Amazon S3 atau server HTTP Anda.
 - b. Server asal mengirim objek kembali ke lokasi tepi.
 - c. Segera setelah byte pertama tiba dari asal, CloudFront mulai meneruskan objek ke pengguna. CloudFront juga menambahkan objek ke cache untuk lain kali seseorang memintanya.



Cara CloudFront bekerja dengan cache tepi regional

CloudFront titik kehadiran (juga dikenal sebagai POPs atau lokasi tepi) memastikan bahwa konten populer dapat disajikan dengan cepat kepada pemirsa Anda. CloudFront juga memiliki cache tepi regional yang membawa lebih banyak konten Anda lebih dekat ke pemirsa Anda, bahkan ketika konten tidak cukup populer untuk tetap di POP, untuk membantu meningkatkan kinerja konten tersebut.

Cache edge regional membantu semua jenis konten, terutama konten yang cenderung menjadi kurang populer seiring waktu. Contohnya meliputi konten yang dihasilkan pengguna, seperti video, foto, atau karya seni; aset e-commerce seperti foto dan video produk; serta berita dan konten terkait acara yang mungkin secara tiba-tiba menemukan popularitas baru.

Cara kerja cache regional

Cache tepi regional adalah CloudFront lokasi yang digunakan secara global, dekat dengan pemirsa Anda. Mereka terletak di antara server asal Anda dan POPs —lokasi edge global yang menayangkan konten langsung ke pemirsa. Ketika objek menjadi kurang populer, individu POPs mungkin menghapus benda-benda itu untuk memberi ruang bagi konten yang lebih populer. Cache

tepi regional memiliki cache yang lebih besar daripada POP individu, sehingga objek tetap berada di dalam cache lebih lama di lokasi cache tepi regional terdekat. Ini membantu menjaga lebih banyak konten Anda lebih dekat dengan pemirsa Anda, mengurangi kebutuhan CloudFront untuk kembali ke server asal Anda, dan meningkatkan kinerja keseluruhan untuk pemirsa.

Saat penonton membuat permintaan di situs web atau aplikasi Anda, DNS mengirimkan permintaan ke POP yang dapat melayani permintaan pengguna dengan sebaik-baiknya. Lokasi ini biasanya merupakan lokasi CloudFront tepi terdekat dalam hal latensi. Di POP, CloudFront memeriksa cache-nya untuk objek yang diminta. Jika objek dalam cache, CloudFront mengembalikannya ke pengguna. Jika objek tidak ada dalam cache, POP biasanya pergi ke cache tepi regional terdekat untuk mengambilnya. Untuk informasi selengkapnya tentang kapan POP melewati cache tepi regional dan langsung menuju ke asal, lihat catatan berikut.

Di lokasi cache tepi regional, periksa CloudFront lagi cache-nya untuk objek yang diminta. Jika objek ada di cache, CloudFront teruskan ke POP yang memintanya. Segera setelah byte pertama tiba dari lokasi cache tepi regional, CloudFront mulai meneruskan objek ke pengguna. CloudFront juga menambahkan objek ke cache di POP untuk lain kali seseorang memintanya.

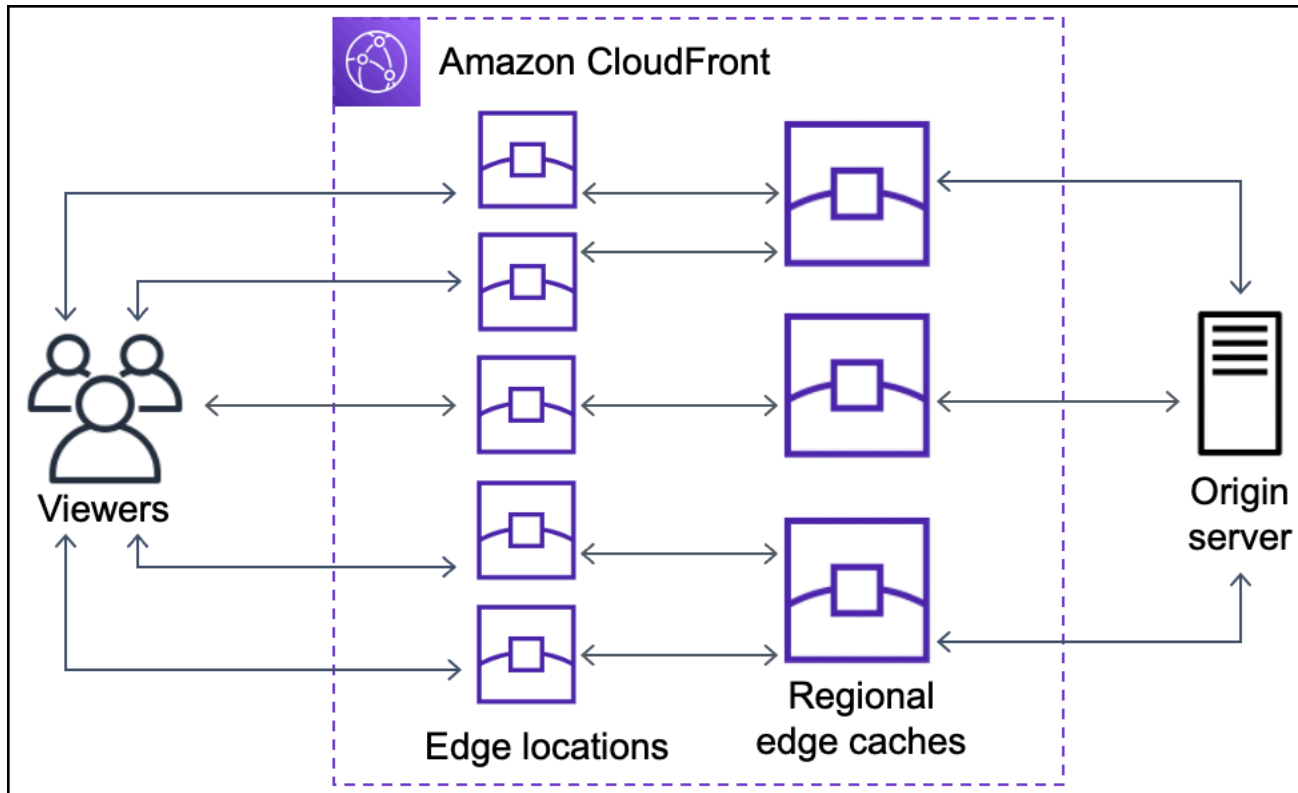
Untuk objek yang tidak di-cache di POP atau lokasi cache tepi regional, CloudFront bandingkan permintaan dengan spesifikasi di distribusi Anda dan teruskan permintaan ke server asal. Setelah server asal Anda mengirim objek kembali ke lokasi cache tepi regional, itu diteruskan ke POP, dan kemudian CloudFront meneruskannya ke pengguna. Dalam hal ini, tambahkan CloudFront juga objek ke cache di lokasi cache tepi regional selain POP untuk waktu berikutnya pemirsa memintanya. Ini memastikan bahwa semua POPs di wilayah berbagi cache lokal, menghilangkan beberapa permintaan ke server asal. CloudFront juga menjaga koneksi persisten dengan server asal sehingga objek diambil dari asal secepat mungkin.

Note

- Cache tepi regional memiliki paritas fitur dengan POPs. Misalnya, permintaan pembatalan cache menghapus objek baik dari cache POP maupun edge cache regional sebelum kedaluwarsa. Saat berikutnya penampil meminta objek, CloudFront kembali ke asal untuk mengambil versi terbaru dari objek.
- Metode HTTP proxy (PUTPOST, PATCH, OPTIONS, dan DELETE) langsung menuju ke asal dari POPs dan jangan proxy melalui cache tepi regional.
- Permintaan dinamis, sebagaimana ditentukan pada waktu permintaan, tidak mengalir melalui cache tepi regional, tetapi langsung menuju asal.

- Jika asalnya adalah bucket Amazon S3 dan cache tepi regional optimal permintaan Wilayah AWS sama dengan bucket S3, POP melewati cache tepi regional dan langsung menuju ke bucket S3.

Diagram berikut menggambarkan bagaimana permintaan dan tanggapan mengalir melalui lokasi CloudFront tepi dan cache tepi regional.



Lokasi dan rentang alamat IP server CloudFront edge

Untuk daftar lokasi server CloudFront edge, lihat halaman [Amazon CloudFront Global Edge Network](#).

Amazon Web Services (AWS) menerbitkan rentang alamat IP saat ini dalam format JSON. Untuk melihat rentang saat ini, unduh [ip-ranges.json](#). Untuk informasi selengkapnya, lihat [Rentang alamat IP AWS](#) di Referensi Umum Amazon Web Services.

Untuk menemukan rentang alamat IP yang terkait dengan server CloudFront edge, cari ip-ranges.json untuk string berikut:

```
"region": "GLOBAL",
```

```
"service": "CLOUDFRONT"
```

Atau, Anda hanya dapat melihat rentang CloudFront IP di <https://d7uri8nf7uskq.cloudfront.net/tools/list-cloudfront-ips>.

Gunakan daftar awalan CloudFront terkelola

Daftar awalan CloudFront terkelola berisi rentang alamat IP dari semua server yang menghadap CloudFront asal terdistribusi secara global. Jika asal Anda di-host AWS dan dilindungi oleh [grup keamanan](#) VPC Amazon, Anda dapat menggunakan daftar awalan CloudFront terkelola untuk mengizinkan lalu lintas masuk ke asal Anda hanya dari CloudFront server yang menghadap asal, mencegah CloudFront non-lalu lintas mencapai asal Anda. CloudFront mempertahankan daftar awalan terkelola sehingga selalu up to date dengan alamat IP dari semua server global yang CloudFront menghadap asal. Dengan daftar awalan CloudFront terkelola, Anda tidak perlu membaca atau memelihara daftar rentang alamat IP sendiri.

Misalnya, bayangkan asal Anda adalah instans Amazon EC2 di Wilayah Eropa (London) (eu-west-2). Jika instance ada di VPC, Anda dapat membuat aturan grup keamanan yang memungkinkan akses HTTPS masuk dari daftar awalan CloudFront terkelola. Ini memungkinkan semua server global yang menghadap ke asal untuk mencapai instance. CloudFront Jika Anda menghapus semua aturan masuk lainnya dari grup keamanan, Anda mencegah CloudFront non-lalu lintas mencapai instans.

Daftar awalan CloudFront terkelola adalah sebagai berikut:

- `com.amazonaws.global.cloudfront.origin-facing` (IPv4)
- `com.amazonaws.global.ipv6.cloudfront.origin-facing` (IPv6)

Untuk informasi selengkapnya, lihat [Menggunakan daftar awalan AWS-terkelola](#) di Panduan Pengguna Amazon VPC.

Important

Daftar awalan CloudFront terkelola unik dalam cara penerapannya pada kuota VPC Amazon. Untuk informasi selengkapnya, lihat [AWS bobot daftar awalan terkelola](#) di Panduan Pengguna Amazon VPC.

Menggunakan CloudFront dengan AWS SDK

AWS kit pengembangan perangkat lunak (SDKs) tersedia untuk banyak bahasa pemrograman populer. Setiap SDK menyediakan API, contoh kode, dan dokumentasi yang memudahkan developer untuk membangun aplikasi dalam bahasa pilihan mereka.

Dokumentasi SDK	Contoh kode
AWS SDK untuk C++	AWS SDK untuk C++ contoh kode
AWS CLI	AWS CLI contoh kode
AWS SDK untuk Go	AWS SDK untuk Go contoh kode
AWS SDK untuk Java	AWS SDK untuk Java contoh kode
AWS SDK untuk JavaScript	AWS SDK untuk JavaScript contoh kode
AWS SDK untuk Kotlin	AWS SDK untuk Kotlin contoh kode
AWS SDK untuk .NET	AWS SDK untuk .NET contoh kode
AWS SDK untuk PHP	AWS SDK untuk PHP contoh kode
Alat AWS untuk PowerShell	Alat AWS untuk PowerShell contoh kode
AWS SDK untuk Python (Boto3)	AWS SDK untuk Python (Boto3) contoh kode
AWS SDK untuk Ruby	AWS SDK untuk Ruby contoh kode
AWS SDK for Rust	AWS SDK for Rust contoh kode
AWS SDK for SAP ABAP	AWS SDK for SAP ABAP contoh kode
AWS SDK for Swift	AWS SDK for Swift contoh kode

 Ketersediaan contoh

Tidak dapat menemukan apa yang Anda butuhkan? Minta contoh kode menggunakan tautan Berikan umpan balik di bagian bawah halaman ini.

CloudFront sumber daya teknis

Gunakan sumber daya berikut untuk mendapatkan jawaban atas pertanyaan teknis tentang CloudFront:

- [AWS Re:post](#) — Situs tanya jawab berbasis komunitas bagi pengembang untuk mendiskusikan pertanyaan teknis yang terkait dengan CloudFront
- [Dukungan Pusat](#) — Situs ini mencakup informasi tentang kasus dukungan terbaru Anda dan hasil dari AWS Trusted Advisor dan pemeriksaan kesehatan. Ini juga menyediakan tautan ke forum diskusi, teknis FAQs, dasbor kesehatan layanan, dan informasi tentang Dukungan rencana.
- [AWS Dukungan Premium](#) — Pelajari tentang AWS Premium Support one-on-one, saluran dukungan respons cepat yang membantu Anda membangun dan menjalankan aplikasi. AWS
- [AWS IQ](#) — Dapatkan bantuan dari para profesional dan ahli AWS bersertifikat.

Memulai dengan CloudFront

Topik di bagian ini menunjukkan cara memulai mengirimkan konten Anda dengan Amazon CloudFront.

[Siapkan Akun AWS](#) Topik ini menjelaskan prasyarat untuk tutorial berikut, seperti membuat Akun AWS dan membuat pengguna dengan akses administratif.

Tutorial distribusi dasar menunjukkan cara mengatur kontrol akses asal (OAC) untuk mengirim permintaan yang diautentikasi ke asal Amazon S3.

Tutorial situs web statis aman menunjukkan cara membuat situs web statis yang aman untuk nama domain Anda menggunakan OAC dengan asal Amazon S3. Tutorial menggunakan template Amazon CloudFront (CloudFront) untuk konfigurasi dan penyebaran.

Topik

- [Siapkan Akun AWS](#)
- [Memulai dengan distribusi CloudFront standar](#)
- [Memulai dengan distribusi standar \(AWS CLI\)](#)
- [Memulai dengan situs web statis yang aman](#)

Siapkan Akun AWS

Topik ini menjelaskan langkah-langkah awal, seperti membuat Akun AWS, untuk mempersiapkan Anda menggunakan Amazon CloudFront.

Topik

- [Mendaftar untuk Akun AWS](#)
- [Pilih cara mengakses CloudFront](#)

Mendaftar untuk Akun AWS

Untuk memulai AWS, Anda membutuhkan Akun AWS. Untuk informasi tentang membuat Akun AWS, lihat [Memulai dengan Akun AWS](#) di Panduan AWS Account Management Referensi.

Pilih cara mengakses CloudFront

Anda dapat mengakses Amazon dengan CloudFront cara berikut:

- **Konsol Manajemen AWS**— Prosedur di seluruh panduan ini menjelaskan cara menggunakan Konsol Manajemen AWS untuk melakukan tugas.
- **AWS SDK** — Jika Anda menggunakan bahasa pemrograman yang AWS menyediakan SDK, Anda dapat menggunakan SDK untuk mengakses CloudFront SDK menyederhanakan otentikasi, mengintegrasikan dengan mudah dengan lingkungan pengembangan Anda, dan menyediakan akses ke perintah. CloudFront Untuk informasi selengkapnya, lihat [Menggunakan CloudFront dengan AWS SDK](#).
- **CloudFront API** — Jika Anda menggunakan bahasa pemrograman yang tidak tersedia untuk SDK, lihat [Referensi Amazon CloudFront API](#) untuk informasi tentang tindakan API dan tentang cara membuat permintaan API.
- **AWS CLI**— The AWS Command Line Interface (AWS CLI) adalah alat terpadu untuk mengelola Layanan AWS. Untuk informasi tentang cara menginstal dan mengonfigurasi AWS CLI, lihat [Menginstal atau memperbarui ke versi terbaru dari](#) Panduan AWS Command Line Interface Pengguna. AWS CLI
- **Tools untuk Windows PowerShell** — Jika Anda memiliki pengalaman dengan Windows PowerShell, Anda mungkin lebih suka menggunakannya AWS Tools for Windows PowerShell. Untuk informasi selengkapnya, lihat [Menginstal AWS Tools for Windows PowerShell](#) dalam Panduan Pengguna Alat AWS untuk PowerShell .

Memulai dengan distribusi CloudFront standar

Prosedur di bagian ini menunjukkan kepada Anda cara menggunakan CloudFront untuk menyiapkan distribusi standar yang melakukan hal berikut:

- Membuat bucket S3 untuk digunakan sebagai asal distribusi Anda.
- Menyimpan versi asli objek Anda di bucket Amazon Simple Storage Service (Amazon S3) Simple Storage Service (Amazon S3).
- Menggunakan kontrol akses asal (OAC) untuk mengirim permintaan yang diautentikasi ke asal Amazon S3 Anda. OAC mengirimkan permintaan CloudFront untuk mencegah pemirsa mengakses bucket S3 Anda secara langsung. Untuk informasi lebih lanjut tentang OAC, lihat [Batasi akses ke asal Amazon S3](#).

- Menggunakan nama CloudFront domain di URL untuk objek Anda (misalnya, `https://d111111abcdef8.cloudfront.net/index.html`).
- Menyimpan objek Anda di lokasi CloudFront tepi selama durasi default 24 jam (durasi minimum adalah 0 detik).

Sebagian besar ini dikonfigurasi secara otomatis untuk Anda saat Anda membuat CloudFront distribusi.

Topik

- [Prasyarat](#)
- [Buat bucket Amazon S3.](#)
- [Unggah konten ke ember](#)
- [Buat CloudFront distribusi yang menggunakan asal Amazon S3 dengan OAC](#)
- [Akses konten Anda melalui CloudFront](#)
- [Bersihkan](#)
- [Tingkatkan distribusi dasar Anda](#)

Prasyarat

Sebelum memulai, pastikan bahwa Anda telah menyelesaikan langkah-langkah dalam [Siapkan Akun AWS](#).

Buat bucket Amazon S3.

Bucket Amazon S3 adalah wadah untuk file (objek) atau folder. CloudFront dapat mendistribusikan hampir semua jenis file untuk Anda ketika ember S3 adalah sumbernya. Misalnya, CloudFront dapat mendistribusikan teks, gambar, dan video. Tidak ada maksimum untuk jumlah data yang dapat disimpan di Amazon S3.

Untuk tutorial ini, Anda membuat bucket S3 dengan `hello world` file sampel yang disediakan yang akan Anda gunakan untuk membuat halaman web dasar.

Untuk membuat ember

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon S3 di <https://console.aws.amazon.com/s3/>

2. Kami menyarankan Anda menggunakan sampel Hello World kami untuk Memulai ini. Unduh halaman web hello world: [hello-world-html.zip](#). Buka zip dan simpan css folder dan index file di lokasi yang nyaman, seperti desktop tempat Anda menjalankan browser.
3. Pilih Buat bucket.
4. Masukkan nama Bucket unik yang sesuai dengan [aturan penamaan bucket tujuan umum](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.
5. Untuk Wilayah, kami sarankan memilih Wilayah AWS yang secara geografis dekat dengan Anda. (Ini mengurangi latensi dan biaya.)
 - Memilih Wilayah yang berbeda juga berfungsi. Anda dapat melakukan ini untuk mengatasi persyaratan peraturan, misalnya.
6. Biarkan semua pengaturan lain pada defaultnya, lalu pilih Buat bucket.

Unggah konten ke ember

Setelah Anda membuat bucket Amazon S3, unggah konten file yang tidak di-zip ke dalamnya `hello world`. (Anda mengunduh dan membuka `hello world` file ini.) [Buat bucket Amazon S3](#).

Untuk mengunggah konten ke Amazon S3

1. Di bagian Bucket tujuan umum, pilih nama bucket baru Anda.
2. Pilih Unggah.
3. Pada halaman Unggah, seret css folder dan index file ke area drop.
4. Biarkan semua pengaturan lain pada defaultnya, lalu pilih Unggah.

Buat CloudFront distribusi yang menggunakan asal Amazon S3 dengan OAC

Untuk tutorial ini, Anda akan membuat CloudFront distribusi yang menggunakan asal Amazon S3 dengan kontrol akses asal (OAC). OAC membantu Anda mengirim permintaan yang diautentikasi dengan aman ke asal Amazon S3 Anda. Untuk informasi lebih lanjut tentang OAC, lihat [Batasi akses ke asal Amazon S3](#).

Untuk membuat CloudFront distribusi dengan asal Amazon S3 yang menggunakan OAC

1. Buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.

2. Pilih Buat Distribusi.
3. Masukkan nama Distribusi untuk distribusi standar. Nama akan muncul sebagai nilai untuk Name kunci sebagai tag. Anda dapat mengubah nilai ini nanti. Anda dapat menambahkan hingga 50 tag untuk distribusi standar Anda. Untuk informasi selengkapnya, lihat [Menandai CloudFront sumber daya](#).
4. Pilih situs web atau aplikasi tunggal, Berikutnya.
5. Pilih Berikutnya.
6. Untuk halaman tipe Origin, pilih Amazon S3.
7. Untuk asal S3, pilih Browse S3 dan pilih bucket S3 yang Anda buat untuk tutorial ini.
8. Untuk Pengaturan, pilih Gunakan pengaturan asal yang disarankan. CloudFront akan menggunakan pengaturan cache dan asal yang direkomendasikan default untuk asal Amazon S3 Anda, termasuk mengatur Origin Access Control (OAC). Untuk informasi selengkapnya tentang pengaturan yang disarankan, lihat [Referensi pengaturan distribusi yang telah dikonfigurasi sebelumnya](#).
9. Pilih Berikutnya.
10. Pada halaman Aktifkan perlindungan keamanan, pilih apakah akan mengaktifkan perlindungan AWS WAF keamanan.
11. Pilih Berikutnya.
12. Pilih Buat distribusi. CloudFront memperbarui kebijakan bucket S3 untuk Anda.
13. Tinjau bagian Detail untuk distribusi baru Anda. Ketika distribusi Anda selesai digunakan, bidang terakhir diubah dari Deploying ke tanggal dan waktu.
14. Catat nama domain yang CloudFront ditetapkan untuk distribusi Anda. Itu terlihat serupa dengan yang berikut ini: `d111111abcdef8.cloudfront.net`.

Sebelum menggunakan bucket distribusi dan S3 dari tutorial ini di lingkungan produksi, pastikan untuk mengonfigurasinya untuk memenuhi kebutuhan spesifik Anda. Untuk informasi tentang mengonfigurasi akses di lingkungan produksi, lihat [Konfigurasi akses aman dan batasi akses ke konten](#).

Akses konten Anda melalui CloudFront

Untuk mengakses konten Anda CloudFront, gabungkan nama domain untuk CloudFront distribusi Anda dengan halaman utama untuk konten Anda. (Anda mencatat nama domain distribusi Anda di [Buat CloudFront distribusi yang menggunakan asal Amazon S3 dengan OAC](#).)

- Nama domain distribusi Anda mungkin terlihat seperti ini: `d111111abcdef8.cloudfront.net`.
- Jalur ke halaman utama situs web biasanya `/index.html`.

Oleh karena itu, URL untuk mengakses konten Anda CloudFront mungkin terlihat seperti ini:

`https://d111111abcdef8.cloudfront.net/index.html`.

Jika Anda mengikuti langkah-langkah sebelumnya dan menggunakan halaman web hello world, Anda akan melihat halaman web yang bertuliskan Hello world! .

Saat mengunggah lebih banyak konten ke bucket S3 ini, Anda dapat mengakses konten CloudFront melalui menggabungkan nama domain CloudFront distribusi dengan jalur ke objek di bucket S3. Misalnya, jika Anda mengunggah file baru yang diberi nama `new-page.html` ke root bucket S3 Anda, URL akan terlihat seperti ini:

`https://d111111abcdef8.cloudfront.net/new-page.html`.

Bersihkan

Jika Anda membuat bucket distribusi dan S3 hanya sebagai latihan pembelajaran, hapuslah agar Anda tidak lagi dikenakan biaya. Hapus distribusi terlebih dahulu. Untuk informasi selengkapnya, lihat tautan berikut:

- [Menghapus sebuah distribusi](#)
- [Menghapus ember](#)

Tingkatkan distribusi dasar Anda

Tutorial Memulai ini menyediakan kerangka kerja minimal untuk membuat distribusi. Kami menyarankan Anda menjelajahi penyempurnaan berikut:

- Anda dapat menggunakan fitur konten CloudFront pribadi untuk membatasi akses ke konten di bucket Amazon S3. Untuk informasi selengkapnya tentang distribusi konten pribadi, lihat [Sajikan konten pribadi dengan URL yang ditandatangani dan cookie yang ditandatangani](#).
- Anda dapat mengonfigurasi CloudFront distribusi Anda untuk menggunakan nama domain kustom (misalnya, `www.example.com` bukannya `d111111abcdef8.cloudfront.net`). Untuk informasi selengkapnya, lihat [Gunakan URL khusus](#).

- Tutorial ini menggunakan asal Amazon S3 dengan kontrol akses asal (OAC). Namun, Anda tidak dapat menggunakan OAC jika asal Anda adalah bucket S3 yang dikonfigurasi sebagai titik akhir [situs web](#). Jika itu masalahnya, Anda harus mengatur bucket Anda CloudFront sebagai custom origin. Untuk informasi selengkapnya, lihat [Menggunakan bucket Amazon S3 yang dikonfigurasi sebagai titik akhir situs web](#). Untuk informasi lebih lanjut tentang OAC, lihat [Batasi akses ke asal Amazon S3](#).

Memulai dengan distribusi standar (AWS CLI)

Prosedur di bagian ini menunjukkan kepada Anda cara menggunakan AWS CLI with CloudFront untuk menyiapkan konfigurasi dasar yang melibatkan hal-hal berikut:

- Membuat bucket Amazon S3 untuk digunakan sebagai asal distribusi Anda.
- Menyimpan versi asli objek Anda di bucket S3.
- Menggunakan kontrol akses asal (OAC) untuk mengirim permintaan yang diautentikasi ke asal Amazon S3 Anda. OAC mengirimkan permintaan CloudFront untuk mencegah pemirsa mengakses bucket S3 Anda secara langsung. Untuk informasi lebih lanjut tentang OAC, lihat [Batasi akses ke asal Amazon S3](#).
- Menggunakan nama CloudFront domain URLs untuk objek Anda (misalnya, `https://d111111abcdef8.cloudfront.net/index.html`).
- Menjaga objek Anda di lokasi CloudFront tepi selama durasi default 24 jam (durasi minimum adalah 0 detik).

Sebagian besar opsi ini dapat disesuaikan. Untuk informasi tentang cara menyesuaikan opsi CloudFront distribusi, lihat [Buat distribusi](#).

Prasyarat

Sebelum memulai, pastikan bahwa Anda telah menyelesaikan langkah-langkah dalam [Siapkan Akun AWS](#).

Instal AWS CLI dan konfigurasikan dengan kredensial Anda. Untuk informasi selengkapnya, lihat [Memulai AWS CLI](#) dalam Panduan Pengguna AWS CLI .

Buat bucket Amazon S3.

Bucket Amazon S3 adalah wadah untuk file (objek) atau folder. CloudFront dapat mendistribusikan hampir semua jenis file untuk Anda ketika ember S3 adalah sumbernya. Misalnya, CloudFront dapat mendistribusikan teks, gambar, dan video. Tidak ada maksimum untuk jumlah data yang dapat disimpan di Amazon S3.

Untuk tutorial ini, Anda membuat bucket S3 dan mengunggah file HTML yang akan Anda gunakan untuk membuat halaman web dasar.

```
aws s3 mb s3://amzn-s3-demo-bucket/ --region us-east-1
```

Ganti *amzn-s3-demo-bucket* dengan nama bucket yang unik secara global. Untuk itu Wilayah AWS, kami sarankan memilih Wilayah yang secara geografis dekat dengan Anda. Ini mengurangi latensi dan biaya, tetapi memilih Wilayah yang berbeda juga berfungsi. Misalnya, Anda dapat melakukan ini untuk memenuhi persyaratan peraturan.

Unggah konten ke ember

Untuk tutorial ini, unduh dan ekstrak file konten sampel untuk halaman web “Hello World” dasar.

```
# Create a temporary directory
mkdir -p ~/cloudfront-demo

# Download the sample Hello World files
curl -o ~/cloudfront-demo/hello-world-html.zip https://docs.aws.amazon.com/
AmazonCloudFront/latest/DeveloperGuide/samples/hello-world-html.zip

# Extract the zip file
unzip ~/cloudfront-demo/hello-world-html.zip -d ~/cloudfront-demo/hello-world
```

Ini membuat direktori dengan `index.html` file dan `css` folder. Unggah file-file ini ke bucket S3 Anda.

```
aws s3 cp ~/cloudfront-demo/hello-world/ s3://amzn-s3-demo-bucket/ --recursive
```

Membuat Origin Access Control (OAC)

Untuk tutorial ini, Anda akan membuat kontrol akses asal (OAC). OAC membantu Anda mengirim permintaan yang diautentikasi dengan aman ke asal Amazon S3 Anda. Untuk informasi lebih lanjut tentang OAC, lihat [Batasi akses ke asal Amazon S3](#).

```
aws cloudfront create-origin-access-control \
  --origin-access-control-config Name="oac-for-
s3",SigningProtocol=sigv4,SigningBehavior=always,OriginAccessControlOriginType=s3
```

Simpan ID OAC dari output sebagai variabel lingkungan. Ganti nilai contoh dengan ID OAC Anda sendiri. Anda akan menggunakan ini di langkah berikutnya.

```
OAC_ID="E1ABCD2EFGHIJ"
```

Buat distribusi standar

Buat file konfigurasi distribusi bernama `distribution-config.json`. Ganti contoh nama bucket dengan nama bucket Anda untuk `Id`, `DomainName`, dan `TargetOriginId` nilai.

```
cat > distribution-config.json << EOF
{
  "CallerReference": "cli-example-$(date +%s)",
  "Origins": {
    "Quantity": 1,
    "Items": [
      {
        "Id": "S3-amzn-s3-demo-bucket",
        "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
        "S3OriginConfig": {
          "OriginAccessIdentity": ""
        },
        "OriginAccessControlId": "$OAC_ID"
      }
    ]
  },
  "DefaultCacheBehavior": {
    "TargetOriginId": "S3-amzn-s3-demo-bucket",
    "ViewerProtocolPolicy": "redirect-to-https",
    "AllowedMethods": {
```

```
    "Quantity": 2,
    "Items": ["GET", "HEAD"],
    "CachedMethods": {
        "Quantity": 2,
        "Items": ["GET", "HEAD"]
    }
},
"DefaultTTL": 86400,
"MinTTL": 0,
"MaxTTL": 31536000,
"Compress": true,
"ForwardedValues": {
    "QueryString": false,
    "Cookies": {
        "Forward": "none"
    }
}
},
"Comment": "CloudFront distribution for S3 bucket",
"Enabled": true
}
EOF
```

Buat distribusi standar.

```
aws cloudfront create-distribution --distribution-config file://distribution-
config.json
```

Simpan ID distribusi dan nama domain dari output sebagai variabel lingkungan. Ganti Contoh nilai dengan nilai Anda sendiri. Anda akan menggunakan ini nanti dalam tutorial ini.

```
DISTRIBUTION_ID="EABCD1234XMPL"
DOMAIN_NAME="d111111abcdef8.cloudfront.net"
```

Sebelum menggunakan bucket distribusi dan S3 dari tutorial ini di lingkungan produksi, pastikan untuk mengonfigurasinya untuk memenuhi kebutuhan spesifik Anda. Untuk informasi tentang mengonfigurasi akses di lingkungan produksi, lihat [Konfigurasi akses aman dan batasi akses ke konten](#).

Perbarui kebijakan bucket S3 Anda

Perbarui kebijakan bucket S3 Anda CloudFront agar dapat mengakses objek. Ganti contoh nama bucket dengan nama bucket Anda.

```
# Get your AWS account ID
ACCOUNT_ID=$(aws sts get-caller-identity --query 'Account' --output text)

# Create the bucket policy
cat > bucket-policy.json << EOF
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCloudFrontServicePrincipal",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudfront.amazonaws.com"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
      "Condition": {
        "StringEquals": {
          "AWS:SourceArn": "arn:aws:cloudfront::$ACCOUNT_ID:distribution/
$DISTRIBUTION_ID"
        }
      }
    }
  ]
}
EOF

# Apply the bucket policy
aws s3api put-bucket-policy \
  --bucket amzn-s3-demo-bucket \
  --policy file://bucket-policy.json
```

Konfirmasikan penyebaran distribusi

Setelah Anda membuat distribusi Anda, itu akan memakan waktu untuk menyelesaikan penerapan. Ketika status distribusi berubah dari InProgress ke Deployed, lanjutkan ke langkah berikutnya.

```
aws cloudfront get-distribution --id $DISTRIBUTION_ID --query 'Distribution.Status'
```

Atau, Anda dapat menggunakan wait perintah untuk menunggu penyebaran distribusi.

```
aws cloudfront wait distribution-deployed --id $DISTRIBUTION_ID
```

Akses konten Anda melalui CloudFront

Untuk mengakses konten Anda CloudFront, gabungkan nama domain untuk CloudFront distribusi Anda dengan halaman utama untuk konten Anda. Ganti contoh nama CloudFront domain dengan nama Anda sendiri.

```
https://d1111111abcdef8.cloudfront.net/index.html
```

Jika Anda mengikuti langkah-langkah sebelumnya dan membuat file HTML, Anda akan melihat halaman web yang mengatakan Hello world! .

Saat mengunggah lebih banyak konten ke bucket S3 ini, Anda dapat mengakses konten CloudFront melalui menggabungkan nama domain CloudFront distribusi dengan jalur ke objek di bucket S3. Misalnya, jika Anda mengunggah file baru yang diberi nama new-page.html ke root bucket S3 Anda, URL akan terlihat seperti ini:

```
https://d1111111abcdef8.cloudfront.net/new-page.html.
```

Bersihkan

Jika Anda membuat bucket distribusi dan S3 hanya sebagai latihan pembelajaran, hapuslah agar Anda tidak lagi dikenakan biaya. Nonaktifkan dan hapus distribusi terlebih dahulu.

Untuk menonaktifkan dan menghapus distribusi standar (AWS CLI)

1. Pertama, nonaktifkan distribusi.

```
# Get the current configuration and ETag
ETAG=$(aws cloudfront get-distribution-config --id $DISTRIBUTION_ID --query 'ETag'
--output text)

# Create a modified configuration with Enabled=false
aws cloudfront get-distribution-config --id $DISTRIBUTION_ID | \
jq '.DistributionConfig.Enabled = false' > temp_disabled_config.json
```



```
# Update the distribution to disable it
aws cloudfront update-distribution \
  --id $DISTRIBUTION_ID \
  --distribution-config file://<(jq '.DistributionConfig'
temp_disabled_config.json) \
  --if-match $ETAG
```

2. Tunggu distribusi dinonaktifkan.

```
aws cloudfront wait distribution-deployed --id $DISTRIBUTION_ID
```

3. Hapus distribusi.

```
# Get the current ETag
ETAG=$(aws cloudfront get-distribution-config --id $DISTRIBUTION_ID --query 'ETag'
--output text)

# Delete the distribution
aws cloudfront delete-distribution --id $DISTRIBUTION_ID --if-match $ETAG
```

Untuk menghapus bucket S3 (AWS CLI)

- Hapus ember S3 dan isinya. Ganti contoh nama bucket dengan nama Anda sendiri.

```
# Delete the bucket contents
aws s3 rm s3://amzn-s3-demo-bucket --recursive

# Delete the bucket
aws s3 rb s3://amzn-s3-demo-bucket
```

Untuk membersihkan file lokal yang dibuat untuk tutorial ini, jalankan perintah berikut:

```
# Clean up local files
rm -f distribution-config.json bucket-policy.json temp_disabled_config.json
rm -rf ~/cloudfront-demo
```

Secara opsional, Anda dapat menghapus OAC yang Anda buat untuk tutorial ini.

```
# Get the OAC ETag
```

```
OAC_ETAG=$(aws cloudfront get-origin-access-control --id $OAC_ID --query 'ETag' --
output text)

# Delete the OAC
aws cloudfront delete-origin-access-control --id $OAC_ID --if-match $OAC_ETAG
```

Memulai dengan situs web statis yang aman

Anda dapat memulai Amazon CloudFront dengan menggunakan solusi yang dijelaskan dalam topik ini untuk membuat situs web statis yang aman untuk nama domain Anda. Situs web statis hanya menggunakan file statis — seperti HTML, CSS, gambar JavaScript, dan video — dan tidak memerlukan server atau pemrosesan sisi server. Dengan solusi ini, situs web Anda akan mendapatkan manfaat sebagai berikut:

- Menggunakan penyimpanan yang tahan lama [Amazon Simple Storage Service \(Amazon S3\)](#) – Solusi ini menciptakan bucket Amazon S3 untuk me-host konten situs web statis Anda. Untuk memperbarui situs web Anda, cukup unggah file baru Anda ke bucket S3.
- Dipercepat oleh jaringan pengiriman CloudFront konten Amazon — Solusi ini menciptakan CloudFront distribusi untuk melayani situs web Anda kepada pemirsa dengan latensi rendah. Distribusi dikonfigurasi dengan [kontrol akses asal](#) (OAC) untuk memastikan bahwa situs web hanya dapat diakses melalui CloudFront, tidak langsung dari S3.
- Diamankan oleh HTTPS dan header keamanan — Solusi ini membuat SSL/TLS sertifikat di [AWS Certificate Manager \(ACM\)](#), dan menempelkannya ke distribusi. CloudFront Sertifikat ini memungkinkan distribusi untuk melayani situs web domain Anda secara aman dengan HTTPS.
- Dikonfigurasi dan digunakan dengan [AWS CloudFormation](#)— Solusi ini menggunakan CloudFormation template untuk menyiapkan semua komponen, sehingga Anda dapat lebih fokus pada konten situs web Anda dan lebih sedikit pada konfigurasi komponen.

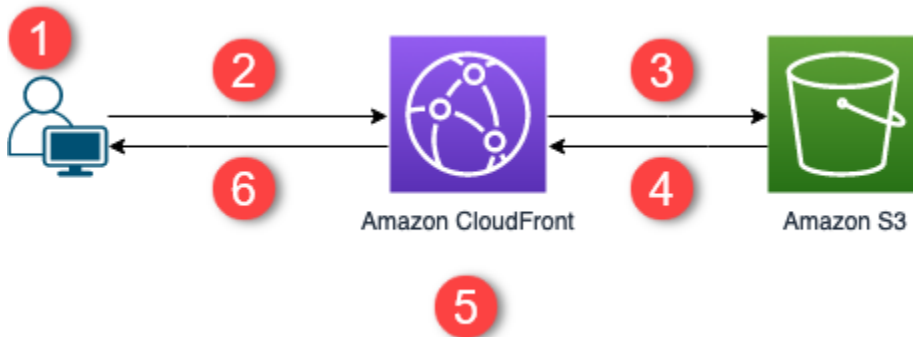
Solusi ini bersifat open source GitHub. Untuk melihat kode, mengirim permintaan penarikan, atau membuka masalah, kunjungi <https://github.com/aws-samples/amazon-cloudfront-secure-static-site>.

Topik

- [Ikhtisar solusi](#)
- [Terapkan solusinya](#)

Ikhtisar solusi

Diagram berikut menunjukkan ikhtisar tentang cara kerja solusi situs web statis ini:



1. Penampil meminta situs web di `www.example.com`.
2. Jika objek yang diminta di-cache, CloudFront mengembalikan objek dari cache ke penampil.
3. Jika objek tidak dalam CloudFront cache, CloudFront meminta objek dari asal (ember S3).
4. S3 mengembalikan objek ke CloudFront.
5. CloudFront cache objek.
6. Objek dikembalikan ke pemirsa. Permintaan selanjutnya untuk objek yang datang ke lokasi CloudFront tepi yang sama disajikan dari CloudFront cache.

Terapkan solusinya

Untuk menyebar solusi situs web statis yang aman ini, Anda dapat memilih salah satu opsi berikut:

- Gunakan CloudFormation konsol untuk menerapkan solusi dengan konten default, lalu unggah konten situs web Anda ke Amazon S3.
- Gandakan solusi ke komputer Anda untuk menambahkan konten situs web Anda. Kemudian, sebar solusi dengan AWS Command Line Interface (AWS CLI).

Note

Anda harus menggunakan Wilayah AS Timur (Virginia N.) untuk menyebarkan template. CloudFormation

Topik

- [Prasyarat](#)
- [Gunakan CloudFormation konsol](#)
- [Kloning solusinya secara lokal](#)
- [Menemukan log akses](#)

Prasyarat

Untuk menggunakan solusi ini, Anda harus memiliki prasyarat berikut:

- Nama domain terdaftar, seperti contoh.com, yang mengarah ke zona hosting Amazon Route 53. Zona yang dihosting harus berada di Akun AWS tempat yang sama di mana Anda menerapkan solusi ini. Jika Anda tidak memiliki nama domain terdaftar, Anda dapat [daftarkan satu dengan Route 53](#). Jika Anda memiliki nama domain terdaftar tetapi tidak menunjuk ke zona hosting Route 53, [mengonfigurasi Route 53 sebagai layanan DNS Anda](#).
- AWS Identity and Access Management (IAM) izin untuk meluncurkan CloudFormation template yang membuat peran IAM, dan izin untuk membuat semua AWS sumber daya dalam solusi. Untuk informasi selengkapnya, lihat [Mengontrol akses dengan AWS Identity and Access Management](#) dalam Panduan Pengguna AWS CloudFormation .

Anda bertanggung jawab atas biaya yang timbul saat menggunakan solusi ini. Untuk informasi lebih lanjut tentang biaya, lihat [halaman harga untuk masing-masing Layanan AWS](#).

Gunakan CloudFormation konsol

Untuk menyebarkan menggunakan konsol CloudFormation

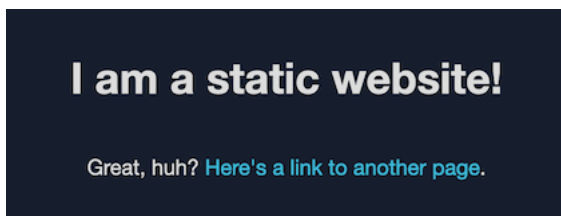
1. [Luncurkan solusi ini di CloudFormation konsol](#). Jika perlu, masuk ke Anda Akun AWS.
2. Wizard Buat tumpukan terbuka di CloudFormation konsol, dengan bidang yang telah diisi sebelumnya yang menentukan templat solusi ini. CloudFormation

Di bagian bawah halaman, pilih Selanjutnya.

3. Di Tentukan detail tumpukan , masukkan nilai untuk kolom berikut:
 - SubDomain – Masukkan subdomain yang akan digunakan untuk situs web Anda. Misalnya, jika subdomain adalah www, situs web Anda tersedia di **www.example.com**. (Ganti example.com dengan nama domain Anda, seperti yang dijelaskan dalam bullet berikut.)

- **DomainName**— Masukkan nama domain Anda, seperti *example.com*. Domain ini harus diarahkan ke zona yang di-hosting Route 53.
 - **HostedZoneId**— ID zona yang dihosting Route 53 dari nama domain Anda.
 - **CreateApex**— (Opsional) Buat alias ke puncak domain (example.com) dalam konfigurasi Anda. CloudFront
4. Setelah selesai, pilih Selanjutnya.
 5. (Opsional) Pada Mengonfigurasi opsi tumpukan yang sangat penting, [tambahkan tag dan opsi tumpukan lainnya](#).
 6. Setelah selesai, pilih Selanjutnya.
 7. Di Peninjauan , gulir ke bagian bawah halaman, kemudian pilih dua kotak di Kemampuan bagian. Kemampuan ini memungkinkan CloudFormation untuk membuat peran IAM yang memungkinkan akses ke sumber daya tumpukan, dan memberi nama sumber daya secara dinamis.
 8. Pilih Membuat tumpukan.
 9. Tunggu tumpukan selesai membuat. Tumpukan ini menciptakan tumpukan yang terbentuk dan dapat memakan waktu beberapa menit untuk selesai. Setelah selesai, Status perubahan pada BUAT_SEMBANG.

Ketika statusnya CREATE_COMPLETE, buka <https://www.example.com> untuk melihat situs web Anda (ganti www.example.com dengan subdomain dan nama domain yang Anda tentukan pada langkah 3). Anda akan melihat konten default situs web:



Untuk mengganti konten default situs web dengan konten Anda sendiri

1. Buka konsol Amazon S3 di <https://console.aws.amazon.com/s3/>
2. Pilih ember yang namanya dimulai dengan amazon-cloudfront-secure-static-site-s3bucketroot -.

Note

Pastikan untuk memilih bucket dengan s3bucketroot atas namanya, bukan s3bucketlog. Ember dengan s3bucketroot dalam namanya berisi konten situs web. Satu dengan s3bucketlog hanya berisi file log.

3. Hapus konten default situs web, lalu unggah konten Anda sendiri.

Note

Jika Anda melihat situs web Anda dengan konten default solusi ini, kemungkinan beberapa konten default di-cache di lokasi CloudFront tepi. Untuk memastikan bahwa pemirsa melihat konten situs web Anda yang diperbarui, batalkan file untuk menghapus salinan yang di-cache dari CloudFront lokasi tepi. Untuk informasi selengkapnya, lihat [Membatalkan file untuk menghapus konten](#).

Kloning solusinya secara lokal

Prasyarat

Untuk menambahkan konten situs web Anda sebelum menerapkan solusi ini, Anda harus mengemas artefak solusi secara lokal, yang memerlukan Node.js dan npm. Untuk informasi selengkapnya, lihat <https://www.npmjs.com/get-npm>.

Untuk menambahkan konten situs web dan menerapkan solusi

1. Gandakan atau unduh solusi dari <https://github.com/aws-samples/amazon-cloudfront-secure-static-site>. Setelah Anda mengkloning atau mengunduhnya, buka perintah perintah atau terminal dan navigasi ke amazon-cloudfront-secure-static-site folder.
2. Jalankan perintah berikut untuk memasang dan mengemas artefak solusi:

```
make package-static
```

3. Salin konten situs web Anda ke www , menimpa konten situs web default.
4. Jalankan AWS CLI perintah berikut untuk membuat bucket Amazon S3 untuk menyimpan artefak solusi. Ganti *amzn-s3-demo-bucket-for-artifacts* dengan nama bucket Anda sendiri.

```
aws s3 mb s3://amzn-s3-demo-bucket-for-artifacts --region us-east-1
```

5. Jalankan AWS CLI perintah berikut untuk mengemas artefak solusi sebagai CloudFormation template. Ganti *amzn-s3-demo-bucket-for-artifacts* dengan nama bucket yang Anda buat pada langkah sebelumnya.

```
aws cloudformation package \  
  --region us-east-1 \  
  --template-file templates/main.yaml \  
  --s3-bucket amzn-s3-demo-bucket-for-artifacts \  
  --output-template-file packaged.template
```

6. Jalankan perintah berikut untuk menyebarkan solusi dengan CloudFormation, mengganti nilai-nilai berikut:

- *your-CloudFormation-stack-name*— Ganti dengan nama untuk CloudFormation tumpukan.
- *example.com*— Ganti dengan nama domain Anda. Domain ini harus diarahkan ke zona yang dihosting Route 53 di tempat yang sama Akun AWS.
- *www*— Ganti dengan subdomain yang akan digunakan untuk situs web Anda. Misalnya, jika subdomain *www*, situs web Anda tersedia di *www.example.com*.
- *hosted-zone-ID*— Ganti dengan ID zona yang dihosting Route 53 dari nama domain Anda.

```
aws cloudformation deploy \  
  --region us-east-1 \  
  --stack-name your-CloudFormation-stack-name \  
  --template-file packaged.template \  
  --capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND \  
  --parameter-overrides DomainName=example.com SubDomain=www HostedZoneId=hosted-  
zone-ID
```

- (Opsional) Untuk menyebarkan tumpukan dengan apex domain, jalankan perintah berikut sebagai gantinya.

```
aws --region us-east-1 cloudformation deploy \  
  --stack-name your-CloudFormation-stack-name \  
  --template-file packaged.template \  
  --capabilities CAPABILITY_NAMED_IAM CAPABILITY_AUTO_EXPAND \  
  --parameter-overrides DomainName=example.com
```

```
--parameter-overrides DomainName=example.com SubDomain=www  
HostedZoneId=hosted-zone-ID CreateApex=yes
```

7. Tunggu CloudFormation tumpukan selesai dibuat. Tumpukan ini menciptakan tumpukan yang terbentuk dan dapat memakan waktu beberapa menit untuk selesai. Setelah selesai, Status perubahan pada BUAT_SEMBANG.

Ketika status berubah menjadi CREATE_COMPLETE, buka <https://www.example.com> untuk melihat situs web Anda (ganti www.example.com dengan subdomain dan nama domain yang Anda tentukan pada langkah sebelumnya). Anda akan melihat konten situs web Anda.

Menemukan log akses

Solusi ini memungkinkan [log akses](#) untuk CloudFront distribusi. Selesaikan langkah-langkah berikut untuk menemukan log akses distribusi.

Untuk menemukan log akses distribusi

1. Buka konsol Amazon S3 di <https://console.aws.amazon.com/s3/>
2. Pilih ember yang namanya dimulai dengan `amazon-cloudfront-secure-static-site-s3bucketlogs` -.

Note

Pastikan untuk memilih buket dengan `s3bucketlog` atas namanya, bukan `s3bucketroot`. Ember dengan `s3bucketlog` dalam namanya mengandung file log. Satu dengan `s3bucketroot` berisi konten situs web.

3. Folder bernama `cdn` berisi log CloudFront akses.

CloudFront paket harga flat-rate

CloudFront Paket harga flat-rate menggabungkan jaringan pengiriman konten CloudFront global Amazon (CDN) dengan beberapa fitur Layanan AWS dan menjadi harga bulanan tanpa biaya kelebihan, terlepas dari lonjakan atau serangan lalu lintas.

Flat-rate paket harga mencakup fitur-fitur berikut untuk harga bulanan sederhana:

- CloudFront CDN
- AWS WAF dan perlindungan DDoS
- Manajemen dan analitik bot
- Amazon Route 53 DNS
- Amazon CloudWatch Logs konsumsi
- Sertifikat TLS
- Perhitungan tepi tanpa server
- Kredit penyimpanan Amazon S3 setiap bulan

Paket tersedia dalam tingkatan Gratis, Pro, Bisnis, dan Premium agar sesuai dengan kebutuhan aplikasi Anda. Paket tidak memerlukan komitmen tahunan untuk mendapatkan harga terbaik yang tersedia. Mulailah dengan paket Gratis dan tingkatkan untuk mengakses lebih banyak kemampuan dan tunjangan penggunaan yang lebih besar.

Topik

- [Manfaat paket harga CloudFront flat-rate](#)
- [Fitur berdasarkan tingkat paket harga](#)
- [Tunjangan penggunaan bulanan](#)
- [Biaya yang ditanggung oleh paket Anda](#)
- [Kurangi secara keseluruhan AWS biaya dengan paket harga](#)
- [Kelola paket harga flat-rate Anda](#)
- [Izin](#)
- [Flat-rate kuota rencana harga](#)
- [Fitur yang tidak didukung](#)

Manfaat paket harga CloudFront flat-rate

Rencana CloudFront penetapan harga memberikan beberapa manfaat utama:

- Layanan dan harga terkonsolidasi

Gabungkan beberapa Layanan AWS dan fitur ke dalam satu paket untuk satu tarif tetap. Dirancang untuk menghilangkan pembelian layanan terpisah dan perhitungan harga di muka.

- Tidak ada kelebihan

Tidak ada biaya kelebihan terlepas dari lonjakan lalu lintas atau serangan.

- Hapus tunjangan penggunaan

Setiap paket mencakup tunjangan penggunaan yang dipublikasikan yang dirancang untuk kinerja optimal pada tingkat tersebut. Pantau penggunaan Anda, terima pemberitahuan proaktif, dan tingkatkan berdasarkan kebutuhan aplikasi Anda, tanpa komitmen jangka panjang.

- Melindungi dari serangan DDoS

CloudFront dan AWS WAF menyerap dan memblokir serangan sebelum mereka mencapai infrastruktur Anda. Cadangan komputasi, database, dan pemanfaatan infrastruktur Anda hanya untuk lalu lintas yang sah. Serangan dan permintaan DDoS yang diblokir diblokir dengan AWS WAF tidak pernah dihitung terhadap tunjangan penggunaan Anda.

- Kurangi AWS biaya keseluruhan Anda

Transfer data dari AWS aplikasi yang berjalan pada layanan seperti Amazon S3, AWS Application Load Balancer (ALB), atau Amazon API Gateway CloudFront terus gratis. Ketika Anda melayani AWS aplikasi Anda melalui CloudFront alih-alih langsung ke internet, paket tarif tetap Anda mencakup biaya transfer data antara aplikasi Anda dan pemirsa Anda dengan harga bulanan sederhana tanpa khawatir kelebihan. Lebih sedikit permintaan yang mencapai asal Anda juga mengurangi biaya Anda pada layanan yang mengenakan biaya berdasarkan penggunaan.

Fitur berdasarkan tingkat paket harga

Setiap paket harga mencakup satu CloudFront distribusi dengan hingga satu domain puncak (root) yang menggabungkan fitur dan layanan penting menjadi satu harga bulanan. Setiap paket juga mencakup kredit penyimpanan S3 tambahan.

Paket pada tingkatan yang lebih tinggi mencakup semua fitur dari paket tingkat yang lebih rendah serta fitur tambahan.

- Gratis — Untuk penggemar, pelajar, dan pengembang yang memulai.
- Pro — Luncurkan dan kembangkan situs web, blog, dan aplikasi kecil.
- Bisnis — Melindungi dan mempercepat aplikasi bisnis.
- Premium - Skala dan lindungi aplikasi bisnis dan misi penting.

Pilih tingkat paket yang menyertakan fitur dan konfigurasi yang Anda perlukan untuk aplikasi Anda. Lihat fitur berikut per paket harga.

Fitur paket harga

Tabel berikut menunjukkan fitur CloudFront, AWS WAF dan DDoS, Amazon Route 53, Amazon CloudWatch, dan Amazon S3 yang disertakan dalam setiap tingkat paket harga.

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
CDN Global	Ya	Ya	Ya	Ya
Gunakan CloudFront 750+ lokasi tepi global sebagai titik masuk tunggal yang besar dan terdistribusi untuk aplikasi web Anda. Mempercepat aplikasi statis, dinamis, dan tidak dapat di-cache.				
	Ya	Ya	Ya	Ya

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Caching konten	✔	✔	✔	✔
Simpan salinan konten Anda di CloudFront 750+ lokasi edge di seluruh dunia, kirimkan ke pengguna dari lokasi terdekat. Mengurangi waktu muat, melindungi aplikasi Anda dari lonjakan lalu lintas, dan menghemat biaya dengan menyajikan permintaan berulang secara lokal, bukan dari server aplikasi Anda.				
Pembatalan cache cepat	Ya	Ya	Ya	Ya
Hapus atau perbarui konten yang di-cache di semua lokasi tepi dalam hitungan detik.				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Pembatalan oleh tag cache		Ya	Ya	Ya
Tetapkan tag khusus ke konten Anda dan hapus semua konten cache yang berbagi tag tertentu, tanpa melacak jalur file individual.				
Perutean cerdas	Ya	Ya	Ya	Ya
Secara cerdas mengarahkan pengguna ke lokasi tepi optimal menggunakan data jaringan real-time, dan terhubung ke AWS asal Anda melalui jaringan AWS pribadi untuk kinerja yang lebih baik.				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Caching berjenjang	Ya	Ya	Ya	Ya
Cache edge regional berada di antara lokasi tepi dan aplikasi Anda untuk menyimpan konten lebih lama, mengurangi beban pada aplikasi Anda dan mempertahankan pengiriman cepat.				
Aturan caching default	Ya	Ya	Ya	Ya
Membuat keputusan caching yang efektif untuk cache sebagian besar aplikasi web tanpa konfigurasi khusus.				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium

Aturan caching kustom			Ya	Ya



[Kontrol cara CloudFront menyimpan konten dengan menentukan nilai permintaan mana yang akan digunakan, mengoptimalkan kinerja, personalisasi, dan kebutuhan kesegaran aplikasi Anda menggunakan kebijakan cache.](#)

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
High-speed perutean asal				Ya
Dengan Origin Shield , permintaan dinamis dialihkan dari lokasi tepi ke lokasi asal Anda menggunakan jaringan CloudFront pribadi untuk jalur berkinerja tinggi ke asal Anda.				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Pengurangan beban asal				Ya
Menambahkan lapisan caching tambahan yang dekat dengan aplikasi web Anda menggunakan Origin Shield . Origin Shield menggabungkan permintaan dari semua lokasi tepi, mengurangi beban pada aplikasi Anda terutama selama lonjakan lalu lintas.				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Failover asal otomatis				Ya
Secara otomatis merutekan lalu lintas ke asal cadangan jika asal utama Anda gagal, menjaga ketersediaan tinggi tanpa mengganggu pengguna.				
Aturan permintaan asal default	Ya	Ya	Ya	Ya
Kontrol informasi mana dari permintaan penampil yang secara otomatis disertakan dalam permintaan ke asal Anda, menggunakan kebijakan permintaan asal AWS terkelola yang dioptimalkan untuk skenario umum.				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Aturan header respons default	Ya	Ya	Ya	Ya
Gunakan kebijakan header respons AWS terkelola untuk menambah atau menghapus header HTTP dalam respons terhadap pemirsa, yang telah dikonfigurasi sebelumnya untuk header keamanan umum, setelan CORS, dan kasus penggunaan standar lainnya.				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Aturan permintaan asal kustom			Ya	Ya
Buat kebijakan permintaan asal Anda sendiri untuk menentukan dengan tepat string kueri URL, header, dan cookie mana yang diteruskan ke asal Anda, memungkinkan analisis kustom dan penanganan permintaan.				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Aturan header respon kustom			Ya	Ya
Buat kebijakan header respons Anda sendiri untuk mengontrol dengan tepat header HTTP mana yang CloudFront ditambahkan atau dihapus dalam tanggapan terhadap pemirsa, seperti header keamanan, Kebijakan Keamanan Konten (CSP), pengaturan CORS, dan header aplikasi kustom.				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Jumlah perilaku cache	5	10	50	100
Konfigurasi perilaku cache untuk mengontrol cara CloudFront menangani permintaan untuk pola URL tertentu, termasuk asal mana yang menyajikan konten, cara konten di-cache, dan apakah HTTPS atau URL yang ditandatangani diperlukan.				
Keamanan dan Perlindungan				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Always-on	Ya	Ya	Ya	Ya
Perlindungan DDoS	✔	✔	✔	✔
Lindungi dari serangan DDoS yang menargetkan situs web atau aplikasi Anda.				
Perlindungan DDoS Tingkat Lanjut			Ya	Ya
Identifikasi dan blokir serangan DDoS dalam hitungan detik menggunakan AntidDoS AMR. AWS mempelajari pola aplikasi unik Anda untuk membedakan antara serangan dan lonjakan alami dari pengguna yang sah.			✔	✔

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Firewall Aplikasi Web (WAF)	Ya	Ya	Ya	Ya
Lindungi dari kerentanan aplikasi umum dan potensi ancaman berdasarkan intelijen ancaman internal Amazon. Permintaan diblokir sebelum mencapai server Anda.				
Jumlah aturan WAF	5	25	50	75
Jumlah total aturan keamanan yang dapat Anda buat dan aktifkan dalam konfigurasi WAF, termasuk aturan kustom dan Aturan AWS Terkelola.				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Perlindungan untuk WordPress, PHP, dan database SQL		Ya	Ya	Ya
Use-case aturan keamanan berbasis untuk melindungi aplikasi umum dan sistem operasi seperti WordPress, PHP, database SQL, Linux, dan Windows.				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Analisis lalu lintas AI		Ya	Ya	Ya
Pantau pola akses, volume permintaan, dan jalur populer bot AI yang berinteraksi dengan konten Anda. Bot AI diklasifikasikan berdasarkan niat untuk mendeteksi aktivitas seperti pengikisan yang berpotensi tidak sah yang digunakan untuk pelatihan.				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
IP-based	Ya	Ya	Ya	Ya
membatasi tingkat				
Secara otomatis memblokir alamat IP yang melebihi jumlah permintaan yang dapat dikonfigurasi selama periode 5 menit, melindungi i terhadap serangan banjir HTTP dan upaya Denial of Service (DoS).				
Pemblokiran lalu lintas geografis	Ya	Ya	Ya	Ya
				
Blokir permintaan dari negara atau wilayah tertentu.				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Tantangan CAPTCHA		Ya	Ya	Ya
Memerlukan permintaan yang cocok dengan aturan keamanan tertentu untuk memecahkan teka-teki CAPTCHA untuk membuktikan bahwa manusia mengirim permintaan.				
Header-based penyaringan ancaman		Ya	Ya	Ya
Buat aturan keamanan WAF yang memfilter ancaman berdasarkan header permintaan HTTP.				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Regex-based penyaringan ancaman			Ya 	Ya 
Buat aturan keamanan WAF menggunakan ekspresi reguler untuk mencocokkan jalur URI dan atribut permintaan HTTP.				
JavaScript tantangan			Ya 	Ya 
Blokir ancaman otomatis dengan mengharuskan browser menyelesaikan JavaScript tantangan yang memverifikasi pengguna yang sah.				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Manajemen dan analitik bot (+bot AI)			Ya	Ya
Mendeteksi dan menganalisis lalu lintas AWS WAF bot dengan Bot Control untuk bot umum. Menyediakan kontrol untuk memblokir, menantang, atau mengizinkan bot yang tidak diverifikasi sambil mengidentifikasi dan membedakan bot yang diverifikasi seperti mesin pencari. Pantau bot AI dan ambil tindakan berdasarkan niat bot.				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Tanggapan WAF khusus		Ya 	Ya 	Ya 
Tetapkan kode status HTTP tertentu dan HTML kustom opsional, teks biasa, atau respons JSON saat permintaan diblokir oleh aturan.				
Penyisipan Header	Ya 	Ya 	Ya 	Ya 
Tambahkan header HTTP kustom ke permintaan yang lulus inspeksi WAF, memungkinkan aplikasi hilir memproses permintaan secara berbeda atau menandainya untuk dianalisis.				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Minta inspeksi badan	16 KB	16 KB	64 KB	64 KB
Ukuran maksimum konten isi permintaan HTTP yang dapat diperiksa oleh AWS WAF ancaman dan pola jahat.				
Asal pribadi dalam VPC			Ya	Ya
Tingkatkan keamanan dengan menyimpan aplikasi Anda di subnet pribadi VPC, hanya dapat diakses melalui CloudFront distribusi Anda dan disembunyi-kan dari internet publik, menggunakan asal VPC.				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Mutual TLS (asal)			Ya	Ya
Batasi akses tidak sah ke aplikasi Anda (asal) menggunakan sertifikat TLS-based klien, memastikan hanya CloudFront distribusi resmi Anda yang dapat membuat koneksi ke aplikasi Anda.				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Kontrol Akses Asal (OAC)	Ya	Ya	Ya	Ya
Pertahankan bucket S3 pribadi dan hanya izinkan akses melalui CloudFront distribusi yang Anda tentukan, pastikan konten Anda dilindungi oleh aturan WAF, batas tarif, dan kontrol keamanan lainnya yang dikonfigurasi dalam distribusi Anda CloudFront.				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Sertifikat TLS gratis	Ya	Ya	Ya	Ya
Sertifikat TLS gratis untuk domain Anda dengan perpanjangan otomatis melalui AWS Certificate Manager				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
URL yang ditandatangani	Ya	Ya	Ya	Ya
Buat URL aman yang menyediakan akses sementara ke konten pribadi untuk pengguna tertentu. Umumnya digunakan untuk berbagi dokumen pribadi dengan pengguna yang berwenang atau memberikan akses aman ke konten yang dilindungi setelah verifikasi pembayaran.				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Mutual TLS (mTL)				Ya
Batasi akses ke aplikasi Anda menggunakan otentikasi mTLS, memastikan hanya klien tepercaya dengan sertifikat valid yang dapat terhubung.				
Komputasi Tepi				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Perhitungan tepi tanpa server	Ya	Ya	Ya	Ya
Jalankan ringan JavaScript di tepi untuk memodifikasi URL, header HTTP, dan request/response elemen dalam milidetik menggunakan Fungsi. CloudFront Lambda @Edge juga dapat digunakan dengan semua tingkatan paket, tetapi tidak seperti CloudFront Fungsi, pemanggilan Lambda @Edge ditagih berdasarkan pay-as-you-go.				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Toko nilai kunci tepi		Ya	Ya	Ya
Simpan data di tepi menggunakan Key-Value Store kustomisasi konten yang cepat dan dinamis dengan CloudFront Fungsi.				
Dukungan Jaringan dan Protokol				
IPv6	Ya	Ya	Ya	Ya
Memberikan konten melalui IPv6 modern dan koneksi IPv4 tradisional dari CloudFront ke pemirsa dan asal. Mengaktifkan dukungan IPv6 end-to-end untuk aplikasi Anda.				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
HTTP/2	Ya	Ya	Ya	Ya
Aktifkan pemuatan halaman yang lebih cepat melalui fitur protokol modern seperti multiplexing, kompresi header, dan prioritas aliran. Secara otomatis digunakan ketika didukung oleh browser dan klien.				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
HTTP/3	Ya	Ya	Ya	Ya
Kirimkan konten menggunakan QUIC ke browser dan klien yang mendukungnya, memungkinkan koneksi yang lebih cepat dan peningkatan kinerja. Terutama menguntungkan pengguna seluler dan memelihara koneksi ketika kondisi jaringan berubah.	✔	✔	✔	✔

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
TLS 1.3	Ya	Ya	Ya	Ya
Memberikan koneksi HTTPS yang lebih cepat melalui proses jabat tangan yang membutuhkan satu pulang-pergi dibandingkan dengan dua di TLS 1.2. Mengurangi latensi byte pertama hingga 33% dibandingkan dengan versi TLS sebelumnya. Diaktifkan end-to-end untuk aplikasi Anda.	Ya	Ya	Ya	Ya

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
WebSockets	Ya	Ya	Ya	Ya
Aktifkan komunikasi dua arah real-time dan persisten antara browser dan server. Ideal untuk aplikasi obrolan AI, game multi-pemain, ruang kerja kolaboratif, dan umpan data real-time seperti platform perdagangan keuangan.	✔	✔	✔	✔
Penebangan dan Pemantauan				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Akses Log		Ya	Ya	Ya
Akses log CloudFront permintaan terperinci untuk memahami pola lalu lintas keamanan dan pengiriman, dengan konsumsi Amazon CloudWatch Logs disertakan tanpa biaya tambahan.				
Log permintaan WAF		Ya	Ya	Ya
Akses log AWS WAF permintaan terperinci untuk memahami pola lalu lintas keamanan dan pengiriman. CloudWatch Konsumsi Amazon Logs disertakan tanpa biaya tambahan.				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Dasbor keamanan	Ya	Ya	Ya	Ya
Pantau peristiwa keamanan, selidiki ancaman, dan lakukan tindakan pemblokiran langsung menggunakan analisis visual tanpa menulis aturan keamanan. Pro dan di atasnya termasuk peng analisis log visual untuk memahami pola lalu lintas dengan cepat tanpa meminta log.				
DNS				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Amazon Route 53 DNS	Ya	Ya	Ya	Ya
Layanan DNS otoritatif publik yang cepat dan andal menggunakan Route 53.				
Catatan per Zona yang Dihosting	50	100	1000	5000
Jumlah maksimum catatan DNS yang diizinkan di zona yang dihosting.				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
DNSSEC	Ya	Ya	Ya	Ya
Lindungi domain Anda dari spoofing DNS dan serangan man-in-the-middle di mana penyerang mencegah kueri DNS dan mengarahkan pengunjung ke situs web palsu. Mengamankan lalu lintas DNS dengan menandatangani catatan DNS Anda secara kriptografis.				
Penyimpanan				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Penyimpanan Amazon S3	5 GB	50 GB	1 TB	5 TB
Kredit penyimpanan Amazon S3 yang mengimbangi biaya penyimpanan Standar S3 apa pun di Akun AWS Tidak terbatas pada CloudFront konten atau tunduk pada rencana tunjangan penggunaan.				
Support dan Reliability				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Dukungan akun dan penagihan 24x7	Ya 	Ya 	Ya 	Ya 
One-on-one tanggapan terhadap pertanyaan akun dan penagihan.				
Jika Anda memiliki paket dukungan berbayar, Anda memenuhi syarat untuk menerima dukungan pada semua paket tarif tetap.				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Dokumenta	Ya	Ya	Ya	Ya
si dan AWS				
Dukungan forum				
Akses dokumenta				
si produk,				
makalah teknis,				
panduan praktik				
terbaik, forum				
AWS re:Post				
komunitas,				
dan informasi				
kesehatan				
layanan untuk				
membantu Anda				
merencanakan				
dan memecahkan masalah.				

Kinerja dan Pengiriman	Kosong	Pro	Bisnis	Premium
Uptime SLA			Ya	Ya
Perjanjian Tingkat Layanan (SLA) untuk Amazon CloudFront, AWS WAF, Amazon Route 53, dan Amazon CloudWatch memberikan komitmen ketersediaan layanan. Jika AWS tidak memenuhi komitmen SLA terkait, Anda akan memenuhi syarat untuk menerima kredit layanan.				

Tunjangan penggunaan bulanan

Setiap paket tarif tetap mencakup tunjangan penggunaan bulanan yang dirancang untuk kinerja optimal di tingkat tersebut. Tunjangan penggunaan bukanlah batas yang sulit — mereka mewakili penggunaan dasar yang dirancang untuk didukung oleh rencana Anda. Anda dapat melacak penggunaan Anda di CloudFront konsol kapan saja dan akan menerima pemberitahuan email otomatis ketika Anda mencapai 50%, 80%, dan 100% dari uang saku Anda.

Tunjangan penggunaan bulanan per tingkat paket

	Kosong	Pro	Bisnis	Premium
Permintaan	1 M	10 M	125 M	500 M (dapat dikonfigurasi hingga 6 B untuk harga flat rate baru)
Transfer data	100 GB	50 TB	50 TB	50 TB (dapat dikonfigurasi hingga 600 TB untuk harga flat rate baru)

Apa yang diperhitungkan dalam tunjangan penggunaan Anda

Serangan dan permintaan DDoS yang diblokir dengan AWS WAF tidak pernah dihitung terhadap tunjangan penggunaan Anda. Hanya lalu lintas yang melewati aturan AWS WAF keamanan Anda yang diperhitungkan dalam uang saku Anda. Ini berarti serangan dan lalu lintas yang tidak diinginkan tidak akan diperhitungkan terhadap Anda, dan Anda mempertahankan kemampuan untuk menentukan dengan tepat lalu lintas apa yang diblokir atau diizinkan aplikasi Anda.

Apa yang terjadi ketika penggunaan melebihi tunjangan

Rencana dirancang untuk menjadi fleksibel dan mengakomodasi pola lalu lintas dunia nyata seperti variabilitas lalu lintas, pertumbuhan organik, dan peristiwa viral. Yang paling penting: Anda tidak akan dikenakan biaya kelebihan, terlepas dari berapa banyak Anda melebihi uang saku Anda. Jika penggunaan Anda melebihi tunjangan Anda melalui pertumbuhan berkelanjutan selama beberapa bulan atau melalui penggunaan yang luar biasa tinggi dalam satu bulan, kami dapat menyesuaikan cara kami mengirimkan lalu lintas Anda. Sebagian besar pelanggan tidak pernah mengalami penyesuaian kinerja.

Begini cara kerjanya:

- Lonjakan lalu lintas pertama Anda hingga 3x tunjangan bulanan Anda tidak akan memengaruhi layanan Anda bulan itu. Akomodasi satu kali ini menangani peristiwa tak terduga seperti konten viral atau peluncuran produk yang sukses tanpa menghukum kesuksesan Anda.
- Penggunaan berkelanjutan di atas tunjangan Anda dievaluasi selama 2-3 bulan atau lebih, tidak segera. Fluktuasi kecil dan pertumbuhan sedang diharapkan dan ditampung. Hanya penggunaan berlebih yang substansif dan berkelanjutan yang menunjukkan aplikasi Anda telah melampaui tingkat Anda.
- Anda akan menerima pemberitahuan setiap bulan saat Anda mendekati dan melebihi tunjangan Anda. Jika penggunaan Anda secara konsisten dan signifikan melebihi paket Anda, kami sarankan untuk meningkatkan ke tingkat yang sesuai dengan pertumbuhan Anda dan memastikan kinerja optimal saat Anda menskalakan. Anda mengontrol kinerja Anda dengan memutakhirkan ketika pola penggunaan dasar Anda berubah.
- Jika Anda terus melebihi tunjangan penggunaan paket Anda tanpa melakukan upgrade, kami dapat menyesuaikan cara kami mengirimkan lalu lintas Anda. Misalnya, kami dapat melayani lalu lintas Anda dari lokasi tepi yang lebih sedikit atau lebih jauh atau menyesuaikan kinerja. Tingkat penyesuaian proporsional—penggunaan berlebih yang kecil melihat perubahan minimal, kelebihan berkelanjutan yang lebih besar melihat perubahan yang lebih nyata. Upgrade mengembalikan kinerja penuh.

Sebagian besar pelanggan tidak pernah mengalami penyesuaian kinerja. Rencana dirancang untuk mengakomodasi pola pertumbuhan normal.

Tunjangan penggunaan yang dapat dikonfigurasi pada paket Premium

Paket Premium menawarkan tunjangan penggunaan bulanan yang dapat dikonfigurasi per distribusi. Saat Anda berlangganan atau mengelola paket Premium, Anda dapat memilih tunjangan penggunaan bulanan yang lebih tinggi dari level berikut:

Tingkat penggunaan premium

Tingkat penggunaan premium	Transfer data bulanan	Permintaan bulanan	Flat-rate harga per bulan
Premium (default)	50 TB	500 M	\$1.000
Premium	75 TB	750 M	\$1.450
Premium	125 TB	1,25 B	\$2.250

Tingkat penggunaan premium	Transfer data bulanan	Permintaan bulanan	Flat-rate harga per bulan
Premium	200 TB	2 B	\$3.500
Premium	350 TB	3,5 B	\$6.000
Premium	600 TB	6 B	\$10.000

Ketika Anda memilih tingkat penggunaan yang lebih tinggi, harga bulanan Anda meningkat dan tunjangan penggunaan bulanan Anda meningkat sesuai dengan itu. Fitur dan layanan yang disertakan dalam paket Premium tetap sama di setiap tingkat penggunaan. Anda hanya mengubah tunjangan penggunaan dan harga flat rate. Kebijakan tunjangan penggunaan yang sama berlaku di semua tingkat penggunaan. Lihat perinciannya di [Apa yang terjadi ketika penggunaan melebihi tunjangan](#).

Anda dapat mengubah tingkat penggunaan Anda kapan saja di CloudFront konsol. Ketika Anda meningkatkan tingkat penggunaan Anda, perubahan akan segera berlaku. Harga dan tunjangan penggunaan Anda diprorata. Saat Anda mengurangi tingkat penggunaan, perubahan akan berlaku di awal siklus penagihan berikutnya.

Jika penggunaan dasar aplikasi Anda melebihi permintaan 6 B atau 600 TB per bulan, [hubungi kami untuk harga](#) khusus.

Note

Tunjangan penggunaan yang dapat dikonfigurasi hanya tersedia pada paket Premium. Tunjangan penggunaan pada paket Gratis, Pro, dan Bisnis tidak dapat dikonfigurasi.

Kelayakan berdasarkan penggunaan historis

CloudFront Penggunaan historis Anda dapat memengaruhi kelayakan Anda untuk mendaftar atau menurunkan versi ke tingkatan paket tertentu. Jika penggunaan terakhir Anda melebihi tunjangan penggunaan tingkat paket, Anda mungkin perlu memilih tingkat yang lebih tinggi yang lebih selaras dengan beban kerja Anda.

Biaya yang ditanggung oleh paket Anda

Paket Anda mencakup biaya untuk:

- CloudFront Distribusi Anda
- ACL AWS WAF web, aturan khusus, Aturan AWS Terkelola, dan biaya permintaan untuk ACL web yang terkait dengan distribusi Anda
- CloudWatch Konsumsi log untuk log CloudFront akses distribusi Anda dan log WAF terkait
- Zona yang dihosting Route 53, catatan DNS, dan kueri DNS saat dilampirkan ke paket distribusi Anda

Anda juga akan menerima kredit S3 untuk mengimbangi penggunaan penyimpanan Standar S3 di akun pembayar Anda, baik bucket S3 digunakan sebagai asal untuk distribusi Anda atau tidak. CloudFront

Rute 53 manajemen DNS dan rencana Anda

Jika Anda menggunakan Route 53 untuk DNS dan melampirkan zona ke paket Anda, paket tarif tetap Anda dapat mencakup biaya zona yang dihosting Route 53 Anda. Anda dapat melampirkan zona ke paket Anda di bagian Kelola Rencana CloudFront distribusi Anda. Ketika zona Anda dilampirkan ke paket, paket Anda menanggung biaya standar zona host Anda, termasuk biaya zona host bulanan, catatan DNS, dan biaya kueri DNS yang tunduk pada tunjangan masing-masing per tingkat, yang disediakan di bawah ini. Zona yang dihosting harus memenuhi persyaratan berikut:

- Ada di AWS akun yang sama dengan CloudFront distribusi Anda
- Pertahankan jumlah catatan yang diizinkan per zona yang dihosting untuk tingkat paket Anda
- Tutupi domain yang digunakan oleh CloudFront distribusi Anda

Jika zona host Anda tidak melekat pada paket Anda, itu akan tetap pada harga pay-as-you-go, di mana Anda bertanggung jawab atas semua biaya standar Route 53.

Memahami tunjangan kueri DNS bulanan

Ketika zona host Anda dilampirkan ke paket Anda, Anda mendapatkan:

1. [Kueri DNS ke catatan ALIAS yang menunjuk ke distribusi Anda CloudFront dan dukungan lainnya Layanan AWS](#)

2. Tunjangan bulanan tambahan untuk jenis catatan DNS lainnya

	Kosong	Pro	Bisnis	Premium
Kueri DNS ke catatan ALIAS (CloudFront dan dukungan Layanan AWS lainnya) per bulan	Tidak ada batas	Tidak ada batas	Tidak ada batas	Tidak ada batas
Tunjangan kueri DNS tambahan per bulan	1 M	5 M	20 M	100 M

Note

Untuk memaksimalkan manfaat paket Anda, gunakan catatan ALIAS untuk menunjuk ke CloudFront distribusi Anda. Catatan ALIAS yang menunjuk ke CloudFront dan [dukungan lainnya](#) tidak dihitung terhadap Layanan AWS tunjangan kueri DNS bulanan Anda. Semua kueri DNS lainnya, termasuk catatan CNAME ke CloudFront, dihitung terhadap tunjangan kueri DNS Anda.

Melebihi tunjangan kueri DNS

Jika penggunaan kueri DNS Anda melebihi tunjangan bulanan paket Anda, AWS dapat memberi tahu Anda. Pada saat itu, Anda dapat melepaskan zona yang dihosting dari paket di bagian Kelola Paket CloudFront distribusi Anda untuk mengembalikan zona yang dihosting ke harga bayar sesuai penggunaan. Jika Anda tidak melepaskan zona yang dihosting setelah menerima pemberitahuan ini, AWS dapat secara otomatis mentransisikan zona yang dihosting ke harga pay-as-you-go. Ketika zona yang dihosting pindah ke harga pay-as-you-go, Anda bertanggung jawab atas semua biaya Route 53 standar. CloudFront Distribusi Anda dan semua manfaat rencana lainnya terus tidak berubah.

Kurangi secara keseluruhan AWS biaya dengan paket harga

CloudFront Paket harga flat-rate dapat mengurangi AWS biaya keseluruhan Anda dengan tiga cara:

Pertama, biaya transfer data antara CloudFront dan AWS aplikasi Anda yang berjalan pada layanan seperti Amazon S3, AWS Application Load Balancer (ALB), atau Amazon API Gateway secara otomatis dibebaskan. Ketika Anda melayani AWS aplikasi Anda melalui CloudFront alih-alih langsung ke internet, paket tarif tetap Anda mencakup biaya transfer data antara aplikasi Anda dan pemirsa Anda dengan harga bulanan sederhana tanpa khawatir kelebihan.

Kedua, CloudFront mengurangi biaya komputasi dan database Anda dengan melindungi infrastruktur aplikasi Anda dan mengurangi jumlah permintaan yang mencapai asal Anda. Ini menyajikan konten yang di-cache dari lokasi tepi atau cache tepi regional, menciutkan permintaan duplikat, dan memblokir lalu lintas berbahaya dan tidak diinginkan sebelum mencapai layanan backend Anda. Ini berarti lebih sedikit permintaan yang mengenai server aplikasi Anda, database, dan lainnya Layanan AWS yang mengenakan biaya berdasarkan penggunaan, yang mengurangi biaya Anda.

Terakhir, setiap paket menyertakan kredit penyimpanan Standar Amazon S3 untuk mengimbangi penggunaan penyimpanan untuk Anda. Akun AWS

Untuk memaksimalkan penghematan ini, konfigurasi AWS asal Anda untuk hanya menerima lalu lintas dari CloudFront. Untuk S3, gunakan [Origin Access Control OAC](#) dengan bucket pribadi untuk memberikan akses ke distribusi yang Anda tentukan. CloudFront Untuk instans Application Load Balancer, Network Load Balancer, dan Amazon EC2 di subnet pribadi, [batasi akses ke distribusi yang Anda tentukan menggunakan VPC](#) Origins. CloudFront

Kelola paket harga flat-rate Anda

Ikuti prosedur ini di CloudFront konsol untuk berlangganan, meningkatkan, menurunkan versi, atau membatalkan paket harga untuk distribusi Anda.

Berlangganan distribusi baru ke paket harga

Saat Anda membuat distribusi baru, Anda dapat berlangganan paket harga.

Untuk berlangganan distribusi baru ke paket harga

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Distribusi, lalu ikuti langkah-langkah untuk membuat distribusi.

3. Pilih paket harga distribusi Anda. Perhatikan bahwa beberapa fitur tidak tersedia per tingkat paket harga. Tinjau fitur per paket dan pilih paket harga yang Anda butuhkan untuk aplikasi Anda.
4. Selesaikan langkah-langkah untuk [membuat distribusi Anda](#).

Berlangganan distribusi yang ada ke paket harga

Saat memperbarui distribusi, Anda dapat berlangganan paket harga. Sebelum memilih paket harga, pastikan konfigurasi distribusi Anda kompatibel dengan paket yang Anda inginkan.

Tip

Jika distribusi Anda saat ini menggunakan [fitur yang tidak didukung](#), Anda harus menonaktifkan fitur tersebut sebelum dapat berlangganan paket harga. Ini termasuk menonaktifkan fitur seperti log akses real-time.

Setelah konfigurasi distribusi Anda kompatibel, Anda dapat memilih paket harga yang Anda inginkan sambil memperbarui distribusi.

Untuk berlangganan distribusi yang ada ke paket harga

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Distribusi, lalu ikuti langkah-langkah untuk memperbarui distribusi yang ada.
3. Pilih paket harga distribusi Anda. Perhatikan bahwa beberapa fitur tidak tersedia per tingkat paket harga. Tinjau fitur per paket dan pilih paket harga yang Anda butuhkan untuk aplikasi Anda.
4. Selesaikan langkah-langkah untuk [memperbarui distribusi Anda](#).

Tingkatkan paket harga

Kami menyarankan Anda meningkatkan paket jika Anda mendekati atau telah melampaui tunjangan penggunaan bulanan Anda, atau jika Anda ingin mengaktifkan fitur yang tersedia di tingkat berikutnya.

Saat Anda meningkatkan ke tingkat paket yang lebih tinggi, perubahan segera berlaku. Harga dan tunjangan penggunaan Anda diprorata. Distribusi Anda dan sumber daya terkait akan memiliki akses ke fitur yang tersedia dan tunjangan penggunaan yang lebih tinggi dari paket baru Anda.

Untuk meng-upgrade paket harga

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Distribusi.
3. Pilih distribusi Anda yang berlangganan paket harga yang ada.
4. Ikuti petunjuk untuk meningkatkan paket harga distribusi Anda.
5. Selesaikan langkah-langkah untuk [memperbarui distribusi yang ada](#).

Ubah tingkat penggunaan paket Premium

Jika Anda menggunakan paket Premium, Anda dapat mengubah tingkat penggunaan untuk menambah atau mengurangi tunjangan penggunaan Anda.

Ketika Anda meningkatkan tingkat penggunaan Anda, perubahan akan segera berlaku dan harga Anda diprorata. Saat Anda mengurangi tingkat penggunaan, perubahan akan berlaku di awal siklus penagihan berikutnya.

Untuk mengubah tingkat penggunaan paket Premium

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Distribusi.
3. Pilih distribusi Anda yang berlangganan paket harga Premium.
4. Pilih tombol Kelola Paket.
5. Di bagian Kelola paket, pilih Ubah paket.
6. Ikuti petunjuk untuk mengubah tingkat penggunaan paket harga Premium. Konsol menampilkan tingkat penggunaan yang tersedia dan harga bulanan yang sesuai.

Turunkan paket harga

Kami menyarankan Anda menurunkan versi ke tingkat paket yang lebih rendah jika Anda tidak memerlukan fitur tambahan pada tingkat yang ada. Misalnya, Anda mungkin menurunkan versi jika Anda berharap aplikasi Anda akan mengalami lalu lintas yang lebih rendah.

Jika Anda menurunkan versi ke tingkat yang lebih rendah, perubahan penagihan Anda akan berlaku di awal siklus penagihan berikutnya.

Jika distribusi Anda saat ini melebihi tunjangan penggunaan untuk paket, Anda dapat menurunkan versi setelah penggunaan Anda berada dalam batas penggunaan untuk tingkat yang Anda inginkan. Agar tidak dikenakan biaya untuk tingkat paket yang ada pada siklus penagihan berikutnya, turunkan versi sebelum akhir bulan.

Untuk menurunkan versi paket harga

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Distribusi.
3. Pilih distribusi Anda yang berlangganan paket harga yang ada.
4. Ikuti petunjuk untuk menurunkan versi paket harga distribusi Anda. Jika Anda memiliki fitur yang tidak didukung, Anda harus menghapus fitur atau sumber daya dari distribusi.
5. Selesaikan langkah-langkah untuk [memperbarui distribusi yang ada](#).

Batalkan paket harga

Ketika Anda membatalkan paket harga, Anda akan mempertahankan harga flat-rate Anda hingga akhir siklus penagihan Anda saat ini. Distribusi Anda dan semua sumber daya paket terkait kemudian akan beralih ke harga pay-as-you-go pada awal siklus penagihan berikutnya.

Untuk membatalkan paket harga

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Distribusi.
3. Pilih distribusi Anda yang berlangganan paket harga yang ada.

4. Ikuti petunjuk untuk membatalkan paket harga distribusi Anda. Jika Anda memiliki fitur yang tidak didukung, Anda harus menghapus fitur atau sumber daya dari distribusi.
5. Selesaikan langkah-langkah untuk [memperbarui distribusi yang ada](#).

Membatalkan perubahan paket yang tertunda

Jika Anda menurunkan atau membatalkan paket harga flat-rate Anda, Anda harus menunggu hingga akhir siklus penagihan saat ini sebelum perubahan Anda berlaku. Untuk mempertahankan paket harga flat-rate yang ada, meningkatkan, atau menurunkan paket harga Anda lagi, Anda harus terlebih dahulu membatalkan perubahan paket yang tertunda.

Untuk membatalkan perubahan paket harga yang tertunda

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Distribusi.
3. Pilih distribusi Anda yang berlangganan paket harga yang ada.
4. Ikuti petunjuk untuk membatalkan perubahan paket tertunda distribusi Anda.
5. Pilih paket harga yang Anda inginkan untuk distribusi Anda.
6. Selesaikan langkah-langkah untuk memperbarui distribusi yang ada.

Menghapus distribusi dengan paket harga

Anda tidak dapat menghapus distribusi yang berlangganan paket harga. Anda harus terlebih dahulu membatalkan paket harga dan kemudian setelah siklus penagihan saat ini, hapus distribusi.

Untuk menghapus distribusi dengan paket harga

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Distribusi.
3. Ikuti langkah-langkah sebelumnya untuk membatalkan paket harga distribusi.
4. Ikuti langkah-langkah untuk [menghapus distribusi](#).

Note

Anda dapat menonaktifkan distribusi yang berlangganan paket harga, tetapi Anda masih akan dikenakan biaya untuk paket itu. Untuk menghentikan biaya untuk paket Anda, Anda harus membatalkannya terlebih dahulu.

Izin

Untuk melihat atau mengelola langganan paket harga untuk CloudFront distribusi Anda, Anda harus memiliki izin yang diperlukan. Untuk informasi selengkapnya, lihat [AWS kebijakan terkelola: CloudFrontFullAccess](#) dan [AWS kebijakan terkelola: CloudFrontReadOnlyAccess](#).

Flat-rate kuota rencana harga

Tabel berikut menunjukkan kuota dan batasan untuk paket harga CloudFront flat-rate.

Note

Kuota ini tidak dapat ditingkatkan untuk Anda Akun AWS.

Account-level kuota	Kuota
Paket harga per Akun AWS	100
Paket gratis per Akun AWS	3
Apex-level domain per paket	1

Fitur yang tidak didukung

Sebelum Anda dapat mengaitkan distribusi dengan paket harga, Anda harus memastikan bahwa fitur tertentu dinonaktifkan dan asosiasi dihapus.

Catatan

- Jika distribusi atau akun Anda memiliki salah satu batasan ini, Anda harus menyelesaikannya sebelum dapat menggunakan paket harga. Setelah Anda membuat perubahan pada distribusi Anda, tunggu perubahan menyebar ke semua lokasi tepi.
- Anda harus memiliki ACL AWS WAF Web yang terkait dengan distribusi Anda jika Anda menggunakan paket harga. Sumber daya ini tidak dapat dihapus atau dipisahkan dari distribusi Anda kecuali Anda beralih ke harga pay-as-you-go untuk distribusi tersebut.

Fitur yang tidak didukung

Anda tidak dapat berlangganan distribusi ke paket harga jika konfigurasinya berisi fitur yang tidak didukung berikut. Anda dapat menonaktifkan fitur yang tidak didukung dan menggunakan opsi alternatif, atau menyimpan pay-as-you-go untuk distribusi Anda.

Fitur yang tidak didukung	Opsi alternatif	Layanan AWS
Multi-tenant distribusi	Gunakan distribusi standar atau harga pay-as-you-go	CloudFront
Penyebaran berkelanjutan dan distribusi Pementasan	Gunakan harga pay-as-you-go	CloudFront
Konfigurasi daftar IP anycast	Gunakan harga pay-as-you-go	CloudFront
Real-time log akses	Gunakan log akses standar atau harga bayar sesuai penggunaan	CloudFront
Bot yang Ditargetkan	Gunakan bot umum atau harga pay-as-you-go	AWS WAF
Aturan Dikelola Mitra	Gunakan harga pay-as-you-go	AWS WAF
	Gunakan harga pay-as-you-go	AWS WAF

Fitur yang tidak didukung	Opsi alternatif	Layanan AWS
Pencegahan Penipuan Pembuatan Akun		
Perlindungan Pengambilalihan Akun	Gunakan harga pay-as-you-go	AWS WAF
Grup Aturan	Buat aturan individual (grup aturan adalah AWS WAF aturan bersama yang dapat diterapkan ke ACL web, mirip dengan kebijakan pada CloudFront)	AWS WAF
Fitur warisan		
Forwarded Values konfigurasi	Gunakan kebijakan permintaan Origin	CloudFront
Berdedikasi IP/SSL	Gunakan harga pay-as-you-go	CloudFront
Enkripsi tingkat lapangan	Gunakan harga pay-as-you-go	CloudFront
AWS Identity and Access Management (IAM) sertifikat server	Gunakan AWS Certificate Manager sertifikat (ACM)	CloudFront
Identitas akses asal (OAI)	Gunakan kontrol akses Origin (OAC)	CloudFront
Pengaturan cache lama	Gunakan kebijakan cache dan kebijakan permintaan asal .	CloudFront

Asosiasi yang tidak didukung

Anda tidak dapat berlangganan distribusi ke paket harga jika distribusi sudah dikaitkan dengan salah satu sumber daya berikut yang sudah dikaitkan dengan distribusi lain. Sumber daya yang terkait dengan distribusi yang berlangganan paket harga hanya dapat digunakan untuk distribusi itu. Misalnya, jika Anda memiliki CloudFront fungsi yang menggunakan penyimpanan nilai kunci, baik fungsi maupun penyimpanan nilai kunci tidak dapat dibagikan untuk distribusi yang ada pada paket harga.

- CloudFront Fungsi
- CloudFront Fungsi yang terkait dengan penyimpanan nilai kunci
- AWS WAF ACL Web

Untuk berlangganan distribusi ke paket harga, hapus sumber daya terkait atau ganti dengan yang lain.

Account-level kendala

Akun AWS tidak memenuhi syarat untuk paket harga jika memenuhi salah satu dari ketentuan berikut:

- Anda mencapai jumlah langganan maksimum yang diizinkan. Lihat [Flat-rate kuota rencana harga](#).
- Akun Anda menggunakan AWS Tingkat Gratis.

Resource-level kendala

Distribusi tidak memenuhi syarat untuk paket harga jika memenuhi salah satu ketentuan berikut:

- Distribusi Anda telah diaktifkan AWS Shield Advanced
- Distribusi Anda telah mengaktifkan [Layanan Firewall Manager](#) untuk ACL web Anda. Firewall Manager tidak akan mengelola WebACL CloudFront distribusi Anda dalam paket harga.

Fitur tambahan yang dapat memengaruhi paket harga Anda

Flat-rate Paket harga memungkinkan Anda membayar tarif tetap untuk CloudFront distribusi Anda dan fitur-fitur yang tercantum di atas yang keduanya termasuk dalam paket Anda dan terkait dengan

distribusi Anda CloudFront . Semua fitur lainnya dapat dikenakan biaya tambahan, termasuk namun tidak terbatas pada hal-hal berikut:

CloudFront

- Pemanggilan fungsi Lambda @Edge

AWS WAF

- Teka-teki CAPTCHA yang dibuat menggunakan [JavaScript API ditagih menggunakan harga pay-as-you-go](#). Tanggapan CAPTCHA yang dikonfigurasi dalam AWS WAF aturan Anda (kasus penggunaan yang paling umum) disertakan dalam paket Anda tanpa biaya tambahan.

Route 53

- Route 53 DNSSEC memiliki biaya AWS KMS
- Blok Route 53 IP (CIDR) (1.000 pertama gratis per Akun AWS)
- Route 53 Health Checks (50 yang pertama gratis per Akun AWS)

Fitur logging

- Route 53 Log Kueri DNS, log CloudFront Fungsi, dan Log Fungsi CloudFront Koneksi
- AWS WAF pengiriman log ke Amazon S3
- CloudFront atau AWS WAF log pengiriman ke Amazon Data Firehose
- CloudWatch Metrik tambahan untuk CloudFront
- CloudFront mengakses log dalam format Parquet

Note

Paket Anda mencakup konsumsi CloudWatch Log Amazon untuk log CloudFront standar (log akses) dan log WAF tanpa biaya tambahan. Semua CloudWatch biaya lain seperti penyimpanan dan kueri tidak ditanggung oleh paket Anda. Semua log lainnya juga ditagih secara terpisah.

 Note

Paket Anda mencakup DNS otoritatif publik dari Route 53. Ketika zona yang dihosting Route 53 Anda dilampirkan ke paket harga Anda, paket Anda menanggung biaya standar zona host Anda, termasuk biaya zona host bulanan, catatan DNS, dan biaya kueri DNS yang tunduk pada masing-masing tunjangan per tingkat. Semua biaya lain dari penggunaan Route 53 dan fitur yang tidak tercantum di atas seperti yang termasuk dalam paket Anda tidak ditanggung oleh paket Anda.

Paket harga vs. harga pay-as-you-go

Flat-rate paket dan harga pay-as-you-go menawarkan keuntungan yang berbeda berdasarkan kebutuhan Anda. Dengan paket flat-rate, Anda membayar satu harga yang mencakup beberapa Layanan AWS seperti CloudFront, Route 53 AWS WAF, dan konsumsi CloudWatch Log dan tidak pernah menghadapi biaya berlebihan, bahkan selama lonjakan lalu lintas atau serangan.

Dengan harga pay-as-you-go, Anda ditagih secara terpisah untuk setiap layanan dan fitur berdasarkan penggunaan aktual Anda. Meskipun ini memberikan fleksibilitas lengkap dalam pemilihan dan konfigurasi layanan, biaya Anda dapat bervariasi dari bulan ke bulan berdasarkan pola lalu lintas, dan Anda perlu memantau penggunaan di berbagai layanan untuk mengelola biaya.

Flat-rate paket sangat ideal jika Anda ingin penagihan bulanan gabungan, konfigurasi layanan yang disederhanakan, dan fitur keamanan bawaan tanpa khawatir tentang biaya kelebihan biaya. Pay-as-you-go penetapan harga adalah pilihan yang lebih baik jika Anda memerlukan kontrol penuh atas fitur layanan individual, konfigurasi khusus, akses ke fitur yang tidak tersedia dalam paket tarif tetap, atau jika Anda berharap untuk menangani lonjakan lalu lintas yang besar dan dapat diprediksi. Paket harga CloudFront flat-rate Amazon tidak dapat digabungkan dengan penawaran, promosi, atau diskon lainnya.

Konfigurasikan distribusi

Anda membuat CloudFront distribusi Amazon untuk mengetahui CloudFront dari mana Anda ingin konten dikirimkan, dan detail tentang cara melacak dan mengelola pengiriman konten.

Saat Anda membuat CloudFront distribusi, CloudFront secara otomatis mengonfigurasi sebagian besar setelan distribusi untuk Anda, berdasarkan jenis asal konten Anda. Untuk detail selengkapnya tentang pengaturan yang telah dikonfigurasi sebelumnya, lihat [Referensi pengaturan distribusi yang telah dikonfigurasi sebelumnya](#). Secara opsional, Anda dapat memilih untuk mengedit pengaturan distribusi secara manual. Untuk informasi selengkapnya, lihat [Semua referensi pengaturan distribusi](#).

Pengaturan berikut dapat dikonfigurasi:

- Asal konten Anda —Bucket Amazon S3 AWS Elemental MediaPackage , channel, container AWS Elemental MediaStore , penyeimbang beban Elastic Load Balancing, atau server HTTP tempat file CloudFront didistribusikan. Anda dapat menentukan kombinasi hingga 25 asal untuk satu distribusi.
- Akses —Apakah Anda ingin akses ke file tersedia untuk semua orang atau dibatasi untuk beberapa pengguna.
- Keamanan — Apakah Anda ingin mengaktifkan AWS WAF perlindungan dan mengharuskan pengguna menggunakan HTTPS untuk mengakses konten Anda. Untuk distribusi multi-penyewa, hanya daftar kontrol akses web (ACL) AWS WAF V2 yang didukung.
- Kunci cache —Nilai mana, jika ada, yang ingin Anda sertakan dalam kunci cache. Kunci cache secara unik mengidentifikasi setiap file dalam cache untuk distribusi tertentu.
- Pengaturan permintaan asal —Apakah Anda CloudFront ingin menyertakan header HTTP, cookie, atau string kueri dalam permintaan yang dikirimkan ke asal Anda.
- Pembatasan geografis —Apakah Anda CloudFront ingin mencegah pengguna di negara tertentu mengakses konten Anda.
- Log —Apakah Anda CloudFront ingin membuat log standar atau log akses real-time yang menampilkan aktivitas penampil.

Untuk informasi selengkapnya, lihat [Semua referensi pengaturan distribusi](#).

Untuk jumlah distribusi maksimum saat ini yang dapat Anda buat untuk masing-masing Akun AWS, lihat [Kuota umum di distribusi](#). Tidak ada jumlah maksimum file yang dapat Anda gunakan per distribusi.

Anda dapat menggunakan distribusi untuk melayani konten berikut melalui HTTP atau HTTPS:

- Konten unduhan statis dan dinamis, seperti HTML, CSS JavaScript, dan file gambar, menggunakan HTTP atau HTTPS.
- Video sesuai permintaan dalam berbagai format, seperti Apple HTTP Live Streaming (HLS) dan Microsoft Smooth Streaming. (Untuk distribusi multi-tenant, Smooth Streaming tidak didukung.) Untuk informasi selengkapnya, lihat [Mengirimkan video sesuai permintaan dengan CloudFront](#).
- Acara langsung, seperti pertemuan, konferensi, atau konser, secara waktu nyata. Untuk streaming langsung, Anda dapat membuat distribusi secara otomatis dengan menggunakan CloudFormation tumpukan. Untuk informasi selengkapnya, lihat [Memberikan streaming video dengan CloudFront dan Layanan AWS Media](#).

Topik berikut memberikan rincian lebih lanjut tentang CloudFront distribusi dan cara mengonfigurasinya untuk memenuhi kebutuhan bisnis Anda. Untuk informasi lebih lanjut tentang pembuatan distribusi, lihat [Buat distribusi](#).

Topik

- [Memahami cara kerja distribusi multi-tenant](#)
- [Buat distribusi](#)
- [Referensi pengaturan distribusi yang telah dikonfigurasi sebelumnya](#)
- [Semua referensi pengaturan distribusi](#)
- [Uji distribusi](#)
- [Perbarui distribusi](#)
- [Menandai CloudFront sumber daya](#)
- [Menghapus sebuah distribusi](#)
- [Gunakan berbagai asal dengan CloudFront distribusi](#)
- [Aktifkan IPv6 untuk distribusi CloudFront](#)
- [Gunakan penerapan CloudFront berkelanjutan untuk menguji perubahan konfigurasi CDN dengan aman](#)
- [Gunakan URL kustom dengan menambahkan nama domain alternatif \(CNames\)](#)
- [Gunakan WebSockets dengan CloudFront distribusi](#)
- [Minta IP statis Anycast untuk digunakan untuk daftar yang diizinkan](#)

- [Menggunakan gRPC dengan distribusi CloudFront](#)

Memahami cara kerja distribusi multi-tenant

Anda dapat membuat distribusi CloudFront multi-tenant dengan pengaturan yang dapat digunakan kembali di beberapa penyewa distribusi. Dengan distribusi multi-tenant, Anda dapat CloudFront mengonfigurasi pengaturan distribusi untuk Anda, berdasarkan jenis asal konten Anda. Untuk detail selengkapnya tentang pengaturan yang telah dikonfigurasi sebelumnya, lihat [Referensi pengaturan distribusi yang telah dikonfigurasi sebelumnya](#).

Manfaat menggunakan distribusi multi-tenant alih-alih distribusi standar meliputi:

- Mengurangi beban operasional.
- Konfigurasi yang dapat digunakan kembali untuk admin web dan penyedia perangkat lunak untuk mengelola CloudFront distribusi untuk beberapa aplikasi web yang mengirimkan konten kepada pengguna akhir.
- Integrasi yang disempurnakan dengan yang lain Layanan AWS untuk menghadirkan manajemen sertifikat otomatis, kontrol keamanan terpadu, dan kontrol konfigurasi tanpa kerumitan dalam skala besar.
- Mempertahankan pola sumber daya yang konsisten di seluruh implementasi Anda. Tentukan pengaturan yang harus dibagikan dan kemudian tentukan penyesuaian untuk pengaturan yang akan diganti.
- Pengaturan asal dan keamanan yang dapat disesuaikan untuk memenuhi kebutuhan spesifik di tingkat penyewa distribusi.
- Atur penyewa distribusi Anda ke dalam tingkatan yang berbeda. Misalnya, jika beberapa penyewa distribusi memerlukan Origin Shield dan beberapa tidak, Anda dapat mengelompokkan penyewa distribusi ke dalam distribusi multi-penyewa yang berbeda.
- Berbagi konfigurasi DNS umum di beberapa domain.

Tidak seperti distribusi standar, distribusi multi-tenant tidak dapat diakses secara langsung karena tidak memiliki titik akhir CloudFront routing. Oleh karena itu, harus digunakan bersama dengan grup koneksi dan satu atau lebih penyewa distribusi. Sementara distribusi standar memiliki CloudFront titik akhir mereka sendiri dan dapat langsung diakses oleh pengguna akhir, mereka tidak dapat digunakan sebagai template untuk distribusi lain.

Untuk informasi tentang kuota distribusi multi-penyewa, lihat. [Kuota pada distribusi multi-penyewa](#)

Topik

- [Cara kerjanya](#)
- [Istilah](#)
- [Fitur yang tidak didukung](#)
- [Kustomisasi penyewa distribusi](#)
- [Minta sertifikat untuk penyewa CloudFront distribusi Anda](#)
- [Buat grup koneksi khusus \(opsional\)](#)
- [Migrasi ke distribusi multi-tenant](#)

Cara kerjanya

Dalam distribusi standar, distribusi berisi semua pengaturan yang ingin Anda aktifkan untuk situs web atau aplikasi Anda, seperti konfigurasi asal, perilaku cache, dan pengaturan keamanan. Jika Anda ingin membuat situs web terpisah dan menggunakan banyak pengaturan yang sama, Anda perlu membuat distribusi baru setiap kali.

CloudFront distribusi multi-tenant berbeda karena Anda dapat membuat distribusi multi-tenant awal. Untuk setiap situs web baru, Anda membuat penyewa distribusi yang secara otomatis mewarisi nilai yang ditentukan dari distribusi sumbernya. Kemudian, Anda menyesuaikan pengaturan khusus untuk penyewa distribusi Anda.

Ikhtisar

1. Untuk memulai, Anda terlebih dahulu membuat distribusi multi-tenant. CloudFront mengonfigurasi pengaturan distribusi untuk Anda, berdasarkan jenis asal konten Anda. Anda dapat menyesuaikan pengaturan untuk semua asal kecuali asal VPC. Pengaturan asal VPC disesuaikan pada sumber daya asal VPC itu sendiri. Untuk informasi selengkapnya tentang pengaturan distribusi multi-tenant yang dapat Anda sesuaikan, lihat. [Referensi pengaturan distribusi yang telah dikonfigurasi sebelumnya](#)
 - Sertifikat TLS yang Anda gunakan untuk distribusi multi-tenant dapat diwarisi oleh penyewa distribusi Anda. Distribusi multi-tenant itu sendiri tidak dapat dirutekan, sehingga tidak akan memiliki nama domain yang terkait dengannya.
2. Secara default, CloudFront buat grup koneksi untuk Anda. Grup koneksi mengontrol cara permintaan penampil untuk konten terhubung CloudFront. Anda dapat menyesuaikan beberapa pengaturan perutean di grup koneksi.

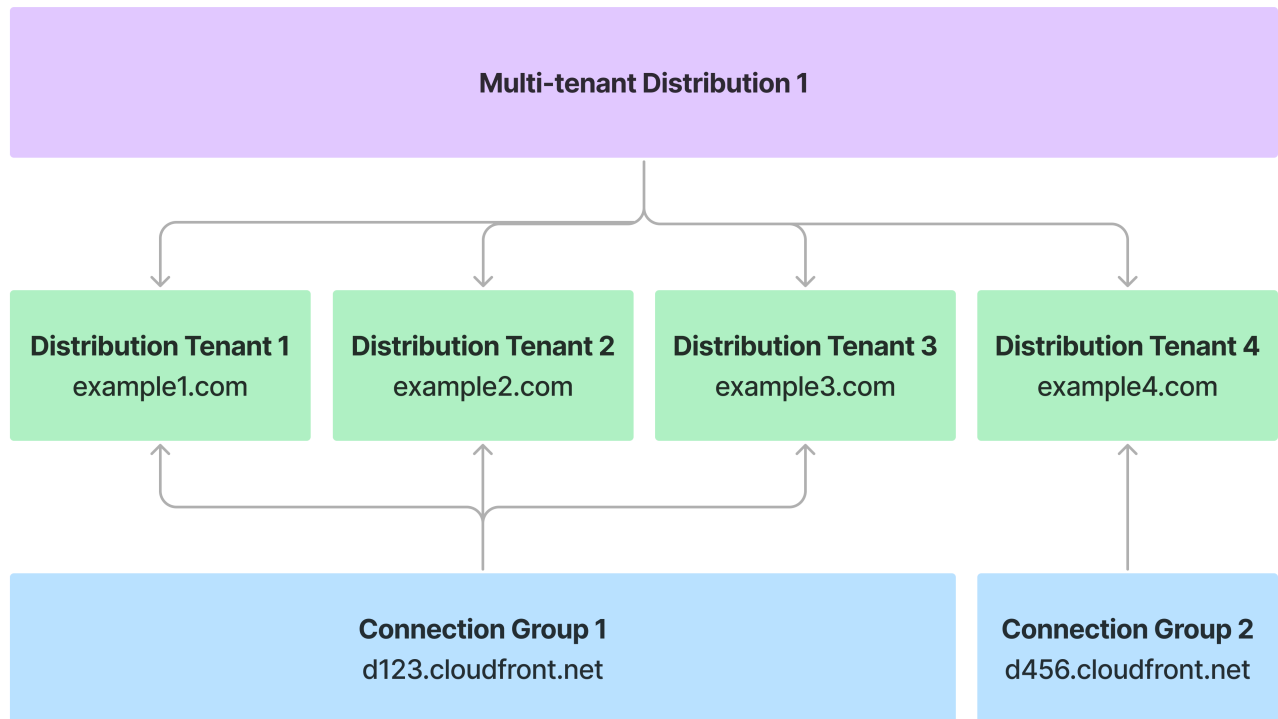
Anda dapat mengubahnya dengan membuat grup koneksi Anda sendiri secara manual. Untuk informasi selengkapnya, lihat [Buat grup koneksi khusus \(opsional\)](#).

3. Kemudian, Anda membuat satu atau lebih penyewa distribusi. Penyewa distribusi adalah “pintu depan” bagi pemirsa untuk mengakses konten Anda. Setiap penyewa distribusi mereferensikan distribusi multi-tenant dan secara otomatis dikaitkan dengan grup koneksi yang CloudFront dibuat untuk Anda. Penyewa distribusi mendukung domain atau subdomain individual.
4. Anda kemudian dapat menyesuaikan beberapa pengaturan penyewa distribusi, seperti domain vanity dan jalur asal. Untuk informasi selengkapnya, lihat [Kustomisasi penyewa distribusi](#).
5. Terakhir, Anda harus memperbarui catatan DNS di host DNS Anda untuk merutekan lalu lintas ke penyewa distribusi. Untuk melakukan ini, dapatkan nilai CloudFront endpoint dari grup koneksi Anda dan buat catatan CNAME yang menunjuk ke titik akhir CloudFront .

Example Contoh

Grafik berikut menunjukkan bagaimana distribusi multi-tenant, penyewa distribusi, dan grup koneksi bekerja sama untuk mengirimkan konten kepada pemirsa Anda untuk beberapa domain.

1. Distribusi multi-tenant mendefinisikan pengaturan yang diwariskan untuk setiap penyewa distribusi. Anda menggunakan distribusi multi-tenant sebagai template.
2. Setiap penyewa distribusi yang dibuat dari distribusi multi-tenant memiliki domain sendiri.
3. Penyewa distribusi secara otomatis ditambahkan ke grup koneksi yang CloudFront dibuat untuk Anda saat Anda membuat distribusi multi-penyewa. Grup koneksi mengontrol bagaimana permintaan penampil terhubung ke CloudFront jaringan.



Untuk instruksi pembuatan distribusi multi-penyewa yang mendetail, lihat. [Buat CloudFront distribusi di konsol](#)

Istilah

Konsep-konsep berikut menjelaskan komponen distribusi multi-tenant:

Multi-tenant distribusi

Cetak biru, distribusi multi-tenant yang menentukan semua pengaturan konfigurasi bersama untuk penyewa distribusi apa pun, termasuk perilaku cache, perlindungan keamanan, dan asal. Multi-tenant distribusi tidak dapat melayani lalu lintas secara langsung. Mereka harus digunakan bersama dengan kelompok koneksi dan penyewa distribusi.

Distribusi standar

Distribusi yang tidak memiliki fungsi multi-tenant. Distribusi ini paling baik untuk mendukung situs web atau aplikasi tunggal.

Penyewa distribusi

Penyewa distribusi mewarisi konfigurasi distribusi multi-tenant. Beberapa pengaturan konfigurasi dapat disesuaikan pada tingkat penyewa distribusi. Penyewa distribusi harus memiliki sertifikat TLS yang valid, yang dapat diwarisi dari distribusi multi-tenant selama mencakup domain penyewa distribusi atau subdomain.

Penyewa distribusi harus dikaitkan dengan grup koneksi. CloudFront membuat grup koneksi untuk Anda saat Anda membuat penyewa distribusi, dan secara otomatis menetapkan penyewa distribusi apa pun ke grup koneksi tersebut.

Multi-tenancy

Anda dapat menggunakan distribusi multi-tenant untuk menayangkan konten di beberapa domain, sambil berbagi konfigurasi dan infrastruktur. Pendekatan ini memungkinkan domain yang berbeda (disebut penyewa) untuk berbagi pengaturan umum dari distribusi multi-penyewa, sambil mempertahankan kustomisasi mereka sendiri.

Grup koneksi

Menyediakan titik akhir CloudFront perutean yang menyajikan konten kepada pemirsa. Anda harus mengaitkan setiap penyewa distribusi ke grup koneksi untuk mendapatkan titik akhir CloudFront perutean yang sesuai untuk catatan CNAME yang Anda buat untuk domain penyewa distribusi atau subdomain Anda. Grup koneksi dapat dibagi di beberapa penyewa distribusi. Grup koneksi mengelola pengaturan perutean untuk penyewa distribusi, seperti pengaturan daftar IP IPv6 dan Anycast.

Parameter

Daftar pasangan kunci-nilai untuk nilai placeholder, seperti jalur asal dan nama domain. Anda dapat menentukan parameter dalam distribusi multi-tenant Anda dan memberikan nilai untuk parameter tersebut di tingkat penyewa distribusi. Anda memilih apakah nilai parameter harus dimasukkan untuk penyewa distribusi.

Jika Anda tidak memberikan nilai untuk parameter opsional dalam penyewa distribusi, maka nilai default dari distribusi multi-penyewa digunakan sebagai nilai.

CloudFront titik akhir perutean

DNS Canonical untuk grup koneksi, seperti `d123.cloudfront.net`. Digunakan dalam catatan CNAME untuk domain penyewa distribusi atau subdomain Anda.

Kustomisasi

Anda dapat menyesuaikan penyewa distribusi Anda sehingga mereka menggunakan pengaturan yang berbeda dari distribusi multi-penyewa. Untuk setiap penyewa distribusi, Anda dapat menentukan daftar kontrol akses AWS WAF web (ACL), sertifikat TLS, dan batasan geografis yang berbeda.

Fitur yang tidak didukung

Fitur berikut tidak dapat digunakan dengan distribusi multi-tenant. Jika Anda ingin membuat distribusi multi-tenant baru menggunakan pengaturan yang sama dengan distribusi standar Anda, perhatikan bahwa beberapa pengaturan tidak tersedia.

Catatan

- Saat ini, AWS Firewall Manager kebijakan hanya berlaku untuk distribusi standar Anda. Firewall Manager akan menambahkan dukungan untuk distribusi multi-tenant di rilis mendatang.
- Tidak seperti distribusi standar, Anda menentukan nama domain Anda (alias) di tingkat penyewa distribusi. Untuk informasi selengkapnya, lihat [Minta sertifikat untuk penyewa CloudFront distribusi Anda](#) dan operasi [CreateDistributionTenantAPI](#).

- [Penyebaran berkelanjutan](#)
- [Origin Access Identity \(OAI\)](#) — Gunakan [Origin Access Control \(OAC\)](#) sebagai gantinya.
- [Dukungan SSL khusus IP khusus](#) - Hanya sni-only metode yang didukung.
- [AWS WAF Classic \(V1\) web ACL](#) — Hanya ACL web AWS WAF V2 yang didukung.
- [Pencatatan standar \(warisan\)](#)
- [TTL minimum](#)
- [TTL standar](#)
- [TTL maksimum](#)
- [ForwardedValues](#)
- [PriceClass](#)
- [Penandatanganan Terpercaya](#)

- [Streaming lancar](#)
- [AWS Identity and Access Management \(IAM\) sertifikat server](#)
- [Alamat IP khusus](#)
- [Versi Protokol Minimum SSLv3](#)

Pengaturan berikut tidak dapat dikonfigurasi dalam distribusi multi-tenant atau penyewa distribusi. Sebagai gantinya, tetapkan nilai yang Anda inginkan dalam grup koneksi. Semua penyewa distribusi yang terkait dalam grup koneksi akan menggunakan pengaturan ini. Untuk informasi selengkapnya, lihat [Buat grup koneksi khusus \(opsional\)](#).

- [Aktifkan IPv6 \(permintaan penampil\)](#)
- [Daftar IP statis anycast](#)

Kustomisasi penyewa distribusi

Saat menggunakan distribusi multi-tenant, penyewa distribusi Anda mewarisi konfigurasi distribusi multi-tenant. Namun, Anda dapat menyesuaikan beberapa pengaturan di tingkat penyewa distribusi.

Anda dapat menyesuaikan yang berikut ini:

- **Parameter** — Parameter adalah pasangan kunci-nilai yang dapat Anda gunakan untuk domain asal atau jalur asal. Lihat [Bagaimana parameter bekerja dengan penyewa distribusi](#).
- **AWS WAF web ACL (V2)** - Anda dapat menentukan ACL web terpisah untuk penyewa distribusi, yang akan mengganti ACL web yang digunakan untuk distribusi multi-penyewa. Anda juga dapat menonaktifkan pengaturan ini untuk penyewa distribusi tertentu, yang berarti penyewa distribusi tidak akan mewarisi perlindungan ACL web dari distribusi multi-penyewa. Untuk informasi selengkapnya, lihat [AWS WAF web ACL](#).
- **Pembatasan geografis** — Pembatasan geografis yang Anda tentukan untuk penyewa distribusi akan mengesampingkan batasan geografis apa pun untuk distribusi multi-penyewa. Misalnya, jika Anda memblokir Jerman (DE) dalam distribusi multi-penyewa Anda, semua penyewa distribusi terkait juga akan memblokir DE. Namun, jika Anda mengizinkan DE untuk penyewa distribusi tertentu, pengaturan penyewa distribusi tersebut akan mengganti pengaturan untuk distribusi multi-penyewa. Untuk informasi selengkapnya, lihat [Batasi distribusi geografis konten Anda](#).
- **Jalur pembatalan** - Tentukan jalur file ke konten yang ingin Anda batalkan untuk penyewa distribusi. Untuk informasi selengkapnya, lihat [Membatalkan file](#).

- Sertifikat TLS kustom — AWS Certificate Manager (ACM) sertifikat yang Anda tentukan untuk penyewa distribusi adalah tambahan untuk sertifikat yang disediakan dalam distribusi multi-penyewa. Namun, jika domain yang sama dicakup oleh distribusi multi-penyewa dan sertifikat penyewa distribusi, maka sertifikat penyewa digunakan. Untuk informasi selengkapnya, lihat [Minta sertifikat untuk penyewa CloudFront distribusi Anda](#).
- Nama domain — Anda harus menentukan setidaknya satu nama domain per penyewa distribusi.

Bagaimana parameter bekerja dengan penyewa distribusi

Parameter adalah pasangan kunci-nilai yang dapat Anda gunakan untuk nilai placeholder. Tentukan parameter yang ingin Anda gunakan dalam distribusi multi-tenant Anda dan tentukan apakah parameter tersebut diperlukan.

Ketika Anda menentukan parameter dalam distribusi multi-tenant Anda, Anda memilih apakah parameter tersebut harus dimasukkan pada tingkat penyewa distribusi.

- Jika Anda menentukan parameter seperti yang dipersyaratkan dalam distribusi multi-penyewa, maka parameter tersebut harus dimasukkan pada tingkat penyewa distribusi. (Mereka tidak diwariskan).
- Jika parameter tidak diperlukan, maka Anda dapat memberikan nilai default dalam distribusi multi-penyewa yang diwarisi oleh penyewa distribusi.

Anda dapat menggunakan parameter dalam properti berikut:

- Nama domain asal
- Jalur asal

Dalam distribusi multi-tenant, Anda dapat menentukan hingga dua parameter untuk masing-masing properti sebelumnya.

Parameter contoh

Lihat contoh berikut untuk menggunakan parameter untuk nama domain dan jalur asal.

Parameter nama domain

Dalam konfigurasi distribusi multi-tenant, Anda dapat menentukan parameter untuk nama domain asal seperti contoh berikut:

Amazon S3

- `{{parameter1}}.amzn-s3-demo-logging-bucket.s3.us-east-1.amazonaws.com`
- `{{parameter1}}-amzn-s3-demo-logging-bucket.s3.us-east-1.amazonaws.com`

Asal kustom

- `{{parameter1}}.lambda-url.us-east-1.on.aws`
- `{{parameter1}}.mediapackagev2.ap-south-1.amazonaws.com`

Saat Anda membuat penyewa distribusi, tentukan nilai yang akan digunakan *parameter1*.

```
"Parameters": [  
  {  
    "Name": "parameter1",  
    "Value": "mycompany-website"  
  }  
]
```

Menggunakan contoh sebelumnya yang ditentukan dalam distribusi multi-tenant, nama domain asal untuk penyewa distribusi menyelesaikan sebagai berikut:

- `mycompany-website.amzn-s3-demo-bucket3.s3.us-east-1.amazonaws.com`
- `mycompany-website-amzn-s3-demo-bucket3.s3.us-east-1.amazonaws.com`
- `mycompany-website.lambda-url.us-east-1.on.aws`
- `mycompany-website.mediapackagev2.ap-south-1.amazonaws.com`

Parameter jalur asal

Demikian pula, Anda dapat menentukan parameter untuk jalur asal dalam distribusi multi-penyewa seperti contoh berikut:

- `/{{parameter2}}`
- `/{{parameter2}}/test`
- `/public/{{parameter2}}/test`
- `/search?name={{parameter2}}`

Saat Anda membuat penyewa distribusi, tentukan nilai yang akan digunakan *parameter2*.

```
"Parameters": [  
  {  
    "Name": "parameter2",  
    "Value": "myBrand"  
  }  
]
```

Menggunakan contoh sebelumnya yang ditentukan dalam distribusi multi-tenant, jalur asal untuk penyewa distribusi menyelesaikan sebagai berikut:

- */myBrand*
- */myBrand/test*
- */public/myBrand/test*
- */search?name=myBrand*

Example Contoh

Anda ingin membuat beberapa situs web (penyewa) untuk pelanggan Anda, dan Anda perlu memastikan bahwa setiap sumber penyewa distribusi menggunakan nilai yang benar.

1. Anda membuat distribusi multi-tenant dan menyertakan dua parameter untuk konfigurasi penyewa distribusi.
2. Untuk nama domain asal, Anda membuat parameter bernama *customer-name* dan menentukan bahwa itu diperlukan. Anda memasukkan parameter sebelum bucket S3, sehingga muncul sebagai:

```
{{customer-name}}.amzn-s3-demo-bucket3.s3.us-east-1.amazonaws.com.
```

3. Untuk jalur asal, Anda membuat parameter kedua bernama *my-theme*, dan menentukan bahwa itu opsional, dengan nilai default *basic*. Jalur asal Anda muncul sebagai: */{{my-theme}}*
4. Saat Anda membuat penyewa distribusi:
 - Untuk nama domain, Anda harus menentukan nilai untuk *customer-name*, karena itu ditandai sebagai diperlukan dalam distribusi multi-penyewa.
 - Untuk jalur asal, Anda dapat secara opsional menentukan nilai untuk *my-theme* atau menggunakan nilai default.

Minta sertifikat untuk penyewa CloudFront distribusi Anda

Saat Anda membuat penyewa distribusi, penyewa mewarisi sertifikat bersama AWS Certificate Manager (ACM) dari distribusi multi-penyewa. Sertifikat bersama ini menyediakan HTTPS untuk semua penyewa yang terkait dengan distribusi multi-penyewa.

Saat membuat atau memperbarui penyewa CloudFront distribusi untuk menambahkan domain, Anda dapat menambahkan CloudFront sertifikat terkelola dari ACM. CloudFront kemudian mendapatkan HTTP-validated sertifikat dari ACM atas nama Anda. Anda dapat menggunakan sertifikat ACM tingkat penyewa ini untuk konfigurasi domain kustom. CloudFront merampingkan alur kerja pembaruan untuk membantu menjaga sertifikat tetap mutakhir dan mengamankan pengiriman konten tanpa gangguan.

Note

Anda memiliki sertifikat, tetapi hanya dapat digunakan dengan CloudFront sumber daya dan kunci pribadi tidak dapat diekspor.

Anda dapat meminta sertifikat saat membuat atau memperbarui penyewa distribusi.

Topik

- [Tambahkan domain dan sertifikat \(penyewa distribusi\)](#)
- [Pengaturan domain lengkap](#)
- [Arahkan domain ke CloudFront](#)
- [Pertimbangan domain \(penyewa distribusi\)](#)
- [Domain wildcard \(penyewa distribusi\)](#)

Tambahkan domain dan sertifikat (penyewa distribusi)

Prosedur berikut menunjukkan cara menambahkan domain dan memperbarui sertifikat untuk penyewa distribusi.

Untuk menambahkan domain dan sertifikat (penyewa distribusi)

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di bawah SaaS, pilih Penyewa distribusi.

3. Cari penyewa distribusi. Gunakan menu tarik-turun di bilah pencarian untuk memfilter berdasarkan domain, nama, ID distribusi, ID sertifikat, ID grup koneksi, atau ID ACL web.
4. Pilih nama penyewa distribusi.
5. Untuk Domain, pilih Kelola domain.
6. Untuk Sertifikat, pilih apakah Anda menginginkan sertifikat TLS Kustom untuk penyewa distribusi Anda. Sertifikat memverifikasi apakah Anda berwenang untuk menggunakan nama domain. Sertifikat harus ada di Wilayah AS Timur (Virginia N.).
7. Untuk Domain, pilih Tambahkan domain dan masukkan nama domain. Bergantung pada domain Anda, pesan berikut akan muncul di bawah nama domain yang Anda masukkan.
 - Domain ini dicakup oleh sertifikat.
 - Domain ini dicakup oleh sertifikat, menunggu validasi.
 - Domain ini tidak dicakup oleh sertifikat. (Ini berarti Anda harus memverifikasi kepemilikan domain.)
8. Pilih Perbarui penyewa distribusi.

Pada halaman detail penyewa, di bawah Domain, Anda dapat melihat bidang berikut:

- Kepemilikan domain — Status kepemilikan domain. Sebelum CloudFront dapat menyajikan konten, kepemilikan domain Anda harus diverifikasi dengan menggunakan validasi sertifikat TLS.
 - Status DNS — Catatan DNS domain Anda harus mengarah CloudFront ke rute lalu lintas dengan benar.
9. Jika kepemilikan domain Anda tidak diverifikasi, pada halaman detail penyewa, di bawah Domain, pilih Selesaikan pengaturan domain, lalu selesaikan prosedur berikut untuk mengarahkan catatan DNS ke nama domain Anda CloudFront .

Pengaturan domain lengkap

Ikuti prosedur ini untuk memverifikasi bahwa Anda memiliki domain untuk penyewa distribusi Anda. Tergantung pada domain Anda, pilih salah satu prosedur berikut.

Note

Jika domain Anda sudah ditunjuk CloudFront dengan catatan alias Amazon Route 53, Anda harus menambahkan catatan DNS TXT Anda dengan `_cf-challenge.` di depan nama

domain. Catatan TXT ini memverifikasi bahwa nama domain Anda ditautkan. CloudFront Ulangi langkah ini untuk setiap domain. Berikut ini menunjukkan cara memperbarui catatan TXT Anda:

- Nama rekam: `_cf-challenge.DomainName`
- Jenis rekaman: TXT
- Nilai rekam: `CloudFrontRoutingEndpoint`

Misalnya, catatan TXT Anda mungkin terlihat seperti: `_cf-challenge.example.com TXT d111111abcdef8.cloudfront.net`

Anda dapat menemukan titik akhir CloudFront perutean di konsol di halaman detail penyewa distribusi atau menggunakan tindakan [ListConnectionGroups](#) API di Referensi CloudFront API Amazon untuk menemukannya.

Tip

Jika Anda penyedia SaaS dan Anda ingin mengizinkan penerbitan sertifikat tanpa mengharuskan pelanggan Anda (penyewa) untuk menambahkan catatan TXT langsung ke DNS mereka, lakukan hal berikut:

1. Jika Anda memiliki `domainexample-saas-provider.com`, tetapkan subdomain ke penyewa Anda, seperti `customer-123.example-saas-provider.com`
2. Di DNS Anda, tambahkan catatan `_cf-challenge.customer-123.example-saas-provider.com TXT d111111abcdef8.cloudfront.net` TXT ke konfigurasi DNS Anda.
3. Selanjutnya, pelanggan Anda (penyewa) kemudian dapat memperbarui catatan DNS mereka sendiri untuk memetakan nama domain mereka ke subdomain yang Anda berikan.

```
www.customer-domain.com CNAME customer-123.example-saas-provider.com
```

I have existing traffic

Pilih opsi ini jika domain Anda tidak dapat mentolerir downtime. Anda harus memiliki akses ke origin/web server Anda. Gunakan prosedur berikut untuk memvalidasi kepemilikan domain.

Untuk menyelesaikan pengaturan domain saat Anda memiliki lalu lintas yang ada

1. Untuk Tentukan lalu lintas web Anda, pilih Saya memiliki lalu lintas yang ada dan kemudian pilih Berikutnya.
2. Untuk Verifikasi kepemilikan domain, pilih salah satu opsi berikut:
 - Gunakan sertifikat yang ada — Cari sertifikat ACM yang ada atau masukkan sertifikat ARN yang mencakup domain yang terdaftar.
 - Pengunggahan file manual - Pilih apakah Anda memiliki akses langsung untuk mengunggah file ke server web Anda.

Untuk setiap domain, buat file teks biasa yang berisi token validasi Anda dari lokasi Token dan unggah ke asal Anda di jalur File yang ditentukan di server yang ada. Jalur ke file ini mungkin terlihat seperti contoh berikut: `/.well-known/pki-validation/acm_9c2a7b2ec0524d09fa6013efb73ad123.txt`. Setelah Anda menyelesaikan langkah itu, ACM memverifikasi token dan kemudian mengeluarkan sertifikat TLS untuk domain tersebut.

- Pengalihan HTTP - Pilih apakah Anda tidak memiliki akses langsung untuk mengunggah file ke server web Anda, atau jika Anda menggunakan layanan CDN atau proxy.

Untuk setiap domain, buat pengalihan 301 di server yang ada. Salin jalur terkenal di bawah Redirect dari dan arahkan ke titik akhir sertifikat yang ditentukan di bawah Redirect to. Pengalihan Anda mungkin terlihat seperti contoh berikut:

```
If the URL matches: example.com/.well-known/pki-validation/
leabe938a4fe077b31e1ff62b781c123.txt
Then the settings are:Forwarding URL
Then 301 Permanent Redirect:To validation.us-east-1.acm-
validations.aws/123456789012/.well-known/pki-validation/
leabe938a4fe077b31e1ff62b781c123.txt
```

 Note

Anda dapat memilih Periksa status sertifikat untuk memverifikasi kapan ACM mengeluarkan sertifikat untuk domain.

3. Pilih Berikutnya.

4. Selesaikan langkah-langkahnya [Arahkan domain ke CloudFront](#).

I don't have traffic

Pilih opsi ini jika Anda menambahkan domain baru. CloudFront akan mengelola validasi sertifikat untuk Anda.

Untuk menyelesaikan pengaturan domain jika Anda tidak memiliki lalu lintas

1. Untuk Tentukan lalu lintas web Anda, pilih Saya belum memiliki lalu lintas.
2. Untuk setiap nama domain, selesaikan langkah-langkahnya [Arahkan domain ke CloudFront](#).
3. Setelah memperbarui catatan DNS untuk setiap nama domain, pilih Berikutnya.
4. Tunggu sertifikat dikeluarkan.

Note

Anda dapat memilih Periksa status sertifikat untuk memverifikasi kapan ACM mengeluarkan sertifikat untuk domain.

5. Pilih Kirim.

Arahkan domain ke CloudFront

Perbarui catatan DNS Anda untuk merutekan lalu lintas dari setiap domain ke titik akhir CloudFront perutean. Anda dapat memiliki beberapa nama domain, tetapi semuanya harus menyelesaikan ke titik akhir ini.

Untuk mengarahkan domain ke CloudFront

1. Salin nilai titik akhir CloudFront perutean, seperti d111111abcdef8.cloudfront.net.
2. Perbarui catatan DNS Anda untuk merutekan lalu lintas dari setiap domain ke titik akhir CloudFront perutean.
 1. Masuk ke registrar domain atau konsol manajemen penyedia DNS Anda.
 2. Arahkan ke bagian manajemen DNS untuk domain Anda.
 - Untuk subdomain — Buat catatan CNAME. Contoh:
 - Nama — Subdomain Anda (seperti www atau app)

- Nilai/Target - Titik akhir CloudFront perutean
 - Jenis rekaman - CNAME
 - TTL - 3600 (atau apa pun yang sesuai untuk kasus penggunaan Anda)
 - Untuk apex/root domain — Ini memerlukan konfigurasi DNS yang unik, karena catatan CNAME standar tidak dapat digunakan pada tingkat domain root atau apex. Karena sebagian besar penyedia DNS tidak mendukung catatan ALIAS, sebaiknya buat catatan ALIAS di Route 53. Contoh:
 - Nama — domain puncak Anda (seperti `example.com`)
 - Jenis rekaman - A
 - Alias - Ya
 - Target alias - Titik akhir CloudFront perutean Anda
 - Kebijakan perutean — Sederhana (atau apa pun yang sesuai untuk kasus penggunaan Anda)
3. Verifikasi bahwa perubahan DNS telah disebarkan. (Ini biasanya terjadi ketika TTL kedaluwarsa. Terkadang bisa memakan waktu 24-48 jam.) Gunakan alat seperti `dig` atau `nslookup`.

```
dig www.example.com
# Should eventually return a CNAME pointing to your CloudFront routing endpoint
```

3. Kembali ke CloudFront konsol dan pilih Kirim. Saat domain Anda aktif, CloudFront perbarui status domain untuk menunjukkan bahwa domain Anda siap melayani lalu lintas.

Untuk informasi selengkapnya, lihat dokumentasi untuk penyedia DNS Anda:

- [Cloudflare](#)
- [ClouDNS](#)
- [DNSimple](#)
- [Gandi.net](#)
- [GoDaddy](#)
- [Google Cloud DNS](#)
- [Namecheap](#)

Pertimbangan domain (penyewa distribusi)

Ketika domain aktif, kontrol domain telah dibuat dan CloudFront akan menanggapi semua permintaan penampil ke domain ini. Setelah diaktifkan, domain tidak dapat dinonaktifkan atau diubah menjadi status tidak aktif. Domain tidak dapat dikaitkan dengan CloudFront sumber daya lain saat sudah digunakan. Untuk mengaitkan domain dengan distribusi lain, gunakan [UpdateDomainAssociation](#) permintaan untuk memindahkan domain dari satu CloudFront sumber daya ke sumber daya lainnya.

Ketika domain tidak aktif, tidak CloudFront akan menanggapi permintaan penampil ke domain. Meskipun domain tidak aktif, perhatikan hal berikut:

- Jika Anda memiliki permintaan sertifikat yang tertunda, CloudFront akan menanggapi permintaan untuk jalur terkenal. Sementara permintaan tertunda, domain tidak dapat dikaitkan dengan CloudFront sumber daya lain.
- Jika Anda tidak memiliki permintaan sertifikat yang tertunda, CloudFront tidak akan menanggapi permintaan untuk domain tersebut. Anda dapat mengaitkan domain dengan CloudFront sumber daya lain.
- Anda hanya dapat memiliki satu permintaan sertifikat yang tertunda per penyewa distribusi. Sebelum Anda dapat meminta sertifikat lain untuk domain tambahan, Anda harus membatalkan permintaan tertunda yang ada. Membatalkan permintaan sertifikat yang ada tidak menghapus sertifikat ACM terkait. Anda dapat menghapusnya dengan menggunakan ACM API.
- Jika Anda menerapkan sertifikat baru ke penyewa distribusi Anda, ini akan memisahkan sertifikat sebelumnya. Anda dapat menggunakan kembali sertifikat untuk menutupi domain untuk penyewa distribusi lain.

Seperti halnya perpanjangan untuk DNS-validated sertifikat, Anda akan diberi tahu ketika perpanjangan sertifikat berhasil. Namun, Anda tidak perlu melakukan hal lain. CloudFront akan mengelola perpanjangan sertifikat untuk domain Anda secara otomatis.

Note

Anda tidak perlu memanggil operasi ACM API untuk membuat atau memperbarui sumber daya sertifikat Anda. Anda dapat mengelola sertifikat dengan menggunakan operasi [CreateDistributionTenant](#) dan [UpdateDistributionTenant](#) API untuk menentukan detail permintaan sertifikat terkelola Anda.

 Note

Kompatibilitas pembatasan geografis — validasi sertifikat CloudFront terkelola kompatibel dengan batasan geografis. Bila Anda mengaktifkan pembatasan geografis pada distribusi yang menggunakan sertifikat CloudFront terkelola, permintaan validasi dikecualikan dari aturan pembatasan geografis. Otoritas Sertifikat memverifikasi kepemilikan domain dari berbagai perspektif jaringan yang didistribusikan secara geografis (Multi Perspective Issuance Corroboration) untuk melindungi dari pembajakan BGP. CloudFront memastikan permintaan validasi ini berhasil terlepas dari negara mana yang Anda blokir atau izinkan. Tidak ada tindakan yang diperlukan untuk menggunakan fungsi ini. Jika sebelumnya Anda menghapus batasan geografis sebagai solusi untuk kegagalan validasi sertifikat, Anda dapat menerapkannya kembali dengan aman.

Domain wildcard (penyewa distribusi)

Domain wildcard didukung untuk penyewa distribusi dalam situasi berikut:

- Bila wildcard disertakan dalam sertifikat bersama yang diwarisi dari distribusi multi-penyewa induk
- Bila Anda menggunakan sertifikat TLS kustom yang valid untuk penyewa distribusi Anda

Buat grup koneksi khusus (opsional)

Secara default, CloudFront buat grup koneksi untuk Anda saat Anda membuat distribusi multi-tenant. Grup koneksi mengontrol cara permintaan penampil untuk konten terhubung CloudFront.

Kami menyarankan Anda menggunakan grup koneksi default. Namun, jika Anda perlu mengisolasi aplikasi perusahaan atau mengelola grup penyewa distribusi secara terpisah, Anda dapat memilih untuk membuat grup koneksi khusus. Misalnya, Anda mungkin perlu memindahkan penyewa distribusi ke grup koneksi terpisah jika mengalami serangan DDoS. Dengan cara ini, Anda dapat melindungi penyewa distribusi lainnya dari dampak.

Buat grup koneksi khusus (opsional)

Secara opsional, Anda dapat memilih untuk membuat grup koneksi khusus untuk penyewa distribusi Anda.

Untuk membuat grup koneksi kustom (opsional)

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Pada panel navigasi, silakan pilih Pengaturan.
3. Aktifkan pengaturan grup Koneksi.
4. Di panel navigasi, pilih Grup koneksi, lalu pilih Buat grup koneksi.
5. Untuk nama grup Koneksi, masukkan nama untuk grup koneksi. Anda tidak dapat memperbarui nama ini setelah membuat grup koneksi.
6. Untuk IPv6, tentukan apakah Anda ingin mengaktifkan protokol IP ini. Untuk informasi selengkapnya, lihat [Aktifkan IPv6 \(permintaan penampil\)](#).
7. Untuk daftar IP statis Anycast, tentukan apakah Anda ingin mengirimkan lalu lintas ke penyewa distribusi Anda dari satu set alamat IP. Untuk informasi selengkapnya, lihat [Daftar IP statis Anycast](#).
8. (Opsional) Tambahkan tag ke grup koneksi Anda.
9. Pilih Buat grup koneksi.

Ketika grup koneksi Anda dibuat, Anda dapat menemukan pengaturan yang Anda tentukan, dan juga ARN dan titik akhir.

- ARN terlihat seperti contoh berikut: `arn:aws:cloudfront::123456789012:connection-group/cg_2uVbA9KeWaADTbKzhj91cKDoM25`
- Titik akhir terlihat seperti contoh berikut: `d1111111abcdef8.cloudfront.net`

Anda dapat mengedit atau menghapus grup koneksi kustom Anda setelah Anda membuatnya. Sebelum Anda dapat menghapus grup koneksi, Anda harus terlebih dahulu menghapus semua penyewa distribusi terkait darinya. Anda tidak dapat menonaktifkan atau menghapus grup koneksi default yang CloudFront dibuat untuk Anda saat Anda membuat distribusi multi-penyewa.

Important

Jika Anda mengubah grup koneksi untuk penyewa distribusi, CloudFront akan terus membawa lalu lintas untuk penyewa distribusi, tetapi dengan peningkatan latensi. Kami menyarankan Anda memperbarui catatan DNS untuk penyewa distribusi untuk menggunakan titik akhir CloudFront perutean dari grup koneksi baru.

Sampai Anda memperbarui catatan DNS, CloudFront akan merutekan berdasarkan pengaturan yang ditentukan untuk titik akhir perutean yang saat ini ditunjuk situs web dengan DNS. Misalnya, asumsikan bahwa grup koneksi default Anda tidak menggunakan IP statis Anycast tetapi grup koneksi khusus Anda yang baru melakukannya. Anda harus memperbarui catatan DNS sebelum CloudFront akan menggunakan IP statis Anycast untuk penyewa distribusi di grup koneksi kustom Anda.

Migrasi ke distribusi multi-tenant

Jika Anda memiliki distribusi CloudFront standar dan ingin bermigrasi ke distribusi multi-tenant, ikuti langkah-langkah berikut.

Untuk bermigrasi dari distribusi standar ke distribusi multi-tenant

1. Tinjau [Fitur yang tidak didukung](#).
2. Buat distribusi multi-tenant dengan konfigurasi yang sama dengan distribusi standar Anda, dikurangi fitur yang tidak didukung. Untuk informasi selengkapnya, lihat [Buat CloudFront distribusi di konsol](#).
3. Buat penyewa distribusi dan tambahkan nama domain alternatif yang Anda miliki.

Warning

Jangan gunakan nama domain saat ini yang terkait dengan distribusi standar Anda. Tambahkan domain placeholder sebagai gantinya. Anda akan memindahkan domain Anda nanti. Untuk informasi selengkapnya tentang membuat penyewa distribusi, lihat [Buat CloudFront distribusi di konsol](#).

4. Berikan sertifikat yang ada untuk domain penyewa distribusi. Ini adalah sertifikat yang akan mencakup domain placeholder dan domain yang ingin Anda pindahkan.
5. Salin titik akhir CloudFront perutean dari halaman detail penyewa distribusi di konsol. Atau, temukan dengan menggunakan tindakan [ListConnectionGroups](#) API di Referensi Amazon CloudFront API.
6. Untuk memverifikasi kepemilikan domain, buat catatan DCV TXT dengan awalan garis bawah (_) yang menunjuk ke titik akhir CloudFront perutean untuk penyewa distribusi Anda. Untuk informasi selengkapnya, lihat [Arahkan domain ke CloudFront](#).

7. Ketika perubahan Anda telah menyebar, perbarui penyewa distribusi Anda untuk menggunakan domain yang sebelumnya Anda gunakan untuk distribusi standar Anda.
 - Konsol — Untuk petunjuk terperinci, lihat [Tambahkan domain dan sertifikat \(penyewa distribusi\)](#).
 - API — Gunakan tindakan [UpdateDomainAssociation](#) API di Referensi Amazon CloudFront API.

 Important

Ini mengatur ulang kunci cache untuk konten Anda. Setelah ini, CloudFront mulai caching konten Anda menggunakan kunci cache baru. Untuk informasi selengkapnya, lihat [Memahami kunci cache](#).

8. Perbarui data DNS Anda untuk mengarahkan domain Anda ke titik akhir CloudFront perutean untuk penyewa distribusi Anda. Setelah Anda menyelesaikan langkah ini, domain Anda akan siap melayani lalu lintas ke penyewa distribusi Anda. Untuk informasi selengkapnya, lihat [Arahkan domain ke CloudFront](#).
9. (Opsional) Setelah berhasil memigrasikan domain ke penyewa distribusi, Anda dapat menggunakan sertifikat CloudFront terkelola lain yang mencakup nama domain untuk penyewa distribusi. Untuk meminta sertifikat terkelola, buat catatan TXT terpisah untuk menerbitkan sertifikat dan ikuti langkah-langkahnya di sini. [Pengaturan domain lengkap](#)

Buat distribusi

Topik ini menjelaskan cara menggunakan CloudFront konsol untuk membuat distribusi.

Ikhtisar

1. Buat satu atau beberapa bucket Amazon S3, atau konfigurasi server HTTP sebagai server asal Anda. Asal adalah lokasi tempat Anda menyimpan versi asli konten Anda. Ketika CloudFront mendapat permintaan untuk file Anda, ia pergi ke asal untuk mendapatkan file yang didistribusikan di lokasi tepi. Anda dapat menggunakan kombinasi bucket Amazon S3 dan server HTTP sebagai server asal Anda.
 - Jika Anda menggunakan Amazon S3, nama bucket Anda harus semua huruf kecil dan tidak dapat berisi spasi.

- Jika Anda menggunakan server Amazon EC2 atau asal kustom lainnya, tinjau. [Gunakan Amazon EC2 \(atau asal kustom lainnya\)](#)
 - Untuk jumlah maksimum asal saat ini yang dapat Anda buat untuk distribusi, atau untuk meminta kuota yang lebih tinggi, lihat [Kuota umum di distribusi](#).
2. Unggah konten Anda ke server asal Anda. Anda membuat objek dapat dibaca publik, atau Anda dapat menggunakan URL yang CloudFront ditandatangani untuk membatasi akses ke konten Anda.

 Important

Anda bertanggung jawab untuk memastikan keamanan server asal Anda. Anda harus memastikan bahwa CloudFront memiliki izin untuk mengakses server dan bahwa pengaturan keamanan melindungi konten Anda.

3. Buat CloudFront distribusi Anda:
- Untuk prosedur terperinci yang membuat distribusi di CloudFront konsol, lihat [Buat CloudFront distribusi di konsol](#).
 - Untuk informasi tentang membuat distribusi menggunakan CloudFront API, lihat [CreateDistribution](#) di Referensi Amazon CloudFront API.
4. (Opsional) Jika Anda menggunakan CloudFront konsol untuk membuat distribusi, buat lebih banyak perilaku cache atau asal untuk distribusi. Untuk informasi lebih lanjut tentang perilaku dan asal-usul, lihat [Untuk memperbarui distribusi multi-tenant](#).
5. Uji distribusi Anda. Untuk informasi lebih lanjut tentang pengujian, lihat [Uji distribusi](#).
6. Kembangkan situs web atau aplikasi Anda untuk mengakses konten Anda menggunakan nama domain yang CloudFront dikembalikan setelah Anda membuat distribusi di Langkah 3. Misalnya, jika CloudFront mengembalikan `d111111abcdef8.cloudfront.net` sebagai nama domain untuk distribusi Anda, URL untuk file di bucket `image.jpg` Amazon S3 atau di direktori root pada server HTTP adalah `https://d111111abcdef8.cloudfront.net/image.jpg`

Jika Anda menentukan satu atau lebih nama domain alternatif (CNAMEs) ketika membuat distribusi Anda, Anda dapat menggunakan nama domain Anda sendiri. Dalam hal ini, URL untuk `image.jpg` dapat menjadi `https://www.example.com/image.jpg`.

Perhatikan hal-hal berikut:

- Jika Anda ingin menggunakan URL yang ditandatangani untuk membatasi akses ke konten Anda, lihat [Sajikan konten pribadi dengan URL yang ditandatangani dan cookie yang ditandatangani](#).
- Jika Anda ingin menyajikan konten terkompresi, lihat [Sajikan file terkompresi](#).
- Untuk informasi tentang perilaku CloudFront permintaan dan respons untuk Amazon S3 dan asal kustom, lihat. [Perilaku permintaan dan respons](#)

Topik

- [Buat CloudFront distribusi di konsol](#)
- [Nilai yang CloudFront ditampilkan di konsol](#)
- [Tautan tambahan](#)
- [Tambahkan domain ke distribusi CloudFront standar Anda](#)

Buat CloudFront distribusi di konsol

Saat Anda membuat distribusi, CloudFront mengonfigurasi setelan distribusi untuk Anda, berdasarkan jenis asal konten Anda. Untuk detail selengkapnya tentang pengaturan yang telah dikonfigurasi sebelumnya, lihat [Referensi pengaturan distribusi yang telah dikonfigurasi sebelumnya](#). Anda juga dapat membuat distribusi multi-tenant dengan pengaturan yang dapat digunakan kembali di beberapa penyewa distribusi. Untuk informasi selengkapnya, lihat [Memahami cara kerja distribusi multi-tenant](#). Atau, Anda dapat mengonfigurasi pengaturan distribusi Anda sendiri secara manual.

Multi-tenant

Untuk membuat distribusi multi-tenant

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Distribusi, lalu pilih Buat distribusi.
3. Pilih Multi-tenant arsitektur, Berikutnya.
4. Masukkan nama Distribusi untuk distribusi multi-tenant. Nama akan muncul sebagai nilai untuk Name kunci. Anda dapat mengubah nilai ini nanti. Anda dapat menambahkan hingga 50 tag untuk distribusi multi-tenant Anda. Untuk informasi selengkapnya, lihat [Menandai CloudFront sumber daya](#).

5. (Opsional) Untuk sertifikat Wildcard, pilih sertifikat AWS Certificate Manager (ACM) yang akan mencakup semua subdomain di bawah domain root, seperti. **.example.com* Sertifikat harus berada di Wilayah AS Timur (Virginia N.).
6. Pilih Berikutnya.
7. Pada halaman Tentukan asal, pilih jenis asal yang CloudFront akan mendapatkan konten Anda. CloudFront akan menggunakan pengaturan yang disarankan untuk jenis asal tersebut untuk distribusi multi-penyewa Anda. Untuk informasi selengkapnya tentang pengaturan yang disarankan, lihat [Referensi pengaturan distribusi yang telah dikonfigurasi sebelumnya](#).
8. Untuk Origin, di bawah tipe asal yang Anda pilih, pilih atau masukkan asal yang akan digunakan.
9. Untuk jalur Origin, masukkan karakter garis miring (/) maju, diikuti oleh jalur asal.
10. (Opsional) Untuk menambahkan parameter, pilih Sisipkan parameter untuk nama domain asal atau jalur asal. Anda dapat memasukkan hingga dua parameter untuk setiap bidang.
 - a. Pilih Buat parameter baru.
 - b. Pada kotak dialog Buat parameter baru, untuk nama Parameter, masukkan nama unik untuk parameter dan, secara opsional, deskripsi.
 - c. Untuk parameter Diperlukan, pilih kotak centang untuk membuat nilai parameter ini diperlukan pada tingkat penyewa distribusi. Jika tidak diperlukan, masukkan nilai Default yang akan diwarisi oleh penyewa distribusi.
 - d. Pilih Buat parameter. Parameter ini muncul di bidang yang sesuai.
11. Untuk Opsi, pilih salah satu opsi berikut:
 - Gunakan pengaturan asal yang disarankan — Gunakan pengaturan cache dan asal yang direkomendasikan default untuk jenis asal yang Anda pilih.
 - Sesuaikan pengaturan asal - Sesuaikan pengaturan cache dan asal. Jika Anda memilih opsi ini, tentukan nilai Anda sendiri yang muncul.
12. Pilih Berikutnya.
13. Pada halaman Aktifkan perlindungan keamanan, pilih apakah akan mengaktifkan perlindungan AWS WAF keamanan. Anda dapat menyesuaikan ACL web untuk penyewa distribusi tertentu nanti. Untuk informasi selengkapnya, lihat [AWS WAF Aktifkan distribusi baru](#).
14. Pilih Berikutnya, Buat distribusi.

15. Pada halaman Distribusi, distribusi multi-tenant Anda muncul dalam daftar sumber daya. Anda dapat memilih dropdown Semua distribusi untuk difilter berdasarkan distribusi standar atau distribusi multi-penyewa. Anda juga dapat memilih kolom Type untuk memfilter berdasarkan distribusi standar atau multi-tenant.

Secara default, CloudFront buat grup koneksi untuk Anda. Grup koneksi mengontrol cara permintaan penampil untuk konten terhubung CloudFront. Anda dapat menyesuaikan beberapa pengaturan perutean di grup koneksi. Untuk informasi selengkapnya, lihat [Memahami cara kerja distribusi multi-tenant](#).

Anda dapat membuat penyewa distribusi tambahan menggunakan distribusi multi-penyewa sebagai templat.

Untuk membuat penyewa distribusi

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, lakukan salah satu hal berikut:
 - Pilih Distribusi, pilih distribusi multi-tenant, lalu pilih Buat penyewa.
 - Pilih Penyewa distribusi dan kemudian pilih Buat penyewa.
3. Untuk nama penyewa distribusi, masukkan nama. Nama harus unik di Anda Akun AWS dan tidak dapat diubah setelah Anda membuatnya.
4. Untuk distribusi Template, pilih ID distribusi multi-tenant dari daftar.
5. Untuk Kelola tag, tambahkan hingga 50 pasangan nilai kunci untuk penyewa distribusi. Untuk mengetahui informasi selengkapnya, lihat [Menandai CloudFront sumber daya](#).
6. Pilih Berikutnya.
7. Pada halaman Tambah domain, untuk Sertifikat, pilih apakah Anda menginginkan sertifikat TLS Kustom untuk penyewa distribusi Anda. Sertifikat memverifikasi apakah Anda berwenang untuk menggunakan nama domain. Sertifikat harus ada di Wilayah AS Timur (Virginia N.).
8. Untuk Domain, masukkan nama domain Anda.

 Note

Jika Anda memasukkan nama domain yang tidak tercakup dalam sertifikat, Anda harus memverifikasi bahwa Anda memiliki domain tersebut. Anda masih dapat

membuat penyewa distribusi untuk saat ini dan memverifikasi kepemilikan domain nanti. Untuk informasi selengkapnya, lihat [Minta sertifikat untuk penyewa CloudFront distribusi Anda](#).

9. Pilih Berikutnya.
10. Pada halaman Tentukan parameter, parameter yang Anda tentukan dalam distribusi multi-penyewa muncul. Untuk parameter yang diperlukan, masukkan nilai di sebelah nama parameter dan simpan perubahan Anda.
11. Untuk menambahkan parameter lain, pilih Tambah parameter dan masukkan nama dan nilai.
12. Pilih Berikutnya.
13. (Opsional) Untuk kustomisasi Keamanan, jika Anda memilih untuk Mengganti pengaturan distribusi, pilih opsi untuk kasus penggunaan Anda.
14. (Opsional) Untuk kustomisasi pembatasan geografis, jika Anda memilih untuk Mengganti pengaturan distribusi, pilih jenis Pembatasan yang sesuai dan Negara untuk penyewa distribusi. Untuk informasi selengkapnya, lihat [Batasi distribusi geografis konten Anda](#).
15. Pilih Berikutnya.
16. Pilih Buat penyewa distribusi.

Anda dapat menemukan semua penyewa distribusi Anda di halaman Penyewa distribusi. Anda dapat memfilter dengan yang berikut ini:

Asosiasi

- ID Distribusi
- ID Sertifikat
- ID grup koneksi
- ID ACL Web

Sifat-sifat

- Nama
- Domain

Anda dapat mengedit penyewa distribusi Anda untuk menyesuaikan pengaturan tertentu. Untuk informasi selengkapnya, lihat [Kustomisasi penyewa distribusi](#).

Standard

Untuk membuat distribusi standar

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Distribusi, lalu pilih Buat distribusi.
3. Masukkan nama Distribusi untuk distribusi standar. Nama akan muncul sebagai nilai untuk Name kunci sebagai tag. Anda dapat mengubah nilai ini nanti. Anda dapat menambahkan hingga 50 tag untuk distribusi standar Anda. Untuk informasi selengkapnya, lihat [Menandai CloudFront sumber daya](#).
4. Pilih situs web atau aplikasi tunggal, Berikutnya.
5. (Opsional) Untuk pengaturan Domain, masukkan domain yang sudah terdaftar dengan Route 53 di Akun AWS, atau daftarkan domain baru. Selesaikan langkah-langkah pengaturan.
 - Jika domain Anda menggunakan penyedia DNS selain Route 53, Anda masih dapat menambahkan domain, tetapi Anda harus melakukannya setelah membuat distribusi. Lewati pengaturan domain untuk saat ini untuk melanjutkan pembuatan distribusi. Anda harus mengonfigurasi domain dan sertifikat TLS secara manual nanti. Untuk informasi selengkapnya, lihat [Tambahkan domain ke distribusi CloudFront standar Anda](#).
6. Pilih Berikutnya.
7. Pada halaman Tentukan asal, pilih jenis asal yang CloudFront akan mendapatkan konten Anda. CloudFront akan menggunakan pengaturan yang disarankan untuk jenis asal tersebut untuk distribusi Anda. Untuk informasi selengkapnya tentang pengaturan yang disarankan, lihat [Referensi pengaturan distribusi yang telah dikonfigurasi sebelumnya](#).
8. Untuk Origin, pilih atau masukkan asal Anda.
9. Untuk Pengaturan, pilih salah satu opsi berikut:
 - Gunakan pengaturan asal yang disarankan — Gunakan pengaturan cache dan asal yang direkomendasikan default untuk jenis asal yang Anda pilih.
 - Sesuaikan pengaturan asal - Sesuaikan pengaturan cache dan asal. Jika Anda memilih opsi ini, tentukan nilai Anda sendiri.
10. Pilih Berikutnya.

11. Pada halaman Aktifkan perlindungan keamanan, pilih apakah akan mengaktifkan perlindungan AWS WAF keamanan.
12. Pilih Berikutnya.
13. (Opsional) Jika Anda menggunakan Route 53 untuk domain Anda, Anda akan melihat halaman sertifikat TLS. Jika tidak CloudFront dapat menemukan sertifikat yang ada AWS Certificate Manager (ACM) untuk domain Anda di Akun AWS dalam us-east-1 Wilayah AWS, Anda dapat memilih untuk secara otomatis membuat sertifikat atau membuatnya secara manual. Setelah sertifikat dibuat, pilih Berikutnya.
14. Tinjau detail distribusi Anda dan pilih Buat distribusi.
15. Setelah CloudFront membuat distribusi Anda, nilai kolom Status untuk distribusi Anda akan berubah dari Deploying ke tanggal dan waktu distribusi diterapkan.

Nama domain yang CloudFront ditetapkan untuk distribusi Anda muncul dalam daftar distribusi. (Ini juga muncul di Umum untuk distribusi yang dipilih.)

 Tip

Anda dapat menggunakan nama domain alternatif, bukan nama yang diberikan kepada Anda oleh CloudFront, dengan mengikuti langkah-langkah di [Gunakan URL kustom dengan menambahkan nama domain alternatif \(CNames\)](#).

16. Ketika distribusi Anda digunakan, konfirmasi bahwa Anda dapat mengakses konten Anda dengan menggunakan CloudFront URL baru Anda (d111111abcdef8.cloudfront.net) atau CNAME. Untuk informasi selengkapnya, lihat [Uji distribusi](#).
17. Pastikan untuk memperbarui catatan DNS Anda untuk menunjuk CloudFront kapan Anda siap mengirim lalu lintas ke distribusi Anda. Untuk informasi selengkapnya, lihat [Arahkan domain ke CloudFront \(distribusi standar\)](#).

Nilai yang CloudFront ditampilkan di konsol

Saat Anda membuat distribusi baru atau memperbarui distribusi yang ada, CloudFront menampilkan informasi berikut di CloudFront konsol.

 Note

Penandatanganan tepercaya aktif, Akun AWS yang memiliki CloudFront key pair aktif dan dapat digunakan untuk membuat URL bertanda tangan yang valid, saat ini tidak terlihat di CloudFront konsol.

ID Distribusi

Saat Anda melakukan tindakan pada distribusi menggunakan CloudFront API, Anda menggunakan ID distribusi untuk menentukan distribusi mana yang akan digunakan, misalnya, EDFDVBD6EXAMPLE. Anda tidak dapat mengubah ID distribusi distribusi.

Penyebaran dan status

Saat Anda menerapkan distribusi, Anda melihat status Deploying di bawah kolom Terakhir dimodifikasi. Tunggu distribusi selesai digunakan dan pastikan kolom Status ditampilkan Diaktifkan. Untuk informasi selengkapnya, lihat [Status distribusi](#).

Terakhir dimodifikasi

Tanggal dan waktu distribusi terakhir diubah, menggunakan format ISO 8601, misalnya, 2012-05-19. T19:37:58Z Untuk informasi selengkapnya, lihat <https://www.w3.org/TR/NOTE-datetime>.

Nama domain

Anda menggunakan nama domain distribusi dalam tautan ke objek Anda. Misalnya, jika nama domain distribusi Anda adalah d111111abcdef8.cloudfront.net, tautan ke /images/image.jpg akan menjadi <https://d111111abcdef8.cloudfront.net/images/image.jpg>. Anda tidak dapat mengubah nama CloudFront domain untuk distribusi Anda. Untuk informasi selengkapnya tentang CloudFront URL untuk tautan ke objek Anda, lihat [Sesuaikan format URL untuk file di CloudFront](#).

Jika Anda menentukan satu atau beberapa nama domain alternatif (CNames), Anda dapat menggunakan nama domain Anda sendiri untuk tautan ke objek Anda alih-alih menggunakan nama CloudFront domain. Untuk informasi selengkapnya tentang CNAME, lihat [Nama domain alternatif \(CNames\)](#).

 Note

CloudFront Nama domain adalah unik. Nama domain distribusi Anda tidak pernah digunakan untuk distribusi sebelumnya dan tidak akan pernah digunakan kembali untuk distribusi lain di masa mendatang.

Tautan tambahan

Untuk informasi selengkapnya tentang membuat distribusi, lihat tautan berikut.

- Untuk mempelajari cara membuat distribusi yang menggunakan asal bucket Amazon Simple Storage Service (Amazon S3) dengan origin access control (OAC), lihat [Memulai dengan distribusi CloudFront standar](#)
- Untuk informasi tentang penggunaan CloudFront API untuk membuat distribusi, lihat [CreateDistribution](#) di Referensi CloudFront API Amazon.
- Untuk informasi tentang memperbarui distribusi (misalnya, untuk menambahkan perilaku cache ke distribusi standar, atau untuk menyesuaikan penyewa distribusi), lihat [Perbarui distribusi](#)
- Untuk melihat jumlah maksimum distribusi saat ini yang dapat Anda buat untuk setiap akun AWS , atau untuk meminta kuota yang lebih tinggi (sebelumnya disebut batas), lihat [Kuota umum di distribusi](#).

Tambahkan domain ke distribusi CloudFront standar Anda

Setelah Anda membuat distribusi CloudFront standar baru, Anda dapat menambahkan domain ke dalamnya. Secara opsional, Anda dapat mengatur domain Amazon Route 53 untuk distribusi standar saat Anda membuatnya. Untuk informasi selengkapnya, lihat [Buat CloudFront distribusi di konsol](#).

Menambahkan domain ke distribusi standar yang ada

Untuk menambahkan domain ke distribusi standar Anda

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Distribusi, lalu pilih ID distribusi.
3. Di bawah Pengaturan, Nama domain alternatif, pilih Tambahkan domain.

4. Masukkan hingga lima domain untuk dilayani.
5. Pilih Berikutnya.
6. Untuk sertifikat TLS, jika tidak CloudFront dapat menemukan sertifikat AWS Certificate Manager (ACM) yang ada untuk domain Anda di Akun AWS dalam us-east-1 Wilayah AWS, Anda dapat membuatnya.
 - Jika Anda menggunakan Amazon Route 53 (Route 53), CloudFront secara otomatis membuat sertifikat untuk Anda.
7. Ketika sertifikat Anda disediakan, Anda harus memperbarui catatan DNS Anda dengan penyedia DNS Anda untuk membuktikan kepemilikan domain. Kemudian, pilih Validasi sertifikat. Untuk informasi selengkapnya, lihat [Arahkan domain ke CloudFront \(distribusi standar\)](#).
 - Jika Anda menggunakan Route 53, CloudFront perbarui catatan DNS Anda untuk Anda.
8. Pilih Berikutnya.
9. Tinjau perubahan Anda dan pilih Tambahkan domain.
10. Sebelum Anda mengirim lalu lintas ke distribusi Anda, pastikan untuk memperbarui catatan DNS Anda untuk ditunjuk. CloudFront Untuk informasi selengkapnya, pilih Rutekan domain ke bagian Pengaturan CloudFront di halaman detail distribusi Anda.
 - Jika Anda menggunakan Route 53, Anda dapat CloudFront mengatur perutean DNS untuk Anda secara otomatis.

Arahkan domain ke CloudFront (distribusi standar)

Perbarui catatan DNS Anda untuk merutekan lalu lintas dari setiap domain ke nama CloudFront host. Anda dapat memiliki beberapa nama domain, tetapi semuanya harus menyelesaikan nama host ini.

Untuk mengarahkan domain ke CloudFront

1. Salin nilai CloudFront nama host, seperti d111111abcdef8.cloudfront.net.
2. Perbarui catatan DNS Anda untuk merutekan lalu lintas dari setiap domain ke nama CloudFront host.
 1. Masuk ke registrar domain atau konsol manajemen penyedia DNS Anda.
 2. Arahkan ke bagian manajemen DNS untuk domain Anda.
 - Untuk subdomain — Buat catatan CNAME. Contoh:

- Nama — Subdomain Anda (seperti `www` atau `app`)
 - Nilai/Target - Nama CloudFront host Anda
 - Jenis rekaman - CNAME
 - TTL - 3600 (atau apa pun yang sesuai untuk kasus penggunaan Anda)
- Untuk apex/root domain — Ini memerlukan konfigurasi DNS yang unik, karena catatan CNAME standar tidak dapat digunakan pada tingkat domain root atau apex. Karena sebagian besar penyedia DNS tidak mendukung catatan ALIAS, sebaiknya buat catatan ALIAS di Route 53. Contoh:
 - Nama — domain puncak Anda (seperti `example.com`)
 - Jenis rekaman - A
 - Alias - Ya
 - Target alias - Nama host Anda CloudFront
 - Kebijakan perutean — Sederhana (atau apa pun yang sesuai untuk kasus penggunaan Anda)
3. Verifikasi bahwa perubahan DNS telah disebarkan. (Ini biasanya terjadi ketika TTL kedaluwarsa. Terkadang bisa memakan waktu 24-48 jam.) Gunakan alat seperti `dig` atau `nslookup`.

```
dig www.example.com
# Should eventually return a CNAME pointing to your CloudFront hostname
```

3. Kembali ke CloudFront konsol dan pilih Kirim. Saat domain Anda aktif, CloudFront perbarui status domain untuk menunjukkan bahwa domain Anda siap melayani lalu lintas.

Untuk informasi selengkapnya, lihat dokumentasi untuk penyedia DNS Anda:

- [Cloudflare](#)
- [ClouDNS](#)
- [DNSimple](#)
- [Gandi.net](#)
- [GoDaddy](#)
- [Google Cloud DNS](#)
- [Namecheap](#)

Referensi pengaturan distribusi yang telah dikonfigurasi sebelumnya

Saat Anda membuat CloudFront distribusi, CloudFront secara otomatis mengonfigurasi sebagian besar setelan distribusi untuk Anda, berdasarkan jenis asal konten Anda. Secara opsional, Anda dapat memilih untuk mengedit pengaturan distribusi secara manual. Untuk informasi selengkapnya, lihat [Semua referensi pengaturan distribusi](#).

Bagian berikut menjelaskan pengaturan pra-konfigurasi default untuk distribusi, dan pengaturan yang dapat Anda sesuaikan.

Asal Amazon S3

Berikut ini adalah pengaturan asal yang CloudFront telah dikonfigurasi sebelumnya untuk asal Amazon S3 Anda dalam distribusi multi-penyewa.

Pengaturan asal (dikonfigurasi sebelumnya)

- Origin Access Control (hanya konsol) — CloudFront atur ini untuk Anda. CloudFront mencoba menambahkan kebijakan bucket S3 untuk distribusi standar dan untuk distribusi multi-penyewa tanpa parameter yang digunakan dalam domain asal.
- Tambahkan header kustom - None
- Aktifkan Origin Shield - Tidak
- Upaya koneksi - 3

Berikut ini adalah pengaturan cache yang CloudFront telah dikonfigurasi sebelumnya untuk asal Amazon S3 Anda dalam distribusi multi-penyewa.

Pengaturan cache (dikonfigurasi sebelumnya)

- Kompres objek secara otomatis - Ya
- Kebijakan protokol penampil - Alihkan ke HTTPS
- Metode HTTP yang diizinkan - GET, HEAD
- Batasi akses penampil - Tidak
- Kebijakan cache - CachingOptimized
- Kebijakan permintaan asal - Tidak ada

- Kebijakan header respon - Tidak ada
- Streaming Halus - Tidak
- Enkripsi tingkat lapangan - Tidak
- Aktifkan log akses real-time - Tidak
- Fungsi - Tidak

Berikut ini adalah pengaturan yang dapat Anda sesuaikan untuk asal Amazon S3 Anda dalam distribusi multi-penyewa.

Pengaturan yang dapat disesuaikan

- Akses S3 - CloudFront mengatur ini untuk Anda, berdasarkan pengaturan bucket S3 Anda:
 - Jika bucket Anda bersifat publik — Tidak diperlukan kebijakan Origin Access Control (OAC).
 - Jika bucket Anda bersifat pribadi — Anda dapat memilih atau membuat kebijakan OAC untuk digunakan.
- Aktifkan Origin Shield - Tidak
- Kompres objek secara otomatis - Ya
 - Jika Anda memilih Ya, maka kebijakan `CachingOptimized` caching digunakan.
 - Jika Anda memilih Tidak, maka kebijakan `CachingOptimizedForUncompressedObjects` caching digunakan.

Asal API Gateway

Berikut ini adalah pengaturan asal yang CloudFront telah dikonfigurasi sebelumnya untuk asal API Gateway Anda dalam distribusi multi-tenant.

Pengaturan asal (dikonfigurasi sebelumnya)

- Protokol — HTTPS saja
- Port HTTPS - 443
- Protokol SSL asal minimum - TLSv1.2
- Jalur asal - Tidak ada
- Origin Access Control (hanya konsol) — CloudFront mengatur ini untuk Anda
- Tambahkan header kustom - None

- Aktifkan Origin Shield - Tidak
- Upaya koneksi - 3
- Batas waktu respons - 30
- Keep-alive batas waktu — 5

Berikut ini adalah pengaturan cache yang CloudFront telah dikonfigurasi sebelumnya untuk asal API Gateway Anda dalam distribusi multi-tenant.

Pengaturan cache (dikonfigurasi sebelumnya)

- Kompres objek secara otomatis - Ya
- Kebijakan protokol penampil - Alihkan ke HTTPS
- Metode HTTP yang diizinkan - GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE
- Metode Cache HTTP - Tidak
- Izinkan permintaan gRPC berakhir HTTP/2 - Tidak
- Batasi akses penampil - Tidak
- Kebijakan cache - CachingDisabled (Nilai yang mungkin: UseOriginCacheControlHeaders, UseOriginCacheControlHeaders-QueryString)
- Kebijakan permintaan asal - AllViewerExceptHostHeader (Nilai yang mungkin: AllViewer, AllViewerandCloudFrontHeaders-2022-06)
- Kebijakan header respon - Tidak ada
- Streaming Halus - Tidak
- Enkripsi tingkat lapangan - Tidak
- Aktifkan log akses real-time - Tidak
- Fungsi - Tidak

Berikut ini adalah pengaturan yang dapat Anda sesuaikan untuk asal API Gateway Anda dalam distribusi multi-tenant.

Pengaturan yang dapat disesuaikan

- Aktifkan Origin Shield - (Default: Tidak)

- Kompres objek secara otomatis - (Default: Ya)

Asal kustom dan instans EC2

Berikut ini adalah pengaturan asal yang CloudFront telah dikonfigurasi sebelumnya untuk asal kustom Anda dalam distribusi multi-tenant.

Pengaturan asal (dikonfigurasi sebelumnya)

- Protokol - Penampil pertandingan
- Port HTTP - 80
- Port HTTPS - 443
- Protokol SSL asal minimum - TLSv1.2
- Jalur asal - Tidak ada
- Tambahkan header kustom - None
- Aktifkan Origin Shield - Tidak
- Upaya koneksi - 3
- Batas waktu respons - 30
- Keep-alive batas waktu — 5

Berikut ini adalah pengaturan cache yang CloudFront telah dikonfigurasi sebelumnya untuk custom origin dan instans EC2 Anda dalam distribusi multi-tenant.

Pengaturan cache (dikonfigurasi sebelumnya)

- Kompres objek secara otomatis - Ya
- Kebijakan protokol penampil - Alihkan ke HTTPS
- Metode HTTP yang diizinkan - GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE
- Metode Cache HTTP - Tidak
- Izinkan permintaan gRPC berakhir HTTP/2 - Tidak
- Batasi akses penampil - Tidak
- Kebijakan cache - UseOriginCacheControlHeaders (Nilai yang mungkin: UseOriginCacheControlHeaders-QueryStrings, CachingDisabled, CacheOptimized, CachingOptimizedForUncompressedObjects)

- Kebijakan permintaan asal - AllViewer (Nilai yang mungkin:AllViewerExceptHostHeader,AllViewerandCloudFrontHeaders-2022-06)
- Kebijakan header respon - Tidak ada
- Streaming Halus - Tidak
- Enkripsi tingkat lapangan - Tidak
- Aktifkan log akses real-time - Tidak
- Fungsi - Tidak

Berikut ini adalah pengaturan yang dapat Anda sesuaikan untuk asal kustom dan instans EC2 Anda dalam distribusi multi-tenant.

Pengaturan yang dapat disesuaikan

- Aktifkan Origin Shield - (Default: Tidak)
- Kompres objek secara otomatis - (Default: Ya)
- Caching - (Default:Cache by Default)
 - Jika Cache by Default dipilih, kebijakan UseOriginCacheControlHeaders cache digunakan.
 - Jika Do Not Cache by Default dipilih, kebijakan CachingDisabled cache digunakan.
- Sertakan string kueri dalam cache - (Default: Ya, jika Cache by Default sudah dipilih)
 - Jika Do Not Cache by Default sudah dipilih dan Anda kemudian memilih untuk menyertakan string kueri dalam cache, kebijakan UseOriginCacheControlHeaders-QueryStrings cache digunakan.

Asal Elastic Load Balancing

Berikut ini adalah pengaturan asal yang CloudFront telah dikonfigurasi sebelumnya untuk asal Elastic Load Balancing Anda dalam distribusi multi-tenant.

Pengaturan asal (dikonfigurasi sebelumnya)

- Protokol — HTTPS saja
- Port HTTPS - 443
- Protokol SSL asal minimum - TLSv1.2

- Jalur asal - Tidak ada
- Tambahkan header kustom - None
- Aktifkan Origin Shield - Tidak
- Upaya koneksi - 3
- Batas waktu respons - 30
- Keep-alive batas waktu — 5

Berikut ini adalah pengaturan cache yang CloudFront telah dikonfigurasi untuk asal Elastic Load Balancing Anda dalam distribusi multi-tenant.

Pengaturan cache (dikonfigurasi sebelumnya)

- Kompres objek secara otomatis - Ya
- Kebijakan protokol penampil - Alihkan ke HTTPS
- Metode HTTP yang diizinkan - GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE
- Metode Cache HTTP - Tidak
- Izinkan permintaan gRPC berakhir HTTP/2 - Tidak
- Batasi akses penampil - Tidak
- Caching - (Default: Cache by Default)
 - Jika Cache by Default dipilih, kebijakan UseOriginCacheControlHeaders cache digunakan.
 - Jika Do Not Cache by Default dipilih, kebijakan CachingDisabled cache digunakan.
- Sertakan string kueri dalam cache - (Default: Ya, jika Cache by Default sudah dipilih)
 - Jika Do Not Cache by Default sudah dipilih dan Anda kemudian memilih untuk menyertakan string kueri dalam cache, kebijakan UseOriginCacheControlHeaders-QueryStrings cache digunakan.
- Kebijakan permintaan asal - All Viewer (Nilai yang mungkin: AllViewerExceptHostHeader, AllViewerandCloudFrontHeaders-2022-06)
- Kebijakan header respon - Tidak ada
- Streaming Halus - Tidak
- Enkripsi tingkat lapangan - Tidak
- Aktifkan log akses real-time - Tidak

- Fungsi - Tidak

Berikut ini adalah pengaturan yang dapat Anda sesuaikan untuk asal Elastic Load Balancing Anda dalam distribusi multi-tenant.

Pengaturan yang dapat disesuaikan

- Aktifkan Origin Shield - (Default: Tidak)
- Kompres objek secara otomatis - (Default: Ya)
- Caching - (Default: Cache by Default)
 - Jika Cache by Default dipilih, kebijakan UseOriginCacheControlHeaders cache digunakan.
 - Jika Do Not Cache by Default dipilih, kebijakan CachingDisabled cache digunakan.
- Sertakan string kueri dalam cache - (Default: Ya, jika Cache by Default sudah dipilih)
 - Jika Do Not Cache by Default sudah dipilih dan Anda kemudian memilih untuk menyertakan string kueri dalam cache, kebijakan UseOriginCacheControlHeaders-QueryString cache digunakan.

MediaPackage asal v1

Berikut ini adalah pengaturan asal yang CloudFront telah dikonfigurasi sebelumnya untuk asal MediaPackage v1 Anda dalam distribusi multi-tenant.

Pengaturan asal (dikonfigurasi sebelumnya)

- Protokol — HTTPS saja
- Port HTTPS - 443
- Protokol SSL asal minimum - TLSv1.2
- Jalur asal — Anda memberikan ini dengan memasukkan MediaPackage URL Anda.
- Tambahkan header kustom - None
- Aktifkan Origin Shield - Tidak
- Upaya koneksi - 3
- Batas waktu respons - 30
- Keep-alive batas waktu — 5

Berikut ini adalah pengaturan cache yang CloudFront telah dikonfigurasi sebelumnya untuk asal MediaPackage v1 Anda dalam distribusi multi-tenant.

Pengaturan cache (dikonfigurasi sebelumnya)

- Kompres objek secara otomatis - Ya
- Kebijakan protokol penampil - Alihkan ke HTTPS
- Metode HTTP yang diizinkan - GET, HEAD
- Metode Cache HTTP - Tidak
- Izinkan permintaan gRPC berakhir HTTP/2 - Tidak
- Batasi akses penampil - Tidak
- Kebijakan cache - Elemental-MediaPackage
- Kebijakan permintaan asal - Tidak ada
- Kebijakan header respon - Tidak ada
- Streaming Halus - Tidak
- Enkripsi tingkat lapangan - Tidak
- Aktifkan log akses real-time - Tidak
- Fungsi - Tidak

MediaPackage v2 asal

Berikut ini adalah pengaturan asal yang CloudFront telah dikonfigurasi sebelumnya untuk asal MediaPackage v2 Anda dalam distribusi multi-tenant.

Pengaturan asal (dikonfigurasi sebelumnya)

- Origin Access Control — CloudFront mengatur ini untuk Anda dan menambahkan kebijakan
- Protokol — HTTPS saja
- Port HTTPS - 443
- Protokol SSL asal minimum - TLSv1.2
- Jalur asal - Tidak ada
- Tambahkan header kustom - None
- Aktifkan Origin Shield - Tidak

- Upaya koneksi - 3
- Batas waktu respons - 30
- Keep-alive batas waktu — 5

Berikut ini adalah pengaturan cache yang CloudFront telah dikonfigurasi sebelumnya untuk asal MediaPackage v2 Anda dalam distribusi multi-tenant.

Pengaturan cache (dikonfigurasi sebelumnya)

- Kompres objek secara otomatis - Ya
- Kebijakan protokol penampil - Alihkan ke HTTPS
- Metode HTTP yang diizinkan - GET, HEAD
- Metode Cache HTTP - Tidak
- Izinkan permintaan gRPC berakhir HTTP/2 - Tidak
- Batasi akses penampil - Tidak
- Kebijakan cache - Elemental-MediaPackage
- Kebijakan permintaan asal - Tidak ada
- Kebijakan header respon - Tidak ada
- Streaming Halus - Tidak
- Enkripsi tingkat lapangan - Tidak
- Aktifkan log akses real-time - Tidak
- Fungsi - Tidak

MediaTailor asal

Berikut ini adalah pengaturan asal yang CloudFront telah dikonfigurasi sebelumnya untuk MediaTailor asal Anda dalam distribusi multi-tenant.

Pengaturan asal (dikonfigurasi sebelumnya)

- Protokol — HTTPS saja
- Port HTTPS - 443
- Protokol SSL asal minimum - TLSv1.2
- Jalur asal — Anda memberikan ini dengan memasukkan MediaPackage URL Anda.

- Tambahkan header kustom - None
- Aktifkan Origin Shield - Tidak
- Upaya koneksi - 3
- Batas waktu respons - 30
- Keep-alive batas waktu — 5

Berikut ini adalah pengaturan cache yang CloudFront telah dikonfigurasi sebelumnya untuk MediaTailor asal Anda dalam distribusi multi-tenant.

Pengaturan cache (dikonfigurasi sebelumnya)

- Kompres objek secara otomatis - Ya
- Kebijakan protokol penampil - Alihkan ke HTTPS
- Metode HTTP yang diizinkan - GET, HEAD
- Metode Cache HTTP - Tidak
- Izinkan permintaan gRPC berakhir HTTP/2 - Tidak
- Batasi akses penampil - Tidak
- Kebijakan cache - Tidak ada
- Kebijakan permintaan asal - Elemental-MediaTailor-PersonalizedManifests
- Kebijakan header respon - Tidak ada
- Streaming Halus - Tidak
- Enkripsi tingkat lapangan - Tidak
- Aktifkan log akses real-time - Tidak
- Fungsi - Tidak

Semua referensi pengaturan distribusi

Anda dapat memilih untuk mengedit setelan CloudFront distribusi secara manual saat membuat atau memperbarui distribusi. Berikut ini adalah pengaturan yang dapat Anda edit.

Namun, CloudFront mengonfigurasi sebagian besar pengaturan distribusi untuk Anda, berdasarkan jenis asal konten Anda. Untuk informasi selengkapnya, lihat [Referensi pengaturan distribusi yang telah dikonfigurasi sebelumnya](#).

Untuk informasi lebih lanjut tentang membuat atau memperbarui distribusi dengan menggunakan CloudFront konsol, lihat [the section called “Buat distribusi”](#) atau [the section called “Perbarui distribusi”](#).

Topik

- [Pengaturan asal](#)
- [Pengaturan perilaku cache](#)
- [Pengaturan distribusi](#)
- [Halaman kesalahan kustom dan caching kesalahan](#)
- [Pembatasan geografis](#)

Pengaturan asal

Saat Anda menggunakan CloudFront konsol untuk membuat atau memperbarui distribusi, Anda memberikan informasi tentang satu atau beberapa lokasi, yang dikenal sebagai asal, tempat Anda menyimpan versi asli konten web Anda. CloudFront mendapatkan konten web Anda dari asal Anda dan menyajikannya kepada pemirsa melalui jaringan server edge di seluruh dunia.

Untuk jumlah maksimum asal saat ini yang dapat Anda buat untuk distribusi, atau untuk meminta kuota yang lebih tinggi, lihat [the section called “Kuota umum di distribusi”](#).

Jika ingin menghapus asal, Anda harus terlebih dahulu mengedit atau menghapus perilaku cache yang terkait dengan asal tersebut.

Important

Jika Anda menghapus asal, konfirmasi bahwa file yang sebelumnya dilayani oleh asal tersedia di tempat lain dan bahwa perilaku cache Anda sekarang mengirimkan permintaan untuk file tersebut ke sumber baru.

Saat Anda membuat atau memperbarui distribusi, Anda menentukan nilai berikut untuk setiap asal.

Topik

- [Domain asal](#)
- [Protokol \(hanya asal kustom\)](#)

- [Jalur asal](#)
- [Nama](#)
- [Akses asal \(hanya asal Amazon S3\)](#)
- [Tambahkan header kustom](#)
- [Aktifkan Origin Shield](#)
- [Upaya koneksi](#)
- [Batas waktu koneksi](#)
- [Batas waktu respons](#)
- [Batas waktu penyelesaian respons](#)
- [Keep-alive batas waktu \(khusus dan hanya asal VPC\)](#)
- [Tanggapan dan kuota batas waktu tetap hidup](#)

Domain asal

Domain asal adalah nama domain DNS dari sumber daya di mana CloudFront akan mendapatkan objek untuk asal Anda, seperti bucket Amazon S3 atau server HTTP. Contoh:

- Ember Amazon S – *amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com*

Note

Jika Anda baru saja membuat bucket S3, CloudFront distribusi mungkin mengembalikan HTTP 307 Temporary Redirect respons hingga 24 jam. Diperlukan waktu hingga 24 jam agar nama bucket S3 dapat didistribusikan ke semua Wilayah AWS. Ketika propagasi selesai, distribusi secara otomatis berhenti mengirimkan tanggapan pengalihan ini; Anda tidak perlu mengambil tindakan apa pun. Untuk informasi lebih lanjut, lihat [Mengapa saya mendapatkan tanggapan HTTP 307 Temporary Redirect dari Amazon S3?](#) dan [Pengalihan Permintaan Sementara](#).

- Ember Amazon S3 dikonfigurasi sebagai situs web – *amzn-s3-demo-bucket.s3-website.us-west-2.amazonaws.com*
- MediaStore kontainer — *examplemediastore.data.mediastore.us-west-1.amazonaws.com*
- MediaPackage titik akhir — *examplemediapackage.mediapackage.us-west-1.amazonaws.com*

- Amazon EC2 instance – *ec2-203-0-113-25.compute-1.amazonaws.com*
- Load balancer Elastic Load Balancing – *example-load-balancer-1234567890.us-west-2.elb.amazonaws.com*
- Server web Anda sendiri – *www.example.com*

Pilih nama domain di bidang domain Origin, atau ketik nama. Sumber daya dari Wilayah keikutsertaan harus dimasukkan secara manual. Nama domain tidak peka huruf besar/kecil. Domain asal Anda harus memiliki nama DNS yang dapat diselesaikan secara publik yang mengarahkan permintaan dari klien ke target melalui internet.

Jika Anda mengonfigurasi CloudFront untuk menyambung ke asal Anda melalui HTTPS, salah satu nama domain dalam sertifikat harus cocok dengan nama domain yang Anda tentukan untuk Nama Domain Asal. Jika tidak ada nama domain yang cocok, CloudFront mengembalikan kode status HTTP 502 (Bad Gateway) ke penampil. Untuk informasi selengkapnya, lihat [Nama domain dalam CloudFront distribusi dan sertifikat](#) dan [Kegagalan negosiasi SSL/TLS antara CloudFront dan server asal kustom](#).

 Note

Jika Anda menggunakan kebijakan permintaan asal yang meneruskan header host penampil ke asal, asal harus merespons dengan sertifikat yang cocok dengan header host penampil. Untuk informasi selengkapnya, lihat [Tambahkan header CloudFront permintaan](#).

Jika asal Anda adalah bucket Amazon S3, perhatikan hal berikut ini:

- Jika bucket dikonfigurasi sebagai situs web, masukkan titik akhir hosting situs web statis Amazon S3 untuk bucket Anda; jangan pilih nama bucket dari daftar di bidang domain Origin. Titik akhir hosting situs web statis muncul di konsol Amazon S3, di halaman Properti di bawah hosting situs web statis. Untuk informasi selengkapnya, lihat [the section called “Menggunakan bucket Amazon S3 yang dikonfigurasi sebagai titik akhir situs web”](#).
- Jika Anda mengonfigurasi Amazon S3 Transfer Acceleration untuk bucket, jangan tentukan titik akhir untuk **s3-accelerate** domain Origin.
- Jika Anda menggunakan bucket dari AWS akun lain dan jika bucket tidak dikonfigurasi sebagai situs web, masukkan nama, menggunakan format berikut:

bucket-name.s3.region.amazonaws.com

Jika bucket Anda berada di Wilayah AS, dan Anda ingin Amazon S3 merutekan permintaan ke fasilitas di Virginia utara, gunakan format berikut:

bucket-name.s3.us-east-1.amazonaws.com

- File harus dapat dibaca publik kecuali Anda mengamankan konten Anda di Amazon S3 dengan menggunakan CloudFront kontrol akses asal. Untuk informasi selengkapnya tentang kontrol akses, lihat [the section called “Batasi akses ke asal Amazon S3”](#).

Important

Jika asalnya adalah keranjang Amazon S3, nama keranjang harus sesuai dengan persyaratan penamaan DNS. Untuk informasi selengkapnya, buka [Pembatasan dan batasan Bucket](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Saat Anda mengubah nilai domain Origin untuk asal, CloudFront segera mulai mereplikasi perubahan ke lokasi CloudFront tepi. Sampai konfigurasi distribusi diperbarui di lokasi tepi tertentu, CloudFront terus meneruskan permintaan ke asal sebelumnya. Segera setelah konfigurasi distribusi diperbarui di lokasi tepi itu, CloudFront mulai meneruskan permintaan ke asal baru.

Mengubah asal tidak CloudFront perlu mengisi kembali cache tepi dengan objek dari asal baru. Selama permintaan penampil di aplikasi Anda belum berubah, CloudFront terus menyajikan objek yang sudah ada dalam edge cache hingga TTL di setiap objek kedaluwarsa atau hingga objek yang diminta jarang digali.

Protokol (hanya asal kustom)

Note

Ini hanya berlaku untuk asal kustom.

Kebijakan protokol yang CloudFront ingin Anda gunakan saat mengambil objek dari asal Anda.

Pilih salah satu nilai berikut:

- Hanya HTTP: hanya CloudFront menggunakan HTTP untuk mengakses asal.

 Important

Hanya HTTP adalah pengaturan default ketika asalnya adalah titik akhir hosting situs web statis Amazon S3, karena Amazon S3 tidak mendukung koneksi HTTPS untuk titik akhir hosting situs web statis. CloudFront Konsol tidak mendukung perubahan pengaturan ini untuk titik akhir hosting situs web statis Amazon S3.

- Hanya HTTPS: hanya CloudFront menggunakan HTTPS untuk mengakses asal.
- Penampil pencocokan: CloudFront berkomunikasi dengan asal Anda menggunakan HTTP atau HTTPS, tergantung pada protokol permintaan penampil. CloudFront cache objek hanya sekali meskipun pemirsa membuat permintaan menggunakan protokol HTTP dan HTTPS.

 Important

Untuk permintaan penampil HTTPS yang CloudFront diteruskan ke asal ini, salah satu nama domain dalam SSL/TLS sertifikat di server asal Anda harus cocok dengan nama domain yang Anda tentukan untuk domain Origin. Jika tidak, CloudFront menanggapi permintaan penampil dengan kode status HTTP 502 (Bad Gateway) alih-alih mengembalikan objek yang diminta. Untuk informasi selengkapnya, lihat [the section called “Persyaratan untuk menggunakan SSL/TLS sertifikat dengan CloudFront”](#).

Topik

- [Port HTTP](#)
- [Port HTTPS](#)
- [Protokol SSL asal minimum](#)

Port HTTP

 Note

Ini hanya berlaku untuk asal kustom.

(Opsional) Anda dapat menentukan port HTTP tempat asal kustom mendengarkan. Nilai yang valid termasuk port 80, 443, dan 1024 hingga 65535. Nilai default adalah port 80.

 Important

Port 80 adalah pengaturan default ketika asal-usulnya adalah titik akhir hosting situs web statis Amazon S3, karena Amazon S3 hanya mendukung port 80 untuk titik akhir hosting situs web statis. CloudFront Konsol tidak mendukung perubahan pengaturan ini untuk titik akhir hosting situs web statis Amazon S3.

Port HTTPS

 Note

Ini hanya berlaku untuk asal kustom.

(Opsional) Anda dapat menentukan port HTTPS tempat asal kustom mendengarkan. Nilai yang valid termasuk port 80, 443, dan 1024 hingga 65535. Nilai default adalah port 443. Ketika Protokol diatur ke HTTP saja, Anda tidak dapat menentukan nilai untuk port HTTPS.

Protokol SSL asal minimum

 Note

Ini hanya berlaku untuk asal kustom.

Pilih TLS/SSL protokol minimum yang CloudFront dapat digunakan saat membuat koneksi HTTPS ke asal Anda. Protokol TLS yang lebih rendah tidak terlalu aman, sehingga kami menyarankan agar Anda memilih protokol TLS terbaru yang didukung oleh produk asal Anda. Ketika Protokol diatur ke HTTP saja, Anda tidak dapat menentukan nilai untuk protokol SSL asal minimum.

Jika Anda menggunakan CloudFront API untuk mengatur TLS/SSL protokol CloudFront untuk digunakan, Anda tidak dapat menetapkan protokol minimum. Sebagai gantinya, Anda menentukan semua TLS/SSL protokol yang CloudFront dapat digunakan dengan asal Anda. Untuk informasi selengkapnya, lihat [OriginSslProtocols](#) di Referensi Amazon CloudFront API.

Jalur asal

Jika Anda CloudFront ingin meminta konten Anda dari direktori di asal Anda, masukkan jalur direktori, dimulai dengan garis miring (/). CloudFront menambahkan jalur direktori ke nilai domain Origin,

misalnya, **cf-origin.example.com/production/images**. Jangan menambahkan garis miring (/) di ujung jalur.

Misalnya, Anda telah menentukan nilai berikut untuk distribusi Anda:

- Domain asal - Bucket Amazon S3 bernama **amzn-s3-demo-bucket**
- Jalur asal — **/production**
- Nama domain alternatif (CNAME) - **example.com**

Saat pengguna `example.com/index.html` masuk ke browser, CloudFront kirimkan permintaan ke Amazon S3 untuk `amzn-s3-demo-bucket/production/index.html`

Saat pengguna `example.com/acme/index.html` masuk ke browser, CloudFront kirimkan permintaan ke Amazon S3 untuk `amzn-s3-demo-bucket/production/acme/index.html`

Nama

Nama adalah string yang secara unik mengidentifikasi asal ini dalam distribusi ini. Jika Anda membuat perilaku cache selain perilaku cache default, Anda menggunakan nama yang Anda tentukan di sini untuk mengidentifikasi asal yang CloudFront ingin Anda rutekan permintaan saat permintaan cocok dengan pola jalur untuk perilaku cache tersebut.

Akses asal (hanya asal Amazon S3)

Note

Ini hanya berlaku untuk asal bucket Amazon S3 (mereka yang tidak menggunakan titik akhir situs web statis S3)

Pilih pengaturan kontrol akses Origin (disarankan) jika Anda ingin membatasi akses ke asal bucket Amazon S3 hanya CloudFront untuk distribusi tertentu.

Pilih Publik jika asal bucket Amazon S3 dapat diakses publik.

Untuk informasi selengkapnya, lihat [the section called “Batasi akses ke asal Amazon S3”](#).

Untuk informasi tentang cara mengharuskan pengguna mengakses objek pada asal kustom dengan hanya menggunakan CloudFront URL, lihat [the section called “Batasi akses ke file pada asal kustom”](#).

Tambahkan header kustom

Jika Anda CloudFront ingin menambahkan header khusus setiap kali mengirim permintaan ke asal Anda, tentukan nama header dan nilainya. Untuk informasi selengkapnya, lihat [the section called “Tambahkan header khusus ke permintaan asal”](#).

Untuk jumlah maksimum header khusus saat ini yang dapat Anda tambahkan, panjang maksimum nama dan nilai header khusus, dan panjang total maksimum semua nama dan nilai header, lihat [Kuota](#).

Aktifkan Origin Shield

Pilih Ya untuk mengaktifkan CloudFront Origin Shield. Untuk informasi selengkapnya tentang Origin Shield, lihat [the section called “Gunakan Origin Shield”](#).

Upaya koneksi

Anda dapat mengatur berapa kali yang CloudFront mencoba terhubung ke asal. Anda dapat menentukan 1, 2, atau 3 sebagai jumlah percobaan. Angka default (jika tidak disebutkan sebaliknya) adalah 3.

Gunakan pengaturan ini bersama dengan batas waktu Koneksi untuk menentukan berapa lama CloudFront menunggu sebelum mencoba menyambung ke asal sekunder atau mengembalikan respons kesalahan ke penampil. Secara default, CloudFront tunggu selama 30 detik (3 upaya masing-masing 10 detik) sebelum mencoba terhubung ke asal sekunder atau mengembalikan respons kesalahan. Anda dapat mengurangi waktu ini dengan menentukan lebih sedikit percobaan, waktu habis koneksi yang lebih singkat, atau keduanya.

Jika jumlah upaya koneksi yang ditentukan gagal, CloudFront lakukan salah satu hal berikut:

- Jika asal adalah bagian dari kelompok asal, CloudFront cobalah untuk terhubung ke asal sekunder. Jika jumlah percobaan koneksi yang ditentukan ke asal sekunder gagal, maka CloudFront mengembalikan respons kesalahan ke penampil.
- Jika asal bukan bagian dari grup asal, CloudFront mengembalikan respons kesalahan ke penampil.

Untuk custom origin (termasuk bucket Amazon S3 yang dikonfigurasi dengan hosting situs web statis), pengaturan ini juga menentukan berapa kali CloudFront upaya untuk mendapatkan respons dari asal. Untuk informasi selengkapnya, lihat [the section called “Batas waktu respons”](#).

Batas waktu koneksi

Batas waktu koneksi adalah jumlah detik yang CloudFront menunggu ketika mencoba membuat koneksi ke asal. Anda dapat menentukan jumlah detik antara 1 dan 10 (inklusif). Waktu habis default (jika tidak ditentukan lain) adalah 10 detik.

Gunakan pengaturan ini bersama dengan upaya Sambungan untuk menentukan berapa lama CloudFront menunggu sebelum mencoba menyambung ke asal sekunder atau sebelum mengembalikan respons kesalahan ke penampil. Secara default, CloudFront tunggu selama 30 detik (3 upaya masing-masing 10 detik) sebelum mencoba terhubung ke asal sekunder atau mengembalikan respons kesalahan. Anda dapat mengurangi waktu ini dengan menentukan lebih sedikit percobaan, waktu habis koneksi yang lebih singkat, atau keduanya.

Jika CloudFront tidak membuat koneksi ke asal dalam jumlah detik yang ditentukan, CloudFront lakukan salah satu hal berikut:

- Jika jumlah upaya koneksi yang ditentukan lebih dari 1, CloudFront coba lagi untuk membuat koneksi. CloudFront mencoba hingga 3 kali, sebagaimana ditentukan oleh nilai upaya Koneksi.
- Jika semua upaya koneksi gagal dan asal-usulnya adalah bagian dari kelompok asal, CloudFront upaya untuk terhubung ke asal sekunder. Jika jumlah percobaan koneksi yang ditentukan ke asal sekunder gagal, maka CloudFront mengembalikan respons kesalahan ke penampil.
- Jika semua upaya koneksi gagal dan asal bukan bagian dari grup asal, CloudFront mengembalikan respons kesalahan ke penampil.

Batas waktu respons

Waktu habis respons asal, juga dikenal sebagai waktu habis baca asal atau waktu habis permintaan asal, berlaku untuk kedua nilai berikut:

- Berapa lama (dalam detik) CloudFront menunggu respons setelah meneruskan permintaan ke asal.
- Berapa lama (dalam detik) CloudFront menunggu setelah menerima paket respons dari asal dan sebelum menerima paket berikutnya.

 Tip

Jika Anda ingin meningkatkan nilai timeout karena penampil mengalami kesalahan kode status HTTP 504, pertimbangkan untuk mengeksplorasi cara lain untuk menghapus kesalahan tersebut sebelum mengubah nilai timeout. Lihat saran pemecahan masalah di [the section called “Kode status HTTP 504 \(Gateway Timeout\)”](#).

CloudFront perilaku tergantung pada metode HTTP dalam permintaan penampil:

- GET dan HEAD permintaan - Jika asal tidak merespons atau berhenti merespons dalam durasi waktu tunggu respons, hentikan CloudFront koneksi. CloudFront mencoba lagi untuk terhubung sesuai dengan nilai [the section called “Upaya koneksi”](#).
- DELETE, OPTIONS PATCH, PUT, dan POST permintaan — Jika asal tidak merespons selama durasi batas waktu baca, CloudFront hentikan koneksi dan tidak mencoba lagi untuk menghubungi asal. Klien dapat mengirim ulang permintaan bilamana perlu.

Batas waktu penyelesaian respons

 Note

Batas waktu penyelesaian respons tidak mendukung fitur [penerapan berkelanjutan](#).

Waktu (dalam detik) permintaan dari CloudFront ke asal dapat tetap terbuka dan menunggu tanggapan. Jika respons lengkap tidak diterima dari asal saat ini, CloudFront akhiri koneksi.

Tidak seperti batas waktu Respons, yang merupakan waktu tunggu untuk paket respons individu, batas waktu penyelesaian Respons adalah jumlah waktu maksimum yang diizinkan yang CloudFront menunggu respons selesai. Anda dapat menggunakan pengaturan ini untuk memastikan bahwa CloudFront tidak menunggu tanpa batas waktu untuk asal yang lambat atau tidak responsif, bahkan jika pengaturan batas waktu lainnya memungkinkan untuk menunggu lebih lama.

Batas waktu maksimum ini mencakup apa yang Anda tentukan untuk pengaturan batas waktu lainnya dan jumlah upaya Koneksi untuk setiap percobaan ulang. Anda dapat menggunakan pengaturan ini bersama-sama untuk menentukan berapa lama CloudFront menunggu permintaan penuh dan kapan harus mengakhiri permintaan, terlepas apakah itu lengkap atau tidak.

Misalnya, jika Anda mengatur pengaturan berikut:

- Upaya koneksi adalah 3
- Batas waktu koneksi adalah 10 detik
- Batas waktu respons adalah 30 detik
- Batas waktu penyelesaian respons adalah 60 detik

Ini berarti CloudFront akan mencoba untuk terhubung ke asal (hingga 3 upaya total), dengan setiap upaya koneksi waktu habis pada 10 detik. Setelah terhubung, CloudFront akan menunggu hingga 30 detik untuk asal untuk menanggapi permintaan sampai menerima paket terakhir dari respons.

Terlepas dari jumlah upaya koneksi atau batas waktu respons, CloudFront akan mengakhiri koneksi jika respons lengkap dari asal membutuhkan waktu lebih dari 60 detik untuk diselesaikan. CloudFront kemudian akan kembali ke penampil respons [the section called “Kode status HTTP 504 \(Gateway Timeout\)”](#) kesalahan atau respons kesalahan khusus jika Anda menentukannya.

Catatan

- Jika Anda menetapkan nilai untuk batas waktu penyelesaian respons, nilainya harus sama dengan atau lebih besar dari nilai [batas waktu respons \(batas waktu baca asal\)](#).
- Jika Anda tidak menetapkan nilai untuk batas waktu penyelesaian respons, CloudFront tidak akan menerapkan nilai maksimum.

Keep-alive batas waktu (khusus dan hanya asal VPC)

Batas waktu keep-alive adalah berapa lama (dalam hitungan detik) CloudFront mencoba mempertahankan koneksi ke custom origin Anda setelah mendapat paket respons terakhir.

Mempertahankan koneksi yang persisten menghemat waktu yang diperlukan untuk membangun kembali koneksi TCP dan melakukan handshake TLS lain untuk permintaan berikutnya.

Meningkatkan timeout yang terus-menerus membantu meningkatkan metrik permintaan-per-koneksi untuk distribusi.

 Note

Agar nilai Keep-alive batas waktu berpengaruh, asal Anda harus dikonfigurasi untuk memungkinkan koneksi persisten.

Tanggapan dan kuota batas waktu tetap hidup

- Untuk [batas waktu respons](#), defaultnya adalah 30 detik.
- Untuk [keep-alive timeout](#), defaultnya adalah 5 detik.

Jika Anda meminta peningkatan batas waktu untuk Akun AWS, perbarui asal distribusi Anda sehingga mereka memiliki batas waktu respons dan nilai batas waktu tetap hidup yang Anda inginkan. Peningkatan kuota untuk akun Anda tidak memperbarui asal Anda secara otomatis. Misalnya, jika Anda menggunakan fungsi Lambda @Edge untuk mengatur batas waktu keep-alive 90 detik, asal Anda harus sudah memiliki batas waktu keep-alive 90 detik atau lebih. Jika tidak, fungsi Lambda @Edge Anda mungkin gagal dijalankan.

Untuk informasi selengkapnya tentang kuota distribusi, termasuk cara meminta kenaikan, lihat [Kuota umum di distribusi](#).

Pengaturan perilaku cache

Dengan mengatur perilaku cache, Anda dapat mengonfigurasi berbagai CloudFront fungsi untuk pola jalur URL yang diberikan untuk file di situs web Anda. Misalnya, satu perilaku cache mungkin berlaku untuk semua .jpg file dalam images direktori di server web yang Anda gunakan sebagai server asal CloudFront. Fungsionalitas yang Anda dapat mengonfigurasi untuk setiap perilaku cache meliputi:

- Pola jalur
- Jika Anda telah mengonfigurasi beberapa asal untuk CloudFront distribusi Anda, asal yang CloudFront ingin Anda teruskan permintaan
- Apakah akan meneruskan string kueri ke asal Anda
- Apakah mengakses file yang ditentukan memerlukan URL yang ditandatangani
- Apakah mengharuskan pengguna menggunakan HTTPS untuk mengakses file-file tersebut
- Jumlah minimum waktu file-file tersebut tetap berada di CloudFront cache terlepas dari nilai Cache-Control header apa pun yang ditambahkan asal Anda ke file

Saat Anda membuat distribusi baru, Anda menetapkan pengaturan untuk perilaku cache default, yang secara otomatis meneruskan semua permintaan ke asal usul yang Anda tetapkan saat Anda membuat distribusi. Setelah membuat distribusi, Anda dapat membuat perilaku cache tambahan yang menentukan cara CloudFront merespons saat menerima permintaan untuk objek yang cocok dengan pola jalur, misalnya, *.jpg. Jika Anda membuat perilaku cache tambahan, perilaku cache default selalu yang terakhir untuk diproses. Perilaku cache lainnya diproses dalam urutan yang dicantumkan di CloudFront konsol atau, jika Anda menggunakan CloudFront API, urutan pencantuman mereka dalam DistributionConfig elemen untuk distribusi. Untuk informasi selengkapnya, lihat [Pola jalur](#).

Saat Anda membuat perilaku cache, Anda menentukan satu asal dari mana Anda CloudFront ingin mendapatkan objek. Akibatnya, jika Anda CloudFront ingin mendistribusikan objek dari semua asal Anda, Anda harus memiliki setidaknya sebanyak perilaku cache (termasuk perilaku cache default) seperti yang Anda miliki asal. Misalnya, jika Anda memiliki dua asal dan hanya perilaku cache default, perilaku cache default CloudFront menyebabkan objek dari salah satu asal, tetapi asal lainnya tidak pernah digunakan.

Untuk jumlah maksimum perilaku cache saat ini yang dapat Anda tambahkan ke distribusi, atau untuk meminta kuota yang lebih tinggi (sebelumnya dikenal sebagai batas), lihat [Kuota umum di distribusi](#).

Topik

- [Pola jalur](#)
- [Asal atau kelompok asal](#)
- [Kebijakan protokol penampil](#)
- [Metode HTTP yang Diizinkan](#)
- [Field-level konfigurasi enkripsi](#)
- [Metode HTTP cache](#)
- [Izinkan permintaan gRPC berakhir HTTP/2](#)
- [Cache berdasarkan header permintaan yang dipilih](#)
- [Header daftar yang diizinkan](#)
- [Caching objek](#)
- [TTL Minimum](#)
- [TTL Maksimum](#)
- [TTL bawaan](#)

- [Teruskan cookie](#)
- [Daftar cookie yang diizinkan](#)
- [Penerusan string kueri dan caching](#)
- [Daftar izin string kueri](#)
- [Streaming yang Lancar](#)
- [Batasi akses penampil \(gunakan URL yang ditandatangani atau cookie yang ditandatangani\)](#)
- [Penandatanganan tepercaya](#)
- [Akun AWS angka](#)
- [Kompresi objek secara otomatis](#)
- [CloudFront acara](#)
- [Fungsi Lambda ARN](#)
- [Sertakan isi](#)

Pola jalur

Pola jalur (misalnya, `images/* .jpg`) menentukan permintaan mana yang Anda inginkan perilaku cache ini diterapkan. Saat CloudFront menerima permintaan pengguna akhir, jalur yang diminta dibandingkan dengan pola jalur dalam urutan perilaku cache terdaftar dalam distribusi. Kecocokan pertama menentukan perilaku cache mana yang diterapkan pada permintaan tersebut. Misalnya, bayangkan Anda memiliki tiga perilaku cache dengan tiga pola jalur berikut, sesuai urutan ini:

- `images/* .jpg`
- `images/*`
- `*.gif`

Note

Anda dapat secara opsional menyertakan garis miring (/) di awal pola jalur, misalnya, `/images/* .jpg`. CloudFront perilakunya sama dengan atau tanpa pemimpin /. Jika Anda tidak menentukan/di awal jalur, karakter ini secara otomatis tersirat; CloudFront memperlakukan jalur yang sama dengan atau tanpa petunjuk /. Misalnya, CloudFront memperlakukan `/*product .jpg` sama seperti `*product .jpg`.

Permintaan untuk file `images/sample.gif` tidak memenuhi pola jalur pertama, sehingga perilaku cache terkait tidak diterapkan pada permintaan. File memenuhi pola jalur kedua, sehingga perilaku cache yang terkait dengan pola jalur kedua diterapkan meskipun permintaan juga sesuai dengan pola jalur ketiga.

Note

Saat Anda membuat distribusi baru, nilai dari Pola Jalan untuk perilaku cache default diatur menjadi `*` (semua file) dan tidak dapat diubah. Nilai ini menyebabkan CloudFront untuk meneruskan semua permintaan untuk objek Anda ke asal yang Anda tentukan di [Domain asal](#) bidang. Jika permintaan objek tidak cocok dengan pola jalur untuk perilaku cache lainnya, CloudFront terapkan perilaku yang Anda tentukan dalam perilaku cache default.

Important

Tentukan pola jalur dan urutannya dengan cermat atau Anda dapat memberi pengguna akses yang tidak diinginkan ke konten Anda. Misalnya, anggaplah permintaan tersebut sesuai dengan pola jalur untuk perilaku cache. Perilaku singgahan pertama tidak memerlukan URL yang ditandatangani dan perilaku cache kedua benar-benar memerlukan URL yang ditandatangani. Pengguna dapat mengakses objek tanpa menggunakan URL yang ditandatangani karena CloudFront memproses perilaku cache yang terkait dengan kecocokan pertama.

Jika Anda bekerja dengan MediaPackage channel, Anda harus menyertakan pola jalur tertentu untuk perilaku cache yang Anda tentukan untuk tipe titik akhir untuk asal Anda. Misalnya, untuk titik akhir DASH, Anda mengetik `*.mpd` untuk Pola Jalan. Untuk informasi lebih lanjut dan petunjuk spesifik, lihat [Sajikan video langsung yang diformat dengan AWS Elemental MediaPackage](#).

Jalur yang Anda tentukan berlaku untuk permintaan semua file di direktori yang ditentukan dan di subdirektori di bawah direktori yang ditentukan. CloudFront tidak mempertimbangkan string kueri atau cookie saat mengevaluasi pola jalur. Misalnya, jika `images` direktori berisi `product1` dan `product2` subdirektori, pola jalur `images/*.jpg` berlaku bagi permintaan file `.jpg` di `images`, `images/product1`, dan `images/product2` yang berbeda. Jika Anda ingin menerapkan perilaku cache yang berbeda pada file di `images/product1` yang lebih besar dari file dalam `images` dan `images/product2` direktori, membuat perilaku cache terpisah untuk `images/product1` dan memindahkan perilaku cache tersebut ke posisi di atas (sebelum) perilaku cache untuk `images` direktori.

Anda dapat menggunakan karakter wildcard berikut dalam pola jalur Anda:

- * sesuai dengan 0 karakter atau lebih.
- ? persis cocok dengan 1 karakter.

Contoh berikut menunjukkan cara kerja karakter wildcard:

Pola jalur	File yang cocok dengan pola jalur
*.jpg	Semua file.jpg.
images/*.jpg	Semua file.jpg di <code>images</code> direktori dan di subdirektori di bawah direktori <code>images</code>
a*.jpg	• Semua file.jpg yang nama filenya dimulai dengana, misalnya, <code>apple.jpg</code> dan <code>appalachian_trail_2012_05_21.jpg</code> • Semua file .jpg yang berawalan jalur file a, misalnya, <code>abra/cadabra/magic.jpg</code> .
a??.jpg	Semua file.jpg yang nama filenya dimulai a dan diikuti oleh tepat dua karakter lainnya, misalnya, <code>ant.jpg</code> dan <code>abe.jpg</code>
.doc	Semua file dengan ekstensi nama file dimulai <code>.doc</code> , misalnya, <code>.doc</code> , <code>.docx</code> , dan <code>.docm</code> berkas. Anda tidak dapat menggunakan pola jalur <code>*.doc?</code> dalam kasus ini, karena pola jalan tersebut tidak akan berlaku pada permintaan untuk <code>.doc</code> berkas; <code>?</code> karakter wildcard menggantikan persis satu karakter.

Panjang maksimal pola jalur adalah 255 karakter. Nilai dapat berisi salah satu karakter berikut:

- A-Z, a-z

Pola jalur peka huruf besar/kecil, sehingga pola jalur *.jpg tidak berlaku untuk file LOGO.JPG

- 0-9
- _ - . * \$ / ~ " ' @ : +
- &, lulus dan kembali saat &

Normalisasi jalur

CloudFront menormalkan jalur URI yang konsisten dengan [RFC 3986](#) dan kemudian mencocokkan jalur dengan perilaku cache yang benar. Setelah perilaku cache dicocokkan, CloudFront kirimkan jalur URI mentah ke asal. Jika tidak cocok, permintaan akan dicocokkan dengan perilaku cache default Anda.

Beberapa karakter dinormalisasi dan dihapus dari jalur, seperti beberapa garis miring (/) atau periode (.). Ini dapat mengubah URL yang CloudFront digunakan untuk mencocokkan perilaku cache yang dimaksud.

Example Contoh

Anda menentukan /a/b* dan /a* jalur untuk perilaku cache Anda.

- Penampil yang mengirim /a/b?c=1 jalur akan cocok dengan perilaku /a/b* cache.
- Penampil yang mengirim /a/b/.?c=1 jalur akan cocok dengan perilaku /a* cache.

Untuk mengatasi jalur yang dinormalisasi, Anda dapat memperbarui jalur permintaan atau pola jalur untuk perilaku cache.

Asal atau kelompok asal

Pengaturan ini hanya berlaku ketika Anda membuat atau memperbarui perilaku cache untuk distribusi yang ada.

Masukkan nilai asal atau kelompok asal yang ada. Ini mengidentifikasi grup asal atau asal tempat Anda CloudFront ingin merutekan permintaan ketika permintaan (seperti <https://example.com/logo.jpg>) cocok dengan pola jalur untuk perilaku cache (seperti *.jpg) atau untuk perilaku cache default (*).

Kebijakan protokol penampil

Pilih kebijakan protokol yang ingin digunakan pemirsa untuk mengakses konten Anda di lokasi CloudFront tepi:

- HTTP dan HTTPS: Penampil dapat menggunakan kedua protokol.
- Mengalihkan HTTP ke HTTPS: Penampil dapat menggunakan kedua protokol, tetapi permintaan HTTP secara otomatis dialihkan ke permintaan HTTPS.
- Hanya HTTPS: Penampil hanya dapat mengakses konten Anda jika mereka menggunakan HTTPS.

Untuk informasi selengkapnya, lihat [Memerlukan HTTPS untuk komunikasi antara pemirsa dan CloudFront](#).

Metode HTTP yang Diizinkan

Tentukan metode HTTP yang CloudFront ingin Anda proses dan teruskan ke asal Anda:

- GET, HEAD: Anda CloudFront hanya dapat menggunakan untuk mendapatkan objek dari asal Anda atau untuk mendapatkan header objek.
- GET, HEAD, OPTIONS: Anda CloudFront hanya dapat menggunakan untuk mendapatkan objek dari asal Anda, mendapatkan header objek, atau mengambil daftar opsi yang didukung server asal Anda.
- GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE: Anda dapat menggunakan CloudFront untuk mendapatkan, menambah, memperbarui, dan menghapus objek, dan untuk mendapatkan header objek. Selain itu, Anda dapat melakukan operasi POST lainnya seperti mengirimkan data dari formulir web.

Note

Jika Anda menggunakan gRPC dalam beban kerja Anda, Anda harus memilih GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE. Beban kerja gRPC memerlukan metode ini. POST Untuk informasi selengkapnya, lihat [Menggunakan gRPC dengan distribusi CloudFront](#).

CloudFront cache tanggapan GET dan HEAD permintaan dan, secara opsional, OPTIONS permintaan. Respons terhadap OPTIONS permintaan di-cache secara terpisah dari respons GET dan HEAD permintaan (OPTIONSmetode ini disertakan dalam [kunci cache](#) untuk

OPTIONS permintaan). CloudFront tidak menyimpan respons terhadap permintaan yang menggunakan metode lain.

Important

Jika Anda memilih DAPATKAN, KEPALA, OPSI atau DAPATKAN, KEPALA, OPSI, PUT, POST, PATCH, DELETE, Anda mungkin perlu membatasi akses ke bucket Amazon S3 Anda atau ke tempat yang dibuat khusus untuk mencegah pengguna melakukan operasi yang Anda tidak ingin mereka lakukan. Contoh berikut menjelaskan cara membatasi akses:

- Jika Anda menggunakan Amazon S3 sebagai asal untuk distribusi Anda: Buat kontrol akses CloudFront asal untuk membatasi akses ke konten Amazon S3 Anda, dan berikan izin ke kontrol akses asal. Misalnya, jika Anda mengonfigurasi CloudFront untuk menerima dan meneruskan metode ini hanya karena ingin digunakan PUT, Anda tetap harus mengonfigurasi kebijakan bucket Amazon S3 untuk menangani DELETE permintaan dengan tepat. Untuk informasi selengkapnya, lihat [Batasi akses ke asal Amazon S3](#).
- Jika Anda menggunakan asal kustom: Konfigurasi server asal Anda untuk menangani semua metode. Misalnya, jika Anda mengonfigurasi CloudFront untuk menerima dan meneruskan metode ini hanya karena Anda ingin menggunakan POST, Anda masih harus mengonfigurasi server asal Anda untuk menangani DELETE permintaan dengan tepat.

Field-level konfigurasi enkripsi

Jika Anda ingin menerapkan enkripsi tingkat bidang pada bidang data tertentu, dalam daftar tarik-turun, pilih konfigurasi enkripsi tingkat bidang.

Untuk informasi selengkapnya, lihat [Gunakan enkripsi tingkat lapangan untuk membantu melindungi data sensitif](#).

Metode HTTP cache

Tentukan apakah Anda CloudFront ingin menyimpan respons dari asal Anda saat penampil mengirimkan OPTIONS permintaan. CloudFront selalu menyimpan respons GET dan HEAD permintaan.

Izinkan permintaan gRPC berakhir HTTP/2

Tentukan apakah Anda ingin distribusi Anda mengizinkan permintaan gRPC. Untuk mengaktifkan gRPC, pilih pengaturan berikut:

- Untuk [metode HTTP yang Diizinkan](#), pilih metode GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE. gRPC memerlukan metode ini POST.
- Pilih kotak centang gRPC yang muncul setelah Anda memilih metode. POST
- Untuk [Versi HTTP yang didukung](#), pilih HTTP/2.

Untuk informasi selengkapnya, lihat [Menggunakan gRPC dengan distribusi CloudFront](#).

Cache berdasarkan header permintaan yang dipilih

Tentukan apakah Anda CloudFront ingin menyimpan objek berdasarkan nilai header yang ditentukan:

- Tidak ada (meningkatkan caching) - CloudFront tidak men-cache objek Anda berdasarkan nilai header.
- Allowlist — CloudFront cache objek Anda hanya berdasarkan nilai header yang ditentukan. Gunakan Header Allowlist untuk memilih header yang ingin Anda gunakan sebagai dasar CloudFront caching.
- Semua - CloudFront tidak menyimpan cache objek yang terkait dengan perilaku cache ini. Sebagai gantinya, CloudFront kirim setiap permintaan ke asal. (Tidak disarankan untuk asal Amazon S3.)

Terlepas dari opsi yang Anda pilih, CloudFront teruskan header tertentu ke asal Anda dan lakukan tindakan spesifik berdasarkan header yang Anda teruskan. Untuk informasi selengkapnya tentang cara CloudFront menangani penerusan header, lihat. [Header dan CloudFront perilaku permintaan HTTP \(asal kustom dan Amazon S3\)](#)

Untuk informasi selengkapnya tentang cara mengonfigurasi caching CloudFront dengan menggunakan header permintaan, lihat. [Konten cache berdasarkan header permintaan](#)

Header daftar yang diizinkan

Pengaturan ini hanya berlaku ketika Anda memilih Allowlist for Cache Berdasarkan Header Permintaan yang Dipilih.

Tentukan header yang CloudFront ingin Anda pertimbangkan saat menyimpan objek Anda. Pilih header dari daftar header yang tersedia dan pilih Tambahkan. Untuk meneruskan header kustom, masukkan nama header di kolom, lalu pilih Tambahkan Kustom.

Untuk jumlah maksimum header saat ini yang dapat Anda izinkan untuk setiap perilaku cache, atau untuk meminta kuota yang lebih tinggi (sebelumnya dikenal sebagai batas), lihat [Kuota pada header](#)

Caching objek

Jika server asal Anda menambahkan Cache-Control header ke objek Anda untuk mengontrol berapa lama objek tetap berada di CloudFront cache dan jika Anda tidak ingin mengubah Cache-Control nilainya, pilih Gunakan Header Cache Asal.

Untuk menentukan waktu minimum dan maksimum objek Anda tetap berada di CloudFront cache terlepas dari Cache-Control header, dan waktu default objek Anda tetap berada di CloudFront cache saat Cache-Control header hilang dari objek, pilih Sesuaikan. Lalu tentukan nilai dalam TTL Minimum, TTL bawaan, dan TTL Maksimum bidang.

Untuk informasi selengkapnya, lihat [Mengelola berapa lama konten tetap dalam cache \(kedaluwarsa\)](#).

TTL Minimum

Tentukan jumlah waktu minimum, dalam detik, yang Anda inginkan objek tetap berada di CloudFront cache sebelum CloudFront mengirim permintaan lain ke asal untuk menentukan apakah objek telah diperbarui.

Warning

Jika TTL minimum Anda lebih besar dari 0, CloudFront akan menyimpan konten untuk setidaknya durasi yang ditentukan dalam TTL minimum kebijakan cache, bahkan jika Cache-Control: no-cache, no-store, atau private arahan ada di header asal.

Untuk informasi selengkapnya, lihat [Mengelola berapa lama konten tetap dalam cache \(kedaluwarsa\)](#).

TTL Maksimum

Tentukan jumlah waktu maksimum, dalam hitungan detik, agar objek tetap berada di CloudFront cache sebelum CloudFront menanyakan asal Anda untuk melihat apakah objek telah diperbarui.

Nilai yang Anda tentukan untuk TTL Maksimum hanya berlaku saat asal Anda menambahkan header HTTP seperti `Cache-Control max-age`, `Cache-Control s-maxage`, atau `Expires` objek. Untuk informasi selengkapnya, lihat [Mengelola berapa lama konten tetap dalam cache \(kedaluwarsa\)](#).

Untuk menentukan nilai untuk TTL Maksimum, Anda harus memilih Menyesuaikan untuk Mengatasi Objek pengaturan.

Nilai default untuk TTL Maksimum adalah 31536000 detik (satu tahun). Jika Anda mengubah nilai TTL Minimum atau TTL bawaan hingga lebih dari 31536000 detik, kemudian nilai default TTL Maksimum perubahan pada nilai TTL bawaan.

TTL bawaan

Tentukan jumlah waktu default, dalam detik, yang Anda inginkan objek tetap dalam CloudFront cache sebelum CloudFront meneruskan permintaan lain ke asal Anda untuk menentukan apakah objek telah diperbarui. Nilai yang Anda tentukan untuk TTL default hanya berlaku saat asal Anda tidak tambahkan header HTTP seperti `Cache-Control max-age`, `Cache-Control s-maxage`, atau `Expires` ke objek. Untuk informasi selengkapnya, lihat [Mengelola berapa lama konten tetap dalam cache \(kedaluwarsa\)](#).

Untuk menentukan nilai untuk TTL bawaan, Anda harus memilih Menyesuaikan untuk Mengatasi Objek pengaturan.

Nilai default untuk TTL bawaan adalah 86400 detik (satu hari). Jika Anda mengubah nilai TTL Minimum hingga lebih dari 86400 detik, kemudian nilai default TTL bawaan perubahan pada nilai TTL Minimum.

Teruskan cookie

Note

Untuk asal Amazon S3, opsi ini hanya berlaku untuk bucket yang dikonfigurasi sebagai titik akhir situs web.

Tentukan apakah Anda CloudFront ingin meneruskan cookie ke server asal Anda dan, jika demikian, yang mana. Jika Anda memilih untuk meneruskan hanya cookie yang dipilih (daftar cookie yang diizinkan), masukkan nama cookie di bidang Allowlist Cookies. Jika Anda memilih Semua, CloudFront meneruskan semua cookie terlepas dari berapa banyak penggunaan aplikasi Anda.

Amazon S3 tidak memproses cookie, dan meneruskan cookie ke asal mengurangi kemampuan cache. Untuk perilaku cache yang meneruskan permintaan ke asal Amazon S3, pilih Tidak ada untuk Teruskan Cookie.

Untuk informasi lebih lanjut tentang meneruskan cookie ke asal, kunjungi [Konten cache berdasarkan cookie](#).

Daftar cookie yang diizinkan

Note

Untuk asal Amazon S3, opsi ini hanya berlaku untuk bucket yang dikonfigurasi sebagai titik akhir situs web.

Jika Anda memilih Allowlist dalam daftar Forward Cookies, maka di bidang Allowlist Cookies, masukkan nama cookie yang ingin Anda teruskan CloudFront ke server asal Anda untuk perilaku cache ini. Masukkan setiap nama cookie pada baris baru.

Anda dapat menentukan wildcard berikut untuk menentukan nama cookie:

- * sesuai dengan 0 karakter atau lebih dalam nama cookie
- ? persis cocok dengan satu karakter dalam nama cookie

Misalnya, bayangkan permintaan penampil untuk sebuah objek menyertakan cookie bernama:

`userid_member-number`

Dimana setiap pengguna Anda memiliki nilai unik untuk *member-number*. Anda CloudFront ingin men-cache versi terpisah dari objek untuk setiap anggota. Anda dapat melakukannya dengan meneruskan semua cookie ke asal Anda, tetapi permintaan penampil menyertakan beberapa cookie yang tidak ingin CloudFront Anda cache. Atau, Anda dapat menentukan nilai berikut sebagai nama cookie, yang menyebabkan diteruskan CloudFront ke asal semua cookie yang dimulai dengan `userid_`:

`userid_*`

Untuk jumlah maksimum nama cookie saat ini yang dapat Anda daftar untuk setiap perilaku cache, atau untuk meminta kuota yang lebih tinggi (sebelumnya dikenal sebagai batas), lihat. [Kuota pada cookie \(pengaturan cache warisan\)](#)

Penerusan string kueri dan caching

CloudFront dapat menyimpan versi konten yang berbeda berdasarkan nilai parameter string kueri. Pilih salah satu opsi berikut:

Tidak ada (Meningkatkan Caching)

Pilih opsi ini jika asal Anda mengembalikan versi objek yang sama terlepas dari nilai parameter string kueri. Ini meningkatkan kemungkinan yang CloudFront dapat melayani permintaan dari cache, yang meningkatkan kinerja dan mengurangi beban pada asal Anda.

Teruskan semua, cache berdasarkan daftar yang diizinkan

Pilih opsi ini jika server asal Anda mengembalikan versi objek yang berbeda berdasarkan satu atau lebih parameter string kueri. Kemudian tentukan parameter yang CloudFront ingin Anda gunakan sebagai dasar untuk caching di [Daftar izin string kueri](#) lapangan.

Teruskan semua, cache berdasarkan semua

Pilih opsi ini jika server asal Anda mengembalikan versi objek yang berbeda untuk semua parameter string kueri.

Untuk informasi selengkapnya tentang caching berdasarkan parameter string pencarian, termasuk cara meningkatkan kinerja, lihat [Konten cache berdasarkan parameter string kueri](#).

Daftar izin string kueri

Pengaturan ini hanya berlaku ketika Anda memilih Teruskan semua, cache berdasarkan daftar yang diizinkan untuk [Penerusan string kueri dan caching](#). Anda dapat menentukan parameter string kueri yang CloudFront ingin Anda gunakan sebagai dasar untuk caching.

Streaming yang Lancar

Pilih Ya jika Anda ingin mendistribusikan file media dalam format Microsoft Smooth Streaming dan Anda tidak memiliki server IIS.

Pilih Tidak jika Anda memiliki server Microsoft IIS yang ingin Anda gunakan sebagai sumber untuk mendistribusikan file media dalam format Microsoft Smooth Streaming, atau jika Anda tidak mendistribusikan file media Streaming Mulus.

 Note

Jika Anda menentukan Ya, Anda masih dapat mendistribusikan konten lain menggunakan perilaku cache ini jika konten tersebut sesuai dengan nilai Pola Jalan.

Untuk informasi selengkapnya, lihat [Konfigurasi video sesuai permintaan untuk Microsoft Smooth Streaming](#).

Batasi akses penampil (gunakan URL yang ditandatangani atau cookie yang ditandatangani)

Jika Anda ingin permintaan objek yang sesuai dengan PathPattern untuk perilaku cache ini menggunakan URL publik, pilih Tidak.

Jika Anda ingin permintaan objek yang sesuai dengan PathPattern untuk menggunakan URL yang ditandatangani, pilih Ya. Kemudian tentukan AWS akun yang ingin Anda gunakan untuk membuat URL yang ditandatangani; akun ini dikenal sebagai penandatanganan tepercaya.

Untuk informasi lebih lanjut tentang penanda tangan tepercaya, lihat [Tentukan penandatanganan yang dapat membuat URL yang ditandatangani dan cookie yang ditandatangani](#).

Penandatanganan tepercaya

Pengaturan ini hanya berlaku jika Anda memilih Ya untuk Batasi Akses Penampil (Gunakan URL yang Ditandatangani atau Cookie yang Ditandatangani).

Pilih AWS akun mana yang ingin Anda gunakan sebagai penandatanganan tepercaya untuk perilaku cache ini:

- **Mandiri:** Gunakan akun yang saat ini Anda masuki Konsol Manajemen AWS sebagai penandatanganan tepercaya. Jika saat ini Anda masuk sebagai pengguna IAM, AWS akun terkait akan ditambahkan sebagai penandatanganan tepercaya.
- **Menentukan Akun:** Masukkan nomor akun untuk penanda tangan tepercaya di bidang Nomor Akun AWS .

Untuk membuat URL yang ditandatangani, AWS akun harus memiliki setidaknya satu CloudFront key pair aktif.

Important

Jika Anda memperbarui distribusi yang sudah Anda gunakan untuk mendistribusikan konten, tambahkan penanda tangan tepercaya hanya ketika Anda siap untuk mulai membuat URL yang ditandatangani untuk objek Anda. Setelah Anda menambahkan penanda tangan tepercaya ke distribusi, pengguna harus menggunakan URL yang ditandatangani untuk mengakses objek yang sesuai dengan PathPattern untuk perilaku singgahan ini.

Akun AWS angka

Pengaturan ini hanya berlaku jika Anda memilih Tentukan Akun untuk Penandatanganan Tepercaya.

Jika Anda ingin membuat URL yang ditandatangani menggunakan Akun AWS selain atau bukan akun saat ini, masukkan satu Akun AWS nomor per baris di bidang ini. Perhatikan hal-hal berikut:

- Akun yang Anda tentukan harus memiliki setidaknya satu akun aktif CloudFront yang berbeda. Untuk informasi selengkapnya, lihat [Buat pasangan kunci untuk penandatanganan Anda](#).
- Anda tidak dapat membuat pasangan CloudFront kunci untuk pengguna IAM, sehingga Anda tidak dapat menggunakan pengguna IAM sebagai penandatanganan tepercaya.
- Untuk informasi tentang cara mendapatkan Akun AWS nomor akun, lihat [Melihat Akun AWS pengenalan](#) di Panduan Referensi Akun AWS Manajemen.
- Jika Anda memasukkan nomor akun untuk akun saat ini, CloudFront secara otomatis mencentang kotak centang Mandiri dan menghapus nomor akun dari daftar Nomor AWS Akun.

Kompresi objek secara otomatis

Jika Anda ingin CloudFront mengompres file jenis tertentu secara otomatis saat penonton mendukung konten terkompresi, pilih Ya. Saat CloudFront memampatkan konten Anda, unduhan lebih cepat karena file lebih kecil, dan halaman web Anda dirender lebih cepat untuk pengguna Anda. Untuk informasi selengkapnya, lihat [Sajikan file terkompresi](#).

CloudFront acara

Pengaturan ini berlaku untuk Asosiasi Fungsi Lambda.

Anda dapat memilih untuk menjalankan fungsi Lambda saat satu atau beberapa hal berikut CloudFront peristiwa yang terjadi:

- Saat CloudFront menerima permintaan dari penampil (permintaan penampil)
- Sebelum CloudFront meneruskan permintaan ke asal (permintaan asal)
- Ketika CloudFront menerima respons dari asal (respons asal)
- Sebelum CloudFront mengembalikan respons ke penampil (respons penampil)

Untuk informasi selengkapnya, lihat [Pilih acara untuk memicu fungsi](#).

Fungsi Lambda ARN

Pengaturan ini berlaku untuk Asosiasi Fungsi Lambda.

Tentukan Nama Sumber Daya Amazon (ARN) dari fungsi Lambda yang ingin Anda tambahkan pemicunya. Untuk mempelajari cara mendapatkan ARN untuk suatu fungsi, lihat langkah 1 dari prosedur [Menambahkan Pemicu dengan Menggunakan](#) Konsol. CloudFront

Sertakan isi

Pengaturan ini berlaku untuk Asosiasi Fungsi Lambda.

Untuk informasi selengkapnya, lihat [Sertakan isi](#).

Pengaturan distribusi

Nilai berikut berlaku untuk seluruh distribusi.

Topik

- [Kelas harga](#)
- [AWS WAF web ACL](#)
- [Nama domain alternatif \(CNames\)](#)
- [Sertifikat SSL](#)
- [Dukungan klien SSL khusus](#)
- [Kebijakan keamanan \(SSL/TLS versi minimum\)](#)
- [Versi HTTP yang didukung](#)
- [Objek akar default](#)
- [Pencatatan standar](#)
- [Log koneksi](#)
- [Awalan log](#)

- [Pencatatan cookie](#)
- [Aktifkan IPv6 \(permintaan penampil\)](#)
- [Autentikasi bersama](#)
- [Aktifkan IPv6 untuk asal kustom \(permintaan asal\)](#)
- [Komentar](#)
- [Status distribusi](#)

Kelas harga

Pilih kelas harga yang sesuai dengan harga maksimum yang ingin Anda bayar untuk CloudFront layanan. Secara default, CloudFront melayani objek Anda dari lokasi tepi di semua CloudFront Wilayah.

Untuk informasi selengkapnya tentang kelas harga dan tentang bagaimana pilihan kelas harga memengaruhi CloudFront kinerja distribusi Anda, lihat [CloudFront harga](#).

AWS WAF web ACL

Anda dapat melindungi CloudFront distribusi Anda dengan [AWS WAF](#) firewall aplikasi web yang memungkinkan Anda mengamankan aplikasi web dan API Anda untuk memblokir permintaan sebelum mencapai server Anda. Anda bisa [Aktifkan AWS WAF untuk distribusi](#) saat membuat atau mengedit CloudFront distribusi.

Secara opsional, Anda nantinya dapat mengonfigurasi perlindungan keamanan tambahan untuk ancaman lain yang spesifik untuk aplikasi Anda di AWS WAF konsol di <https://console.aws.amazon.com/wafv2/>

Untuk informasi selengkapnya AWS WAF, lihat [Panduan AWS WAF Pengembang](#).

Nama domain alternatif (CNames)

Tidak wajib. Tentukan satu atau beberapa nama domain yang ingin Anda gunakan untuk URL objek Anda, bukan nama domain yang ditetapkan CloudFront saat Anda membuat distribusi. Anda harus memiliki nama domain, atau memiliki otorisasi untuk menggunakannya, yang Anda verifikasi dengan menambahkan SSL/TLS sertifikat.

Misalnya, jika Anda menginginkan URL untuk objek tersebut:

`/images/image.jpg`

Seperti ini:

```
https://www.example.com/images/image.jpg
```

Alih-alih seperti ini:

```
https://d111111abcdef8.cloudfront.net/images/image.jpg
```

Tambahkan CNAME untuk `www.example.com`.

Important

Jika Anda menambahkan CNAME untuk `www.example.com` ke distribusi, Anda juga harus melakukan hal berikut:

- Buat (atau perbarui) catatan CNAME dengan layanan DNS Anda untuk menjawab pertanyaan `www.example.com` untuk `d111111abcdef8.cloudfront.net`.
- Tambahkan sertifikat CloudFront dari otoritas sertifikat tepercaya (CA) yang mencakup nama domain (CNAME) yang Anda tambahkan ke distribusi Anda, untuk memvalidasi otorisasi Anda untuk menggunakan nama domain.

Anda harus memiliki izin untuk membuat catatan CNAME dengan penyedia layanan DNS untuk domain tersebut. Biasanya, ini berarti bahwa Anda memiliki domain, atau bahwa Anda mengembangkan aplikasi untuk pemilik domain.

Untuk jumlah maksimum nama domain alternatif saat ini yang dapat Anda tambahkan ke distribusi, atau untuk meminta kuota yang lebih tinggi (sebelumnya dikenal sebagai batas), lihat [Kuota umum di distribusi](#).

Untuk informasi selengkapnya tentang nama domain alternatif, lihat [Gunakan URL kustom dengan menambahkan nama domain alternatif \(CNames\)](#). Untuk informasi selengkapnya tentang CloudFront URL, lihat [Sesuaikan format URL untuk file di CloudFront](#).

Sertifikat SSL

Jika Anda menentukan nama domain alternatif untuk digunakan dengan distribusi Anda, pilih Sertifikat SSL Kustom, dan kemudian, untuk memvalidasi otorisasi Anda untuk menggunakan nama domain alternatif, pilih sertifikat yang mencakupnya. Jika Anda ingin penampil menggunakan HTTPS untuk mengakses objek Anda, pilih pengaturan yang mendukungnya.

- CloudFront Sertifikat Default (*.cloudfront.net) - Pilih opsi ini jika Anda ingin menggunakan nama CloudFront domain di URL untuk objek Anda, seperti. `https://d1111111abcdef8.cloudfront.net/image1.jpg`
- Sertifikat SSL Kustom – Pilih opsi ini jika Anda ingin menggunakan nama domain Anda sendiri di URL untuk objek Anda sebagai nama domain alternatif, seperti `https://example.com/image1.jpg`. Kemudian pilih sertifikat yang akan digunakan yang mencakup nama domain alternatif. Daftar sertifikat dapat mencakup salah satu dari berikut ini:
 - Sertifikat yang disediakan oleh AWS Certificate Manager
 - Sertifikat yang Anda beli dari otoritas sertifikat pihak ketiga dan diunggah ke ACM
 - Sertifikat yang Anda beli dari otoritas sertifikat pihak ketiga dan diunggah ke toko sertifikat IAM

Jika Anda memilih pengaturan ini, kami sarankan Anda hanya menggunakan nama domain alternatif di URL objek Anda (`https://example.com/logo.jpg`). Jika Anda menggunakan nama domain CloudFront distribusi (`https://d1111111abcdef8.cloudfront.net/logo.jpg`) dan klien menggunakan penampil lama yang tidak mendukung SNI, cara penampil merespons bergantung pada nilai yang Anda pilih untuk Klien yang Didukung:

- Semua Klien: Penampil menampilkan peringatan karena nama CloudFront domain tidak cocok dengan nama domain dalam SSL/TLS sertifikat Anda.
- Hanya Klien yang Mendukung Indikasi Nama Server (SNI): CloudFront menjatuhkan koneksi dengan penampil tanpa mengembalikan objek.

Dukungan klien SSL khusus

Berlaku hanya jika Anda memilih Custom SSL Certificate (example.com) untuk Sertifikat SSL. Jika Anda menentukan satu atau beberapa nama domain alternatif dan sertifikat SSL khusus untuk distribusi, pilih cara Anda CloudFront ingin menyajikan permintaan HTTPS:

- Klien yang Mendukung Indikasi Nama Server (SNI) - (Disarankan) – Dengan pengaturan ini, hampir semua browser web modern dan klien dapat terhubung ke distribusi, karena mereka mendukung SNI. Namun, beberapa pemirsa mungkin menggunakan browser web atau klien lama yang tidak mendukung SNI, yang berarti mereka tidak dapat terhubung ke distribusi.

Untuk menerapkan pengaturan ini menggunakan CloudFront API, tentukan `sni-only` di `SSLSupportMethod` bidang. Dalam CloudFormation, kolom diberi nama `SslSupportMethod` (perhatikan berbagai kapitalisasi).

- Dukungan Klien Warisan – Dengan pengaturan ini, browser web dan klien yang lebih lama yang tidak mendukung SNI dapat terhubung ke distribusi. Namun, pengaturan ini akan dikenakan biaya bulanan tambahan. Untuk harga yang tepat, buka halaman [CloudFront Harga Amazon](#), dan cari halaman untuk SSL khusus IP Khusus.

Untuk menerapkan pengaturan ini menggunakan CloudFront API, tentukan `vip` di `SSLSupportMethod` bidang. Di CloudFormation, bidang diberi nama `SslSupportMethod` (perhatikan kapitalisasi yang berbeda).

Untuk informasi selengkapnya, lihat [Pilih cara CloudFront melayani permintaan HTTPS](#).

Kebijakan keamanan (SSL/TLS versi minimum)

Tentukan kebijakan keamanan yang ingin Anda gunakan CloudFront untuk koneksi HTTPS dengan pemirsa (klien). Kebijakan keamanan menentukan dua pengaturan:

- SSL/TLS Protokol minimum yang CloudFront digunakan untuk berkomunikasi dengan pemirsa.
- Cipher yang CloudFront dapat digunakan untuk mengenkripsi konten yang dikembalikan ke pemirsa.

Untuk informasi lebih lanjut tentang kebijakan keamanan, termasuk protokol dan cipher yang disertakan oleh masing-masing kebijakan, lihat [Protokol dan sandi yang didukung antara pemirsa dan CloudFront](#).

Kebijakan keamanan yang tersedia bergantung pada nilai yang Anda tentukan untuk Sertifikat SSL dan Custom SSL Client Support (dikenal sebagai `CloudFrontDefaultCertificate` dan `SSLSupportMethod` di CloudFront API):

- Ketika Sertifikat SSL adalah Sertifikat Default CloudFront (*.cloudfront.net) (saat **CloudFrontDefaultCertificate** berada **true** di API), CloudFront secara otomatis menetapkan kebijakan keamanan ke TLSv1.
- Jika Sertifikat SSL adalah Sertifikat SSL Kustom (contoh.com) dan Dukungan Klien SSL Kustom adalah Klien yang Mendukung Indikasi Nama Server (SNI) - (Disarankan) (jika `CloudFrontDefaultCertificate` adalah **false** dan `SSLSupportMethod` adalah **sni-only** di API), Anda dapat memilih dari kebijakan keamanan berikut:
 - TLSv1.3_2025
 - TLSv1.2_2025

- TLSv1.2_2021
 - TLSv1.2_2019
 - TLSv1.2_2018
 - TLSv1.1_2016
 - TLSv1_2016
 - TLSV1
- Saat Sertifikat SSL adalah Sertifikat SSL Khusus (contoh.com) dan Dukungan Klien SSL Khusus adalah Dukungan Klien Warisan (saat CloudFrontDefaultCertificate adalah false dan SSLSupportMethod adalah vip di API), Anda dapat memilih dari kebijakan keamanan berikut:
- TLSV1
 - SSLv3

Dalam konfigurasi ini, kebijakan keamanan TLSv1.3_2025, TLSv1.2_2025, TLSv1.2_2021, TLSv1.2_2019, TLSv1.2_2018, TLSv1.1_2016, dan TLSv1_2016 tidak tersedia di konsol atau API. CloudFront Jika Anda ingin menggunakan salah satu kebijakan keamanan ini, Anda memiliki opsi berikut:

- Evaluasi apakah kebutuhan distribusi Anda Dukungan Klien Legacy dengan alamat IP khusus. Jika penampil Anda mendukung [indikasi nama server \(SNI\)](#), kami menyarankan agar Anda memperbarui pengaturan Dukungan Klien SSL Kustom distribusi untuk Klien yang Mendukung Indikasi Nama Server (SNI) (tetapkan SSLSupportMethod untuk sni-only di API). Hal ini memungkinkan Anda untuk menggunakan kebijakan keamanan TLS yang tersedia, dan juga dapat mengurangi CloudFront biaya.
- [Jika Anda harus menyimpan Legacy Clients Support dengan alamat IP khusus, Anda dapat meminta salah satu kebijakan keamanan TLS lainnya \(TLSv1.3_2025, _2025, TLSv1.2_2021, TLSv1.2 TLSv1.2_2019, TLSv1.2_2018, _2016, atau TLSv1.1 TLSv1_2016\) dengan membuat case di Support Center.AWS](#)

Note

Sebelum Anda menghubungi AWS Support untuk meminta perubahan ini, pertimbangkan hal berikut:

- Saat Anda menambahkan salah satu kebijakan keamanan ini (TLSv1.3_2025, _2025, TLSv1.2 TLSv1.2_2021, TLSv1.2_2019, TLSv1.2_2018, _2016, atau TLSv1.1 TLSV1_2016) ke distribusi Dukungan Klien Legacy, kebijakan keamanan diterapkan ke semua permintaan penampil non-SNI untuk semua distribusi Dukungan Klien

Legacy di akun Anda. AWS Namun, jika pemirsa mengirimkan permintaan SNI ke distribusi dengan Dukungan Klien Warisan, kebijakan keamanan distribusi tersebut akan berlaku. Untuk memastikan bahwa kebijakan keamanan yang Anda inginkan diterapkan pada semua permintaan penampil yang dikirim ke semua distribusi Dukungan Klien Legacy di AWS akun Anda, tambahkan kebijakan keamanan yang diinginkan ke setiap distribusi satu per satu.

- Menurut definisi, kebijakan keamanan baru tidak mendukung ciphers dan protokol yang sama dengan yang lama. Misalnya, jika Anda memilih untuk memutakhirkan kebijakan keamanan distribusi dari TLSv1 ke TLSv1.1 _2016, distribusi tersebut tidak akan lagi mendukung cipher. DES-CBC3-SHA Untuk informasi lebih lanjut tentang ciphers dan protokol yang didukung oleh setiap kebijakan keamanan, lihat [Protokol dan sandi yang didukung antara pemirsa dan CloudFront](#).

Versi HTTP yang didukung

Pilih versi HTTP yang ingin didukung distribusi saat pemirsa berkomunikasi CloudFront.

Untuk pemirsa dan CloudFront penggunaan HTTP/2, pemirsa harus mendukung TLSv1.2 atau lebih baru, dan Indikasi Nama Server (SNI).

Untuk pemirsa dan CloudFront penggunaan HTTP/3, pemirsa harus mendukung TLSv1.3 dan Indikasi Nama Server (SNI). CloudFront mendukung migrasi HTTP/3 koneksi untuk memungkinkan penampil beralih jaringan tanpa kehilangan koneksi. Untuk informasi selengkapnya tentang migrasi koneksi, lihat [Migrasi Sambungan](#) di RFC 9000.

Note

Untuk informasi selengkapnya tentang TLSv1.3 cipher yang didukung, lihat. [Protokol dan sandi yang didukung antara pemirsa dan CloudFront](#)

Note

Jika Anda menggunakan Amazon Route 53, Anda dapat menggunakan catatan HTTPS untuk mengizinkan negosiasi protokol sebagai bagian dari pencarian DNS jika klien mendukungnya. Untuk informasi selengkapnya, lihat [Create alias resource record set](#).

Objek akar default

Tidak wajib. Objek yang CloudFront ingin Anda minta dari asal Anda (misalnya, `index.html`) ketika penampil meminta URL root distribusi Anda (`https://www.example.com/`), bukan objek dalam distribusi Anda (`https://www.example.com/product-description.html`). Menentukan objek akar default menghindari pemaparan konten distribusi Anda.

Panjang maksimum nama adalah 255 karakter. Nama dapat berisi salah satu karakter berikut:

- A-Z, a-z
- 0-9
- `_ - . * $ / ~ " ' &`, lulus dan kembali saat `&`;

Saat Anda menetapkan objek akar default, masukkan nama objek saja, misalnya, `index.html`. Jangan menambahkan `/` sebelum nama objek.

Untuk informasi selengkapnya, lihat [Tentukan objek root default](#).

Pencatatan standar

Tentukan apakah Anda CloudFront ingin mencatat informasi tentang setiap permintaan untuk objek dan menyimpan file log. Anda dapat mengaktifkan atau menonaktifkan log kapan saja. Tidak ada biaya tambahan jika Anda mengaktifkan logging, tetapi Anda dapat dikenakan biaya untuk menyimpan dan mengakses file. Anda dapat menghapus log kapan saja.

CloudFront mendukung opsi logging standar berikut:

- [Pencatatan standar \(v2\)](#) - Anda dapat mengirim log ke tujuan pengiriman, termasuk Amazon CloudWatch Logs, Amazon Data Firehose, dan Amazon Simple Storage Service (Amazon S3).
- [Pencatatan standar \(lama\)](#) - Anda hanya dapat mengirim log ke bucket Amazon S3.

Log koneksi

Saat Anda mengaktifkan [otentikasi timbal balik](#) untuk distribusi Anda, CloudFront berikan log koneksi yang menangkap atribut tentang permintaan yang dikirim ke distribusi Anda. Log koneksi berisi informasi seperti alamat IP klien dan port, informasi sertifikat klien, hasil koneksi, dan cipher

TLS yang digunakan. Log koneksi ini kemudian dapat digunakan untuk meninjau pola permintaan dan tren lainnya.

Untuk mempelajari lebih lanjut tentang log koneksi, lihat [Observabilitas menggunakan log koneksi](#).

Awalan log

(Opsional) Jika Anda mengaktifkan logging standar (legacy), tentukan string, jika ada, yang ingin Anda awalan CloudFront ke nama file log akses untuk distribusi ini, misalnya, `exampleprefix/`. The trailing slash (/) adalah opsional tetapi dianjurkan untuk menyederhanakan browsing file log Anda. Untuk informasi selengkapnya, lihat [Mengonfigurasi pencatatan standar \(lama\)](#).

Pencatatan cookie

Jika Anda CloudFront ingin menyertakan cookie di log akses, pilih Aktif. Jika Anda memilih untuk memasukkan cookie dalam CloudFront log, mencatat semua cookie terlepas dari bagaimana Anda mengonfigurasi perilaku cache untuk distribusi ini: meneruskan semua cookie, tidak meneruskan cookie, atau meneruskan daftar cookie tertentu ke asal.

Amazon S3 tidak memproses cookie, jadi kecuali distribusi Anda juga menyertakan Amazon EC2 atau asal kustom lainnya, kami menyarankan agar Anda memilih Nonaktif untuk nilai Pencatatan Cookie.

Untuk informasi selengkapnya tentang cookie, lihat [Konten cache berdasarkan cookie](#).

Aktifkan IPv6 (permintaan penampil)

Jika Anda ingin CloudFront menanggapi permintaan penampil dari alamat IP IPv4 dan IPv6, pilih Aktifkan IPv6. Untuk informasi selengkapnya, lihat [Aktifkan IPv6 untuk distribusi CloudFront](#).

Autentikasi bersama

Tidak wajib. Anda dapat memilih untuk mengaktifkan otentikasi timbal balik untuk CloudFront distribusi Anda. Untuk informasi selengkapnya, lihat [Otentikasi TLS timbal balik dengan CloudFront \(Viewer mTLS\)](#).

Aktifkan IPv6 untuk asal kustom (permintaan asal)

Bila Anda menggunakan custom origin (tidak termasuk asal Amazon S3 dan VPC), Anda dapat menyesuaikan pengaturan asal untuk distribusi Anda untuk memilih CloudFront cara

menghubungkan ke asal Anda menggunakan alamat IPv4 atau IPv6. Untuk informasi selengkapnya, lihat [Aktifkan IPv6 untuk distribusi CloudFront](#).

Komentar

Tidak wajib. Saat membuat distribusi, Anda dapat menyertakan komentar berisi hingga 128 karakter. Anda dapat memperbarui komentar kapan saja.

Status distribusi

Menunjukkan apakah Anda ingin distribusi diaktifkan atau dinonaktifkan setelah diterapkan:

- Diaktifkan berarti bahwa segera setelah distribusi sepenuhnya diterapkan, Anda dapat menerapkan tautan yang menggunakan nama domain distribusi dan pengguna dapat mengambil konten. Setiap kali distribusi diaktifkan, CloudFront menerima dan menangani setiap permintaan pengguna akhir untuk konten yang menggunakan nama domain yang terkait dengan distribusi tersebut.

Saat Anda membuat, memodifikasi, atau menghapus CloudFront distribusi, perlu waktu untuk perubahan Anda menyebar ke CloudFront database. Permintaan segera untuk informasi tentang distribusi mungkin tidak menunjukkan perubahan tersebut. Propagasi biasanya selesai dalam beberapa menit, tetapi beban sistem tinggi atau partisi jaringan dapat meningkatkan waktu ini.

- Nonaktif berarti bahwa meskipun distribusi mungkin diterapkan dan siap digunakan, pengguna tidak dapat menggunakannya. Setiap kali distribusi dinonaktifkan, CloudFront tidak menerima permintaan pengguna akhir apa pun yang menggunakan nama domain yang terkait dengan distribusi tersebut. Sampai Anda mengalihkan distribusi dari dinonaktifkan ke diaktifkan (dengan memperbarui konfigurasi distribusi), tidak ada yang dapat menggunakannya.

Anda dapat mengubah distribusi antara dinonaktifkan dan diaktifkan sesering yang Anda inginkan. Ikuti proses untuk memperbarui konfigurasi distribusi. Untuk informasi selengkapnya, lihat [Perbarui distribusi](#).

Halaman kesalahan kustom dan caching kesalahan

Anda dapat CloudFront mengembalikan objek ke penampil (misalnya, file HTML) saat Amazon S3 atau custom origin mengembalikan kode status HTTP 4xx atau 5xx. CloudFront Anda juga dapat menentukan berapa lama respons kesalahan dari asal Anda atau halaman kesalahan kustom di-cache di cache CloudFront tepi. Untuk informasi selengkapnya, lihat [Buat halaman kesalahan khusus untuk kode status HTTP tertentu](#).

 Note

Nilai-nilai berikut tidak disertakan dalam wizard Create Distribution, sehingga Anda dapat mengonfigurasi halaman kesalahan khusus hanya ketika Anda memperbarui distribusi.

Topik

- [Kode kesalahan HTTP](#)
- [Jalur halaman respons](#)
- [Kode tanggapan HTTP](#)
- [Kesalahan caching minimum TTL \(detik\)](#)

Kode kesalahan HTTP

Kode status HTTP yang CloudFront ingin Anda kembalikan halaman kesalahan kustom. Anda dapat mengonfigurasi CloudFront untuk mengembalikan halaman kesalahan khusus untuk tidak ada, beberapa, atau semua kode status HTTP yang CloudFront di-cache.

Jalur halaman respons

Jalur ke halaman kesalahan kustom (misalnya, `/4xx-errors/403-forbidden.html`) yang Anda inginkan CloudFront untuk kembali ke penampil saat asal Anda mengembalikan kode status HTTP yang Anda tentukan untuk Kode Kesalahan (misalnya, 403). Jika Anda ingin menyimpan objek Anda dan halaman kesalahan kustom Anda di lokasi yang berbeda, distribusi Anda harus menyertakan perilaku cache yang menyatakan hal berikut benar:

- Nilai dari Pola Jalan sesuai dengan jalur ke pesan kesalahan khusus Anda. Misalnya, Anda menyimpan halaman kesalahan kustom untuk 4xx kesalahan dalam bucket Amazon S3 di direktori bernama `/4xx-errors`. Distribusi Anda harus menyertakan perilaku cache yang pola jalur yang mengarahkan permintaan halaman kesalahan kustom Anda ke lokasi tersebut, misalnya, `/4xx-errors/*`.
- Nilai dari Asal menentukan nilai dari ID Asal untuk asal yang berisi halaman kesalahan kustom Anda.

Kode tanggapan HTTP

Kode status HTTP yang ingin Anda kembalikan CloudFront ke penampil bersama dengan halaman kesalahan kustom.

Kesalahan caching minimum TTL (detik)

Jumlah minimum waktu yang Anda CloudFront ingin cache respons kesalahan dari server asal Anda.

Pembatasan geografis

Jika Anda perlu mencegah pengguna di negara tertentu mengakses konten Anda, Anda dapat mengonfigurasi CloudFront distribusi Anda dengan Daftar Izinkan atau daftar Blokir. Tidak ada biaya tambahan untuk mengonfigurasi batasan geografis. Untuk informasi selengkapnya, lihat [Batasi distribusi geografis konten Anda](#).

Uji distribusi

Setelah Anda membuat distribusi, CloudFront ketahui di mana server asal Anda berada, dan Anda tahu nama domain yang terkait dengan distribusi. Untuk menguji distribusi Anda, lakukan hal berikut:

1. Tunggu hingga distribusi Anda diterapkan.
 - Lihat Detail distribusi Anda di konsol. Ketika distribusi Anda selesai digunakan, bidang terakhir diubah dari Deploying ke tanggal dan waktu.
2. Buat tautan ke objek Anda dengan nama CloudFront domain dengan menggunakan prosedur berikut.
3. Uji tautannya. CloudFront melayani objek ke halaman web atau aplikasi Anda.

Buat tautan ke objek Anda

Gunakan prosedur berikut untuk membuat tautan pengujian untuk objek dalam distribusi CloudFront web Anda.

Untuk membuat tautan ke objek di distribusi web

1. Salin kode HTML berikut ke file baru, ganti *domain-name* dengan nama domain distribusi Anda, dan ganti *object-name* dengan nama objek Anda.

```
<html>
<head>
  <title>My CloudFront Test</title>
</head>
<body>
  <p>My text content goes here.</p>
  <p></p>
</body>
</html>
```

Misalnya, jika nama domain Anda `d111111abcdef8.cloudfront.net` dan objek Anda `image.jpg`, URL untuk tautan ini adalah:

`https://d111111abcdef8.cloudfront.net/image.jpg`.

Jika objek Anda ada dalam folder di server asal, maka folder tersebut juga harus disertakan di URL. Misalnya, jika `gambar.jpg` terletak di folder `gambar` di server asal Anda, URL akan berupa:

`https://d111111abcdef8.cloudfront.net/images/image.jpg`

2. Simpan kode HTML dalam file yang memiliki ekstensi nama file `.html`.
3. Buka halaman web Anda di browser untuk memastikan bahwa Anda dapat melihat objek Anda.

Browser mengembalikan halaman Anda dengan file gambar yang disematkan, disajikan dari lokasi tepi yang CloudFront ditentukan sesuai untuk melayani objek.

Perbarui distribusi

Di CloudFront konsol, Anda dapat melihat CloudFront distribusi yang terkait dengan Anda Akun AWS, melihat pengaturan untuk distribusi, dan memperbarui sebagian besar pengaturan. Perlu diketahui bahwa perubahan pengaturan yang Anda lakukan tidak akan berpengaruh hingga distribusi telah menyebar ke lokasi edge AWS .

Perbarui distribusi di konsol

Prosedur berikut menunjukkan cara memperbarui CloudFront distribusi di konsol.

Multi-tenant

Untuk memperbarui distribusi multi-tenant

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Cari dan pilih ID distribusi multi-tenant.
3. Pilih tab untuk pengaturan yang ingin Anda perbarui.
4. Lakukan pembaruan, lalu, untuk menyimpan perubahan, pilih Simpan perubahan. Untuk informasi selengkapnya tentang pengaturan yang dapat Anda perbarui, lihat [Referensi pengaturan distribusi yang telah dikonfigurasi sebelumnya](#).

Anda juga dapat memperbarui distribusi dengan menggunakan CloudFront API:

- Untuk memperbarui distribusi, lihat [UpdateDistribution](#) di Referensi Amazon CloudFront API.

Important

Saat Anda memperbarui distribusi, ketahuilah bahwa sejumlah bidang tambahan diperlukan yang tidak diperlukan saat Anda pertama kali membuat distribusi. Untuk membantu memastikan bahwa semua bidang yang diperlukan disertakan saat Anda menggunakan CloudFront API untuk memperbarui distribusi, ikuti langkah-langkah yang dijelaskan [UpdateDistribution](#) dalam Referensi Amazon CloudFront API.

Untuk mengubah distribusi multi-tenant untuk penyewa distribusi, Anda memperbarui penyewa distribusi. Anda juga memperbarui penyewa distribusi untuk memperbarui nilai domain, sertifikat, kustomisasi, atau parameternya. Untuk detail selengkapnya tentang memperbarui sertifikat penyewa distribusi, lihat [Tambahkan domain dan sertifikat \(penyewa distribusi\)](#).

Untuk memperbarui penyewa distribusi

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di bawah SaaS, pilih Penyewa distribusi.
3. Cari penyewa distribusi. Gunakan menu tarik-turun di bilah pencarian untuk memfilter berdasarkan domain, nama, ID distribusi, ID sertifikat, ID grup koneksi, atau ID ACL web.

4. Pilih nama penyewa distribusi.
5. Untuk memperbarui Detail umum, pilih Edit, buat pembaruan, dan pilih Perbarui penyewa distribusi.
6. Pilih tab yang sesuai untuk pengaturan lain untuk memperbarui, membuat pembaruan Anda, dan menyimpannya. Untuk informasi selengkapnya tentang pengaturan penyewa distribusi yang dapat Anda sesuaikan, lihat [Kustomisasi penyewa distribusi](#).

Standard

Untuk memperbarui distribusi standar

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Pilih ID distribusi. Daftar ini mencakup semua distribusi yang terkait dengan AWS akun yang Anda gunakan untuk masuk ke CloudFront konsol.
3. Untuk memperbarui pengaturan umum, pilih Edit. Jika tidak, pilih tab untuk pengaturan yang ingin Anda perbarui.
4. Lakukan pembaruan, lalu pilih Simpan perubahan. Untuk informasi tentang bidang, lihat topik berikut:
 - Pengaturan umum: [Pengaturan distribusi](#)
 - Pengaturan asal: [Pengaturan asal](#)
 - Pengaturan perilaku Cache: [Pengaturan perilaku cache](#)
5. Jika Anda ingin menghapus asal di distribusi Anda, lakukan hal berikut:
 - a. Pilih Perilaku, dan kemudian pastikan Anda telah memindahkan perilaku cache default yang terkait dengan asal ke asal yang lain.
 - b. Pilih Asal, lalu pilih asal.
 - c. Pilih Hapus.

Anda juga dapat memperbarui distribusi dengan menggunakan CloudFront API:

- Untuk memperbarui distribusi, lihat [UpdateDistribution](#) di Referensi Amazon CloudFront API.

 Important

Saat Anda memperbarui distribusi, perhatikan bahwa sejumlah bidang tambahan diperlukan yang tidak diperlukan untuk membuat distribusi. Untuk membantu memastikan bahwa semua bidang yang diperlukan disertakan saat Anda menggunakan CloudFront API untuk memperbarui distribusi, ikuti langkah-langkah yang dijelaskan [UpdateDistribution](#) dalam Referensi Amazon CloudFront API.

Saat Anda menyimpan perubahan pada konfigurasi distribusi Anda, CloudFront mulai menyebarkan perubahan ke semua lokasi tepi. Perubahan konfigurasi berturut-turut menyebar dalam urutannya masing-masing. Sampai konfigurasi Anda diperbarui di lokasi edge, CloudFront terus melayani konten Anda dari lokasi tersebut berdasarkan konfigurasi sebelumnya. Setelah konfigurasi Anda diperbarui di lokasi edge, CloudFront segera mulai menyajikan konten Anda dari lokasi tersebut berdasarkan konfigurasi baru.

Perubahan Anda tidak menyebar ke setiap lokasi tepi secara bersamaan. Saat CloudFront menyebarkan perubahan Anda, kami tidak dapat menentukan apakah lokasi tepi tertentu menyajikan konten Anda berdasarkan konfigurasi sebelumnya atau konfigurasi baru.

 Note

Dalam kasus yang jarang terjadi ketika host atau tautan jaringan terganggu, beberapa lalu lintas penyewa distribusi mungkin dilayani menggunakan konfigurasi lama untuk waktu yang singkat sampai perubahan Anda membuat jalan mereka melalui jaringan.

Untuk melihat kapan perubahan Anda disebarkan, lihat Detail distribusi Anda di konsol. Bidang yang terakhir dimodifikasi berubah dari Deploying ke tanggal dan waktu saat penerapan selesai.

Menandai CloudFront sumber daya

Tag adalah kata atau frasa yang dapat Anda gunakan untuk mengidentifikasi dan mengatur AWS sumber daya Anda. Anda dapat menambahkan beberapa tanda ke setiap sumber daya, dan setiap tanda mencakup kunci dan nilai yang Anda tentukan. Misalnya, kunci dapat berupa "domain" dan nilai dapat berupa "contoh.com". Anda dapat mencari dan memfilter sumber daya Anda berdasarkan tanda yang Anda tambahkan. Untuk informasi selengkapnya tentang kasus penggunaan tag, lihat [Strategi penandaan umum](#) di AWS Sumber Daya Penandaan dan Panduan Editor Tag.

Anda dapat menggunakan tag dengan CloudFront, seperti contoh berikut:

- Menerapkan izin berbasis tag pada distribusi. CloudFront Untuk informasi selengkapnya, lihat [ABAC dengan CloudFront](#).
- Lacak informasi penagihan dalam berbagai kategori. Saat Anda menerapkan tag ke CloudFront distribusi atau AWS sumber daya lainnya (seperti instans Amazon EC2 atau bucket Amazon S3) dan mengaktifkan tag AWS , buat laporan alokasi biaya sebagai nilai yang dipisahkan koma (file CSV) dengan penggunaan dan biaya yang dikumpulkan oleh tag aktif Anda.

Anda dapat menerapkan tanda yang mewakili kategori bisnis (seperti pusat biaya, nama aplikasi, atau pemilik) untuk mengatur biaya Anda di berbagai layanan. Untuk informasi selengkapnya tentang penggunaan tanda untuk alokasi biaya, lihat [Menggunakan Tanda Alokasi Biaya](#) dalam Panduan Pengguna AWS Billing .

Catatan

- Anda dapat menandai distribusi, tetapi Anda tidak dapat menandai identitas atau ketidakabsahan akses asal.
- [Editor Tag](#) dan [grup Sumber Daya](#) saat ini tidak didukung untuk CloudFront.
- Untuk jumlah tanda maksimum saat ini yang dapat Anda tambahkan ke distribusi, lihat [Kuota umum](#).

Daftar Isi

- [Pembatasan tag](#)
- [Menandai sumber daya Anda untuk penagihan](#)
- [Menambahkan, mengedit, dan menghapus tag](#)
- [Penandaan terprogram](#)

Pembatasan tag

Batasan dasar berikut berlaku untuk tag:

- Untuk jumlah maksimum tag per distribusi, lihat [Kuota umum](#).
- Panjang kunci maksimum – 128 karakter Unicode

- Panjang nilai maksimum – 256 karakter Unicode
- Nilai yang valid untuk kunci dan nilai - a-z, A-Z, 0-9, spasi, dan karakter berikut: _./= + - dan @
- Kunci dan nilai tanda peka huruf besar kecil
- Jangan gunakan aws : sebagai awalan untuk kunci. Awalan ini dicadangkan untuk penggunaan AWS .

Menandai sumber daya Anda untuk penagihan

Anda dapat menggunakan tag untuk mengatur AWS tagihan Anda untuk mencerminkan struktur biaya Anda sendiri. Untuk melakukan ini, Anda harus terlebih dahulu mengaktifkan tag untuk alokasi biaya di konsol Billing and Cost Management. Untuk informasi selengkapnya tentang menyiapkan laporan alokasi biaya dengan tag, lihat Laporan [alokasi biaya bulanan di Panduan Pengguna AWS Penagihan](#). Untuk melihat biaya sumber daya gabungan, Anda dapat mengatur informasi penagihan berdasarkan sumber daya yang memiliki nilai kunci tanda yang sama. Misalnya, Anda dapat menandai beberapa sumber daya dengan nama aplikasi tertentu, kemudian mengatur informasi penagihan untuk melihat biaya total aplikasi tersebut pada beberapa layanan. Untuk informasi selengkapnya, lihat [Menggunakan tag alokasi biaya](#) di Panduan Pengguna AWS Penagihan.

Note

Jika Anda baru saja mengaktifkan pelaporan, data untuk bulan yang berjalan dapat dilihat setelah 24 jam.

Menambahkan, mengedit, dan menghapus tag

Anda dapat menggunakan CloudFront konsol untuk mengelola tag untuk CloudFront sumber daya Anda.

Untuk menambahkan tag, mengedit, atau menghapus tag untuk sumber daya

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Arahkan ke sumber daya Anda:
 - Untuk distribusi: Pilih ID untuk distribusi yang ingin Anda perbarui.
 - Untuk fungsi: Pilih Fungsi, lalu pilih nama fungsi yang ingin Anda perbarui.

- Untuk KeyValueStore: Pilih Fungsi, pilih KeyValueStore tab, lalu pilih nama KeyValueStore yang ingin Anda perbarui.
3. Pilih tab Tanda.
 4. Pilih Kelola tanda.
 5. Pada halaman Kelola tag, Anda dapat melakukan hal berikut:
 - Untuk menambahkan tag, masukkan kunci dan, secara opsional, nilai untuk tag. Pilih Tambahkan tag baru untuk menambahkan lebih banyak tag.
 - Untuk mengedit tag, ubah kunci tag atau nilainya, atau keduanya. Anda dapat menghapus nilai untuk tag, tetapi kuncinya diperlukan.
 - Untuk menghapus sebuah tag, pilih Hapus.
 6. Pilih Simpan perubahan.

Penandaan terprogram

Anda juga dapat menggunakan CloudFront API, AWS Command Line Interface (AWS CLI), AWS SDK, dan AWS Tools for Windows PowerShell untuk menerapkan tag. Untuk informasi selengkapnya, lihat topik berikut:

- CloudFront Operasi API:
 - [ListTagsForResource](#)
 - [TagResource](#)
 - [UntagResource](#)
- AWS CLI — Lihat [cloudfront di Referensi](#) Perintah AWS CLI
- AWS [SDK](#) — Lihat dokumentasi SDK yang berlaku di halaman [Dokumentasi AWS](#)
- Alat untuk Windows PowerShell - Lihat [Amazon CloudFront di Referensi Alat AWS untuk PowerShell Cmdlet](#)

Menghapus sebuah distribusi

Prosedur berikut menghapus distribusi dengan menggunakan CloudFront konsol. Untuk informasi tentang menghapus dengan CloudFront API, lihat [DeleteDistribution](#) di Referensi Amazon CloudFront API.

Jika Anda perlu menghapus distribusi dengan OAC yang terpasang pada bucket S3, lihat detail [Hapus distribusi dengan OAC yang terpasang pada bucket S3](#) penting.

Warning

- Sebelum Anda dapat menghapus distribusi, Anda harus menonaktifkannya, yang memerlukan izin untuk memperbarui distribusi. Setelah dihapus, distribusi tidak dapat dipulihkan.
- Jika Anda menonaktifkan distribusi yang memiliki nama domain alternatif yang terkait dengannya, CloudFront berhenti menerima lalu lintas untuk nama domain tersebut (seperti `www.example.com`), meskipun distribusi lain memiliki nama domain alternatif dengan wildcard (*) yang cocok dengan domain yang sama (seperti `*.example.com`).
- Anda tidak dapat menghapus distribusi yang berlangganan paket harga [CloudFront flat-rate](#). Anda akan menerima kesalahan yang mengatakan, “Anda tidak dapat menghapus distribusi ini saat berlangganan paket harga.” Anda harus terlebih dahulu membatalkan paket harga dan kemudian setelah siklus penagihan saat ini, hapus distribusi.

Multi-tenant

Sebelum Anda dapat menghapus distribusi multi-penyewa, Anda harus terlebih dahulu menghapus semua penyewa distribusi terkait darinya.

Untuk menghapus distribusi multi-tenant

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel kanan CloudFront konsol, pilih nama distribusi multi-penyewa yang ingin Anda hapus.
3. Untuk Penyewa, pilih dan hapus semua penyewa distribusi terkait.
4. Pilih Nonaktifkan untuk menonaktifkan distribusi, dan pilih Nonaktifkan distribusi untuk mengonfirmasi.
5. Tunggu sampai timestamp baru muncul di bawah kolom Terakhir dimodifikasi.
 - Mungkin perlu beberapa menit CloudFront untuk menyebarkan perubahan Anda ke semua lokasi tepi.
6. Pilih Hapus, Hapus distribusi.

Untuk menghapus penyewa distribusi

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di bawah SaaS, pilih Penyewa distribusi.
3. Cari penyewa distribusi. Gunakan menu tarik-turun di bilah pencarian untuk memfilter berdasarkan domain, nama, ID distribusi, ID sertifikat, ID grup koneksi, atau ID ACL web.
4. Pilih penyewa distribusi untuk dihapus.
5. Pilih Hapus penyewa, Hapus penyewa distribusi.

Standard

Untuk menghapus distribusi standar

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel kanan CloudFront konsol, temukan distribusi yang ingin Anda hapus.
 - Jika kolom Status menunjukkan bahwa distribusi sudah Dinonaktifkan, lewati ke Langkah 6.
 - Jika Status menunjukkan Diaktifkan tetapi distribusi masih menunjukkan Deploying di kolom Terakhir dimodifikasi, tunggu hingga penerapan selesai sebelum melanjutkan ke langkah 3.
3. Di panel kanan CloudFront konsol, pilih kotak centang untuk distribusi yang ingin Anda hapus.
4. Pilih Nonaktifkan untuk menonaktifkan distribusi, dan memilih Ya, Nonaktifkan untuk mengonfirmasi. Lalu, pilih Tutup.
 - Nilai kolom Status segera berubah menjadi Dinonaktifkan.
5. Tunggu sampai timestamp baru muncul di bawah kolom Terakhir dimodifikasi.
 - Mungkin perlu beberapa menit CloudFront untuk menyebarkan perubahan Anda ke semua lokasi tepi.
6. Pilih kotak centang untuk distribusi yang ingin Anda hapus.
7. Pilih Delete, Delete.

- Jika opsi Hapus tidak tersedia, itu berarti CloudFront masih menyebarkan perubahan Anda ke lokasi tepi. Tunggu hingga timestamp baru muncul di bawah kolom Terakhir dimodifikasi, lalu ulangi langkah 6-7.

Gunakan berbagai asal dengan CloudFront distribusi

Saat Anda membuat distribusi, Anda menentukan asal tempat CloudFront mengirim permintaan untuk file. Anda dapat menggunakan beberapa jenis asal dengan CloudFront. Misalnya, Anda dapat menggunakan bucket Amazon S3, MediaStore container, MediaPackage channel, Application Load Balancer, atau AWS Lambda URL fungsi. Saat Anda membuat CloudFront distribusi, CloudFront secara otomatis mengonfigurasi sebagian besar setelan distribusi untuk Anda, berdasarkan jenis asal konten Anda. Untuk informasi selengkapnya, lihat [Referensi pengaturan distribusi yang telah dikonfigurasi sebelumnya](#).

Jika Anda memiliki Application Load Balancer, Network Load Balancer, atau instans EC2 di subnet pribadi, Anda dapat menggunakannya sebagai asal VPC. Dengan asal VPC, aplikasi Anda hanya dapat diakses di subnet pribadi dengan CloudFront distribusi, yang mencegah aplikasi Anda diakses di internet publik. Untuk informasi selengkapnya, lihat [the section called “Batasi akses dengan asal VPC”](#).

Note

Anda dapat menggunakan fungsi tepi untuk secara dinamis memilih asal yang sesuai untuk setiap permintaan. Dengan menggunakan CloudFront Fungsi atau Lambda @Edge, Anda dapat merutekan permintaan ke asal yang berbeda berdasarkan faktor-faktor seperti lokasi geografis pemirsa, header permintaan, atau parameter string kueri. Untuk informasi selengkapnya, lihat [Sesuaikan di tepi dengan fungsi](#).

Topik

- [Gunakan bucket Amazon S3](#)
- [Gunakan MediaStore wadah atau MediaPackage saluran](#)
- [Menggunakan Application Load Balancer](#)
- [Menggunakan Network Load Balancer](#)
- [Gunakan URL fungsi Lambda](#)

- [Gunakan Amazon EC2 \(atau asal kustom lainnya\)](#)
- [Gunakan grup CloudFront asal](#)
- [Gunakan Amazon API Gateway](#)

Gunakan bucket Amazon S3

Topik berikut menjelaskan berbagai cara Anda dapat menggunakan bucket Amazon S3 sebagai asal distribusi. CloudFront

Topik

- [Gunakan bucket Amazon S3 standar](#)
- [Gunakan Amazon S3 Object Lambda](#)
- [Gunakan Titik Akses Amazon S3](#)
- [Menggunakan bucket Amazon S3 yang dikonfigurasi sebagai titik akhir situs web](#)
- [Tambahkan CloudFront ke bucket Amazon S3 yang ada](#)
- [Pindahkan bucket Amazon S3 ke bucket yang berbeda Wilayah AWS](#)

Gunakan bucket Amazon S3 standar

Saat Anda menggunakan Amazon S3 sebagai asal untuk distribusi Anda, Anda menempatkan objek yang CloudFront ingin Anda kirimkan dalam ember Amazon S3. Anda dapat menggunakan metode apa pun yang didukung oleh Amazon S3 untuk memasukkan objek Anda ke Amazon S3. Misalnya, Anda dapat menggunakan konsol Amazon S3 atau API, atau alat pihak ketiga. Anda dapat membuat hierarki di bucket untuk menyimpan objek, seperti yang Anda lakukan dengan bucket Amazon S3 standar lainnya.

Menggunakan bucket Amazon S3 yang ada karena server CloudFront asal Anda tidak mengubah bucket dengan cara apa pun; Anda masih dapat menggunakannya seperti biasa untuk menyimpan dan mengakses objek Amazon S3 dengan harga Amazon S3 standar. Anda dikenakan biaya Amazon S3 untuk menyimpan objek di dalam bucket. Untuk informasi selengkapnya tentang tagihan yang akan digunakan CloudFront, lihat [CloudFrontHarga Amazon](#). Untuk informasi selengkapnya tentang penggunaan CloudFront dengan bucket S3 yang ada, lihat [the section called “Tambahkan CloudFront ke bucket Amazon S3 yang ada”](#).

 Important

Agar bucket dapat digunakan CloudFront, nama harus sesuai dengan persyaratan penamaan DNS. Untuk informasi selengkapnya, buka [Aturan penamaan Bucket](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Saat Anda menentukan bucket Amazon S3 sebagai asal CloudFront, sebaiknya gunakan format berikut:

bucket-name.s3.*region*.amazonaws.com

Saat Anda menentukan nama keranjang dalam format ini, Anda dapat menggunakan hal berikut CloudFront fitur:

- Konfigurasi CloudFront untuk berkomunikasi dengan bucket Amazon S3 Anda menggunakan SSL/TLS Untuk informasi selengkapnya, lihat [the section called “Gunakan HTTPS dengan CloudFront”](#).
- Gunakan kontrol akses asal untuk mengharuskan pemirsa mengakses konten Anda menggunakan CloudFront URL, bukan dengan menggunakan URL Amazon S3. Untuk informasi selengkapnya, lihat [the section called “Batasi akses ke asal Amazon S3”](#).
- Perbarui konten bucket Anda dengan mengirimkan POST dan PUT meminta. CloudFront Untuk informasi selengkapnya, lihat [the section called “Metode HTTP”](#) dalam topik [the section called “Cara CloudFront memproses dan meneruskan permintaan ke asal Amazon S3 Anda”](#).

Jangan tentukan bucket menggunakan format berikut:

- Gaya jalur Amazon S3: s3.amazonaws.com/*bucket-name*
- Amazon S3 CNAME

 Note

CloudFront mendukung asal S3 menggunakan kelas penyimpanan apa pun, termasuk Intelligent-Tiering S3. Saat CloudFront meminta objek dari asal S3, objek diambil terlepas dari tingkat penyimpanan tempat mereka berada saat ini. Menggunakan CloudFront dengan S3 Intelligent-Tiering tidak memengaruhi kinerja atau fungsionalitas distribusi Anda. Untuk

informasi selengkapnya, lihat [Mengelola biaya penyimpanan dengan Amazon S3 Intelligent-Tiering](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Gunakan Amazon S3 Object Lambda

Saat Anda [membuat Titik Akses Objek Lambda](#), Amazon S3 secara otomatis menghasilkan [alias unik untuk Titik Akses](#) Objek Lambda Anda. Anda dapat [menggunakan alias ini](#) alih-alih nama bucket Amazon S3 sebagai asal distribusi Anda. CloudFront

Bila Anda menggunakan alias Object Lambda Access Point sebagai asal untuk CloudFront, kami sarankan Anda menggunakan format berikut:

alias.s3.*region*.amazonaws.com

Untuk informasi selengkapnya tentang menemukan *alias*, lihat [Cara menggunakan alias gaya ember untuk S3 bucket Object Lambda Access Point di Panduan Pengguna Amazon S3](#).

Important

Bila Anda menggunakan Object Lambda Access Point sebagai asal CloudFront, Anda harus menggunakan kontrol [akses asal](#).

Untuk contoh kasus penggunaan, lihat Menggunakan [Lambda Objek Amazon S3 dengan CloudFront Amazon untuk Menyesuaikan Konten untuk Pengguna Akhir](#).

CloudFront memperlakukan asal Object Lambda Access Point sama dengan asal [bucket Amazon S3 standar](#).

Jika Anda menggunakan Amazon S3 Object Lambda sebagai asal untuk distribusi Anda, Anda harus mengonfigurasi empat izin berikut.

Object Lambda Access Point

Untuk menambahkan izin untuk Object Lambda Access Point

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon S3 di. <https://console.aws.amazon.com/s3/>
2. Di panel navigasi, pilih Titik Akses Objek Lambda.

3. Pilih Object Lambda Access Point yang ingin Anda gunakan.
4. Pilih tab Izin.
5. Pilih Edit di bagian kebijakan Titik Akses Objek Lambda.
6. Tempelkan kebijakan berikut ke bidang Kebijakan.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudfront.amazonaws.com"
      },
      "Action": "s3-object-lambda:Get*",
      "Resource": "arn:aws:s3-object-lambda:us-east-1:123456789012:accesspoint/Object-Lambda-Access-Point-name",
      "Condition": {
        "StringEquals": {
          "aws:SourceArn":
            "arn:aws:cloudfront::123456789012:distribution/CloudFront-distribution-ID"
        }
      }
    }
  ]
}
```

7. Pilih Simpan perubahan.

Amazon S3 Access Point

Untuk menambahkan izin untuk Titik Akses Amazon S3

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon S3 di <https://console.aws.amazon.com/s3/>
2. Di panel navigasi, pilih Access Points.
3. Pilih Titik Akses Amazon S3 yang ingin Anda gunakan.

4. Pilih tab Izin.
5. Pilih Edit di bagian kebijakan Access Point.
6. Tempelkan kebijakan berikut ke bidang Kebijakan.

JSON

```
{
  "Version": "2012-10-17",
  "Id": "default",
  "Statement": [
    {
      "Sid": "s3objlambda",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudfront.amazonaws.com"
      },
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:us-east-1:123456789012:accesspoint/Access-Point-name",
        "arn:aws:s3:us-east-1:123456789012:accesspoint/Access-Point-name/object/*"
      ],
      "Condition": {
        "ForAnyValue:StringEquals": {
          "aws:CalledVia": "s3-object-lambda.amazonaws.com"
        }
      }
    }
  ]
}
```

7. Pilih Simpan.

Amazon S3 bucket

Untuk menambahkan izin ke bucket Amazon S3

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon S3 di. <https://console.aws.amazon.com/s3/>

2. Di panel navigasi, pilih Bucket.
3. Pilih bucket Amazon S3 yang ingin Anda gunakan.
4. Pilih tab Izin.
5. Pilih Edit di bagian Kebijakan Bucket.
6. Tempelkan kebijakan berikut ke bidang Kebijakan.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "*",
      "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ],
      "Condition": {
        "StringEquals": {
          "s3:DataAccessPointAccount": "AWS-account-ID"
        }
      }
    }
  ]
}
```

7. Pilih Simpan perubahan.

AWS Lambda function

Untuk menambahkan izin ke fungsi Lambda

1. Masuk ke Konsol Manajemen AWS dan buka AWS Lambda konsol di <https://console.aws.amazon.com/lambda/>.
2. Di panel navigasi, pilih Fungsi.

3. Pilih AWS Lambda fungsi yang ingin Anda gunakan.
4. Pilih tab Konfigurasi, lalu pilih Izin.
5. Pilih Tambahkan izin di bagian pernyataan Resource-based kebijakan.
6. Pilih Akun AWS.
7. Masukkan nama untuk ID Pernyataan.
8. Masuk `cloudfront.amazonaws.com` untuk Kepala Sekolah.
9. Pilih `lambda:InvokeFunction` dari menu dropdown Action.
10. Pilih Simpan.

Gunakan Titik Akses Amazon S3

Saat Anda [menggunakan Titik Akses S3](#), Amazon S3 secara otomatis menghasilkan alias unik untuk Anda. Anda dapat menggunakan alias ini alih-alih nama bucket Amazon S3 sebagai asal distribusi Anda. CloudFront

Saat Anda menggunakan alias Titik Akses Amazon S3 sebagai asal CloudFront, kami sarankan Anda menggunakan format berikut:

alias.s3.*region*.amazonaws.com

Untuk informasi selengkapnya tentang menemukan *alias*, lihat [Menggunakan alias gaya ember untuk titik akses bucket S3 di](#) Panduan Pengguna Amazon S3.

Important

Saat Anda menggunakan Titik Akses Amazon S3 sebagai asal CloudFront, Anda harus menggunakan kontrol [akses asal](#).

CloudFront memperlakukan asal Titik Akses Amazon S3 sama dengan asal bucket [Amazon S3 standar](#).

Jika Anda menggunakan Amazon S3 Object Lambda sebagai asal untuk distribusi Anda, Anda harus mengonfigurasi dua izin berikut.

Amazon S3 Access Point

Untuk menambahkan izin untuk Titik Akses Amazon S3

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon S3 di. <https://console.aws.amazon.com/s3/>
2. Di panel navigasi, pilih Access Points.
3. Pilih Titik Akses Amazon S3 yang ingin Anda gunakan.
4. Pilih tab Izin.
5. Pilih Edit di bagian kebijakan Access Point.
6. Tempelkan kebijakan berikut ke bidang Kebijakan.

JSON

```
{
  "Version": "2012-10-17",
  "Id": "default",
  "Statement": [
    {
      "Sid": "s3objlambda",
      "Effect": "Allow",
      "Principal": {"Service": "cloudfront.amazonaws.com"},
      "Action": "s3:*",
      "Resource": [
        "arn:aws:s3:us-east-1:123456789012:accesspoint/Access-Point-name",
        "arn:aws:s3:us-east-1:123456789012:accesspoint/Access-Point-name/object/*"
      ],
      "Condition": {
        "StringEquals": {"aws:SourceArn": "arn:aws:cloudfront::123456789012:distribution/CloudFront-distribution-ID"}
      }
    }
  ]
}
```

7. Pilih Simpan.

Amazon S3 bucket

Untuk menambahkan izin ke bucket Amazon S3

1. Masuk ke Konsol Manajemen AWS dan buka konsol Amazon S3 di. <https://console.aws.amazon.com/s3/>
2. Di panel navigasi, pilih Bucket.
3. Pilih bucket Amazon S3 yang ingin Anda gunakan.
4. Pilih tab Izin.
5. Pilih Edit di bagian Kebijakan Bucket.
6. Tempelkan kebijakan berikut ke bidang Kebijakan.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "*"
      },
      "Action": "*",
      "Resource": [
        "arn:aws:s3:::bucket-name",
        "arn:aws:s3:::bucket-name/*"
      ],
      "Condition": {
        "StringEquals": {
          "s3:DataAccessPointAccount": "AWS-account-ID"
        }
      }
    }
  ]
}
```

7. Pilih Simpan perubahan.

Menggunakan bucket Amazon S3 yang dikonfigurasi sebagai titik akhir situs web

Anda dapat menggunakan bucket Amazon S3 yang dikonfigurasi sebagai titik akhir situs web sebagai asal kustom. CloudFront Saat mengonfigurasi CloudFront distribusi, untuk asal, masukkan titik akhir hosting situs web statis Amazon S3 untuk bucket Anda. Nilai ini muncul di [konsol Amazon S3](#), pada tab Properties, di panel hosting situs web statis. Contoh:

```
http://bucket-name.s3-website-region.amazonaws.com
```

Untuk informasi selengkapnya tentang menentukan titik akhir situs web statis Amazon S3, [lihat Titik akhir situs web](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Saat Anda menentukan nama bucket dalam format ini sebagai asal Anda, Anda dapat menggunakan pengalihan Amazon S3 dan dokumen kesalahan kustom Amazon S3. Untuk informasi selengkapnya, lihat [Mengonfigurasi dokumen kesalahan kustom](#) dan [Mengonfigurasi pengalihan di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon](#). (CloudFront juga menyediakan halaman kesalahan kustom. Untuk informasi lebih lanjut, lihat [the section called “Buat halaman kesalahan khusus untuk kode status HTTP tertentu”](#).)

Menggunakan bucket Amazon S3 sebagai server CloudFront asal Anda tidak mengubah bucket dengan cara apa pun. Anda masih dapat menggunakannya seperti biasa dan Anda mengeluarkan biaya Amazon S3 biasa. Untuk informasi selengkapnya tentang tagihan yang akan digunakan CloudFront, lihat [CloudFront Harga Amazon](#).

Note

Jika Anda menggunakan CloudFront API untuk membuat distribusi dengan bucket Amazon S3 yang dikonfigurasi sebagai titik akhir situs web, Anda harus mengonfigurasinya dengan menggunakan `CustomOriginConfig`, meskipun situs web dihosting di bucket Amazon S3. Untuk informasi selengkapnya tentang membuat distribusi menggunakan CloudFront API, lihat [CreateDistribution](#) di Referensi Amazon CloudFront API.

Tambahkan CloudFront ke bucket Amazon S3 yang ada

Jika Anda menyimpan objek Anda di bucket Amazon S3, Anda dapat meminta pengguna mendapatkan objek Anda langsung dari S3, atau Anda dapat mengonfigurasi CloudFront untuk mendapatkan objek Anda dari S3 dan kemudian mendistribusikannya ke pengguna Anda. Penggunaan CloudFront bisa lebih hemat biaya jika pengguna Anda sering mengakses objek Anda karena, pada penggunaan yang lebih tinggi, harga untuk transfer CloudFront data lebih

rendah daripada harga untuk transfer data Amazon S3. Selain itu, unduhan lebih cepat CloudFront dibandingkan dengan Amazon S3 saja karena objek Anda disimpan lebih dekat ke pengguna Anda.

Note

Jika Anda CloudFront ingin menghormati pengaturan berbagi sumber daya lintas asal Amazon S3, konfigurasi CloudFront untuk meneruskan `Origin` header ke Amazon S3. Untuk informasi selengkapnya, lihat [the section called “Konten cache berdasarkan header permintaan”](#).

Jika saat ini Anda mendistribusikan konten langsung dari bucket Amazon S3 menggunakan nama domain Anda sendiri (seperti `example.com`), bukan nama domain bucket Amazon S3 Anda (seperti `amzn-s3-demo-bucket.s3.us-west-2.amazonaws.com`), Anda dapat menambahkan tanpa gangguan dengan menggunakan prosedur berikut. CloudFront

Untuk menambahkan CloudFront ketika Anda sudah mendistribusikan konten Anda dari Amazon S3

1. Buat CloudFront distribusi. Untuk informasi selengkapnya, lihat [the section called “Buat distribusi”](#).

Saat Anda membuat distribusi, sebutkan nama bucket Amazon S3 Anda sebagai server asal.

Important

Agar bucket dapat digunakan CloudFront, nama harus sesuai dengan persyaratan penamaan DNS. Untuk informasi selengkapnya, buka [Aturan penamaan Bucket](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Jika Anda menggunakan CNAME dengan Amazon S3, sebutkan pula CNAME untuk distribusi Anda.

2. Buat halaman web uji yang berisi tautan ke objek yang dapat dibaca publik di bucket Amazon S3, dan uji tautan. Untuk pengujian awal ini, gunakan nama CloudFront domain distribusi Anda di URL objek, misalnya, `https://d1111111abcdef8.cloudfront.net/images/image.jpg`.

Untuk informasi selengkapnya tentang format CloudFront URL, lihat [the section called “Kustomisasi URL file”](#).

3. Jika Anda menggunakan Amazon S3 CNames, aplikasi Anda menggunakan nama domain Anda (misalnya, example.com) untuk mereferensikan objek di bucket Amazon S3 alih-alih menggunakan nama bucket Anda (misalnya, amzn-s3-demo-bucket.s3.amazonaws.com). Untuk terus menggunakan nama domain Anda ke objek referensi alih-alih menggunakan nama CloudFront domain untuk distribusi Anda (misalnya, d111111abcdef8.cloudfront.net), Anda perlu memperbarui pengaturan Anda dengan penyedia layanan DNS Anda.

Untuk Amazon S3 CNAMEs agar berfungsi, penyedia layanan DNS Anda harus memiliki catatan sumber daya CNAME untuk domain Anda yang saat ini merutekan kueri domain ke bucket Amazon S3. Misalnya, jika pengguna meminta objek ini:

```
https://example.com/images/image.jpg
```

Permintaan secara otomatis dialihkan rutenya, dan pengguna melihat objek ini:

```
https://amzn-s3-demo-bucket.s3.amazonaws.com/images/image.jpg
```

Untuk merutekan kueri ke CloudFront distribusi, bukan bucket Amazon S3, Anda harus menggunakan metode yang disediakan oleh penyedia layanan DNS untuk memperbarui kumpulan data sumber daya CNAME untuk domain Anda. Catatan CNAME yang diperbarui ini mengalihkan kueri DNS dari domain Anda ke nama domain untuk distribusi Anda. CloudFront Untuk informasi lebih lanjut, lihat dokumentasi yang disediakan oleh penyedia layanan DNS Anda.

Note

Jika Anda menggunakan Route 53 sebagai layanan DNS Anda, Anda dapat menggunakan set data sumber daya CNAME atau rekaman sumber daya alias. Untuk informasi tentang mengedit kumpulan rekaman sumber daya, lihat [Mengedit catatan](#). Untuk informasi tentang kumpulan rekaman sumber daya alias, lihat [Memilih antara catatan alias dan non-alias](#). Kedua topik tersebut berada dalam Panduan Pengembang Amazon Route 53.

Untuk informasi selengkapnya tentang menggunakan CNames with CloudFront, lihat [the section called “Gunakan URL khusus”](#).

Setelah Anda memperbarui kumpulan catatan sumber daya CNAME, mungkin diperlukan waktu hingga 72 jam untuk perubahan menyebar ke seluruh sistem DNS, meskipun biasanya hal

itu terjadi lebih cepat. Selama waktu ini, beberapa permintaan untuk konten Anda akan terus dikirimkan ke keranjang Amazon S3, dan yang lainnya akan diarahkan ke CloudFront.

Pindahkan bucket Amazon S3 ke bucket yang berbeda Wilayah AWS

Jika Anda menggunakan Amazon S3 sebagai asal CloudFront distribusi dan Anda memindahkan bucket ke yang lain Wilayah AWS, CloudFront dapat memakan waktu hingga satu jam untuk memperbarui catatannya agar menggunakan Wilayah baru jika kedua hal berikut ini benar:

- Anda menggunakan identitas akses CloudFront asal (OAI) untuk membatasi akses ke bucket.
- Anda memindahkan bucket ke Amazon S3 Region yang memerlukan Signature Version 4 untuk otentikasi.

Saat Anda menggunakan OAI, CloudFront gunakan Region (di antara nilai lainnya) untuk menghitung tanda tangan yang digunakan untuk meminta objek dari bucket Anda. Untuk informasi selengkapnya tentang cara membuat pekerjaan OAI, lihat [the section called “Gunakan identitas akses asal \(warisan, tidak disarankan\)”](#). Untuk daftar dukungan Signature Version 2, lihat [proses penandatanganan Signature Version 2](#) di Referensi Umum Amazon Web Services. Wilayah AWS

Untuk memaksa pembaruan yang lebih cepat ke CloudFront catatan, Anda dapat memperbarui CloudFront distribusi Anda, misalnya, dengan memperbarui bidang Deskripsi pada tab Umum di CloudFront konsol. Saat Anda memperbarui distribusi, CloudFront segera periksa Wilayah tempat bucket Anda berada. Perbanyak perubahan ke semua lokasi tepi hanya memakan waktu beberapa menit.

Gunakan MediaStore wadah atau MediaPackage saluran

Untuk melakukan streaming video CloudFront, Anda dapat menyiapkan bucket Amazon S3 yang dikonfigurasi sebagai MediaStore wadah, atau membuat saluran dan titik akhir dengan MediaPackage. Kemudian Anda membuat dan mengkonfigurasi distribusi CloudFront untuk streaming video.

Untuk informasi selengkapnya dan petunjuk langkah demi langkah, lihat topik berikut:

- [the section called “Sajikan video dengan menggunakan AWS Elemental MediaStore sebagai asal”](#)
- [the section called “Sajikan video langsung yang diformat dengan AWS Elemental MediaPackage”](#)

Menggunakan Application Load Balancer

Anda dapat menggunakan CloudFront untuk merutekan lalu lintas ke Penyeimbang Beban Aplikasi internal dan yang menghadap ke internet.

Jika asal Anda adalah satu atau beberapa server HTTP (S) (server web) yang dihosting pada satu atau beberapa instans Amazon EC2, Anda dapat memilih untuk menggunakan Application Load Balancer yang menghadap ke internet untuk mendistribusikan lalu lintas ke instans. Penyeimbang beban yang menghadap ke internet memiliki nama DNS yang dapat diselesaikan secara publik dan mengarahkan permintaan dari klien ke target melalui internet.

Untuk informasi selengkapnya tentang menggunakan Application Load Balancer yang menghadap ke internet sebagai asal CloudFront Anda, termasuk cara memastikan bahwa pemirsa hanya dapat mengakses server web Anda CloudFront melalui dan bukan dengan mengakses penyeimbang beban secara langsung, lihat. [the section called “Batasi akses ke Application Load Balancers”](#)

Atau, Anda dapat menggunakan asal VPC untuk mengirimkan konten dari aplikasi yang di-host dengan Application Load Balancer internal di subnet pribadi virtual private cloud (VPC) Anda. Asal VPC mencegah aplikasi Anda diakses di internet publik. Untuk informasi selengkapnya, lihat [Batasi akses dengan asal VPC](#).

Menggunakan Network Load Balancer

Anda dapat menggunakan Network Load Balancer internal dan internet dengan Amazon. CloudFront Anda dapat menggunakan Network Load Balancer internal di dalam subnet pribadi dengan CloudFront menggunakan asal VPC. CloudFront Asal VPC memungkinkan Anda untuk menyajikan konten dari aplikasi yang dihosting di subnet VPC pribadi tanpa memaparkannya ke internet publik. Untuk informasi selengkapnya, lihat [Batasi akses dengan asal VPC](#).

Atau, Anda juga dapat menggunakan CloudFront untuk mengirimkan lalu lintas dari Network Load Balancers yang menghadap ke internet. Penyeimbang beban yang menghadap ke internet memiliki nama DNS yang dapat diselesaikan secara publik dan dapat menerima permintaan dari klien di internet dan distribusi. CloudFront

Gunakan URL fungsi Lambda

[URL fungsi Lambda](#) adalah titik akhir HTTPS khusus untuk fungsi Lambda. Anda dapat menggunakan URL fungsi Lambda untuk membangun aplikasi web tanpa server sepenuhnya di

dalam Lambda. Anda dapat memanggil aplikasi web Lambda secara langsung melalui URL fungsi, tanpa perlu mengintegrasikan dengan API Gateway atau Application Load Balancer.

Jika Anda membangun aplikasi web tanpa server dengan menggunakan fungsi Lambda dengan URL fungsi, Anda dapat menambahkan CloudFront untuk mendapatkan manfaat berikut:

- Mempercepat aplikasi Anda dengan menyimpan konten yang lebih dekat dengan pemirsa
- Gunakan nama domain khusus untuk aplikasi web Anda
- Rutekan jalur URL yang berbeda ke fungsi Lambda yang berbeda menggunakan perilaku cache CloudFront
- Blokir permintaan tertentu menggunakan batasan CloudFront geografis atau AWS WAF (atau keduanya)
- Gunakan AWS WAF dengan CloudFront untuk membantu melindungi aplikasi Anda dari bot berbahaya, membantu mencegah eksploitasi aplikasi umum, dan meningkatkan perlindungan dari serangan DDoS

Untuk menggunakan URL fungsi Lambda sebagai asal CloudFront distribusi, tentukan nama domain lengkap URL fungsi Lambda sebagai domain asal. Nama domain URL fungsi Lambda menggunakan format berikut:

`function-URL-ID.lambda-url.AWS-Region.on.aws`

Saat Anda menggunakan URL fungsi Lambda sebagai asal CloudFront distribusi, URL fungsi harus dapat diakses publik. Untuk melakukannya, gunakan salah satu opsi berikut:

- Jika Anda menggunakan kontrol akses asal (OAC), AuthType parameter URL fungsi Lambda harus menggunakan `AWS_IAM` nilai dan mengizinkan serta izin dalam `lambda:InvokeFunctionUrl` kebijakan `lambda:InvokeFunction` berbasis sumber daya. Untuk informasi selengkapnya tentang penggunaan URL fungsi Lambda untuk OAC, lihat. [Batasi akses ke asal URL AWS Lambda fungsi](#)
- Jika Anda tidak menggunakan OAC, Anda dapat menyetel AuthType parameter URL fungsi `NONE` dan mengizinkan `lambda:InvokeFunctionUrl` izin dalam kebijakan berbasis sumber daya.

Anda juga dapat [menambahkan header asal kustom](#) ke permintaan yang CloudFront dikirim ke asal, dan menulis kode fungsi untuk mengembalikan respons kesalahan jika header tidak ada dalam

permintaan. Ini membantu memastikan bahwa pengguna hanya dapat mengakses aplikasi web Anda melalui CloudFront, tidak secara langsung menggunakan URL fungsi Lambda.

Untuk informasi selengkapnya tentang URL fungsi Lambda, lihat topik berikut di Panduan Pengembang AWS Lambda :

- [URL fungsi Lambda - Gambaran umum fitur URL](#) fungsi Lambda
- [Memanggil URL fungsi Lambda](#) - Termasuk detail tentang permintaan dan muatan respons yang akan digunakan untuk pengkodean aplikasi web tanpa server Anda
- [Model keamanan dan autentikasi untuk URL fungsi Lambda](#) - Termasuk detail tentang jenis autentikasi Lambda

Gunakan Amazon EC2 (atau asal kustom lainnya)

Anda dapat menggunakan instans EC2 internal dan internet dengan Amazon. CloudFront Anda dapat menggunakan instans EC2 internal di dalam subnet pribadi dengan CloudFront menggunakan asal VPC. CloudFront Asal VPC memungkinkan Anda untuk menyajikan konten dari aplikasi yang dihosting di subnet VPC pribadi tanpa memaparkannya ke internet publik. Untuk informasi selengkapnya, lihat [Batasi akses dengan asal VPC](#).

Asal kustom adalah server web HTTP (S) dengan nama DNS yang dapat diselesaikan secara publik yang merutekan permintaan dari klien ke target melalui internet. Server HTTP (S) dapat di-host di AWS—misalnya, instans Amazon EC2—atau dihosting di tempat lain. Asal Amazon S3 yang dikonfigurasi sebagai titik akhir situs web juga dianggap sebagai asal kustom. Untuk informasi selengkapnya, lihat [the section called “Menggunakan bucket Amazon S3 yang dikonfigurasi sebagai titik akhir situs web”](#).

Bila Anda menggunakan server HTTP Anda sendiri sebagai custom origin, Anda menentukan nama DNS server, bersama dengan port HTTP dan HTTPS dan protokol yang CloudFront ingin Anda gunakan saat mengambil objek dari asal Anda.

Sebagian besar CloudFront fitur didukung saat Anda menggunakan custom origin dengan pengecualian konten pribadi. Meskipun Anda dapat menggunakan URL yang ditandatangani untuk mendistribusikan konten dari asal kustom, CloudFront untuk mengakses asal kustom, asal harus tetap dapat diakses publik. Untuk informasi selengkapnya, lihat [the section called “Batasi konten dengan URL yang ditandatangani dan cookie yang ditandatangani”](#).

Ikuti panduan ini untuk menggunakan instans Amazon EC2 dan asal usul lainnya dengan CloudFront.

- Host dan sajikan konten yang sama di semua server yang menyajikan konten untuk CloudFront asal yang sama. Untuk informasi selengkapnya, lihat [the section called “Pengaturan asal”](#) dalam topik [the section called “Semua pengaturan distribusi”](#).
- Log entri X-Amz-Cf-Id header di semua server jika Anda membutuhkan Dukungan atau CloudFront menggunakan nilai ini untuk debugging.
- Batasi permintaan ke port HTTP dan HTTPS tempat asal kustom Anda mendengarkan.
- Sinkronkan jam semua server dalam implementasi Anda. Perhatikan bahwa CloudFront menggunakan Coordinated Universal Time (UTC) untuk URL yang ditandatangani dan cookie yang ditandatangani, untuk log, dan laporan. Selain itu, jika Anda memantau CloudFront aktivitas menggunakan CloudWatch metrik, perhatikan bahwa CloudWatch juga menggunakan UTC.
- Gunakan server redundan untuk menangani kegagalan.
- Untuk informasi tentang penggunaan asal kustom untuk menyajikan konten pribadi, lihat [the section called “Batasi akses ke file pada asal kustom”](#).
- Untuk informasi tentang permintaan dan perilaku respons dan tentang kode status HTTP yang didukung, lihat [Perilaku permintaan dan respons](#).

Jika Anda menggunakan Amazon EC2 untuk asal kustom, kami sarankan Anda melakukan hal berikut:

- Gunakan Amazon Machine Image yang secara otomatis menginstal perangkat lunak untuk server web. Untuk informasi lebih lanjut, lihat [Dokumentasi Amazon EC2](#).
- Gunakan load balancer Elastic Load Balancing untuk menangani lalu lintas di beberapa instans Amazon EC2 dan untuk mengisolasi aplikasi Anda dari perubahan ke instans Amazon EC2. Misalnya, jika Anda menggunakan load balancer, Anda bisa menambahkan dan menghapus instans Amazon EC2 tanpa mengubah aplikasi Anda. Untuk informasi lebih lanjut, lihat [Dokumentasi Penyeimbangan Beban Elastis](#).
- Saat Anda membuat CloudFront distribusi, tentukan URL penyeimbang beban untuk nama domain server asal Anda. Untuk informasi selengkapnya, lihat [the section called “Buat distribusi”](#).

Gunakan grup CloudFront asal

Anda dapat menentukan grup asal untuk CloudFront asal Anda jika, misalnya, Anda ingin mengonfigurasi failover asal untuk skenario saat Anda membutuhkan ketersediaan tinggi. Gunakan failover asal untuk menentukan asal primer CloudFront ditambah asal kedua yang CloudFront secara otomatis beralih ke saat asal utama mengembalikan respons kegagalan kode status HTTP tertentu.

Untuk informasi selengkapnya, termasuk langkah-langkah untuk menyiapkan grup asal, lihat [the section called “Tingkatkan ketersediaan dengan failover asal”](#).

Gunakan Amazon API Gateway

Anda dapat menggunakan API Gateway sebagai custom origin untuk CloudFront distribusi Anda. Untuk informasi selengkapnya, lihat topik berikut:

- [Mengamankan Amazon API Gateway dengan cipher aman menggunakan posting blog Amazon CloudFront AWS](#)
- [Bagaimana cara mengatur API Gateway dengan CloudFront distribusi saya sendiri?](#) AWS re:Post

Aktifkan IPv6 untuk distribusi CloudFront

Amazon CloudFront mendukung IPv4 dan IPv6 dari klien ke lokasi edge. AWS CloudFront juga mendukung konektivitas IPv6 dan dual-stack (IPv4 dan IPv6) menuju asal. Ini membantu Anda mencapai pengiriman IPv6 ujung ke ujung.

IPv6 adalah protokol internet generasi berikutnya yang dirancang untuk menggantikan IPv4. Sementara IPv4 menggunakan alamat 32-bit (seperti 192.0.2.44), IPv6 menggunakan alamat 128-bit (seperti 2001:0 db 8:85 a3: :8a2e: 0370:7334). IPv6 menyediakan ruang alamat yang diperluas untuk mengakomodasi lebih banyak perangkat yang terhubung ke internet.

Topik

- [Permintaan penampil IPv6](#)
- [Permintaan asal IPv6](#)

Permintaan penampil IPv6

Secara umum, Anda harus mengaktifkan IPv6 jika Anda memiliki pengguna pada IPv6 yang ingin mengakses konten Anda. Namun, jika Anda menggunakan URL yang ditandatangani atau cookie yang ditandatangani untuk membatasi akses ke konten, dan jika Anda menggunakan kebijakan khusus yang mencakup parameter `IpAddress` untuk membatasi alamat IP yang dapat mengakses konten Anda, tidak mengaktifkan IPv6. Jika Anda ingin membatasi akses ke sebagian konten dengan alamat IP dan tidak membatasi akses ke konten lain (atau membatasi akses tetapi tidak oleh alamat IP), Anda dapat membuat dua distribusi. Untuk informasi tentang pembuatan URL yang

ditandatangani dengan menggunakan kebijakan kustom, lihat [Membuat URL yang ditandatangani menggunakan kebijakan khusus](#). Untuk informasi tentang membuat cookie yang ditandatangani dengan menggunakan kebijakan kustom, lihat [Tetapkan cookie yang ditandatangani menggunakan kebijakan khusus](#).

Jika Anda menggunakan catatan sumber daya Route 53 alias diatur untuk mengarahkan lalu lintas ke CloudFront Anda perlu membuat rekaman sumber daya alias kedua jika kedua hal berikut benar:

- Anda mengaktifkan IPv6 untuk distribusi
- Anda menggunakan nama domain alternatif di URL untuk objek Anda

Untuk informasi selengkapnya, lihat [Merutekan lalu lintas ke CloudFront distribusi Amazon dengan menggunakan nama domain Anda](#) di Panduan Pengembang Amazon Route 53.

Jika Anda membuat data sumber daya CNAME, baik dengan Route 53 atau dengan layanan DNS lainnya, Anda tidak perlu melakukan perubahan apa pun. CNAME mencatat lalu lintas ke distribusi Anda tanpa memandang format alamat IP penampil.

Jika Anda mengaktifkan IPv6 dan CloudFront mengakses log, `c-ip` kolom menyertakan nilai dalam format IPv4 dan IPv6. Untuk informasi selengkapnya, lihat [Bidang file log](#).

Note

Untuk menjaga ketersediaan pelanggan yang tinggi, CloudFront tanggap permintaan pemirsa dengan menggunakan IPv4 jika data kami menunjukkan bahwa IPv4 akan memberikan pengalaman pengguna yang lebih baik. Untuk mengetahui persentase permintaan yang CloudFront disajikan melalui IPv6, aktifkan CloudFront pencatatan untuk distribusi Anda dan uraikan `c-ip` kolom, yang berisi alamat IP penampil yang membuat permintaan. Persentase ini harus tumbuh seiring berjalannya waktu, tetapi akan tetap menjadi minoritas lalu lintas seperti IPv6 belum didukung oleh semua jaringan penampil secara global. Beberapa jaringan penampil memiliki keunggulan IPv6 tetapi yang lain tidak mendukung IPv6 sama sekali. (Jaringan penampil serupa dengan internet rumah atau operator nirkabel Anda.)

[Untuk informasi lebih lanjut tentang dukungan kami untuk IPv6, lihat FAQ. CloudFront](#) Untuk informasi tentang mengaktifkan log akses, lihat [Awalan log](#) bidang [Pencatatan standar](#) dan.

Permintaan asal IPv6

Bila Anda menggunakan custom origin (tidak termasuk asal Amazon S3 dan VPC), Anda dapat menyesuaikan pengaturan asal untuk distribusi Anda untuk memilih CloudFront cara menghubungkan ke asal Anda menggunakan alamat IPv4 atau IPv6. Untuk asal kustom (tidak termasuk asal Amazon S3 dan VPC), Anda memiliki opsi konektivitas berikut:

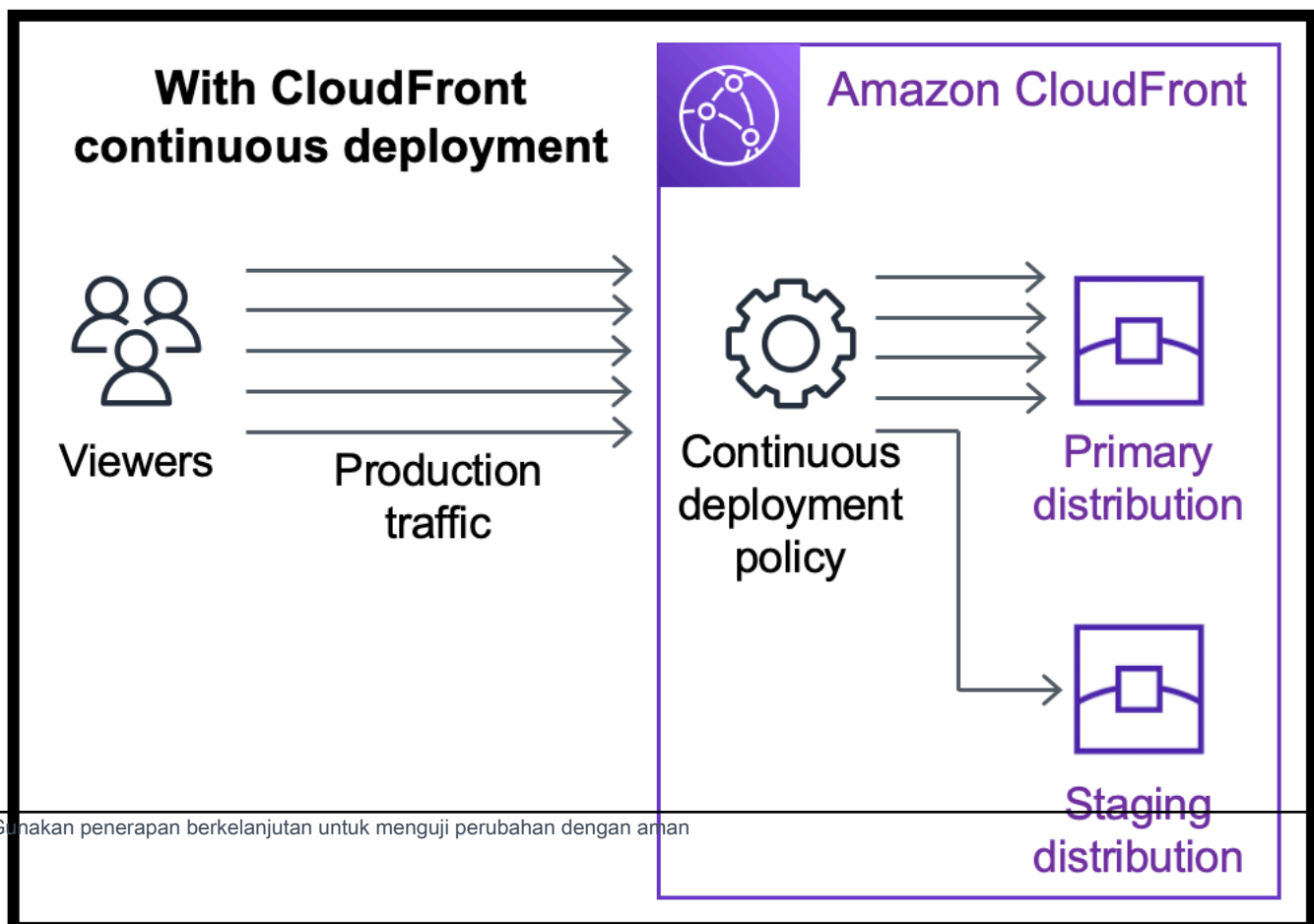
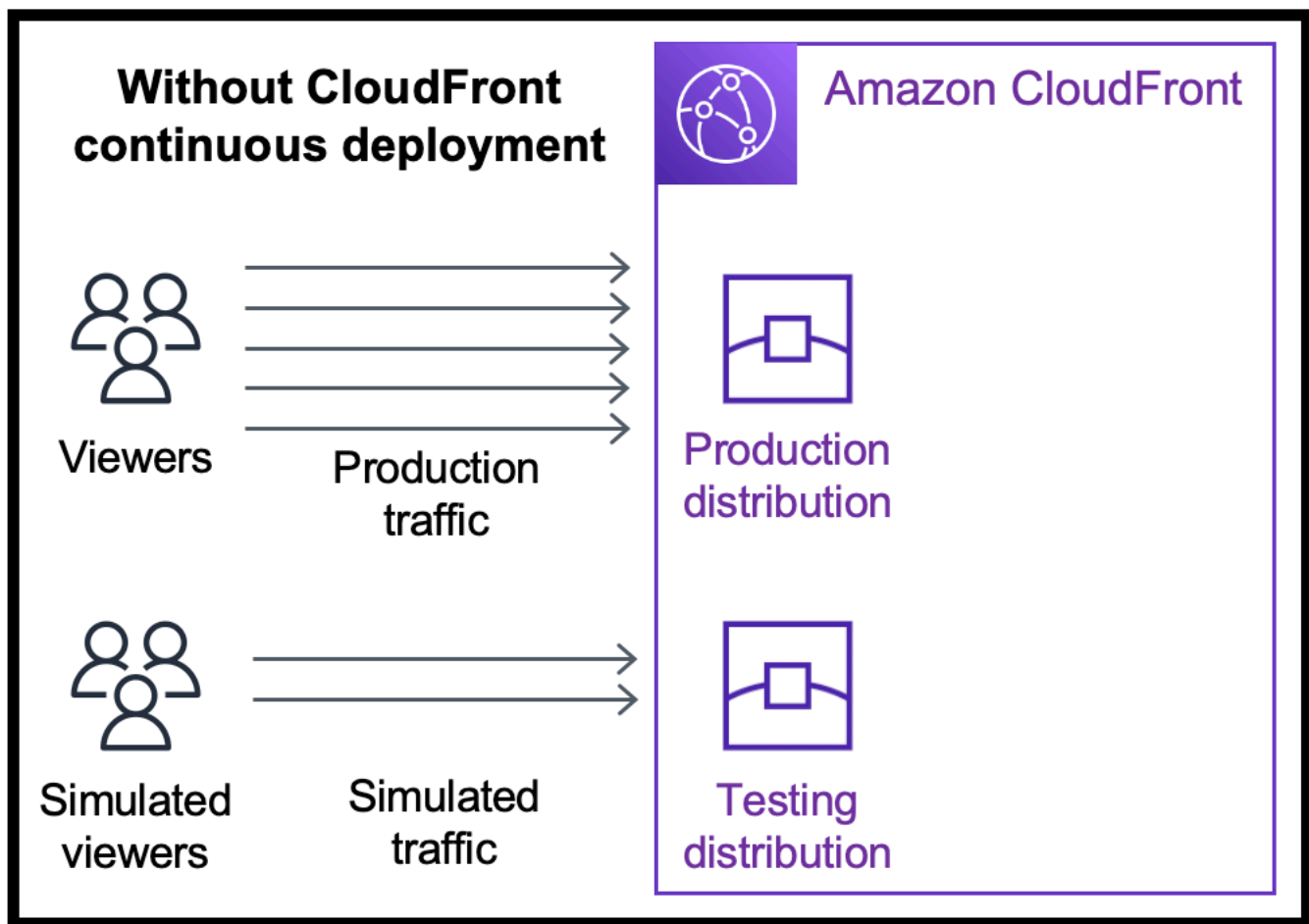
- Hanya IPv4 (default) - Ini adalah konfigurasi default yang CloudFront digunakan untuk terhubung ke asal melalui IPv4.
- Hanya IPv6 - Memerlukan domain asal Anda untuk menyelesaikan ke alamat IPv6. CloudFront akan secara eksklusif menggunakan alamat IPv6 untuk koneksi asal.
- Dual-stack— Memungkinkan koneksi melalui IPv4 dan IPv6. CloudFront Secara otomatis memilih konektivitas asal IPv4 atau IPv6 untuk memprioritaskan kinerja dan ketersediaan sehingga Anda dapat menggunakan CloudFront sebagai gateway internet dual-stack IPv6 dan IPv4 untuk aplikasi web Anda.

Pilih opsi yang sesuai dengan konfigurasi jaringan asal Anda dan persyaratan konektivitas. Untuk informasi selengkapnya, lihat [Merancang DNS untuk pertimbangan keamanan dan pemantauan IPv6](#) dan [IPv6](#).

Gunakan penerapan CloudFront berkelanjutan untuk menguji perubahan konfigurasi CDN dengan aman

Dengan penerapan CloudFront berkelanjutan Amazon, Anda dapat menerapkan perubahan ke konfigurasi CDN dengan aman dengan menguji terlebih dahulu dengan subset lalu lintas produksi. Anda dapat menggunakan distribusi pementasan dan kebijakan penerapan berkelanjutan untuk mengirim beberapa lalu lintas dari pemirsa nyata (produksi) ke konfigurasi CDN baru dan memvalidasi bahwa itu berfungsi seperti yang diharapkan. Anda dapat memantau kinerja konfigurasi baru secara real time, dan mempromosikan konfigurasi baru untuk melayani semua lalu lintas melalui distribusi utama saat Anda siap.

Diagram berikut menunjukkan manfaat menggunakan penerapan CloudFront berkelanjutan. Tanpa itu, Anda harus menguji perubahan konfigurasi CDN dengan lalu lintas simulasi. Dengan penerapan berkelanjutan, Anda dapat menguji perubahan dengan subset lalu lintas produksi, lalu mempromosikan perubahan ke distribusi utama saat Anda siap.



Pelajari lebih lanjut tentang bekerja dengan penerapan berkelanjutan dalam topik berikut.

Topik

- [CloudFront alur kerja penerapan berkelanjutan](#)
- [Bekerja dengan distribusi pementasan dan kebijakan penyebaran berkelanjutan](#)
- [Pantau distribusi pementasan](#)
- [Pelajari cara kerja penerapan berkelanjutan](#)
- [Kuota dan pertimbangan lain untuk penyebaran berkelanjutan](#)

CloudFront alur kerja penerapan berkelanjutan

Alur kerja tingkat tinggi berikut menjelaskan cara menguji dan menerapkan perubahan konfigurasi dengan aman dengan CloudFront penerapan berkelanjutan.

1. Pilih distribusi yang ingin Anda gunakan sebagai distribusi utama. Distribusi utama adalah salah satu yang saat ini melayani lalu lintas produksi.
2. Dari distribusi primer, buat distribusi pementasan. Distribusi pementasan dimulai sebagai salinan distribusi utama.
3. Buat konfigurasi lalu lintas di dalam kebijakan penerapan berkelanjutan, dan lampirkan ke distribusi utama. Ini menentukan bagaimana CloudFront rute lalu lintas ke distribusi pementasan. Untuk informasi selengkapnya tentang permintaan perutean ke distribusi pementasan, lihat [the section called “Permintaan rute ke distribusi pementasan”](#).
4. Perbarui konfigurasi distribusi pementasan. Untuk informasi selengkapnya tentang pengaturan yang dapat Anda perbarui, lihat [the section called “Perbarui distribusi primer dan pementasan”](#).
5. Pantau distribusi pementasan untuk menentukan apakah perubahan konfigurasi berfungsi seperti yang diharapkan. Untuk informasi selengkapnya tentang pemantauan distribusi pementasan, lihat [the section called “Pantau distribusi pementasan”](#).

Saat Anda memantau distribusi pementasan, Anda dapat:

- Perbarui konfigurasi distribusi pementasan lagi, untuk melanjutkan pengujian perubahan konfigurasi.
 - Perbarui kebijakan penerapan berkelanjutan (konfigurasi lalu lintas) untuk mengirim lebih banyak atau lebih sedikit lalu lintas ke distribusi pementasan.
6. Bila Anda puas dengan kinerja distribusi pementasan, promosikan konfigurasi distribusi pementasan ke distribusi utama, yang menyalin konfigurasi distribusi pementasan ke distribusi

utama. Ini juga menonaktifkan kebijakan penerapan berkelanjutan yang berarti bahwa CloudFront merutekan semua lalu lintas ke distribusi utama.

Anda dapat membangun otomatisasi yang memantau kinerja distribusi pementasan (langkah 5) dan mempromosikan konfigurasi secara otomatis (langkah 6) ketika kriteria tertentu terpenuhi.

Setelah Anda mempromosikan konfigurasi, Anda dapat menggunakan kembali distribusi pementasan yang sama saat berikutnya Anda ingin menguji perubahan konfigurasi.

Untuk informasi selengkapnya tentang bekerja dengan distribusi pementasan dan kebijakan penerapan berkelanjutan di CloudFront konsol, API AWS CLI, atau CloudFront API, lihat bagian berikut.

Bekerja dengan distribusi pementasan dan kebijakan penyebaran berkelanjutan

Anda dapat membuat, memperbarui, dan memodifikasi distribusi pementasan dan kebijakan penerapan berkelanjutan di CloudFront konsol, dengan AWS Command Line Interface (AWS CLI), atau dengan API. CloudFront

Membuat distribusi pementasan dengan kebijakan penerapan berkelanjutan

Prosedur berikut menunjukkan kepada Anda cara membuat distribusi pementasan dengan kebijakan penerapan berkelanjutan.

Console

Anda dapat membuat distribusi pementasan dengan kebijakan penerapan berkelanjutan dengan menggunakan Konsol Manajemen AWS

Untuk membuat distribusi pementasan dan kebijakan penerapan berkelanjutan (konsol)

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Distribusi.
3. Pilih distribusi yang ingin Anda gunakan sebagai distribusi utama. Distribusi utama adalah distribusi yang saat ini melayani lalu lintas produksi, yang darinya Anda akan membuat distribusi pementasan.

4. Di bagian Penerapan berkelanjutan, pilih Buat distribusi pementasan. Ini membuka wizard distribusi pementasan Buat.
5. Di wizard Create staging distribution, lakukan hal berikut:
 - a. (Opsional) Ketik deskripsi untuk distribusi pementasan.
 - b. Pilih Berikutnya.
 - c. Ubah konfigurasi distribusi pementasan. Untuk informasi selengkapnya tentang pengaturan yang dapat Anda perbarui, lihat [the section called “Perbarui distribusi primer dan pementasan”](#).

Setelah Anda selesai memodifikasi konfigurasi distribusi pementasan, pilih Berikutnya.

- d. Gunakan konsol untuk menentukan konfigurasi Lalu Lintas. Ini menentukan bagaimana CloudFront rute lalu lintas ke distribusi pementasan. (CloudFront menyimpan konfigurasi lalu lintas dalam kebijakan penerapan berkelanjutan.)

Untuk informasi selengkapnya tentang opsi dalam konfigurasi Lalu Lintas, lihat [the section called “Permintaan rute ke distribusi pementasan”](#).

Setelah Anda selesai dengan konfigurasi Lalu Lintas, pilih Berikutnya.

- e. Tinjau konfigurasi untuk distribusi pementasan, termasuk konfigurasi lalu lintas, lalu pilih Buat distribusi pementasan.

Saat Anda menyelesaikan wizard distribusi pementasan Buat di CloudFront konsol, CloudFront lakukan hal berikut:

- Membuat distribusi pementasan dengan pengaturan yang Anda tentukan (pada langkah 5c)
- Membuat kebijakan penerapan berkelanjutan dengan konfigurasi lalu lintas yang Anda tentukan (pada langkah 5d)
- Melampirkan kebijakan penerapan berkelanjutan ke distribusi utama tempat Anda membuat distribusi pementasan

Ketika konfigurasi distribusi utama, dengan kebijakan penerapan berkelanjutan terlampir, disebarkan ke lokasi tepi, CloudFront mulai mengirimkan bagian lalu lintas yang ditentukan ke distribusi pementasan berdasarkan konfigurasi lalu lintas.

CLI

Untuk membuat distribusi pementasan dan kebijakan penyebaran berkelanjutan dengan AWS CLI, gunakan prosedur berikut.

Untuk membuat distribusi pementasan (CLI)

1. Gunakan grep perintah `aws cloudfront get-distribution` dan bersama-sama untuk mendapatkan ETag nilai distribusi yang ingin Anda gunakan sebagai distribusi utama. Distribusi utama adalah salah satu yang saat ini melayani lalu lintas produksi, dari mana Anda akan membuat distribusi pementasan.

Perintah berikut menunjukkan sebuah contoh. Dalam contoh berikut, ganti *primary_distribution_ID* dengan ID distribusi primer.

```
aws cloudfront get-distribution --id primary_distribution_ID | grep 'ETag'
```

Salin ETag nilainya karena Anda membutuhkannya untuk langkah berikut.

2. Gunakan `aws cloudfront copy-distribution` perintah untuk membuat distribusi pementasan. Contoh perintah berikut menggunakan karakter escape (\) dan jeda baris untuk keterbacaan, tetapi Anda harus menghilangkan ini dari perintah. Dalam contoh perintah berikut:
 - Ganti *primary_distribution_ID* dengan ID distribusi primer.
 - Ganti *primary_distribution_ETag* dengan ETag nilai distribusi primer (yang Anda dapatkan di langkah sebelumnya).
 - (Opsional) Ganti *CLI_example* dengan ID referensi penelepon yang diinginkan.

```
aws cloudfront copy-distribution --primary-distribution-id primary_distribution_ID \  
                                --if-match primary_distribution_ETag \  
                                --staging \  
                                --caller-reference 'CLI_example'
```

Output perintah menunjukkan informasi tentang distribusi pementasan dan konfigurasinya. Salin nama CloudFront domain distribusi pementasan karena Anda membutuhkannya untuk langkah berikut.

Untuk membuat kebijakan penerapan berkelanjutan (CLI dengan file input)

1. Gunakan perintah berikut untuk membuat file bernama `continuous-deployment-policy.yaml` yang berisi semua parameter input untuk `create-continuous-deployment-policy` perintah. Perintah berikut menggunakan karakter escape (`\`) dan jeda baris untuk keterbacaan, tetapi Anda harus menghilangkan ini dari perintah.

```
aws cloudfront create-continuous-deployment-policy --generate-cli-skeleton yaml-  
input \  
  
                                > continuous-deployment-  
policy.yaml
```

2. Buka file dengan nama `continuous-deployment-policy.yaml` yang baru Anda buat. Edit file untuk menentukan pengaturan kebijakan penerapan berkelanjutan yang Anda inginkan, lalu simpan file tersebut. Saat Anda mengedit file:

- Di `StagingDistributionDnsNames` bagian:
 - Ubah nilai `Quantity` ke 1.
 - Untuk `Items`, tempel nama CloudFront domain dari distribusi pementasan (yang Anda simpan dari langkah sebelumnya).
- Di `TrafficConfig` bagian:
 - Pilih `Type`, salah satu `SingleWeight` atau `SingleHeader`.
 - Hapus pengaturan untuk jenis lainnya. Misalnya, jika Anda menginginkan konfigurasi lalu lintas berbasis berat, atur `Type` ke lalu hapus `SingleWeight` pengaturannya `SingleHeaderConfig`.
 - Untuk menggunakan konfigurasi lalu lintas berbasis berat, tetapkan nilai `Weight` ke angka desimal antara .01 (satu persen) dan .15 (lima belas persen).

Untuk informasi selengkapnya tentang opsi di `TrafficConfig`, lihat [the section called “Permintaan rute ke distribusi pementasan”](#) dan [the section called “Sesi lengket untuk konfigurasi berbasis berat”](#).

3. Gunakan perintah berikut untuk membuat kebijakan penerapan berkelanjutan menggunakan parameter input dari `continuous-deployment-policy.yaml` file.

```
aws cloudfront create-continuous-deployment-policy --cli-input-yaml file://continuous-deployment-policy.yaml
```

Salin Id nilai dalam output perintah. Ini adalah ID kebijakan penerapan berkelanjutan, dan Anda memerlukannya dalam langkah berikut.

Untuk melampirkan kebijakan penerapan berkelanjutan ke distribusi utama (CLI dengan file input)

1. Gunakan perintah berikut untuk menyimpan konfigurasi distribusi utama ke file bernama `primary-distribution.yaml`. Ganti *primary_distribution_ID* dengan ID distribusi utama.

```
aws cloudfront get-distribution-config --id primary_distribution_ID --output  
yaml > primary-distribution.yaml
```

2. Buka file dengan nama `primary-distribution.yaml` yang baru saja Anda buat. Edit file akan membuat perubahan berikut:
 - Tempelkan ID kebijakan penerapan berkelanjutan (yang Anda salin dari langkah sebelumnya) ke dalam bidang `ContinuousDeploymentPolicyId`
 - Ubah nama ETag bidang menjadi `IfMatch`, tetapi jangan ubah nilai bidang.

Simpan file setelah selesai.

3. Gunakan perintah berikut untuk memperbarui distribusi utama agar menggunakan kebijakan penerapan berkelanjutan. Ganti *primary_distribution_ID* dengan ID distribusi utama.

```
aws cloudfront update-distribution --id primary_distribution_ID --cli-input-yaml  
file://primary-distribution.yaml
```

Ketika konfigurasi distribusi utama, dengan kebijakan penerapan berkelanjutan terlampir, disebarkan ke lokasi tepi, CloudFront mulai mengirimkan bagian lalu lintas yang ditentukan ke distribusi pementasan berdasarkan konfigurasi lalu lintas.

API

Untuk membuat distribusi pementasan dan kebijakan penerapan berkelanjutan dengan CloudFront API, gunakan operasi API berikut:

- [CopyDistribution](#)
- [CreateContinuousDeploymentPolicy](#)

Untuk informasi selengkapnya tentang bidang yang Anda tentukan dalam panggilan API ini, lihat berikut ini:

- [the section called “Permintaan rute ke distribusi pementasan”](#)
- [the section called “Sesi lengket untuk konfigurasi berbasis berat”](#)
- Dokumentasi referensi API untuk AWS SDK atau klien API lainnya

Setelah Anda membuat distribusi pementasan dan kebijakan penerapan berkelanjutan, gunakan [UpdateDistribution](#) (pada distribusi utama) untuk melampirkan kebijakan penerapan berkelanjutan ke distribusi utama.

Perbarui distribusi pementasan

Prosedur berikut menunjukkan cara memperbarui distribusi pementasan dengan kebijakan penerapan berkelanjutan.

Console

Anda dapat memperbarui konfigurasi tertentu untuk distribusi primer dan pementasan. Untuk informasi selengkapnya, lihat [Perbarui distribusi primer dan pementasan](#).

Untuk memperbarui distribusi pementasan (konsol)

1. Buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Distribusi.
3. Pilih distribusi utama. Ini adalah distribusi yang saat ini melayani lalu lintas produksi, yang darinya Anda membuat distribusi pementasan.
4. Pilih Lihat distribusi pementasan.

- Gunakan konsol untuk memodifikasi konfigurasi distribusi pementasan. Untuk informasi selengkapnya tentang pengaturan yang dapat Anda perbarui, lihat [the section called “Perbarui distribusi primer dan pementasan”](#).

Segera setelah konfigurasi distribusi pementasan diterapkan ke lokasi tepi, itu berlaku untuk lalu lintas masuk yang diarahkan ke distribusi pementasan.

CLI

Untuk memperbarui distribusi pementasan (CLI dengan file input)

- Gunakan perintah berikut untuk menyimpan konfigurasi distribusi pementasan ke file bernama `staging-distribution.yaml`. Ganti `staging_distribution_ID` dengan ID distribusi pementasan.

```
aws cloudfront get-distribution-config --id staging_distribution_ID --output  
yaml > staging-distribution.yaml
```

- Buka file dengan nama `staging-distribution.yaml` yang baru saja Anda buat. Edit file akan membuat perubahan berikut:
 - Ubah konfigurasi distribusi pementasan. Untuk informasi selengkapnya tentang pengaturan yang dapat Anda perbarui, lihat [the section called “Perbarui distribusi primer dan pementasan”](#).
 - Ubah nama ETag bidang menjadi `IfMatch`, tetapi jangan ubah nilai bidang.

Simpan file setelah selesai.

- Gunakan perintah berikut untuk memperbarui konfigurasi distribusi pementasan. Ganti `staging_distribution_ID` dengan ID distribusi pementasan.

```
aws cloudfront update-distribution --id staging_distribution_ID --cli-input-yaml  
file://staging-distribution.yaml
```

Segera setelah konfigurasi distribusi pementasan diterapkan ke lokasi tepi, itu berlaku untuk lalu lintas masuk yang diarahkan ke distribusi pementasan.

API

Untuk memperbarui konfigurasi distribusi pementasan, gunakan [UpdateDistribution](#) (pada distribusi pementasan) untuk memodifikasi konfigurasi distribusi pementasan. Untuk informasi selengkapnya tentang pengaturan yang dapat Anda perbarui, lihat [the section called “Perbarui distribusi primer dan pementasan”](#).

Memperbarui kebijakan penerapan berkelanjutan

Prosedur berikut menunjukkan cara memperbarui kebijakan penerapan berkelanjutan.

Console

Anda dapat memperbarui konfigurasi lalu lintas distribusi dengan memperbarui kebijakan penerapan berkelanjutan.

Untuk memperbarui kebijakan penerapan berkelanjutan (konsol)

1. Buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Distribusi.
3. Pilih distribusi utama. Ini adalah distribusi yang saat ini melayani lalu lintas produksi, yang darinya Anda membuat distribusi pementasan.
4. Di bagian Penerapan berkelanjutan, pilih Edit kebijakan.
5. Ubah konfigurasi lalu lintas dalam kebijakan penerapan berkelanjutan. Setelah selesai, pilih Simpan perubahan.

Ketika konfigurasi distribusi utama dengan kebijakan penerapan berkelanjutan yang diperbarui diterapkan ke lokasi tepi, CloudFront mulai mengirimkan lalu lintas ke distribusi pementasan berdasarkan konfigurasi lalu lintas yang diperbarui.

CLI

Untuk memperbarui kebijakan penerapan berkelanjutan (CLI dengan file input)

1. Gunakan perintah berikut untuk menyimpan konfigurasi kebijakan penerapan berkelanjutan ke file bernama `continuous-deployment-policy.yaml`. Ganti *continuous_deployment_policy_ID* dengan ID kebijakan penerapan berkelanjutan. Perintah berikut menggunakan karakter escape (`\`) dan jeda baris untuk keterbacaan, tetapi Anda harus menghilangkan ini dari perintah.

```
aws cloudfront get-continuous-deployment-policy-config --  
id continuous_deployment_policy_ID \  
continuous-deployment-policy.yaml --output yaml >
```

2. Buka file dengan nama `continuous-deployment-policy.yaml` yang baru saja Anda buat. Edit file akan membuat perubahan berikut:
 - Ubah konfigurasi kebijakan penerapan berkelanjutan sesuai keinginan. Misalnya, Anda dapat mengubah dari menggunakan konfigurasi lalu lintas berbasis header ke berbasis berat, atau Anda dapat mengubah persentase lalu lintas (bobot) untuk konfigurasi berbasis berat. Untuk informasi selengkapnya, lihat [the section called “Permintaan rute ke distribusi pementasan”](#) dan [the section called “Sesi lengket untuk konfigurasi berbasis berat”](#).
 - Ubah nama ETag bidang menjadi `IfMatch`, tetapi jangan ubah nilai bidang.

Simpan file setelah selesai.

3. Gunakan perintah berikut untuk memperbarui kebijakan penerapan berkelanjutan. Ganti *continuous_deployment_policy_ID* dengan ID kebijakan penerapan berkelanjutan. Perintah berikut menggunakan karakter escape (`\`) dan jeda baris untuk keterbacaan, tetapi Anda harus menghilangkan ini dari perintah.

```
aws cloudfront update-continuous-deployment-policy --  
id continuous_deployment_policy_ID \  
continuous-deployment-policy.yaml --cli-input-yaml file://
```

Ketika konfigurasi distribusi utama dengan kebijakan penerapan berkelanjutan yang diperbarui diterapkan ke lokasi tepi, CloudFront mulai mengirimkan lalu lintas ke distribusi pementasan berdasarkan konfigurasi lalu lintas yang diperbarui.

API

Untuk memperbarui kebijakan penerapan berkelanjutan, gunakan [UpdateContinuousDeploymentPolicy](#).

Mempromosikan konfigurasi distribusi pementasan

Prosedur berikut menunjukkan cara mempromosikan konfigurasi distribusi pementasan.

Console

Saat Anda mempromosikan distribusi pementasan, CloudFront salin konfigurasi dari distribusi pementasan ke distribusi utama. CloudFront juga menonaktifkan kebijakan penyebaran berkelanjutan dan merutekan semua lalu lintas ke distribusi utama.

Setelah Anda mempromosikan konfigurasi, Anda dapat menggunakan kembali distribusi pementasan yang sama saat berikutnya Anda ingin menguji perubahan konfigurasi.

Untuk mempromosikan konfigurasi distribusi pementasan (konsol)

1. Buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Distribusi.
3. Pilih distribusi utama. Ini adalah distribusi yang saat ini melayani lalu lintas produksi, yang darinya Anda membuat distribusi pementasan.
4. Di bagian Penerapan berkelanjutan, pilih Promosikan.
5. Ketik **confirm** dan kemudian pilih Promosikan.

CLI

Saat Anda mempromosikan distribusi pementasan, CloudFront salin konfigurasi dari distribusi pementasan ke distribusi utama. CloudFront juga menonaktifkan kebijakan penyebaran berkelanjutan dan merutekan semua lalu lintas ke distribusi utama.

Setelah Anda mempromosikan konfigurasi, Anda dapat menggunakan kembali distribusi pementasan yang sama saat berikutnya Anda ingin menguji perubahan konfigurasi.

Untuk mempromosikan konfigurasi distribusi pementasan (CLI)

- Gunakan `aws cloudfront update-distribution-with-staging-config` perintah untuk mempromosikan konfigurasi distribusi pementasan ke distribusi utama. Contoh perintah berikut menggunakan karakter escape (`\`) dan jeda baris untuk keterbacaan, tetapi Anda harus menghilangkan ini dari perintah. Dalam contoh perintah berikut:
 - Ganti *primary_distribution_ID* dengan ID distribusi primer.

- Ganti *staging_distribution_ID* dengan ID distribusi pementasan.
- Ganti *primary_distribution_ETag* dan *staging_distribution_ETag* dengan ETag nilai-nilai distribusi primer dan pementasan. Pastikan nilai distribusi primer adalah yang pertama, seperti yang ditunjukkan pada contoh.

```
aws cloudfront update-distribution-with-staging-config --
id primary_distribution_ID \
                                     --staging-distribution-
id staging_distribution_ID \
                                     --if-match
'primary_distribution_ETag,staging_distribution_ETag'
```

API

Untuk mempromosikan konfigurasi distribusi pementasan ke distribusi utama, gunakan [UpdateDistributionWithStagingConfig](#).

Pantau distribusi pementasan

Untuk memantau kinerja distribusi pementasan, Anda dapat menggunakan [metrik, log, dan laporan](#) yang sama yang CloudFront menyediakan semua distribusi. Contoh:

- Anda dapat melihat [metrik CloudFront distribusi default](#) (seperti total permintaan dan tingkat kesalahan) di CloudFront konsol, dan Anda dapat [mengaktifkan metrik tambahan \(seperti tingkat klik cache dan tingkat kesalahan berdasarkan kode status\)](#) dengan biaya tambahan. Anda juga dapat membuat alarm berdasarkan metrik ini.
- Anda dapat melihat [log standar](#) dan [log akses real-time](#) untuk mendapatkan informasi rinci tentang permintaan yang diterima oleh distribusi pementasan. Log standar berisi dua bidang berikut yang membantu Anda mengidentifikasi distribusi utama yang awalnya dikirim ke permintaan sebelum CloudFront dirutekan ke distribusi pementasan: `primary-distribution-id` dan `primary-distribution-dns-name`
- Anda dapat melihat dan mengunduh [laporan](#) di CloudFront konsol, misalnya laporan statistik cache.

Pelajari cara kerja penerapan berkelanjutan

Topik berikut menjelaskan cara kerja penerapan CloudFront berkelanjutan.

Topik

- [Permintaan rute ke distribusi pementasan](#)
- [Sesi lengket untuk konfigurasi berbasis berat](#)
- [Perbarui distribusi primer dan pementasan](#)
- [Distribusi primer dan pementasan tidak berbagi cache](#)

Permintaan rute ke distribusi pementasan

Saat Anda menggunakan penerapan CloudFront berkelanjutan, Anda tidak perlu mengubah apa pun tentang permintaan penampil. Pemirsa tidak dapat mengirim permintaan langsung ke distribusi pementasan menggunakan nama DNS, alamat IP, atau CNAME. Sebagai gantinya, pemirsa mengirim permintaan ke distribusi utama (produksi), dan CloudFront merutekan beberapa permintaan tersebut ke distribusi pementasan berdasarkan pengaturan konfigurasi lalu lintas dalam kebijakan penerapan berkelanjutan. Ada dua jenis konfigurasi lalu lintas:

Weight-based

Konfigurasi berbasis bobot merutekan persentase permintaan penampil yang ditentukan ke distribusi pementasan. Saat menggunakan konfigurasi berbasis bobot, Anda juga dapat mengaktifkan kelengketan sesi, yang membantu memastikan permintaan dari CloudFront penampil yang sama sebagai bagian dari satu sesi. Untuk informasi selengkapnya, lihat [the section called “Sesi lengket untuk konfigurasi berbasis berat”](#).

Header-based

Konfigurasi berbasis header merutekan permintaan ke distribusi pementasan saat permintaan penampil berisi header HTTP tertentu (Anda menentukan header dan nilainya). Permintaan yang tidak berisi header dan nilai yang ditentukan dirutekan ke distribusi utama. Konfigurasi ini berguna untuk pengujian lokal, atau ketika Anda memiliki kontrol atas permintaan penampil.

Note

Header yang dirutekan ke distribusi pementasan Anda harus berisi awalan. `aws-cf-cd-`

Sesi lengket untuk konfigurasi berbasis berat

Saat Anda menggunakan konfigurasi berbasis bobot untuk merutekan lalu lintas ke distribusi pementasan, Anda juga dapat mengaktifkan kelengketan sesi, yang membantu memastikan bahwa permintaan dari penampil yang sama CloudFront diperlakukan sebagai satu sesi. Saat Anda mengaktifkan kekakuan sesi, CloudFront tetapkan cookie sehingga semua permintaan dari penampil yang sama dalam satu sesi disajikan oleh satu distribusi, baik yang utama maupun pementasan.

Saat Anda mengaktifkan kelengketan sesi, Anda juga dapat menentukan durasi idle. Jika pemirsa menganggur (tidak mengirim permintaan) untuk jumlah waktu ini, sesi akan kedaluwarsa dan memperlakukan permintaan CloudFront future dari penampil ini sebagai sesi baru. Anda menentukan durasi idle sebagai jumlah detik, dari 300 (lima menit) hingga 3600 (satu jam).

Dalam kasus berikut, CloudFront atur ulang semua sesi (bahkan yang aktif) dan menganggap semua permintaan sebagai sesi baru:

- Anda menonaktifkan atau mengaktifkan kebijakan penerapan berkelanjutan
- Anda menonaktifkan atau mengaktifkan pengaturan lengket sesi

Perbarui distribusi primer dan pementasan

Jika distribusi primer memiliki kebijakan penerapan berkelanjutan yang dilampirkan, perubahan konfigurasi berikut tersedia untuk distribusi primer dan pementasan:

- Semua pengaturan perilaku cache, termasuk perilaku cache default
- Semua pengaturan asal (asal dan grup asal)
- Tanggapan kesalahan kustom (halaman kesalahan)
- Pembatasan geografis
- Objek akar default
- Pengaturan logging
- Deskripsi (komentar)

Anda juga dapat memperbarui sumber daya eksternal yang direferensikan dalam konfigurasi distribusinya—seperti kebijakan cache, kebijakan header respons, fungsi CloudFront , atau fungsi Lambda @Edge.

Distribusi primer dan pementasan tidak berbagi cache

Distribusi primer dan staging tidak berbagi cache. Saat CloudFront mengirim permintaan pertama ke distribusi pementasan, cache-nya kosong. Saat permintaan tiba di distribusi pementasan, itu memulai respons caching (jika dikonfigurasi untuk melakukannya).

Kuota dan pertimbangan lain untuk penyebaran berkelanjutan

CloudFront penyebaran berkelanjutan tunduk pada kuota berikut dan pertimbangan lainnya.

Kuota

- Jumlah maksimum distribusi pementasan per Akun AWS: 20
- Jumlah maksimum kebijakan penerapan berkelanjutan per Akun AWS: 20
- Persentase maksimum lalu lintas yang dapat Anda kirim ke distribusi pementasan dalam konfigurasi berbasis berat: 15%
- Nilai minimum dan maksimum untuk durasi idle lengket sesi: 300-3600 detik

Untuk informasi selengkapnya, lihat [Kuota](#).

Note

Saat menggunakan penerapan berkelanjutan dan distribusi utama Anda disetel dengan OAC untuk akses bucket S3, perbarui kebijakan bucket S3 Anda untuk mengizinkan akses distribusi pementasan. Misalnya kebijakan bucket S3, lihat [the section called “Berikan CloudFront izin untuk mengakses bucket S3”](#).

AWS WAF ACL web

Jika Anda mengaktifkan penerapan berkelanjutan untuk distribusi Anda, pertimbangan berikut berlaku untuk: AWS WAF

- Anda tidak dapat mengaitkan daftar kontrol akses AWS WAF web (ACL) ke distribusi jika ini adalah pertama kalinya ACL dikaitkan dengan distribusi.
- Anda tidak dapat memisahkan ACL AWS WAF web dari distribusi.

Sebelum Anda dapat melakukan tugas-tugas sebelumnya, Anda harus menghapus kebijakan penerapan berkelanjutan untuk distribusi produksi Anda. Ini juga menghapus distribusi pementasan. Untuk informasi selengkapnya, lihat [Gunakan AWS WAF perlindungan](#).

Kasus saat CloudFront mengirim semua permintaan ke distribusi utama

Dalam kasus tertentu, seperti periode pemanfaatan sumber daya yang tinggi, CloudFront dapat mengirim semua permintaan ke distribusi utama terlepas dari apa yang ditentukan dalam kebijakan penerapan berkelanjutan.

CloudFront mengirimkan semua permintaan ke distribusi utama selama jam lalu lintas puncak, terlepas dari apa yang ditentukan dalam kebijakan penerapan berkelanjutan. Lalu lintas puncak mengacu pada lalu lintas pada CloudFront layanan, dan bukan lalu lintas pada distribusi Anda.

HTTP/3

Anda tidak dapat menggunakan penerapan berkelanjutan dengan distribusi yang mendukung HTTP/3.

Gunakan URL kustom dengan menambahkan nama domain alternatif (CNames)

Saat Anda membuat distribusi, CloudFront berikan nama domain untuknya, seperti d111111abcdef8.cloudfront.net. Alih-alih menggunakan nama domain yang disediakan ini, Anda dapat menggunakan nama domain alternatif (juga dikenal sebagai CNAME).

Untuk mempelajari cara menggunakan nama domain Anda sendiri, seperti www.example.com, lihat topik berikut:

Topik

- [Persyaratan untuk menggunakan nama domain alternatif](#)
- [Pembatasan penggunaan nama domain alternatif](#)
- [Tambahkan nama domain alternatif](#)
- [Memindahkan nama domain alternatif](#)
- [Hapus nama domain alternatif](#)
- [Gunakan wildcard dalam nama domain alternatif](#)

Persyaratan untuk menggunakan nama domain alternatif

Saat Anda menambahkan nama domain alternatif, seperti `www.example.com`, ke CloudFront distribusi, berikut ini adalah persyaratan:

Nama domain alternatif harus huruf kecil

Semua nama domain alternatif (CNames) harus huruf kecil.

Nama domain alternatif harus dicakup oleh sertifikat TLS yang valid

Untuk menambahkan nama domain alternatif (CNAME) ke CloudFront distribusi, Anda harus melampirkan sertifikat TLS tepercaya dan valid yang mencakup nama domain alternatif. Ini memastikan bahwa hanya orang yang memiliki akses ke sertifikat domain Anda yang dapat mengasosiasikan dengan CNAME CloudFront yang terkait dengan domain Anda.

Sertifikat tepercaya adalah sertifikat yang dikeluarkan oleh AWS Certificate Manager (ACM) atau oleh otoritas sertifikat lain yang valid (CA). Anda dapat menggunakan sertifikat yang ditandatangani sendiri untuk memvalidasi CNAME yang ada, tetapi tidak untuk CNAME baru. CloudFront mendukung otoritas sertifikat yang sama dengan Mozilla. Untuk daftar saat ini, lihat [Daftar Sertifikat CA yang Disertakan Mozilla](#). Untuk informasi tentang sertifikat perantara saat menggunakan CA pihak ketiga, lihat [Sertifikat Menengah](#).

Untuk memverifikasi nama domain alternatif dengan menggunakan sertifikat yang Anda lampirkan, termasuk nama domain alternatif yang menyertakan wildcard, CloudFront periksa nama alternatif subjek (SAN) pada sertifikat. Nama domain alternatif yang Anda tambahkan harus dicakup oleh SAN.

Note

Hanya satu sertifikat yang dapat dilampirkan ke CloudFront distribusi pada satu waktu.

Anda membuktikan bahwa Anda berwenang menambahkan nama domain alternatif tertentu ke distribusi Anda dengan melakukan salah satu hal berikut:

- Melampirkan sertifikat yang menyertakan nama domain alternatif, seperti `product-name.example.com`.
- Melampirkan sertifikat yang mencakup * wildcard di awal nama domain, untuk menutupi beberapa subdomain dengan satu sertifikat. Saat menentukan wildcard, Anda dapat menambahkan beberapa subdomain sebagai nama domain alternatif di CloudFront.

Contoh berikut menggambarkan cara menggunakan wildcard dalam nama domain dalam karya sertifikat untuk memungkinkan Anda menambahkan nama domain alternatif khusus di CloudFront.

- Anda ingin menambahkan `marketing.example.com` sebagai nama domain alternatif. Anda mencantumkan dalam sertifikat Anda nama domain berikut: `*.example.com`. Ketika Anda melampirkan sertifikat ini CloudFront, Anda dapat menambahkan nama domain alternatif untuk distribusi Anda yang menggantikan wildcard pada tingkat itu, termasuk `marketing.example.com`. Anda juga dapat, misalnya, menambahkan nama domain alternatif berikut:
 - `product.example.com`
 - `api.example.com`

Namun, Anda tidak dapat menambahkan nama domain alternatif yang lebih tinggi atau lebih rendah dari wildcard. Misalnya, Anda tidak dapat menambahkan nama domain alternatif `example.com` atau `marketing.product.example.com`.

- Anda ingin menambahkan `example.com` sebagai nama domain alternatif. Untuk melakukan ini, Anda harus mencantumkan nama domain `example.com` itu sendiri pada sertifikat yang Anda lampirkan ke distribusi Anda.
- Anda ingin menambahkan `marketing.product.example.com` sebagai nama domain alternatif. Untuk melakukan ini, Anda dapat mencantumkan `*.product.example.com` pada sertifikat, atau Anda dapat mencantumkan `marketing.product.example.com` itu sendiri pada sertifikat.

Izin untuk mengubah konfigurasi DNS

Saat menambahkan nama domain alternatif, Anda harus membuat catatan CNAME untuk merutekan kueri DNS untuk nama domain alternatif ke distribusi Anda. CloudFront Untuk melakukannya, Anda harus memiliki izin untuk membuat catatan CNAME dengan penyedia layanan DNS untuk nama domain alternatif yang Anda gunakan. Biasanya, ini berarti Anda memiliki domain, tetapi Anda mungkin mengembangkan aplikasi untuk pemilik domain.

Nama domain alternatif dan HTTPS

Jika Anda ingin penampil menggunakan HTTPS dengan nama domain alternatif, Anda harus menyelesaikan beberapa konfigurasi tambahan. Untuk informasi selengkapnya, lihat [Gunakan nama domain alternatif dan HTTPS](#).

Pembatasan penggunaan nama domain alternatif

Perhatikan pembatasan berikut saat menggunakan nama domain alternatif:

Jumlah maksimum nama domain alternatif

Untuk jumlah maksimum nama domain alternatif saat ini yang dapat Anda tambahkan ke distribusi, atau untuk meminta kuota yang lebih tinggi (sebelumnya dikenal sebagai batas), lihat [Kuota umum di distribusi](#).

Menggandakan dan menindih nama domain alternatif

Anda tidak dapat menambahkan nama domain alternatif ke CloudFront distribusi jika nama domain alternatif yang sama sudah ada di CloudFront distribusi lain, bahkan jika Anda Akun AWS memiliki distribusi lainnya.

Namun, Anda dapat menambahkan nama domain alternatif wildcard, seperti *.example.com, yang menyertakan (yang tumpang tindih dengan) nama domain alternatif non-wildcard, seperti www.example.com. Jika Anda memiliki nama domain alternatif yang tumpang tindih dalam dua distribusi, CloudFront kirimkan permintaan ke distribusi dengan pencocokan nama yang lebih spesifik, terlepas dari distribusi yang ditunjuk oleh catatan DNS. Misalnya, marketing.domain.com lebih spesifik daripada *.domain.com.

Jika Anda memiliki entri DNS wildcard yang mengarah ke CloudFront distribusi dan Anda menerima kesalahan DNS yang tidak dikonfigurasi dengan benar saat mencoba menambahkan CNAME baru dengan nama yang lebih spesifik, lihat [CloudFront mengembalikan kesalahan catatan DNS yang tidak dikonfigurasi dengan benar ketika saya mencoba menambahkan CNAME baru](#)

Fronting domain

CloudFront memiliki perlindungan terhadap fronting domain yang terjadi di berbagai Akun AWS tempat. Ini adalah skenario ketika klien non-standar membuat TLS/SSL koneksi ke nama domain dalam satu Akun AWS, dan kemudian membuat permintaan HTTPS untuk nama domain yang tidak terkait di yang lain. Akun AWS

Misalnya, koneksi TLS mungkin terhubung ke www.example.com, dan kemudian mengeluarkan permintaan untuk www.example.org.

Untuk menentukan apakah permintaan sedang di-fronted domain, CloudFront lakukan pemeriksaan berikut:

- Ekstensi SNI sama dengan header permintaan Host HTTP
- Sertifikat milik Akun AWS sama dengan distribusi untuk permintaan
- Permintaan HTTP Host dicakup oleh sertifikat yang disajikan selama jabat tangan TLS

Jika tidak satu pun dari kondisi ini terpenuhi, CloudFront tentukan permintaannya adalah fronting domain. CloudFront akan menolak permintaan dengan respons kesalahan 421 HTTP.

 Note

Jika klien tidak menyediakan ekstensi SNI dan mendapatkan sertifikat default *.cloudfront.net sebagai gantinya, akan menerima permintaan yang masuk. CloudFront

Bagaimana CloudFront mengidentifikasi distribusi untuk permintaan

CloudFront mengidentifikasi distribusi untuk permintaan HTTP berdasarkan Host header. CloudFront tidak bergantung pada alamat CloudFront IP yang Anda sambungkan atau jabat tangan SNI apa yang disediakan selama jabat tangan TLS.

Ketika CloudFront menerima permintaan, itu akan menggunakan nilai Host header untuk mencocokkan permintaan dengan distribusi tertentu.

Misalnya, Anda memiliki dua distribusi dan Anda telah memperbarui konfigurasi DNS Anda sehingga nama domain alternatif merutekan ke titik akhir berikut:

- primary.example.com menunjuk ke d111111primary.cloudfront.net
- secondary.example.com menunjuk ke d222222secondary.cloudfront.net

Jika Anda membuat permintaan untuk `https://primary.example.com` tetapi menentukan Host header sebagai `secondary.example.com`, seperti, permintaan akan merutekan ke distribusi sekunder sebagai `curl https://primary.example.com -H "Host: secondary.example.com"` gantinya.

Menambahkan nama domain alternatif pada simpul atas (apex zona) untuk domain

Saat menambahkan nama domain alternatif ke distribusi, Anda biasanya membuat catatan CNAME dalam konfigurasi DNS untuk merutekan kueri DNS untuk nama domain ke distribusi Anda. CloudFront Namun, Anda tidak dapat membuat catatan CNAME untuk node atas dari ruang nama DNS, yang juga dikenal sebagai apex zona; protokol DNS tidak mengizinkannya. Misalnya, jika Anda mendaftarkan nama DNS `example.com`, zone apex-nya adalah `example.com`. Anda tidak dapat membuat catatan CNAME untuk `example.com`, tetapi Anda dapat membuat catatan CNAME untuk `www.example.com`, `newproduct.example.com`, dan sebagainya.

Jika Anda menggunakan Route 53 sebagai layanan DNS, Anda dapat membuat kumpulan catatan sumber daya alias, yang memiliki keunggulan berikut dibandingkan catatan CNAME:

- Anda dapat membuat rekaman sumber daya alias yang ditetapkan untuk nama domain di simpul atas (contoh.com).
- Anda dapat membuat catatan HTTPS untuk nama domain alternatif untuk memungkinkan negosiasi protokol sebagai bagian dari pencarian DNS jika klien mendukungnya. Untuk informasi selengkapnya, lihat [Create alias resource record set](#).
- Anda tidak membayar kueri Route 53 saat menggunakan kumpulan catatan sumber daya alias.

 Note

Jika Anda mengaktifkan IPv6, Anda harus membuat dua set catatan sumber daya alias: satu untuk merutekan IPv4 lalu lintas (catatan A) dan satu untuk merutekan lalu lintas (catatan AAAA). Untuk informasi lebih lanjut, lihat [Aktifkan IPv6 \(permintaan penampil\)](#) dalam [Semua referensi pengaturan distribusi](#) topik.

Untuk informasi selengkapnya, lihat [Merutekan lalu lintas ke distribusi CloudFront web Amazon dengan menggunakan nama domain Anda](#) di Panduan Pengembang Amazon Route 53.

Jika Anda tidak menggunakan Route 53 untuk DNS Anda, Anda dapat meminta alamat IP statis Anycast untuk merutekan domain apex seperti example.com ke CloudFront. Untuk informasi selengkapnya, lihat [Minta IP statis Anycast untuk digunakan untuk daftar yang diizinkan](#).

Tambahkan nama domain alternatif

Daftar tugas berikut menjelaskan cara menggunakan CloudFront konsol untuk menambahkan nama domain alternatif ke distribusi Anda sehingga Anda dapat menggunakan nama domain Anda sendiri di tautan Anda, bukan nama CloudFront domain. Untuk informasi tentang memperbarui distribusi menggunakan CloudFront API, lihat [Konfigurasi distribusi](#).

 Note

Jika Anda ingin penampil menggunakan HTTPS dengan nama domain alternatif Anda, lihat [Gunakan nama domain alternatif dan HTTPS](#).

Sebelum memulai: Pastikan Anda melakukan hal berikut sebelum memperbarui distribusi untuk menambahkan nama domain alternatif:

- Daftarkan nama domain dengan Route 53 atau registrar domain lain.
- Dapatkan sertifikat TLS dari otoritas sertifikat resmi (CA) yang mencakup nama domain. Tambahkan sertifikat ke distribusi Anda untuk memvalidasi bahwa Anda berwenang untuk menggunakan domain. Untuk informasi selengkapnya, lihat [Persyaratan untuk menggunakan nama domain alternatif](#).

Tambahkan nama domain alternatif

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Pilih ID untuk distribusi yang ingin Anda perbarui.
3. Pada tab Umum, pilih Tambahkan domain.
4. Masukkan hingga lima domain untuk dilayani.
5. Pilih Berikutnya.
6. Untuk sertifikat TLS, jika tidak CloudFront dapat menemukan sertifikat AWS Certificate Manager (ACM) yang ada untuk domain Anda di dalam us-east-1 Wilayah AWS, Anda Akun AWS dapat memilih untuk secara otomatis membuat sertifikat atau membuatnya secara manual di ACM.
7. Ketika sertifikat Anda disediakan, Anda harus memperbarui catatan DNS Anda dengan penyedia DNS Anda untuk membuktikan kepemilikan domain. Entri yang perlu Anda buat ke catatan DNS Anda disediakan untuk Anda di CloudFront konsol.
8. Setelah memperbarui catatan DNS, pilih Validasi sertifikat.
9. Ketika sertifikat divalidasi, pilih Berikutnya.
10. Tinjau perubahan Anda dan pilih Tambahkan domain.
11. Di Umum untuk distribusi, konfirmasi bahwa Status Distribusi telah berubah menjadi Diterapkan. Jika Anda mencoba menggunakan nama domain alternatif sebelum pembaruan ke distribusi Anda telah diterapkan, tautan yang Anda buat pada langkah-langkah berikut mungkin tidak akan berfungsi.
12. Konfigurasi layanan DNS untuk nama domain alternatif (seperti `www.example.com`) untuk merutekan lalu lintas ke nama CloudFront domain untuk distribusi Anda (seperti `d111111abcdef8.cloudfront.net`). Metode yang Anda gunakan tergantung pada apakah Anda

menggunakan Route 53 sebagai penyedia layanan DNS untuk domain atau penyedia lain. Untuk informasi selengkapnya, lihat [Tambahkan domain ke distribusi CloudFront standar Anda](#).

Route 53

Buat set catatan sumber daya alias. Dengan seperangkat catatan sumber daya alias, Anda tidak membayar kueri Route 53. Anda juga dapat membuat catatan sumber daya alias yang ditetapkan untuk nama domain root (example.com), yang DNS tidak mengizinkan untuk CNames. Untuk petunjuk cara membuat kumpulan catatan sumber daya alias, lihat [Merutekan lalu lintas ke distribusi CloudFront web Amazon menggunakan nama domain Anda](#) di Panduan Pengembang Amazon Route 53.

Secara opsional, Anda dapat membuat catatan HTTPS untuk nama domain alternatif untuk memungkinkan negosiasi protokol sebagai bagian dari pencarian DNS jika klien mendukungnya.

Untuk membuat catatan sumber daya alias yang ditetapkan dengan catatan HTTPS (opsional)

1. Aktifkan HTTP/2 atau HTTP/3 dalam pengaturan CloudFront distribusi Anda. Untuk informasi selengkapnya, lihat [Versi HTTP yang didukung](#) dan [Perbarui distribusi](#).
2. Di konsol Route 53, buat kumpulan catatan sumber daya alias. Ikuti [lalu lintas perutean ke distribusi CloudFront web Amazon dengan menggunakan prosedur nama domain Anda](#).
3. Saat Anda membuat kumpulan catatan sumber daya alias, buat catatan alias dengan tipe rekaman HTTPS.

Penyedia layanan DNS lainnya

Gunakan metode yang disediakan oleh penyedia layanan DNS Anda untuk menambahkan catatan CNAME untuk domain Anda. Catatan CNAME baru ini akan mengarahkan kueri DNS dari nama domain alternatif Anda (misalnya, www.example.com) ke nama domain untuk distribusi Anda (misalnya, CloudFront d111111abcdef8.cloudfront.net). Untuk informasi lebih lanjut, lihat dokumentasi yang disediakan oleh penyedia layanan DNS Anda.

 Important

Jika Anda sudah memiliki catatan CNAME yang ada untuk nama domain alternatif Anda, perbarui catatan tersebut atau ganti dengan yang baru yang menunjuk ke nama CloudFront domain untuk distribusi Anda.

13. Menggunakan `dig` atau alat DNS serupa, konfirmasi bahwa konfigurasi DNS yang Anda buat pada langkah sebelumnya menunjuk ke nama domain untuk distribusi Anda.

Contoh berikut menunjukkan `dig` permintaan di domain `www.example.com`, serta bagian terkait dari respons.

```
PROMPT> dig www.example.com

; <<> DiG 9.3.3rc2 <<> www.example.com
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15917
;; flags: qr rd ra; QUERY: 1, ANSWER: 9, AUTHORITY: 2, ADDITIONAL: 0

;; QUESTION SECTION:
;www.example.com.      IN      A

;; ANSWER SECTION:
www.example.com. 10800 IN CNAME d111111abcdef8.cloudfront.net.
...
```

Bagian jawaban menunjukkan catatan CNAME yang merutekan kueri untuk `www.example.com` ke nama domain distribusi `d111111abcdef8.cloudfront.net`. CloudFront Jika nama di sisi kanan CNAME adalah nama domain untuk CloudFront distribusi Anda, catatan CNAME dikonfigurasi dengan benar. Jika ada nilai lain, misalnya, nama domain untuk bucket Amazon S3 Anda, maka catatan CNAME tidak dikonfigurasi dengan benar. Dalam hal ini, kembali ke langkah 7 dan perbaiki catatan CNAME untuk menunjuk ke nama domain untuk distribusi Anda.

14. Uji nama domain alternatif dengan mengunjungi URL dengan nama domain Anda, bukan nama CloudFront domain untuk distribusi Anda.
15. Dalam aplikasi Anda, ubah URL objek Anda untuk menggunakan nama domain alternatif Anda, bukan nama domain CloudFront distribusi Anda.

Memindahkan nama domain alternatif

Jika Anda mencoba menambahkan nama domain alternatif ke distribusi standar atau penyewa distribusi, dan nama domain alternatif sudah dikaitkan dengan sumber daya yang berbeda, Anda akan mendapatkan pesan kesalahan.

Misalnya, Anda akan mendapatkan pesan `CNAMEAlreadyExists` kesalahan (Satu atau lebih dari CNames yang Anda berikan sudah dikaitkan dengan sumber daya yang berbeda) ketika Anda mencoba menambahkan `www.example.com` ke distribusi standar atau penyewa distribusi, tetapi nama domain alternatif itu sudah dikaitkan dengan sumber daya yang berbeda.

Dalam hal ini, Anda mungkin ingin memindahkan nama domain alternatif yang ada dari satu sumber daya ke sumber lainnya. Ini adalah distribusi sumber dan distribusi target. Anda dapat memindahkan nama domain alternatif antara penyewa distribusi and/or distribusi standar.

Untuk memindahkan nama domain alternatif, lihat topik berikut:

Topik

- [Mengatur target distribusi standar atau penyewa distribusi](#)
- [Temukan sumber distribusi standar atau penyewa distribusi](#)
- [Pindahkan nama domain alternatif](#)

Mengatur target distribusi standar atau penyewa distribusi

Sebelum Anda dapat memindahkan nama domain alternatif, Anda harus mengatur sumber daya target. Ini adalah target distribusi standar atau penyewa distribusi tempat Anda memindahkan nama domain alternatif.

Standard distribution

Untuk mengatur distribusi standar target

1. Minta sertifikat TLS. Sertifikat ini mencakup nama domain alternatif sebagai Subjek atau Subjek Alternatif Domain (SAN), atau wildcard (*) yang mencakup nama domain alternatif yang Anda pindahkan. Jika Anda tidak memilikinya, Anda dapat meminta satu dari AWS Certificate Manager (ACM) atau dari otoritas sertifikat (CA) lain dan mengimpornya ke ACM.

 Note

Anda harus meminta atau mengimpor sertifikat di Wilayah AS Timur (Virginia N.) (us-east-1).

Untuk informasi selengkapnya, lihat [Meminta sertifikat publik menggunakan konsol](#) dan [Mengimpor sertifikat](#) AWS Certificate Manager di Panduan AWS Certificate Manager Pengguna.

2. Jika Anda belum membuat distribusi standar target, buat sekarang. Sebagai bagian dari pembuatan distribusi standar, kaitkan sertifikat dengan distribusi standar ini. Untuk informasi selengkapnya, lihat [Buat distribusi](#).

Jika Anda sudah memiliki distribusi standar target, kaitkan sertifikat dengan distribusi standar. Untuk informasi selengkapnya, lihat [Perbarui distribusi](#).

3. Jika Anda memindahkan nama domain alternatif dalam hal yang sama Akun AWS, lewati langkah ini.

Untuk memindahkan nama domain alternatif dari satu Akun AWS ke yang lain, Anda harus membuat catatan TXT dalam konfigurasi DNS Anda. Langkah verifikasi ini membantu mencegah transfer domain yang tidak sah. CloudFront menggunakan catatan TXT ini untuk memvalidasi kepemilikan Anda atas nama domain alternatif.

Dalam konfigurasi DNS Anda, buat catatan DNS TXT yang mengaitkan nama domain alternatif dengan distribusi standar target. Format rekaman TXT dapat bervariasi, tergantung pada jenis domain.

- Untuk subdomain, tentukan garis bawah (_) di depan nama domain alternatif. Berikut ini menunjukkan contoh catatan TXT.

```
_www.example.com TXT d111111abcdef8.cloudfront.net
```

- Untuk apex (atau domain root), tentukan garis bawah dan periode (._) di depan nama domain. Berikut ini menunjukkan contoh catatan TXT.

```
_.example.com TXT d111111abcdef8.cloudfront.net
```

Distribution tenant

Untuk mengatur penyewa distribusi target

1. Minta sertifikat TLS. Sertifikat ini mencakup nama domain alternatif sebagai Subjek atau Subjek Alternatif Domain (SAN), atau wildcard (*) yang mencakup nama domain alternatif yang Anda pindahkan. Jika Anda tidak memilikinya, Anda dapat meminta satu dari AWS Certificate Manager (ACM) atau dari otoritas sertifikat (CA) lain dan mengimpornya ke ACM.

Note

Anda harus meminta atau mengimpor sertifikat di Wilayah AS Timur (Virginia N.) (us-east-1).

Untuk informasi selengkapnya, lihat [Meminta sertifikat publik menggunakan konsol](#) dan [Mengimpor sertifikat](#) AWS Certificate Manager di Panduan AWS Certificate Manager Pengguna.

2. Jika Anda belum membuat penyewa distribusi target, buat sekarang. Sebagai bagian dari pembuatan penyewa distribusi, kaitkan sertifikat dengan penyewa distribusi. Untuk informasi selengkapnya, lihat [Buat distribusi](#).

Jika Anda sudah memiliki penyewa distribusi target, kaitkan sertifikat dengan penyewa distribusi. Untuk informasi selengkapnya, lihat [Tambahkan domain dan sertifikat \(penyewa distribusi\)](#).

3. Jika Anda memindahkan nama domain alternatif dalam hal yang sama Akun AWS, lewati langkah ini.

Untuk memindahkan nama domain alternatif dari satu Akun AWS ke yang lain, Anda harus membuat catatan TXT dalam konfigurasi DNS Anda. Langkah verifikasi ini membantu mencegah transfer domain yang tidak sah, dan CloudFront menggunakan catatan TXT ini untuk memvalidasi kepemilikan Anda atas nama domain alternatif.

Dalam konfigurasi DNS Anda, buat catatan DNS TXT yang mengaitkan nama domain alternatif dengan penyewa distribusi target. Format rekaman TXT dapat bervariasi, tergantung pada jenis domain.

- Untuk subdomain, tentukan garis bawah (_) di depan nama domain alternatif. Berikut ini menunjukkan contoh catatan TXT.

```
_www.example.com TXT d111111abcdef8.cloudfront.net
```

- Untuk apex (atau domain root), tentukan garis bawah dan periode (_.) di depan nama domain. Berikut ini menunjukkan contoh catatan TXT.

```
_.example.com TXT d111111abcdef8.cloudfront.net
```

Selanjutnya, lihat topik berikut untuk menemukan distribusi standar sumber atau penyewa distribusi yang sudah dikaitkan dengan nama domain alternatif.

Temukan sumber distribusi standar atau penyewa distribusi

Sebelum Anda dapat memindahkan nama domain alternatif dari satu distribusi (standar atau penyewa) ke yang lain, cari distribusi sumber. Ini adalah sumber daya yang sudah dikaitkan dengan nama domain alternatif. Ketika Anda mengetahui Akun AWS ID sumber daya distribusi sumber dan target, Anda dapat menentukan cara memindahkan nama domain alternatif.

Catatan

- Kami menyarankan Anda menggunakan operasi [ListDomainConflicts](#) API, karena mendukung distribusi standar dan penyewa distribusi.
- Operasi [ListConflictingAliases](#) API hanya mendukung distribusi standar.

Ikuti contoh-contoh ini untuk menemukan distribusi sumber (standar atau penyewa).

list-domain-conflicts

Tip

- Untuk distribusi standar, Anda harus memiliki `cloudfront:GetDistribution` dan `cloudfront:ListDomainConflicts` izin.

- Untuk penyewa distribusi, Anda harus memiliki `cloudfront:GetDistributionTenant` dan `cloudfront:ListDomainConflicts` izin.

Untuk menggunakan **daftar-domain-konflik** untuk menemukan distribusi standar sumber atau penyewa distribusi

1. Gunakan perintah `list-domain-conflicts` seperti pada contoh berikut.
 - a. Ganti `www.example.com` dengan nama domain.
 - b. Untuk `domain-control-validation-resource`, tentukan ID distribusi standar target atau penyewa distribusi [yang Anda atur sebelumnya](#). Anda harus memiliki penyewa distribusi atau distribusi standar yang terkait dengan sertifikat yang mencakup domain yang ditentukan.
 - c. Jalankan perintah ini menggunakan kredensial yang Akun AWS sama dengan distribusi standar target atau penyewa distribusi.

Permintaan

Contoh ini menentukan penyewa distribusi.

```
aws cloudfront list-domain-conflicts \
--domain www.example.com \
--domain-control-validation-resource
"DistributionTenantId=dt_2x9GhoK0TZRsohWzv1b9It8JABC"
```

Respons

Untuk setiap nama domain dalam output perintah, Anda dapat melihat yang berikut:

- Jenis sumber daya yang dikaitkan dengan domain
- ID sumber daya
- Akun AWS ID yang memiliki sumber daya

ID sumber daya dan ID akun sebagian disembunyikan. Ini memungkinkan Anda untuk mengidentifikasi distribusi standar atau penyewa distribusi milik akun Anda, dan membantu melindungi informasi yang tidak Anda miliki.

```
{
  "DomainConflicts": [
    {
      "Domain": "www.example.com",
      "ResourceType": "distribution-tenant",
      "ResourceId": "*****ohWzv1b9It8JABC",
      "AccountId": "*****112233"
    }
  ]
}
```

Respons mencantumkan semua nama domain yang bertentangan atau tumpang tindih dengan nama yang Anda tentukan.

Contoh

- Jika Anda menentukan *tenant1.example.com*, responsnya mencakup tenant1.example.com dan nama domain alternatif wildcard yang tumpang tindih (*.example.com jika ada).
 - Jika Anda menentukan **.tenant1.example.com*, responsnya mencakup *.tenant1.example.com dan nama domain alternatif apa pun yang dicakup oleh wildcard tersebut (misalnya, test.tenant1.example.com, dev.tenant1.example.com, dan sebagainya).
2. Dalam tanggapan, temukan distribusi standar sumber atau penyewa distribusi untuk nama domain alternatif yang Anda pindahkan, dan catat Akun AWS ID.
 3. Bandingkan ID akun dari penyewa distribusi atau distribusi standar sumber dengan ID akun tempat Anda membuat target distribusi standar atau penyewa distribusi pada langkah [sebelumnya](#). Anda kemudian dapat menentukan apakah sumber dan targetnya sama Akun AWS. Ini membantu Anda menentukan cara memindahkan nama domain alternatif.

Untuk informasi selengkapnya, lihat [list-domain-conflicts](#) perintah di AWS Command Line Interface Referensi.

list-conflicting-aliases (standard distributions only)

Tip

Anda harus memiliki `cloudfront:GetDistribution` dan `cloudfront:ListConflictingAliases` izin pada distribusi standar target.

Untuk menggunakan **list-conflicting-aliases** untuk menemukan distribusi standar sumber

1. Gunakan perintah `list-conflicting-aliases` seperti pada contoh berikut.
 - a. Ganti `www.example.com` dengan nama domain alternatif, dan `EDFDVBD6EXAMPLE` dengan ID distribusi standar target [yang Anda atur sebelumnya](#).
 - b. Jalankan perintah ini menggunakan kredensial yang Akun AWS sama dengan distribusi standar target.

Permintaan

Contoh ini menentukan distribusi standar.

```
aws cloudfront list-conflicting-aliases \  
--alias www.example.com \  
--distribution-id EDFDVBD6EXAMPLE
```

Respons

Untuk setiap nama domain alternatif dalam output perintah, Anda dapat melihat ID distribusi standar yang terkait dengannya, dan Akun AWS ID yang memiliki distribusi standar. Distribusi standar dan ID akun sebagian disembunyikan, yang memungkinkan Anda mengidentifikasi distribusi standar dan akun yang Anda miliki, dan membantu melindungi informasi yang tidak Anda miliki.

```
{  
  "ConflictingAliasesList": {  
    "MaxItems": 100,  
    "Quantity": 1,  
    "Items": [  

```

```
{
  "Alias": "www.example.com",
  "DistributionId": "*****EXAMPLE",
  "AccountId": "*****112233"
}
```

Respons mencantumkan nama domain alternatif yang bertentangan atau tumpang tindih dengan nama domain yang Anda tentukan.

Contoh

- Jika Anda menentukan `www.example.com`, responsnya mencakup `www.example.com` dan nama domain alternatif wildcard yang tumpang tindih (`*.example.com`) jika ada.
 - Jika Anda menentukan `*.example.com`, responsnya mencakup `*.example.com` dan nama domain alternatif apa pun yang dicakup oleh wildcard tersebut (misalnya, `www.example.com`, `test.example.com`, `dev.example.com`, dan sebagainya).
2. Temukan distribusi standar untuk nama domain alternatif yang Anda pindahkan, dan catat Akun AWS ID. Bandingkan ID akun ini dengan ID akun tempat Anda membuat distribusi standar target pada [langkah sebelumnya](#). Anda kemudian dapat menentukan apakah kedua distribusi standar ini sama Akun AWS dan bagaimana cara memindahkan nama domain alternatif.

Untuk informasi selengkapnya, lihat [list-conflicting-aliases](#) perintah di AWS Command Line Interface Referensi.

Selanjutnya, lihat topik berikut untuk memindahkan nama domain alternatif.

Pindahkan nama domain alternatif

Tergantung pada situasi Anda, pilih dari cara-cara berikut untuk memindahkan nama domain alternatif:

Distribusi sumber dan target (standar atau penyewa) sama Akun AWS

Gunakan `update-domain-association` perintah di AWS Command Line Interface (AWS CLI) untuk memindahkan nama domain alternatif.

Perintah ini berfungsi untuk semua gerakan akun yang sama, termasuk ketika nama domain alternatif adalah domain puncak (juga disebut domain root, seperti `example.com`).

Distribusi sumber dan target (standar atau penyewa) berbeda Akun AWS

Jika Anda memiliki akses ke distribusi standar sumber atau penyewa distribusi, nama domain alternatif bukan domain puncak, dan Anda belum menggunakan wildcard yang tumpang tindih dengan nama domain alternatif tersebut, gunakan wildcard untuk memindahkan nama domain alternatif. Untuk informasi selengkapnya, lihat [the section called “Gunakan wildcard untuk memindahkan nama domain alternatif”](#).

Jika Anda tidak memiliki akses ke Akun AWS yang memiliki distribusi standar sumber atau penyewa distribusi, Anda dapat mencoba menggunakan `update-domain-association` perintah untuk memindahkan nama domain alternatif. Penyewa distribusi atau distribusi standar sumber harus dinonaktifkan sebelum Anda dapat memindahkan nama domain alternatif. Untuk bantuan tambahan, lihat [the section called “Kontak AWS Dukungan untuk memindahkan nama domain alternatif”](#).

Note

Anda dapat menggunakan `associate-alias` perintah, tetapi perintah ini hanya mendukung distribusi standar. Lihat [AssociateAlias](#) di Referensi CloudFront API Amazon.

`update-domain-association` (standard distributions and distribution tenants)

Untuk menggunakan **`update-domain-association`** untuk memindahkan nama domain alternatif

1. Gunakan perintah `update-domain-association` seperti pada contoh berikut.
 - a. Ganti *example.com* dengan nama domain alternatif, dan tentukan ID dari distribusi standar target atau penyewa distribusi.
 - b. Jalankan perintah ini menggunakan kredensial yang Akun AWS sama dengan distribusi standar target atau penyewa distribusi.

 Perhatikan batasan berikut

- Selain `cloudfront:UpdateDomainAssociation` izin, Anda harus memiliki `cloudfront:UpdateDistribution` izin untuk memperbarui distribusi standar. Untuk memperbarui penyewa distribusi, Anda harus memiliki `cloudfront:UpdateDistributionTenant` izin.
- Jika distribusi sumber dan target (standar atau penyewa) berbeda Akun AWS, sumber harus dinonaktifkan sebelum Anda dapat memindahkan domain.
- Distribusi target harus diatur seperti yang dijelaskan dalam [the section called “Mengatur target distribusi standar atau penyewa distribusi”](#).

Permintaan

```
aws cloudfront update-domain-association \  
  --domain "www.example.com" \  
  --target-resource DistributionTenantId=dt_9Fd3xTZq7H12KABC \  
  --if-match E3UN6WX5ABC123
```

Respons

```
{  
  "ETag": "E7Xp1Y3N9DABC",  
  "Domain": "www.example.com",  
  "ResourceId": "dt_9Fd3xTZq7H12KABC"  
}
```

Perintah ini menghapus nama domain alternatif dari distribusi standar sumber atau penyewa distribusi dan menambahkannya ke distribusi standar target atau penyewa distribusi.

2. Setelah distribusi target sepenuhnya diterapkan, perbarui konfigurasi DNS Anda untuk mengarahkan nama domain Anda ke titik akhir CloudFront perutean. Misalnya, catatan DNS Anda akan mengarahkan nama domain alternatif Anda (`www.example.com`) ke nama domain yang CloudFront disediakan `d11111abcdef8.cloudfront.net`. Jika targetnya adalah penyewa distribusi, tentukan titik akhir grup koneksi. Untuk informasi selengkapnya, lihat [Arahkan domain ke CloudFront](#).

associate-alias (standard distributions only)

Untuk menggunakan **alias asosiasi** untuk memindahkan nama domain alternatif

1. Gunakan perintah `associate-alias` seperti pada contoh berikut.
 - a. Ganti `www.example.com` dengan nama domain alternatif, dan `EDFDVBD6EXAMPLE` dengan ID distribusi standar target.
 - b. Jalankan perintah ini menggunakan kredensial yang Akun AWS sama dengan distribusi standar target.

 Perhatikan batasan berikut

- Anda harus memiliki `cloudfront:AssociateAlias` dan `cloudfront:UpdateDistribution` izin pada distribusi standar target.
- Jika distribusi standar sumber dan target sama Akun AWS, Anda harus memiliki `cloudfront:UpdateDistribution` izin pada distribusi standar sumber.
- Jika distribusi standar sumber dan distribusi standar target berbeda Akun AWS, Anda harus menonaktifkan distribusi standar sumber terlebih dahulu.
- Distribusi standar target harus diatur seperti yang dijelaskan dalam [the section called “Mengatur target distribusi standar atau penyewa distribusi”](#).

Permintaan

```
aws cloudfront associate-alias \  
--alias www.example.com \  
--target-distribution-id EDFDVBD6EXAMPLE
```

Perintah ini menghapus nama domain alternatif dari distribusi standar sumber dan memindahkannya ke distribusi standar target.

2. Setelah distribusi standar target sepenuhnya diterapkan, perbarui konfigurasi DNS Anda untuk mengarahkan data DNS nama domain alternatif ke nama domain distribusi distribusi standar target. Misalnya, catatan DNS Anda akan mengarahkan nama domain alternatif Anda (`www.example.com`) ke nama domain yang CloudFront disediakan `d111111abcdef8.cloudfront.net`.

Untuk informasi selengkapnya, lihat [associate-alias](#) perintah di AWS CLI Command Reference.

Gunakan wildcard untuk memindahkan nama domain alternatif

Jika distribusi sumber berbeda Akun AWS dari distribusi target, dan distribusi sumber diaktifkan, Anda dapat menggunakan wildcard untuk memindahkan nama domain alternatif.

 Note

Anda tidak dapat menggunakan wildcard untuk memindahkan domain apex (seperti example.com). Untuk memindahkan domain apex ketika distribusi sumber dan target berbeda Akun AWS, hubungi. Dukungan Untuk informasi selengkapnya, lihat [the section called “Kontak AWS Dukungan untuk memindahkan nama domain alternatif”](#).

Untuk menggunakan wildcard untuk memindahkan nama domain alternatif

 Note

Proses ini melibatkan beberapa pembaruan untuk distribusi Anda. Tunggu hingga setiap distribusi menerapkan perubahan terbaru sepenuhnya sebelum melanjutkan ke langkah berikutnya.

1. Perbarui distribusi target untuk menambahkan nama domain alternatif wildcard yang mencakup nama domain alternatif yang Anda pindahkan. Misalnya, jika nama domain alternatif yang Anda pindahkan adalah www.example.com, tambahkan nama domain alternatif*.example.com ke distribusi target. Untuk melakukan ini, SSL/TLS sertifikat pada distribusi target harus menyertakan nama domain wildcard. Untuk informasi selengkapnya, lihat [the section called “Perbarui distribusi”](#).
2. Perbarui pengaturan DNS untuk nama domain alternatif untuk menunjuk ke nama domain dari distribusi target. Misalnya, jika nama domain alternatif yang Anda pindahkan adalah www.example.com, perbarui data DNS untuk www.example.com untuk merutekan lalu lintas ke nama domain distribusi target (misalnya d111111abcdef8.cloudfront.net).

 Note

Bahkan setelah Anda memperbarui pengaturan DNS, nama domain alternatif masih disajikan oleh distribusi sumber karena di situlah nama domain alternatif saat ini dikonfigurasi.

3. Perbarui distribusi sumber untuk menghapus nama domain alternatif. Untuk informasi selengkapnya, lihat [Perbarui distribusi](#).
4. Perbarui distribusi target untuk menambahkan nama domain alternatif. Untuk informasi selengkapnya, lihat [Perbarui distribusi](#).
5. Gunakan dig (atau alat kueri DNS serupa) untuk memvalidasi bahwa catatan DNS untuk nama domain alternatif menyelesaikan nama domain dari distribusi target.
6. (Opsional) Perbarui distribusi target untuk menghapus nama domain alternatif wildcard.

Kontak AWS Dukungan untuk memindahkan nama domain alternatif

Jika distribusi sumber dan target berbeda Akun AWS, dan Anda tidak memiliki akses ke distribusi sumber Akun AWS atau tidak dapat menonaktifkan distribusi sumber, Anda dapat menghubungi Dukungan untuk memindahkan nama domain alternatif.

Untuk menghubungi Dukungan untuk memindahkan nama domain alternatif

1. Siapkan distribusi target, termasuk catatan DNS TXT yang mengarah ke distribusi target. Untuk informasi selengkapnya, lihat [Mengatur target distribusi standar atau penyewa distribusi](#).
2. [Hubungi Dukungan](#) untuk meminta mereka memverifikasi bahwa Anda memiliki domain, dan memindahkan domain ke CloudFront distribusi baru untuk Anda.
3. Setelah distribusi target sepenuhnya digunakan, perbarui konfigurasi DNS Anda untuk mengarahkan catatan DNS nama domain alternatif ke nama domain distribusi distribusi target.

Hapus nama domain alternatif

Jika Anda ingin menghentikan perutean lalu lintas untuk domain atau subdomain ke CloudFront distribusi, ikuti langkah-langkah di bagian ini untuk memperbarui konfigurasi DNS dan distribusi. CloudFront

Penting bagi Anda untuk menghapus nama domain alternatif dari distribusi serta memperbarui konfigurasi DNS Anda. Ini membantu mencegah masalah nanti jika Anda ingin mengaitkan nama domain dengan CloudFront distribusi lain. Jika nama domain alternatif sudah terkait dengan satu distribusi, itu tidak dapat diatur dengan yang lain.

 Note

Jika Anda ingin menghapus nama domain alternatif dari distribusi ini sehingga Anda dapat menambahkannya ke yang lain, ikuti langkah-langkah di [Memindahkan nama domain alternatif](#). Jika Anda mengikuti langkah-langkah di sini sebagai gantinya (untuk menghapus domain) dan kemudian menambahkan domain ke distribusi lain, akan ada periode waktu di mana domain tidak akan menautkan ke distribusi baru karena CloudFront menyebarkan ke pembaruan ke lokasi tepi.

Untuk menghapus nama domain alternatif dari distribusi

1. Untuk memulai, arahkan lalu lintas internet untuk domain Anda ke sumber daya lain yang bukan CloudFront distribusi Anda, seperti penyeimbang beban Elastic Load Balancing. Atau Anda dapat menghapus catatan DNS yang merutekan lalu lintas ke CloudFront

Lakukan salah satu langkah berikut, tergantung pada layanan DNS untuk domain Anda:

- Jika Anda menggunakan Route 53, memperbarui atau menghapus catatan alias atau catatan CNAME. Untuk informasi lebih lanjut, lihat [Mengedit rekaman](#) atau [Menghapus catatan](#).
 - Jika Anda menggunakan penyedia layanan DNS lain, gunakan metode yang disediakan oleh penyedia layanan DNS untuk memperbarui atau menghapus catatan CNAME yang mengarahkan lalu lintas ke CloudFront Untuk informasi lebih lanjut, lihat dokumentasi yang disediakan oleh penyedia layanan DNS Anda.
2. Setelah Anda memperbarui catatan DNS domain Anda, tunggu hingga perubahan telah menyebar dan penutup DNS sedang mengirimkan lalu lintas ke sumber daya baru. Anda dapat memeriksa untuk melihat kapan ini selesai dengan membuat beberapa tautan uji yang menggunakan domain Anda di URL.
 3. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>, dan perbarui CloudFront distribusi Anda untuk menghapus nama domain dengan melakukan hal berikut:
 - a. Pilih ID untuk distribusi yang ingin Anda perbarui.

- b. Di Umum pilih, pilih Edit.
- c. Di Nama Domain Alternatif (CNAME), hapus nama domain alternatif (atau nama domain) yang tidak ingin Anda gunakan lagi untuk distribusi Anda.
- d. Pilih Ya, Edit.

Gunakan wildcard dalam nama domain alternatif

Saat menambahkan nama domain alternatif, Anda dapat menggunakan * wildcard di awal nama domain dan bukan menambahkan subdomain secara individu. Misalnya, dengan nama domain alternatif*.example.com, Anda dapat menggunakan nama domain apa pun yang diakhiri dengan example.com di URL Anda, seperti www.example.com, product-name.example.com, marketing.product-name.example.com, dan sebagainya. Jalur ke objek adalah sama terlepas dari nama domain, misalnya:

- www.contoh. com/images/image.jpg
- product-name.example. com/images/image.jpg
- marketing.product-name.example. com/images/image.jpg

Ikuti persyaratan ini untuk nama domain alternatif yang mencakup wildcard:

- Nama domain alternatif harus dimulai dengan tanda bintang dan titik (*.).
- Anda tidak dapat menggunakan wildcard untuk mengganti bagian dari nama subdomain, seperti ini: *domain.example.com.
- Anda tidak dapat mengganti subdomain di tengah nama domain, seperti ini: subdomain.*.example.com.
- Semua nama domain alternatif, termasuk nama domain alternatif yang menggunakan wildcard, harus dicakup oleh nama alternatif subjek (SAN) pada sertifikat.

Nama domain alternatif wildcard, seperti *.example.com, dapat menyertakan nama domain alternatif lain yang sedang digunakan, seperti example.com.

Gunakan WebSockets dengan CloudFront distribusi

Amazon CloudFront mendukung penggunaan WebSocket, TCP-based protokol yang berguna ketika Anda membutuhkan koneksi dua arah yang berumur panjang antara klien dan server. Koneksi yang

persisten sering kali merupakan persyaratan dengan aplikasi waktu nyata. Skenario yang mungkin Anda gunakan WebSockets termasuk platform obrolan sosial, ruang kerja kolaborasi online, game multi-pemain, dan layanan yang menyediakan umpan data real-time seperti platform perdagangan keuangan. Data melalui WebSocket koneksi dapat mengalir di kedua arah untuk komunikasi dupleks penuh.

WebSocket fungsionalitas secara otomatis diaktifkan untuk bekerja dengan distribusi apa pun. Untuk menggunakan WebSockets, konfigurasi salah satu dari berikut ini dalam perilaku cache yang dilampirkan ke distribusi Anda:

- Teruskan semua header permintaan penampil ke asal Anda. Anda dapat menggunakan [kebijakan permintaan asal AllViewer terkelola](#).
- Teruskan header Sec-WebSocket-Key dan Sec-WebSocket-Version minta secara khusus dalam kebijakan permintaan asal Anda.

Bagaimana WebSocket protokol bekerja

WebSocket Protokol ini adalah TCP-based protokol independen yang memungkinkan Anda menghindari beberapa overhead—dan berpotensi meningkatkan latensi—HTTP.

Untuk membuat WebSocket koneksi, klien mengirimkan permintaan HTTP reguler yang menggunakan semantik upgrade HTTP untuk mengubah protokol. Server kemudian dapat menyelesaikan proses. WebSocket Koneksi tetap terbuka dan klien atau server dapat mengirim bingkai data satu sama lain tanpa harus membuat koneksi baru setiap kali.

Secara default, WebSocket protokol menggunakan port 80 untuk WebSocket koneksi reguler dan port 443 untuk WebSocket koneksi melalui TLS. Opsi yang Anda pilih untuk Anda CloudFront [Kebijakan protokol penampil](#) dan [Protokol \(hanya asal kustom\)](#) berlaku untuk WebSocket koneksi dan juga lalu lintas HTTP.

WebSocket persyaratan

WebSocket permintaan harus mematuhi [RFC 6455](#) dalam format standar berikut.

Example Contoh permintaan klien

```
GET /chat HTTP/1.1
Host: server.example.com
Upgrade: websocket
```

```
Connection: Upgrade
Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==
Origin: https://example.com
Sec-WebSocket-Protocol: chat, superchat
Sec-WebSocket-Version: 13
```

Example Contoh respon server

```
HTTP/1.1 101 Switching Protocols
Upgrade: websocket
Connection: Upgrade
Sec-WebSocket-Accept: s3pPLMBiTxaQ9kYGzzhZRbK+x0o=
Sec-WebSocket-Protocol: chat
```

Jika WebSocket koneksi terputus oleh klien atau server, atau oleh gangguan jaringan, aplikasi klien diharapkan untuk memulai kembali koneksi dengan server.

WebSocket Header yang direkomendasikan

Untuk menghindari masalah terkait kompresi yang tidak terduga saat menggunakan WebSockets, sebaiknya sertakan header berikut dalam kebijakan permintaan [asal](#):

- Sec-WebSocket-Key
- Sec-WebSocket-Version
- Sec-WebSocket-Protocol
- Sec-WebSocket-Accept
- Sec-WebSocket-Extensions

Note

Saat ini, CloudFront hanya mendukung WebSocket koneksi melalui HTTP/1.1 protokol.

Minta IP statis Anycast untuk digunakan untuk daftar yang diizinkan

Anda dapat meminta IP statis Anycast CloudFront untuk digunakan dengan distribusi Anda. Daftar IP statis anycast hanya dapat berisi alamat IP IPv4 atau alamat IP IPv4 dan IPv6. Alamat IP ini didedikasikan untuk Anda Akun AWS dan tersebar di seluruh wilayah geografis.

Anda dapat meminta 21 alamat IP statis Anycast untuk diizinkan masuk ke penyedia jaringan sehingga Anda dapat membebaskan biaya data untuk pemirsa yang mengakses aplikasi Anda. Atau, Anda dapat menggunakan IP statis ini dalam firewall keamanan keluar untuk mengontrol pertukaran lalu lintas dengan aplikasi yang disetujui. Daftar IP statis anycast dapat digunakan dengan satu atau lebih distribusi.

Jika Anda ingin mengaktifkan perutean domain apex (seperti `example.com`) langsung ke CloudFront distribusi Anda, Anda dapat meminta 3 alamat IP statis Anycast untuk kasus penggunaan ini. Kemudian, tambahkan catatan A di DNS Anda untuk mengarahkan domain apex ke CloudFront

IP statis anycast bekerja dengan [Server Name Indication \(SNI\)](#). Untuk informasi selengkapnya, lihat [Gunakan SNI untuk melayani permintaan HTTPS \(berfungsi untuk sebagian besar klien\)](#).

Prasyarat

Untuk menggunakan daftar IP statis Anycast dengan CloudFront distribusi Anda, Anda harus memilih Gunakan semua lokasi tepi untuk kelas harga untuk distribusi. Untuk informasi selengkapnya tentang harga, silakan lihat [harga CloudFront](#).

Minta daftar IP statis Anycast

Minta daftar IP statis Anycast untuk digunakan dengan CloudFront distribusi Anda.

Untuk meminta daftar IP statis Anycast

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi kiri, pilih IP Statis.
3. Untuk Permintaan, pilih tautan untuk menghubungi teknik CloudFront dukungan.
4. Berikan informasi beban kerja Anda (minta byte per detik dan permintaan per detik).
5. CloudFront dukungan teknik meninjau permintaan Anda. Proses peninjauan bisa memakan waktu hingga dua hari.

Setelah permintaan Anda disetujui, Anda dapat membuat daftar IP statis Anycast dan mengaitkannya dengan satu atau beberapa distribusi.

Buat daftar IP statis Anycast

Sebelum Anda mulai, minta daftar IP statis Anycast seperti yang dijelaskan di bagian sebelumnya.

Untuk membuat daftar IP statis Anycast

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi kiri, pilih IP Statis.
3. Pilih Buat daftar IP Anycast.
4. Untuk Nama, masukkan nama.
5. Untuk kasus penggunaan IP Statis, pilih kasus penggunaan yang sesuai.
6. Untuk jenis alamat IP, tentukan salah satu opsi berikut:
 - IPv4 — Alokasikan daftar hanya alamat IPv4
 - Dualstack — Alokasikan daftar alamat IPv4 dan IPv6
7. Tinjau persyaratan dan harga layanan, dan pilih Kirim.

Setelah daftar IP statis Anda dibuat, Anda dapat melihat alamat IP yang dialokasikan pada halaman detail daftar IP statis Anda. Anda juga dapat mengaitkan distribusi dengan daftar IP statis.

Kaitkan daftar IP statis Anycast dengan distribusi yang ada

Sebelum Anda mulai, minta dan buat daftar IP statis Anycast seperti yang dijelaskan di bagian sebelumnya.

Verifikasi bahwa pengaturan distribusi berikut kompatibel dengan daftar IP statis Anycast Anda:

- [Kelas harga](#) memiliki pengaturan Gunakan semua lokasi tepi (kinerja terbaik).
- Jika [IPv6](#) diaktifkan, Anda dapat mengaitkan daftar IP statis Anycast dualstack. Daftar IP statis Anycast yang hanya memiliki alamat IPv4 tidak dapat dikaitkan dengan distribusi dengan IPv6 diaktifkan.

Untuk mengaitkan daftar IP statis Anycast dengan distribusi yang ada

- Lakukan salah satu tindakan berikut:
 - Kaitkan daftar IP statis dari halaman detail daftar IP statis:
 1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.

2. Pilih IP Statis di panel navigasi kiri.
 3. Pilih nama daftar IP statis Anda.
 4. Pilih Distribusi asosiasi.
 5. Pilih satu atau beberapa distribusi dan pilih Distribusi asosiasi.
- Kaitkan daftar IP statis dari halaman detail distribusi:
 1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
 2. Pilih Distribusi di panel navigasi kiri.
 3. Pilih nama distribusi Anda.
 4. Pada tab Umum, di bawah Pengaturan, pilih Edit.
 5. Untuk daftar IP Anycast, pilih daftar IP statis Anycast untuk digunakan dengan distribusi ini.
 6. Pilih Simpan perubahan.

Kaitkan daftar IP statis Anycast dengan distribusi baru

Sebelum Anda mulai, minta dan buat daftar IP statis Anycast seperti yang dijelaskan di bagian sebelumnya.

Untuk mengaitkan daftar IP statis Anycast dengan distribusi baru

- Buat distribusi baru. Untuk informasi selengkapnya, lihat [Buat CloudFront distribusi di konsol](#). Untuk Pengaturan, Anda harus membuat pilihan berikut untuk menggunakan daftar IP statis Anycast Anda:
 - Untuk daftar IP Anycast, pilih daftar IP statis Anycast Anda dari daftar dropdown.
 - Untuk kelas Harga, pilih Gunakan semua lokasi tepi (performa terbaik).
 - Catatan: Jika IP statis Anycast Anda hanya menggunakan IPv4 dan bukan dualstack, untuk IPv6, pilih Off.

Selesai membuat distribusi Anda. Anda dapat memilih pengaturan dan konfigurasi lain yang tidak diperlukan untuk daftar IP statis Anycast berdasarkan kebutuhan Anda.

Untuk informasi selengkapnya tentang kuota yang terkait dengan daftar IP statis Anycast, lihat [CloudFront titik akhir dan kuota Amazon](#) di Referensi Umum AWS

Kaitkan daftar IP statis Anycast dengan grup koneksi

Sebelum Anda mulai, minta dan buat daftar IP statis Anycast seperti yang dijelaskan di bagian sebelumnya.

Untuk mengaitkan daftar IP statis Anycast dengan grup koneksi baru

1. Pastikan Anda telah mengaktifkan grup koneksi di bawah Pengaturan.
2. Buat grup koneksi. Untuk informasi selengkapnya, lihat [Membuat grup koneksi kustom](#).
3. Untuk Pengaturan, Anda harus membuat pilihan berikut untuk menggunakan daftar IP statis Anycast Anda.
 - Untuk daftar IP Anycast, pilih daftar IP statis Anycast Anda dari daftar dropdown.
4. Selesai membuat grup koneksi Anda.

Note

Jika IP statis Anycast Anda hanya menggunakan IPv4 dan bukan dualstack, untuk IPv6, pilih Off.

Untuk informasi selengkapnya tentang kuota yang terkait dengan daftar IP statis Anycast, lihat [CloudFront titik akhir dan kuota Amazon](#) di Referensi Umum Amazon Web

Perbarui daftar IP statis Anycast

Setelah Anda membuat alamat IP statis Anycast dan menghubungkannya dengan distribusi, Anda dapat mengubah jenis alamat IP dari daftar IP statis Anycast Anda..

Untuk memperbarui daftar IP statis Anycast

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi kiri, pilih IP Statis.
3. Pilih nama daftar IP statis Anda.
4. Pilih Edit.
5. Untuk jenis alamat IP, tentukan salah satu opsi berikut:

- IPv4 — Alokasikan daftar hanya alamat IPv4
- Dualstack — Alokasikan daftar alamat IPv4 dan IPv6

 Note

Anda tidak dapat memilih IPv4 jika distribusi terkait Anda telah mengaktifkan IPv6. Untuk melakukannya, nonaktifkan IPv6 sebelum Anda dapat memperbarui jenis alamat IP untuk IP statis Anycast Anda. Untuk informasi selengkapnya, lihat [Aktifkan IPv6 untuk distribusi CloudFront](#).

6. Pilih Kirim untuk menyimpan perubahan Anda dan memperbarui daftar IP statis Anycast.

Bawa IP Anda sendiri untuk CloudFront menggunakan IPAM

Tutorial ini menunjukkan cara menggunakan IPAM untuk mengelola CIDR BYOIP Anda untuk CloudFront daftar IP Statis Anycast.

Topik

- [Apa itu BYOIP untuk IP Statis Anycast?](#)
- [Mengapa menggunakan fitur ini?](#)
- [Prasyarat](#)
- [Langkah 1: Minta daftar IP statis Anycast](#)
- [Langkah 2: Buat daftar IP statis Anycast](#)
- [Langkah 3: Buat CloudFront distribusi](#)
- [Langkah 4: Kaitkan dengan CloudFront sumber daya](#)
- [Langkah 5: Bersiaplah untuk Migrasi](#)
- [Langkah 6: Iklankan CIDR secara global](#)

Apa itu BYOIP untuk IP Statis Anycast?

CloudFront mendukung membawa alamat IPv4 dan IPv6 Anda sendiri melalui BYOIP IPAM untuk layanan global. Melalui antarmuka terpadu IPAM, pelanggan dapat membuat kumpulan alamat IP khusus menggunakan alamat IP mereka sendiri (BYOIP) dan menetapkan ke CloudFront

distribusi sambil memanfaatkan jaringan pengiriman konten AWS di seluruh dunia untuk mengirimkan aplikasi dan konten mereka. Alamat IP Anda diiklankan dari beberapa lokasi CloudFront tepi secara bersamaan menggunakan perutean anycast.

Mengapa menggunakan fitur ini?

Kontrol akses jaringan dalam daftar izinkan untuk:

- Allow-list Alamat IP dengan penyedia jaringan untuk membebaskan biaya data bagi pemirsa yang disetujui
- Konfigurasi firewall keamanan keluar untuk membatasi lalu lintas ke aplikasi yang disetujui saja

Sederhanakan operasi dan migrasi

- Rute domain apex (example.com) langsung ke CloudFront dengan menambahkan catatan A yang mengarah ke IP statis Anda
- Migrasi dari CDN lain tanpa memperbarui infrastruktur IP atau konfigurasi firewall
- Pertahankan daftar izin IP yang ada dengan mitra dan klien
- Bagikan satu daftar IP statis Anycast di beberapa distribusi CloudFront

Branding yang konsisten

- Pertahankan ruang alamat IP yang ada untuk branding yang konsisten saat pindah ke AWS

Prasyarat

Untuk menggunakan daftar IP statis Anycast dengan CloudFront distribusi Anda, Anda harus memilih Gunakan semua lokasi tepi untuk kelas harga untuk distribusi. Untuk informasi selengkapnya tentang harga, silakan lihat [harga CloudFront](#). Untuk Bring Your Own IP (BYOIP), Anda juga perlu menonaktifkan IPv6 untuk distribusi atau grup koneksi saat menggunakan BYOIP. IPv4-only Untuk dual-stack BYOIP, kaitkan Daftar IP statis Anycast dual-stack dan aktifkan IPv6 untuk grup distribusi atau koneksi.

Selesaikan langkah-langkah ini sebelum memulai:

- Penyiapan IPAM: Lihat [Integrasikan IPAM dengan akun](#) dan [Buat IPAM](#).
- Verifikasi domain: [Verifikasi kontrol domain](#).

- Buat kumpulan tingkat atas: Ikuti langkah 1 hingga 2 di [Bawa IPv4 CIDR Anda sendiri ke IPAM and/or Bawa IPv6 CIDR Anda sendiri ke IPAM](#).
- Buat kolam IPAM dengan lokal sebagai global untuk digunakan. CloudFront Untuk informasi selengkapnya, lihat [Membawa IP Anda sendiri untuk CloudFront menggunakan IPAM](#).

 Note

Membutuhkan and/or tiga/24/48 blok CIDR IPv4.

Langkah 1: Minta daftar IP statis Anycast

Minta daftar IP statis Anycast untuk digunakan dengan CloudFront distribusi Anda.

Untuk meminta daftar IP statis Anycast

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi kiri, pilih IP Statis.
3. Untuk Permintaan, pilih tautan untuk menghubungi teknik CloudFront dukungan.
4. Berikan informasi beban kerja Anda (minta byte per detik dan permintaan per detik).
5. CloudFront dukungan teknik meninjau permintaan Anda. Proses peninjauan bisa memakan waktu hingga dua hari.
6. Setelah permintaan Anda disetujui, Anda dapat membuat daftar IP statis Anycast dan mengaitkannya dengan satu atau beberapa distribusi.

Langkah 2: Buat daftar IP statis Anycast

Sebelum Anda mulai, minta daftar IP statis Anycast seperti yang dijelaskan di bagian sebelumnya.

Untuk membuat daftar IP statis Anycast

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi kiri, pilih IP Statis.
3. Pilih Buat daftar IP Anycast.

4. Untuk Nama, masukkan nama.
5. Untuk kasus penggunaan IP Statis, pilih BYOIP sebagai kasus penggunaan Anda.
6. Untuk jenis alamat IP, pilih IPv4 atau Dualstack.
7. Untuk kolam IPAM, pilih kolam IPAM yang Anda sediakan melalui grup IPAM dan CIDR dari mereka.

Langkah-langkah berikut berbeda dari proses BYOIP regional standar dan menetapkan pola untuk layanan global:

AWS CLI

Menginstal atau memperbarui ke versi terbaru AWS CLI. Untuk informasi selengkapnya, silakan lihat [Panduan Pengguna AWS Command Line Interface](#).

1. Ambil kolam IPAM tempat blok CIDR Anda disediakan. `IpamPoolArn` Untuk informasi lebih lanjut, lihat [Membawa IPv4 CIDR publik Anda sendiri ke IPAM hanya menggunakan AWS CLI atau Bawa IPv6 CIDR publik Anda sendiri](#) ke IPAM hanya menggunakan CLI. AWS
2. Buat daftar IP Anycast dengan blok CIDR dan konfigurasi IPAM Anda:

Untuk IPv4:

```
aws cloudfront create-anycast-ip-list \
  --name byoip-aip-1 \
  --ip-count 3 \
  --region us-east-1 \
  --ip-address-type ipv4 \
  --ipam-cidr-configs
  '[{"Cidr":"1.1.1.0/24","IpamPoolArn":"arn:aws:ec2::123456789012:ipam-pool/ipam-
pool-005d58a8aa8147abc"},
{"Cidr":"2.2.2.0/24","IpamPoolArn":"arn:aws:ec2::123456789012:ipam-pool/ipam-
pool-005d58a8aa8147abc"},
{"Cidr":"3.3.3.0/24","IpamPoolArn":"arn:aws:ec2::123456789012:ipam-pool/ipam-
pool-005d58a8aa8147abc"}]'
```

Untuk IPv6:

```
aws cloudfront create-anycast-ip-list \
  --name byoip-aip-dualstack \
  --ip-count 3 \
```

```
--region us-east-1 \  
--ip-address-type dualstack \  
--ipam-cidr-configs  
'[{"Cidr":"1.1.1.0/24","IpamPoolArn":"arn:aws:ec2::123456789012:ipam-pool/ipam-pool-005d58a8aa8147abc"},  
{"Cidr":"2.2.2.0/24","IpamPoolArn":"arn:aws:ec2::123456789012:ipam-pool/ipam-pool-005d58a8aa8147abc"},  
{"Cidr":"3.3.3.0/24","IpamPoolArn":"arn:aws:ec2::123456789012:ipam-pool/ipam-pool-005d58a8aa8147abc"},  
{"Cidr":"2600:9000:a100::/48","IpamPoolArn":"arn:aws:ec2::123456789012:ipam-pool/ipam-pool-0a3b7c9e2f41d6789"},  
{"Cidr":"2600:9000:a200::/48","IpamPoolArn":"arn:aws:ec2::123456789012:ipam-pool/ipam-pool-0a3b7c9e2f41d6789"},  
{"Cidr":"2600:9000:a300::/48","IpamPoolArn":"arn:aws:ec2::123456789012:ipam-pool/ipam-pool-0a3b7c9e2f41d6789"}]'
```

Note

Anda tidak dapat memilih alamat IP tertentu dari pool. CloudFront akan melakukan ini secara otomatis.

Langkah 3: Buat CloudFront distribusi

Untuk CloudFront, Anda dapat mengikuti petunjuk untuk [membuat distribusi standar atau menggunakan distribusi multi-penyewa](#).

Langkah 4: Kaitkan dengan CloudFront sumber daya

- [Kaitkan daftar IP statis Anycast dengan distribusi yang ada](#)
- [Kaitkan daftar IP statis Anycast dengan distribusi baru](#)
- [Kaitkan daftar IP statis Anycast dengan grup koneksi](#)

Langkah 5: Bersiaplah untuk Migrasi

Untuk informasi selengkapnya, lihat [Langkah 4: Mempersiapkan migrasi](#) di Panduan Pengguna Amazon VPC.

Langkah 6: Iklankan CIDR secara global

Untuk informasi selengkapnya, lihat [Langkah 5: Mengiklankan CIDR secara global](#) di Panduan Pengguna Amazon VPC.

Menggunakan gRPC dengan distribusi CloudFront

Amazon CloudFront mendukung gRPC, kerangka kerja panggilan prosedur jarak jauh sumber terbuka (RPC) bawaan. HTTP/2 gRPC menawarkan streaming dua arah dan protokol biner yang menyangga muatan, sehingga cocok untuk aplikasi yang memerlukan komunikasi latensi rendah.

CloudFront menerima permintaan gRPC Anda dan memproxinya langsung ke asal Anda. Anda dapat menggunakan CloudFront untuk mem-proxy empat jenis layanan gRPC:

- RPC Unary
- Server streaming RPC
- Klien streaming RPC
- Bi-directional streaming RPC

Bagaimana gRPC bekerja di CloudFront

Untuk mengonfigurasi gRPC CloudFront, tetapkan asal yang menyediakan layanan gRPC sebagai asal distribusi Anda. Anda dapat menggunakan asal yang menyediakan layanan non-gRPC dan gRPC. CloudFront menentukan apakah permintaan yang masuk adalah permintaan gRPC atau HTTP/HTTPS permintaan berdasarkan Content-Type header. Jika Content-Type header permintaan memiliki nilai `application/grpc`, permintaan tersebut dianggap sebagai permintaan gRPC dan CloudFront akan mem-proxy permintaan tersebut ke asal Anda.

Note

Untuk mengaktifkan distribusi untuk menangani permintaan gRPC, sertakan HTTP/2 sebagai salah satu versi HTTP yang didukung, dan izinkan metode HTTP, termasuk. POST Titik akhir asal gRPC Anda harus dikonfigurasi untuk mendukung HTTPS, karena CloudFront hanya mendukung koneksi gRPC aman (). HTTPS-based gRPC hanya mendukung HTTPS ujung ke ujung. Jika Anda menggunakan custom origin, verifikasi bahwa setelah [Protokol](#) Anda mendukung HTTPS.

Untuk mengaktifkan dukungan gRPC untuk distribusi Anda, selesaikan langkah-langkah berikut:

1. Perbarui perilaku cache distribusi Anda untuk mengizinkan metode HTTP, termasuk POST metode.
2. Setelah Anda memilih POST metode, pilih kotak centang gRPC yang muncul.
3. Tentukan HTTP/2 sebagai salah satu versi HTTP yang didukung.

Untuk informasi selengkapnya, lihat topik berikut:

- [Izinkan permintaan gRPC berakhir HTTP/2](#)
- [GrpcConfig](#) di Referensi CloudFront API Amazon

Karena gRPC hanya digunakan untuk lalu lintas API yang tidak dapat di-cache, konfigurasi cache Anda tidak akan memengaruhi permintaan gRPC. Anda dapat menggunakan kebijakan permintaan asal untuk menambahkan header kustom ke permintaan gRPC yang dikirim ke asal gRPC Anda. Anda dapat menggunakannya AWS WAF CloudFront untuk mengelola akses ke distribusi gRPC Anda, mengontrol bot, dan melindungi aplikasi gRPC Anda dari eksploitasi web. CloudFront [gRPC mendukung CloudFront Fungsi](#).

Selain status HTTPS, Anda akan menerima status grpc bersama dengan respons gRPC Anda. Untuk daftar kemungkinan nilai grpc-status, lihat [Kode status dan penggunaannya di gRPC](#).

Catatan

gRPC tidak mendukung fitur-fitur berikut: CloudFront

- [Tanggapan kesalahan kustom](#)
- [Origin failover](#) tidak didukung dengan gRPC, karena gRPC menggunakan metode. POST CloudFront gagal ke asal sekunder hanya jika metode HTTP dari permintaan penampil adalah GET, HEAD, atau OPTIONS.
- CloudFront proxy gRPC meminta langsung ke asal dan melewati Regional Edge Cache (REC). [Karena gRPC melewati REC, gRPC tidak mendukung Lambda @Edge atau Origin Shield](#).
- gRPC tidak mendukung aturan inspeksi badan AWS WAF permintaan. Jika Anda mengaktifkan aturan ini di ACL web untuk distribusi, permintaan apa pun yang menggunakan gRPC akan mengabaikan aturan inspeksi badan permintaan. Semua AWS

WAF aturan lain akan tetap berlaku. Lihat informasi yang lebih lengkap di [Aktifkan AWS WAF untuk distribusi](#).

Bekerja dengan sumber daya bersama di CloudFront

Amazon CloudFront terintegrasi dengan AWS Resource Access Manager (AWS RAM) untuk mengaktifkan berbagi sumber daya. AWS RAM memungkinkan Anda untuk berbagi beberapa CloudFront sumber daya dengan orang lain Akun AWS atau melalui AWS Organizations. Dengan AWS RAM, Anda dapat berbagi sumber daya yang Anda miliki dengan membuat berbagi sumber daya. Pembagian sumber daya menentukan sumber daya yang akan dibagikan, dan konsumen yang akan dibagikan. Konsumen dapat mencakup:

- Khusus Akun AWS di dalam atau di luar organisasinya di AWS Organizations
- Unit organisasi di dalam organisasinya di AWS Organizations
- Seluruh organisasinya di AWS Organizations

Untuk informasi selengkapnya AWS RAM, lihat [Panduan AWS RAM Pengguna](#).

Topik ini menjelaskan cara berbagi sumber daya yang Anda miliki, dan cara menggunakan sumber daya yang dibagikan dengan Anda.

Daftar Isi

- [Prasyarat untuk berbagi sumber daya](#)
- [Berbagi asal VPC](#)
- [Menggunakan asal VPC bersama](#)
- [Mengidentifikasi asal VPC bersama](#)
- [Membatalkan berbagi asal VPC bersama](#)
- [Tanggung jawab dan izin untuk asal VPC bersama](#)
- [Tagihan dan pengukuran](#)
- [Kuota sumber daya bersama](#)

Prasyarat untuk berbagi sumber daya

- Anda harus memiliki kebijakan `AWSRAMDefaultPermissionCloudFront` terkelola untuk memberikan akses hanya-baca pada pembagian sumber daya. Untuk informasi selengkapnya, lihat [AWSRAMDefaultPermissionCloudFront](#).

- Untuk berbagi asal VPC, Anda harus memilikinya di situs Anda. Akun AWS Ini berarti bahwa sumber daya harus dialokasikan atau disediakan di akun Anda. Anda tidak dapat berbagi sumber daya yang telah dibagikan dengan Anda.
- Untuk berbagi sumber daya dengan organisasi Anda atau unit organisasi di AWS Organizations, Anda harus mengaktifkan berbagi dengan AWS Organizations. Untuk informasi selengkapnya, lihat [Mengaktifkan Berbagi dengan AWS Organizations](#) di Panduan AWS RAM Pengguna.

Berbagi asal VPC

Note

Saat ini, CloudFront mendukung berbagi asal VPC. Jika Anda belum membuatnya, lihat [Batasi akses dengan asal VPC](#).

Ketika Anda berbagi asal VPC yang Anda miliki dengan orang lain Akun AWS, Anda memungkinkan mereka untuk menggunakan sumber daya itu sebagai asal untuk distribusi mereka CloudFront .

Untuk berbagi asal VPC, Anda harus menambahkannya ke berbagi sumber daya. Pembagian sumber daya adalah AWS RAM sumber daya yang memungkinkan Anda berbagi sumber daya Anda Akun AWS.

Pembagian sumber daya menentukan hal berikut:

- Sumber daya yang ingin Anda bagikan
- Konsumen dengan siapa mereka berbagi
- Kebijakan terkelola layanan yang menentukan izin ke sumber daya

Saat membagikan asal VPC menggunakan CloudFront konsol, Anda menambahkannya ke pembagian sumber daya yang ada. Jika Anda belum memiliki pembagian sumber daya, Anda dapat membuatnya saat membagikan asal VPC dari konsol. CloudFront Anda juga dapat menggunakan [AWS RAM konsol](#) atau membuatnya AWS CLI secara terpisah.

Anda dapat berbagi asal VPC dengan yang lain Akun AWS dan. AWS Organizations

- Jika Anda berbagi sumber daya dengan AWS organisasi, semua konsumen di organisasi tertentu tersebut diizinkan mengakses asal VPC.

- Jika Anda berbagi dengan Akun AWS atau organisasi yang bukan bagian dari Anda, konsumen akan menerima undangan untuk menerima pembagian sumber daya. Setelah diterima, mereka dapat menggunakan asal VPC.

Anda dapat membagikan asal VPC yang Anda miliki menggunakan CloudFront konsol, AWS RAM konsol, atau. AWS CLI

Untuk membuat pembagian sumber daya dengan menggunakan CloudFront konsol

1. Buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
 2. Di panel navigasi, pilih asal VPC.
 3. Pilih satu atau beberapa sumber daya dan pilih Bagikan asal VPC.
 4. Pilih Buat berbagi sumber daya.
 5. Untuk Nama, masukkan nama untuk berbagi sumber daya.
 6. Untuk tipe Principal, pilih salah satu opsi berikut:
 - Akun AWS— Berikan akses ke yang spesifik Akun AWS.
 - Unit organisasi — Memberikan akses ke unit organisasi tertentu (OU).
 - Organisasi — Berikan akses ke seluruh organisasi Anda, termasuk anaknya OUs dan Akun AWS.
 - a. Jika Anda memilih Akun AWS, masukkan nomor ID akun. Anda dapat memilih Tambahkan akun baru untuk menambahkan hingga 5 Akun AWS.
 - b. Jika Anda memilih unit Organisasi, masukkan unit OU ARN. Anda hanya dapat memasukkan 1 OU.
 - c. Jika Anda memilih Organisasi, masukkan organisasi ARN. Anda hanya dapat memasukkan 1 organisasi.
7. Pilih Bagikan sumber daya.

Secara default, CloudFront menerapkan kebijakan [AWSRAMDefaultPermissionCloudFront](#) AWS terkelola pada pembagian sumber daya. Kebijakan ini memungkinkan tindakan hanya-baca pada pembagian sumber daya, sehingga akun yang menggunakan tidak dapat memperbarui atau menghapus sumber daya bersama. Anda tidak dapat mengedit atau menghapus kebijakan ini dari pembagian sumber daya.

 Tip

Setelah Anda membuat berbagi sumber daya, Anda dapat menambahkan tambahan Akun AWS dari AWS RAM konsol. Untuk informasi selengkapnya, lihat [Memperbarui pembagian sumber daya dalam AWS RAM](#) di Panduan AWS RAM Pengguna.

Untuk berbagi asal VPC yang Anda miliki menggunakan konsol CloudFront

1. Buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih asal VPC.
3. Pilih sumber daya dan pilih Bagikan asal VPC.
4. Di halaman Berbagi asal VPC, Anda dapat memilih pembagian sumber daya yang ada yang ingin Anda tambahkan asal VPC ini.
5. Pilih Bagikan sumber daya.

Pada halaman detail sumber daya, di bawah Dibagikan dengan, Anda dapat melihat bahwa asal VPC Anda dibagikan dengan detail berikut:

- Nama berbagi sumber daya
- Berbagi status
- Tanggal modifikasi terakhir

Setelah Anda membuat dan membagikan pembagian sumber daya dengan akun konsumen, mereka memiliki 12 jam untuk menerima undangan. Untuk informasi selengkapnya, lihat [Menerima dan menolak undangan berbagi sumber daya](#) di Panduan Pengguna AWS RAM

 Important

Untuk mengaktifkan akun pengguna menggunakan asal VPC Anda untuk CloudFront distribusinya, Anda juga harus memberi mereka Elastic Load Balancing asal VPC atau titik akhir Amazon EC2.

Untuk berbagi asal VPC yang Anda miliki menggunakan konsol AWS RAM

Buat berbagi sumber daya dan kemudian pilih CloudFront sumber daya yang ingin Anda tambahkan ke dalamnya. Untuk informasi selengkapnya, lihat [Membuat Pembagian Sumber Daya](#) di Panduan AWS RAM Pengguna.

Untuk berbagi asal VPC yang Anda miliki menggunakan AWS CLI

Gunakan perintah [create-resource-share](#).

Menggunakan asal VPC bersama

Untuk menggunakan asal VPC bersama, akun yang menerima undangan harus menerima pembagian sumber daya. Anda dapat melakukannya dengan menavigasi ke AWS Resource Access Manager konsol di Wilayah AS Timur (Virginia N.) dan menerima permintaan yang tertunda di tab Pending. Untuk informasi selengkapnya, lihat [Menerima sumber daya bersama](#) di Panduan AWS RAM Pengguna.

Setelah Anda menerima pembagian sumber daya, Anda kemudian dapat menggunakan asal VPC sebagai asal distribusi Anda CloudFront .

Untuk menggunakan asal VPC bersama

1. Buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Pada panel navigasi, untuk Distribusi, lakukan salah satu hal berikut:
 - Untuk distribusi baru, pilih Buat distribusi.
 - Untuk distribusi yang ada, pilih ID distribusi.
3. Untuk tipe Origin, pilih asal VPC, lalu tentukan asal VPC yang dibagikan dengan Anda.
4. Untuk titik akhir asal VPC, masukkan nama DNS pribadi instans Amazon EC2 atau penyeimbang beban Elastic Load Balancing, atau domain asal. Jika Anda belum memiliki nilai ini, Anda harus mendapatkannya dari Akun AWS yang memiliki asal VPC. Jika Anda belum memiliki titik akhir ini, Anda bisa mendapatkannya dari Akun AWS yang memiliki asal VPC.
5. Ikuti langkah-langkah konsol lainnya untuk membuat atau memperbarui distribusi Anda.

Mengidentifikasi asal VPC bersama

Pemilik dan konsumen dapat mengidentifikasi asal-usul VPC bersama menggunakan CloudFront konsol dan. AWS CLI

Untuk mengidentifikasi asal VPC bersama menggunakan konsol CloudFront

1. Buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih asal VPC. Anda dapat menggunakan kolom ID Pemilik untuk mengidentifikasi sumber daya milik Akun AWS
3. Pilih sumber daya.
4. Pada halaman detail sumber daya, di bawah Dibagikan dengan, Anda dapat melihat bahwa asal VPC Anda dibagikan dengan detail berikut:
 - Nama berbagi sumber daya
 - Berbagi status
 - Tanggal modifikasi terakhir

Membatalkan berbagi asal VPC bersama

Saat Anda membatalkan pembagian sumber daya, Akun AWS (akun yang dikonsumsi) tidak dapat lagi menggunakan sumber daya tersebut untuk distribusi baru atau memperbarui distribusi yang ada.

Note

Jika Anda membatalkan pembagian sumber daya, distribusi yang ada yang masih menggunakan sumber daya tersebut tetap aktif dan akan terus melayani lalu lintas. Namun, distribusi ini tidak dapat diedit sampai sumber daya yang tidak dibagikan dihapus sebagai asal. Kami menyarankan Anda memastikan bahwa setiap akun yang mengkonsumsi berhenti menggunakan sumber daya yang tidak dibagikan sebelum Anda membatalkan pembagiannya.

Untuk membatalkan berbagi asal VPC bersama yang Anda miliki, Anda harus menghapusnya dari pembagian sumber daya. Anda dapat melakukan ini menggunakan CloudFront konsol, AWS RAM konsol, atau AWS CLI.

Untuk membatalkan berbagi asal VPC bersama yang Anda miliki menggunakan konsol CloudFront

1. Buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih asal VPC.
3. Pilih sumber daya dan pilih Unshare.

4. Tinjau detailnya di kotak dialog Unshare resource lalu pilih Unshare. Prinsipal yang terdaftar tidak akan lagi memiliki akses ke sumber daya bersama Anda.

Untuk membatalkan berbagi asal VPC bersama yang Anda miliki menggunakan konsol AWS RAM

Lihat [Memperbarui Sumber Daya Bersama](#) di Panduan Pengguna AWS RAM .

Untuk membatalkan pembagian asal VPC bersama yang Anda miliki menggunakan AWS CLI

Gunakan perintah [disassociate-resource-share](#).

Tanggung jawab dan izin untuk asal VPC bersama

Izin untuk pemilik

Sebagai akun pemilik sumber daya, pastikan bahwa setiap akun yang mengkonsumsi berhenti menggunakan sumber daya sebelum Anda membatalkan pembagian atau menghapusnya.

Izin untuk konsumen

Akun yang dikonsumsi dapat menggunakan sumber daya bersama sebagai asal untuk CloudFront distribusinya, tetapi akun tersebut tidak dapat mengedit atau menghapus sumber daya. Secara default, kebijakan [AWSRAMDefaultPermissionCloudFront](#) AWS terkelola diterapkan pada pembagian sumber daya di akun berbagi (akun yang memiliki sumber daya).

AWSRAMDefaultPermissionCloudFront

Saat Anda membuat pembagian sumber daya CloudFront, CloudFront gunakan kebijakan AWSRAMDefaultPermissionCloudFront AWS terkelola dan terapkan pada pembagian sumber daya Anda. Kebijakan ini memberikan izin hanya-baca ke CloudFront sumber daya yang dapat dibagikan dari pemilik sumber daya ke akun pengguna.

Untuk informasi selengkapnya tentang mengelola izin AWS RAM, lihat [Mengelola izin AWS RAM di AWS Resource Access Manager](#) Panduan Pengguna.

Tagihan dan pengukuran

Tidak ada biaya tambahan untuk berbagi asal VPC dengan yang lain. Akun AWS Biaya penggunaan lalu lintas untuk distribusi yang menggunakan asal VPC bersama akan masuk ke akun konsumsi yang memiliki distribusi.

Kuota sumber daya bersama

CloudFront menggunakan kuota berbagi sumber daya yang sama seperti yang ditentukan oleh AWS RAM. Dari CloudFront konsol, Anda dapat menambahkan hingga 5 Akun AWS, 1 OU, atau 1 organisasi. Untuk menambahkan lebih banyak, gunakan AWS RAM konsol atau AWS RAM API.

Untuk informasi selengkapnya, lihat [Kuota layanan AWS RAM](#) di Panduan AWS RAM Pengguna.

Caching dan ketersediaan

Anda dapat menggunakannya CloudFront untuk mengurangi jumlah permintaan yang harus ditanggapi oleh server asal Anda secara langsung. Dengan CloudFront caching, lebih banyak objek dilayani dari lokasi CloudFront tepi, yang lebih dekat dengan pengguna Anda. Ini mengurangi beban di server asal Anda dan mengurangi latensi.

Semakin banyak permintaan yang CloudFront dapat ditayangkan dari cache tepi, semakin sedikit permintaan pemirsa yang CloudFront harus diteruskan ke asal Anda untuk mendapatkan versi terbaru atau versi unik dari suatu objek. CloudFront Untuk mengoptimalkan agar permintaan ke asal Anda sesedikit mungkin, pertimbangkan untuk menggunakan CloudFront Origin Shield. Untuk informasi selengkapnya, lihat [Gunakan Amazon CloudFront Origin Shield](#).

Proporsi permintaan yang disajikan langsung dari CloudFront cache dibandingkan dengan semua permintaan disebut rasio hit cache. Anda dapat melihat persentase permintaan pemirsa yang merupakan klik, kesalahan, dan kesalahan di CloudFront konsol. Untuk informasi selengkapnya, lihat [Lihat laporan statistik CloudFront cache](#).

Sejumlah faktor memengaruhi rasio tembolak. Anda dapat menyesuaikan konfigurasi CloudFront distribusi Anda untuk meningkatkan rasio hit cache dengan mengikuti panduan di [Tingkatkan proporsi permintaan yang disajikan langsung dari CloudFront cache \(rasio hit cache\)](#).

Untuk mempelajari tentang menambahkan dan menghapus konten yang CloudFront ingin Anda sajikan, lihat [Menambahkan, menghapus, atau mengganti konten yang CloudFront mendistribusikan](#).

Topik

- [Tingkatkan proporsi permintaan yang disajikan langsung dari CloudFront cache \(rasio hit cache\)](#)
- [Gunakan Amazon CloudFront Origin Shield](#)
- [Optimalkan ketersediaan tinggi dengan failover CloudFront asal](#)
- [Mengelola berapa lama konten tetap dalam cache \(kedaluwarsa\)](#)
- [Konten cache berdasarkan parameter string kueri](#)
- [Konten cache berdasarkan cookie](#)
- [Konten cache berdasarkan header permintaan](#)

Tingkatkan proporsi permintaan yang disajikan langsung dari CloudFront cache (rasio hit cache)

Anda dapat meningkatkan kinerja dengan meningkatkan proporsi permintaan pemirsa Anda yang disajikan langsung dari CloudFront cache alih-alih pergi ke server asal Anda untuk konten. Hal ini dikenal sebagai peningkatan rasio temuan cache.

Bagian berikut menjelaskan cara meningkatkan rasio temuan cache Anda.

Topik

- [Tentukan berapa lama CloudFront cache objek Anda](#)
- [Gunakan Origin Shield](#)
- [Caching berdasarkan parameter string kueri](#)
- [Memisahkan berdasarkan nilai cookie](#)
- [Menyimpan berdasarkan header permintaan](#)
- [Hapus Accept-Encoding header saat kompresi tidak diperlukan](#)
- [Sajikan konten media melalui HTTP](#)

Tentukan berapa lama CloudFront cache objek Anda

Untuk meningkatkan rasio temuan cache, Anda dapat mengonfigurasi asal Anda untuk menambah arahan [Cache-Control max-age](#) ke objek Anda, dan menentukan nilai praktis terpanjang untuk max-age. Semakin pendek durasi cache, semakin sering CloudFront mengirim permintaan ke asal Anda untuk menentukan apakah suatu objek telah berubah dan untuk mendapatkan versi terbaru. Anda dapat melengkapi max-age dengan stale-if-error arahan stale-while-revalidate dan untuk lebih meningkatkan rasio hit cache dalam kondisi tertentu. Untuk informasi selengkapnya, lihat [Mengelola berapa lama konten tetap dalam cache \(kedaluwarsa\)](#).

Gunakan Origin Shield

CloudFront Origin Shield dapat membantu meningkatkan rasio hit cache CloudFront distribusi Anda, karena menyediakan lapisan caching tambahan di depan asal Anda. Saat Anda menggunakan Origin Shield, semua permintaan dari CloudFront semua lapisan caching ke asal Anda berasal dari satu lokasi. CloudFront dapat mengambil setiap objek menggunakan permintaan asal tunggal dari Origin

Shield, dan semua lapisan cache lainnya (lokasi tepi dan CloudFront [cache tepi regional](#)) dapat mengambil objek dari Origin Shield.

Untuk informasi selengkapnya, lihat [Gunakan Amazon CloudFront Origin Shield](#).

Caching berdasarkan parameter string kueri

Jika Anda CloudFront mengkonfigurasi cache berdasarkan parameter string kueri, Anda dapat meningkatkan caching jika Anda melakukan hal berikut:

- Konfigurasi CloudFront untuk meneruskan hanya parameter string kueri yang asal Anda akan mengembalikan objek unik.
- Gunakan kasus yang sama (huruf besar atau kecil) untuk semua kasus parameter yang sama. Misalnya, jika satu permintaan berisi `parameter1=A` dan permintaan lainnya berisi `parameter1=a`, CloudFront teruskan permintaan terpisah ke asal Anda saat permintaan berisi `parameter1=A` dan saat permintaan berisi `parameter1=a`. CloudFront kemudian secara terpisah menyimpan objek terkait yang dikembalikan oleh asal Anda secara terpisah meskipun objeknya identik. Jika Anda hanya menggunakan `A` atau `a`, CloudFront mengirimkan lebih sedikit permintaan ke asal Anda.
- Cantumkan parameter dalam urutan yang sama. Seperti halnya perbedaan dalam kasus, jika satu permintaan untuk objek berisi string kueri `parameter1=a&parameter2=b` dan permintaan lain untuk objek yang sama berisi `parameter2=b&parameter1=a`, CloudFront teruskan kedua permintaan ke asal Anda dan secara terpisah menyimpan objek yang sesuai meskipun keduanya identik. Jika Anda selalu menggunakan urutan parameter yang sama, CloudFront teruskan lebih sedikit permintaan ke asal Anda.

Untuk informasi selengkapnya, lihat [Konten cache berdasarkan parameter string kueri](#). Jika Anda ingin meninjau string kueri yang CloudFront diteruskan ke asal Anda, lihat nilai di `cs-uri-query` kolom file CloudFront log Anda. Untuk informasi selengkapnya, lihat [Akses log \(log standar\)](#).

Memisahkan berdasarkan nilai cookie

Jika Anda CloudFront mengonfigurasi cache berdasarkan nilai cookie, Anda dapat meningkatkan caching jika Anda melakukan hal berikut:

- Konfigurasi CloudFront untuk meneruskan hanya cookie tertentu alih-alih meneruskan semua cookie. Untuk cookie yang Anda konfigurasi CloudFront untuk meneruskan ke asal Anda,

CloudFront teruskan setiap kombinasi nama dan nilai cookie. Kemudian dia menyimpan secara terpisah objek yang dikembalikan asal Anda, bahkan jika semuanya identik.

Misalnya, anggaplah bahwa pemirsa menyertakan dua cookie dalam setiap permintaan, bahwa setiap cookie memiliki tiga nilai yang mungkin, dan bahwa semua kombinasi nilai cookie dimungkinkan. CloudFront meneruskan hingga sembilan permintaan berbeda ke asal Anda untuk setiap objek. Jika asal Anda mengembalikan versi objek yang berbeda hanya berdasarkan salah satu cookie, maka meneruskan CloudFront lebih banyak permintaan ke asal Anda daripada yang diperlukan dan tidak perlu menyimpan beberapa versi objek yang identik.

- Buat perilaku cache terpisah untuk konten statis dan dinamis, dan konfigurasi CloudFront untuk meneruskan cookie ke asal Anda hanya untuk konten dinamis.

Misalnya, Anda hanya memiliki satu perilaku cache untuk distribusi Anda dan bahwa Anda menggunakan distribusi baik untuk konten dinamis, seperti `.js` file, dan untuk `.css` file yang jarang berubah. CloudFront cache versi terpisah dari `.css` file Anda berdasarkan nilai cookie, sehingga setiap lokasi CloudFront tepi meneruskan permintaan ke asal Anda untuk setiap nilai cookie baru atau kombinasi nilai cookie.

Jika Anda membuat perilaku cache yang pola jalurnya `*.css` dan yang CloudFront tidak di-cache berdasarkan nilai cookie, maka CloudFront teruskan permintaan `.css` file ke asal Anda hanya untuk permintaan pertama yang diterima lokasi tepi untuk `.css` file tertentu dan untuk permintaan pertama setelah `.css` file kedaluwarsa.

- Jika memungkinkan, buat perilaku cache terpisah untuk konten dinamis ketika nilai cookie unik untuk setiap pengguna (seperti ID pengguna), dan konten dinamis yang bervariasi berdasarkan jumlah nilai unik yang lebih kecil.

Untuk informasi selengkapnya, lihat [Konten cache berdasarkan cookie](#). Jika Anda ingin meninjau cookie yang CloudFront diteruskan ke asal Anda, lihat nilai di `cs(Cookie)` kolom file CloudFront log Anda. Untuk informasi selengkapnya, lihat [Akses log \(log standar\)](#).

Menyimpan berdasarkan header permintaan

Jika Anda CloudFront mengonfigurasi cache berdasarkan header permintaan, Anda dapat meningkatkan caching jika Anda melakukan hal berikut:

- Konfigurasi CloudFront untuk meneruskan dan cache hanya berdasarkan header yang ditentukan, bukan penerusan dan caching berdasarkan semua header. Untuk header yang Anda

tentukan, CloudFront teruskan setiap kombinasi nama dan nilai header. Kemudian ini menyimpan objek secara terpisah yang asal Anda kembali meskipun semuanya identik.

 Note

CloudFront selalu meneruskan ke asal Anda header yang ditentukan dalam topik berikut:

- Cara CloudFront memproses dan meneruskan permintaan ke server asal Amazon S3 Anda > [Header permintaan HTTP yang CloudFront menghapus atau memperbarui](#)
- Cara CloudFront memproses dan meneruskan permintaan ke server asal kustom Anda > [Header dan CloudFront perilaku permintaan HTTP \(asal kustom dan Amazon S3\)](#)

Saat Anda CloudFront mengonfigurasi cache berdasarkan header permintaan, Anda tidak mengubah header yang CloudFront diteruskan, hanya jika CloudFront cache objek berdasarkan nilai header.

- Coba hindari cache berdasarkan header permintaan yang memiliki nilai unik dalam jumlah besar.

Misalnya, jika Anda ingin menyajikan ukuran gambar yang berbeda berdasarkan perangkat pengguna, maka jangan CloudFront mengkonfigurasi cache berdasarkan User-Agent header, yang memiliki sejumlah besar kemungkinan nilai. Sebagai gantinya, konfigurasi CloudFront ke cache berdasarkan header CloudFront tipe perangkat CloudFront-Is-Desktop-Viewer,,, CloudFront-Is-Mobile-Viewer dan. CloudFront-Is-SmartTV-Viewer CloudFront-Is-Tablet-Viewer Selain itu, jika Anda mengembalikan versi citra yang sama untuk tablet dan desktop, maka teruskan header CloudFront-Is-Tablet-Viewer saja, bukan header CloudFront-Is-Desktop-Viewer.

Untuk informasi selengkapnya, lihat [Konten cache berdasarkan header permintaan](#).

Hapus **Accept-Encoding** header saat kompresi tidak diperlukan

Jika kompresi tidak diaktifkan—karena asal tidak mendukungnya, CloudFront tidak mendukungnya, atau konten tidak dapat dikompresikan—Anda dapat meningkatkan rasio hit cache dengan mengaitkan perilaku cache dalam distribusi Anda ke asal yang menetapkan sebagai berikut: Custom Origin Header

- Nama header: Accept-Encoding
- Nilai header: (Biarkan kosong)

Saat Anda menggunakan konfigurasi ini, CloudFront hapus Accept-Encoding header dari kunci cache dan tidak menyertakan header dalam permintaan asal. Konfigurasi ini berlaku untuk semua konten yang CloudFront berfungsi dengan distribusi dari asal itu.

Sajikan konten media melalui HTTP

Untuk informasi tentang mengoptimalkan video sesuai permintaan (VOD) dan konten video streaming, lihat [Video sesuai permintaan dan video streaming langsung dengan CloudFront](#).

Gunakan Amazon CloudFront Origin Shield

CloudFront Origin Shield adalah lapisan tambahan dalam infrastruktur CloudFront caching yang membantu meminimalkan beban asal Anda, meningkatkan ketersediaannya, dan mengurangi biaya operasinya. Dengan CloudFront Origin Shield, Anda mendapatkan manfaat berikut:

Rasio temuan cache yang lebih baik

Origin Shield dapat membantu meningkatkan rasio hit cache CloudFront distribusi Anda karena menyediakan lapisan caching tambahan di depan asal Anda. Saat Anda menggunakan Origin Shield, semua permintaan dari CloudFront semua lapisan caching ke asal Anda melalui Origin Shield, meningkatkan kemungkinan terkena cache. CloudFront dapat mengambil setiap objek dengan permintaan asal tunggal dari Origin Shield ke asal Anda, dan semua lapisan cache lainnya (lokasi tepi dan CloudFront [cache tepi regional](#)) dapat mengambil objek dari Origin Shield.

Pengurangan muatan asal

Origin Shield dapat mengurangi lebih lanjut jumlah [permintaan secara bersamaan](#) yang dikirimkan ke tempat asal Anda untuk objek yang sama. Permintaan konten yang tidak berada dalam cache Origin Shield digabungkan dengan permintaan lain untuk objek yang sama, yang mengakibatkan sedikit mungkin permintaan yang dikirimkan ke asal Anda. Menangani lebih sedikit permintaan di tempat asal Anda dapat mempertahankan ketersediaan asal Anda selama beban puncak atau lonjakan lalu lintas yang tidak terduga, dan dapat mengurangi biaya untuk hal-hal seperti just-in-time pengemasan, transformasi gambar, dan transfer data keluar (DTO).

Kinerja jaringan yang lebih baik

Bila Anda mengaktifkan Origin Shield di AWS Wilayah [yang memiliki latensi terendah ke asal Anda, Anda](#) bisa mendapatkan performa jaringan yang lebih baik. Untuk asal-usul di suatu AWS Wilayah, lalu lintas CloudFront jaringan tetap berada di CloudFront jaringan throughput tinggi

sampai ke asal Anda. Untuk asal di luar AWS, lalu lintas CloudFront jaringan tetap berada di CloudFront jaringan sampai ke Origin Shield, yang memiliki koneksi latensi rendah ke asal Anda.

Anda akan dikenakan biaya tambahan untuk menggunakan Origin Shield. Untuk informasi selengkapnya, lihat [harga CloudFront](#).

Note

Origin Shield tidak didukung dengan permintaan gRPC. Jika distribusi yang mendukung gRPC mengaktifkan Origin Shield, permintaan gRPC akan terus berfungsi. Namun, permintaan akan diproksi langsung ke asal gRPC tanpa melalui Origin Shield. Untuk informasi selengkapnya, lihat [Menggunakan gRPC dengan distribusi CloudFront](#).

Topik

- [Gunakan kasus untuk Tameng Asal](#)
- [Pilih AWS Wilayah untuk Origin Shield](#)
- [Aktifkan Origin Shield](#)
- [Perkirakan biaya Origin Shield](#)
- [Ketersediaan tinggi Origin Shield](#)
- [Bagaimana Origin Shield berinteraksi dengan fitur lain CloudFront](#)

Gunakan kasus untuk Tameng Asal

CloudFront Origin Shield dapat bermanfaat untuk banyak kasus penggunaan, termasuk yang berikut:

- Penampil yang tersebar di berbagai wilayah geografis
- Origins yang menyediakan just-in-time kemasan untuk streaming langsung atau pemrosesan on-the-fly gambar
- Asal di lokasi dengan kapasitas atau keterbatasan bandwidth
- Beban kerja yang menggunakan beberapa jaringan pengiriman konten () CDNs

Tameng Asal mungkin tidak cocok untuk kasus lain, seperti konten dinamis yang disadarkan dengan asal, konten dengan kemampuan cache rendah, atau konten yang tidak sering diminta.

Bagian berikut menjelaskan manfaat Tameng Asal untuk kasus penggunaan berikut.

Kasus penggunaan

- [Penampil di wilayah geografis yang berbeda](#)
- [Berganda CDNs](#)

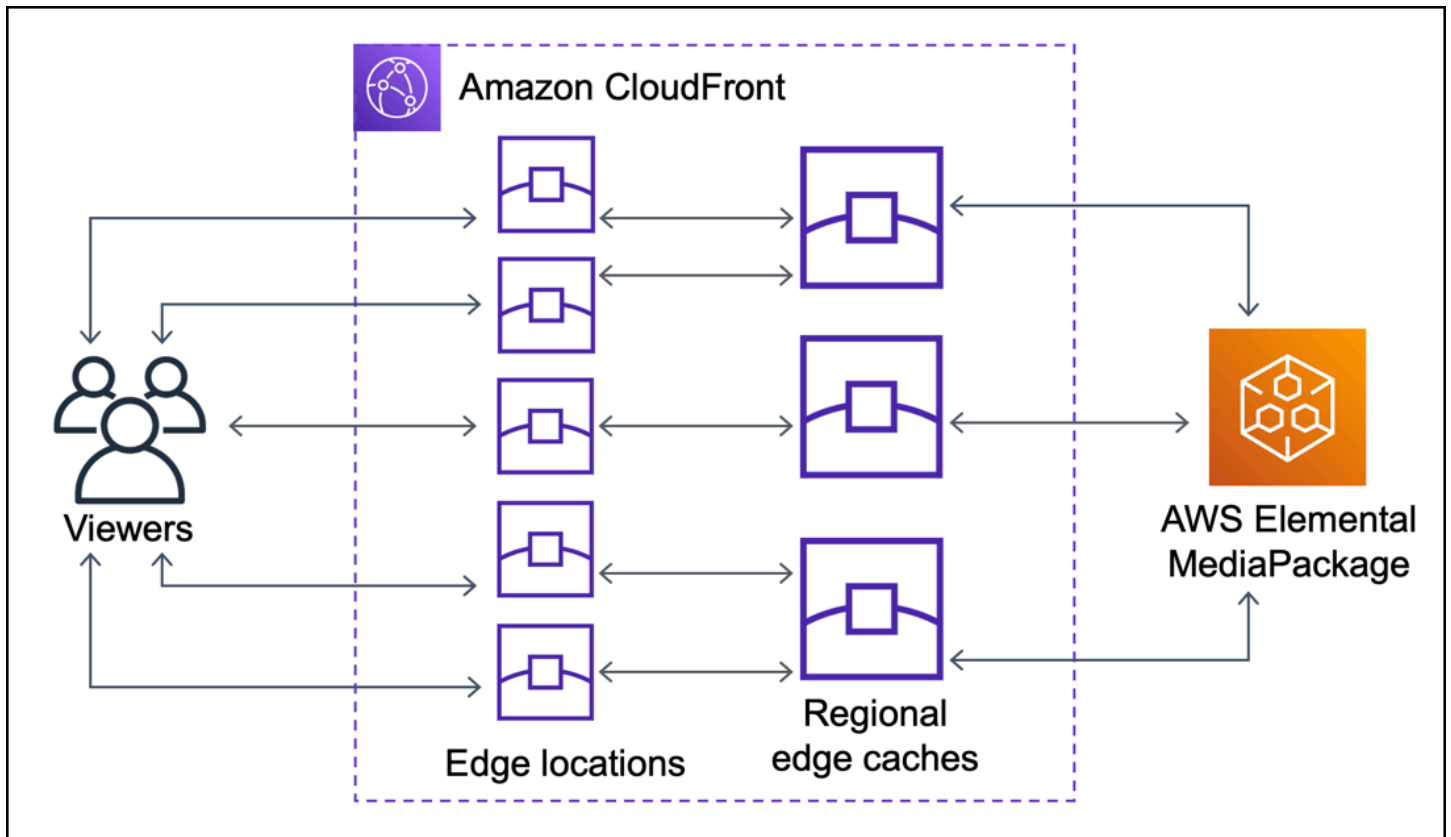
Penampil di wilayah geografis yang berbeda

Dengan Amazon CloudFront, Anda secara inheren mendapatkan pengurangan beban pada asal Anda karena permintaan yang CloudFront dapat ditayangkan dari cache tidak masuk ke asal Anda. Selain [jaringan global lokasi edge, cache edge regional berfungsi sebagai lapisan caching](#) tingkat menengah untuk memberikan klik cache dan mengkonsolidasikan permintaan asal untuk pemirsa di wilayah geografis terdekat. CloudFront Permintaan penampil dirutekan terlebih dahulu ke lokasi terdekat CloudFront lokasi tepian, dan jika objek tidak tersimpan di lokasi tersebut, permintaan dikirim ke cache tepi regional.

Ketika penampil berada di wilayah geografis yang berbeda, permintaan dapat diarahkan melalui cache edge regional yang berbeda, yang masing-masing dapat mengirim permintaan ke asal Anda untuk konten yang sama. Tetapi dengan Origin Shield, Anda mendapatkan lapisan tambahan cache antara edge cache regional dan tempat asal Anda. Semua permintaan dari semua cache tepi regional masuk melalui Origin Shield, lebih lanjut mengurangi muatan di tempat asal Anda. Diagram berikut menggambarkan hal ini. Dalam diagram berikut, asalnya adalah AWS Elemental MediaPackage

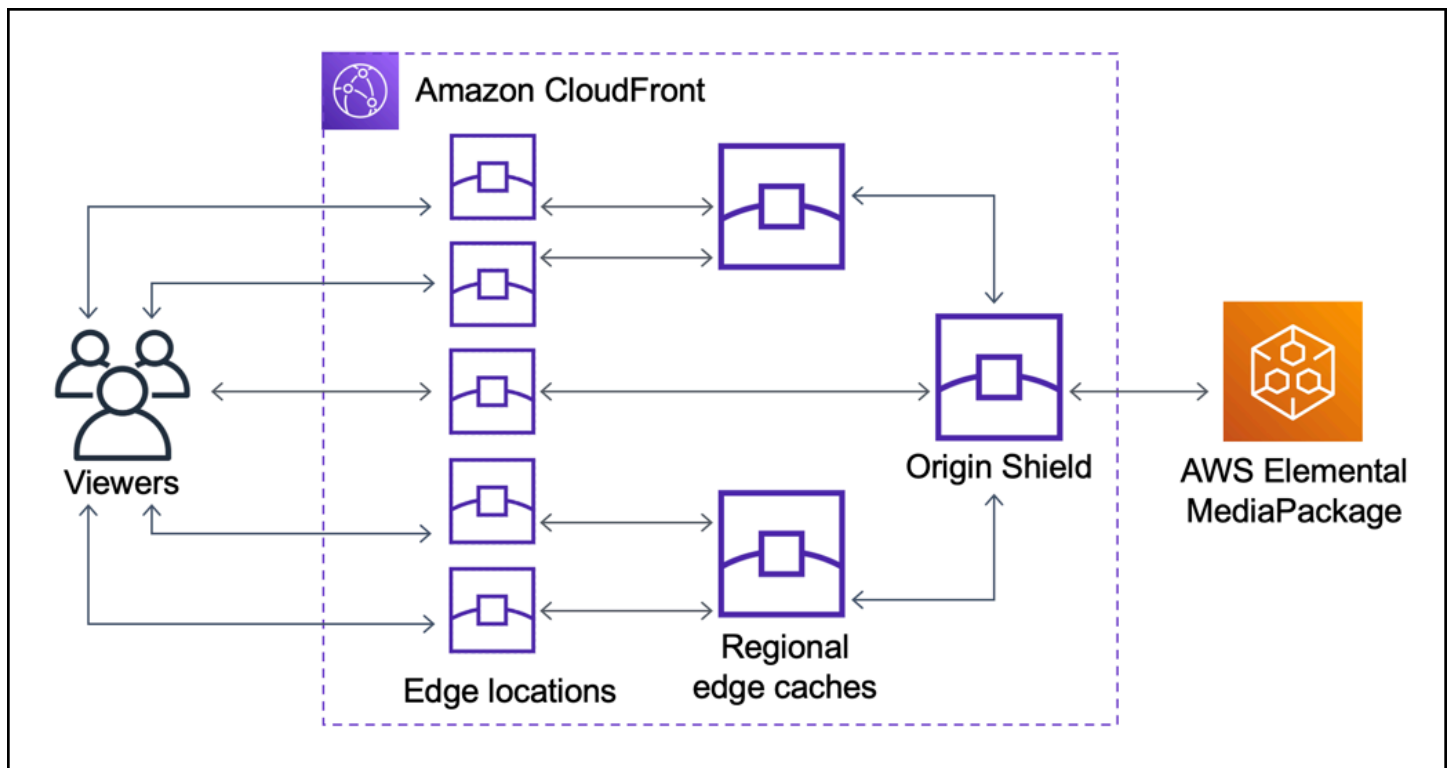
Tanpa Origin Shield

Tanpa Origin Shield, negara asal Anda mungkin menerima permintaan duplikat untuk konten yang sama, seperti yang ditunjukkan pada diagram berikut.



Dengan Origin Shield

Menggunakan Tameng Asal dapat membantu mengurangi beban pada asal Anda, seperti yang ditunjukkan dalam diagram berikut.



Berganda CDNs

Untuk menayangkan acara video langsung atau konten sesuai permintaan populer, Anda dapat menggunakan beberapa jaringan pengiriman konten (CDNs). Menggunakan beberapa CDNs dapat menawarkan keuntungan tertentu, tetapi itu juga berarti bahwa asal Anda mungkin menerima banyak permintaan duplikat untuk konten yang sama, masing-masing berasal dari lokasi yang berbeda CDNs atau berbeda dalam CDN yang sama. Permintaan berlebihan ini dapat mempengaruhi ketersediaan asal Anda atau menyebabkan biaya operasi tambahan untuk proses seperti just-in-time pengemasan atau transfer data (DTO) ke internet.

Ketika Anda menggabungkan Origin Shield dengan menggunakan CloudFront distribusi Anda sebagai asal untuk yang lain CDNs, Anda bisa mendapatkan manfaat berikut:

- Lebih sedikit permintaan berlebihan yang diterima di negara asal Anda, yang membantu mengurangi dampak negatif penggunaan banyak CDNs.
- Umum [kunci cache](#) di seluruh CDNs, dan manajemen terpusat untuk fitur asal-usul.
- Peningkatan kinerja jaringan. Lalu lintas jaringan dari CDNs yang lain dihentikan di lokasi CloudFront tepi terdekat, yang mungkin memberikan hit dari cache lokal. Jika objek yang diminta tidak berada di cache lokasi tepi, permintaan ke asal tetap ada di CloudFront jaringan sampai ke Origin Shield, yang memberikan throughput tinggi dan latensi rendah ke asal. Jika objek

yang diminta berada dalam cache Origin Shield, permintaan kepada asal Anda dapat dihindari sepenuhnya.

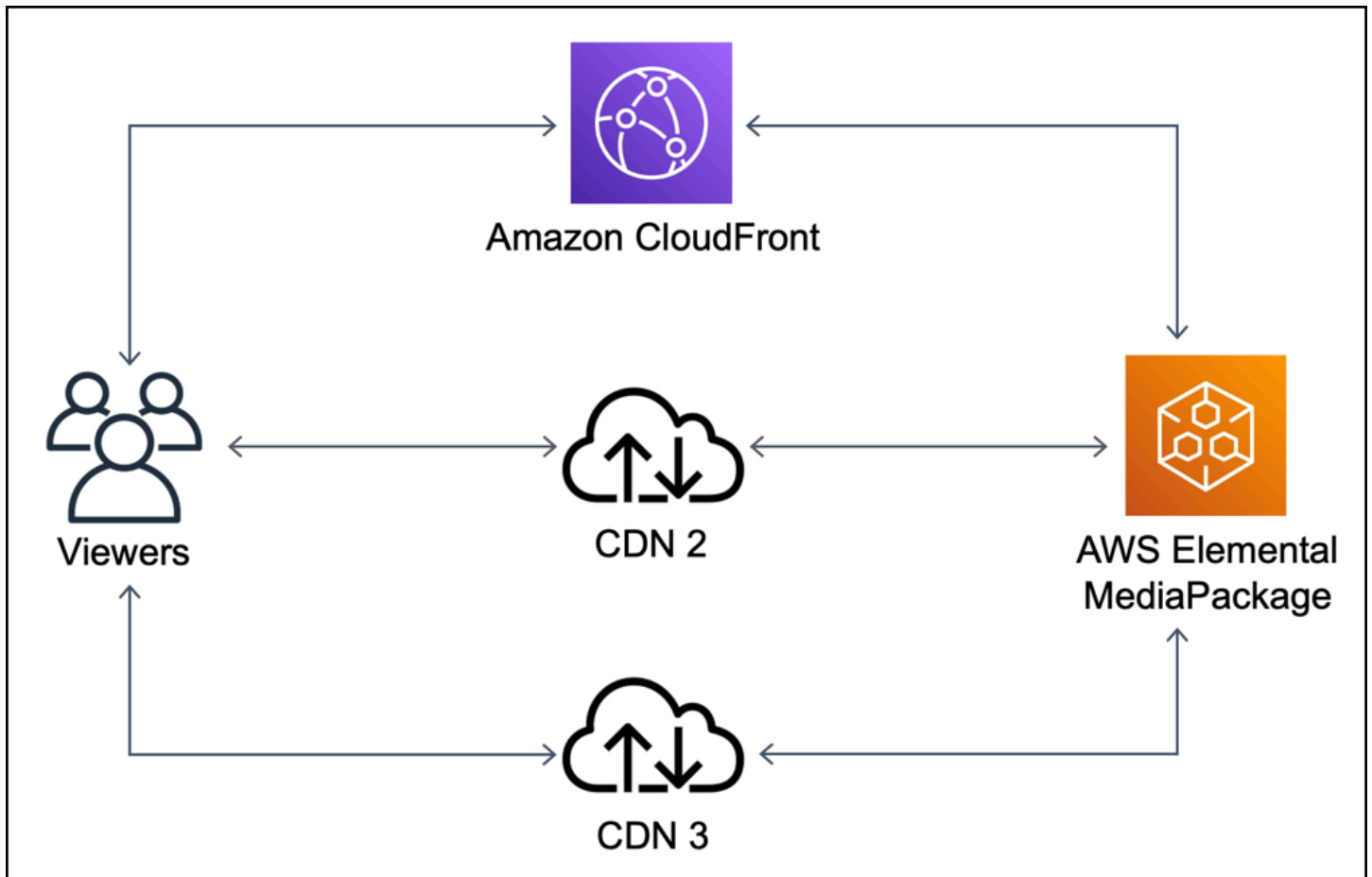
 Important

Jika Anda tertarik untuk menggunakan Origin Shield dalam arsitektur multi-CDN, dan memiliki harga diskon, [hubungi kami](#) atau perwakilan AWS penjualan Anda untuk informasi lebih lanjut. Mungkin dapat dikenakan biaya tambahan.

Diagram berikut menunjukkan bagaimana konfigurasi ini dapat membantu meminimalkan beban pada asal Anda saat Anda menyajikan acara video langsung populer dengan beberapa CDNs. Dalam diagram berikut, asalnya adalah AWS Elemental MediaPackage.

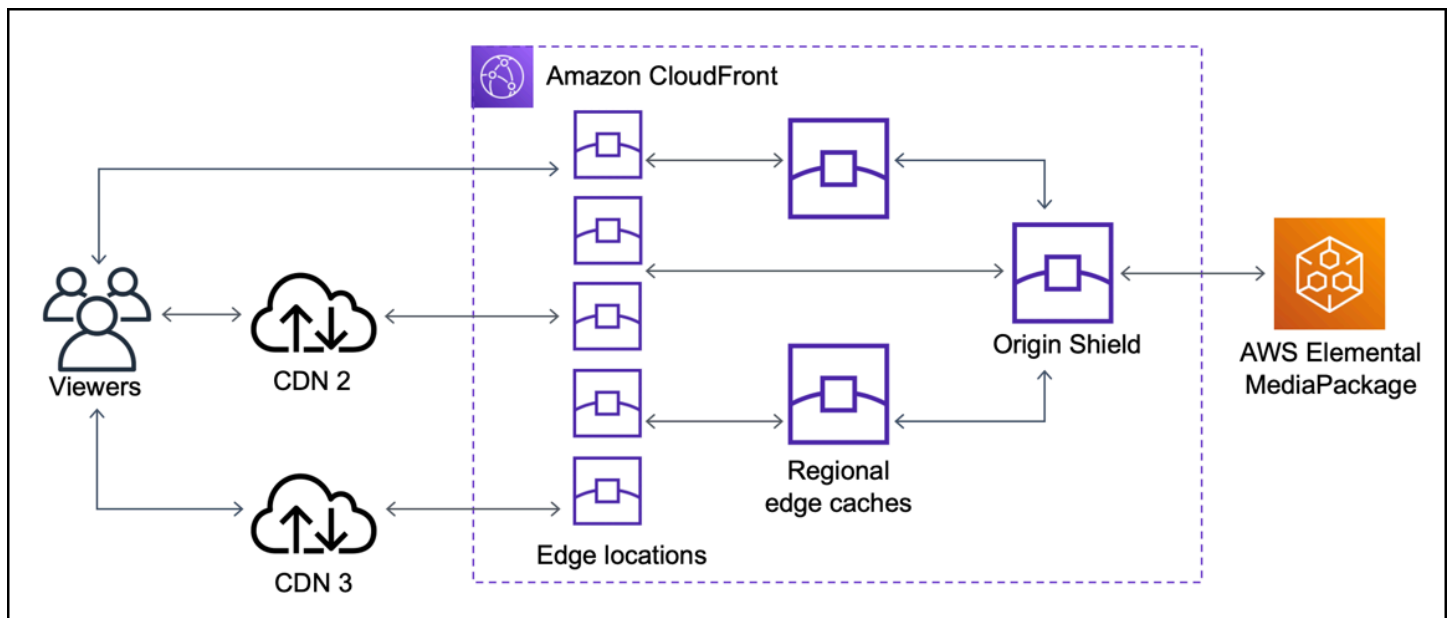
Tanpa Origin Shield (beberapa CDNs)

Tanpa Origin Shield, negara asal Anda mungkin menerima banyak permintaan duplikat untuk konten yang sama, yang masing-masing berasal dari CDN yang berbeda, seperti yang ditunjukkan pada diagram berikut.



Dengan Origin Shield (beberapa CDNs)

Menggunakan Origin Shield, dengan CloudFront sebagai asal untuk yang lain CDNs, dapat membantu mengurangi beban pada asal Anda, seperti yang ditunjukkan pada diagram berikut.



Pilih AWS Wilayah untuk Origin Shield

Amazon CloudFront menawarkan Origin Shield di AWS Wilayah yang CloudFront memiliki [cache tepi regional](#). Saat mengaktifkan Origin Shield, Anda memilih AWS Region for Origin Shield. Anda harus memilih Wilayah AWS yang memiliki latensi terendah dengan asal Anda. Anda dapat menggunakan Origin Shield dengan asal yang ada di AWS Wilayah, dan dengan asal yang tidak ada AWS.

Untuk asal-usul di suatu AWS Wilayah

Jika asal Anda berada di suatu AWS Wilayah, tentukan terlebih dahulu apakah asal Anda berada di Wilayah yang CloudFront menawarkan Origin Shield. CloudFront menawarkan Origin Shield di AWS Wilayah berikut.

- AS Timur (Ohio) – us-east-2
- AS Timur (Virginia Utara) – us-east-1
- AS Barat (Oregon) – us-west-2
- Asia Pasifik (Mumbai)–ap-south-1
- Asia Pasifik (Seoul)–ap-northeast-2
- Asia Pasifik (Singapura)–ap-southeast-1
- Asia Pasifik (Sydney)–ap-southeast-2
- Asia Pasifik (Tokyo) – ap-northeast-1

- Eropa (Frankfurt) – eu-central-1
- Eropa (Irlandia)–eu-west-1
- Eropa (London) – eu-west-2
- Amerika Selatan (São Paulo) – sa-east-1
- Timur Tengah (UEA) - me-central-1

Jika asal Anda berada di AWS Wilayah yang CloudFront menawarkan Origin Shield

Jika asal Anda berada di AWS Wilayah yang CloudFront menawarkan Origin Shield (lihat daftar sebelumnya), aktifkan Origin Shield di Wilayah yang sama dengan asal Anda.

Jika asal Anda tidak berada di AWS Wilayah yang CloudFront menawarkan Origin Shield

Jika asal Anda tidak berada di AWS Wilayah yang CloudFront menawarkan Origin Shield, lihat tabel berikut untuk menentukan Wilayah mana yang akan mengaktifkan Origin Shield.

Jika asal Anda berada di...	Aktifkan Origin Shield di...
AS Barat (N. California) – us-west-1	AS Barat (Oregon) – us-west-2
Afrika (Cape Town) – af-south-1	Eropa (Irlandia) – eu-west-1
Asia Pasifik (Hong Kong) – ap-east-1	Asia Pasifik (Singapura) – ap-southeast-1
Kanada (Pusat) – ca-central-1	AS Timur (Virginia Utara) – us-east-1
Eropa (Milan) – eu-south-1	Eropa (Frankfurt) – eu-central-1
Eropa (Paris) – eu-west-3	Eropa (London) – eu-west-2
Eropa (Stockholm) – eu-north-1	Eropa (London) – eu-west-2
Timur Tengah (Bahrain) – me-south-1	Asia Pasifik (Mumbai) – ap-south-1

Untuk asal-usul di luar AWS

Anda dapat menggunakan Origin Shield dengan asal yang berada di lokasi atau tidak berada di Wilayah AWS . Dalam hal ini, aktifkan Origin Shield di AWS Wilayah yang memiliki latensi terendah

ke asal Anda. Jika Anda tidak yakin AWS Wilayah mana yang memiliki latensi terendah ke asal Anda, Anda dapat menggunakan saran berikut untuk membantu Anda menentukan.

- Anda dapat melihat tabel sebelumnya untuk perkiraan Wilayah AWS yang mungkin memiliki latensi terendah ke asal Anda, berdasarkan lokasi geografis asal Anda.
- Anda dapat meluncurkan instans Amazon EC2 di beberapa AWS Wilayah berbeda yang secara geografis dekat dengan asal Anda, dan menjalankan beberapa pengujian ping untuk mengukur latensi jaringan tipikal antara Wilayah tersebut dan asal Anda.

Aktifkan Origin Shield

Anda dapat mengaktifkan Origin Shield untuk meningkatkan rasio tekan cache, mengurangi beban pada asal Anda, dan membantu meningkatkan kinerja. Untuk mengaktifkan Origin Shield, ubah pengaturan asal dalam CloudFront distribusi. Tameng Asal adalah milik asal usul. Untuk setiap asal dalam CloudFront distribusi Anda, Anda dapat mengaktifkan Origin Shield secara terpisah di AWS Wilayah mana pun yang memberikan kinerja terbaik untuk asal tersebut.

Anda dapat mengaktifkan Origin Shield di CloudFront konsol CloudFormation, dengan, atau dengan CloudFront API.

Console

Untuk mengaktifkan Tameng Asal untuk asal yang ada (konsol)

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Pilih distribusi yang memiliki asal yang ingin Anda perbarui.
3. Pilih tab Origins.
4. Pilih asal pembaruan, lalu pilih Edit.
5. Untuk Aktifkan Perisai Asal, pilih Ya.
6. Untuk Wilayah Origin Shield, pilih Wilayah AWS tempat Anda ingin mengaktifkan Origin Shield. Untuk bantuan memilih Wilayah, lihat [Pilih AWS Wilayah untuk Origin Shield](#).
7. Pilih Simpan perubahan.

Saat status distribusi Anda Diterapkan, Tameng Asal sudah siap. Ini memerlukan waktu beberapa menit.

Untuk mengaktifkan Shield Asal untuk asal baru (konsol)

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Untuk membuat asal baru dalam distribusi yang sudah ada, lakukan hal berikut ini:
 1. Pilih distribusi tempat Anda ingin membuat asal.
 2. Pilih Buat Asal, lalu lanjutkan ke langkah 3.

Untuk membuat asal baru dalam distribusi standar baru, lakukan hal berikut:

1. Ikuti langkah-langkah untuk membuat distribusi standar di konsol. Untuk informasi selengkapnya, lihat [Buat CloudFront distribusi di konsol](#).
2. Di bagian Pengaturan, pilih Sesuaikan pengaturan asal. Lanjutkan ke langkah 3.
3. Untuk Aktifkan Perisai Asal, pilih Ya.
4. Untuk Wilayah Origin Shield, pilih Wilayah AWS tempat Anda ingin mengaktifkan Origin Shield. Untuk bantuan memilih Wilayah, lihat [Pilih AWS Wilayah untuk Origin Shield](#).
5. Ikuti langkah-langkah di konsol untuk menyelesaikan pembuatan asal atau distribusi Anda.

Saat status distribusi Anda Diterapkan, Tameng Asal sudah siap. Ini memerlukan waktu beberapa menit.

CloudFormation

Untuk mengaktifkan Origin Shield dengan CloudFormation, gunakan `OriginShield` properti dalam jenis `Origin` properti dalam `AWS::CloudFront::Distribution` sumber daya. Anda dapat menambahkan `OriginShield` menjadi `Origin`, atau masukkan saat Anda membuat `Origin`.

Contoh berikut menunjukkan sintaks, dalam format YAML, untuk memungkinkan `OriginShield` di Wilayah (Oregon) Barat AS (`us-west-2`). Untuk bantuan memilih Wilayah, lihat [the section called "Pilih AWS Wilayah untuk Origin Shield"](#). Contoh ini hanya menunjukkan `Origin` jenis properti, bukan keseluruhan `AWS::CloudFront::Distribution` sumber daya.

```
Origins:
- DomainName: 3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com
  Id: Example-EMP-3ae97e9482b0d011
  OriginShield:
```

```
Enabled: true
OriginShieldRegion: us-west-2
CustomOriginConfig:
  OriginProtocolPolicy: match-viewer
  OriginSSLProtocols: TLSv1
```

Untuk informasi selengkapnya, lihat [AWS::CloudFront::Distribution Asal](#) di bagian referensi sumber daya dan properti di Panduan AWS CloudFormation Pengguna.

API

Untuk mengaktifkan Origin Shield dengan CloudFront API menggunakan AWS SDKs or AWS Command Line Interface (AWS CLI), gunakan `OriginShield` tipe. Anda menentukan `OriginShield` dalam `Origin`, dalam `DistributionConfig`. Untuk informasi tentang `OriginShield` jenisnya, lihat informasi berikut di Referensi Amazon CloudFront API.

- [OriginShield](#) (jenis)
- [Asal](#) (jenis)
- [DistributionConfig](#) (jenis)
- [UpdateDistribution](#) (operasi)
- [CreateDistribution](#) (operasi)

Sintaks spesifik untuk menggunakan jenis dan operasi ini berbeda-beda berdasarkan klien SDK, CLI, atau API. Untuk informasi lebih lanjut, lihat dokumentasi referensi untuk SDK, CLI, atau klien Anda.

Perkirakan biaya Origin Shield

Anda mengakumulasikan biaya untuk Tameng Asal berdasarkan jumlah permintaan yang masuk ke Tameng Asal sebagai lapisan tambahan.

Untuk permintaan dinamis (non-cacheable) yang berhubungan dengan asal usul, Shield Asal selalu merupakan lapisan tambahan. Permintaan dinamis menggunakan metode HTTP PUT, POST, PATCH, dan DELETE.

GET dan HEAD permintaan yang memiliki pengaturan waktu untuk hidup (TTL) kurang dari 3600 detik dianggap sebagai permintaan dinamis. Selain itu, GET dan HEAD permintaan yang telah menonaktifkan caching juga dianggap permintaan dinamis.

Untuk memperkirakan biaya untuk Shield Asal untuk permintaan dinamis, gunakan rumus berikut:

Total jumlah permintaan dinamis x Biaya Tameng Asal per 10.000 permintaan / 10.000

Untuk permintaan non-dinamis dengan metode HTTPGET,, dan HEADOPTIONS, Origin Shield terkadang merupakan lapisan tambahan. Ketika Anda mengaktifkan Origin Shield, Anda memilih Wilayah AWS untuk Origin Shield. Untuk permintaan yang secara alami masuk ke [cache tepi regional](#) di Region yang sama dengan Origin Shield, Origin Shield bukanlah lapisan tambahan. Anda tidak dikenakan biaya Origin Shield untuk permintaan ini. Untuk permintaan yang masuk ke cache edge regional di Region yang berbeda dari Origin Shield, dan kemudian pergi ke Origin Shield, Origin Shield adalah lapisan tambahan. Anda mengakumulasi biaya Tameng Asal untuk permintaan ini.

Untuk memperkirakan biaya Anda untuk Perisai Asal untuk permintaan yang dapat disimpan, gunakan rumus berikut:

Total jumlah permintaan yang dapat disimpan x (1 – laju ketukan cache) x persentase permintaan yang masuk ke Origin Shield dari edge cache regional di wilayah yang berbeda x Biaya Tameng Asal per 10.000 permintaan / 10.000

Untuk informasi lebih lanjut tentang biaya per 10.000 permintaan untuk Origin Shield, lihat [CloudFront Harga](#).

Ketersediaan tinggi Origin Shield

Origin Shield memanfaatkan fitur [cache edge CloudFront regional](#). Masing-masing cache edge ini dibangun di AWS Wilayah menggunakan setidaknya tiga [Availability Zone dengan armada instans](#) Amazon EC2 auto-scaling. Koneksi dari CloudFront Lokasi ke Tameng Asal juga menggunakan pelacakan kesalahan aktif untuk setiap permintaan untuk secara otomatis mengirimkan permintaan ke lokasi Tameng Asal sekunder jika lokasi Perisai Asal primer tidak tersedia.

Bagaimana Origin Shield berinteraksi dengan fitur lain CloudFront

Bagian berikut menjelaskan bagaimana Origin Shield berinteraksi dengan CloudFront yang berbeda.

Origin Shield dan CloudFront logging

Untuk melihat kapan Origin Shield menangani permintaan, Anda harus mengaktifkan salah satu dari yang berikut:

- [CloudFront log standar \(log akses\)](#). Catatan standar disediakan secara gratis.

- [CloudFront log akses real-time](#). Anda dikenakan biaya tambahan untuk menggunakan log akses real-time. Lihat [Amazon CloudFront Harga](#).

Cache hits dari Origin Shield muncul seperti `OriginShieldHit` di `x-edge-detailed-result-type` bidang di CloudFront log. Origin Shield memanfaatkan [cache edge regional](#) Amazon CloudFront. Jika permintaan dirutekan dari lokasi CloudFront tepi ke cache tepi regional yang bertindak sebagai Origin Shield, permintaan tersebut dilaporkan sebagai a Hit di log, bukan sebagai `OriginShieldHit`.

Origin Shield dan kelompok asal

Origin Shield kompatibel dengan [CloudFront kelompok asal](#). Karena Origin Shield adalah properti asal, permintaan selalu melakukan perjalanan melalui Origin Shield untuk setiap asal bahkan ketika asal-usul adalah bagian dari kelompok asal. Untuk permintaan tertentu, CloudFront merutekan permintaan ke asal utama dalam grup asal melalui Origin Shield asal utama. Jika permintaan tersebut gagal (sesuai dengan kriteria failover grup asal), CloudFront rutekan permintaan ke asal sekunder melalui Origin Shield asal sekunder.

Perisai Asal dan Lambda@Edge

Origin Shield tidak memengaruhi fungsionalitas dari fungsi [Lambda@Edge](#) namun dapat memengaruhi Wilayah AWS tempat fungsi tersebut dijalankan.

Saat Anda menggunakan Origin Shield dengan Lambda @Edge, [pemicu yang menghadap ke](#) asal (permintaan asal dan respons asal) berjalan di Wilayah AWS tempat Origin Shield diaktifkan. Jika lokasi Origin Shield utama tidak tersedia dan CloudFront merutekan permintaan ke lokasi Origin Shield sekunder, pemicu Lambda @Edge yang menghadap ke asal juga akan bergeser untuk menggunakan lokasi Origin Shield sekunder.

Pemicu yang berhadapan dengan penampil tidak terpengaruh.

Optimalkan ketersediaan tinggi dengan failover CloudFront asal

Anda dapat mengatur CloudFront dengan failover asal untuk skenario yang memerlukan ketersediaan tinggi. Untuk memulai, Anda membuat grup asal dengan dua asal usul: primer dan sekunder. Jika asal primer tidak tersedia, atau mengembalikan kode status respons HTTP tertentu yang menunjukkan kegagalan, CloudFront secara otomatis beralih ke asal sekunder.

Untuk mengatur failover asal, Anda harus memiliki distribusi dengan setidaknya dua asal.

Selanjutnya, Anda membuat grup asal untuk distribusi Anda yang mencakup dua asal, menetapkan satu sebagai yang utama. Terakhir, Anda membuat atau memperbarui perilaku cache untuk menggunakan grup asal.

Untuk melihat langkah-langkah pengaturan grup asal dan konfigurasi opsi failover asal tertentu, lihat [Buat grup asal](#).

Setelah Anda mengonfigurasi failover asal untuk perilaku cache, CloudFront lakukan hal berikut untuk permintaan penampil:

- Ketika ada hit cache, CloudFront mengembalikan objek yang diminta.
- Saat ada cache yang hilang, CloudFront rutekan permintaan ke asal utama di grup asal.
- Ketika asal utama mengembalikan kode status yang tidak dikonfigurasi untuk failover, seperti kode status HTTP 2xx atau 3xx, CloudFront menyajikan objek yang diminta ke penampil.
- Jika terjadi hal-hal berikut:
 - Asal utama mengembalikan kode status HTTP yang telah Anda konfigurasikan untuk failover
 - CloudFront gagal terhubung ke asal utama (ketika 503 disetel sebagai kode failover)
 - Respons dari asal utama memakan waktu terlalu lama (waktu habis) (ketika 504 disetel sebagai kode failover)

Kemudian CloudFront rute permintaan ke asal sekunder di grup asal.

Note

Untuk beberapa kasus penggunaan, seperti streaming konten video, Anda mungkin CloudFront ingin gagal ke asal sekunder dengan cepat. Untuk menyesuaikan seberapa cepat CloudFront gagal ke asal sekunder, lihat [Kontrol batas waktu dan upaya asal](#).

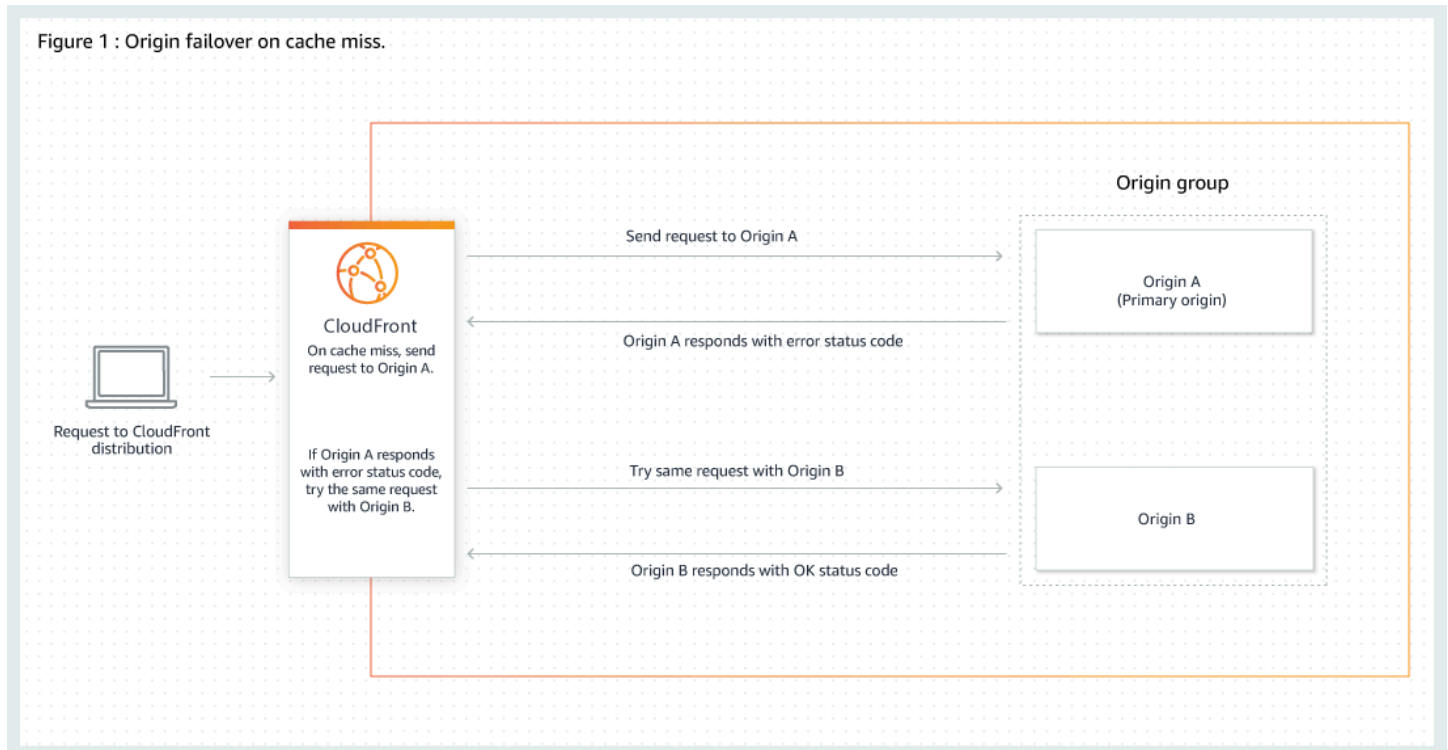
CloudFront merutekan semua permintaan yang masuk ke asal primer, bahkan ketika permintaan sebelumnya gagal ke asal sekunder. CloudFront hanya mengirim permintaan ke asal sekunder setelah permintaan ke asal primer gagal.

CloudFront gagal ke asal sekunder hanya jika metode HTTP dari permintaan penampil adalah GET, HEAD, atau OPTIONS. CloudFront tidak gagal ketika penampil mengirim metode HTTP yang berbeda (misalnya POST, PUT, dan sebagainya).

Note

CloudFront tidak akan failover jika tidak OPTIONS disetel sebagai [Metode HTTP cache](#) perilaku cache Anda.

Diagram berikut menggambarkan cara kerja failover asal.

**Topik**

- [Buat grup asal](#)
- [Kontrol batas waktu dan upaya asal](#)
- [Gunakan failover asal dengan fungsi Lambda@Edge](#)
- [Gunakan halaman kesalahan kustom dengan failover asal](#)

Buat grup asal

Untuk membuat grup asal

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.

2. Pilih distribusi yang ingin Anda buat sebagai grup asal.
3. Pilih tab Origins.
4. Pastikan distribusi memiliki lebih dari satu asal. Jika tidak, tambahkan asal kedua.
5. Pada tab Origins, di panel Grup asal, pilih Buat grup asal.
6. Pilih asal untuk grup asal. Setelah Anda menambahkan asal, gunakan panah untuk menetapkan prioritas—yaitu, asal mana yang utama dan yang kedua.
7. Masukkan nama untuk grup asal.
8. Pilih kode status HTTP untuk digunakan sebagai kriteria failover. Anda dapat memilih kombinasi kode status berikut: 400, 403, 404, 416, 429, 500, 502, 503, atau 504. Ketika CloudFront menerima respons dengan salah satu kode status yang Anda tentukan, itu gagal ke asal sekunder.

 Note

CloudFront gagal ke asal sekunder hanya jika metode HTTP dari permintaan penampil adalah GET, HEAD, atau OPTIONS. CloudFront tidak gagal ketika penampil mengirim metode HTTP yang berbeda (misalnya POST, PUT, dan sebagainya).

9. Di bawah Kriteria pemilihan Asal, tentukan bagaimana asal Anda dipilih saat distribusi Anda merutekan permintaan penampil. Anda dapat memilih opsi berikut.

Default

CloudFront akan menggunakan prioritas asal default yang Anda tentukan di halaman Pengaturan.

Skor kualitas media

CloudFront melacak dan menggunakan skor ini untuk menentukan asal pertama untuk meneruskan permintaan ke. Ini juga mengizinkan CloudFront untuk membuat HEAD permintaan asinkron ke asal alternatif di grup asal untuk menentukan skor kualitas medianya. Anda hanya dapat memilih opsi ini untuk asal AWS Elemental MediaPackage v2. Untuk informasi selengkapnya, lihat [Ketahanan sadar kualitas media](#).

10. Pilih Buat grup asal.

Pastikan untuk menetapkan grup asal Anda sebagai asal untuk perilaku cache distribusi Anda. Untuk informasi selengkapnya, lihat [Nama](#).

Kontrol batas waktu dan upaya asal

Secara default, CloudFront mencoba untuk terhubung ke asal utama dalam grup asal selama 30 detik (3 upaya koneksi masing-masing 10 detik) sebelum gagal ke asal sekunder. Untuk beberapa kasus penggunaan, seperti streaming konten video, Anda mungkin CloudFront ingin gagal ke asal sekunder lebih cepat. Anda dapat menyesuaikan pengaturan berikut untuk memengaruhi seberapa cepat CloudFront gagal ke asal sekunder. Jika asal adalah asal sekunder, atau asal yang bukan bagian dari grup asal, pengaturan ini memengaruhi seberapa cepat CloudFront mengembalikan respons HTTP 504 ke penampil.

Untuk gagal dengan lebih cepat, tentukan waktu koneksi yang lebih singkat, lebih sedikit upaya koneksi, atau keduanya. Untuk asal kustom (termasuk asal bucket Amazon S3 yang adalah dikonfigurasi dengan hosting situs web statis), Anda juga dapat menyesuaikan waktu habis respons asal.

Waktu habis koneksi asal

Pengaturan batas waktu koneksi asal memengaruhi berapa lama CloudFront menunggu ketika mencoba membuat koneksi ke asal. Secara default, CloudFront tunggu 10 detik untuk membuat koneksi, tetapi Anda dapat menentukan 1-10 detik (inklusif). Untuk informasi selengkapnya, lihat [Batas waktu koneksi](#).

Upaya koneksi asal

Pengaturan upaya koneksi asal mempengaruhi berapa kali CloudFront upaya untuk terhubung ke asal. Secara default, CloudFront coba 3 kali untuk terhubung, tetapi Anda dapat menentukan 1-3 (inklusif). Untuk informasi selengkapnya, lihat [Upaya koneksi](#).

Untuk custom origin (termasuk bucket Amazon S3 yang dikonfigurasi dengan hosting situs web statis), pengaturan ini juga memengaruhi berapa kali CloudFront upaya mendapatkan respons dari asal jika terjadi batas waktu respons asal.

Waktu habis untuk respons asal

Batas waktu respons asal, juga dikenal sebagai batas waktu baca asal, memengaruhi berapa lama CloudFront menunggu untuk menerima respons (atau untuk menerima respons lengkap) dari asal. Secara default, CloudFront menunggu selama 30 detik, tetapi Anda dapat menentukan 1-120 detik (inklusif). Untuk informasi selengkapnya, lihat [Batas waktu respons](#).

Cara mengubah pengaturan ini

Untuk mengubah pengaturan ini di [CloudFrontkonsol](#)

- Untuk asal baru atau distribusi baru, Anda menentukan nilai ini saat membuat sumber daya.
- Untuk asal yang sudah ada dalam distribusi yang sudah ada, Anda menentukan nilai ini saat mengedit asal.

Untuk informasi selengkapnya, lihat [Semua referensi pengaturan distribusi](#).

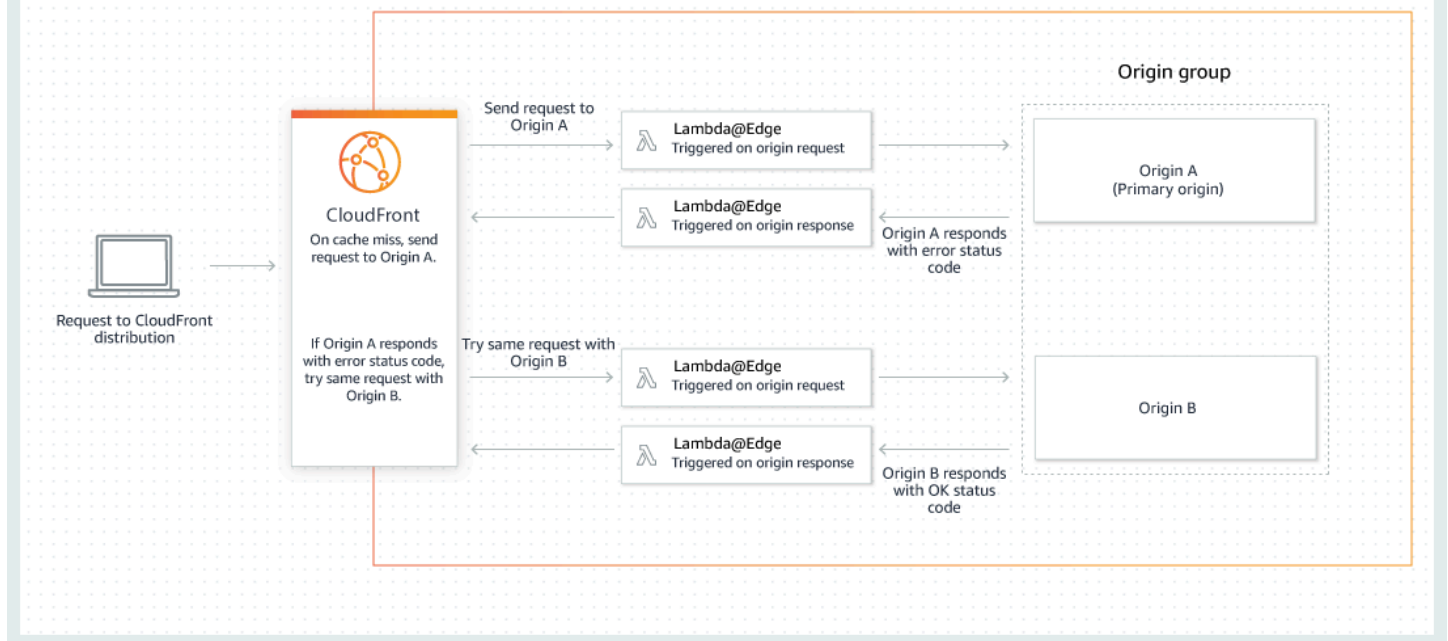
Gunakan failover asal dengan fungsi Lambda@Edge

Anda dapat menggunakan fungsi Lambda @Edge dengan CloudFront distribusi yang telah Anda atur dengan grup asal. Untuk menggunakan fungsi Lambda, tentukan di [permintaan asal usul atau pemicu respons asal](#) untuk grup asal ketika Anda membuat perilaku cache. Saat Anda menggunakan fungsi Lambda@Edge dengan grup asal, fungsi ini dapat dipicu dua kali untuk permintaan penampil tunggal. Misalnya, pertimbangkan skenario ini:

1. Anda membuat fungsi Lambda@Edge dengan pemicu permintaan asal.
2. Fungsi Lambda dipicu sekali saat CloudFront mengirim permintaan ke asal utama (pada cache yang hilang).
3. Asal utama merespons dengan kode status HTTP yang dikonfigurasi untuk failover.
4. Fungsi Lambda dipicu lagi saat CloudFront mengirim permintaan yang sama ke asal sekunder.

Diagram berikut menggambarkan cara kerja asal-usul saat Anda menyertakan fungsi Lambda@Edge dalam permintaan asal usul atau pemicu respons.

Figure 2 : Origin failover with Lambda@Edge functions triggered on origin request and response events.



Untuk informasi lebih lanjut tentang menggunakan pemacu Lambda@Edge, lihat [the section called “Tambahkan pemacu untuk fungsi Lambda @Edge”](#).

Untuk informasi selengkapnya tentang mengelola failover DNS, lihat [Mengonfigurasi failover DNS](#) di Panduan Pengembang Amazon Route 53.

Gunakan halaman kesalahan kustom dengan failover asal

Anda dapat menggunakan halaman kesalahan kustom dengan grup asal yang serupa dengan cara Anda menggunakannya dengan asal yang tidak disiapkan untuk failover asal.

Saat Anda menggunakan failover asal, Anda dapat mengonfigurasi CloudFront untuk mengembalikan halaman kesalahan kustom untuk asal primer atau sekunder (atau keduanya):

- Mengembalikan halaman kesalahan kustom untuk asal utama — Jika asal utama mengembalikan kode status HTTP yang tidak dikonfigurasi untuk failover, CloudFront mengembalikan halaman kesalahan kustom ke pemirsa.
- Mengembalikan halaman kesalahan kustom untuk asal sekunder - Jika CloudFront menerima kode status kegagalan dari asal sekunder, CloudFront mengembalikan halaman kesalahan kustom.

Untuk informasi selengkapnya tentang menggunakan halaman kesalahan kustom dengan CloudFront, lihat [Hasilkan respons kesalahan khusus](#).

Mengelola berapa lama konten tetap dalam cache (kedaluwarsa)

Anda dapat mengontrol berapa lama file Anda berada dalam CloudFront cache sebelum CloudFront meneruskan permintaan lain ke asal Anda. Mengurangi durasi memungkinkan Anda untuk melayani konten dinamis. Peningkatan durasi berarti bahwa pengguna Anda mendapatkan kinerja yang lebih baik karena file Anda lebih mungkin dilayani secara langsung dari edge cache. Durasi yang lebih lama juga mengurangi beban yang berasal dari Anda.

Biasanya, CloudFront menyajikan file dari lokasi tepi hingga durasi cache yang Anda tentukan lewat — yaitu, hingga file kedaluwarsa. Setelah kedaluwarsa, saat berikutnya lokasi tepi mendapat permintaan untuk file tersebut, CloudFront teruskan permintaan ke asal untuk memverifikasi bahwa cache berisi versi terbaru dari file tersebut. Tanggapan dari sumber tergantung pada apakah file telah berubah:

- Jika CloudFront cache sudah memiliki versi terbaru, asal mengembalikan kode status `304 Not Modified`.
- Jika CloudFront cache tidak memiliki versi terbaru, asal mengembalikan kode status `200 OK` dan versi terbaru file.

Jika file di lokasi tepi tidak sering diminta, CloudFront mungkin mengusir file — hapus file sebelum tanggal kedaluwarsa — untuk memberi ruang bagi file yang telah diminta baru-baru ini.

Sebaiknya mengelola durasi cache Anda dengan memperbarui kebijakan cache distribusi Anda. Jika Anda memilih untuk tidak menggunakan kebijakan cache, TTL default (Time to Live) adalah 24 jam, tetapi Anda dapat memperbarui pengaturan berikut untuk mengganti default:

- Untuk mengubah durasi cache untuk semua file yang cocok dengan pola jalur yang sama, Anda dapat mengubah CloudFront pengaturan untuk TTL Minimum, TTL Maksimum, dan TTL Default untuk perilaku cache. Untuk informasi tentang pengaturan individual, lihat [TTL Minimum](#), [TTL Maksimum](#), dan [TTL bawaan](#).
- Untuk mengubah durasi cache untuk file individual, Anda dapat mengonfigurasi asal Anda untuk menambahkan Cache-Control header dengan max-age atau s-maxage direktif, atau Expires header ke file. Untuk informasi selengkapnya, lihat [Gunakan header untuk mengontrol durasi cache untuk masing-masing objek](#).

Untuk informasi selengkapnya tentang bagaimana TTL Minimum, TTL Default, dan TTL Maksimum berinteraksi dengan s-maxage arahan max-age dan bidang Expires header, lihat. [the section called “Tentukan jumlah waktu yang menyimpan objek dalam CloudFront cache”](#)

Anda juga dapat mengontrol berapa lama kesalahan (misalnya, 404 Not Found) tinggal di CloudFront cache sebelum CloudFront mencoba lagi untuk mendapatkan objek yang diminta dengan meneruskan permintaan lain ke asal Anda. Untuk informasi selengkapnya, lihat [the section called “Bagaimana CloudFront memproses kode status HTTP 4xx dan 5xx dari asal Anda”](#).

Topik

- [Gunakan header untuk mengontrol durasi cache untuk masing-masing objek](#)
- [Sajikan konten basi \(kedaluwarsa\)](#)
- [Tentukan jumlah waktu yang menyimpan objek dalam CloudFront cache](#)
- [Tambahkan header ke objek Anda dengan menggunakan konsol Amazon S3](#)

Gunakan header untuk mengontrol durasi cache untuk masing-masing objek

Anda dapat menggunakan Cache-Control dan Expires header untuk mengontrol berapa lama objek tetap berada di dalam cache. Pengaturan untuk TTL Minimum, TTL bawaan, dan TTL Maksimum juga memengaruhi durasi cache, tetapi berikut ini gambaran umum tentang bagaimana header dapat memengaruhi durasi cache:

- Cache-Control max-ageDirektif memungkinkan Anda menentukan berapa lama (dalam detik) bahwa Anda ingin objek tetap berada di cache sebelum CloudFront mendapatkan objek lagi dari server asal. CloudFrontDukungan waktu kedaluwarsa minimum adalah 0 detik. Nilai maksimumnya adalah 100 tahun. Tentukan nilai dalam format berikut:

Cache-Control: max-age=*seconds*

Misalnya, arahan berikut memberitahu CloudFront untuk menyimpan objek terkait dalam cache selama 3600 detik (satu jam):

Cache-Control: max-age=3600

Jika Anda ingin objek tetap berada di cache CloudFront tepi untuk durasi yang berbeda dari yang berada di cache browser, Anda dapat menggunakan Cache-Control s-maxage arahan Cache-

Control max-age dan bersama-sama. Untuk informasi selengkapnya, lihat [Tentukan jumlah waktu yang menyimpan objek dalam CloudFront cache](#).

- Expires kolom header memungkinkan Anda menentukan tanggal dan waktu kedaluwarsa menggunakan format yang ditentukan dalam [RFC 2616, Protokol Transfer Hiperteks -- HTTP/1.1 Bagian 3.3.1, Tanggal Penuh](#), misalnya:

Sat, 27 Jun 2015 23:59:59 GMT

Kami sarankan Anda menggunakan Cache-Control max-age lebih langsung, bukan Expires kolom header untuk mengontrol caching objek. Jika Anda menentukan nilai baik untuk Cache-Control max-age dan untuk Expires, hanya CloudFront menggunakan nilai Cache-Control max-age.

Untuk informasi selengkapnya, lihat [Tentukan jumlah waktu yang menyimpan objek dalam CloudFront cache](#).

Anda tidak dapat menggunakan bidang HTTP Cache-Control atau Pragma header dalam GET permintaan dari penampil CloudFront untuk memaksa kembali ke server asal untuk objek tersebut. CloudFront mengabaikan bidang header tersebut dalam permintaan penampil.

Untuk informasi lebih lanjut tentang Cache-Control dan Expires kolom header, lihat bagian berikut di RFC 2616, Protokol Transfer Hiperteks -- HTTP/1.1:

- [Bagian 14.9 Kontrol Cache](#)
- [Bagian 14.21 Kedaluwarsa](#)

Sajikan konten basi (kedaluwarsa)

CloudFront mendukung arahan kontrol Stale-While-Revalidate dan Stale-If-Error cache. Anda dapat menggunakan arahan ini untuk menentukan berapa lama konten basi tersedia untuk pemirsa.

Topik

- [Stale-While-Revalidate](#)
- [Stale-If-Error](#)
- [Gunakan kedua arahan](#)

Stale-While-Revalidate

Arahan ini memungkinkan CloudFront untuk menyajikan konten basi dari cache sementara secara CloudFront asinkron mengambil versi baru dari asal. Ini meningkatkan latensi karena pemirsa menerima tanggapan langsung dari lokasi tepi tanpa harus menunggu pengambilan latar belakang. Konten segar dimuat di latar belakang untuk permintaan future.

Example Contoh: **Stale-While-Revalidate**

CloudFront melakukan hal berikut ketika Anda mengatur Cache-Control header untuk menggunakan arahan ini.

```
Cache-Control: max-age=3600, stale-while-revalidate=600
```

1. CloudFront akan menyimpan respons selama satu jam (max-age=3600).
2. Jika permintaan dibuat setelah durasi ini, CloudFront menyajikan konten basi, sementara secara bersamaan mengirim permintaan ke asal untuk memvalidasi ulang dan menyegarkan konten yang di-cache.
3. Saat konten sedang divalidasi ulang, CloudFront menyajikan konten basi hingga 10 menit ().
stale-while-revalidate=600

Note

CloudFront akan menyajikan konten basi hingga nilai stale-while-revalidate direktif atau nilai [TTL CloudFront maksimum](#), mana yang kurang. Setelah durasi TTL maksimum, objek basi tidak akan tersedia dari cache tepi, terlepas dari nilainya. stale-while-revalidate

Stale-If-Error

Arahan ini memungkinkan CloudFront untuk menyajikan konten basi dari cache jika asal tidak dapat dijangkau atau mengembalikan kode kesalahan antara 500 dan 600. Ini memastikan bahwa pemirsa dapat mengakses konten bahkan selama pemadaman asal.

Example Contoh: **Stale-If-Error**

CloudFront melakukan hal berikut ketika Anda mengatur Cache-Control header untuk menggunakan arahan ini.

```
Cache-Control: max-age=3600, stale-if-error=86400
```

1. CloudFront cache respons selama satu jam (max-age=3600).
2. Jika asal tidak aktif atau mengembalikan kesalahan setelah durasi ini, CloudFront terus menyajikan konten basi hingga 24 jam () stale-if-error=86400
3. Jika Anda mengonfigurasi respons kesalahan kustom, CloudFront akan mencoba menyajikan konten basi jika terjadi kesalahan dalam stale-if-error durasi yang ditentukan. Jika konten basi tidak tersedia, maka CloudFront akan menyajikan respons kesalahan kustom yang Anda konfigurasi untuk kode status kesalahan yang sesuai. Untuk informasi selengkapnya, lihat [Hasilkan respons kesalahan khusus](#).

Catatan

- CloudFront akan menyajikan konten basi hingga nilai stale-if-error direktif atau nilai [TTL CloudFront maksimum](#), mana yang kurang. Setelah durasi TTL maksimum, objek basi tidak akan tersedia dari cache tepi, terlepas dari nilainya. stale-if-error
- Jika Anda tidak mengonfigurasi stale-if-error atau respons kesalahan kustom, CloudFront akan mengembalikan objek basi atau meneruskan respons kesalahan kembali ke penampil, tergantung pada apakah objek yang diminta ada di cache tepi atau tidak. Untuk informasi selengkapnya, lihat [Cara CloudFront memproses kesalahan jika Anda belum mengonfigurasi halaman kesalahan kustom](#).

Gunakan kedua arahan

Keduanya stale-while-revalidate dan stale-if-error merupakan arahan kontrol cache independen yang dapat Anda gunakan bersama untuk mengurangi latensi dan menambahkan buffer agar asal Anda merespons atau memulihkan.

Example Contoh: Menggunakan kedua arahan

CloudFront melakukan hal berikut ketika Anda mengatur Cache-Control header untuk menggunakan arahan berikut.

```
Cache-Control: max-age=3600, stale-while-revalidate=600, stale-if-error=86400
```

1. CloudFront cache respons selama satu jam (max-age=3600).
2. Jika permintaan dibuat setelah durasi ini, CloudFront menyajikan konten basi hingga 10 menit (stale-while-revalidate=600) saat konten sedang divalidasi ulang.
3. Jika server asal mengembalikan kesalahan saat CloudFront mencoba memvalidasi ulang konten, CloudFront akan terus menyajikan konten basi hingga 24 jam (). stale-if-error=86400

Caching adalah keseimbangan antara kinerja dan kesegaran. Menggunakan arahan seperti stale-while-revalidate dan stale-if-error dapat meningkatkan kinerja dan pengalaman pengguna, tetapi pastikan konfigurasi selaras dengan seberapa segar konten yang Anda inginkan. Arahan konten basi paling cocok untuk kasus penggunaan di mana konten perlu disegarkan tetapi memiliki versi terbaru tidak penting. Selain itu, jika konten Anda tidak berubah atau jarang berubah, stale-while-revalidate dapat menambahkan permintaan jaringan yang tidak perlu. Sebagai gantinya, pertimbangkan untuk mengatur durasi cache yang panjang.

Tentukan jumlah waktu yang menyimpan objek dalam CloudFront cache

Untuk mengontrol jumlah waktu yang CloudFront menyimpan objek dalam cache sebelum mengirim permintaan lain ke asal, Anda dapat:

- Tetapkan nilai TTL minimum, maksimum, dan default dalam perilaku cache CloudFront distribusi. Anda dapat mengatur nilai-nilai ini dalam [kebijakan cache](#) yang melekat pada perilaku cache (disarankan), atau dalam pengaturan cache warisan.
- Sertakan Cache-Control atau Expires header dalam tanggapan dari asal. Header ini juga membantu menentukan berapa lama browser menyimpan objek di cache browser sebelum mengirim permintaan lain. CloudFront

Tabel berikut menjelaskan bagaimana header Cache-Control dan Expires yang dikirim dari asal digunakan bersama dengan pengaturan TTL dalam perilaku cache untuk memengaruhi caching.

Header asal	TTL minimum = 0	TTL Minimum > 0
Asal menambahkan Cache-Control: max-age direktif ke objek	CloudFront caching CloudFront cache objek untuk yang lebih rendah dari nilai <code>Cache-Control: max-age</code> direktif atau nilai TTL maksimum. CloudFront Caching browser Browser menyimpan cache objek untuk nilai arahan <code>Cache-Control: max-age</code>.	CloudFront caching CloudFront caching tergantung pada nilai TTL CloudFront minimum dan TTL maksimum dan arahan: <code>Cache-Control max-age</code> • Jika minimum TTL < max-age < maksimum TTL, maka CloudFront cache objek untuk nilai direktif. <code>Cache-Control: max-age</code> • Jika max-age < minimum TTL, maka CloudFront cache objek untuk nilai TTL CloudFront minimum. • Jika max-age > TTL maksimum, maka CloudFront cache objek untuk nilai TTL CloudFront maksimum. Caching browser Browser menyimpan cache objek untuk nilai arahan <code>Cache-Control: max-age</code>.

Header asal	TTL minimum = 0	TTL Minimum > 0
Asal tidak menambahkan Cache-Control: max-age direktif ke objek	CloudFront caching CloudFront cache objek untuk nilai TTL CloudFront default. Caching browser Tergantung pada peramban.	CloudFront caching CloudFront cache objek untuk nilai TTL CloudFront minimum atau TTL default yang lebih besar. Caching browser Tergantung pada peramban.

Header asal	TTL minimum = 0	TTL Minimum > 0
Asal menambahkan Cache-Control: max-age dan Cache-Control: s-maxage mengarahkan ke objek	CloudFront caching CloudFront cache objek untuk yang lebih rendah dari nilai Cache-Control: s-maxage direktif atau nilai TTL maksimum. CloudFront Caching browser Browser menyimpan cache objek untuk nilai arahan Cache-Control max-age.	CloudFront caching CloudFront caching tergantung pada nilai TTL CloudFront minimum dan TTL maksimum dan arahan: Cache-Control: s-maxage • Jika minimum TTL < s-maxage < maksimum TTL, maka CloudFront cache objek untuk nilai direktif. Cache-Control: s-maxage • Jika s-maxage < minimum TTL, maka CloudFront cache objek untuk nilai TTL CloudFront minimum. • Jika s-maxage > TTL maksimum, maka CloudFront cache objek untuk nilai TTL CloudFront maksimum. Caching browser Browser menyimpan cache objek untuk nilai arahan Cache-Control: max-age.

Header asal	TTL minimum = 0	TTL Minimum > 0
Asal menambahkan Expires header ke objek	CloudFront caching CloudFront cache objek sampai tanggal di Expires header atau untuk nilai TTL CloudFront maksimum, mana yang lebih cepat. Caching browser Browser menyimpan cache objek hingga tanggal di header Expires.	CloudFront caching CloudFront caching tergantung pada nilai TTL CloudFront minimum dan TTL maksimum dan header: Expires • Jika minimum TTL < Expires < maksimum TTL, maka CloudFront cache objek sampai tanggal dan waktu di header. Expires • Jika Expires < minimum TTL, maka CloudFront cache objek untuk nilai TTL CloudFront minimum. • Jika Expires > TTL maksimum, maka CloudFront cache objek untuk nilai TTL CloudFront maksimum. Caching browser Browser menyimpan cache objek hingga tanggal dan waktu di header Expires.

Header asal	TTL minimum = 0	TTL Minimum > 0
Origin menambahkan Cache-Control: no-cache, no-store,, dan/atau private arahan ke objek	CloudFront dan browser menghormati header.	CloudFront caching CloudFront cache objek untuk nilai TTL CloudFront minimum. Lihat peringatan di bawah tabel ini. Caching browser Browser mematuhi header.

Warning

- Jika TTL minimum Anda lebih besar dari 0, CloudFront gunakan TTL minimum kebijakan cache, meskipun and/or private arahan `Cache-Control: no-cache, no-store,` ada di header asal.
- Jika asal dapat dijangkau, CloudFront dapatkan objek dari asal dan kembalikan ke penampil.
- Jika asal tidak dapat dijangkau dan nilai TTL minimum atau maksimum lebih besar dari 0, CloudFront akan melayani objek yang didapat dari asal sebelumnya.

Untuk menghindari perilaku ini, sertakan arahan `Cache-Control: stale-if-error=0` dengan objek yang dikembalikan dari asal. Hal ini menyebabkan CloudFront untuk mengembalikan kesalahan dalam menanggapi permintaan future jika asal tidak dapat dijangkau, daripada mengembalikan objek yang didapatnya dari asal sebelumnya.

- CloudFront tidak menyimpan kode status HTTP 501 (Tidak Diimplementasikan) dari asal S3 saat header asal menyertakan arahan `Cache-Control: no-cache,,no-store. and/or private` Ini adalah perilaku default untuk asal S3, bahkan jika pengaturan [TTL minimum](#) Anda lebih besar dari 0.

Untuk informasi tentang cara mengubah pengaturan untuk distribusi menggunakan CloudFront konsol, lihat [Perbarui distribusi](#). Untuk informasi tentang cara mengubah setelan distribusi menggunakan CloudFront API, lihat [UpdateDistribution](#).

Tambahkan header ke objek Anda dengan menggunakan konsol Amazon S3

Anda dapat menambahkan bidang **Cache-Control** atau **Expires** header ke objek Amazon S3 Anda. Untuk melakukannya, Anda memodifikasi bidang metadata untuk objek.

Untuk menambahkan bidang **Cache-Control** atau **Expires** header ke objek Amazon S3

1. Ikuti prosedur di bagian Mengganti metadata yang ditentukan sistem pada metadata [objek pengeditan dalam topik konsol Amazon S3 di Panduan Pengguna Amazon S3](#).
2. Untuk Kunci, pilih nama header yang Anda tambahkan (Kontrol Cache atau Kedaluwarsa).
3. Untuk Nilai, masukkan nilai header. Misalnya, untuk header **Cache-Control**, Anda bisa memasukkan `max-age=86400`. Untuk **Expires**, Anda dapat memasukkan tanggal kedaluwarsa dan waktu seperti `Wed, 30 Jun 2021 09:28:00 GMT`.
4. Ikuti prosedur lainnya untuk menyimpan perubahan metadata Anda.

Konten cache berdasarkan parameter string kueri

Beberapa aplikasi web menggunakan string pencarian untuk mengirimkan informasi ke sumber. String kueri adalah bagian dari permintaan web yang muncul setelah `?` karakter; string dapat berisi satu atau lebih parameter, dipisahkan oleh `&` karakter. Dalam contoh berikut, string query mencakup dua parameter, `color=red` dan `size=large`:

`https://d1111111abcdef8.cloudfront.net/images/image.jpg?color=red&size=large`

Untuk distribusi, Anda dapat memilih apakah Anda CloudFront ingin meneruskan string kueri ke asal Anda dan apakah akan menyimpan konten Anda berdasarkan semua parameter atau parameter yang dipilih. Mengapa hal ini mungkin berguna? Pertimbangkan contoh berikut.

Misalkan situs web Anda tersedia dalam lima bahasa. Struktur direktori dan nama file untuk kelima versi situs web ini adalah identik. Saat pengguna melihat situs web Anda, permintaan yang diteruskan untuk CloudFront menyertakan parameter string kueri bahasa berdasarkan bahasa yang dipilih pengguna. Anda dapat mengonfigurasi CloudFront untuk meneruskan string kueri ke

asal dan cache berdasarkan parameter bahasa. Jika Anda mengonfigurasi server web Anda untuk mengembalikan versi halaman tertentu yang sesuai dengan bahasa yang dipilih, CloudFront cache setiap versi bahasa secara terpisah, berdasarkan nilai parameter string kueri bahasa.

Dalam contoh ini, jika halaman utama untuk situs web `Andamain.html`, lima permintaan berikut CloudFront menyebabkan cache `main.html` lima kali, sekali untuk setiap nilai parameter string kueri bahasa:

- `https://d111111abcdef8.cloudfront.net/main.html?language=de`
- `https://d111111abcdef8.cloudfront.net/main.html?language=en`
- `https://d111111abcdef8.cloudfront.net/main.html?language=es`
- `https://d111111abcdef8.cloudfront.net/main.html?language=fr`
- `https://d111111abcdef8.cloudfront.net/main.html?language=jp`

Perhatikan hal-hal berikut:

- Beberapa server HTTP tidak memproses parameter string kueri dan, oleh karena itu, tidak mengembalikan versi objek yang berbeda berdasarkan nilai parameter. Untuk asal-usul ini, jika Anda mengonfigurasi CloudFront untuk meneruskan parameter string kueri ke asal, CloudFront tetap cache berdasarkan nilai parameter meskipun asal mengembalikan versi objek yang identik CloudFront untuk setiap nilai parameter.
- Untuk parameter string pencarian agar berfungsi seperti yang dijelaskan dalam contoh di atas dengan bahasa, Anda harus menggunakan `&` karakter sebagai pembatas antara parameter string pencarian. Jika Anda menggunakan pembatas yang berbeda, Anda mungkin mendapatkan hasil yang tidak terduga, tergantung pada parameter mana yang Anda tentukan CloudFront untuk digunakan sebagai dasar untuk caching, dan urutan parameter yang muncul dalam string kueri.

Contoh berikut menunjukkan apa yang terjadi jika Anda menggunakan pembatas yang berbeda dan Anda mengonfigurasi CloudFront ke cache hanya berdasarkan parameter: `color`

- Dalam permintaan berikut, CloudFront cache konten Anda berdasarkan nilai `color` parameter, tetapi CloudFront menafsirkan nilai sebagai: `red;size=large`

```
https://d111111abcdef8.cloudfront.net/images/  
image.jpg?color=red;size=large
```

- Dalam permintaan berikut, CloudFront cache konten Anda tetapi tidak mendasarkan caching pada parameter string kueri. Ini karena Anda CloudFront mengonfigurasi cache berdasarkan

color parameter, tetapi CloudFront menafsirkan string berikut sebagai hanya berisi size parameter yang memiliki nilai *large; color=red*:

```
https://d1111111abcdef8.cloudfront.net/images/  
image.jpg?size=large; color=red
```

Anda dapat mengonfigurasi CloudFront untuk melakukan salah satu hal berikut:

- Jangan meneruskan string kueri ke asal sama sekali. Jika Anda tidak meneruskan string kueri, CloudFront tidak cache berdasarkan parameter string kueri.
- Teruskan string kueri ke asal, dan simpan berdasarkan semua parameter dalam string kueri.
- Teruskan string kueri ke asal, dan cache berdasarkan parameter yang ditentukan dalam string kueri.

Untuk informasi selengkapnya, lihat [the section called “Optimalkan caching”](#).

Topik

- [Pengaturan konsol dan API untuk penerusan string dan caching kueri](#)
- [Optimalkan caching](#)
- [Parameter string kueri dan log CloudFront standar \(log akses\)](#)

Pengaturan konsol dan API untuk penerusan string dan caching kueri

Saat Anda membuat distribusi di CloudFront konsol, CloudFront mengonfigurasi penerusan string kueri dan caching untuk Anda berdasarkan tipe asal Anda. Secara opsional, Anda dapat mengedit pengaturan ini secara manual. Untuk informasi selengkapnya, lihat pengaturan berikut di [the section called “Semua pengaturan distribusi”](#):

- [the section called “Penerusan string kueri dan caching”](#)
- [the section called “Daftar izin string kueri”](#)

Untuk mengonfigurasi penerusan dan caching string kueri dengan CloudFront API, lihat [CachePolicy](#) dan di [OriginRequestPolicy](#) Referensi Amazon CloudFront API.

Optimalkan caching

Saat Anda CloudFront mengonfigurasi cache berdasarkan parameter string kueri, Anda dapat mengambil langkah-langkah berikut untuk mengurangi jumlah permintaan yang CloudFront diteruskan ke asal Anda. Saat lokasi CloudFront tepi menyajikan objek, Anda mengurangi beban di server asal dan mengurangi latensi karena objek dilayani dari lokasi yang lebih dekat dengan pengguna Anda.

Cache hanya berdasarkan parameter yang asal Anda mengembalikan versi objek yang berbeda

Untuk setiap parameter string kueri yang diteruskan aplikasi web Anda CloudFront, CloudFront teruskan permintaan ke asal Anda untuk setiap nilai parameter dan cache versi terpisah dari objek untuk setiap nilai parameter. Hal ini berlaku bahkan jika asal Anda selalu mengembalikan objek yang sama terlepas dari nilai parameter. Untuk parameter multipel, jumlah permintaan dan jumlah objek berlipat ganda.

Sebaiknya Anda CloudFront mengonfigurasi cache hanya berdasarkan parameter string kueri yang asal Anda mengembalikan versi yang berbeda, dan Anda mempertimbangkan dengan cermat manfaat caching berdasarkan setiap parameter. Misalnya, Anda memiliki situs web ritel. Ada gambar jaket dengan enam warna berbeda, dan jaket ini tersedia dalam 10 ukuran berbeda. Gambar yang Anda miliki pada jaket menunjukkan warna yang berbeda tetapi tidak berbeda ukuran. Untuk mengoptimalkan caching, Anda harus CloudFront mengkonfigurasi cache hanya berdasarkan parameter warna, bukan pada parameter ukuran. Ini meningkatkan kemungkinan yang CloudFront dapat melayani permintaan dari cache, yang meningkatkan kinerja dan mengurangi beban pada asal Anda.

Selalu daftar parameter dalam urutan yang sama

Urutan parameter penting dalam string kueri. Dalam contoh berikut, string kueri identik kecuali bahwa parameter berada dalam urutan yang berbeda. Hal ini menyebabkan CloudFront untuk meneruskan dua permintaan terpisah untuk `image.jpg` ke asal Anda dan untuk cache dua versi terpisah dari objek:

- `https://d111111abcdef8.cloudfront.net/images/image.jpg?color=red&size=large`
- `https://d111111abcdef8.cloudfront.net/images/image.jpg?size=large&color=red`

Kami menyarankan Anda untuk selalu mencantumkan nama parameter dalam urutan yang sama, seperti urutan abjad.

Selalu gunakan kasus yang sama untuk nama dan nilai parameter

CloudFront mempertimbangkan kasus nama parameter dan nilai saat caching berdasarkan parameter string kueri. Dalam contoh berikut, string pencarian identik kecuali untuk kasus nama dan nilai parameter. Hal ini menyebabkan CloudFront untuk meneruskan empat permintaan terpisah untuk image.jpg ke asal Anda dan men-cache empat versi terpisah dari objek:

- `https://d111111abcdef8.cloudfront.net/images/image.jpg?color=red`
- `https://d111111abcdef8.cloudfront.net/images/image.jpg?color=Red`
- `https://d111111abcdef8.cloudfront.net/images/image.jpg?Color=red`
- `https://d111111abcdef8.cloudfront.net/images/image.jpg?Color=Red`

Kami menyarankan agar Anda menggunakan kasus secara konsisten untuk nama dan nilai parameter, seperti semua huruf kecil.

Jangan gunakan nama parameter yang bertentangan dengan tanda tangan URLs

Jika Anda menggunakan tanda tangan URLs untuk membatasi akses ke konten Anda (jika Anda menambahkan tanda tangan tepercaya ke distribusi Anda), CloudFront hapus parameter string kueri berikut sebelum meneruskan sisa URL ke asal Anda:

- Expires
- Key-Pair-Id
- Policy
- Signature

Jika Anda menggunakan tanda tangan URLs dan ingin mengonfigurasi CloudFront untuk meneruskan string kueri ke asal Anda, parameter string kueri Anda sendiri tidak dapat diberi namaExpires,, Key-Pair-IdPolicy, atauSignature.

Parameter string kueri dan log CloudFront standar (log akses)

Jika Anda mengaktifkan logging, CloudFront mencatat URL lengkap, termasuk parameter string kueri. Ini benar terlepas dari apakah Anda telah mengonfigurasi CloudFront untuk meneruskan string kueri ke asal. Untuk informasi selengkapnya tentang CloudFront pencatatan, lihat [the section called “Akses log \(log standar\)”](#).

Konten cache berdasarkan cookie

Secara default, CloudFront tidak mempertimbangkan cookie saat memproses permintaan dan tanggapan, atau saat menyimpan objek Anda di lokasi tepi. Jika CloudFront menerima dua permintaan yang identik kecuali untuk apa yang ada di Cookie header, maka, secara default, CloudFront memperlakukan permintaan sebagai identik dan mengembalikan objek yang sama untuk kedua permintaan.

Anda dapat mengonfigurasi CloudFront untuk meneruskan ke asal Anda beberapa atau semua cookie dalam permintaan penampil, dan untuk menyimpan versi terpisah dari objek Anda berdasarkan nilai cookie yang diteruskannya. Saat Anda melakukan ini, CloudFront gunakan beberapa atau semua cookie dalam permintaan penampil — mana pun yang dikonfigurasi untuk diteruskan—untuk mengidentifikasi objek dalam cache secara unik.

Sebagai contoh, anggaplah bahwa permintaan untuk `locations.html` berisi sebuah cookie `country` yang memiliki nilai `uk` atau `fr`. Saat Anda mengonfigurasi CloudFront untuk menyimpan objek Anda berdasarkan nilai `country` cookie, CloudFront teruskan permintaan `locations.html` ke asal dan sertakan `country` cookie dan nilainya. Asal Anda kembali `locations.html`, dan CloudFront menyimpan objek satu kali untuk permintaan di mana nilai `country` cookie berada `uk` dan sekali untuk permintaan di mana nilainya `fr`.

Important

Amazon S3 dan beberapa server HTTP tidak memproses cookie. Jangan mengonfigurasi CloudFront untuk meneruskan cookie ke asal yang tidak memproses cookie atau tidak mengubah responsnya berdasarkan cookie. Itu dapat menyebabkan CloudFront untuk meneruskan lebih banyak permintaan ke asal untuk objek yang sama, yang memperlambat kinerja dan meningkatkan beban pada asal. Jika, mengingat contoh sebelumnya, asal Anda tidak memproses `country` cookie atau selalu mengembalikan versi yang sama dari `locations.html` ke CloudFront terlepas dari nilai `country` cookie, jangan konfigurasi CloudFront untuk meneruskan cookie itu.

Sebaliknya, jika asal kustom Anda bergantung pada cookie tertentu atau mengirimkan tanggapan berbeda berdasarkan cookie, pastikan Anda mengonfigurasi CloudFront untuk meneruskan cookie tersebut ke asal. Jika tidak, CloudFront hapus cookie sebelum meneruskan permintaan ke asal Anda.

Untuk mengonfigurasi penerusan cookie, Anda memperbarui perilaku cache distribusi. Untuk informasi lebih lanjut tentang perilaku cache, lihat [Pengaturan perilaku cache](#), terutama [Teruskan cookie](#) dan [Daftar cookie yang diizinkan](#) bagian.

Anda dapat mengonfigurasi setiap perilaku cache untuk melakukan salah satu hal berikut:

- Teruskan semua cookie ke asal Anda — CloudFront termasuk semua cookie yang dikirim oleh pemirsa saat meneruskan permintaan ke asal. Saat asal Anda mengembalikan respons, CloudFront cache respons menggunakan nama dan nilai cookie dalam permintaan penampil. Jika respons asal menyertakan Set-Cookie header, CloudFront mengembalikannya ke penampil dengan objek yang diminta. CloudFront juga menyimpan Set-Cookie header dengan objek yang dikembalikan dari asal, dan mengirimkan Set-Cookie header tersebut ke pemirsa di semua klik cache.
- Teruskan satu set cookie yang Anda tentukan — CloudFront menghapus cookie apa pun yang dikirim penampil yang tidak ada dalam daftar yang diizinkan sebelum meneruskan permintaan ke asal. CloudFront cache respons menggunakan nama dan nilai cookie yang tercantum dalam permintaan penampil. Jika respons asal menyertakan Set-Cookie header, CloudFront mengembalikannya ke penampil dengan objek yang diminta. CloudFront juga menyimpan Set-Cookie header dengan objek yang dikembalikan dari asal, dan mengirimkan Set-Cookie header tersebut ke pemirsa di semua klik cache.

Untuk informasi tentang menentukan wildcard dalam nama cookie, lihat. [Daftar cookie yang diizinkan](#)

Untuk kuota saat ini pada jumlah nama cookie yang dapat Anda teruskan untuk setiap perilaku cache, atau untuk meminta kuota yang lebih tinggi, lihat. [Kuota pada string kueri \(pengaturan cache warisan\)](#)

- Jangan meneruskan cookie ke asal Anda — CloudFront tidak menyimpan objek Anda berdasarkan cookie yang dikirim oleh pemirsa. Selain itu, CloudFront menghapus cookie sebelum meneruskan permintaan ke asal Anda, dan menghapus Set-Cookie header dari tanggapan sebelum mengembalikan tanggapan ke pemirsa Anda. Karena ini bukan cara optimal untuk menggunakan sumber daya asal Anda, ketika Anda memilih perilaku cache ini, Anda harus memastikan bahwa asal Anda tidak menyertakan cookie dalam respons asal secara default.

Perhatikan hal berikut tentang menentukan cookie yang ingin Anda teruskan:

Log akses

Jika Anda CloudFront mengonfigurasi permintaan log dan mencatat cookie, CloudFront mencatat semua cookie dan semua atribut cookie, bahkan jika Anda mengonfigurasi untuk CloudFront tidak meneruskan cookie ke asal Anda atau jika Anda mengonfigurasi CloudFront untuk meneruskan hanya cookie tertentu. Untuk informasi selengkapnya tentang CloudFront pencatatan, lihat [Akses log \(log standar\)](#).

Sensitivitas kasus

Nama dan nilai cookie bersifat peka huruf besar-kecil. Misalnya, jika CloudFront dikonfigurasi untuk meneruskan semua cookie, dan dua permintaan penampil untuk objek yang sama memiliki cookie yang identik kecuali untuk kasus, CloudFront cache objek dua kali.

CloudFront mengurutkan cookie

Jika CloudFront dikonfigurasi untuk meneruskan cookie (semua atau sebagian), CloudFront urutkan cookie dalam urutan alami berdasarkan nama cookie sebelum meneruskan permintaan ke asal Anda.

Note

Nama cookie yang dimulai dengan \$ karakter tidak didukung. CloudFront akan menghapus cookie sebelum meneruskan permintaan ke asal. Anda dapat menghapus \$ karakter atau menentukan karakter yang berbeda di awal nama cookie.

If-Modified-Since dan If-None-Match

If-Modified-Since dan permintaan If-None-Match bersyarat tidak didukung ketika CloudFront dikonfigurasi untuk meneruskan cookie (semua atau subset).

Nama standar-format pasangan nilai diperlukan

CloudFront meneruskan header cookie hanya jika nilainya sesuai dengan format [pasangan nama-nilai standar](#), misalnya: "Cookie: cookie1=value1; cookie2=value2"

Nonaktifkan cache **Set-Cookie** header

Jika CloudFront dikonfigurasi untuk meneruskan cookie ke asal (baik semua atau cookie tertentu), itu juga menyimpan Set-Cookie header yang diterima dalam respons asal. CloudFront termasuk Set-Cookie header ini dalam responsnya terhadap penampil asli, dan juga memasukkannya dalam tanggapan berikutnya yang disajikan dari CloudFront cache.

Jika Anda ingin menerima cookie di asal Anda tetapi CloudFront ingin menyimpan Set-Cookie header di tanggapan asal Anda, konfigurasi asal Anda untuk menambahkan Cache-Control header dengan no-cache arahan yang menentukan Set-Cookie sebagai nama bidang. Sebagai contoh: Cache-Control: no-cache="Set-Cookie". Untuk informasi lebih lanjut, lihat [Cache Respons-Arahan Kontrol](#) dalam standar Protokol Transfer Hiperteks (HTTP/1.1): Caching.

Panjang maksimum nama cookie

Jika Anda mengonfigurasi CloudFront untuk meneruskan cookie tertentu ke asal Anda, jumlah total byte di semua nama cookie yang Anda konfigurasi CloudFront untuk diteruskan tidak dapat melebihi 512 dikurangi jumlah cookie yang Anda teruskan. Misalnya, jika Anda mengonfigurasi CloudFront untuk meneruskan 10 cookie ke asal Anda, panjang gabungan nama 10 cookie tidak dapat melebihi 502 byte (512 — 10).

Jika Anda mengonfigurasi CloudFront untuk meneruskan semua cookie ke asal Anda, panjang nama cookie tidak menjadi masalah.

Untuk informasi tentang menggunakan CloudFront konsol untuk memperbarui distribusi sehingga CloudFront meneruskan cookie ke asal, lihat [Perbarui distribusi](#). Untuk informasi tentang menggunakan CloudFront API untuk memperbarui distribusi, lihat [UpdateDistribution](#) di Referensi CloudFront API Amazon.

Konten cache berdasarkan header permintaan

CloudFront memungkinkan Anda memilih apakah Anda CloudFront ingin meneruskan header ke asal Anda dan untuk menyimpan versi terpisah dari objek tertentu berdasarkan nilai header dalam permintaan penampil. Ini memungkinkan Anda untuk menyajikan versi konten yang berbeda berdasarkan perangkat yang digunakan pengguna, lokasi penampil, bahasa yang digunakan penampil, dan berbagai kriteria lainnya.

Topik

- [Header dan distribusi – ikhtisar](#)
- [Pilih header untuk mendasarkan caching](#)
- [Konfigurasi CloudFront untuk menghormati pengaturan CORS](#)
- [Konfigurasi caching berdasarkan jenis perangkat](#)
- [Konfigurasi caching berdasarkan bahasa pemirsa](#)

- [Konfigurasi caching berdasarkan lokasi penampil](#)
- [Konfigurasi caching berdasarkan protokol permintaan](#)
- [Konfigurasi caching untuk file terkompresi](#)
- [Bagaimana caching berdasarkan header memengaruhi kinerja](#)
- [Bagaimana kasus nilai header dan header memengaruhi caching](#)
- [Header yang CloudFront kembali ke penampil](#)

Header dan distribusi – ikhtisar

Secara default, CloudFront tidak mempertimbangkan header saat menyimpan objek Anda di lokasi tepi. Jika asal Anda mengembalikan dua objek dan mereka hanya berbeda dengan nilai di header permintaan, CloudFront cache hanya satu versi objek.

Anda dapat mengonfigurasi CloudFront untuk meneruskan header ke asal, yang menyebabkan CloudFront cache beberapa versi objek berdasarkan nilai dalam satu atau beberapa header permintaan. CloudFront Untuk mengonfigurasi objek cache berdasarkan nilai header tertentu, Anda menentukan pengaturan perilaku cache untuk distribusi Anda. Untuk informasi lebih lanjut, lihat [Cache Berdasarkan Header Permintaan yang Dipilih](#).

Misalnya, bayangkan permintaan penampil untuk `logo.jpg` berisi sebuah header `Product` kustom yang memiliki nilai `Acme` atau `Apex`. Saat Anda mengonfigurasi CloudFront untuk menyimpan objek Anda berdasarkan nilai `Product` header, CloudFront teruskan permintaan `logo.jpg` ke asal dan sertakan nilai `Product` header dan header. CloudFront cache `logo.jpg` sekali untuk permintaan di mana nilai `Product` header adalah `Acme` dan sekali untuk permintaan di mana nilainya `Apex`.

Anda dapat mengonfigurasi setiap perilaku cache dalam distribusi untuk melakukan salah satu hal berikut:

- Teruskan semua header ke asal Anda

Note

Untuk pengaturan cache lama — Jika Anda mengonfigurasi CloudFront untuk meneruskan semua header ke asal Anda, CloudFront tidak akan menyimpan objek yang terkait dengan perilaku cache ini. Alih-alih, email mengirimkan setiap permintaan ke sumber.

- Teruskan daftar header yang Anda tentukan. CloudFront cache objek Anda berdasarkan nilai di semua header yang ditentukan. CloudFront juga meneruskan header yang diteruskan secara default, tetapi cache objek Anda hanya berdasarkan header yang Anda tentukan.
- Teruskan hanya header default. Dalam konfigurasi ini, CloudFront tidak men-cache objek Anda berdasarkan nilai di header permintaan.

Untuk kuota saat ini pada jumlah header yang dapat Anda teruskan untuk setiap perilaku cache atau untuk meminta kuota yang lebih tinggi, lihat. [Kuota pada header](#)

Untuk informasi tentang menggunakan CloudFront konsol untuk memperbarui distribusi sehingga CloudFront meneruskan header ke asal, lihat. [Perbarui distribusi](#) Untuk informasi tentang penggunaan CloudFront API untuk memperbarui distribusi yang ada, lihat [Memperbarui Distribusi](#) di Referensi CloudFront API Amazon.

Pilih header untuk mendasarkan caching

Header yang dapat Anda teruskan ke asal dan yang CloudFront mendasari caching bergantung pada apakah asal Anda adalah ember Amazon S3 atau asal khusus.

- Amazon S3 - Anda dapat mengonfigurasi CloudFront untuk meneruskan dan menyimpan objek berdasarkan sejumlah header tertentu (lihat daftar pengecualian berikut). Namun, kami menyarankan agar Anda menghindari penerusan header dengan asal Amazon S3 kecuali Anda perlu menerapkan berbagi sumber daya lintas asal (CORS) atau Anda ingin mempersonalisasi konten dengan menggunakan Lambda @Edge dalam acara yang menghadap ke asal.
- Untuk mengonfigurasi CORS, Anda harus meneruskan header yang memungkinkan CloudFront untuk mendistribusikan konten untuk situs web yang diaktifkan untuk berbagi sumber daya lintas asal (CORS). Untuk informasi selengkapnya, lihat [Konfigurasi CloudFront untuk menghormati pengaturan CORS](#).
- Untuk mempersonalisasi konten dengan menggunakan header yang Anda teruskan ke asal Amazon S3, Anda menulis dan menambahkan fungsi Lambda @Edge dan mengaitkannya dengan distribusi CloudFront Anda untuk dipicu oleh peristiwa yang menghadap ke asal. Untuk informasi lebih lanjut tentang bekerja dengan header untuk mempersonalisasi konten, lihat [Personalisasi konten berdasarkan header negara atau jenis perangkat - contoh](#).

Kami menyarankan Anda menghindari penerusan header yang tidak Anda gunakan untuk mempersonalisasi konten karena meneruskan header tambahan dapat mengurangi rasio hit

cache Anda. Artinya, tidak CloudFront dapat melayani banyak permintaan dari cache tepi, sebagai proporsi dari semua permintaan.

- Asal yang disesuaikan – Anda dapat mengonfigurasi CloudFront untuk cache berdasarkan nilai header permintaan apa pun kecuali hal berikut:
 - Connection
 - Cookie – Jika Anda ingin meneruskan dan menyimpan berdasarkan cookie, Anda menggunakan pengaturan terpisah dalam distribusi Anda. Untuk informasi selengkapnya, lihat [Konten cache berdasarkan cookie](#).
 - Host (for Amazon S3 origins)
 - Proxy-Authorization
 - TE
 - Upgrade

Anda dapat CloudFront mengonfigurasi objek cache berdasarkan nilai di User-Agent header Date dan, tetapi kami tidak merekomendasikannya. Header ini memiliki banyak nilai yang mungkin, dan caching berdasarkan nilainya dapat menyebabkan CloudFront untuk meneruskan lebih banyak permintaan secara signifikan ke asal Anda.

Untuk daftar lengkap header permintaan HTTP dan cara CloudFront memprosesnya, lihat [Header dan CloudFront perilaku permintaan HTTP \(asal kustom dan Amazon S3\)](#).

Konfigurasi CloudFront untuk menghormati pengaturan CORS

Jika Anda telah mengaktifkan cross-origin resource sharing (CORS) pada bucket Amazon S3 atau origin khusus, Anda harus memilih header tertentu untuk diteruskan, untuk menghargai pengaturan CORS. Header yang harus Anda teruskan berbeda tergantung asal (Amazon S3 atau custom) dan apakah Anda ingin melakukan cache OPTIONS tanggapan mereka.

Amazon S3

- Jika Anda ingin OPTIONS respons yang harus disimpan, lakukan hal berikut:
 - Pilih opsi untuk pengaturan perilaku cache default yang mengaktifkan cache untuk OPTIONS tanggapan mereka.
 - Konfigurasi CloudFront untuk meneruskan header berikut: `Origin`, `Access-Control-Request-Headers`, dan `Access-Control-Request-Method`.

- Jika Anda tidak ingin OPTIONS respons untuk disimpan, dikonfigurasi CloudFront untuk meneruskan Origin header, bersama dengan header lain yang diwajibkan oleh negara asal Anda (misalnya, Access-Control-Request-Headers, Access-Control-Request-Method, atau lainnya).

Asal yang disesuaikan – Meneruskan Origin beserta header lainnya yang diperlukan sesuai dengan asal Anda.

CloudFront Untuk mengonfigurasi respons cache berdasarkan CORS, Anda harus mengonfigurasi CloudFront untuk meneruskan header dengan menggunakan kebijakan cache. Untuk informasi selengkapnya, lihat [Mengontrol kunci cache dengan kebijakan](#).

Untuk informasi selengkapnya tentang CORS dan Amazon S3, [lihat Menggunakan berbagi sumber daya lintas asal \(CORS\)](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Konfigurasi caching berdasarkan jenis perangkat

Jika Anda CloudFront ingin menyimpan versi objek yang berbeda berdasarkan perangkat yang digunakan pengguna untuk melihat konten Anda, konfigurasi CloudFront untuk meneruskan header yang berlaku ke asal kustom Anda:

- CloudFront-Is-Desktop-Viewer
- CloudFront-Is-Mobile-Viewer
- CloudFront-Is-SmartTV-Viewer
- CloudFront-Is-Tablet-Viewer

Berdasarkan nilai User-Agent header, CloudFront tetapkan nilai header ini ke true atau false sebelum meneruskan permintaan ke asal Anda. Jika perangkat termasuk dalam lebih dari satu kategori, lebih dari satu nilai mungkin true. Misalnya, untuk beberapa perangkat tablet, CloudFront mungkin mengatur keduanya CloudFront-Is-Mobile-Viewer dan CloudFront-Is-Tablet-Viewer ke true.

Konfigurasi caching berdasarkan bahasa pemirsa

Jika Anda CloudFront ingin menyimpan versi objek yang berbeda berdasarkan bahasa yang ditentukan dalam permintaan, konfigurasi CloudFront untuk meneruskan Accept-Language header ke asal Anda.

Konfigurasi caching berdasarkan lokasi penampil

Jika Anda CloudFront ingin menyimpan versi objek yang berbeda berdasarkan negara asal permintaan, konfigurasi CloudFront untuk meneruskan `CloudFront-Viewer-Country` header ke asal Anda. CloudFront secara otomatis mengubah alamat IP tempat permintaan berasal menjadi kode negara dua huruf. Untuk easy-to-use daftar kode negara, dapat diurutkan berdasarkan kode dan nama negara, lihat entri Wikipedia [ISO 3166-1 alpha-2](#).

Konfigurasi caching berdasarkan protokol permintaan

Jika Anda CloudFront ingin menyimpan versi objek yang berbeda berdasarkan protokol permintaan, HTTP atau HTTPS, konfigurasi CloudFront untuk meneruskan `CloudFront-Forwarded-Proto` header ke asal Anda.

Konfigurasi caching untuk file terkompresi

Jika asal Anda mendukung kompresi Brotli, Anda dapat melakukan cache berdasarkan header. `Accept-Encoding` Konfigurasi cache berdasarkan pada `Accept-Encoding` hanya jika asal Anda melayani konten yang berbeda berdasarkan header.

Bagaimana caching berdasarkan header memengaruhi kinerja

Saat Anda CloudFront mengonfigurasi cache berdasarkan satu atau beberapa header dan header memiliki lebih dari satu nilai yang mungkin, CloudFront teruskan lebih banyak permintaan ke server asal Anda untuk objek yang sama. Hal ini memperlambat kinerja dan meningkatkan beban pada server asal Anda. Jika server asal Anda mengembalikan objek yang sama terlepas dari nilai header yang diberikan, sebaiknya Anda tidak CloudFront mengonfigurasi cache berdasarkan header tersebut.

Jika Anda mengonfigurasi CloudFront untuk meneruskan lebih dari satu header, urutan header dalam permintaan penampil tidak memengaruhi caching selama nilainya sama. Misalnya, jika satu permintaan berisi header A: 1, B: 2 dan permintaan lain berisi B: 2, A: 1, cache hanya satu salinan objek. CloudFront

Bagaimana kasus nilai header dan header memengaruhi caching

Ketika CloudFront cache berdasarkan nilai header, itu tidak mempertimbangkan kasus nama header, tetapi mempertimbangkan kasus nilai header:

- Jika permintaan penampil menyertakan keduanya `Product:Acme` dan `product:Acme`, CloudFront cache objek hanya sekali. Satu-satunya perbedaan di antara keduanya adalah kasus nama header, yang tidak memengaruhi caching.
- Jika permintaan penampil menyertakan keduanya `Product:Acme` dan `Product:acme`, CloudFront cache objek dua kali, karena nilainya ada Acme di beberapa permintaan dan acme permintaan lainnya.

Header yang CloudFront kembali ke penampil

Mengkonfigurasi CloudFront untuk meneruskan dan menyimpan header tidak memengaruhi header mana yang CloudFront kembali ke penampil. CloudFront mengembalikan semua header yang didapatnya dari asal dengan beberapa pengecualian. Untuk informasi selengkapnya, lihat topik yang berlaku:

- Asal Amazon S3 – Lihat [Header respons HTTP yang CloudFront menghapus atau memperbarui](#).
- Asal yang disesuaikan – Lihat [Header respons HTTP yang CloudFront menghapus atau menggantikan](#).

Mengontrol kunci cache dengan kebijakan

Dengan kebijakan CloudFront cache, Anda dapat menentukan header HTTP, cookie, dan string kueri yang CloudFront disertakan dalam kunci cache untuk objek yang di-cache di CloudFront lokasi tepi. Kunci cache adalah pengidentifikasi unik untuk setiap objek dalam cache, dan menentukan apakah permintaan HTTP penampil menghasilkan hit cache.

Hit cache terjadi ketika permintaan penampil menghasilkan kunci cache yang sama dengan permintaan sebelumnya, dan objek untuk kunci cache tersebut berada di cache lokasi tepi dan valid. Ketika ada cache hit, objek disajikan ke penampil dari lokasi CloudFront tepi, yang memiliki manfaat sebagai berikut:

- Berkurangnya beban pada server asal Anda
- Berkurangnya latensi untuk penampil

Menyertakan nilai yang lebih sedikit dalam kunci cache meningkatkan kemungkinan hit cache. Ini dapat memberi Anda kinerja yang lebih baik dari situs web atau aplikasi Anda karena ada rasio hit cache yang lebih tinggi (proporsi permintaan pemirsa yang lebih tinggi yang menghasilkan hit cache). Untuk informasi selengkapnya, lihat [Memahami kunci cache](#).

Untuk mengontrol kunci cache, Anda menggunakan kebijakan CloudFront cache. Anda melampirkan kebijakan cache ke satu atau beberapa perilaku cache dalam CloudFront distribusi.

Anda juga dapat menggunakan kebijakan cache untuk menentukan pengaturan time to live (TTL) untuk objek dalam CloudFront cache, dan memungkinkan CloudFront untuk meminta dan menyimpan objek terkompresi.

Note

Pengaturan cache tidak memengaruhi permintaan gRPC karena lalu lintas gRPC tidak dapat di-cache. Untuk informasi selengkapnya, lihat [Menggunakan gRPC dengan distribusi CloudFront](#).

Topik

- [Memahami kebijakan cache](#)
- [Buat kebijakan cache](#)

- [Gunakan kebijakan cache terkelola](#)
- [Memahami kunci cache](#)

Memahami kebijakan cache

Anda dapat menggunakan kebijakan cache untuk meningkatkan rasio hit cache Anda dengan mengontrol nilai (string kueri URL, header HTTP, dan cookie) yang disertakan dalam kunci cache. CloudFront menyediakan beberapa kebijakan cache yang telah ditentukan sebelumnya, yang dikenal sebagai kebijakan terkelola, untuk kasus penggunaan umum. Anda dapat menggunakan kebijakan terkelola ini, atau Anda dapat membuat kebijakan cache sendiri yang khusus untuk kebutuhan Anda. Untuk informasi selengkapnya tentang kebijakan terkelola, lihat [Gunakan kebijakan cache terkelola](#).

Kebijakan cache berisi pengaturan berikut, yang dikategorikan menjadi informasi kebijakan, waktu ke pengaturan langsung (TTL), dan pengaturan kunci cache.

Informasi kebijakan

Nama

Nama untuk mengidentifikasi kebijakan cache. Di konsol, Anda menggunakan nama untuk melampirkan kebijakan cache ke perilaku cache.

Deskripsi

Komentar untuk menjelaskan kebijakan cache. Ini opsional, tetapi dapat membantu Anda mengidentifikasi tujuan kebijakan cache.

Waktu ke pengaturan langsung (TTL)

Pengaturan time to live (TTL) bekerja sama dengan header Cache-Control dan Expires HTTP (jika berada dalam respons asal) untuk menentukan berapa lama objek dalam CloudFront cache tetap valid.

TTL Minimum

Jumlah waktu minimum, dalam hitungan detik, Anda ingin objek tetap berada di CloudFront cache sebelum CloudFront memeriksa dengan asal untuk melihat apakah objek telah diperbarui. Untuk informasi selengkapnya, lihat [Mengelola berapa lama konten tetap dalam cache \(kedaluwarsa\)](#).

 Warning

Jika TTL minimum Anda lebih besar dari 0, CloudFront akan menyimpan konten untuk setidaknya durasi yang ditentukan dalam TTL minimum kebijakan cache, bahkan jika `Cache-Control: no-cache, no-store`, atau `private` arahan ada di header asal.

TTL Maksimum

Jumlah waktu maksimum, dalam hitungan detik, objek tetap berada di CloudFront cache sebelum CloudFront memeriksa dengan asal untuk melihat apakah objek telah diperbarui. CloudFront menggunakan pengaturan ini hanya ketika asal mengirim `Cache-Control` atau `Expires` header dengan objek. Untuk informasi selengkapnya, lihat [Mengelola berapa lama konten tetap dalam cache \(kedaluwarsa\)](#).

TTL Default

Jumlah waktu default, dalam hitungan detik, yang Anda inginkan objek tetap berada di CloudFront cache sebelum CloudFront memeriksa dengan asal untuk melihat apakah objek telah diperbarui. CloudFront menggunakan nilai pengaturan ini sebagai TTL objek hanya ketika asal tidak mengirim `Cache-Control` atau `Expires` header dengan objek. Untuk informasi selengkapnya, lihat [Mengelola berapa lama konten tetap dalam cache \(kedaluwarsa\)](#).

 Note

Jika pengaturan TTL Minimum, TTL Maksimum, dan TTL Default semuanya diatur ke 0, ini menonaktifkan caching. CloudFront

Pengaturan tombol Cache

Pengaturan kunci cache menentukan nilai dalam permintaan penampil yang CloudFront disertakan dalam kunci cache. Nilai dapat mencakup string kueri URL, header HTTP, dan cookie. Nilai yang Anda sertakan dalam kunci cache secara otomatis disertakan dalam permintaan yang CloudFront mengirim ke asal, dikenal sebagai permintaan asal. Untuk informasi tentang pengendalian permintaan asal tanpa memengaruhi kunci cache, lihat [Kontrol permintaan asal dengan kebijakan](#).

Pengaturan kunci Cache meliputi:

- [Header](#)
- [Cookie](#)
- [String kueri](#)
- [Dukungan kompresi](#)

Header

Header HTTP dalam permintaan penampil yang CloudFront termasuk dalam kunci cache dan permintaan asal. Untuk header, Anda dapat memilih salah satu pengaturan berikut:

- Tidak ada – Header HTTP dalam permintaan penampil adalah tidak yang termasuk dalam kunci cache dan tidak secara otomatis disertakan dalam permintaan asal.
- Sertakan header berikut - Anda menentukan header HTTP mana dalam permintaan penampil yang disertakan dalam kunci cache dan secara otomatis disertakan dalam permintaan asal.

Saat Anda menggunakan setelan Sertakan header berikut, Anda menentukan header HTTP berdasarkan namanya, bukan nilainya. Misalnya, pertimbangkan HTTP header berikut:

```
Accept-Language: en-US,en;q=0.5
```

Dalam hal ini, Anda menentukan header sebagai Accept-Language, bukan sebagai Accept-Language: en-US,en;q=0.5. Namun, CloudFront termasuk header lengkap, termasuk nilainya, dalam kunci cache dan permintaan asal.

Anda juga dapat menyertakan header tertentu yang dihasilkan oleh CloudFront dalam kunci cache. Untuk informasi selengkapnya, lihat [the section called “Tambahkan header CloudFront permintaan”](#).

Cookie

Cookie dalam permintaan penampil yang CloudFront termasuk dalam kunci cache dan permintaan asal. Untuk cookie, Anda dapat memilih salah satu pengaturan berikut:

- Tidak ada – Cookie di permintaan penampil adalah tidak yang termasuk dalam kunci cache dan tidak secara otomatis disertakan dalam permintaan asal.
- Semua – Semua cookie di permintaan penampil disertakan dalam kunci cache dan secara otomatis disertakan dalam permintaan asal.
- Sertakan cookie tertentu - Anda menentukan cookie mana dalam permintaan penampil yang disertakan dalam kunci cache dan secara otomatis disertakan dalam permintaan asal.

- Sertakan semua cookie kecuali — Anda menentukan cookie mana dalam permintaan penampil yang tidak termasuk dalam kunci cache dan tidak secara otomatis disertakan dalam permintaan asal. Semua cookie lain, kecuali cookie yang Anda tentukan, adalah disertakan dalam kunci cache dan secara otomatis disertakan dalam permintaan asal.

Ketika Anda menggunakan Sertakan cookie yang ditentukan atau Sertakan semua cookie kecuali pengaturan, Anda menentukan cookie dengan namanya, bukan nilainya. Misalnya, pertimbangkan berikut ini Cookie header:

```
Cookie: session_ID=abcd1234
```

Dalam hal ini, Anda menentukan cookie sebagai `session_ID`, bukan sebagai `session_ID=abcd1234`. Namun, CloudFront termasuk cookie lengkap, termasuk nilainya, dalam kunci cache dan permintaan asal.

String kueri

String kueri URL dalam permintaan penampil yang CloudFront disertakan dalam kunci cache dan permintaan asal. Untuk string kueri, Anda dapat memilih salah satu pengaturan berikut:

- Tidak ada – String kueri pada permintaan pemirsa adalah tidak yang termasuk dalam kunci cache dan tidak secara otomatis disertakan dalam permintaan asal.
- Semua – Semua string kueri dalam permintaan penampil disertakan dalam kunci cache dan juga secara otomatis disertakan dalam permintaan asal.
- Sertakan string kueri yang ditentukan - Anda menentukan string kueri mana dalam permintaan penampil yang disertakan dalam kunci cache dan secara otomatis disertakan dalam permintaan asal.
- Sertakan semua string kueri kecuali - Anda menentukan string kueri mana dalam permintaan penampil yang tidak disertakan dalam kunci cache dan tidak secara otomatis disertakan dalam permintaan asal. Semua string kueri lainnya, kecuali untuk yang Anda tentukan, adalah disertakan dalam kunci cache dan secara otomatis disertakan dalam permintaan asal.

Saat Anda menggunakan Sertakan string kueri yang ditentukan atau Sertakan semua string kueri kecuali setelan, Anda menentukan string kueri berdasarkan namanya, bukan nilainya. Misalnya, pertimbangkan alur URL berikut ini:

```
/content/stories/example-story.html?split-pages=false
```

Dalam hal ini, Anda menentukan string kueri sebagai `split-pages`, bukan sebagai `split-pages=false`. Namun, CloudFront termasuk string kueri lengkap, termasuk nilainya, dalam kunci cache dan permintaan asal.

 Note

Untuk pengaturan kunci cache, CloudFront perlakukan karakter tanda bintang (*) untuk header, string kueri, dan cookie sebagai string literal, bukan sebagai wildcard.

Dukungan kompresi

Pengaturan ini memungkinkan CloudFront untuk meminta dan menyimpan objek cache yang dikompresi dalam format kompresi Gzip atau Brotli, ketika penampil mendukungnya. Pengaturan ini juga memungkinkan [CloudFront kompresi](#) untuk bekerja. Penampil menunjukkan dukungan mereka untuk format kompresi ini dengan Accept-Encoding Header HTTP.

 Note

Browser web Chrome dan Firefox mendukung kompresi Brotli hanya jika permintaan dikirim menggunakan HTTPS. Browser ini tidak mendukung Brotli dengan permintaan HTTP.

Aktifkan pengaturan ini jika salah satu di bawah ini benar:

- Asal Anda mengembalikan objek terkompresi Gzip ketika penampil mendukungnya (permintaan berisi header HTTP Accept-Encoding dengan gzip sebagai nilai). Dalam hal ini, gunakan pengaturan berkemampuan Gzip (disetel `EnableAcceptEncodingGzip` ke `true` dalam CloudFront API,, AWS SDKs AWS CLI, atau CloudFormation).
- Asal Anda mengembalikan objek terkompresi Brotli saat penampil mendukungnya (permintaan memuat Accept-Encoding Header HTTP dengan br sebagai nilai). Dalam kasus ini, gunakan setelan berkemampuan Brotli (disetel `EnableAcceptEncodingBrotli` ke `true` dalam CloudFront API,, AWS SDKs AWS CLI, atau CloudFormation).
- Perilaku cache yang terkait dengan kebijakan cache ini dikonfigurasi dengan [CloudFront kompresi](#). Dalam hal ini, Anda dapat mengaktifkan caching untuk Gzip atau Brotli, atau keduanya. Saat CloudFront kompresi diaktifkan, mengaktifkan caching untuk kedua format dapat membantu mengurangi biaya transfer data ke internet.

 Note

Jika Anda mengaktifkan caching untuk salah satu atau kedua format kompresi ini, jangan sertakan `Accept-Encoding` header dalam [kebijakan permintaan asal](#) yang terkait dengan perilaku cache yang sama. CloudFront selalu menyertakan header ini dalam permintaan asal saat caching diaktifkan untuk salah satu format ini, jadi termasuk `Accept-Encoding` dalam kebijakan permintaan asal tidak berpengaruh.

Jika server asal Anda tidak mengembalikan objek terkompresi Gzip atau Brotli, atau perilaku cache tidak dikonfigurasi dengan CloudFront kompresi, jangan aktifkan caching untuk objek terkompresi. Jika Anda melakukannya, hal tersebut dapat menyebabkan penurunan [rasio tembolok](#).

Berikut ini menjelaskan bagaimana pengaturan ini mempengaruhi CloudFront distribusi. Semua skenario berikut ini mengasumsikan bahwa permintaan pemirsa mencakup `Accept-Encoding` header. Ketika permintaan penampil tidak menyertakan `Accept-Encoding` header, CloudFront tidak menyertakan header ini di kunci cache dan tidak menyertakannya dalam permintaan asal yang sesuai.

Saat caching, objek terkompresi diaktifkan untuk kedua format kompresi

Jika penampil mendukung Gzip dan Brotli—yaitu, jika `br` nilai `gzip` dan keduanya ada di `Accept-Encoding` header dalam permintaan penampil—lakukan hal berikut: CloudFront

- Normalisasi header ke `Accept-Encoding: br,gzip` dan mencakup header yang dinormalisasi dalam tombol cache. Kunci cache tidak menyertakan nilai lain yang ada di `Accept-Encoding` header yang dikirim oleh penampil.
- Jika lokasi tepi memiliki objek terkompresi Brotli atau Gzip di cache yang sesuai dengan permintaan dan tidak kedaluwarsa, lokasi tepi mengembalikan objek ke penampil.
- Jika lokasi tepi tidak memiliki objek terkompresi Brotli atau Gzip di cache yang cocok dengan permintaan dan tidak kedaluwarsa, CloudFront sertakan header (`Accept-Encoding: br,gzip`) yang dinormalisasi dalam permintaan asal yang sesuai. Permintaan asal tidak menyertakan nilai lain yang ada di `Accept-Encoding` header yang dikirim oleh penampil.

Jika penampil mendukung satu format kompresi tetapi tidak yang lain—misalnya, jika `gzip` adalah nilai di `Accept-Encoding` header dalam permintaan penampil tetapi `br` CloudFront tidak—lakukan hal berikut:

- Normalisasi header ke `Accept-Encoding: gzip` dan mencakup header yang dinormalisasi dalam tombol cache. Kunci cache tidak menyertakan nilai lain yang ada di `Accept-Encoding` header yang dikirim oleh penampil.
- Jika lokasi tepi memiliki objek terkompresi Gzip di cache yang cocok dengan permintaan dan tidak kedaluwarsa, lokasi tepi akan mengembalikan objek ke penampil.
- Jika lokasi tepi tidak memiliki objek terkompresi Gzip di cache yang cocok dengan permintaan dan tidak kedaluwarsa, CloudFront sertakan header (`Accept-Encoding: gzip`) yang dinormalisasi dalam permintaan asal yang sesuai. Permintaan asal tidak menyertakan nilai lain yang ada di `Accept-Encoding` header yang dikirim oleh penampil.

Untuk memahami CloudFront apa yang dilakukan jika penampil mendukung Brotli tetapi tidak Gzip, ganti dua format kompresi satu sama lain dalam contoh sebelumnya.

Jika penampil tidak mendukung Brotli atau GZip—yaitu, `Accept-Encoding` header dalam permintaan penampil tidak berisi `br` atau sebagai nilai—: `gzip` CloudFront

- Tidak termasuk `Accept-Encoding` header dalam kunci cache.
- Termasuk `Accept-Encoding: identity` dalam permintaan asal terkait. Permintaan asal tidak menyertakan nilai lain yang ada di `Accept-Encoding` header yang dikirim oleh penampil.

Saat caching, objek terkompresi diaktifkan untuk satu format kompresi, tetapi tidak lainnya

Jika penampil mendukung format yang mengaktifkan caching — misalnya, jika caching objek terkompresi diaktifkan untuk Gzip dan penampil mendukung Gzip (`gzip` adalah salah satu nilai di `Accept-Encoding` header dalam permintaan penampil) — lakukan hal berikut: CloudFront

- Normalisasi header ke `Accept-Encoding: gzip` dan mencakup header yang dinormalisasi dalam tombol cache.
- Jika lokasi tepi memiliki objek terkompresi Gzip di cache yang cocok dengan permintaan dan tidak kedaluwarsa, lokasi tepi akan mengembalikan objek ke penampil.
- Jika lokasi tepi tidak memiliki objek terkompresi Gzip di cache yang cocok dengan permintaan dan tidak kedaluwarsa, CloudFront sertakan header (`Accept-Encoding: gzip`) yang dinormalisasi dalam permintaan asal yang sesuai. Permintaan asal tidak menyertakan nilai lain yang ada di `Accept-Encoding` header yang dikirim oleh penampil.

Perilaku ini sama saat penampil mendukung Gzip dan Brotli (header `Accept-Encoding` di permintaan penampil mencakup keduanya `gzip` dan `br` sebagai nilai), karena dalam skenario ini, caching di objek terkompresi untuk Brotli tidak diaktifkan.

Untuk memahami CloudFront apa yang terjadi jika caching objek terkompresi diaktifkan untuk Brotli tetapi tidak Gzip, ganti dua format kompresi satu sama lain dalam contoh sebelumnya.

Jika penampil tidak mendukung format kompresi yang caching diaktifkan (Accept-Encodingheader dalam permintaan penampil tidak berisi nilai untuk format tersebut), CloudFront:

- Tidak termasuk Accept-Encoding header dalam kunci cache.
- Termasuk Accept-Encoding: identity dalam permintaan asal terkait. Permintaan asal tidak menyertakan nilai lain yang ada di Accept-Encoding header yang dikirim oleh penampil.

Saat caching, objek terkompresi dinonaktifkan untuk kedua format kompresi

Saat caching objek terkompresi dinonaktifkan untuk kedua format kompresi CloudFront , perlakukan Accept-Encoding header sama seperti header HTTP lainnya dalam permintaan penampil. Secara default, itu tidak termasuk dalam kunci cache dan tidak termasuk dalam permintaan asal. Anda dapat memasukkannya ke dalam daftar header dalam kebijakan cache atau kebijakan permintaan asal sama seperti header HTTP lainnya.

Buat kebijakan cache

Anda dapat menggunakan kebijakan cache untuk meningkatkan rasio ketukan cache dengan mengendalikan nilai (string kueri URL, header HTTP, dan cookie) yang disertakan dalam kunci cache. Anda dapat membuat kebijakan cache di CloudFront konsol, dengan AWS Command Line Interface (AWS CLI), atau dengan CloudFront API.

Setelah membuat kebijakan cache, Anda melampirkannya pada satu atau beberapa perilaku cache di CloudFront distribusi.

Console

Untuk membuat kebijakan cache (konsole)

1. Masuk ke Konsol Manajemen AWS dan buka halaman Kebijakan di CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home?#/policies>.
2. Pilih Buat kebijakan cache.
3. Pilih pengaturan yang diinginkan untuk kebijakan cache ini. Untuk informasi selengkapnya, lihat [Memahami kebijakan cache](#).

4. Setelah selesai, pilih Buat.

Setelah membuat kebijakan cache, Anda dapat memasangnya ke perilaku cache.

Untuk melampirkan kebijakan cache ke distribusi yang sudah ada (konsole)

1. Membuka Distribusi di halaman dalam CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home#/distributions>.
2. Pilih distribusi untuk diperbarui, lalu pilih Perilaku tab.
3. Pilih perilaku cache untuk diperbarui, lalu pilih Edit.

Atau, untuk membuat perilaku cache baru, pilih Buat perilaku.

4. Di bagian Kunci cache dan permintaan asal, pastikan kebijakan Cache dan kebijakan permintaan asal dipilih.
5. Untuk kebijakan Cache, pilih kebijakan cache untuk dilampirkan ke perilaku cache ini.
6. Di bagian bawah halaman, pilih Simpan perubahan.

Untuk melampirkan kebijakan cache ke distribusi baru (konsole)

1. Buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Pilih Buat Distribusi.
3. Di bagian Kunci cache dan permintaan asal, pastikan kebijakan Cache dan kebijakan permintaan asal dipilih.
4. Untuk kebijakan Cache, pilih kebijakan cache untuk dilampirkan ke perilaku cache default distribusi ini.
5. Pilih pengaturan yang diinginkan untuk asal, perilaku cache default, dan pengaturan distribusi lainnya. Untuk informasi selengkapnya, lihat [Semua referensi pengaturan distribusi](#).
6. Setelah selesai, pilih Buat distribusi.

CLI

Untuk membuat kebijakan cache dengan AWS Command Line Interface (AWS CLI), gunakan `aws cloudfront create-cache-policy` perintah. Anda dapat menggunakan file input untuk memberikan parameter input perintah, daripada menentukan setiap parameter individu sebagai input baris perintah.

Untuk membuat kebijakan cache (CLI dengan file input)

1. Gunakan perintah berikut untuk membuat file dengan nama `cache-policy.yaml` yang berisi semua parameter input untuk `create-cache-policy` perintah.

```
aws cloudfront create-cache-policy --generate-cli-skeleton yml-input > cache-policy.yaml
```

2. Buka file dengan nama `cache-policy.yaml` yang baru saja Anda buat. Edit file untuk menentukan pengaturan kebijakan cache yang diinginkan, lalu simpan file. Anda dapat menghapus bidang opsional dari file, tetapi jangan menghapus bidang yang diperlukan.

Untuk informasi lebih lanjut tentang pengaturan kebijakan cache, lihat [Memahami kebijakan cache](#).

3. Gunakan perintah berikut untuk membuat kebijakan cache menggunakan parameter input dari `cache-policy.yaml` file Anda.

```
aws cloudfront create-cache-policy --cli-input-yaml file://cache-policy.yaml
```

Catat Id nilai dalam output perintah. Ini adalah ID kebijakan cache, dan Anda memerlukannya untuk melampirkan kebijakan cache ke perilaku cache CloudFront distribusi.

Untuk melampirkan kebijakan cache ke distribusi yang ada (CLI dengan file input)

1. Gunakan perintah berikut untuk menyimpan konfigurasi distribusi untuk CloudFront distribusi yang ingin Anda perbarui. Ganti *distribution_ID* dengan ID distribusi.

```
aws cloudfront get-distribution-config --id distribution_ID --output yaml > dist-config.yaml
```

2. Buka file dengan nama `dist-config.yaml` yang baru saja Anda buat. Edit file, membuat perubahan berikut pada setiap perilaku cache yang Anda perbarui untuk menggunakan kebijakan cache.
 - Dalam perilaku cache, tambahkan bidang bernama `CachePolicyId`. Untuk nilai bidang, gunakan ID kebijakan cache yang Anda catat setelah membuat kebijakan.

- Hapus MinTTL, MaxTTL, DefaultTTL, dan ForwardedValues bidang dari perilaku cache. Pengaturan ini ditentukan dalam kebijakan cache, sehingga Anda tidak dapat menyertakan bidang ini dan kebijakan cache dalam perilaku cache yang sama.
- Ubah nama ETag bidang menjadi IfMatch, tetapi jangan ubah nilai bidang.

Simpan file setelah selesai.

3. Gunakan perintah berikut untuk memperbarui distribusi untuk menggunakan kebijakan cache. Ganti *distribution_ID* dengan ID distribusi.

```
aws cloudfront update-distribution --id distribution_ID --cli-input-yaml file://dist-config.yaml
```

Untuk melampirkan kebijakan cache ke distribusi baru (CLI dengan file input)

1. Gunakan perintah berikut untuk membuat file dengan nama `distribution.yaml` yang berisi semua parameter input untuk `create-distribution` perintah.

```
aws cloudfront create-distribution --generate-cli-skeleton yaml-input > distribution.yaml
```

2. Buka file dengan nama `distribution.yaml` yang baru saja Anda buat. Dalam perilaku cache default, di bidang `CachePolicyId`, masukkan ID kebijakan cache yang Anda catat setelah membuat kebijakan. Lanjutkan mengedit file untuk menentukan pengaturan distribusi yang Anda inginkan, kemudian simpan file setelah selesai.

Untuk informasi lebih lanjut tentang pengaturan distribusi, lihat [Semua referensi pengaturan distribusi](#).

3. Gunakan perintah berikut untuk membuat distribusi menggunakan parameter input dari `distribution.yaml` file Anda.

```
aws cloudfront create-distribution --cli-input-yaml file://distribution.yaml
```


API

Untuk membuat kebijakan cache dengan CloudFront API, gunakan [CreateCachePolicy](#). Untuk informasi selengkapnya tentang bidang yang Anda tentukan dalam panggilan API ini, lihat [Memahami kebijakan cache](#) dan dokumentasi referensi API untuk AWS SDK atau klien API lainnya.

Setelah membuat kebijakan cache, Anda dapat memasangnya ke perilaku cache, menggunakan salah satu panggilan API berikut:

- Untuk memasangnya ke perilaku cache dalam distribusi yang ada, gunakan [UpdateDistribution](#).
- Untuk memasangnya ke perilaku cache dalam distribusi baru, gunakan [CreateDistribution](#).

Untuk kedua panggilan API ini, berikan ID kebijakan cache di `CachePolicyId` bidang, di dalam perilaku cache. Untuk informasi selengkapnya tentang bidang lain yang Anda tentukan dalam panggilan API ini, lihat [Semua referensi pengaturan distribusi](#) dan dokumentasi referensi API untuk AWS SDK atau klien API lainnya.

Gunakan kebijakan cache terkelola

CloudFront menyediakan serangkaian kebijakan cache terkelola yang dapat Anda lampirkan ke perilaku cache distribusi Anda. Dengan kebijakan cache terkelola, Anda tidak perlu menulis atau memelihara kebijakan cache Anda sendiri. Kebijakan terkelola menggunakan pengaturan yang dioptimalkan untuk kasus penggunaan spesifik.

Untuk menggunakan kebijakan cache terkelola, Anda melampirkannya ke perilaku cache dalam distribusi Anda. Prosesnya sama seperti ketika Anda membuat kebijakan cache, tetapi daripada membuat yang baru, Anda hanya perlu melampirkan salah satu kebijakan cache terkelola. Anda melampirkan kebijakan baik dengan nama (dengan konsol) atau dengan ID (dengan AWS CLI atau SDKs). Nama-nama dan IDs tercantum di bagian berikut.

Untuk informasi selengkapnya, lihat [Buat kebijakan cache](#).

Topik berikut menjelaskan kebijakan cache terkelola yang dapat Anda gunakan.

Topik

- [Amplify](#)

- [CachingDisabled](#)
- [CachingOptimized](#)
- [CachingOptimizedForUncompressedObjects](#)
- [Elemen- MediaPackage](#)
- [UseOriginCacheControlHeaders](#)
- [UseOriginCacheControlHeaders-QueryString](#)

Amplify

[Lihat kebijakan ini di CloudFront konsol](#)

Kebijakan ini dirancang untuk digunakan dengan asal yang merupakan aplikasi [AWS Amplify](#) web.

Saat menggunakan CloudFormation, the AWS CLI, atau CloudFront API, ID untuk kebijakan ini adalah:

2e54312d-136d-493c-8eb9-b001f22f67d2

Kebijakan ini memiliki pengaturan berikut:

- TTL minimum: 2 detik
- TTL maksimum: 600 detik (10 menit)
- Default TTL: 2 detik
- Header termasuk dalam kunci cache:
 - Authorization
 - CloudFront-Viewer-Country
 - Host

Accept-EncodingHeader yang dinormalisasi juga disertakan karena pengaturan objek terkompresi cache diaktifkan. Untuk informasi selengkapnya, lihat [Dukungan kompresi](#).

- Cookie termasuk dalam kunci cache: Semua cookie disertakan.
- String kueri disertakan dalam kunci cache: Semua string kueri disertakan.
- Pengaturan objek terkompresi cache: Diaktifkan. Untuk informasi selengkapnya, lihat [Dukungan kompresi](#).

 Warning

Karena kebijakan ini memiliki TTL minimum lebih besar dari 0, CloudFront akan menyimpan konten untuk setidaknya durasi yang ditentukan dalam TTL minimum kebijakan cache, bahkan jika `Cache-Control: no-cache, no-store`, atau `private` arahan ada di header asal.

AWS Amplify Kebijakan cache hosting

Amplify menggunakan kebijakan cache terkelola berikut untuk mengoptimalkan konfigurasi cache default untuk aplikasi pelanggan:

- [Amplify-Default](#)
- [Amplify- DefaultNoCookies](#)
- [Amplify- ImageOptimization](#)
- [Amplify- StaticContent](#)

 Note

Kebijakan ini hanya digunakan oleh Amplify. Kami tidak menyarankan Anda menggunakan kebijakan ini untuk distribusi Anda.

Untuk informasi selengkapnya tentang mengelola konfigurasi cache untuk aplikasi yang dihosting Amplify, lihat [Mengelola konfigurasi cache di Panduan Pengguna](#) Amplify Hosting.

CachingDisabled

[Lihat kebijakan ini di CloudFront konsol](#)

Kebijakan ini menonaktifkan caching. Kebijakan ini berguna untuk konten dinamis dan untuk permintaan yang tidak dapat disimpan.

Saat menggunakan CloudFormation, the AWS CLI, atau CloudFront API, ID untuk kebijakan ini adalah:

4135ea2d-6df8-44a3-9df3-4b5a84be39ad

Kebijakan ini memiliki pengaturan berikut:

- TTL minimum: 0 detik
- TTL maksimum: 0 detik
- Default TTL: 0 detik
- Header termasuk dalam kunci cache: Tidak ada
- Cookie termasuk dalam kunci cache: Tidak ada
- String kueri termasuk dalam kunci cache: Tidak ada
- Pengaturan objek terkompresi Cache: Dinonaktifkan

CachingOptimized

[Lihat kebijakan ini di CloudFront konsol](#)

Kebijakan ini dirancang untuk mengoptimalkan efisiensi cache dengan meminimalkan nilai yang CloudFront disertakan dalam kunci cache. CloudFront tidak menyertakan string kueri atau cookie apa pun di kunci cache, dan hanya menyertakan header yang dinormalisasi `Accept-Encoding`. Hal ini memungkinkan CloudFront untuk secara terpisah cache objek dalam format kompresi Gzip dan Brotli ketika asal mengembalikannya atau ketika kompresi [CloudFront tepi](#) diaktifkan.

Saat menggunakan CloudFormation, the AWS CLI, atau CloudFront API, ID untuk kebijakan ini adalah:

658327ea-f89d-4fab-a63d-7e88639e58f6

Kebijakan ini memiliki pengaturan berikut:

- Minimum TTL: 1 detik.
- TTL maksimum: 31.536.000 detik (365 hari).
- Default TTL: 86.400 detik (24 jam).
- Header yang disertakan dalam kunci cache: Tidak ada yang disertakan secara eksplisit. Dinormalkan `Accept-Encoding` header disertakan karena pengaturan objek terkompresi cache diaktifkan. Untuk informasi selengkapnya, lihat [Dukungan kompresi](#).
- Cookie yang disertakan dalam kunci cache: Tidak ada.
- String kueri yang disertakan dalam kunci cache: Tidak ada.

- Pengaturan objek terkompresi cache: Diaktifkan. Untuk informasi selengkapnya, lihat [Dukungan kompresi](#).

 Warning

Karena kebijakan ini memiliki TTL minimum lebih besar dari 0, CloudFront akan menyimpan konten untuk setidaknya durasi yang ditentukan dalam TTL minimum kebijakan cache, bahkan jika `Cache-Control: no-cache, no-store`, atau `private` arahan ada di header asal.

CachingOptimizedForUncompressedObjects

[Lihat kebijakan ini di CloudFront konsol](#)

Kebijakan ini dirancang untuk mengoptimalkan efisiensi cache dengan meminimalkan nilai yang disertakan dalam kunci cache. Tidak ada string kueri, header, atau cookie yang disertakan. Kebijakan ini identik dengan yang sebelumnya, tetapi menonaktifkan pengaturan objek terkompresi cache.

Saat menggunakan CloudFormation, the AWS CLI, atau CloudFront API, ID untuk kebijakan ini adalah:

b2884449-e4de-46a7-ac36-70bc7f1ddd6d

Kebijakan ini memiliki pengaturan berikut:

- Minimum TTL: 1 detik
- TTL maksimum: 31.536.000 detik (365 hari)
- TTL default: 86.400 detik (24 jam)
- Header termasuk dalam kunci cache: Tidak ada
- Cookie termasuk dalam kunci cache: Tidak ada
- String kueri termasuk dalam kunci cache: Tidak ada
- Pengaturan objek terkompresi Cache: Dinonaktifkan

⚠ Warning

Karena kebijakan ini memiliki TTL minimum lebih besar dari 0, CloudFront akan menyimpan konten untuk setidaknya durasi yang ditentukan dalam TTL minimum kebijakan cache, bahkan jika `Cache-Control: no-cache, no-store`, atau `private` arahan ada di header asal.

Elemen- MediaPackage

[Lihat kebijakan ini di CloudFront konsol](#)

Kebijakan ini dirancang untuk digunakan dengan asal yang merupakan AWS Elemental MediaPackage titik akhir.

Saat menggunakan CloudFormation, the AWS CLI, atau CloudFront API, ID untuk kebijakan ini adalah:

08627262-05a9-4f76-9ded-b50ca2e3a84f

Kebijakan ini memiliki pengaturan berikut:

- TTL minimum: 0 detik
- TTL maksimum: 31.536.000 detik (365 hari)
- TTL default: 86.400 detik (24 jam)
- Header termasuk dalam kunci cache:
 - `Origin`

`Accept-Encoding` header yang dinormalisasi juga disertakan karena pengaturan objek terkompresi cache diaktifkan untuk Gzip. Untuk informasi selengkapnya, lihat [Dukungan kompresi](#).

- Cookie termasuk dalam kunci cache: Tidak ada
- String kueri termasuk dalam kunci cache:
 - `aws.manifestfilter`
 - `start`
 - `end`
 - `m`

- Pengaturan objek terkompresi cache: Diaktifkan untuk Gzip. Untuk informasi selengkapnya, lihat [Dukungan kompresi](#).

UseOriginCacheControlHeaders

[Lihat kebijakan ini di CloudFront konsol](#)

Kebijakan ini dirancang untuk digunakan dengan origin yang menampilkan header respons Cache-Control HTTP dan tidak menyajikan konten yang berbeda berdasarkan nilai yang ada dalam string kueri. Jika asal Anda menyajikan konten yang berbeda berdasarkan nilai yang ada dalam string kueri, pertimbangkan untuk menggunakan [UseOriginCacheControlHeaders-QueryStrings](#).

Saat menggunakan CloudFormation, the AWS CLI, atau CloudFront API, ID untuk kebijakan ini adalah:

83da9c7e-98b4-4e11-a168-04f0df8e2c65

Kebijakan ini memiliki pengaturan berikut:

- TTL minimum: 0 detik
- TTL maksimum: 31.536.000 detik (365 hari)
- Default TTL: 0 detik
- Header termasuk dalam kunci cache:
 - Host
 - Origin
 - X-HTTP-Method-Override
 - X-HTTP-Method
 - X-Method-Override

Accept-EncodingHeader yang dinormalisasi juga disertakan karena pengaturan objek terkompresi cache diaktifkan. Untuk informasi selengkapnya, lihat [Dukungan kompresi](#).

- Cookie termasuk dalam kunci cache: Semua cookie disertakan.
- String kueri yang disertakan dalam kunci cache: Tidak ada.
- Pengaturan objek terkompresi cache: Diaktifkan. Untuk informasi selengkapnya, lihat [Dukungan kompresi](#).

UseOriginCacheControlHeaders-QueryStrings

[Lihat kebijakan ini di CloudFront konsol](#)

Kebijakan ini dirancang untuk digunakan dengan origin yang menampilkan header respons Cache-Control HTTP dan menyajikan konten berbeda berdasarkan nilai yang ada dalam string kueri. Jika asal Anda tidak menyajikan konten yang berbeda berdasarkan nilai yang ada dalam string kueri, pertimbangkan untuk menggunakan [UseOriginCacheControlHeaders](#).

Saat menggunakan CloudFormation, the AWS CLI, atau CloudFront API, ID untuk kebijakan ini adalah:

4cc15a8a-d715-48a4-82b8-cc0b614638fe

Kebijakan ini memiliki pengaturan berikut:

- TTL minimum: 0 detik
- TTL maksimum: 31.536.000 detik (365 hari)
- Default TTL: 0 detik
- Header termasuk dalam kunci cache:
 - Host
 - Origin
 - X-HTTP-Method-Override
 - X-HTTP-Method
 - X-Method-Override

Accept-EncodingHeader yang dinormalisasi juga disertakan karena pengaturan objek terkompresi cache diaktifkan. Untuk informasi selengkapnya, lihat [Dukungan kompresi](#).

- Cookie termasuk dalam kunci cache: Semua cookie disertakan.
- String kueri disertakan dalam kunci cache: Semua string kueri disertakan.
- Pengaturan objek terkompresi cache: Diaktifkan. Untuk informasi selengkapnya, lihat [Dukungan kompresi](#).

Memahami kunci cache

kunci cache menentukan apakah penampil meminta CloudFront lokasi tepi menghasilkan temuan cache. Kunci cache adalah pengidentifikasi unik untuk objek dalam cache. Setiap objek dalam cache memiliki kunci cache unik.

Terjebak terjadi ketika permintaan penampil menghasilkan kunci cache yang sama dengan permintaan sebelumnya, dan objek untuk kunci cache tersebut ada di cache lokasi edge dan valid. Ketika ada cache hit, objek yang diminta disajikan ke penampil dari lokasi CloudFront tepi, yang memiliki manfaat sebagai berikut:

- Berkurangnya beban pada server asal Anda
- Berkurangnya latensi untuk penampil

Anda bisa mendapatkan kinerja yang lebih baik dari situs web atau aplikasi saat Anda memiliki rasio tembolok (proporsi yang lebih tinggi dari permintaan penampil yang menghasilkan ketukan cache). Salah satu cara untuk meningkatkan rasio ketukan cache Anda adalah dengan hanya memasukkan nilai minimum yang diperlukan dalam kunci cache. Untuk informasi lebih lanjut, lihat bagian berikut.

Anda dapat memodifikasi nilai (string kueri URL, header HTTP, dan cookie) dalam kunci cache menggunakan [kebijakan cache](#). (Anda juga dapat memodifikasi kunci cache dengan menggunakan [Lambda @Edgefunction](#) atau [CloudFront Fungsi](#) pada permintaan penampil.) Sebelum memodifikasi kunci cache, penting untuk memahami bagaimana aplikasi Anda dirancang dan kapan dan bagaimana aplikasi tersebut dapat memberikan respons yang berbeda berdasarkan karakteristik permintaan penampil. Jika suatu nilai dalam permintaan penampil menentukan respons bahwa asal Anda akan kembali, Anda harus menyertakan nilai tersebut dalam kunci cache. Tetapi jika Anda menyertakan nilai dalam kunci cache yang tidak memengaruhi respons bahwa asal Anda kembali, Anda mungkin akhirnya menyimpan objek duplikat.

Kunci cache default

Secara default, kunci cache untuk CloudFront distribusi mencakup informasi berikut:

- Nama domain CloudFront distribusi (misalnya, d111111abcdef8.cloudfront.net)
- Jalur URL objek yang diminta (misalnya, /content/stories/example-story.html)

 Note

OPTIONSMetode ini termasuk dalam kunci cache untuk OPTIONS permintaan. Ini berarti bahwa respons terhadap OPTIONS permintaan di-cache secara terpisah dari respons GET dan HEAD permintaan.

Nilai lain dari permintaan penampil tidak disertakan dalam kunci cache, secara default. Pertimbangkan permintaan HTTP berikut dari peramban web.

```
HTTP/1.1 GET /content/stories/example-story.html?ref=0123abc&split-pages=false
Host: d111111abcdef8.cloudfront.net
User-Agent: Mozilla/5.0 Gecko/20100101 Firefox/68.0
Accept: text/html, */*
Accept-Language: en-US,en
Cookie: session_id=01234abcd
Referer: https://news.example.com/
```

Saat permintaan penampil seperti contoh ini masuk ke lokasi CloudFront tepi, CloudFront gunakan kunci cache untuk menentukan apakah ada cache yang terkena. Secara default, hanya komponen permintaan berikut yang disertakan dalam kunci cache: `/content/stories/example-story.html` dan `d111111abcdef8.cloudfront.net`. Jika objek yang diminta tidak ada dalam cache (cache hilang), maka CloudFront kirimkan permintaan ke asal untuk mendapatkan objek. Setelah mendapatkan objek, CloudFront mengembalikannya ke penampil dan menyimpannya di cache lokasi tepi.

Ketika CloudFront menerima permintaan lain untuk objek yang sama, sebagaimana ditentukan oleh kunci cache, CloudFront menyajikan objek cache ke penampil segera, tanpa mengirim permintaan ke asal. Misalnya, pertimbangkan permintaan HTTP berikut yang muncul setelah permintaan sebelumnya.

```
HTTP/1.1 GET /content/stories/example-story.html?ref=xyz987&split-pages=true
Host: d111111abcdef8.cloudfront.net
User-Agent: Mozilla/5.0 AppleWebKit/537.36 Chrome/83.0.4103.116
Accept: text/html, */*
```

```
Accept-Language: en-US,en  
Cookie: session_id=wxyz9876  
Referer: https://rss.news.example.net/
```

Permintaan ini untuk objek yang sama seperti permintaan sebelumnya, tetapi berbeda dengan permintaan sebelumnya. Ada string kueri URL yang berbeda, berbeda User-Agent dan Referer dan session_id cookie. Namun, nilai-nilai ini tidak menjadi bagian dari kunci cache secara default, sehingga permintaan kedua ini menghasilkan temuan tembolok.

Sesuaikan kunci cache

Dalam beberapa kasus, Anda mungkin ingin memasukkan lebih banyak informasi dalam kunci cache, meskipun melakukannya mungkin hanya akan menghasilkan lebih sedikit ketukan cache. Anda menentukan apa yang harus disertakan dalam kunci cache menggunakan [kebijakan cache](#).

Misalnya, jika server asal Anda menggunakan Accept-Language Header HTTP dalam permintaan penampil untuk mengembalikan konten yang berbeda berdasarkan bahasa penampil, Anda mungkin ingin menyertakan header ini dalam tombol cache. Saat Anda melakukannya, CloudFront gunakan header ini untuk menentukan klik cache, dan sertakan header dalam permintaan asal (permintaan yang CloudFront dikirim ke asal saat ada cache yang hilang).

Salah satu konsekuensi potensial yang mencakup nilai tambahan dalam kunci cache adalah bahwa CloudFront mungkin berakhir saat menyimpan objek duplikat karena variasi yang dapat terjadi dalam permintaan penampil. Misalnya, penampil dapat mengirim nilai-nilai berikut untuk header Accept-Language:

- en-US,en
- en,en-US
- en-US, en
- en-US

Semua nilai yang berbeda ini menunjukkan bahwa bahasa pemirsa adalah bahasa Inggris, tetapi variasi dapat CloudFront menyebabkan cache objek yang sama beberapa kali. Hal ini dapat mengurangi temuan cache dan meningkatkan jumlah permintaan asal usul. Anda dapat menghindari duplikasi ini dengan tidak menyertakan Accept-Language header dalam kunci cache, dan sebagai gantinya mengonfigurasi situs web atau aplikasi Anda untuk menggunakan konten yang

berbeda URLs dalam bahasa yang berbeda (misalnya, `/en-US/content/stories/example-story.html`).

Untuk setiap nilai yang ingin Anda sertakan dalam kunci cache, Anda harus memastikan bahwa Anda memahami berapa banyak variasi nilai tersebut yang mungkin muncul dalam permintaan penampil. Untuk nilai permintaan tertentu, tidak ada salahnya memasukkan nilai tersebut dalam kunci cache. Misalnya, `User-Agent` header dapat memiliki ribuan variasi unik, jadi umumnya bukan merupakan kandidat yang baik untuk dimasukkan dalam kunci cache. Cookie yang memiliki nilai khusus pengguna atau khusus sesi dan unik untuk ribuan (atau bahkan jutaan) permintaan juga bukan kandidat yang baik untuk inklusi kunci cache. Jika Anda memasukkan nilai-nilai ini ke dalam kunci cache, setiap variasi unik menghasilkan salinan objek lain di dalam cache. Jika salinan objek ini tidak unik, atau jika Anda berakhir dengan sejumlah besar objek yang sedikit berbeda, maka setiap objek hanya mendapat sejumlah kecil tembolok, Anda mungkin ingin mempertimbangkan pendekatan yang berbeda. Anda dapat mengecualikan nilai yang sangat variabel ini dari kunci cache, atau Anda dapat menandai objek sebagai tidak dapat dicache.

Berhati-hatilah saat menyesuaikan kunci cache. Terkadang hal ini diinginkan, tetapi dapat memiliki konsekuensi yang tidak diinginkan seperti menyimpan objek duplikat, menurunkan rasio hit cache Anda, dan meningkatkan jumlah permintaan asal. Jika situs web atau aplikasi asal Anda perlu menerima nilai tertentu dari permintaan penampil untuk analitik, telemetri, atau penggunaan lainnya, tetapi nilai ini tidak mengubah objek yang kembali ke objek asal, gunakan [kebijakan permintaan asal](#) untuk memasukkan nilai-nilai ini dalam permintaan asal, tetapi tidak memasukkannya ke dalam kunci cache.

Kontrol permintaan asal dengan kebijakan

Ketika permintaan penampil untuk CloudFront menghasilkan cache yang hilang (objek yang diminta tidak di-cache di lokasi tepi), CloudFront mengirimkan permintaan ke asal untuk mengambil objek. Ini disebut permintaan asal perjalanan. Permintaan asal usul selalu menyertakan informasi berikut dari permintaan penampil:

- Jalur URL (jalur saja, tanpa string kueri URL atau nama domain)
- Isi permohonan (jika ada)
- Header HTTP yang CloudFront secara otomatis menyertakan dalam setiap permintaan asal, termasuk `Host`, `User-Agent`, dan `X-Amz-Cf-Id`

Informasi lain dari permintaan penampil, seperti string kueri URL, header HTTP, dan cookie, tidak disertakan dalam permintaan asal secara default. (Pengecualian: Dengan pengaturan cache lama, CloudFront teruskan header ke asal Anda secara default.) Namun, Anda mungkin ingin menerima beberapa informasi lain ini di tempat asal, misalnya untuk mengumpulkan data untuk analitik atau telemetri. Anda dapat menggunakan kebijakan permintaan asal untuk mengontrol informasi yang disertakan dalam permintaan asal.

Kebijakan permintaan asal terpisah dari [kebijakan cache](#), yang mengontrol kunci cache. Dengan cara ini, Anda dapat menerima informasi tambahan di tempat asal dan juga mempertahankan rasio hit cache yang baik (proporsi permintaan pemirsa yang menghasilkan hit cache). Anda melakukannya dengan secara terpisah mengontrol informasi mana saja yang termasuk dalam permintaan asal (menggunakan kebijakan permintaan asal) dan yang disertakan dalam kunci cache (menggunakan kebijakan cache).

Meskipun dua jenis kebijakan terpisah, kebijakan tersebut berkaitan. Semua string kueri URL, header HTTP, dan cookie yang Anda sertakan dalam kunci cache (menggunakan kebijakan cache) secara otomatis disertakan dalam permintaan asal. Gunakan kebijakan permintaan asal usul untuk menentukan informasi yang ingin Anda masukkan ke permintaan asal keberangkatan, tetapi tidak di dalam kunci cache. Sama seperti kebijakan cache, Anda melampirkan kebijakan permintaan asal ke satu atau beberapa perilaku cache dalam CloudFront distribusi.

Anda juga dapat menggunakan kebijakan permintaan asal untuk menambahkan header HTTP tambahan ke permintaan asal yang tidak disertakan dalam permintaan penampil. Header tambahan ini ditambahkan oleh CloudFront sebelum mengirim permintaan asal, dengan nilai header yang

ditentukan secara otomatis berdasarkan permintaan penampil. Untuk informasi selengkapnya, lihat [the section called “Tambahkan header CloudFront permintaan”](#).

Topik

- [Memahami kebijakan permintaan asal](#)
- [Buat kebijakan permintaan asal](#)
- [Gunakan kebijakan permintaan asal terkelola](#)
- [Tambahkan header CloudFront permintaan](#)
- [Memahami bagaimana kebijakan permintaan asal dan kebijakan cache bekerja sama](#)

Memahami kebijakan permintaan asal

CloudFront menyediakan beberapa kebijakan permintaan asal yang telah ditentukan sebelumnya, yang dikenal sebagai kebijakan terkelola, untuk kasus penggunaan umum. Anda dapat menggunakan kebijakan terkelola ini, atau Anda dapat membuat kebijakan permintaan asal Anda sendiri yang khusus untuk kebutuhan Anda. Untuk informasi lebih lanjut tentang kebijakan terkelola, lihat [Gunakan kebijakan permintaan asal terkelola](#).

Kebijakan permohonan asal memuat pengaturan berikut, yang dikategorikan menjadi informasi kebijakan dan pengaturan permintaan asal.

Informasi kebijakan

Nama

Nama untuk mengidentifikasi kebijakan permintaan asal usul. Pada konsol, Anda menggunakan nama untuk melampirkan kebijakan permintaan asal untuk perilaku cache.

Deskripsi

Komentar untuk menguraikan kebijakan permintaan asal. Ini opsional.

Pengaturan permintaan asal

Pengaturan permintaan asal menentukan nilai dalam permintaan penampil yang disertakan dalam permintaan yang CloudFront dikirim ke asal (dikenal sebagai permintaan asal). Nilai dapat mencakup string kueri URL, header HTTP, dan cookie. Nilai yang Anda tetapkan termasuk dalam permintaan

asal, tetapi tidak termasuk dalam kunci cache. Untuk informasi tentang pengontrolan kunci cache, lihat [Mengontrol kunci cache dengan kebijakan](#).

Header

Header HTTP dalam permintaan penampil yang CloudFront menyertakan permintaan asal. Untuk header, Anda dapat memilih salah satu pengaturan berikut:

- Tidak ada – Header HTTP dalam permintaan penampil adalah tidak yang termasuk dalam permintaan asal usul.
- Semua header penampil – Semua header HTTP di permintaan penampil disertakan dalam permintaan asal.
- Semua header penampil dan CloudFront header berikut - Semua header HTTP dalam permintaan penampil disertakan dalam permintaan asal. Selain itu, Anda menentukan CloudFront header mana yang ingin Anda tambahkan ke permintaan asal. Untuk informasi selengkapnya tentang CloudFront header, lihat [the section called “Tambahkan header CloudFront permintaan”](#).
- Sertakan header berikut - Anda menentukan header HTTP mana yang disertakan dalam permintaan asal.

Note

Jangan tentukan header yang sudah disertakan dalam pengaturan Origin Custom Header Anda. Untuk informasi selengkapnya, lihat [Konfigurasi CloudFront untuk menambahkan header khusus ke permintaan asal](#).

- Semua header penampil kecuali - Anda menentukan header HTTP mana yang tidak termasuk dalam permintaan asal. Semua header HTTP lainnya dalam permintaan penampil, kecuali yang ditentukan, disertakan.

Saat Anda menggunakan header Semua penampil dan header berikut, Sertakan CloudFront header berikut, atau Semua header penampil kecuali setelah, Anda menentukan header HTTP hanya dengan nama header. CloudFront termasuk header lengkap, termasuk nilainya, dalam permintaan asal.

 Note

Saat Anda menggunakan header Semua penampil kecuali setelan untuk menghapus Host header penampil, CloudFront tambahkan Host header baru dengan nama domain asal ke permintaan asal.

Cookie

Cookie dalam permintaan penampil yang CloudFront mencakup permintaan asal. Untuk cookie, Anda dapat memilih salah satu pengaturan berikut:

- Tidak ada – Cookie di permintaan penampil adalah tidak yang termasuk dalam permintaan asal usul.
- Semua – Semua cookie di permintaan pemirsa disertakan dalam permintaan asal.
- Sertakan cookie berikut - Anda menentukan cookie mana dalam permintaan penampil yang disertakan dalam permintaan asal.
- Semua cookie kecuali — Anda menentukan cookie mana dalam permintaan penampil yang tidak termasuk dalam permintaan asal. Semua cookie lain dalam permintaan pemirsa disertakan.

Ketika Anda menggunakan Sertakan cookie berikut atau Semua cookie kecuali pengaturan, Anda menentukan cookie dengan nama mereka saja. CloudFront termasuk cookie lengkap, termasuk nilainya, dalam permintaan asal.

String kueri

String kueri URL dalam permintaan penampil yang CloudFront menyertakan permintaan asal. Untuk string kueri, Anda dapat memilih salah satu pengaturan berikut:

- Tidak ada – String kueri pada permintaan pemirsa adalah tidak yang termasuk dalam permintaan asal usul.
- Semua – Semua string kueri dalam permintaan penampil akan disertakan dalam permintaan asal.
- Sertakan string kueri berikut - Anda menentukan string kueri mana dalam permintaan penampil yang disertakan dalam permintaan asal.
- Semua string kueri kecuali - Anda menentukan string kueri mana dalam permintaan penampil yang tidak termasuk dalam permintaan asal. Semua string kueri lainnya disertakan.

Bila Anda menggunakan Sertakan string kueri berikut atau Semua string kueri kecuali setelan, Anda menentukan string kueri berdasarkan namanya saja. CloudFront termasuk string kueri lengkap, termasuk nilainya, dalam permintaan asal.

Buat kebijakan permintaan asal

Anda dapat menggunakan kebijakan permintaan asal untuk mengontrol nilai (string kueri URL, header HTTP, dan cookie) yang disertakan dalam permintaan yang CloudFront dikirim ke asal Anda. Anda dapat membuat kebijakan permintaan asal di CloudFront konsol, dengan AWS Command Line Interface (AWS CLI), atau dengan CloudFront API.

Setelah membuat kebijakan permintaan asal, Anda melampirkannya ke satu atau beberapa perilaku cache dalam CloudFront distribusi.

Kebijakan permintaan asal tidak diperlukan. Saat perilaku cache tidak memiliki kebijakan permintaan asal yang dilampirkan, permintaan asal mencakup semua nilai yang ditentukan dalam [kebijakan cache](#), tetapi tidak lebih.

Note

Untuk menggunakan kebijakan permintaan asal, perilaku cache juga harus menggunakan [kebijakan cache](#). Anda tidak dapat menggunakan kebijakan permintaan asal dalam perilaku cache tanpa kebijakan cache.

Console

Untuk membuat kebijakan permintaan asal (konsol)

1. Masuk ke Konsol Manajemen AWS dan buka halaman Kebijakan di CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home?#/policies>.
2. Pilih Permintaan asal, lalu pilih Buat kebijakan permintaan asal.
3. Pilih pengaturan yang diinginkan untuk kebijakan permintaan asal ini. Untuk informasi selengkapnya, lihat [Memahami kebijakan permintaan asal](#).
4. Setelah selesai, pilih Buat.

Setelah membuat kebijakan permintaan asal, Anda dapat melampirkannya ke perilaku cache.

Untuk melampirkan kebijakan permintaan asal ke distribusi yang ada (konsol)

1. Buka halaman Distribusi di CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home#/distributions>.
2. Pilih distribusi untuk diperbarui, lalu pilih Perilaku tab.
3. Pilih perilaku cache untuk diperbarui, lalu pilih Edit.

Atau, untuk membuat perilaku cache baru, pilih Buat perilaku.

4. Di bagian Kunci cache dan permintaan asal, pastikan kebijakan Cache dan kebijakan permintaan asal dipilih.
5. Untuk kebijakan permintaan Origin, pilih kebijakan permintaan asal untuk dilampirkan ke perilaku cache ini.
6. Di bagian bawah halaman, pilih Simpan perubahan.

Untuk melampirkan kebijakan permintaan asal usul ke distribusi baru (konsol)

1. Buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Pilih Buat Distribusi.
3. Di bagian Kunci cache dan permintaan asal, pastikan kebijakan Cache dan kebijakan permintaan asal dipilih.
4. Untuk kebijakan permintaan Origin, pilih kebijakan permintaan asal untuk dilampirkan ke perilaku cache default distribusi ini.
5. Pilih pengaturan yang diinginkan untuk asal, perilaku cache default, dan pengaturan distribusi lainnya. Untuk informasi selengkapnya, lihat [Semua referensi pengaturan distribusi](#).
6. Setelah selesai, pilih Buat distribusi.

CLI

Untuk membuat kebijakan permintaan asal dengan AWS Command Line Interface (AWS CLI), gunakan `aws cloudfront create-origin-request-policy` perintah. Anda dapat menggunakan file input untuk memberikan parameter input perintah, daripada menentukan setiap parameter individu sebagai input baris perintah.

Untuk membuat kebijakan permintaan asal (CLI dengan file input)

1. Gunakan perintah berikut untuk membuat file dengan nama `origin-request-policy.yaml` yang berisi semua parameter input untuk `create-origin-request-policy` perintah.

```
aws cloudfront create-origin-request-policy --generate-cli-skeleton yml-input >
origin-request-policy.yaml
```

2. Buka file dengan nama `origin-request-policy.yaml` yang baru Anda buat. Edit file untuk menentukan pengaturan kebijakan permintaan asal yang diinginkan, lalu simpan file. Anda dapat menghapus bidang opsional dari file, tetapi jangan menghapus bidang yang diperlukan.

Untuk informasi selengkapnya tentang pengaturan kebijakan permintaan asal, lihat [Memahami kebijakan permintaan asal](#).

3. Gunakan perintah berikut untuk membuat kebijakan permintaan asal dengan menggunakan parameter input dari `origin-request-policy.yaml` file.

```
aws cloudfront create-origin-request-policy --cli-input-yml file://origin-
request-policy.yaml
```

Catat Id nilai dalam output perintah. Ini adalah ID kebijakan permintaan asal, dan Anda memerlukannya untuk melampirkan kebijakan permintaan asal ke perilaku cache CloudFront distribusi.

Untuk melampirkan kebijakan permintaan asal ke distribusi yang ada (CLI dengan file masukan)

1. Gunakan perintah berikut untuk menyimpan konfigurasi distribusi untuk CloudFront distribusi yang ingin Anda perbarui. Ganti *distribution_ID* dengan ID distribusi.

```
aws cloudfront get-distribution-config --id distribution_ID --output yml >
dist-config.yaml
```

2. Buka file dengan nama `dist-config.yaml` yang baru Anda buat. Edit file, membuat perubahan berikut pada setiap perilaku cache yang Anda perbarui untuk menggunakan kebijakan permintaan asal usul.
 - Dalam perilaku cache, tambahkan bidang bernama `OriginRequestPolicyId`. Untuk nilai bidang, gunakan ID kebijakan permintaan asal yang Anda catat setelah membuat kebijakan.
 - Ubah nama `ETag` bidang menjadi `IfMatch`, tetapi jangan ubah nilai bidang.

Simpan file setelah selesai.

3. Gunakan perintah berikut untuk memperbarui distribusi untuk menggunakan kebijakan permintaan asal usul. Ganti `distribution_ID` dengan ID distribusi.

```
aws cloudfront update-distribution --id distribution_ID --cli-input-yaml file://  
dist-config.yaml
```

Untuk melampirkan kebijakan permintaan asal ke distribusi baru (CLI dengan file input)

1. Gunakan perintah berikut untuk membuat file dengan nama `distribution.yaml` yang berisi semua parameter input untuk `create-distribution` perintah.

```
aws cloudfront create-distribution --generate-cli-skeleton yaml-input >  
distribution.yaml
```

2. Buka file dengan nama `distribution.yaml` yang baru Anda buat. Dalam perilaku cache default, di `OriginRequestPolicyId`, masukkan ID kebijakan permintaan asal yang Anda catat setelah membuat kebijakan. Lanjutkan mengedit file untuk menentukan pengaturan distribusi yang Anda inginkan, kemudian simpan file setelah selesai.

Untuk informasi lebih lanjut tentang pengaturan distribusi, lihat [Semua referensi pengaturan distribusi](#).

3. Gunakan perintah berikut untuk membuat distribusi menggunakan parameter input dari `distribution.yaml` file Anda.

```
aws cloudfront create-distribution --cli-input-yaml file://distribution.yaml
```

API

Untuk membuat kebijakan permintaan asal dengan CloudFront API, gunakan [CreateOriginRequestPolicy](#). Untuk informasi selengkapnya tentang bidang yang Anda tentukan dalam panggilan API ini, lihat [Memahami kebijakan permintaan asal](#) dan dokumentasi referensi API untuk AWS SDK atau klien API lainnya.

Setelah Anda membuat kebijakan permintaan asal, Anda dapat melampirkannya ke perilaku cache, menggunakan salah satu panggilan API berikut:

- Untuk memasangnya ke perilaku cache dalam distribusi yang ada, gunakan [UpdateDistribution](#).
- Untuk memasangnya ke perilaku cache dalam distribusi baru, gunakan [CreateDistribution](#).

Untuk kedua panggilan API ini, berikan ID kebijakan permintaan asal di `OriginRequestPolicyId` bidang, di dalam perilaku cache. Untuk informasi selengkapnya tentang bidang lain yang Anda tentukan dalam panggilan API ini, lihat [Semua referensi pengaturan distribusi](#) dan dokumentasi referensi API untuk AWS SDK atau klien API lainnya.

Gunakan kebijakan permintaan asal terkelola

CloudFront menyediakan serangkaian kebijakan permintaan asal terkelola yang dapat Anda lampirkan ke salah satu perilaku cache distribusi Anda. Dengan kebijakan permintaan asal terkelola, Anda tidak perlu menulis atau mempertahankan kebijakan permintaan asal Anda sendiri. Kebijakan terkelola menggunakan pengaturan yang dioptimalkan untuk kasus penggunaan spesifik.

Untuk menggunakan kebijakan permintaan asal terkelola, Anda melampirkannya ke perilaku cache dalam distribusi Anda. Prosesnya sama seperti ketika Anda membuat kebijakan permintaan asal, tetapi alih-alih membuat yang baru, Anda hanya melampirkan salah satu kebijakan permintaan asal terkelola. Anda melampirkan kebijakan baik dengan nama (dengan konsol) atau dengan ID (dengan AWS CLI atau SDKs). Nama-nama dan IDs tercantum di bagian berikut.

Untuk informasi selengkapnya, lihat [Buat kebijakan permintaan asal](#).

Topik berikut menjelaskan kebijakan permintaan asal terkelola yang dapat Anda gunakan.

Topik

- [AllViewer](#)
- [AllViewerAndCloudFrontHeaders-2022-06](#)
- [AllViewerExceptHostHeader](#)
- [CORS- CustomOrigin](#)
- [CORS-S3asal](#)
- [Elemental- - MediaTailor PersonalizedManifests](#)
- [HostHeaderOnly](#)
- [UserAgentRefererHeaders](#)

AllViewer

[Lihat kebijakan ini di CloudFront konsol](#)

Kebijakan ini mencakup semua nilai (header, cookie, dan string kueri) dari permintaan penampil.

Saat menggunakan CloudFormation, the AWS CLI, atau CloudFront API, ID untuk kebijakan ini adalah:

216adef6-5c7f-47e4-b989-5492eafa07d3

Kebijakan ini memiliki pengaturan berikut:

- Header yang disertakan dalam permintaan asal: Semua header di permintaan penampil
- Cookie yang disertakan dalam permintaan asal: Semua
- String kueri yang disertakan dalam permintaan asal: Semua

AllViewerAndCloudFrontHeaders-2022-06

[Lihat kebijakan ini di CloudFront konsol](#)

Kebijakan ini mencakup semua nilai (header, cookie, dan string kueri) dari permintaan pemirsa, dan semua [CloudFront header](#) yang dirilis hingga Juni 2022 (CloudFront header yang dirilis setelah Juni 2022 tidak disertakan).

Saat menggunakan CloudFormation, the AWS CLI, atau CloudFront API, ID untuk kebijakan ini adalah:

33f36d7e-f396-46d9-90e0-52428a34d9dc

Kebijakan ini memiliki pengaturan berikut:

- Header yang disertakan dalam permintaan asal: Semua header dalam permintaan penampil, dan header berikut: CloudFront
 - CloudFront-Forwarded-Proto
 - CloudFront-Is-Android-Viewer
 - CloudFront-Is-Desktop-Viewer
 - CloudFront-Is-IOS-Viewer
 - CloudFront-Is-Mobile-Viewer
 - CloudFront-Is-SmartTV-Viewer
 - CloudFront-Is-Tablet-Viewer
 - CloudFront-Viewer-Address
 - CloudFront-Viewer-ASN
 - CloudFront-Viewer-City
 - CloudFront-Viewer-Country
 - CloudFront-Viewer-Country-Name
 - CloudFront-Viewer-Country-Region
 - CloudFront-Viewer-Country-Region-Name
 - CloudFront-Viewer-Http-Version
 - CloudFront-Viewer-Latitude
 - CloudFront-Viewer-Longitude
 - CloudFront-Viewer-Metro-Code
 - CloudFront-Viewer-Postal-Code
 - CloudFront-Viewer-Time-Zone
 - CloudFront-Viewer-TLS
- Cookie yang disertakan dalam permintaan asal: Semua
- String kueri yang disertakan dalam permintaan asal: Semua

AllViewerExceptHostHeader

Kebijakan ini tidak menyertakan Host header dari permintaan penampil, tetapi menyertakan semua nilai lainnya (header, cookie, dan string kueri) dari permintaan penampil.

Kebijakan ini juga mencakup [header CloudFront permintaan](#) tambahan untuk protokol HTTP, versi HTTP, versi TLS, dan semua jenis perangkat dan header lokasi penampil.

Kebijakan ini ditujukan untuk digunakan dengan Amazon API Gateway dan asal URL AWS Lambda fungsi. Asal ini mengharapkan Host header berisi nama domain asal, bukan nama domain CloudFront distribusi. Meneruskan Host header dari permintaan penampil ke asal-usul ini dapat mencegahnya berfungsi.

Note

Saat Anda menggunakan kebijakan permintaan asal terkelola ini untuk menghapus Host header penampil, CloudFront tambahkan Host header baru dengan nama domain asal ke permintaan asal.

Saat menggunakan CloudFormation, the AWS CLI, atau CloudFront API, ID untuk kebijakan ini adalah:

b689b0a8-53d0-40ab-baf2-68738e2966ac

Kebijakan ini memiliki pengaturan berikut:

- Header termasuk dalam permintaan asal: Semua header dalam permintaan penampil kecuali untuk header Host
- Cookie yang disertakan dalam permintaan asal: Semua
- String kueri yang disertakan dalam permintaan asal: Semua

CORS- CustomOrigin

[Lihat kebijakan ini di CloudFront konsol](#)

Kebijakan ini mencakup tajuk yang memungkinkan permintaan pembagian sumber daya lintas negara asal (CORS) ketika asal merupakan asal usul khusus.

Saat menggunakan CloudFormation, the AWS CLI, atau CloudFront API, ID untuk kebijakan ini adalah:

59781a5b-3903-41f3-afcb-af62929ccde1

Kebijakan ini memiliki pengaturan berikut:

- Header termasuk dalam permintaan asal:
 - Origin
- Cookie yang disertakan dalam permintaan asal: Tidak ada
- String kueri yang disertakan dalam permintaan asal: Tidak ada

CORS-S3asal

[Lihat kebijakan ini di CloudFront konsol](#)

Kebijakan ini mencakup header yang memungkinkan permintaan pembagian sumber daya lintas negara asal (CORS) ketika asalnya adalah keranjang Amazon S3.

Saat menggunakan CloudFormation, the AWS CLI, atau CloudFront API, ID untuk kebijakan ini adalah:

88a5eaf4-2fd4-4709-b370-b4c650ea3fcf

Kebijakan ini memiliki pengaturan berikut:

- Header termasuk dalam permintaan asal:
 - Origin
 - Access-Control-Request-Headers
 - Access-Control-Request-Method
- Cookie yang disertakan dalam permintaan asal: Tidak ada
- String kueri yang disertakan dalam permintaan asal: Tidak ada

Elemental- - MediaTailor PersonalizedManifests

[Lihat kebijakan ini di CloudFront konsol](#)

Kebijakan ini dimaksudkan untuk digunakan dengan asal yang merupakan AWS Elemental MediaTailor titik akhir.

Saat menggunakan CloudFormation, the AWS CLI, atau CloudFront API, ID untuk kebijakan ini adalah:

775133bc-15f2-49f9-abea-afb2e0bf67d2

Kebijakan ini memiliki pengaturan berikut:

- Header termasuk dalam permintaan asal:
 - Origin
 - Access-Control-Request-Headers
 - Access-Control-Request-Method
 - User-Agent
 - X-Forwarded-For
- Cookie yang disertakan dalam permintaan asal: Tidak ada
- String kueri yang disertakan dalam permintaan asal: Semua

HostHeaderOnly

[Lihat kebijakan ini di CloudFront konsol](#)

Kebijakan ini hanya mencakup Host header dari permintaan asal. Itu tidak termasuk string kueri atau cookie.

Saat menggunakan CloudFormation, the AWS CLI, atau CloudFront API, ID untuk kebijakan ini adalah:

bf0718e1-ba1e-49d1-88b1-f726733018ae

Kebijakan ini memiliki pengaturan berikut:

- Header termasuk dalam permintaan asal: Host
- Cookie yang disertakan dalam permintaan asal: Tidak ada
- String kueri yang disertakan dalam permintaan asal: Tidak ada

UserAgentRefererHeaders

[Lihat kebijakan ini di CloudFront konsol](#)

Kebijakan ini hanya mencakup `User-Agent` dan `Referer` yang berbeda. Itu tidak termasuk string kueri atau cookie.

Saat menggunakan CloudFormation, the AWS CLI, atau CloudFront API, ID untuk kebijakan ini adalah:

```
acba4595-bd28-49b8-b9fe-13317c0390fa
```

Kebijakan ini memiliki pengaturan berikut:

- Header termasuk dalam permintaan asal:
 - `User-Agent`
 - `Referer`
- Cookie yang disertakan dalam permintaan asal: Tidak ada
- String kueri yang disertakan dalam permintaan asal: Tidak ada

Tambahkan header CloudFront permintaan

Anda dapat mengonfigurasi CloudFront untuk menambahkan header HTTP tertentu ke permintaan yang CloudFront diterima dari pemirsa dan meneruskan ke fungsi asal atau [tepi](#) Anda. Nilai header HTTP ini didasarkan pada karakteristik penampil atau permintaan penampil. Header memberikan informasi tentang jenis perangkat penampil, alamat IP, lokasi geografis, protokol permintaan (HTTP atau HTTPS), versi HTTP, detail koneksi TLS, [JA3 sidik jari, dan sidik jari](#). JA4 Anda juga dapat mengonfigurasi perilaku cache distribusi Anda untuk meneruskan WebSocket header. Untuk informasi selengkapnya, lihat [Gunakan WebSockets dengan CloudFront distribusi](#).

Dengan header ini, asal Anda atau fungsi tepi Anda dapat menerima informasi tentang penampil tanpa perlu Anda menulis kode Anda sendiri untuk menentukan informasi ini. Jika asal Anda mengembalikan respons yang berbeda berdasarkan informasi di header ini, Anda dapat memasukkannya ke dalam kunci CloudFront cache sehingga menyimpan respons secara terpisah. Misalnya, asal Anda mungkin merespons dengan konten dalam bahasa tertentu berdasarkan negara tempat pemirsa berada, atau dengan konten yang disesuaikan dengan jenis perangkat tertentu. Origin Anda mungkin juga menulis header ini ke file log, yang dapat Anda gunakan untuk menentukan informasi tentang tempat pemirsa Anda berada, jenis perangkat yang mereka gunakan, dan banyak lagi.

Untuk menyertakan header ini di tombol cache, gunakan kebijakan cache. Untuk informasi selengkapnya, lihat [Mengontrol kunci cache dengan kebijakan](#) dan [the section called “Memahami kunci cache”](#).

Untuk menerima header ini di asal Anda tetapi tidak memasukkannya ke dalam kunci cache, gunakan kebijakan permintaan asal. Untuk informasi selengkapnya, lihat [Kontrol permintaan asal dengan kebijakan](#).

Topik

- [Header jenis perangkat](#)
- [Header lokasi penampil](#)
- [Header untuk menentukan struktur header pemirsa](#)
- [Header terkait TLS](#)
- [CloudFront Header lainnya](#)

Header jenis perangkat

Anda dapat menambahkan header berikut untuk menentukan jenis perangkat pemirsa. Berdasarkan nilai User-Agent header, CloudFront tetapkan nilai header ini ke `true` atau `false`. Jika perangkat jatuh ke dalam lebih dari satu kategori, lebih dari satu nilai bisa `true`. Misalnya, untuk beberapa perangkat tablet, CloudFront atur keduanya `CloudFront-Is-Mobile-Viewer` dan `CloudFront-Is-Tablet-Viewer` ke `true`.

- `CloudFront-Is-Android-Viewer` – Dipasang pada `true` saat CloudFront menentukan bahwa penampil adalah perangkat dengan sistem operasi Android.
- `CloudFront-Is-Desktop-Viewer` – Dipasang pada `true` saat CloudFront menentukan bahwa penampil adalah perangkat desktop.
- `CloudFront-Is-IOS-Viewer`— Setel ke `true` kapan CloudFront menentukan bahwa penampil adalah perangkat dengan sistem operasi seluler Apple, seperti iPhone, iPod touch, dan beberapa perangkat iPad.
- `CloudFront-Is-Mobile-Viewer` – Dipasang pada `true` saat CloudFront menentukan bahwa penampil adalah perangkat seluler.
- `CloudFront-Is-SmartTV-Viewer` – Dipasang pada `true` saat CloudFront menentukan bahwa pengunjung adalah TV pintar.

- `CloudFront-Is-Tablet-Viewer` – Dipasang pada `true` saat CloudFront menentukan apakah penampil adalah tablet.

Header lokasi penampil

Anda dapat menambahkan header berikut untuk menentukan lokasi pemirsa. CloudFront menentukan nilai untuk header ini berdasarkan alamat IP penampil. [Untuk karakter non-ASCII dalam nilai header ini, CloudFront persentase mengkodekan karakter menurut bagian 1.2 dari RFC 3986.](#)

- `CloudFront-Viewer-Address`— Berisi alamat IP penampil dan port sumber permintaan. Misalnya, nilai header `198.51.100.10:46532` berarti alamat IP penampil adalah `198.51.100.10` dan port sumber permintaan adalah `46532`.
- `CloudFront-Viewer-ASN`— Berisi nomor sistem otonom (ASN) dari pemirsa.

Note

`CloudFront-Viewer-Address` dan `CloudFront-Viewer-ASN` dapat ditambahkan dalam kebijakan permintaan asal, tetapi tidak dalam kebijakan cache.

- `CloudFront-Viewer-Country`— Berisi kode negara dua huruf untuk negara pemirsa. Untuk daftar kode negara, lihat [ISO 3166-1 alpha-2](#).
- `CloudFront-Viewer-City`— Berisi nama kota pemirsa.

Saat Anda menambahkan header berikut, CloudFront terapkan ke semua permintaan kecuali yang berasal dari jaringan: AWS

- `CloudFront-Viewer-Country-Name`— Berisi nama negara pemirsa.
- `CloudFront-Viewer-Country-Region`— Berisi kode (hingga tiga karakter) yang mewakili wilayah pemirsa. Wilayah ini adalah subdivisi tingkat pertama (terluas atau paling tidak spesifik) dari [kode ISO 3166-2](#).
- `CloudFront-Viewer-Country-Region-Name`— Berisi nama wilayah pemirsa. Wilayah ini adalah subdivisi tingkat pertama (terluas atau paling tidak spesifik) dari [kode ISO 3166-2](#).
- `CloudFront-Viewer-Latitude`— Berisi perkiraan garis lintang pemirsa.
- `CloudFront-Viewer-Longitude`— Berisi perkiraan bujur pemirsa.
- `CloudFront-Viewer-Metro-Code`— Berisi kode metro pemirsa. Ini hanya ada saat penampil berada di Amerika Serikat.

- `CloudFront-Viewer-Postal-Code`— Berisi kode pos pemirsa.
- `CloudFront-Viewer-Time-Zone` Berisi zona waktu pemirsa, dalam [format database zona waktu IANA](#) (misalnya, `America/Los_Angeles`).

Note

`CloudFront-Viewer-City`, `CloudFront-Viewer-Metro-Code`, dan `CloudFront-Viewer-Postal-Code` mungkin tidak tersedia untuk setiap alamat IP. Beberapa alamat IP tidak dapat geolokasi dengan spesifisitas yang cukup untuk mendapatkan informasi itu.

Header untuk menentukan struktur header pemirsa

Anda dapat menambahkan header berikut untuk membantu mengidentifikasi penampil berdasarkan header yang dikirimkannya. Misalnya, browser yang berbeda dapat mengirim header HTTP dalam urutan tertentu. Jika browser yang ditentukan di `User-Agent` header tidak cocok dengan urutan header yang diharapkan browser tersebut, Anda dapat menolak permintaan tersebut. Selain itu, jika `CloudFront-Viewer-Header-Count` nilainya tidak sesuai dengan jumlah header `CloudFront-Viewer-Header-Order`, Anda dapat menolak permintaan tersebut.

- `CloudFront-Viewer-Header-Order`— Berisi nama header pemirsa dalam urutan yang diminta, dipisahkan oleh titik dua. Sebagai contoh: `CloudFront-Viewer-Header-Order: Host:User-Agent:Accept:Accept-Encoding`. Header di luar batas karakter 7.680 terpotong.
- `CloudFront-Viewer-Header-Count`— Berisi jumlah total header pemirsa.

Header terkait TLS

Anda dapat menambahkan header berikut untuk menentukan JA3 sidik jari, sidik jari, JA4 dan detail koneksi TLS pemirsa:

- `CloudFront-Viewer-JA3-Fingerprint`— Berisi [JA3 sidik jari](#) penampil. JA3 Sidik jari dapat membantu Anda menentukan apakah permintaan tersebut berasal dari klien yang dikenal, apakah itu malware atau bot berbahaya, atau aplikasi yang diharapkan (diizinkan terdaftar).
- `CloudFront-Viewer-JA4-Fingerprint`— Berisi JA4 sidik jari penampil. Mirip dengan JA3 [JA4 sidik jari](#), [sidik jari](#) dapat membantu Anda menentukan apakah permintaan tersebut berasal dari klien yang dikenal, apakah itu malware atau bot berbahaya, atau aplikasi yang diharapkan

(diizinkan terdaftar). Anda dapat menggunakan sidik jari untuk membangun database aktor baik dan buruk yang dikenal untuk diterapkan saat memeriksa permintaan HTTP. Anda kemudian dapat memeriksa nilai header di server web aplikasi Anda atau di [Lambda @Edge CloudFront dan Fungsi Anda](#) untuk membandingkan nilai header dengan daftar sidik jari malware yang diketahui untuk memblokir klien jahat.

- **CloudFront-Viewer-TLS**— Berisi SSL/TLS versi, sandi, dan informasi tentang SSL/TLS jabat tangan yang digunakan untuk koneksi antara pemirsa dan CloudFront. Nilai header dalam format berikut:

```
SSL/TLS_version:cipher:handshake_information
```

Untuk *handshake_information*, header dapat berisi nilai-nilai berikut:

- **fullHandshake**— Jabat tangan penuh dilakukan untuk SSL/TLS sesi tersebut.
- **sessionResumed**— SSL/TLS Sesi sebelumnya dilanjutkan.
- **connectionReused**— SSL/TLS Koneksi sebelumnya digunakan kembali.

Berikut ini adalah beberapa contoh nilai untuk header ini:

```
TLSv1.3:TLS_AES_128_GCM_SHA256:sessionResumed
```

```
TLSv1.2:ECDHE-ECDSA-AES128-GCM-SHA256:connectionReused
```

```
TLSv1.1:ECDHE-RSA-AES128-SHA256:fullHandshake
```

```
TLSv1:ECDHE-RSA-AES256-SHA:fullHandshake
```

Untuk daftar lengkap kemungkinan SSL/TLS versi dan cipher yang dapat berada dalam nilai header ini, lihat [the section called “Protokol dan sandi yang didukung antara pemirsa dan CloudFront”](#)

Catatan

- JA4 Sidik jari JA3 dan sidik jari berasal dari SSL/TLS Client Hello paket. Mereka hanya hadir untuk permintaan HTTPS.

- [Untuk header terkait TLS ini, Anda dapat menambahkannya ke kebijakan permintaan asal, tetapi tidak dalam kebijakan cache.](#)

CloudFront Header lainnya

Anda dapat menambahkan header berikut untuk menentukan URI permintaan asli penampil, parameter dan nilai string kueri permintaan asli, protokol, dan versi:

- `CloudFront-Error-Uri`— Berisi URI permintaan asli yang diterima dari penampil.
- `CloudFront-Error-Args`— Berisi parameter dan nilai string permintaan permintaan asli.
- `CloudFront-Forwarded-Proto`— Berisi protokol permintaan pemirsa (HTTP atau HTTPS).
- `CloudFront-Viewer-Http-Version`— Berisi versi HTTP dari permintaan pemirsa.

Memahami bagaimana kebijakan permintaan asal dan kebijakan cache bekerja sama

Anda dapat menggunakan [kebijakan permintaan CloudFront asal](#) untuk mengontrol permintaan yang CloudFront dikirim ke asal, yang disebut permintaan asal. Untuk menggunakan kebijakan permintaan asal, Anda harus melampirkan [kebijakan cache](#) ke perilaku cache yang sama. Anda tidak dapat menggunakan kebijakan permintaan asal dalam perilaku cache tanpa kebijakan cache. Untuk informasi selengkapnya, lihat [Kontrol permintaan asal dengan kebijakan](#).

Kebijakan permintaan asal dan kebijakan cache bekerja sama untuk menentukan nilai yang CloudFront disertakan dalam permintaan asal. Semua string kueri URL, header HTTP, dan cookie yang Anda tentukan dalam kunci cache (menggunakan kebijakan cache) secara otomatis disertakan dalam permintaan asal. Setiap string kueri tambahan, header, dan cookie yang Anda tentukan dalam kebijakan permintaan asal juga disertakan dalam permintaan asal (tetapi tidak dalam kunci cache).

Kebijakan permintaan asal dan kebijakan cache memiliki pengaturan yang mungkin tampak bertentangan satu sama lain. Misalnya, satu kebijakan mungkin mengizinkan nilai tertentu sementara kebijakan lain memblokirnya. Tabel berikut menjelaskan nilai mana yang CloudFront disertakan dalam permintaan asal saat Anda menggunakan setelan kebijakan permintaan asal dan kebijakan cache secara bersamaan. Pengaturan ini umumnya berlaku untuk semua jenis nilai (string kueri, header, dan cookie), dengan pengecualian bahwa Anda tidak dapat menentukan semua header atau menggunakan daftar blok header dalam kebijakan cache.

	Kebijakan permintaan asal			
	Tidak ada	Semua	Izinkan daftar	Daftar blokir

Kebijakan cache

Tidak ada	Tidak ada nilai dari permintaan penampil yang disertakan dalam permintaan asal, kecuali untuk default yang disertakan dalam setiap permintaan asal. Untuk informasi selengkapnya, lihat Kontrol permintaan asal dengan kebijakan .	Semua nilai dari permintaan penampil disertakan dalam permintaan asal.	Hanya nilai yang ditentukan dalam kebijakan permintaan asal yang disertakan dalam permintaan asal.	Semua nilai dari permintaan penampil kecuali yang ditentukan dalam kebijakan permintaan asal disertakan dalam permintaan asal.
Semua Catatan: Anda tidak dapat menentukan semua header dalam kebijakan cache.	Semua string kueri dan cookie dari permintaan penampil disertakan dalam permintaan asal.	Semua nilai dari permintaan penampil disertakan dalam permintaan asal.	Semua string kueri dan cookie dari permintaan penampil, dan header apa pun yang ditentukan dalam kebijakan permintaan asal, disertakan dalam permintaan asal.	Semua string kueri dan cookie dari permintaan penampil disertakan dalam permintaan asal, bahkan yang ditentukan dalam daftar blokir kebijakan permintaan asal. Pengaturan kebijakan cache akan

	Kebijakan permintaan asal			
	Tidak ada	Semua	Izinkan daftar	Daftar blokir
				mengganti daftar blokir kebijakan permintaan asal.
Izinkan daftar	Hanya nilai yang ditentukan dari permintaan penampil yang disertakan dalam permintaan asal.	Semua nilai dari permintaan penampil disertakan dalam permintaan asal.	Semua nilai yang ditentukan dalam kebijakan cache atau kebijakan permintaan asal disertakan dalam permintaan asal.	Nilai yang ditentukan dalam kebijakan cache disertakan dalam permintaan asal, meskipun nilai yang sama tersebut ditentukan dalam daftar blokir kebijakan permintaan asal. Daftar izin kebijakan cache mengesampingkan daftar blokir kebijakan permintaan asal.

	Kebijakan permintaan asal			
	Tidak ada	Semua	Izinkan daftar	Daftar blokir
Daftar blokir Catatan: Anda tidak dapat menentukan header dalam daftar blok kebijakan cache.	Semua string kueri dan cookie dari permintaan penampil kecuali yang ditentukan disertakan dalam permintaan asal.	Semua nilai dari permintaan penampil disertakan dalam permintaan asal.	Nilai yang ditentukan dalam kebijakan permintaan asal disertakan dalam permintaan asal, meskipun nilai yang sama tersebut ditentukan dalam daftar blok kebijakan cache. Daftar izin kebijakan permintaan asal mengesampingkan daftar blokir kebijakan cache.	Semua nilai dari permintaan penampil kecuali yang ditentukan dalam kebijakan cache atau kebijakan permintaan asal disertakan dalam permintaan asal.

Menambahkan atau menghapus header HTTP dalam CloudFront tanggapan dengan kebijakan

Anda dapat mengonfigurasi CloudFront untuk memodifikasi header HTTP dalam tanggapan yang dikirimkan ke pemirsa (browser web dan klien lain). CloudFront dapat menghapus header yang diterima dari asal, atau menambahkan header ke respons, sebelum mengirim respons ke pemirsa. Membuat perubahan ini tidak memerlukan penulisan kode atau mengubah asal.

Misalnya, Anda dapat menghapus header seperti `X-Powered-By` dan `Vary` agar CloudFront tidak menyertakan header ini dalam tanggapan yang dikirimkan ke pemirsa. Atau, Anda dapat menambahkan header HTTP seperti berikut ini:

- `Cache-ControlHeader` untuk mengontrol caching browser.
- `Access-Control-Allow-OriginHeader` untuk mengaktifkan berbagi sumber daya lintas asal (CORS). Anda juga dapat menambahkan header CORS lainnya.
- Satu set header keamanan umum, seperti, `Strict-Transport-SecurityContent-Security-Policy`, dan `X-Frame-Options`.
- `Server-TimingHeader` untuk melihat informasi yang terkait dengan kinerja dan perutean permintaan dan respons melalui CloudFront.

Untuk menentukan header yang CloudFront menambahkan atau menghapus dalam respons HTTP, Anda menggunakan kebijakan header respons. Anda melampirkan kebijakan header respons ke satu perilaku cache lainnya, dan CloudFront memodifikasi header dalam respons yang dikirimkan ke permintaan yang cocok dengan perilaku cache. CloudFront memodifikasi header dalam tanggapan yang dilayaninya dari cache dan yang diteruskan dari asal. Jika respons asal menyertakan satu atau beberapa header yang ditambahkan dalam kebijakan header respons, kebijakan dapat menentukan apakah CloudFront menggunakan header yang diterimanya dari asal atau menimpa header tersebut dengan header dalam kebijakan header respons.

Note

Jika Anda menambahkan header yang mengontrol caching browser ke kebijakan header respons Anda, misalnya `Cache-Control`, CloudFront tambahkan header ini ke respons penampil saja. Header ini tidak memengaruhi cara CloudFront cache objek yang diminta.

CloudFront menyediakan kebijakan header respons yang telah ditentukan sebelumnya, yang dikenal sebagai kebijakan terkelola, untuk kasus penggunaan umum. Anda dapat [menggunakan kebijakan terkelola ini](#) atau membuat kebijakan Anda sendiri. Anda dapat melampirkan kebijakan header respons tunggal ke beberapa perilaku cache di beberapa distribusi di Anda. Akun AWS

Untuk informasi selengkapnya, lihat informasi berikut:

Topik

- [Memahami kebijakan header respons](#)
- [Buat kebijakan header respons](#)
- [Menggunakan kebijakan header respons terkelola](#)

Memahami kebijakan header respons

Anda dapat menggunakan kebijakan header respons untuk menentukan header HTTP yang CloudFront dihapus atau ditambahkan Amazon dalam tanggapan yang dikirimkan ke pemirsa. Untuk informasi selengkapnya tentang kebijakan header respons dan alasan menggunakannya, lihat [Menambah atau menghapus header respons dengan kebijakan](#).

Topik berikut menjelaskan pengaturan dalam kebijakan header respons. Pengaturan dikelompokkan ke dalam kategori, yang diwakili dalam topik berikut.

Topik

- [Rincian kebijakan \(metadata\)](#)
- [Header CORS](#)
- [Header keamanan](#)
- [Header kustom](#)
- [Hapus header](#)
- [Header Pengaturan Waktu Server](#)

Rincian kebijakan (metadata)

Setelan detail kebijakan berisi metadata tentang kebijakan header respons.

- Nama — Nama untuk mengidentifikasi kebijakan header respons. Di konsol, Anda menggunakan nama untuk melampirkan kebijakan ke perilaku cache.

- Deskripsi (opsional) — Komentar untuk menjelaskan kebijakan header respons. Ini opsional, tetapi dapat membantu Anda mengidentifikasi tujuan kebijakan.

Header CORS

Setelan berbagi sumber daya lintas asal (CORS) memungkinkan Anda menambahkan dan mengonfigurasi header CORS dalam kebijakan header respons.

Daftar ini berfokus pada cara menentukan setelan dan nilai yang valid dalam kebijakan header respons. [Untuk informasi selengkapnya tentang masing-masing header ini dan bagaimana mereka digunakan untuk permintaan dan tanggapan CORS dunia nyata, lihat berbagi sumber daya lintas asal di MDN Web Docs dan spesifikasi protokol CORS.](#)

Access-Control-Allow-Credentials

Ini adalah pengaturan Boolean (`true` atau `false`) yang menentukan apakah CloudFront menambahkan `Access-Control-Allow-Credentials` header sebagai respons terhadap permintaan CORS. Saat pengaturan ini disetel ke `true`, CloudFront tambahkan `Access-Control-Allow-Credentials: true` header sebagai respons terhadap permintaan CORS. Jika CloudFront tidak, tidak menambahkan header ini ke tanggapan.

Access-Control-Allow-Headers

Menentukan nama header yang CloudFront digunakan sebagai nilai untuk `Access-Control-Allow-Headers` header dalam tanggapan terhadap permintaan preflight CORS. Nilai yang valid untuk pengaturan ini termasuk nama header HTTP atau karakter wildcard (*), yang menunjukkan bahwa semua header diperbolehkan.

Note

`AuthorizationHeader` tidak dapat menggunakan wildcard dan harus terdaftar secara eksplisit.

Contoh penggunaan karakter wildcard yang valid

Contoh	Akan cocok	Tidak akan cocok
<code>x-amz-*</code>	<code>x-amz-test</code>	<code>x-amz</code>

Contoh	Akan cocok	Tidak akan cocok
	x-amz-	
x-* -amz	x-test-amz x--amz	
*	Semua header kecuali Authorization	Authorization

Access-Control-Allow-Methods

Menentukan metode HTTP yang CloudFront digunakan sebagai nilai untuk Access-Control-Allow-Methods header dalam tanggapan terhadap permintaan preflight CORS. Nilai yang valid adalah GET,DELETE,HEAD,OPTIONS,PATCH,POST,PUT, danALL. ALL adalah nilai khusus yang mencakup semua metode HTTP yang terdaftar.

Access-Control-Allow-Origin

Menentukan nilai-nilai yang CloudFront dapat digunakan dalam header Access-Control-Allow-Origin respon. Nilai yang valid untuk pengaturan ini mencakup asal tertentu (seperti `http://www.example.com`) atau karakter wildcard (*), yang menunjukkan bahwa semua asal diperbolehkan.

Catatan

- Karakter wildcard (*) diizinkan sebagai subdomain paling kiri (). `*.example.org`
- Karakter wildcard (*) tidak diizinkan di posisi berikut:
 - Domain tingkat atas () `example.*`
 - Di sebelah kanan subdomain (`test.*.example.org`) atau dalam subdomain apa pun () `*test.example.org`
 - Di dalam istilah (`exa*mple.org`)

Untuk contoh cara menggunakan karakter wildcard, lihat tabel berikut.

Contoh	Akan cocok	Tidak akan cocok
http://*.example.org	http://www.example.org	https://test.example.org
	http://test.example.org	https://test.example.org:123
		http://test.example.org:123
*.example.org	test.example.org	http://test.example.org:123
	test.test.example.org	https://test.example.org:123
	.example.org	
	http://test.example.org	
example.org	https://test.example.org	
example.org	http://example.org	
	https://example.org	
http://example.org		https://example.org
		http://example.org:123
http://example.org:*	http://example.org:123	
	http://example.org	
http://example.org:1*3	http://example.org:123	

Contoh	Akan cocok	Tidak akan cocok
	<code>http://example.org:1893</code> <code>http://example.org:13</code>	
<code>*.example.org:1*</code>	<code>test.example.org:123</code>	

Access-Control-Expose-Header

Menentukan nama header yang CloudFront menggunakan sebagai nilai untuk `Access-Control-Expose-Headers` header dalam tanggapan terhadap permintaan CORS. Nilai yang valid untuk pengaturan ini termasuk nama header HTTP atau karakter wildcard (*).

Akses-Kontrol-Max-Age

Beberapa detik, yang CloudFront digunakan sebagai nilai untuk `Access-Control-Max-Age` header dalam menanggapi permintaan preflight CORS.

Pengesampingan asal

Setelan Boolean yang menentukan bagaimana CloudFront perilaku ketika respons dari asal berisi salah satu header CORS yang juga ada dalam kebijakan.

- Jika disetel ke `true` dan respons asal berisi header CORS yang juga ada di kebijakan, CloudFront tambahkan header CORS dalam kebijakan ke respons. CloudFront kemudian mengirimkan respons itu ke pemirsa. CloudFront mengabaikan header yang diterimanya dari asal.
- Saat disetel ke `false` dan respons asal berisi header CORS (terlepas dari apakah header CORS ada dalam kebijakan), CloudFront sertakan header CORS yang diterimanya dari asal ke respons. CloudFront tidak menambahkan header CORS apa pun dalam kebijakan ke respons yang dikirim ke pemirsa.

Header keamanan

Anda dapat menggunakan pengaturan header keamanan untuk menambahkan dan mengonfigurasi beberapa header respons HTTP terkait keamanan dalam kebijakan header respons.

Daftar ini menjelaskan cara menentukan setelan dan nilai yang valid dalam kebijakan header respons. Untuk informasi selengkapnya tentang masing-masing header ini dan bagaimana mereka digunakan dalam respons HTTP dunia nyata, lihat tautan ke MDN Web Docs.

Content-Security-Policy

Menentukan arahan kebijakan keamanan konten yang CloudFront digunakan sebagai nilai untuk header Content-Security-Policy respons.

Untuk informasi selengkapnya tentang header ini dan arahan kebijakan yang valid, lihat [Content-Security-Policy](#) di MDN Web Docs.

Note

Nilai Content-Security-Policy header dibatasi hingga 1783 karakter.

Kebijakan Perujuk

Menentukan arahan kebijakan perujuk yang CloudFront menggunakan sebagai nilai untuk header respon. Referrer-Policy Nilai yang valid untuk pengaturan ini adalah `no-referrer`, `no-referrer-when-downgrade`, `origin`, `origin-when-cross-origin`, `same-origin`, `strict-origin`, `strict-origin-when-cross-origin`, dan `unsafe-url`.

Untuk informasi selengkapnya tentang header ini dan arahan ini, lihat [Referrer-Policy](#) di MDN Web Docs.

Strict-Transport-Security

Menentukan arahan dan pengaturan yang CloudFront menggunakan sebagai nilai untuk header Strict-Transport-Security respon. Untuk pengaturan ini, Anda secara terpisah menentukan:

- Sejumlah detik, yang CloudFront digunakan sebagai nilai untuk `max-age` direktif header ini
- Pengaturan Boolean (`true` atau `false`) untuk `preload`, yang menentukan apakah CloudFront menyertakan `preload` direktif dalam nilai header ini
- Pengaturan Boolean (`true` atau `false`) untuk `includeSubDomains`, yang menentukan apakah CloudFront menyertakan `includeSubDomains` direktif dalam nilai header ini

Untuk informasi selengkapnya tentang header ini dan arahan ini, lihat [Strict-Transport-Security](#) di MDN Web Docs.

X-Content-Type-Options

Ini adalah pengaturan Boolean (`true` atau `false`) yang menentukan apakah CloudFront menambahkan `X-Content-Type-Options` header ke respons. Saat pengaturan ini `true`, CloudFront tambahkan `X-Content-Type-Options: nosniff` header ke respons. Jika CloudFront tidak, tidak menambahkan header ini.

Untuk informasi selengkapnya tentang header ini, lihat [X-Content-Type-Options](#) di MDN Web Docs.

X-Frame-Options

Menentukan direktif yang CloudFront menggunakan sebagai nilai untuk header `X-Frame-Options` respon. Nilai yang valid untuk pengaturan ini adalah `DENY` atau `SAMEORIGIN`.

Untuk informasi selengkapnya tentang header ini dan arahan ini, lihat [X-Frame-Options](#) di MDN Web Docs.

Perlindungan-XSS-X

Menentukan arahan dan pengaturan yang CloudFront menggunakan sebagai nilai untuk header `X-XSS-Protection` respon. Untuk pengaturan ini, Anda secara terpisah menentukan:

- `X-XSS-Protection` Pengaturan `0` (menonaktifkan penyaringan XSS) atau `1` (memungkinkan penyaringan XSS)
- Pengaturan Boolean (`true` atau `false`) untuk `block`, yang menentukan apakah CloudFront menyertakan `mode=block` direktif dalam nilai untuk header ini
- URI pelaporan, yang menentukan apakah CloudFront menyertakan `report=reporting URI` direktif dalam nilai untuk header ini

Anda dapat menentukan `true` untuk `block`, atau Anda dapat menentukan URI pelaporan, tetapi Anda tidak dapat menentukan keduanya bersama-sama. Untuk informasi selengkapnya tentang header ini dan arahan ini, lihat [X-XSS-Protection](#) di MDN Web Docs.

Pengesampingan asal

Setiap pengaturan header keamanan ini berisi setelan Boolean (`true` atau `false`) yang menentukan bagaimana CloudFront perilaku ketika respons dari asal berisi header tersebut.

Jika setelan ini disetel ke `true` dan respons asal berisi header, CloudFront tambahkan header dalam kebijakan ke respons yang dikirimkan ke penampil. Ini mengabaikan header yang diterimanya dari asal.

Saat pengaturan ini disetel ke `false` dan respons asal berisi header, CloudFront sertakan header yang diterimanya dari asal dalam respons yang dikirimkan ke penampil.

Jika respons asal tidak berisi header, CloudFront tambahkan header dalam kebijakan ke respons yang dikirimkan ke penampil. CloudFront melakukan ini ketika pengaturan ini diatur ke `true` atau `false`.

Header kustom

Anda dapat menggunakan pengaturan header khusus untuk menambahkan dan mengonfigurasi header HTTP kustom dalam kebijakan header respons. CloudFront menambahkan header ini ke setiap respons yang dikembalikan ke pemirsa. Untuk setiap header kustom, Anda juga menentukan nilai untuk header, meskipun menentukan nilai adalah opsional. Hal ini karena CloudFront dapat menambahkan header respon tanpa nilai.

Setiap header kustom juga memiliki pengaturan penggantian Origin sendiri:

- Jika setelan ini disetel ke `true` dan respons asal berisi header kustom yang ada di kebijakan, CloudFront tambahkan header kustom dalam kebijakan ke respons yang dikirimkan ke penampil. Ini mengabaikan header yang diterimanya dari asal.
- Bila setelan ini `false` dan respons asal berisi header kustom yang ada di kebijakan, CloudFront sertakan header kustom yang diterimanya dari asal dalam respons yang dikirimkan ke penampil.
- Jika respons asal tidak berisi header kustom yang ada di kebijakan, CloudFront tambahkan header kustom dalam kebijakan ke respons yang dikirimkan ke penampil. CloudFront melakukan ini ketika pengaturan ini diatur ke `true` atau `false`.

Hapus header

Anda dapat menentukan header yang CloudFront ingin Anda hapus dari respons yang diterimanya dari asal sehingga header tidak disertakan dalam tanggapan yang CloudFront kirim ke pemirsa. CloudFront menghapus header dari setiap respons yang dikirimkan ke pemirsa, baik objek disajikan dari CloudFront cache atau dari asal. Misalnya, Anda dapat menghapus header yang tidak berguna untuk browser, seperti `X-Powered-By` atau `Vary`, sehingga CloudFront menghapus header ini dari tanggapan yang dikirimkan ke pemirsa.

Saat Anda menentukan header yang akan dihapus menggunakan kebijakan header respons, CloudFront hapus header terlebih dahulu, lalu tambahkan header apa pun yang ditentukan di bagian

lain dari kebijakan header respons (header CORS, header keamanan, header khusus, dll.). Jika Anda menentukan header yang akan dihapus tetapi juga menambahkan header yang sama di bagian lain kebijakan, CloudFront sertakan header dalam respons yang dikirimkan ke pemirsa.

Note

Anda dapat menggunakan kebijakan header respons untuk menghapus `Server` dan `Date` header yang CloudFront diterima dari asal, sehingga header ini (sebagaimana diterima dari asal) tidak disertakan dalam tanggapan yang CloudFront dikirim ke pemirsa. Namun, jika Anda melakukannya, CloudFront tambahkan versinya sendiri dari header ini ke tanggapan yang dikirimkan ke pemirsa. Untuk `Server` header yang CloudFront menambahkan, nilai header adalah `CloudFront`.

Header yang tidak dapat Anda hapus

Anda tidak dapat menghapus header berikut menggunakan kebijakan header respons. Jika Anda menentukan header ini di bagian Hapus header dari kebijakan header respons (`ResponseHeadersPolicyRemoveHeadersConfig` di API), Anda akan menerima kesalahan.

- `Connection`
- `Content-Encoding`
- `Content-Length`
- `Expect`
- `Host`
- `Keep-Alive`
- `Proxy-Authenticate`
- `Proxy-Authorization`
- `Proxy-Connection`
- `Trailer`
- `Transfer-Encoding`
- `Upgrade`
- `Via`
- `Warning`
- `X-Accel-Buffering`

- X-Accel-Charset
- X-Accel-Limit-Rate
- X-Accel-Redirect
- X-Amz-Cf-.*
- X-Amzn-Auth
- X-Amzn-Cf-Billing
- X-Amzn-Cf-Id
- X-Amzn-Cf-Xff
- X-Amzn-ErrorType
- X-Amzn-Fle-Profile
- X-Amzn-Header-Count
- X-Amzn-Header-Order
- X-Amzn-Lambda-Integration-Tag
- X-Amzn-RequestId
- X-Cache
- X-Edge-.*
- X-Forwarded-Proto
- X-Real-IP

Header Pengaturan Waktu Server

Gunakan pengaturan Server-Timing header untuk mengaktifkan Server-Timing header dalam respons HTTP yang dikirim dari CloudFront. Anda dapat menggunakan header ini untuk melihat metrik yang dapat membantu Anda mendapatkan wawasan tentang perilaku dan kinerja CloudFront dan asal Anda. Misalnya, Anda dapat melihat layer cache mana yang menyajikan hit cache. Atau, Anda dapat melihat latensi byte pertama dari asal jika ada cache yang hilang. Metrik di Server-Timing header dapat membantu Anda memecahkan masalah atau menguji efisiensi konfigurasi asal atau asal Anda CloudFront .

Untuk informasi selengkapnya tentang menggunakan Server-Timing header dengan CloudFront, lihat topik berikut.

Untuk mengaktifkan Server-Timing header, [buat \(atau edit\) kebijakan header respons](#).

Topik

- [Tingkat pengambilan sampel dan header permintaan Pragma](#)
- [Header Server-Timing dari asal](#)
- [Metrik header Waktu Server](#)
- [Contoh header Server-Timing](#)

Tingkat pengambilan sampel dan header permintaan Pragma

Saat mengaktifkan `Server-Timing` header dalam kebijakan header respons, Anda juga menentukan laju pengambilan sampel. Sampling rate adalah angka 0-100 (inklusif) yang menentukan persentase respons yang CloudFront ingin Anda tambahkan header. `Server-Timing` Saat Anda menyetel laju pengambilan sampel ke 100, CloudFront tambahkan `Server-Timing` header ke respons HTTP untuk setiap permintaan yang cocok dengan perilaku cache yang dilampirkan oleh kebijakan header respons. Saat Anda menyetelnya ke 50, CloudFront tambahkan header ke 50% respons untuk permintaan yang cocok dengan perilaku cache. Anda dapat mengatur laju pengambilan sampel ke angka 0-100 dengan hingga empat tempat desimal.

Ketika laju pengambilan sampel diatur ke angka yang lebih rendah dari 100, Anda tidak dapat mengontrol respons mana yang CloudFront menambahkan `Server-Timing` header, hanya persentasenya. Namun, Anda dapat menambahkan `Pragma` header dengan nilai yang disetel ke `server-timing` dalam permintaan HTTP untuk menerima `Server-Timing` header dalam respons terhadap permintaan tersebut. Ini berfungsi tidak peduli berapa laju pengambilan sampel diatur. Bahkan ketika laju pengambilan sampel diatur ke nol (0), CloudFront tambahkan `Server-Timing` header ke respons jika permintaan berisi `Pragma: server-timing` header.

Header Server-Timing dari asal

Ketika ada cache yang hilang dan CloudFront meneruskan permintaan ke asal, asal mungkin menyertakan `Server-Timing` header dalam responsnya. CloudFront Dalam hal ini, CloudFront tambahkan [metriknya](#) ke `Server-Timing` header yang diterimanya dari asal. Respons yang CloudFront dikirim ke penampil berisi satu `Server-Timing` header yang mencakup nilai yang berasal dari asal dan metrik yang CloudFront ditambahkan. Nilai header dari asal mungkin berada di akhir, atau di antara dua set metrik yang CloudFront ditambahkan ke header.

Ketika ada hit cache, respons yang CloudFront dikirim ke penampil berisi satu `Server-Timing` header yang hanya menyertakan CloudFront metrik dalam nilai header (nilai dari asal tidak disertakan).

Metrik header Waktu Server

Saat CloudFront menambahkan `Server-Timing` header ke respons HTTP, nilai header berisi satu atau beberapa metrik yang dapat membantu Anda mendapatkan wawasan tentang perilaku dan kinerja CloudFront dan asal Anda. Daftar berikut berisi semua metrik dan nilai potensinya. `Server-TimingHeader` hanya berisi beberapa metrik ini, tergantung pada sifat permintaan dan respons melalui CloudFront.

Beberapa metrik ini disertakan dalam `Server-Timing` header dengan nama saja (tidak ada nilai). Yang lain adalah nama dan nilai. Ketika metrik memiliki nilai, nama dan nilai dipisahkan oleh titik koma (`.`). ; Ketika header berisi lebih dari satu metrik, metrik dipisahkan oleh koma (`,`).

`cdn-cache-hit`

CloudFront memberikan respons dari cache tanpa membuat permintaan ke asal.

`cdn-cache-refresh`

CloudFront memberikan respons dari cache setelah mengirim permintaan ke asal untuk memverifikasi bahwa objek yang di-cache masih valid. Dalam hal ini, CloudFront tidak mengambil objek lengkap dari asal.

`cdn-cache-miss`

CloudFront tidak memberikan respons dari cache. Dalam hal ini, CloudFront meminta objek lengkap dari asal sebelum mengembalikan respon.

`cdn-pop`

Berisi nilai yang menjelaskan CloudFront point of presence (POP) mana yang menangani permintaan.

`cdn-rid`

Berisi nilai dengan pengenal CloudFront unik untuk permintaan tersebut. Anda dapat menggunakan pengenal permintaan (RID) ini saat memecahkan masalah dengan Dukungan

`cdn-hit-layer`

Metrik ini hadir saat CloudFront memberikan respons dari cache tanpa membuat permintaan ke asal. Ini berisi salah satu nilai berikut:

- `EDGE` — CloudFront memberikan respons cache dari lokasi POP.
- `REC` — CloudFront memberikan respons cache dari lokasi [regional edge cache](#) (REC).

- Origin Shield — CloudFront memberikan respons cache dari REC yang bertindak sebagai Origin [Shield](#).

cdn-upstream-layer

Saat CloudFront meminta objek lengkap dari asal, metrik ini hadir dan berisi salah satu nilai berikut:

- EDGE — Lokasi POP mengirim permintaan langsung ke asal.
- REC — Lokasi REC mengirim permintaan langsung ke asal.
- Origin Shield — REC yang bertindak sebagai [Origin Shield](#) mengirim permintaan langsung ke asal.

cdn-upstream-dns

Berisi nilai dengan jumlah milidetik yang dihabiskan untuk mengambil catatan DNS untuk asal. Nilai nol (0) menunjukkan bahwa CloudFront menggunakan hasil DNS cache atau menggunakan kembali koneksi yang ada.

cdn-upstream-connect

Berisi nilai dengan jumlah milidetik antara saat permintaan DNS asal selesai dan koneksi TCP (dan TLS, jika ada) ke asal selesai. Nilai nol (0) menunjukkan bahwa CloudFront menggunakan kembali koneksi yang ada.

cdn-upstream-fbl

Berisi nilai dengan jumlah milidetik antara saat permintaan HTTP asal selesai dan ketika byte pertama diterima dalam respons dari asal (latensi byte pertama).

cdn-downstream-fbl

Berisi nilai dengan jumlah milidetik antara saat lokasi tepi selesai menerima permintaan dan saat mengirim byte pertama respons ke penampil.

Contoh header Server-Timing

Berikut ini adalah contoh Server-Timing header yang mungkin diterima oleh penampil CloudFront saat pengaturan Server-Timing header diaktifkan.

Example— cache terlewatkan

Contoh berikut menunjukkan Server-Timing header yang mungkin diterima oleh penampil saat objek yang diminta tidak ada dalam CloudFront cache.

```
Server-Timing: cdn-upstream-layer;desc="EDGE",cdn-upstream-dns;dur=0,cdn-upstream-
connect;dur=114,cdn-upstream-fbl;dur=177,cdn-cache-miss,cdn-pop;desc="PHX50-C2",cdn-
rid;desc="yNPsyYn7skvTzwWkq3Wcc8Nj_foxUjQe9H1ifslzWhb0w7aLbFvGg==",cdn-downstream-
fbl;dur=436
```

Server-TimingHeader ini menunjukkan hal berikut:

- Permintaan asal dikirim dari lokasi CloudFront point of presence (POP) (cdn-upstream-layer;desc="EDGE").
- CloudFront menggunakan hasil DNS cache untuk asal ()cdn-upstream-dns;dur=0.
- Butuh 114 milidetik CloudFront untuk menyelesaikan koneksi TCP (dan TLS, jika ada) ke origin ().
cdn-upstream-connect;dur=114
- Butuh 177 milidetik CloudFront untuk menerima byte pertama respons dari asal, setelah menyelesaikan permintaan ()cdn-upstream-fbl;dur=177.
- Objek yang diminta tidak CloudFront ada di cache (cdn-cache-miss).
- Permintaan diterima di lokasi tepi yang diidentifikasi oleh kode PHX50-C2 (cdn-pop;desc="PHX50-C2").
- ID CloudFront unik untuk permintaan ini adalah
yNPsyYn7skvTzwWkq3Wcc8Nj_foxUjQe9H1ifslzWhb0w7aLbFvGg== (cdn-rid;desc="yNPsyYn7skvTzwWkq3Wcc8Nj_foxUjQe9H1ifslzWhb0w7aLbFvGg==").
- Butuh 436 milidetik CloudFront untuk mengirim byte pertama respons ke pemirsa, setelah menerima permintaan penampil ()cdn-downstream-fbl;dur=436.

Example— tembolok

Contoh berikut menunjukkan Server-Timing header yang mungkin diterima oleh penampil saat objek yang diminta berada dalam CloudFront cache.

```
Server-Timing: cdn-cache-hit,cdn-pop;desc="SEA19-C1",cdn-
rid;desc="nQBz4aJU2kP9iC3KHEq7vFxfMozu-VYBwGzkW9di0peVc7xsrlKj-g==",cdn-hit-
layer;desc="REC",cdn-downstream-fbl;dur=137
```

Server-TimingHeader ini menunjukkan hal berikut:

- Objek yang diminta ada di cache (cdn-cache-hit).

- Permintaan diterima di lokasi tepi yang diidentifikasi oleh kode SEA19-C1 (`cdn-pop;desc="SEA19-C1"`).
- ID CloudFront unik untuk permintaan ini adalah `nQBz4aJU2kP9iC3KHEq7vFxfMozu-VYBwGzkW9di0peVc7xsrLKj-g==` (`cdn-rid;desc="nQBz4aJU2kP9iC3KHEq7vFxfMozu-VYBwGzkW9di0peVc7xsrLKj-g=="`).
- Objek yang diminta di-cache di lokasi cache tepi regional (REC) (`cdn-hit-layer;desc="REC"`).
- Butuh 137 milidetik CloudFront untuk mengirim byte pertama respons ke pemirsa, setelah menerima permintaan penampil (`cdn-downstream-fbl;dur=137`).

Buat kebijakan header respons

Anda dapat menggunakan kebijakan header respons untuk menentukan header HTTP yang CloudFront ditambahkan atau dihapus Amazon dalam respons HTTP. Untuk informasi selengkapnya tentang kebijakan header respons dan alasan menggunakannya, lihat [Menambah atau menghapus header respons dengan kebijakan](#).

Anda dapat membuat kebijakan header respons di CloudFront konsol. Atau Anda dapat membuatnya dengan menggunakan AWS CloudFormation, AWS Command Line Interface (AWS CLI), atau CloudFront API. Setelah membuat kebijakan header respons, Anda melampirkannya ke satu atau beberapa perilaku cache dalam CloudFront distribusi.

Sebelum membuat kebijakan header respons kustom, periksa apakah salah satu [kebijakan header respons terkelola sesuai dengan kasus](#) penggunaan Anda. Jika ada, Anda dapat melampirkannya ke perilaku cache Anda. Dengan begitu, Anda tidak perlu membuat atau mengelola kebijakan header respons Anda sendiri.

Console

Untuk membuat kebijakan header respons (konsol)

1. Masuk ke Konsol Manajemen AWS, lalu buka tab Header respons pada halaman Kebijakan di CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home#/policies/responseHeaders>.
2. Pilih Buat kebijakan header respons.
3. Dalam formulir kebijakan Buat header respons, lakukan hal berikut:

- a. Di panel Detail, masukkan Nama untuk kebijakan header respons dan (opsional) Deskripsi yang menjelaskan tujuan kebijakan tersebut.
- b. Di panel Cross-origin resource sharing (CORS), pilih toggle Configure CORS dan konfigurasi header CORS apa pun yang ingin Anda tambahkan ke kebijakan. Jika Anda ingin header yang dikonfigurasi mengganti header yang CloudFront menerima dari asal, pilih kotak centang Origin override.

Untuk informasi selengkapnya tentang setelan header CORS, lihat [the section called “Header CORS”](#).

- c. Di panel Security header, pilih toggle dan konfigurasi setiap header keamanan yang ingin ditambahkan ke kebijakan.

Untuk informasi selengkapnya tentang pengaturan header keamanan, lihat [the section called “Header keamanan”](#).

- d. Di panel Custom header, tambahkan header kustom apa pun yang ingin Anda sertakan dalam kebijakan.

Untuk informasi selengkapnya tentang setelan header khusus, lihat [the section called “Header kustom”](#).

- e. Di panel Hapus header, tambahkan nama header apa pun yang CloudFront ingin Anda hapus dari respons asal dan tidak termasuk dalam respons yang CloudFront dikirim ke pemirsa.

Untuk informasi selengkapnya tentang pengaturan hapus header, lihat [the section called “Hapus header”](#).

- f. Di panel header Server-Timing, pilih sakelar Aktifkan dan masukkan laju pengambilan sampel (angka antara 0 dan 100, inklusif).

Untuk informasi selengkapnya tentang Server-Timing header, lihat [the section called “Header Pengaturan Waktu Server”](#).

4. Pilih Buat untuk membuat kebijakan.

Setelah membuat kebijakan header respons, Anda dapat melampirkannya ke perilaku cache dalam CloudFront distribusi.

Untuk melampirkan kebijakan header respons ke distribusi yang ada (konsol)

1. Buka halaman Distribusi di CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home#/distributions>.
2. Pilih distribusi untuk diperbarui, lalu pilih Perilaku tab.
3. Pilih perilaku cache yang akan diperbarui, lalu pilih Edit.

Atau, untuk membuat perilaku cache baru, pilih Buat perilaku.

4. Untuk kebijakan header Response, pilih kebijakan yang akan ditambahkan ke perilaku cache.
5. Pilih Simpan perubahan untuk memperbarui perilaku cache. Jika Anda membuat perilaku cache baru, pilih Buat perilaku.

Untuk melampirkan kebijakan header respons ke distribusi baru (konsol)

1. Buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Pilih Buat Distribusi.
3. Untuk kebijakan header Response, pilih kebijakan yang akan ditambahkan ke perilaku cache.
4. Pilih pengaturan lain untuk distribusi Anda. Untuk informasi selengkapnya, lihat [the section called “Semua pengaturan distribusi”](#).
5. Pilih Buat distribusi untuk membuat distribusi.

CloudFormation

Untuk membuat kebijakan header respons dengan CloudFormation, gunakan jenis `AWS::CloudFront::ResponseHeadersPolicy` sumber daya. Contoh berikut menunjukkan sintaks CloudFormation template, dalam format YAMAL, untuk membuat kebijakan header respons.

```
Type: AWS::CloudFront::ResponseHeadersPolicy
Properties:
  ResponseHeadersPolicyConfig:
    Name: EXAMPLE-Response-Headers-Policy
    Comment: Example response headers policy for the documentation
  CorsConfig:
    AccessControlAllowCredentials: false
    AccessControlAllowHeaders:
      Items:
```

```

    - '*'
  AccessControlAllowMethods:
    Items:
      - GET
      - OPTIONS
  AccessControlAllowOrigins:
    Items:
      - https://example.com
      - https://docs.example.com
  AccessControlExposeHeaders:
    Items:
      - '*'
  AccessControlMaxAgeSec: 600
  OriginOverride: false
  CustomHeadersConfig:
    Items:
      - Header: Example-Custom-Header-1
        Value: value-1
        Override: true
      - Header: Example-Custom-Header-2
        Value: value-2
        Override: true
  SecurityHeadersConfig:
    ContentSecurityPolicy:
      ContentSecurityPolicy: default-src 'none'; img-src 'self'; script-src
'self'; style-src 'self'; object-src 'none'; frame-ancestors 'none'
      Override: false
    ContentTypeOptions: # You don't need to specify a value for 'X-Content-Type-
Options'.
                        # Simply including it in the template sets its value to
'nosniff'.
      Override: false
  FrameOptions:
    FrameOption: DENY
    Override: false
  ReferrerPolicy:
    ReferrerPolicy: same-origin
    Override: false
  StrictTransportSecurity:
    AccessControlMaxAgeSec: 63072000
    IncludeSubdomains: true
    Preload: true
    Override: false
  XSSProtection:

```

```
ModeBlock: true # You can set ModeBlock to 'true' OR set a value for
ReportUri, but not both
Protection: true
Override: false
ServerTimingHeadersConfig:
  Enabled: true
  SamplingRate: 50
RemoveHeadersConfig:
  Items:
    - Header: Vary
    - Header: X-Powered-By
```

Untuk informasi selengkapnya, lihat [AWS::CloudFront::ResponseHeadersKebijakan](#) di Panduan AWS CloudFormation Pengguna.

CLI

Untuk membuat kebijakan header respons dengan AWS Command Line Interface (AWS CLI), gunakan `aws cloudfront create-response-headers-policy` perintah. Anda dapat menggunakan file input untuk memberikan parameter input untuk perintah, daripada menentukan setiap parameter individu sebagai input baris perintah.

Untuk membuat kebijakan header respons (CLI dengan file input)

1. Gunakan perintah berikut untuk membuat file yang diberi nama `response-headers-policy.yaml`. File ini berisi semua parameter input untuk `create-response-headers-policy` perintah.

```
aws cloudfront create-response-headers-policy --generate-cli-skeleton yml-input
> response-headers-policy.yaml
```

2. Buka `response-headers-policy.yaml` file yang baru saja Anda buat. Edit file untuk menentukan nama kebijakan dan konfigurasi kebijakan header respons yang diinginkan, lalu simpan file tersebut.

Untuk informasi selengkapnya tentang setelan kebijakan header respons, lihat [the section called “Memahami kebijakan header respons”](#).

3. Gunakan perintah berikut untuk membuat kebijakan header respons. Kebijakan yang Anda buat menggunakan parameter input dari `response-headers-policy.yaml` file.

```
aws cloudfront create-response-headers-policy --cli-input-yaml file://response-headers-policy.yaml
```

Catat Id nilai dalam output perintah. Ini adalah ID kebijakan header respons. Anda memerlukannya untuk melampirkan kebijakan ke perilaku cache CloudFront distribusi.

Untuk melampirkan kebijakan header respons ke distribusi yang ada (CLI dengan file input)

1. Gunakan perintah berikut untuk menyimpan konfigurasi distribusi untuk CloudFront distribusi yang ingin Anda perbarui. Ganti *distribution_ID* dengan ID distribusi.

```
aws cloudfront get-distribution-config --id distribution_ID --output yaml > dist-config.yaml
```

2. Buka file yang diberi nama `dist-config.yaml` yang baru saja Anda buat. Edit file, buat perubahan berikut pada perilaku cache untuk membuatnya menggunakan kebijakan header respons.
 - Dalam perilaku cache, tambahkan bidang yang diberi nama `ResponseHeadersPolicyId`. Untuk nilai bidang, gunakan ID kebijakan header respons yang Anda catat setelah membuat kebijakan.
 - Ubah nama `ETag` bidang menjadi `IfMatch`, tetapi jangan ubah nilai bidang.

Simpan file setelah selesai.

3. Gunakan perintah berikut untuk memperbarui distribusi agar menggunakan kebijakan header respons. Ganti *distribution_ID* dengan ID distribusi.

```
aws cloudfront update-distribution --id distribution_ID --cli-input-yaml file://dist-config.yaml
```


Untuk melampirkan kebijakan header respons ke distribusi baru (CLI dengan file input)

1. Gunakan perintah berikut untuk membuat file yang diberi nama `distribution.yaml`. File ini berisi semua parameter input untuk `create-distribution` perintah.

```
aws cloudfront create-distribution --generate-cli-skeleton yaml-input >
distribution.yaml
```

2. Buka `distribution.yaml` file yang baru saja Anda buat. Di perilaku cache default, di `ResponseHeadersPolicyId` bidang, masukkan ID kebijakan header respons yang Anda catat setelah membuat kebijakan. Lanjutkan mengedit file untuk menentukan pengaturan distribusi yang Anda inginkan, kemudian simpan file setelah selesai.

Untuk informasi lebih lanjut tentang pengaturan distribusi, lihat [Semua referensi pengaturan distribusi](#).

3. Gunakan perintah berikut untuk membuat distribusi menggunakan parameter input dari `distribution.yaml` file Anda.

```
aws cloudfront create-distribution --cli-input-yaml file://distribution.yaml
```

API

Untuk membuat kebijakan header respons dengan CloudFront API, gunakan [CreateResponseHeadersPolicy](#). Untuk informasi selengkapnya tentang bidang yang Anda tentukan dalam panggilan API ini, lihat [the section called “Memahami kebijakan header respons”](#) dan dokumentasi referensi API untuk AWS SDK atau klien API lainnya.

Setelah membuat kebijakan header respons, Anda dapat melampirkannya ke perilaku cache, menggunakan salah satu panggilan API berikut:

- Untuk memasangnya ke perilaku cache dalam distribusi yang ada, gunakan [UpdateDistribution](#).
- Untuk memasangnya ke perilaku cache dalam distribusi baru, gunakan [CreateDistribution](#).

Untuk kedua panggilan API ini, berikan ID kebijakan header respons di `ResponseHeadersPolicyId` bidang, di dalam perilaku cache. Untuk informasi selengkapnya

tentang bidang lain yang Anda tentukan dalam panggilan API ini, lihat [Semua referensi pengaturan distribusi](#) dan dokumentasi referensi API untuk AWS SDK atau klien API lainnya.

Menggunakan kebijakan header respons terkelola

Dengan kebijakan header CloudFront respons, Anda dapat menentukan header HTTP yang CloudFront dihapus atau ditambahkan Amazon dalam tanggapan yang dikirimkan ke pemirsa. Untuk informasi selengkapnya tentang kebijakan header respons dan alasan menggunakannya, lihat [Menambah atau menghapus header respons dengan kebijakan](#).

CloudFront menyediakan kebijakan header respons terkelola yang dapat Anda lampirkan ke perilaku cache dalam CloudFront distribusi Anda. Dengan kebijakan header respons terkelola, Anda tidak perlu menulis atau mempertahankan kebijakan Anda sendiri. Kebijakan terkelola berisi kumpulan header respons HTTP untuk kasus penggunaan umum.

Untuk menggunakan kebijakan header respons terkelola, Anda melampirkannya ke perilaku cache dalam distribusi Anda. Prosesnya sama seperti saat Anda membuat kebijakan header respons kustom. Namun, alih-alih membuat kebijakan baru, Anda melampirkan salah satu kebijakan terkelola. Anda melampirkan kebijakan baik dengan nama (dengan konsol) atau dengan ID (dengan CloudFormation, the AWS CLI, atau AWS SDKs). Nama-nama dan IDs tercantum di bagian berikut.

Untuk informasi selengkapnya, lihat [the section called “Buat kebijakan header respons”](#).

Topik berikut menjelaskan kebijakan header respons terkelola yang dapat Anda gunakan.

Topik

- [CORS-dan- SecurityHeadersPolicy](#)
- [CORS-dengan-preflight](#)
- [CORS- - with-preflight-and SecurityHeadersPolicy](#)
- [SecurityHeadersPolicy](#)
- [SimpleCORS](#)

CORS-dan- SecurityHeadersPolicy

[Lihat kebijakan ini di CloudFront konsol](#)

Gunakan kebijakan terkelola ini untuk mengizinkan permintaan CORS sederhana dari asal mana pun. Kebijakan ini juga menambahkan satu set header keamanan ke semua tanggapan yang

CloudFront dikirimkan ke pemirsa. Kebijakan ini menggabungkan [the section called “SimpleCORS”](#) dan [the section called “SecurityHeadersPolicy”](#) kebijakan menjadi satu.

Saat menggunakan CloudFormation, the AWS CLI, atau CloudFront API, ID untuk kebijakan ini adalah:

e61eb60c-9c35-4d20-a928-2b84e02af89c

Pengaturan kebijakan

	Nama header	Nilai header	Mengesampingkan asal?
Header CORS:	Access-Control-Allow-Origin	*	Tidak
Header keamanan:	Referrer-Policy	strict-origin-when-cross-origin	Tidak
	Strict-Transport-Security	max-age=31536000	Tidak
	X-Content-Type-Options	nosniff	Ya
	X-Frame-Options	SAMEORIGIN	Tidak
	X-XSS-Protection	1; mode=block	Tidak

CORS-dengan-preflight

[Lihat kebijakan ini di CloudFront konsol](#)

Gunakan kebijakan terkelola ini untuk mengizinkan permintaan CORS dari asal mana pun, termasuk permintaan preflight. Untuk permintaan preflight (menggunakan OPTIONS metode HTTP), CloudFront tambahkan ketiga header berikut ke respons. Untuk permintaan CORS sederhana, CloudFront tambahkan hanya Access-Control-Allow-Origin header.

Jika respons yang CloudFront menerima dari asal menyertakan salah satu header ini, CloudFront gunakan header yang diterima (dan nilainya) dalam responsnya terhadap penampil. CloudFront tidak menggunakan header dalam kebijakan ini.

Saat menggunakan CloudFormation, the AWS CLI, atau CloudFront API, ID untuk kebijakan ini adalah:

5cc3b908-e619-4b99-88e5-2cf7f45965bd

Pengaturan kebijakan

	Nama header	Nilai header	Mengesampingkan asal?
Header CORS:	Access-Control-Allow-Methods	DELETE, GET, HEAD, OPTIONS, PATCH, POST, PUT	Tidak
	Access-Control-Allow-Origin	*	
	Access-Control-Expose-Headers	*	

CORS- - with-preflight-and SecurityHeadersPolicy

[Lihat kebijakan ini di CloudFront konsol](#)

Gunakan kebijakan terkelola ini untuk mengizinkan permintaan CORS dari asal mana pun. Ini termasuk permintaan preflight. Kebijakan ini juga menambahkan satu set header keamanan ke semua tanggapan yang CloudFront dikirimkan ke pemirsa. Kebijakan ini menggabungkan [the section called “CORS-dengan-preflight”](#) dan [the section called “SecurityHeadersPolicy”](#) kebijakan menjadi satu.

Saat menggunakan CloudFormation, the AWS CLI, atau CloudFront API, ID untuk kebijakan ini adalah:

eaab4381-ed33-4a86-88ca-d9558dc6cd63

Pengaturan kebijakan

	Nama header	Nilai header	Mengesampingkan asal?
Header CORS:	Access-Control-Allow-Methods	DELETE, GET, HEAD, OPTIONS, PATCH, POST, PUT	Tidak
	Access-Control-Allow-Origin	*	
	Access-Control-Expose-Headers	*	
Header keamanan:	Referrer-Policy	strict-origin-when-cross-origin	Tidak
	Strict-Transport-Security	max-age=31536000	Tidak
	X-Content-Type-Options	nosniff	Ya
	X-Frame-Options	SAMEORIGIN	Tidak
	X-XSS-Protection	1; mode=block	Tidak

SecurityHeadersPolicy

[Lihat kebijakan ini di CloudFront konsol](#)

Gunakan kebijakan terkelola ini untuk menambahkan satu set header keamanan ke semua respons yang CloudFront dikirim ke pemirsa. Untuk informasi selengkapnya tentang header keamanan ini, lihat [pedoman keamanan web Mozilla](#).

Dengan kebijakan header respons ini, CloudFront tambahkan X-Content-Type-Options: nosniff ke semua tanggapan. Ini adalah kasus ketika respons yang CloudFront diterima dari asal menyertakan header ini dan ketika tidak. Untuk semua header lain dalam kebijakan ini, jika respons yang CloudFront diterima dari asal menyertakan header, CloudFront menggunakan header yang

diterima (dan nilainya) sebagai responsnya terhadap penampil. Itu tidak menggunakan header dalam kebijakan ini.

Saat menggunakan CloudFormation, the AWS CLI, atau CloudFront API, ID untuk kebijakan ini adalah:

67f7725c-6f97-4210-82d7-5512b31e9d03

Pengaturan kebijakan

	Nama header	Nilai header	Mengesampingkan asal?
Header keamanan:	Referrer-Policy	strict-origin-when-cross-origin	Tidak
	Strict-Transport-Security	max-age=31536000	Tidak
	X-Content-Type-Options	nosniff	Ya
	X-Frame-Options	SAMEORIGIN	Tidak
	X-XSS-Protection	1; mode=block	Tidak

SimpleCORS

[Lihat kebijakan ini di CloudFront konsol](#)

Gunakan kebijakan terkelola ini untuk mengizinkan [permintaan CORS sederhana](#) dari asal mana pun. Dengan kebijakan ini, CloudFront tambahkan header `Access-Control-Allow-Origin: *` ke semua respons untuk permintaan CORS sederhana.

Jika respons yang CloudFront diterima dari asal menyertakan `Access-Control-Allow-Origin` header, CloudFront gunakan header itu (dan nilainya) dalam responsnya terhadap penampil. CloudFront tidak menggunakan header dalam kebijakan ini.

Saat menggunakan CloudFormation, the AWS CLI, atau CloudFront API, ID untuk kebijakan ini adalah:

60669652-455b-4ae9-85a4-c4c02393f86c

Pengaturan kebijakan

	Nama header	Nilai header	Mengesamp ingkan asal?
Header CORS:	Access-Control-Allow-Origin	*	Tidak

Perilaku permintaan dan respons

Topik berikut menjelaskan bagaimana CloudFront menangani permintaan dan tanggapan.

Anda dapat mempelajari cara CloudFront berinteraksi dengan Amazon S3 atau custom origin, menangani berbagai metode dan header HTTP, memproses kode status, dan mengelola caching dan respons kesalahan.

Topik

- [Bagaimana CloudFront memproses permintaan HTTP dan HTTPS](#)
- [Perilaku permintaan dan respons untuk asal Amazon S3](#)
- [Perilaku permintaan dan respons untuk asal kustom](#)
- [Perilaku permintaan dan respons untuk grup asal](#)
- [Tambahkan header khusus ke permintaan asal](#)
- [Bagaimana CloudFront memproses permintaan sebagian untuk suatu objek \(rentang GETs\)](#)
- [Bagaimana CloudFront memproses kode status HTTP 3xx dari asal Anda](#)
- [Bagaimana CloudFront memproses kode status HTTP 4xx dan 5xx dari asal Anda](#)
- [Hasilkan respons kesalahan khusus](#)

Bagaimana CloudFront memproses permintaan HTTP dan HTTPS

Untuk asal Amazon S3, CloudFront menerima permintaan dalam protokol HTTP dan HTTPS untuk objek dalam distribusi secara default. CloudFront kemudian meneruskan permintaan ke bucket Amazon S3 Anda menggunakan protokol yang sama di mana permintaan dibuat.

Untuk asal kustom, saat membuat distribusi, Anda dapat menentukan cara CloudFront mengakses origin Anda: hanya HTTP, atau mencocokkan protokol yang digunakan oleh penampil. Untuk informasi selengkapnya tentang cara CloudFront menangani permintaan HTTP dan HTTPS untuk asal kustom, lihat [Protokol](#).

Untuk informasi tentang cara membatasi distribusi Anda sehingga pengguna akhir hanya dapat mengakses objek menggunakan HTTPS, lihat [Gunakan HTTPS dengan CloudFront](#).

 Note

Biaya untuk permintaan HTTPS lebih tinggi daripada biaya untuk permintaan HTTP. Untuk informasi selengkapnya tentang tarif penagihan, lihat [CloudFront harga](#).

Perilaku permintaan dan respons untuk asal Amazon S3

Untuk memahami cara CloudFront memproses permintaan dan tanggapan saat Anda menggunakan Amazon S3 sebagai asal, lihat bagian berikut:

Topik

- [Cara CloudFront memproses dan meneruskan permintaan ke asal Amazon S3 Anda](#)
- [Bagaimana CloudFront memproses tanggapan dari asal Amazon S3 Anda](#)

Cara CloudFront memproses dan meneruskan permintaan ke asal Amazon S3 Anda

Pelajari cara CloudFront memproses permintaan penampil dan meneruskan permintaan ke asal Amazon S3 Anda.

Daftar Isi

- [Durasi caching dan TTL minimum](#)
- [Alamat IP Klien](#)
- [Permintaan GET bersyarat](#)
- [Cookie](#)
- [Berbagi sumber daya lintas asal \(CORS\)](#)
- [Permintaan GET yang menyertakan tubuh](#)
- [Metode HTTP](#)
- [Header permintaan HTTP yang CloudFront menghapus atau memperbarui](#)
- [Lama maksimum panjang permintaan dan lama maksimum URL](#)
- [Pemasangan OCSP](#)
- [Protokol](#)
- [String pertanyaan](#)

- [Waktu habis dan upaya koneksi tempat asal](#)
- [Waktu habis untuk respons asal](#)
- [Permintaan simultan untuk objek yang sama \(permintaan runtuh\)](#)

Durasi caching dan TTL minimum

Untuk mengontrol berapa lama objek Anda berada dalam CloudFront cache sebelum CloudFront meneruskan permintaan lain ke asal Anda, Anda dapat:

- Konfigurasi asal Anda untuk menambahkan `Cache-Control` atau `Expires` pada setiap objek.
- Tentukan nilai untuk TTL Minimum dalam perilaku CloudFront cache.
- Gunakan nilai default selama 24 jam.

Untuk informasi selengkapnya, lihat [Mengelola berapa lama konten tetap dalam cache \(kedaluwarsa\)](#).

Alamat IP Klien

Jika penampil mengirim permintaan ke CloudFront dan tidak menyertakan header `X-Forwarded-For` permintaan, CloudFront mendapatkan alamat IP penampil dari koneksi TCP, menambahkan `X-Forwarded-For` header yang menyertakan alamat IP, dan meneruskan permintaan ke asal. Misalnya, jika CloudFront mendapatkan alamat IP `192.0.2.2` dari koneksi TCP, akan meneruskan header berikut ke asalnya:

`X-Forwarded-For: 192.0.2.2`

Jika penampil mengirim permintaan ke CloudFront dan menyertakan header `X-Forwarded-For` permintaan, CloudFront mendapatkan alamat IP penampil dari koneksi TCP, menambahkannya ke akhir `X-Forwarded-For` header, dan meneruskan permintaan ke asal. Misalnya, jika permintaan penampil menyertakan `X-Forwarded-For: 192.0.2.4,192.0.2.3` dan CloudFront mendapatkan alamat IP `192.0.2.2` dari koneksi TCP, itu meneruskan header berikut ke asal:

`X-Forwarded-For: 192.0.2.4,192.0.2.3,192.0.2.2`

Note

`X-Forwarded-For` header berisi IPv4 alamat (seperti `192.0.2.44`) dan IPv6 alamat (seperti `2001:0 db 8:85 a3: :8a2e: 0370:7334`).

Permintaan GET bersyarat

Ketika CloudFront menerima permintaan untuk objek yang telah kedaluwarsa dari cache tepi, ia meneruskan permintaan ke asal Amazon S3 untuk mendapatkan versi terbaru dari objek atau untuk mendapatkan konfirmasi dari Amazon S3 bahwa CloudFront cache edge sudah memiliki versi terbaru. Ketika Amazon S3 awalnya mengirim objek ke CloudFront, itu termasuk ETag nilai dan LastModified nilai dalam respons. Dalam permintaan baru yang CloudFront diteruskan ke Amazon S3 CloudFront, tambahkan satu atau kedua header berikut:

- Header If-Match atau If-None-Match yang memuat ETag untuk versi objek yang kedaluwarsa.
- Header If-Modified-Since yang memuat LastModified untuk versi objek yang kedaluwarsa.

Amazon S3 menggunakan informasi ini untuk menentukan apakah objek telah diperbarui dan, oleh karena itu, apakah akan mengembalikan seluruh objek ke CloudFront atau hanya mengembalikan kode status HTTP 304 (tidak dimodifikasi).

Cookie

Amazon S3 tidak memproses cookie. Jika Anda mengonfigurasi perilaku cache untuk meneruskan cookie ke asal Amazon S3, CloudFront teruskan cookie, tetapi Amazon S3 mengabaikannya. Semua permintaan di masa mendatang untuk objek yang sama, terlepas jika Anda mengubah cookie, dilayani dari objek yang ada di dalam cache.

Berbagi sumber daya lintas asal (CORS)

Jika Anda CloudFront ingin menghormati setelan berbagi sumber daya lintas asal Amazon S3, konfigurasi CloudFront untuk meneruskan header yang dipilih ke Amazon S3. Untuk informasi selengkapnya, lihat [Konten cache berdasarkan header permintaan](#).

Permintaan GET yang menyertakan tubuh

Jika GET permintaan penampil menyertakan isi, CloudFront mengembalikan kode status HTTP 403 (Terlarang) ke penampil.

Metode HTTP

Jika Anda mengonfigurasi CloudFront untuk memproses semua metode HTTP yang didukungnya, CloudFront terima permintaan berikut dari pemirsa dan teruskan ke asal Amazon S3 Anda:

- DELETE
- GET
- HEAD
- OPTIONS
- PATCH
- POST
- PUT

CloudFront selalu menyimpan respons GET dan HEAD permintaan. Anda juga dapat CloudFront mengonfigurasi respons cache terhadap OPTIONS permintaan. CloudFront tidak menyimpan respons terhadap permintaan yang menggunakan metode lain.

Jika Anda ingin menggunakan unggahan multi-bagian untuk menambahkan objek ke bucket Amazon S3, Anda harus menambahkan CloudFront kontrol akses asal (OAC) ke distribusi Anda dan memberikan OAC izin yang diperlukan. Untuk informasi selengkapnya, lihat [the section called “Batasi akses ke asal Amazon S3”](#).

Important

Jika Anda mengonfigurasi CloudFront untuk menerima dan meneruskan ke Amazon S3 semua metode HTTP yang CloudFront mendukung, Anda harus membuat CloudFront OAC untuk membatasi akses ke konten Amazon S3 Anda dan memberikan OAC izin yang diperlukan. Misalnya, jika Anda mengonfigurasi CloudFront untuk menerima dan meneruskan metode ini karena Anda ingin menggunakan PUT metode ini, Anda harus mengonfigurasi kebijakan bucket Amazon S3 untuk menangani DELETE permintaan dengan tepat sehingga pemirsa tidak dapat menghapus sumber daya yang tidak Anda inginkan. Untuk informasi selengkapnya, lihat [the section called “Batasi akses ke asal Amazon S3”](#).

Untuk informasi tentang operasi yang didukung oleh Amazon S3, lihat [Dokumentasi Amazon S3](#).

Header permintaan HTTP yang CloudFront menghapus atau memperbarui

CloudFront menghapus atau memperbarui beberapa header sebelum meneruskan permintaan ke asal Amazon S3 Anda. Untuk sebagian besar header, perilaku ini sama dengan untuk asal kustom. Untuk daftar lengkap header permintaan HTTP dan cara CloudFront memprosesnya, lihat [Header dan CloudFront perilaku permintaan HTTP \(asal kustom dan Amazon S3\)](#).

Lama maksimum panjang permintaan dan lama maksimum URL

Panjang maksimum permintaan, termasuk jalur, string kueri (jika ada), dan header, adalah 32.768 byte.

CloudFront membangun URL dari permintaan. Panjang maksimal URL ini adalah 8192 byte.

Jika URL melebihi panjang maksimum, CloudFront mengembalikan kode status HTTP 414 (URI Terlalu Panjang) ke penampil. Jika permintaan melebihi panjang maksimum karena ukuran header terlampaui, CloudFront mengembalikan kode status HTTP 494 ke penampil. Dalam kedua kasus, CloudFront kemudian mengakhiri koneksi TCP ke penampil.

Pemasangan OCSP

Saat penampil mengirimkan permintaan HTTPS untuk suatu objek, CloudFront atau penampil harus mengonfirmasi dengan otoritas sertifikat (CA) bahwa sertifikat SSL untuk domain tersebut belum dicabut. Stapling OCSP mempercepat validasi sertifikat dengan memungkinkan CloudFront untuk memvalidasi sertifikat dan menyimpan respons dari CA, sehingga klien tidak perlu memvalidasi sertifikat secara langsung dengan CA.

Peningkatan kinerja stapling OCSP lebih terasa ketika CloudFront menerima banyak permintaan HTTPS untuk objek dalam domain yang sama. Setiap server di CloudFront lokasi edge harus menyerahkan permintaan validasi terpisah. Ketika CloudFront menerima banyak permintaan HTTPS untuk domain yang sama, setiap server di lokasi edge segera memiliki respons dari CA yang dapat dijadikan staple ke paket dalam jabat tangan SSL. Ketika pemirsa puas bahwa sertifikat valid, CloudFront dapat melayani objek yang diminta. Jika distribusi Anda tidak terlalu banyak CloudFront lokasi tepian, permintaan baru lebih mungkin diarahkan ke server yang belum memvalidasi sertifikat dengan CA. Dalam hal ini, penampil secara terpisah melakukan langkah validasi dan CloudFront server melayani objek. CloudFront Server itu juga mengirimkan permintaan validasi ke CA, jadi lain kali menerima permintaan yang menyertakan nama domain yang sama, ia memiliki respons validasi dari CA.

Protokol

CloudFront meneruskan permintaan HTTP atau HTTPS ke server asal berdasarkan protokol permintaan penampil, baik HTTP atau HTTPS.

Important

Jika bucket Amazon S3 Anda dikonfigurasi sebagai titik akhir situs web, Anda tidak dapat mengonfigurasi CloudFront untuk menggunakan HTTPS untuk berkomunikasi dengan asal Anda karena Amazon S3 tidak mendukung koneksi HTTPS dalam konfigurasi tersebut.

String pertanyaan

Anda dapat mengonfigurasi apakah CloudFront meneruskan parameter string kueri ke asal Amazon S3 Anda. Untuk informasi selengkapnya, lihat [Konten cache berdasarkan parameter string kueri](#).

Waktu habis dan upaya koneksi tempat asal

Batas waktu koneksi asal adalah jumlah detik yang CloudFront menunggu ketika mencoba membuat koneksi ke asal.

Upaya koneksi asal adalah berapa kali CloudFront upaya untuk terhubung ke asal.

Bersama-sama, pengaturan ini menentukan berapa lama CloudFront mencoba untuk terhubung ke asal sebelum gagal ke asal sekunder (dalam kasus grup asal) atau mengembalikan respons kesalahan ke penampil. Secara default, CloudFront tunggu selama 30 detik (3 upaya masing-masing 10 detik) sebelum mencoba terhubung ke asal sekunder atau mengembalikan respons kesalahan. Anda dapat mengurangi waktu ini dengan menentukan waktu koneksi yang lebih singkat, lebih sedikit percobaan, atau keduanya.

Untuk informasi selengkapnya, lihat [Kontrol batas waktu dan upaya asal](#).

Waktu habis untuk respons asal

waktu habis respons asal, juga dikenal sebagai waktu habis baca asal atau waktu habis permintaan asal, berlaku untuk kedua hal berikut:

- Jumlah waktu, dalam hitungan detik, yang CloudFront menunggu respons setelah meneruskan permintaan ke asal.
- Jumlah waktu, dalam hitungan detik, yang CloudFront menunggu setelah menerima paket respons dari asal dan sebelum menerima paket berikutnya.

CloudFront perilaku tergantung pada metode HTTP dari permintaan penampil:

- GET dan HEAD permintaan – Jika asal tidak merespons dalam waktu 30 detik atau berhenti merespons selama 30 detik, CloudFront akan membuat koneksi terputus. Jika jumlah [upaya koneksi asal](#) yang ditentukan lebih dari 1, CloudFront coba lagi untuk mendapatkan respons lengkap. CloudFront mencoba hingga 3 kali, sebagaimana ditentukan oleh nilai pengaturan upaya koneksi asal. Jika asal tidak merespons selama percobaan terakhir, CloudFront tidak mencoba lagi hingga menerima permintaan konten lain di tempat yang sama.
- DELETE, OPTIONS, PATCHPUT,, dan POST permintaan — Jika asal tidak merespons dalam 30 detik CloudFront, lepaskan koneksi dan tidak mencoba lagi untuk menghubungi asal. Klien dapat mengirim ulang permintaan jika perlu.

Anda tidak dapat mengubah waktu respons untuk asal Amazon S3 (wadah S3 yang tidak dikonfigurasi dengan hosting situs web statis).

Permintaan simultan untuk objek yang sama (permintaan runtuh)

Ketika lokasi CloudFront tepi menerima permintaan untuk objek dan objek tidak dalam cache atau objek yang di-cache kedaluwarsa, CloudFront segera kirim permintaan ke asal. Namun, jika ada permintaan simultan untuk objek yang sama—yaitu, jika permintaan tambahan untuk objek yang sama (dengan kunci cache yang sama) tiba di lokasi tepi sebelum CloudFront menerima respons terhadap permintaan pertama— CloudFront berhenti sebelum meneruskan permintaan tambahan ke asal. Jeda singkat ini membantu mengurangi beban pada titik asal. CloudFront mengirimkan respons dari permintaan asli ke semua permintaan yang diterimanya saat dijeda. Ini disebut request collapsing. Dalam CloudFront log, permintaan pertama diidentifikasi sebagai a Miss di x-edge-result-type bidang, dan permintaan yang diciutkan diidentifikasi sebagai a Hit. Untuk informasi selengkapnya tentang CloudFront log, lihat [the section called “CloudFront dan logging fungsi tepi”](#).

CloudFront hanya menciutkan permintaan yang berbagi [kunci cache](#). Jika permintaan tambahan tidak berbagi kunci cache yang sama karena, misalnya, Anda CloudFront mengonfigurasi cache berdasarkan header permintaan atau cookie atau string kueri, CloudFront teruskan semua permintaan dengan kunci cache unik ke asal Anda.

Jika Anda ingin mencegah semua permintaan runtuh, Anda dapat menggunakan kebijakan cache `CachingDisabled`, yang juga mencegah caching. Untuk informasi selengkapnya, lihat [Gunakan kebijakan cache terkelola](#).

Jika Anda ingin mencegah keruntuhan permintaan untuk objek tertentu, Anda dapat mengatur TTL minimum untuk perilaku cache ke 0 dan mengonfigurasi asal untuk mengirim `Cache-Control: private,, Cache-Control: no-store Cache-Control: no-cache Cache-Control: max-`

age=0, atau. Cache-Control: s-maxage=0 Konfigurasi ini akan meningkatkan beban pada asal Anda dan memperkenalkan latensi tambahan untuk permintaan simultan yang dijeda sementara CloudFront menunggu respons terhadap permintaan pertama.

Bagaimana CloudFront memproses tanggapan dari asal Amazon S3 Anda

Pelajari cara CloudFront memproses respons dari asal Amazon S3 Anda.

Daftar Isi

- [Permintaan dibatalkan](#)
- [Header respons HTTP yang CloudFront menghapus atau memperbarui](#)
- [Ukuran file cache maksimum](#)
- [Mengalihkan](#)

Permintaan dibatalkan

Jika suatu objek tidak berada di cache tepi, dan jika penampil mengakhiri sesi (misalnya, menutup browser) setelah CloudFront mendapatkan objek dari asal Anda tetapi sebelum dapat mengirimkan objek yang diminta, CloudFront tidak men-cache objek di lokasi tepi.

Header respons HTTP yang CloudFront menghapus atau memperbarui

CloudFront menghapus atau memperbarui bidang header berikut sebelum meneruskan respons dari asal Amazon S3 Anda ke penampil:

- X-Amz-Id-2
- X-Amz-Request-Id
- Set-Cookie— Jika Anda mengonfigurasi CloudFront untuk meneruskan cookie, itu akan meneruskan bidang Set-Cookie header ke klien. Untuk informasi selengkapnya, lihat [Konten cache berdasarkan cookie](#).
- Trailer
- Transfer-Encoding— Jika asal Amazon S3 Anda mengembalikan bidang header ini, CloudFront tetapkan nilainya chunked sebelum mengembalikan respons ke penampil.
- Upgrade
- Via— CloudFront menetapkan nilai sebagai berikut dalam respons terhadap penampil:

Via: *http-version alphanumeric-string*.cloudfront.net (CloudFront)

Misalnya, nilainya adalah seperti berikut:

Via: 1.1 1026589cc7887e7a0dc7827b4example.cloudfront.net (CloudFront)

Ukuran file cache maksimum

Ukuran maksimum badan respons yang CloudFront menyimpan dalam cache-nya adalah 50 GB. Ini termasuk respons transfer yang dipotong yang tidak menyebutkan nilai header Content-Length.

Anda dapat CloudFront menggunakan cache objek yang lebih besar dari ukuran ini dengan menggunakan permintaan rentang untuk meminta objek di bagian yang masing-masing 50 GB atau lebih kecil. CloudFront cache bagian-bagian ini karena masing-masing dari mereka adalah 50 GB atau lebih kecil. Setelah penampil mengambil semua bagian objek, ia dapat merekonstruksi objek asli yang lebih besar. Untuk informasi selengkapnya, lihat [Gunakan permintaan rentang untuk menyimpan objek besar](#).

Mengalihkan

Anda dapat mengonfigurasi bucket Amazon S3 untuk mengalihkan semua permintaan ke nama host lain; ini dapat berupa bucket Amazon S3 lain atau server HTTP. Jika Anda mengonfigurasi bucket untuk mengalihkan semua permintaan dan jika bucket adalah asal untuk CloudFront distribusi, sebaiknya Anda mengonfigurasi bucket untuk mengalihkan semua permintaan ke distribusi menggunakan nama domain untuk CloudFront distribusi (misalnya, d111111abcdef8.cloudfront.net) atau nama domain alternatif (CNAME) yang dikaitkan dengan distribusi (misalnya, example.com). Jika tidak, pemirsa meminta bypass CloudFront, dan objek disajikan langsung dari asal baru.

Note

Jika Anda mengarahkan permintaan ke nama domain alternatif, Anda juga harus memperbarui layanan DNS untuk domain Anda dengan menambahkan catatan CNAME. Untuk informasi selengkapnya, lihat [Gunakan URL kustom dengan menambahkan nama domain alternatif \(CNames\)](#).

Inilah yang terjadi saat Anda mengonfigurasi keranjang untuk mengalihkan semua permintaan:

1. Penampil (misalnya, browser) meminta objek dari CloudFront.
2. CloudFront meneruskan permintaan ke bucket Amazon S3 yang merupakan asal distribusi Anda.

3. Amazon S3 mengembalikan kode status HTTP 301 (Moved Permanently) serta lokasi baru.
4. CloudFront cache kode status pengalihan dan lokasi baru, dan mengembalikan nilai ke penampil. CloudFront tidak mengikuti pengalihan untuk mendapatkan objek dari lokasi baru.
5. Penampil mengirimkan permintaan lain untuk objek tersebut, tetapi kali ini penampil menentukan lokasi baru yang CloudFront didapatnya:
 - Jika bucket Amazon S3 mengalihkan semua permintaan ke CloudFront distribusi, menggunakan nama domain untuk distribusi atau nama domain alternatif, CloudFront meminta objek dari bucket Amazon S3 atau server HTTP di lokasi baru. Saat lokasi baru mengembalikan objek, CloudFront mengembalikannya ke penampil dan menyimpannya di lokasi tepi.
 - Jika bucket Amazon S3 mengalihkan permintaan ke lokasi lain, permintaan kedua akan melewati CloudFront Bucket Amazon S3 atau server HTTP di lokasi baru mengembalikan objek langsung ke penampil, sehingga objek tidak pernah di-cache di cache tepi. CloudFront

Perilaku permintaan dan respons untuk asal kustom

Untuk memahami cara CloudFront memproses permintaan dan tanggapan saat Anda menggunakan asal kustom, lihat bagian berikut:

Topik

- [Cara CloudFront memproses dan meneruskan permintaan ke asal kustom Anda](#)
- [Bagaimana CloudFront memproses tanggapan dari asal kustom Anda](#)

Cara CloudFront memproses dan meneruskan permintaan ke asal kustom Anda

Pelajari cara CloudFront memproses permintaan penampil dan meneruskan permintaan ke asal kustom Anda.

Daftar Isi

- [Autentikasi](#)
- [Durasi caching dan TTL minimum](#)
- [Alamat IP Klien](#)
- [Autentikasi SSL sisi-klien](#)

- [Kompresi](#)
- [Permintaan bersyarat](#)
- [Cookie](#)
- [Berbagi sumber daya lintas asal \(CORS\)](#)
- [Enkripsi](#)
- [Permintaan GET yang menyertakan tubuh](#)
- [Metode HTTP](#)
- [Header dan CloudFront perilaku permintaan HTTP \(asal kustom dan Amazon S3\)](#)
- [Versi HTTP](#)
- [Lama maksimum panjang permintaan dan lama maksimum URL](#)
- [Pemasangan OCSP](#)
- [Koneksi persisten](#)
- [Protokol](#)
- [String pertanyaan](#)
- [Waktu habis dan upaya koneksi tempat asal](#)
- [Waktu habis untuk respons asal](#)
- [Permintaan simultan untuk objek yang sama \(permintaan runtuh\)](#)
- [Header User-Agent](#)

Autentikasi

Jika Anda meneruskan `Authorization` header ke asal Anda, Anda kemudian dapat mengonfigurasi server asal Anda untuk meminta otentikasi klien untuk jenis permintaan berikut:

- DELETE
- GET
- HEAD
- PATCH
- PUT
- POST

Untuk OPTIONS permintaan, otentikasi klien hanya dapat dikonfigurasi jika Anda menggunakan CloudFront pengaturan berikut:

- CloudFront dikonfigurasi untuk meneruskan Authorization header ke asal Anda
- CloudFront dikonfigurasi untuk tidak menyimpan respons terhadap OPTIONS permintaan

Untuk informasi selengkapnya, lihat [CloudFront Konfigurasi untuk meneruskan Authorization header](#).

Anda dapat menggunakan HTTP atau HTTPS untuk meneruskan permintaan ke server asal Anda. Untuk informasi selengkapnya, lihat [Gunakan HTTPS dengan CloudFront](#).

Durasi caching dan TTL minimum

Untuk mengontrol berapa lama objek Anda berada dalam CloudFront cache sebelum CloudFront meneruskan permintaan lain ke asal Anda, Anda dapat:

- Konfigurasi asal Anda untuk menambahkan Cache-Control atau Expires pada setiap objek.
- Tentukan nilai untuk TTL Minimum dalam perilaku CloudFront cache.
- Gunakan nilai default selama 24 jam.

Untuk informasi selengkapnya, lihat [Mengelola berapa lama konten tetap dalam cache \(kedaluwarsa\)](#).

Alamat IP Klien

Jika penampil mengirim permintaan ke CloudFront dan tidak menyertakan header X-Forwarded-For permintaan, CloudFront mendapatkan alamat IP penampil dari koneksi TCP, menambahkan X-Forwarded-For header yang menyertakan alamat IP, dan meneruskan permintaan ke asal. Misalnya, jika CloudFront mendapatkan alamat IP 192.0.2.2 dari koneksi TCP, akan meneruskan header berikut ke asalnya:

X-Forwarded-For: 192.0.2.2

Jika penampil mengirim permintaan ke CloudFront dan menyertakan header X-Forwarded-For permintaan, CloudFront mendapatkan alamat IP penampil dari koneksi TCP, menambahkannya ke akhir X-Forwarded-For header, dan meneruskan permintaan ke asal. Misalnya, jika permintaan penampil menyertakan X-Forwarded-For: 192.0.2.4, 192.0.2.3 dan CloudFront mendapatkan alamat IP 192.0.2.2 dari koneksi TCP, itu meneruskan header berikut ke asal:

X-Forwarded-For: 192.0.2.4,192.0.2.3,192.0.2.2

Beberapa aplikasi, seperti load balancer (termasuk Elastic Load Balancing), firewall aplikasi web, reverse proxy, sistem pencegahan intrusi, dan API Gateway, menambahkan CloudFront alamat IP server edge yang meneruskan permintaan ke ujung header. X-Forwarded-For Misalnya, jika CloudFront termasuk X-Forwarded-For: 192.0.2.2 dalam permintaan yang diteruskan ke ELB dan jika alamat IP server CloudFront edge adalah 192.0.2.199, permintaan yang diterima instans EC2 Anda berisi header berikut:

X-Forwarded-For: 192.0.2.2,192.0.2.199

Note

X-Forwarded-ForHeader berisi IPv4 alamat (seperti 192.0.2.44) dan IPv6 alamat (seperti 2001:0 db 8:85 a3: :8a2e: 0370:7334).

Perhatikan juga bahwa X-Forwarded-For header dapat dimodifikasi oleh setiap node di jalur ke server saat ini (CloudFront). Untuk informasi lebih lanjut, lihat bagian 8.1 di [RFC 7239](#). Anda juga dapat memodifikasi header menggunakan fungsi komputasi CloudFront tepi.

Aotentikasi SSL sisi-klien

CloudFront mendukung otentikasi TLS (mTLS) timbal balik di mana klien dan server saling mengotentikasi menggunakan sertifikat. Dengan mTL yang dikonfigurasi, CloudFront dapat memvalidasi sertifikat klien selama jabat tangan TLS dan secara opsional menjalankan CloudFront Fungsi untuk mengimplementasikan logika validasi kustom.

Untuk asal yang meminta sertifikat sisi klien saat mTL tidak dikonfigurasi, hapus permintaanCloudFront .

Untuk informasi selengkapnya tentang mengonfigurasi mTL, lihat. [???](#)

CloudFront tidak mendukung otentikasi klien dengan sertifikat SSL sisi klien. Jika asal meminta sertifikat sisi klien, hapus CloudFront permintaan.

Kompresi

Untuk informasi selengkapnya, lihat [Sajikan file terkompresi](#).

Permintaan bersyarat

Ketika CloudFront menerima permintaan untuk objek yang telah kedaluwarsa dari cache tepi, itu meneruskan permintaan ke asal baik untuk mendapatkan versi terbaru dari objek atau untuk mendapatkan konfirmasi dari asal bahwa cache CloudFront tepi sudah memiliki versi terbaru. Biasanya, ketika asal terakhir mengirim objek ke CloudFront, itu termasuk ETag nilai, LastModified nilai, atau kedua nilai dalam respons. Dalam permintaan baru yang CloudFront diteruskan ke asal, CloudFront tambahkan satu atau kedua hal berikut:

- Header If-Match atau If-None-Match yang memuat ETag untuk versi objek yang kedaluwarsa.
- Header If-Modified-Since yang memuat LastModified untuk versi objek yang kedaluwarsa.

Asal menggunakan informasi ini untuk menentukan apakah objek telah diperbarui dan, oleh karena itu, apakah akan mengembalikan seluruh objek ke CloudFront atau hanya mengembalikan kode status HTTP 304 (tidak dimodifikasi).

Note

If-Modified-Since dan permintaan If-None-Match bersyarat tidak didukung ketika CloudFront dikonfigurasi untuk meneruskan cookie (semua atau subset). Untuk informasi selengkapnya, lihat [Konten cache berdasarkan cookie](#).

Cookie

Anda dapat mengonfigurasi CloudFront untuk meneruskan cookie ke asal Anda. Untuk informasi selengkapnya, lihat [Konten cache berdasarkan cookie](#).

Berbagi sumber daya lintas asal (CORS)

Jika Anda CloudFront ingin menghormati setelan berbagi sumber daya lintas asal, konfigurasi CloudFront untuk meneruskan Origin header ke asal Anda. Untuk informasi selengkapnya, lihat [Konten cache berdasarkan header permintaan](#).

Enkripsi

Anda dapat meminta pemirsa untuk menggunakan HTTPS untuk mengirim permintaan CloudFront dan CloudFront harus meneruskan permintaan ke asal kustom Anda dengan menggunakan protokol yang digunakan oleh penampil. Untuk informasi lebih lanjut, lihat pengaturan distribusi berikut:

- [Kebijakan protokol penampil](#)
- [Protokol \(hanya asal kustom\)](#)

CloudFront meneruskan permintaan HTTPS ke server asal menggunakan protokol SSLv3, TLSv1 .0, TLSv1 .1, TLSv1 .2, dan .3. TLSv1 Untuk asal kustom, Anda dapat memilih protokol SSL yang ingin Anda gunakan saat berkomunikasi CloudFront dengan asal Anda:

- Jika Anda menggunakan CloudFront konsol, pilih protokol menggunakan kotak centang Origin SSL Protocols. Untuk informasi selengkapnya, lihat [Buat distribusi](#).
- Jika Anda menggunakan CloudFront API, tentukan protokol dengan menggunakan elemen. `OriginSslProtocols` Untuk informasi selengkapnya, lihat [OriginSslProtocols](#) dan [DistributionConfig](#) di Referensi Amazon CloudFront API.

Jika asalnya adalah bucket Amazon S3, CloudFront akan default ke TLSv1 .3.

Important

Versi lain dari SSL dan TLS tidak didukung.

Untuk informasi selengkapnya tentang menggunakan HTTPS dengan CloudFront, lihat [Gunakan HTTPS dengan CloudFront](#). Untuk daftar cipher yang CloudFront mendukung komunikasi HTTPS antara pemirsa dan CloudFront, dan antara CloudFront dan asal Anda, lihat. [Protokol dan sandi yang didukung antara pemirsa dan CloudFront](#)

Permintaan GET yang menyertakan tubuh

Jika GET permintaan penampil menyertakan isi, CloudFront mengembalikan kode status HTTP 403 (Terlarang) ke penampil.

Metode HTTP

Jika Anda mengonfigurasi CloudFront untuk memproses semua metode HTTP yang didukungnya, CloudFront terima permintaan berikut dari pemirsa dan teruskan ke asal kustom Anda:

- DELETE
- GET
- HEAD
- OPTIONS
- PATCH
- POST
- PUT

CloudFront selalu menyimpan respons GET dan HEAD permintaan. Anda juga dapat CloudFront mengonfigurasi respons cache terhadap OPTIONS permintaan. CloudFront tidak menyimpan respons terhadap permintaan yang menggunakan metode lain.

Untuk informasi tentang konfigurasi apakah asal kustom Anda memproses metode ini, lihat dokumentasi untuk asal Anda.

Important

Jika Anda mengonfigurasi CloudFront untuk menerima dan meneruskan ke asal Anda semua metode HTTP yang CloudFront mendukung, konfigurasi server asal Anda untuk menangani semua metode. Misalnya, jika Anda mengonfigurasi CloudFront untuk menerima dan meneruskan metode ini karena ingin digunakan POST, Anda harus mengonfigurasi server asal Anda untuk menangani DELETE permintaan dengan tepat sehingga pemirsa tidak dapat menghapus sumber daya yang tidak Anda inginkan. Untuk informasi lebih lanjut, lihat dokumentasi untuk server HTTP Anda.

Header dan CloudFront perilaku permintaan HTTP (asal kustom dan Amazon S3)

Tabel berikut mencantumkan header permintaan HTTP yang dapat Anda teruskan ke asal kustom dan juga Amazon S3 (dengan pengecualian yang dicatat). Untuk setiap header, tabel mencakup informasi tentang hal berikut:

- CloudFront perilaku jika Anda tidak mengonfigurasi CloudFront untuk meneruskan header ke asal Anda, yang CloudFront menyebabkan cache objek Anda berdasarkan nilai header.
- Apakah Anda dapat CloudFront mengonfigurasi objek cache berdasarkan nilai header untuk header itu.

Anda dapat CloudFront mengonfigurasi objek cache berdasarkan nilai di User-Agent header Date dan, tetapi kami tidak merekomendasikannya. Header ini memiliki banyak nilai yang mungkin, dan caching berdasarkan nilainya akan menyebabkan CloudFront untuk meneruskan lebih banyak permintaan secara signifikan ke asal Anda.

Untuk informasi lebih lanjut tentang caching berdasarkan nilai header, lihat [Konten cache berdasarkan header permintaan](#).

Header	Perilaku jika Anda tidak mengonfigurasi CloudFront ke cache berdasarkan nilai header	Caching berdasarkan nilai header didukung
Header yang ditetapkan lainnya	Pengaturan cache lama — CloudFront meneruskan header ke asal Anda.	Ya
Accept	CloudFront menghapus header.	Ya
Accept-Charset	CloudFront menghapus header.	Ya
Accept-Encoding	Jika nilainya berisi gzip atau br, CloudFront meneruskan Accept-Encoding header yang dinormalisasi ke asal Anda. Untuk informasi selengkapnya, lihat Dukungan kompresi dan Sajikan file terkompresi .	Ya
Accept-Language	CloudFront menghapus header.	Ya

Header	Perilaku jika Anda tidak mengonfigurasi CloudFront ke cache berdasarkan nilai header	Caching berdasarkan nilai header didukung
Authorization	• GET dan HEAD permintaan - CloudFront menghapus bidang Authorization header sebelum meneruskan permintaan ke asal Anda. • OPTIONS permintaan — CloudFront menghapus bidang Authorization header sebelum meneruskan permintaan ke asal Anda jika Anda mengonfigurasi respons cache CloudFront terhadap permintaan. OPTIONS CloudFront meneruskan bidang Authorization header ke asal Anda jika Anda tidak mengonfigurasi respons cache CloudFront ke permintaan OPTIONS. • DELETE, PATCH, POST, dan PUT permintaan — CloudFront tidak menghapus bidang header sebelum meneruskan permintaan ke asal Anda.	Ya
Cache-Control	CloudFront meneruskan header ke asal Anda.	Tidak
CloudFront-Forwarded-Proto	CloudFront tidak menambahkan header sebelum meneruskan permintaan ke asal Anda. Untuk informasi selengkapnya, lihat Konfigurasi caching berdasarkan protokol permintaan.	Ya

Header	Perilaku jika Anda tidak mengonfigurasi CloudFront ke cache berdasarkan nilai header	Caching berdasarkan nilai header didukung
CloudFront-Is-Desktop-Viewer	CloudFront tidak menambahkan header sebelum meneruskan permintaan ke asal Anda. Untuk informasi selengkapnya, lihat Konfigurasi caching berdasarkan jenis perangkat .	Ya
CloudFront-Is-Mobile-Viewer	CloudFront tidak menambahkan header sebelum meneruskan permintaan ke asal Anda. Untuk informasi selengkapnya, lihat Konfigurasi caching berdasarkan jenis perangkat .	Ya
CloudFront-Is-Tablet-Viewer	CloudFront tidak menambahkan header sebelum meneruskan permintaan ke asal Anda. Untuk informasi selengkapnya, lihat Konfigurasi caching berdasarkan jenis perangkat .	Ya
CloudFront-Viewer-Country	CloudFront tidak menambahkan header sebelum meneruskan permintaan ke asal Anda.	Ya
Connection	CloudFront menggantikan header ini dengan Connection: Keep-Alive sebelum meneruskan permintaan ke asal Anda.	Tidak
Content-Length	CloudFront meneruskan header ke asal Anda.	Tidak
Content-MD5	CloudFront meneruskan header ke asal Anda.	Ya

Header	Perilaku jika Anda tidak mengonfigurasi CloudFront ke cache berdasarkan nilai header	Caching berdasarkan nilai header didukung
Content-Type	CloudFront meneruskan header ke asal Anda.	Ya
Cookie	Jika Anda mengonfigurasi CloudFront untuk meneruskan cookie, itu akan meneruskan bidang Cookie header ke asal Anda. Jika tidak, CloudFront hapus bidang Cookie header. Untuk informasi selengkapnya, lihat Konten cache berdasarkan cookie .	Tidak
Date	CloudFront meneruskan header ke asal Anda.	Ya, tetapi tidak disarankan
Expect	CloudFront menghapus header.	Ya
From	CloudFront meneruskan header ke asal Anda.	Ya
Host	CloudFront menetapkan nilai ke nama domain asal yang terkait dengan objek yang diminta.	Ya (sesuai aturan)
	Anda tidak dapat melakukan cache berdasarkan judul Host untuk Amazon S3 atau MediaStore asal.	Tidak (S3 dan MediaStore)
If-Match	CloudFront meneruskan header ke asal Anda.	Ya
If-Modified-Since	CloudFront meneruskan header ke asal Anda.	Ya

Header	Perilaku jika Anda tidak mengonfigurasi CloudFront ke cache berdasarkan nilai header	Caching berdasarkan nilai header didukung
If-None-Match	CloudFront meneruskan header ke asal Anda.	Ya
If-Range	CloudFront meneruskan header ke asal Anda.	Ya
If-Unmodified-Since	CloudFront meneruskan header ke asal Anda.	Ya
Max-Forwards	CloudFront meneruskan header ke asal Anda.	Tidak
Origin	CloudFront meneruskan header ke asal Anda.	Ya
Pragma	CloudFront meneruskan header ke asal Anda.	Tidak
Proxy-Authenticate	CloudFront menghapus header.	Tidak
Proxy-Authorization	CloudFront menghapus header.	Tidak
Proxy-Connection	CloudFront menghapus header.	Tidak
Range	CloudFront meneruskan header ke asal Anda. Untuk informasi selengkapnya, lihat Bagaimana CloudFront memproses permintaan sebagian untuk suatu objek (rentang GETs) .	Ya, secara default
Referer	CloudFront menghapus header.	Ya

Header	Perilaku jika Anda tidak mengonfigurasi CloudFront ke cache berdasarkan nilai header	Caching berdasarkan nilai header didukung
Request-Range	CloudFront meneruskan header ke asal Anda.	Tidak
TE	CloudFront menghapus header.	Tidak
Trailer	CloudFront menghapus header.	Tidak
Transfer-Encoding	CloudFront meneruskan header ke asal Anda.	Tidak
Upgrade	CloudFront menghapus header, kecuali Anda telah membuat WebSocket koneksi.	Tidak (kecuali untuk WebSocket koneksi)
User-Agent	CloudFront menggantikan nilai bidang header ini dengan Amazon CloudFront. Jika Anda CloudFront ingin menyimpan konten Anda berdasarkan perangkat yang digunakan pengguna, lihat Konfigurasi caching berdasarkan jenis perangkat .	Ya, tetapi tidak disarankan
Via	CloudFront meneruskan header ke asal Anda.	Ya
Warning	CloudFront meneruskan header ke asal Anda.	Ya

Header	Perilaku jika Anda tidak mengonfigurasi CloudFront ke cache berdasarkan nilai header	Caching berdasarkan nilai header didukung
X-Amz-Cf-Id	CloudFront menambahkan header ke permintaan penampil sebelum meneruskan permintaan ke asal Anda. Nilai header berisi string terenkripsi yang secara unik mengidentifikasi permintaan.	Tidak
X-Edge-*	CloudFront menghapus semua X-Edge-* header.	Tidak
X-Forwarded-For	CloudFront meneruskan header ke asal Anda. Untuk informasi selengkapnya, lihat Alamat IP Klien .	Ya
X-Forwarded-Proto	CloudFront menghapus header.	Tidak
X-HTTP-Method-Override	CloudFront menghapus header.	Ya
X-Real-IP	CloudFront menghapus header.	Tidak

Versi HTTP

CloudFront meneruskan permintaan ke asal kustom Anda menggunakan HTTP/1.1.

Lama maksimum panjang permintaan dan lama maksimum URL

Panjang maksimum permintaan, termasuk jalur, string kueri (jika ada), dan header, adalah 32.768 byte.

CloudFront membangun URL dari permintaan. Panjang maksimal URL ini adalah 8192 byte.

Jika URL melebihi panjang maksimum, CloudFront mengembalikan kode status HTTP 414 (URI Terlalu Panjang) ke penampil. Jika permintaan melebihi panjang maksimum karena ukuran header terlampaui, CloudFront mengembalikan kode status HTTP 494 ke penampil. Dalam kedua kasus, CloudFront kemudian mengakhiri koneksi TCP ke penampil.

Pemasangan OCSP

Saat penampil mengirimkan permintaan HTTPS untuk suatu objek, salah satu CloudFront atau penampil harus mengonfirmasi dengan otoritas sertifikat (CA) bahwa sertifikat SSL untuk domain tersebut belum dicabut. Stapling OCSP mempercepat validasi sertifikat dengan memungkinkan CloudFront untuk memvalidasi sertifikat dan menyimpan respons dari CA, sehingga klien tidak perlu memvalidasi sertifikat secara langsung dengan CA.

Peningkatan performa stapling OCSP lebih jelas ketika CloudFront menerima banyak permintaan HTTPS untuk objek dalam domain yang sama. Setiap server di lokasi CloudFront tepi harus mengirimkan permintaan validasi terpisah. Saat CloudFront menerima banyak permintaan HTTPS untuk domain yang sama, setiap server di lokasi tepi segera memiliki respons dari CA yang dapat "menempatkan" ke paket dalam handshake SSL; ketika penampil puas bahwa sertifikat valid, CloudFront dapat melayani objek yang diminta. Jika distribusi Anda tidak mendapatkan banyak lalu lintas di lokasi CloudFront tepi, permintaan baru lebih mungkin diarahkan ke server yang belum memvalidasi sertifikat dengan CA. Dalam hal ini, penampil secara terpisah melakukan langkah validasi dan CloudFront server melayani objek. CloudFront Server itu juga mengirimkan permintaan validasi ke CA, jadi lain kali menerima permintaan yang menyertakan nama domain yang sama, ia memiliki respons validasi dari CA.

Koneksi persisten

Ketika CloudFront mendapat respons dari asal Anda, ia mencoba mempertahankan koneksi selama beberapa detik jika permintaan lain tiba selama periode itu. Mempertahankan koneksi yang persisten menghemat waktu yang diperlukan untuk membangun kembali koneksi TCP dan melakukan handshake TLS lain untuk permintaan berikutnya.

Untuk informasi lebih lanjut, termasuk cara mengonfigurasi durasi koneksi persisten, lihat [Keep-alive batas waktu \(khusus dan hanya asal VPC\)](#) di bagian [Semua referensi pengaturan distribusi](#).

Protokol

CloudFront meneruskan permintaan HTTP atau HTTPS ke server asal berdasarkan hal berikut:

- Protokol permintaan yang dikirimkan oleh penampil CloudFront, baik HTTP atau HTTPS.

- Nilai bidang Kebijakan Protokol Asal di CloudFront konsol atau, jika Anda menggunakan CloudFront API, `OriginProtocolPolicy` elemen dalam tipe `DistributionConfig` kompleks. Di CloudFront konsol, opsinya adalah HTTP Only, HTTPS Only, dan Match Viewer.

Jika Anda menentukan HTTP Only atau HTTPS Only, CloudFront teruskan permintaan ke server asal menggunakan protokol yang ditentukan, terlepas dari protokol dalam permintaan penampil.

Jika Anda menentukan Penampil Pencocokan, CloudFront teruskan permintaan ke server asal menggunakan protokol dalam permintaan penampil. Perhatikan bahwa CloudFront menyimpan objek hanya sekali jika penampil membuat permintaan menggunakan protokol HTTP dan HTTPS.

Important

Jika CloudFront meneruskan permintaan ke asal menggunakan protokol HTTPS, dan jika server asal mengembalikan sertifikat yang tidak valid atau sertifikat yang ditandatangani sendiri, lepaskan koneksi TCP CloudFront.

Untuk informasi tentang cara memperbarui distribusi menggunakan CloudFront konsol, lihat [Perbarui distribusi](#). Untuk informasi tentang cara memperbarui distribusi menggunakan CloudFront API, buka Referensi CloudFront API Amazon. [UpdateDistribution](#)

String pertanyaan

Anda dapat mengonfigurasi apakah CloudFront meneruskan parameter string kueri ke asal Anda. Untuk informasi selengkapnya, lihat [Konten cache berdasarkan parameter string kueri](#).

Waktu habis dan upaya koneksi tempat asal

Batas waktu koneksi asal adalah jumlah detik yang CloudFront menunggu ketika mencoba membuat koneksi ke asal.

Upaya koneksi asal adalah berapa kali CloudFront upaya untuk terhubung ke asal.

Bersama-sama, pengaturan ini menentukan berapa lama CloudFront mencoba untuk terhubung ke asal sebelum gagal ke asal sekunder (dalam kasus grup asal) atau mengembalikan respons kesalahan ke penampil. Secara default, CloudFront tunggu selama 30 detik (3 upaya masing-masing 10 detik) sebelum mencoba terhubung ke asal sekunder atau mengembalikan respons kesalahan.

Anda dapat mengurangi waktu ini dengan menentukan waktu koneksi yang lebih singkat, lebih sedikit percobaan, atau keduanya.

Untuk informasi selengkapnya, lihat [Kontrol batas waktu dan upaya asal](#).

Waktu habis untuk respons asal

waktu habis respons asal, juga dikenal sebagai waktu habis baca asal atau waktu habis permintaan asal, berlaku untuk kedua hal berikut:

- Jumlah waktu, dalam hitungan detik, yang CloudFront menunggu respons setelah meneruskan permintaan ke asal.
- Jumlah waktu, dalam hitungan detik, yang CloudFront menunggu setelah menerima paket respons dari asal dan sebelum menerima paket berikutnya.

CloudFront perilaku tergantung pada metode HTTP dari permintaan penampil:

- GET dan HEAD permintaan - Jika asal tidak merespons atau berhenti merespons dalam durasi waktu tunggu respons, hentikan CloudFront koneksi. Jika jumlah [upaya koneksi asal](#) yang ditentukan lebih dari 1, CloudFront coba lagi untuk mendapatkan respons lengkap. CloudFront mencoba hingga 3 kali, sebagaimana ditentukan oleh nilai pengaturan upaya koneksi asal. Jika asal tidak merespons selama percobaan terakhir, CloudFront tidak mencoba lagi hingga menerima permintaan konten lain di tempat yang sama.
- DELETE, OPTIONS PATCH, PUT, dan POST permintaan — Jika asal tidak merespons selama durasi batas waktu baca, CloudFront hentikan koneksi dan tidak mencoba lagi untuk menghubungi asal. Klien dapat mengirim ulang permintaan bilamana perlu.

Untuk informasi lebih lanjut, termasuk cara mengonfigurasi waktu habis respons asal, lihat [Batas waktu respons](#).

Permintaan simultan untuk objek yang sama (permintaan runtuh)

Ketika lokasi CloudFront tepi menerima permintaan untuk objek dan objek tidak dalam cache atau objek yang di-cache kedaluwarsa, CloudFront segera kirim permintaan ke asal. Namun, jika ada permintaan simultan untuk objek yang sama—yaitu, jika permintaan tambahan untuk objek yang sama (dengan kunci cache yang sama) tiba di lokasi tepi sebelum CloudFront menerima respons terhadap permintaan pertama— CloudFront berhenti sebelum meneruskan permintaan tambahan

ke asal. Jeda singkat ini membantu mengurangi beban pada titik asal. CloudFront mengirimkan respons dari permintaan asli ke semua permintaan yang diterimanya saat dijeda. Ini disebut request collapsing. Dalam CloudFront log, permintaan pertama diidentifikasi sebagai a Miss di x-edge-result-type bidang, dan permintaan yang diciutkan diidentifikasi sebagai aHit. Untuk informasi selengkapnya tentang CloudFront log, lihat [the section called “CloudFront dan logging fungsi tepi”](#).

CloudFront hanya menciutkan permintaan yang berbagi [kunci cache](#). Jika permintaan tambahan tidak berbagi kunci cache yang sama karena, misalnya, Anda CloudFront mengonfigurasi cache berdasarkan header permintaan atau cookie atau string kueri, CloudFront teruskan semua permintaan dengan kunci cache unik ke asal Anda.

Jika Anda ingin mencegah semua permintaan runtuh, Anda dapat menggunakan kebijakan cache terkelola `CachingDisabled`, yang juga mencegah caching. Untuk informasi selengkapnya, lihat [Gunakan kebijakan cache terkelola](#).

Jika Anda ingin mencegah keruntuhan permintaan untuk objek tertentu, Anda dapat mengatur TTL minimum untuk perilaku cache ke 0 dan mengonfigurasi asal untuk mengirim `Cache-Control: private,, Cache-Control: no-store Cache-Control: no-cache Cache-Control: max-age=0`, atau. `Cache-Control: s-maxage=0` Konfigurasi ini akan meningkatkan beban pada asal Anda dan memperkenalkan latensi tambahan untuk permintaan simultan yang dijeda sementara CloudFront menunggu respons terhadap permintaan pertama.

Header **User-Agent**

Jika Anda CloudFront ingin menyimpan versi objek yang berbeda berdasarkan perangkat yang digunakan pengguna untuk melihat konten Anda, kami sarankan Anda mengonfigurasi CloudFront untuk meneruskan satu atau beberapa header berikut ke asal kustom Anda:

- `CloudFront-Is-Desktop-Viewer`
- `CloudFront-Is-Mobile-Viewer`
- `CloudFront-Is-SmartTV-Viewer`
- `CloudFront-Is-Tablet-Viewer`

Berdasarkan nilai `User-Agent` header, CloudFront tetapkan nilai header ini ke `true` atau `false` sebelum meneruskan permintaan ke asal Anda. Jika perangkat termasuk dalam lebih dari satu kategori, lebih dari satu nilai mungkin `true`. Misalnya, untuk beberapa perangkat tablet, CloudFront mungkin mengatur keduanya `CloudFront-Is-Mobile-Viewer` dan `CloudFront-Is-Tablet-`

Viewer `true`. Untuk informasi selengkapnya tentang CloudFront mengonfigurasi cache berdasarkan header permintaan, lihat. [Konten cache berdasarkan header permintaan](#)

Anda dapat CloudFront mengonfigurasi objek cache berdasarkan nilai di `User-Agent` header, tetapi kami tidak merekomendasikannya. `User-Agent` header memiliki banyak nilai yang mungkin, dan caching berdasarkan nilai-nilai tersebut akan menyebabkan CloudFront untuk meneruskan lebih banyak permintaan secara signifikan ke asal Anda.

Jika Anda tidak CloudFront mengonfigurasi objek cache berdasarkan nilai di `User-Agent` header, CloudFront tambahkan `User-Agent` header dengan nilai berikut sebelum meneruskan permintaan ke asal Anda:

`User-Agent = Amazon CloudFront`

CloudFront menambahkan header ini terlepas dari apakah permintaan dari penampil menyertakan `User-Agent` header. Jika permintaan dari penampil menyertakan `User-Agent` header, CloudFront hapus itu.

Bagaimana CloudFront memproses tanggapan dari asal kustom Anda

Pelajari cara CloudFront memproses respons dari asal kustom Anda.

Daftar Isi

- [100 Continuetanggapan](#)
- [Pembuatan cache](#)
- [Permintaan dibatalkan](#)
- [Negosiasi konten](#)
- [Cookie](#)
- [Koneksi TCP yang terhenti](#)
- [Header respons HTTP yang CloudFront menghapus atau menggantikan](#)
- [Ukuran file cache maksimum](#)
- [Tempat asal tidak tersedia](#)
- [Mengalihkan](#)
- [Header Transfer-Encoding](#)

100 Continue tanggapan

Asal Anda tidak dapat mengirim lebih dari satu respons 100-Lanjutkan ke CloudFront. Setelah respons 100-Continue pertama, CloudFront mengharapkan respons HTTP 200 OK. Jika asal Anda mengirimkan respons 100-Lanjutkan lagi setelah yang pertama, CloudFront akan mengembalikan kesalahan.

Pembuatan cache

- Pastikan server asal menetapkan nilai yang valid dan akurat untuk Date dan Last-Modified bidang header.
- CloudFront biasanya menghormati Cache-Control: no-cache header dalam respons dari asal. Untuk pengecualian, lihat [Permintaan simultan untuk objek yang sama \(permintaan runtuh\)](#).

Permintaan dibatalkan

Jika suatu objek tidak berada di cache tepi, dan jika penampil mengakhiri sesi (misalnya, menutup browser) setelah CloudFront mendapatkan objek dari asal Anda tetapi sebelum dapat mengirimkan objek yang diminta, CloudFront tidak men-cache objek di lokasi tepi.

Negosiasi konten

Jika asal Anda kembali Vary : * dalam respons, dan jika nilai TTL Minimum untuk perilaku cache yang sesuai adalah 0, CloudFront cache objek tetapi masih meneruskan setiap permintaan berikutnya untuk objek ke asal untuk mengonfirmasi bahwa cache berisi versi terbaru dari objek. CloudFront tidak termasuk header bersyarat, seperti If-None-Match atau If-Modified-Since. Akibatnya, asal Anda mengembalikan objek sebagai CloudFront respons terhadap setiap permintaan.

Jika asal Anda kembali Vary : * dalam respons, dan jika nilai TTL Minimum untuk perilaku cache yang sesuai adalah nilai lainnya, CloudFront proses Vary header seperti yang dijelaskan dalam [Header respons HTTP yang CloudFront menghapus atau menggantikan](#).

Cookie

Jika Anda mengaktifkan cookie untuk perilaku cache, dan jika asal mengembalikan cookie dengan objek, CloudFront cache objek dan cookie. Perhatikan bahwa ini mengurangi kemungkinan cache untuk sebuah objek. Untuk informasi selengkapnya, lihat [Konten cache berdasarkan cookie](#).

Koneksi TCP yang terhenti

Jika koneksi TCP antara CloudFront dan asal Anda turun saat asal Anda mengembalikan objek CloudFront, CloudFront perilaku tergantung pada apakah asal Anda menyertakan Content-Length header dalam respons:

- Judul Panjang Konten – CloudFront mengembalikan objek ke penampil saat objek berasal dari negara asal Anda. Namun, jika nilai Content-Length header tidak sesuai dengan ukuran objek, objek CloudFront tidak disimpan dalam cache.
- Transfer-Encoding: Chunked — CloudFront mengembalikan objek ke penampil karena mendapatkan objek dari asal Anda. Namun, jika respon chunked tidak lengkap, CloudFront tidak cache objek.
- Tidak ada header Panjang Konten – CloudFront mengembalikan objek ke penampil dan menyimpannya dalam cache, tetapi objek mungkin tidak lengkap. Tanpa Content-Length header, CloudFront tidak dapat menentukan apakah koneksi TCP jatuh secara tidak sengaja atau dengan sengaja.

Kami menyarankan Anda mengonfigurasi server HTTP Anda untuk menambahkan Content-Length header untuk CloudFront mencegah caching objek sebagian.

Header respons HTTP yang CloudFront menghapus atau menggantikan

CloudFront menghapus atau memperbarui bidang header berikut sebelum meneruskan respons dari asal Anda ke penampil:

- Set-Cookie— Jika Anda mengonfigurasi CloudFront untuk meneruskan cookie, itu akan meneruskan bidang Set-Cookie header ke klien. Untuk informasi selengkapnya, lihat [Konten cache berdasarkan cookie](#).
- Trailer
- Transfer-Encoding— Jika asal Anda mengembalikan bidang header ini, CloudFront tetapkan nilainya chunked sebelum mengembalikan respons ke penampil.
- Upgrade
- Vary – Catat hal berikut:
 - Jika Anda mengonfigurasi CloudFront untuk meneruskan header khusus perangkat ke origin (CloudFront-Is-Desktop-Viewer,,CloudFront-Is-Mobile-Viewer,CloudFront-Is-Tablet-Viewer) dan mengonfigurasi asal Anda untuk kembaliCloudFront-Is-SmartTV-

Viewer, kembali Vary:User-Agent ke CloudFront penampil CloudFront. Vary:User-Agent Untuk informasi selengkapnya, lihat [Konfigurasi caching berdasarkan jenis perangkat](#).

- Jika Anda mengonfigurasi asal Anda untuk menyertakan salah satu Accept-Encoding atau Cookie di Vary header, CloudFront sertakan nilai dalam respons terhadap penampil.
- Jika Anda mengonfigurasi CloudFront untuk meneruskan header ke asal Anda, dan jika Anda mengonfigurasi asal Anda untuk mengembalikan nama header ke CloudFront Vary header (misalnya, Vary:Accept-Charset, Accept-Language), CloudFront mengembalikan Vary header dengan nilai-nilai tersebut ke penampil.
- Untuk informasi tentang cara CloudFront memproses nilai * di Vary header, lihat [Negosiasi konten](#).
- Jika Anda mengonfigurasi asal Anda untuk menyertakan nilai lain di Vary header, CloudFront hapus nilai sebelum mengembalikan respons ke penampil.
- Via— CloudFront menetapkan nilai sebagai berikut dalam respons terhadap penampil:

Via: *http-version alphanumeric-string*.cloudfront.net (CloudFront)

Misalnya, nilainya adalah seperti berikut:

Via: 1.1 1026589cc7887e7a0dc7827b4example.cloudfront.net (CloudFront)

Ukuran file cache maksimum

Ukuran maksimum badan respons yang CloudFront menyimpan dalam cache-nya adalah 50 GB. Ini termasuk respons transfer yang dipotong yang tidak menyebutkan nilai header Content-Length.

Anda dapat CloudFront menggunakan cache objek yang lebih besar dari ukuran ini dengan menggunakan permintaan rentang untuk meminta objek di bagian yang masing-masing 50 GB atau lebih kecil. CloudFront cache bagian-bagian ini karena masing-masing dari mereka adalah 50 GB atau lebih kecil. Setelah penampil mengambil semua bagian objek, ia dapat merekonstruksi objek asli yang lebih besar. Untuk informasi selengkapnya, lihat [Gunakan permintaan rentang untuk menyimpan objek besar](#).

Tempat asal tidak tersedia

Jika server asal Anda tidak tersedia dan CloudFront mendapat permintaan untuk objek yang ada di cache tepi tetapi telah kedaluwarsa (misalnya, karena periode waktu yang ditentukan dalam Cache-Control max-age arahan telah berlalu), CloudFront baik menyajikan versi kedaluwarsa objek atau

menyajikan halaman kesalahan khusus. Untuk informasi selengkapnya tentang CloudFront perilaku saat Anda mengonfigurasi halaman kesalahan kustom, lihat [Bagaimana CloudFront proses kesalahan ketika Anda telah mengonfigurasi halaman kesalahan kustom](#).

Dalam beberapa kasus, objek yang jarang diminta diusir dan tidak lagi tersedia di cache tepi. CloudFront tidak dapat melayani objek yang telah diusir.

Mengalihkan

Jika Anda mengubah lokasi objek di server asal, Anda dapat mengonfigurasi server web Anda untuk mengalihkan permintaan ke lokasi baru. Setelah Anda mengonfigurasi pengalihan, pertama kali penampil mengirimkan permintaan untuk objek, CloudFront mengirim permintaan ke asal, dan asal merespons dengan pengalihan (misalnya, `302 Moved Temporarily`). CloudFront cache pengalihan dan mengembalikannya ke penampil. CloudFront tidak mengikuti pengalihan.

Anda dapat mengonfigurasi server web untuk mengalihkan permintaan ke salah satu lokasi berikut:

- URL baru objek di server asal. Saat penampil mengikuti pengalihan ke URL baru, penampil mem-bypass CloudFront dan langsung menuju ke asal. Akibatnya, kami menyarankan agar Anda tidak mengalihkan permintaan ke URL baru objek pada asal.
- CloudFront URL baru untuk objek. Saat penampil mengirimkan permintaan yang berisi CloudFront URL baru, CloudFront dapatkan objek dari lokasi baru di asal Anda, menyimpannya di lokasi tepi, dan mengembalikan objek ke penampil. Permintaan berikutnya atas objek tersebut akan dilayani oleh lokasi edge. Ini menghindari latensi dan beban yang terkait dengan penampil yang meminta objek dari asal. Namun, setiap permintaan baru untuk objek akan dikenakan biaya untuk dua permintaan untuk CloudFront.

Header **Transfer-Encoding**

CloudFront hanya mendukung chunked nilai `Transfer-Encoding` header. Jika asal Anda kembali `Transfer-Encoding: chunked`, CloudFront mengembalikan objek ke klien sebagai objek diterima di lokasi tepi, dan cache objek dalam format chunked untuk permintaan berikutnya.

Jika penampil membuat `Range GET` permintaan dan asal kembali `Transfer-Encoding: chunked`, CloudFront mengembalikan seluruh objek ke penampil, bukan rentang yang diminta.

Kami sarankan Anda menggunakan pengkodean bertahap jika panjang konten tanggapan Anda tidak dapat ditentukan sebelumnya. Untuk informasi selengkapnya, lihat [Koneksi TCP yang terhenti](#).

Perilaku permintaan dan respons untuk grup asal

Permintaan kepada kelompok asal bekerja sama dengan permintaan ke asal yang tidak diatur sebagai kelompok asal, kecuali saat ada failover asal. Seperti halnya asal lainnya, ketika CloudFront menerima permintaan dan konten sudah di-cache di lokasi tepi, konten disajikan kepada pemirsa dari cache. Ketika ada kesalahan cache dan asal adalah kelompok asal, permintaan penampil diteruskan ke asal utama dalam kelompok asal.

Permintaan dan perilaku respons untuk asal mula utama sama dengan untuk asal yang tidak berada dalam kelompok asal. Untuk informasi lebih lanjut, lihat [Perilaku permintaan dan respons untuk asal Amazon S3](#) dan [Perilaku permintaan dan respons untuk asal kustom](#).

Berikut ini menguraikan perilaku untuk failover asal ketika asal-usul utama mengembalikan kode status HTTP tertentu:

- Kode status HTTP 2xx (sukses): CloudFront menyimpan file dan mengembalikannya ke penampil.
- Kode status HTTP 3xx (pengalihan): CloudFront mengembalikan kode status ke penampil.
- Kode status HTTP 4xx atau 5xx (kesalahan klien/server): Jika kode status yang dikembalikan telah dikonfigurasi untuk failover, CloudFront mengirimkan permintaan yang sama ke asal sekunder di grup asal.
- Kode status HTTP 4xx atau 5xx (kesalahan klien/server): Jika kode status yang dikembalikan belum dikonfigurasi untuk failover, CloudFront mengembalikan kesalahan ke penampil.

CloudFront gagal ke asal sekunder hanya jika metode HTTP dari permintaan penampil adalah GET, HEAD, atau OPTIONS. CloudFront tidak gagal ketika penampil mengirim metode HTTP yang berbeda (misalnya POST, PUT, dan sebagainya).

Saat CloudFront mengirim permintaan ke asal sekunder, perilaku responsnya sama dengan CloudFront asal yang tidak ada dalam grup asal.

Untuk informasi selengkapnya tentang kelompok asal, lihat [Optimalkan ketersediaan tinggi dengan failover CloudFront asal](#).

Tambahkan header khusus ke permintaan asal

Anda dapat mengonfigurasi CloudFront untuk menambahkan header khusus ke permintaan yang dikirimkan ke asal Anda. Anda dapat menggunakan header khusus untuk mengirim dan

mengumpulkan informasi dari asal Anda yang tidak Anda dapatkan dengan permintaan pemirsa biasa. Anda bahkan dapat menyesuaikan header untuk setiap asal. CloudFront mendukung header khusus untuk asal kustom dan asal Amazon S3.

Daftar Isi

- [Kasus penggunaan](#)
- [Konfigurasi CloudFront untuk menambahkan header khusus ke permintaan asal](#)
- [Header khusus yang tidak CloudFront dapat ditambahkan ke permintaan asal](#)
- [CloudFront Konfigurasi untuk meneruskan Authorization header](#)

Kasus penggunaan

Anda dapat menggunakan header khusus, seperti contoh berikut:

Mengidentifikasi permintaan dari CloudFront

Anda dapat mengidentifikasi permintaan yang diterima asal Anda CloudFront. Ini dapat berguna jika Anda ingin tahu apakah pengguna melewati CloudFront, atau jika Anda menggunakan lebih dari satu CDN dan Anda menginginkan informasi tentang permintaan mana yang berasal dari setiap CDN.

Note

Jika Anda menggunakan asal Amazon S3 dan Anda mengaktifkan [Pencatatan akses server Amazon S3](#), log tidak menyertakan informasi header.

Menentukan permintaan yang berasal dari distribusi tertentu

Jika Anda mengonfigurasi lebih dari satu CloudFront distribusi untuk menggunakan asal yang sama, Anda dapat menambahkan header khusus yang berbeda di setiap distribusi. Anda lalu bisa menggunakan log dari tempat asal Anda untuk menentukan permintaan mana yang berasal dari distribusi CloudFront.

Mengaktifkan pembagian sumber daya lintas-asal (CORS)

Jika beberapa penonton tidak mendukung berbagi sumber daya lintas asal (CORS), Anda dapat mengonfigurasi CloudFront agar selalu menambahkan `Origin` header ke permintaan yang

dikirimkan ke asal Anda. Lalu Anda bisa mengonfigurasi asal Anda untuk mengembalikan header `Access-Control-Allow-Origin` untuk setiap permintaan. Anda juga harus [mengonfigurasi CloudFront untuk mengikuti pengaturan CORS](#).

Mengendalikan akses ke konten

Anda bisa menggunakan header kustom untuk mengontrol akses ke konten. Dengan mengonfigurasi asal Anda untuk menanggapi permintaan hanya ketika permintaan tersebut menyertakan header khusus yang ditambahkan CloudFront, Anda mencegah pengguna melewati CloudFront dan mengakses konten Anda langsung di asal. Untuk informasi selengkapnya, lihat [Batasi akses ke file pada asal kustom](#).

Konfigurasi CloudFront untuk menambahkan header khusus ke permintaan asal

Untuk mengonfigurasi distribusi guna menambahkan header kustom ke permintaan yang dikirim ke asal Anda, perbarui konfigurasi asal menggunakan salah satu metode berikut:

- CloudFront konsol — Saat Anda membuat atau memperbarui distribusi, tentukan nama dan nilai header dalam pengaturan Tambahkan header khusus. Untuk informasi selengkapnya, lihat [Tambahkan header kustom](#).
- CloudFront API - Untuk setiap asal yang ingin Anda tambahkan header khusus, tentukan nama dan nilai header di `CustomHeaders` bidang di dalamnya `Origin`. Untuk informasi selengkapnya, lihat [CreateDistribution](#) atau [UpdateDistribution](#) di Referensi Amazon CloudFront API.

Jika nama header dan nilai yang Anda tentukan belum ada dalam permintaan penampil, CloudFront tambahkan ke permintaan asal. Jika ada header, CloudFront timpa nilai header sebelum meneruskan permintaan ke asal.

Untuk kuota yang berlaku untuk header kustom asal, lihat. [Kuota pada header](#)

Header khusus yang tidak CloudFront dapat ditambahkan ke permintaan asal

Anda tidak dapat mengonfigurasi CloudFront untuk menambahkan header berikut ke permintaan yang dikirimkan ke asal Anda:

- `Cache-Control`

- Connection
- Content-Length
- Cookie
- Host
- If-Match
- If-Modified-Since
- If-None-Match
- If-Range
- If-Unmodified-Since
- Max-Forwards
- Pragma
- Proxy-Authenticate
- Proxy-Authorization
- Proxy-Connection
- Range
- Request-Range
- TE
- Trailer
- Transfer-Encoding
- Upgrade
- Via
- Header yang dimulai dengan X-Amz -
- Header yang dimulai dengan X-Edge -
- X-Real-Ip

CloudFront Konfigurasi untuk meneruskan **Authorization** header

Saat CloudFront meneruskan permintaan penampil ke asal Anda, CloudFront hapus beberapa header penampil secara default, termasuk header. Authorization Untuk memastikan bahwa asal Anda selalu menerima header Authorization di permintaan asal, Anda memiliki opsi berikut:

- Tambahkan header `Authorization` ke kunci cache menggunakan kebijakan cache. Semua header di kunci cache secara otomatis disertakan dalam permintaan asal. Untuk informasi selengkapnya, lihat [Mengontrol kunci cache dengan kebijakan](#).
- Tambahkan `Authorization` header satu per satu dalam kebijakan permintaan asal. Untuk informasi selengkapnya, lihat [Kontrol permintaan asal dengan kebijakan](#).
- Gunakan kebijakan permintaan asal yang meneruskan semua header penampil ke asal. CloudFront menyediakan kebijakan permintaan asal terkelola untuk kasus penggunaan ini, yang disebut Managed- AllViewer. Untuk informasi selengkapnya, lihat [Gunakan kebijakan permintaan asal terkelola](#).

Important

Jika Anda meneruskan `Authorization` header ke asal Anda tanpa memasukkannya ke dalam kunci cache, pastikan bahwa asal Anda tidak bergantung pada `Authorization` header untuk kontrol akses konten yang di-cache. Ketika `Authorization` header bukan bagian dari kunci cache, CloudFront dapat menyajikan respons cache yang sama untuk pemirsa yang berwenang dan tidak sah. Sertakan `Authorization` header dalam kunci cache menggunakan kebijakan cache, atau nonaktifkan caching sepenuhnya untuk asal yang memerlukan pemrosesan otorisasi sisi asal.

Bagaimana CloudFront memproses permintaan sebagian untuk suatu objek (rentang GETs)

Untuk objek besar, penampil (browser web atau klien lain) dapat membuat beberapa GET permintaan dan menggunakan header `Range` permintaan untuk mengunduh objek di bagian yang lebih kecil. Permintaan untuk rentang byte, terkadang disebut `Range GET` meminta, meningkatkan efisiensi unduhan sebagian, dan pemulihan dari transfer yang gagal sebagian.

Saat CloudFront menerima `Range GET` permintaan, ia memeriksa cache di lokasi tepi yang menerima permintaan. Jika cache di lokasi tepi itu sudah berisi seluruh objek atau bagian objek yang diminta, CloudFront segera layani rentang yang diminta dari cache.

Jika cache tidak berisi rentang yang diminta, CloudFront teruskan permintaan ke asal. (Untuk mengoptimalkan kinerja, CloudFront dapat meminta rentang yang lebih besar dari yang diminta klien

diRange GET.) Apa yang terjadi selanjutnya tergantung pada apakah asal mendukung permintaan Range GET:

- Jika asal mendukung **Range GET** permintaan — Ini mengembalikan rentang yang diminta. CloudFront melayani rentang yang diminta dan juga menyimpannya dalam cache untuk permintaan masa depan. (Amazon S3 mendukung Range GET permintaan, seperti halnya banyak server HTTP.)
- Jika asal tidak mendukung **Range GET** permintaan - Ini mengembalikan seluruh objek. CloudFront melayani permintaan saat ini dengan mengirimkan seluruh objek sementara juga men-cache untuk permintaan future. Setelah CloudFront cache seluruh objek dalam cache tepi, ia merespons Range GET permintaan baru dengan menyajikan rentang yang diminta.

Dalam kedua kasus tersebut, CloudFront mulailah melayani rentang atau objek yang diminta ke pengguna akhir segera setelah byte pertama tiba dari asal.

Note

Jika penampil membuat Range GET permintaan dan asal kembali `Transfer-Encoding: chunked`, CloudFront mengembalikan seluruh objek ke penampil, bukan rentang yang diminta.

CloudFront umumnya mengikuti spesifikasi RFC untuk Range header. Namun, jika Range header Anda tidak mematuhi persyaratan berikut, CloudFront mengembalikan kode status HTTP 200 dengan objek lengkap, bukan kode status 206 dengan rentang yang ditentukan:

- Rentang harus terdaftar dalam urutan naik. Misalnya, `100-200, 300-400` valid, `300-400, 100-200` tidak valid.
- Rentang tersebut tidak boleh tumpang tindih. Misalnya, `100-200, 150-250` tidak valid.
- Semua spesifikasi rentang harus valid. Misalnya, Anda tidak dapat menentukan nilai negatif sebagai bagian dari rentang.

Untuk informasi selengkapnya tentang header Range permintaan, lihat [Permintaan Rentang](#) di RFC 7233, atau [Rentang di Dokumen](#) Web MDN.

Gunakan permintaan rentang untuk menyimpan objek besar

Ketika caching diaktifkan, CloudFront tidak mengambil atau cache objek yang lebih besar dari 50 GB. Ketika asal menunjukkan bahwa objek lebih besar dari ukuran ini (di header `Content-Length` respons), CloudFront menutup koneksi ke asal dan mengembalikan kesalahan ke penampil. (Dengan caching dinonaktifkan, CloudFront dapat mengambil objek yang lebih besar dari ukuran ini dari asal dan meneruskannya ke penampil. Namun, CloudFront tidak men-cache objek.)

Namun, dengan permintaan rentang, Anda dapat CloudFront menggunakan cache objek yang lebih besar dari ukuran [file cache maksimum](#).

Example Contoh

1. Pertimbangkan asal dengan objek 100 GB. Dengan caching diaktifkan, CloudFront tidak mengambil atau menyimpan objek sebesar ini. Namun, penampil dapat mengirim beberapa permintaan rentang untuk mengambil objek ini dalam beberapa bagian, dengan setiap bagian lebih kecil dari 50 GB.
2. Penampil dapat meminta objek dalam bagian 20 GB dengan mengirimkan permintaan dengan header `Range: bytes=0-21474836480` untuk mengambil bagian pertama, permintaan lain dengan header `Range: bytes=21474836481-42949672960` untuk mengambil bagian berikutnya, dan seterusnya.
3. Ketika pemirsa telah menerima semua bagian, itu dapat menggabungkannya untuk membangun objek 100 GB asli.
4. Dalam hal ini, CloudFront cache masing-masing bagian 20 GB dari objek dan dapat menanggapi permintaan berikutnya untuk bagian yang sama dari cache.

Untuk permintaan rentang untuk objek terkompresi, permintaan rentang byte didasarkan pada ukuran terkompresi, dan bukan ukuran asli objek. Untuk informasi selengkapnya tentang mengompresi file, lihat [Sajikan file terkompresi](#).

Bagaimana CloudFront memproses kode status HTTP 3xx dari asal

Anda

Saat CloudFront meminta objek dari bucket Amazon S3 atau server asal khusus, asal Anda terkadang menampilkan kode status HTTP 3xx. Ini biasanya menunjukkan salah satu hal berikut:

- URL objek telah berubah (misalnya, kode status 301, 302, 307, atau 308)

- Objek tidak berubah sejak terakhir kali CloudFront memintanya (kode status 304)

CloudFront cache tanggapan 3xx sesuai dengan pengaturan dalam CloudFront distribusi Anda dan header dalam respons. CloudFront cache 307 dan 308 tanggapan hanya ketika Anda menyertakan `Cache-Control` header dalam tanggapan dari asal. Untuk informasi selengkapnya, lihat [Mengelola berapa lama konten tetap dalam cache \(kedaluwarsa\)](#).

Jika asal Anda mengembalikan kode status pengalihan (misalnya, 301 atau 307), CloudFront tidak mengikuti pengalihan. CloudFront meneruskan respons 301 atau 307 kepada pemirsa, yang dapat mengikuti pengalihan dengan mengirimkan permintaan baru.

Bagaimana CloudFront memproses kode status HTTP 4xx dan 5xx dari asal Anda

Saat CloudFront meminta objek dari bucket Amazon S3 atau server asal kustom, asal Anda terkadang menampilkan kode status HTTP 4xx atau 5xx, yang menunjukkan bahwa telah terjadi kesalahan. CloudFront perilaku tergantung pada:

- Apakah Anda telah mengkonfigurasi halaman kesalahan kustom
- Apakah Anda telah mengonfigurasi berapa lama Anda CloudFront ingin menyimpan respons kesalahan cache dari asal Anda (kesalahan caching minimum TTL)
- Kode status
- Untuk kode status 5xx, apakah objek yang diminta saat ini berada di cache CloudFront tepi
- Untuk beberapa kode status 4xx, apakah asal mengembalikan header `Cache-Control max-age` atau `Cache-Control s-maxage`

CloudFront selalu menyimpan respons GET dan HEAD permintaan. Anda juga dapat CloudFront mengonfigurasi respons cache terhadap OPTIONS permintaan. CloudFront tidak menyimpan respons terhadap permintaan yang menggunakan metode lain.

Jika asal tidak merespons, CloudFront permintaan ke waktu asal habis yang dianggap sebagai kesalahan HTTP 5xx dari asal, meskipun asal tidak merespons dengan kesalahan itu. Dalam skenario itu, CloudFront terus menyajikan konten yang di-cache. Untuk informasi selengkapnya, lihat [Tempat asal tidak tersedia](#).

Jika Anda telah mengaktifkan logging, CloudFront tulis hasilnya ke log terlepas dari kode status HTTP.

Untuk informasi selengkapnya tentang fitur dan opsi yang terkait dengan pesan galat yang dikembalikan CloudFront, lihat berikut ini:

- Untuk informasi tentang pengaturan untuk halaman kesalahan kustom di CloudFront konsol, lihat [Halaman kesalahan kustom dan caching kesalahan](#).
- Untuk informasi tentang kesalahan cache TTL minimum di CloudFront konsol, lihat. [Kesalahan caching minimum TTL \(detik\)](#)
- Untuk daftar kode status HTTP yang CloudFront di-cache, lihat [Kode status HTTP 4xx dan 5xx yang di-cache CloudFront](#).

Topik

- [Bagaimana CloudFront proses kesalahan ketika Anda telah mengkonfigurasi halaman kesalahan kustom](#)
- [Cara CloudFront memproses kesalahan jika Anda belum mengonfigurasi halaman kesalahan kustom](#)
- [Kode status HTTP 4xx dan 5xx yang di-cache CloudFront](#)

Bagaimana CloudFront proses kesalahan ketika Anda telah mengkonfigurasi halaman kesalahan kustom

Jika Anda telah mengonfigurasi halaman kesalahan kustom, CloudFront perilaku tergantung pada apakah objek yang diminta ada di cache tepi.

Objek yang diminta tidak ada di cache tepi

CloudFront terus mencoba untuk mendapatkan objek yang diminta dari asal Anda ketika semua hal berikut benar:

- Penampil meminta sebuah objek.
- Objek tidak berada dalam cache tepi.
- Asal Anda mengembalikan kode status HTTP 4xx atau 5xx dan salah satu dari yang berikut adalah benar:

- Asal Anda mengembalikan kode status HTTP 5xx, bukan mengembalikan kode status 304 (Not Modified) atau versi terbaru dari objek.
- Kota asal Anda mengembalikan kode status HTTP 4xx yang tidak dibatasi oleh header kontrol cache dan termasuk dalam daftar kode status berikut: [Kode status HTTP 4xx dan 5xx yang di-cache CloudFront](#).
- Asal Anda mengembalikan kode status HTTP 4xx dengan Cache-Control max-age header atau Cache-Control s-maxage header, dan kode status disertakan dalam daftar kode status berikut: [Kontrol Kode status HTTP 4xx yang CloudFront di-cache berdasarkan header Cache-Control](#).

CloudFront melakukan hal berikut:

1. Di cache CloudFront tepi yang menerima permintaan penampil, CloudFront memeriksa konfigurasi distribusi Anda dan dapatkan jalur halaman kesalahan kustom yang sesuai dengan kode status yang dikembalikan asal Anda.
2. CloudFront menemukan perilaku cache pertama dalam distribusi Anda yang memiliki pola jalur yang cocok dengan jalur halaman kesalahan kustom.
3. Lokasi CloudFront tepi mengirimkan permintaan untuk halaman kesalahan kustom ke asal yang ditentukan dalam perilaku cache.
4. Kota asal mengembalikan halaman kesalahan kustom ke lokasi tepi.
5. CloudFront mengembalikan halaman kesalahan kustom ke penampil yang membuat permintaan, dan juga cache halaman kesalahan kustom untuk maksimum berikut ini:
 - Jumlah waktu yang ditentukan oleh kesalahan caching minimum TTL (10 detik secara default)
 - Jumlah waktu yang ditentukan oleh Cache-Control max-age header atau Cache-Control s-maxage header yang dikembalikan oleh asal ketika permintaan pertama menghasilkan kesalahan
6. Setelah waktu caching (ditentukan pada Langkah 5) telah berlalu, CloudFront coba lagi untuk mendapatkan objek yang diminta dengan meneruskan permintaan lain ke asal Anda. CloudFront terus mencoba lagi pada interval yang ditentukan oleh kesalahan caching minimum TTL.

 Note

Jika Anda juga mengonfigurasi perilaku cache untuk halaman kesalahan kustom yang sama, CloudFront gunakan perilaku cache TTL sebagai gantinya. Dalam hal ini, CloudFront akan melakukan hal berikut untuk langkah 5 dan 6:

- Setelah CloudFront mengembalikan halaman kesalahan kustom ke penampil yang membuat permintaan, CloudFront memeriksa perilaku cache TTL (misalnya, Anda mengatur TTL default ke 5 detik). CloudFront kemudian cache halaman kesalahan kustom hingga maksimum itu.
- Setelah 5 detik berlalu, CloudFront ambil halaman kesalahan kustom dari asal lagi. CloudFront akan terus mencoba lagi pada interval yang ditentukan oleh perilaku cache TTL.

Untuk informasi selengkapnya, lihat [pengaturan TTL](#) perilaku cache.

Objek yang diminta ada di cache tepi

CloudFront terus melayani objek yang saat ini berada di cache tepi ketika semua hal berikut benar:

- Penampil meminta sebuah objek.
- Objek berada dalam cache tepi namun telah kedaluwarsa.
- Asal Anda mengembalikan kode status HTTP 5xx, bukan mengembalikan kode status 304 (Not Modified) atau versi terbaru dari objek.

CloudFront melakukan hal berikut:

1. Jika asal Anda mengembalikan kode status 5xx, CloudFront melayani objek meskipun telah kedaluwarsa. Selama durasi kesalahan cache TTL minimum, CloudFront terus menanggapi permintaan penampil dengan menyajikan objek dari cache tepi.

Jika asal Anda mengembalikan kode status 4xx, CloudFront mengembalikan kode status, bukan objek yang diminta, ke penampil.

2. Setelah kesalahan caching minimum TTL telah berlalu, CloudFront coba lagi untuk mendapatkan objek yang diminta dengan meneruskan permintaan lain ke asal Anda. Perhatikan bahwa jika objek tidak sering diminta, CloudFront mungkin mengeluarkannya dari cache tepi saat server asal

Anda masih mengembalikan respons 5xx. Untuk informasi tentang berapa lama objek berada di cache CloudFront tepi, lihat [Mengelola berapa lama konten tetap dalam cache \(kedaluwarsa\)](#).

Cara CloudFront memproses kesalahan jika Anda belum mengonfigurasi halaman kesalahan kustom

Jika Anda belum mengonfigurasi halaman kesalahan kustom, CloudFront perilaku tergantung pada apakah objek yang diminta ada di cache tepi.

Topik

- [Objek yang diminta tidak ada di cache tepi](#)
- [Objek yang diminta ada di cache tepi](#)

Objek yang diminta tidak ada di cache tepi

CloudFront terus mencoba untuk mendapatkan objek yang diminta dari asal Anda ketika semua hal berikut benar:

- Penampil meminta sebuah objek.
- Objek tidak berada dalam cache tepi.
- Asal Anda mengembalikan kode status HTTP 4xx atau 5xx dan salah satu dari yang berikut adalah benar:
 - Asal Anda mengembalikan kode status HTTP 5xx, bukan mengembalikan kode status 304 (Not Modified) atau versi terbaru dari objek.
 - Kota asal Anda mengembalikan kode status HTTP 4xx yang tidak dibatasi oleh header kontrol cache dan termasuk dalam daftar kode status berikut: [Kode status HTTP 4xx dan 5xx yang di-cache CloudFront](#)
 - Asal Anda mengembalikan kode status HTTP 4xx dengan Cache-Control max-age header atau Cache-Control s-maxage header dan kode status disertakan dalam daftar kode status berikut: Kontrol [Kode status HTTP 4xx yang CloudFront di-cache berdasarkan header Cache-Control](#).

CloudFront melakukan hal berikut:

1. CloudFront mengembalikan kode status 4xx atau 5xx ke penampil, dan juga cache kode status di cache tepi yang menerima permintaan untuk maksimum berikut ini:
 - Jumlah waktu yang ditentukan oleh kesalahan caching minimum TTL (10 detik secara default)
 - Jumlah waktu yang ditentukan oleh Cache-Control max-age header atau Cache-Control s-maxage header yang dikembalikan oleh asal ketika permintaan pertama menghasilkan kesalahan
2. Selama durasi waktu caching (ditentukan di Langkah 1), CloudFront merespons permintaan penampil berikutnya untuk objek yang sama dengan kode status 4xx atau 5xx cache.
3. Setelah waktu caching (ditentukan pada Langkah 1) telah berlalu, CloudFront coba lagi untuk mendapatkan objek yang diminta dengan meneruskan permintaan lain ke asal Anda. CloudFront terus mencoba lagi pada interval yang ditentukan oleh kesalahan caching minimum TTL.

Objek yang diminta ada di cache tepi

CloudFront terus melayani objek yang saat ini berada di cache tepi ketika semua hal berikut benar:

- Penampil meminta sebuah objek.
- Objek berada dalam cache tepi namun telah kedaluwarsa. Ini berarti objek itu basi.
- Asal Anda mengembalikan kode status HTTP 5xx, bukan mengembalikan kode status 304 (Not Modified) atau versi terbaru dari objek.

CloudFront melakukan hal berikut:

1. Jika asal Anda mengembalikan kode kesalahan 5xx, CloudFront melayani objek meskipun telah kedaluwarsa. Untuk durasi kesalahan caching minimum TTL (10 detik secara default), CloudFront terus menanggapi permintaan penampil dengan menyajikan objek dari cache tepi.

Jika asal Anda mengembalikan kode status 4xx, CloudFront mengembalikan kode status, bukan objek yang diminta, ke penampil.

2. Setelah kesalahan caching minimum TTL telah berlalu, CloudFront coba lagi untuk mendapatkan objek yang diminta dengan meneruskan permintaan lain ke asal Anda. Jika objek tidak sering diminta, CloudFront mungkin mengeluarkannya dari cache tepi saat server asal Anda masih mengembalikan respons 5xx. Untuk informasi selengkapnya, lihat [Mengelola berapa lama konten tetap dalam cache \(kedaluwarsa\)](#).

 Tip

- Jika Anda mengonfigurasi `stale-if-error` atau `Stale-While-Revalidate` direktif, Anda dapat menentukan berapa lama objek basi tersedia di cache tepi. Ini memungkinkan Anda untuk terus menayangkan konten untuk pemirsa Anda bahkan ketika asal Anda tidak tersedia. Untuk informasi, lihat [Sajikan konten basi \(kedaluwarsa\)](#).
- CloudFront hanya akan melayani objek yang basi hingga nilai [TTL maksimum](#) yang ditentukan. Setelah durasi ini, objek tidak akan tersedia dari cache tepi.

Kode status HTTP 4xx dan 5xx yang di-cache CloudFront

CloudFront cache kode status HTTP 4xx dan 5xx yang dikembalikan oleh asal Anda, tergantung pada kode status tertentu yang dikembalikan dan apakah asal Anda mengembalikan header tertentu dalam respons.

CloudFront cache kode status HTTP 4xx dan 5xx berikut yang dikembalikan oleh asal Anda. Jika Anda mengonfigurasi halaman kesalahan kustom untuk kode status HTTP, CloudFront cache halaman kesalahan kustom.

 Note

Jika Anda menggunakan kebijakan cache [CachingDisabled](#) terkelola, CloudFront tidak akan menyimpan kode status atau halaman kesalahan khusus ini.

404	Tidak Ditemukan
414	Permintaan-URI Terlalu Besar
500	Kesalahan Server Internal
501	Tidak Diterapkan

502	Gateway Buruk
503	Layanan Tidak Tersedia
504	Waktu Habis Gateway

Kode status HTTP 4xx yang CloudFront di-cache berdasarkan header **Cache-Control**

CloudFront hanya menyimpan kode status HTTP 4xx berikut yang dikembalikan oleh asal Anda jika asal Anda mengembalikan header `Cache-Control max-age` atau `Cache-Control s-maxage`. Jika Anda telah mengonfigurasi halaman kesalahan kustom untuk salah satu kode status HTTP ini - dan asal Anda mengembalikan salah satu header kontrol cache - CloudFront cache halaman kesalahan kustom.

400	Permintaan Buruk
403	Dilarang
405	Metode Tidak Diizinkan
412 ¹	Prakondisi Gagal
415 ¹	Jenis Media Tidak Didukung

¹ CloudFront tidak mendukung pembuatan halaman kesalahan khusus untuk kode status HTTP ini.

Hasilkan respons kesalahan khusus

Jika objek yang Anda layani tidak CloudFront tersedia karena alasan tertentu, server web Anda biasanya mengembalikan kode status HTTP yang relevan CloudFront untuk menunjukkan ini. Misalnya, jika penampil meminta URL yang tidak valid, server web Anda mengembalikan kode status

HTTP 404 (Tidak Ditemukan) ke CloudFront, dan kemudian CloudFront mengembalikan kode status tersebut ke penampil. Alih-alih menggunakan respons kesalahan default ini, Anda dapat membuat respons khusus yang CloudFront kembali ke penampil.

Jika Anda mengonfigurasi CloudFront untuk mengembalikan halaman kesalahan kustom untuk kode status HTTP tetapi halaman kesalahan kustom tidak tersedia, CloudFront mengembalikan ke penampil kode status yang CloudFront diterima dari asal yang berisi halaman kesalahan kustom. Misalnya, asal kustom Anda mengembalikan kode status 500 dan Anda telah mengonfigurasi CloudFront untuk mendapatkan halaman kesalahan kustom untuk kode status 500 dari bucket Amazon S3. Namun, seseorang secara tidak sengaja menghapus halaman kesalahan khusus dari bucket Amazon S3 Anda. CloudFront mengembalikan kode status HTTP 404 (Tidak Ditemukan) ke penampil yang meminta objek.

Saat CloudFront mengembalikan halaman kesalahan kustom ke penampil, Anda membayar CloudFront biaya standar untuk halaman kesalahan kustom, bukan biaya untuk objek yang diminta. Untuk informasi selengkapnya tentang CloudFront tagihan, lihat [CloudFrontHarga Amazon](#).

Topik

- [Konfigurasi perilaku respons kesalahan](#)
- [Buat halaman kesalahan khusus untuk kode status HTTP tertentu](#)
- [Menyimpan objek dan halaman kesalahan kustom di lokasi yang berbeda](#)
- [Ubah kode respons yang dikembalikan oleh CloudFront](#)
- [Kontrol berapa lama CloudFront kesalahan cache](#)

Konfigurasi perilaku respons kesalahan

Anda memiliki beberapa opsi untuk mengelola bagaimana CloudFront merespons ketika ada kesalahan. Untuk mengonfigurasi respons kesalahan kustom, Anda dapat menggunakan CloudFront konsol, CloudFront API, atau CloudFormation. Terlepas dari bagaimana Anda memilih untuk memperbarui konfigurasi, pertimbangkan tip dan rekomendasi berikut ini:

- Simpan halaman kesalahan kustom Anda di lokasi yang dapat diakses CloudFront. Kami menyarankan agar Anda menyimpannya di bucket Amazon S3, dan bahwa Anda [jangan menyimpannya di tempat yang sama dengan situs web atau konten aplikasi lainnya](#). Jika Anda menyimpan halaman kesalahan kustom pada asal yang sama dengan situs web atau aplikasi Anda, dan asal mulai mengembalikan kesalahan 5xx, tidak CloudFront bisa mendapatkan

halaman kesalahan kustom karena server asal tidak tersedia. Untuk informasi selengkapnya, lihat [Menyimpan objek dan halaman kesalahan kustom di lokasi yang berbeda](#).

- Pastikan bahwa CloudFront memiliki izin untuk mendapatkan halaman kesalahan kustom Anda. Jika halaman kesalahan kustom disimpan di Amazon S3, halaman harus dapat diakses publik atau Anda harus mengonfigurasi [kontrol akses CloudFront asal](#) (OAC). Jika halaman kesalahan kustom disimpan dalam asal kustom, halaman harus dapat diakses publik.
- (Opsional) Konfigurasi asal Anda untuk menambahkan Cache-Control atau Expires bersama dengan halaman kesalahan kustom, jika Anda ingin. Anda juga dapat menggunakan pengaturan Error Caching Minimum TTL untuk mengontrol berapa lama CloudFront cache halaman kesalahan kustom. Untuk informasi selengkapnya, lihat [Kontrol berapa lama CloudFront kesalahan cache](#).

Konfigurasi respons kesalahan khusus

Untuk mengonfigurasi respons kesalahan khusus di CloudFront konsol, Anda harus memiliki CloudFront distribusi. Di konsol, pengaturan konfigurasi untuk respons kesalahan kustom hanya tersedia untuk distribusi yang ada. Untuk mempelajari cara membuat distribusi, lihat [Memulai dengan distribusi CloudFront standar](#).

Console

Untuk mengonfigurasi respons kesalahan kustom (konsol)

1. Masuk ke Konsol Manajemen AWS dan buka halaman Distribusi di CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home#distributions>.
2. Pada daftar distribusi, pilih distribusi yang akan diperbarui.
3. Pilih tab Halaman Kesalahan, lalu pilih Membuat Respons Kesalahan Kustom.
4. Masukkan nilai yang berlaku. Untuk informasi selengkapnya, lihat [Halaman kesalahan kustom dan caching kesalahan](#).
5. Setelah memasukkan nilai yang diinginkan, pilih Buat.

CloudFront API or CloudFormation

Untuk mengonfigurasi respons kesalahan kustom dengan CloudFront API atau CloudFormation, gunakan CustomErrorResponse tipe dalam distribusi. Untuk informasi selengkapnya, lihat berikut ini:

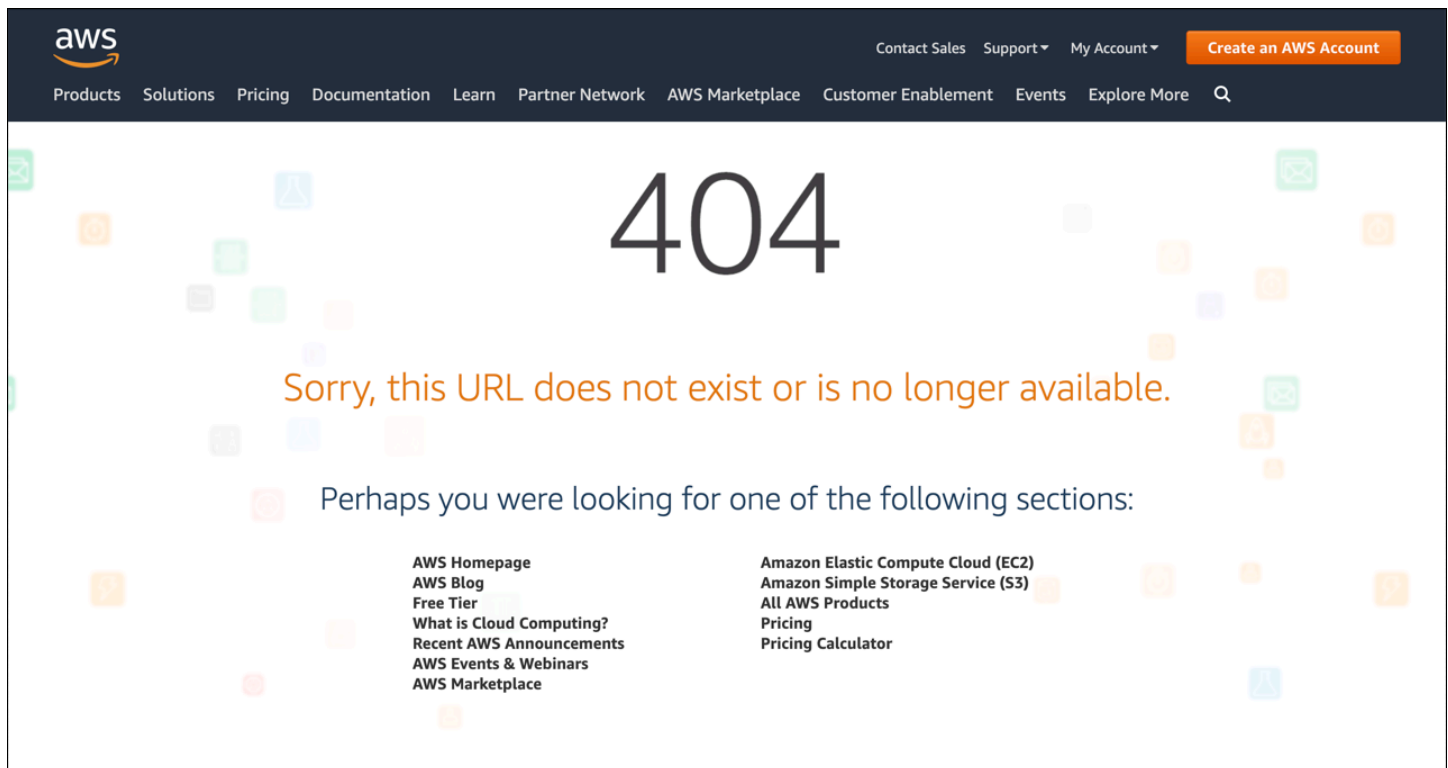
- [AWS::CloudFront::Distribution CustomErrorResponse](#) di Panduan Pengguna AWS CloudFormation
- [CustomErrorResponse](#) di Referensi CloudFront API Amazon

Buat halaman kesalahan khusus untuk kode status HTTP tertentu

Jika Anda lebih suka menampilkan pesan kesalahan kustom daripada pesan default—misalnya, halaman yang menggunakan format yang sama dengan situs web lainnya—Anda dapat CloudFront mengembalikan objek ke penampil (seperti file HTML) yang berisi pesan kesalahan kustom Anda.

Untuk menentukan file yang ingin Anda kembalikan dan kesalahan yang harus dikembalikan file, Anda memperbarui CloudFront distribusi Anda untuk menentukan nilai-nilai tersebut. Untuk informasi selengkapnya, lihat [Konfigurasi perilaku respons kesalahan](#).

Misalnya, berikut ini adalah pesan kesalahan kustom:



Anda dapat menentukan objek yang berbeda untuk setiap kode status HTTP yang didukung, atau Anda dapat menggunakan objek yang sama untuk semua kode status yang didukung. Anda dapat memilih untuk menentukan halaman kesalahan kustom untuk beberapa kode status dan tidak untuk yang lainnya.

Objek yang Anda layani CloudFront bisa tidak tersedia karena berbagai alasan. Hal ini dibagi ke dalam dua kategori luas:

- Kesalahan klien menunjukkan masalah dengan permintaan. Misalnya, objek dengan nama yang ditentukan tidak tersedia, atau pengguna tidak memiliki izin yang diperlukan untuk mendapatkan objek di bucket Amazon S3. Ketika kesalahan klien terjadi, asal mengembalikan kode status HTTP dalam rentang 4xx ke CloudFront.
- Kesalahan server menunjukkan masalah dengan server asal. Misalnya, server HTTP sibuk atau tidak tersedia. Ketika kesalahan server terjadi, server asal Anda mengembalikan kode status HTTP dalam rentang 5xx ke CloudFront, atau CloudFront tidak mendapatkan respons dari server asal Anda untuk jangka waktu tertentu dan mengasumsikan kode status 504 (Gateway Timeout).

Kode status HTTP yang CloudFront dapat mengembalikan halaman kesalahan kustom meliputi yang berikut:

- 400, 403, 404, 405, 414, 416
- 500, 501, 502, 503, 504

Catatan

- Jika CloudFront mendeteksi bahwa permintaan mungkin tidak aman, CloudFront mengembalikan kesalahan 400 (Permintaan Buruk) alih-alih halaman kesalahan khusus.
- Anda dapat membuat halaman kesalahan kustom untuk kode status HTTP 416 (Rentang yang Diminta Tidak Memuaskan), dan Anda dapat mengubah kode status HTTP yang CloudFront kembali ke pemirsa saat asal Anda mengembalikan kode status 416 ke. CloudFront Untuk informasi selengkapnya, lihat [Ubah kode respons yang dikembalikan oleh CloudFront](#). Namun, CloudFront tidak menyimpan kode status 416 respons, jadi meskipun Anda menentukan nilai untuk Error Caching Minimum TTL untuk kode status 416, CloudFront tidak menggunakannya.
- Dalam beberapa kasus, CloudFront tidak mengembalikan halaman kesalahan khusus untuk kode status HTTP 503 bahkan jika Anda mengonfigurasi CloudFront untuk melakukannya. Jika kode CloudFront kesalahan Capacity Exceeded atau Limit Exceeded, CloudFront mengembalikan kode status 503 ke penampil tanpa menggunakan halaman kesalahan kustom Anda.
- Jika Anda membuat halaman kesalahan kustom, CloudFront akan kembali `Connection: close` atau `Connection: keep-alive` untuk kode respons berikut:

- CloudFront pengembalian Connection: `close` untuk kode status: 400, 405, 414, 416, 500, 501
- CloudFront pengembalian Connection: `keep-alive` untuk kode status: 403, 404, 502, 503, 504

Untuk penjelasan rinci tentang cara CloudFront menangani respons kesalahan dari asal Anda, lihat [Bagaimana CloudFront memproses kode status HTTP 4xx dan 5xx dari asal Anda](#).

Menyimpan objek dan halaman kesalahan kustom di lokasi yang berbeda

Jika Anda ingin menyimpan objek Anda dan halaman kesalahan kustom Anda di lokasi yang berbeda, distribusi Anda harus menyertakan perilaku cache yang menyatakan hal berikut benar:

- Nilai dari Pola Jalan sesuai dengan jalur ke pesan kesalahan khusus Anda. Misalnya, Anda menyimpan halaman kesalahan kustom untuk 4xx kesalahan dalam bucket Amazon S3 di direktori bernama `/4xx-errors`. Distribusi Anda harus menyertakan perilaku cache yang pola jalurnya merutekan permintaan untuk halaman kesalahan kustom Anda ke lokasi tersebut, misalnya, `/4xx-errors/*`.
- Nilai dari Asal menentukan nilai dari ID Asal untuk asal yang berisi halaman kesalahan kustom Anda.

Untuk informasi selengkapnya, lihat [Pengaturan perilaku cache](#).

Ubah kode respons yang dikembalikan oleh CloudFront

Anda dapat mengonfigurasi CloudFront untuk mengembalikan kode status HTTP yang berbeda ke penampil daripada yang CloudFront diterima dari asal. Misalnya, jika asal Anda mengembalikan kode status 500 CloudFront, Anda mungkin CloudFront ingin mengembalikan halaman kesalahan kustom dan kode status 200 (OK) ke penampil. Ada berbagai alasan mengapa Anda mungkin ingin CloudFront mengembalikan kode status ke penampil yang berbeda dari yang asal Anda kembalikan CloudFront:

- Beberapa perangkat internet (beberapa firewall dan proksi korporat, misalnya) menangkap kode status HTTP 4xx dan 5xx dan mencegah respons kembali ke penampil. Dalam skenario ini, jika Anda mengganti 200, respon tidak dicegat.

- Jika Anda tidak peduli tentang membedakan antara kesalahan klien atau kesalahan server yang berbeda, Anda dapat menentukan 400 atau 500 sebagai nilai yang CloudFront mengembalikan semua kode status 4xx atau 5xx.
- Anda mungkin ingin mengembalikan kode status 200 (OK) dan situs web statis sehingga pelanggan Anda tidak tahu bahwa situs web Anda sedang tidak aktif.

Jika Anda mengaktifkan [log CloudFront standar](#) dan Anda mengonfigurasi CloudFront untuk mengubah kode status HTTP dalam respons, nilai `sc-status` kolom di log berisi kode status yang Anda tentukan. Namun, nilai kolom `x-edge-result-type` tidak terpengaruh. Ini berisi jenis hasil respons dari asal. Misalnya, Anda mengonfigurasi CloudFront untuk mengembalikan kode status 200 ke penampil saat asal kembali 404 (Tidak Ditemukan) ke CloudFront. Ketika asal merespon permintaan dengan kode status 404, nilai dalam kolom `sc-status` di log menjadi 200, tetapi nilai dalam kolom `x-edge-result-type` menjadi `Error`.

Anda dapat mengonfigurasi CloudFront untuk mengembalikan salah satu kode status HTTP berikut bersama dengan halaman kesalahan kustom:

- 200
- 400, 403, 404, 405, 414, 416
- 500, 501, 502, 503, 504

Kontrol berapa lama CloudFront kesalahan cache

CloudFront cache respons kesalahan untuk durasi default 10 detik. CloudFront kemudian mengirimkan permintaan berikutnya untuk objek ke asal Anda untuk melihat apakah masalah yang menyebabkan kesalahan telah diselesaikan dan objek yang diminta tersedia.

Anda dapat menentukan durasi error caching — Error Caching Minimum TTL — untuk setiap kode status 4xx dan 5xx yang di-cache. CloudFront (Untuk informasi selengkapnya, lihat [Kode status HTTP 4xx dan 5xx yang di-cache CloudFront](#).) Saat Anda menentukan durasi, perhatikan hal berikut:

- Jika Anda menentukan durasi caching kesalahan singkat, CloudFront teruskan lebih banyak permintaan ke asal Anda daripada jika Anda menentukan durasi yang lebih lama. Untuk 5xx kesalahan, ini dapat memperparah masalah yang awalnya menyebabkan asal Anda mengembalikan kesalahan.
- Saat asal Anda mengembalikan kesalahan untuk suatu objek, CloudFront merespons permintaan objek baik dengan respons kesalahan atau dengan halaman kesalahan kustom Anda hingga

durasi caching kesalahan berlalu. Jika Anda menentukan durasi caching kesalahan yang lama, CloudFront mungkin terus menanggapi permintaan dengan respons kesalahan atau halaman kesalahan kustom Anda untuk waktu yang lama setelah objek tersedia kembali.

Note

Anda dapat membuat halaman kesalahan kustom untuk kode status HTTP 416 (Rentang yang Diminta Tidak Memuaskan), dan Anda dapat mengubah kode status HTTP yang CloudFront kembali ke pemirsa saat asal Anda mengembalikan kode status 416 ke CloudFront (Untuk informasi selengkapnya, lihat [Ubah kode respons yang dikembalikan oleh CloudFront](#).) Namun, CloudFront tidak menyimpan kode status 416 respons, jadi meskipun Anda menentukan nilai untuk Error Caching Minimum TTL untuk kode status 416, CloudFront tidak menggunakannya.

Jika Anda ingin mengontrol berapa lama kesalahan CloudFront cache untuk objek individual, Anda dapat mengonfigurasi server asal Anda untuk menambahkan header yang berlaku ke respons kesalahan untuk objek tersebut.

Jika asal menambahkan `Cache-Control: max-age` atau `Cache-Control: s-maxage` direktif, atau `Expires` header, CloudFront cache respons kesalahan untuk nilai yang lebih besar di header atau Error Caching Minimum TTL.

Note

`Cache-Control: s-maxage` Nilai `Cache-Control: max-age` dan tidak boleh lebih besar dari nilai TTL Maksimum yang ditetapkan untuk perilaku cache yang halaman kesalahan diambil.

Jika asal menambahkan `Cache-Control: no-store`, `Cache-Control: no-cache`, atau `Cache-Control: private` direktif untuk kode kesalahan 404, 410, 414, atau 501, CloudFront tidak men-cache respons kesalahan. Untuk semua kode kesalahan lainnya, CloudFront abaikan **private** arahan **no-store**, **no-cache**, dan cache respons kesalahan untuk nilai Error Caching Minimum TTL.

Jika asal menambahkan **Cache-Control** arahan lain atau tidak menambahkan header, CloudFront cache respons kesalahan untuk nilai Error Caching Minimum TTL.

Jika waktu kedaluwarsa untuk kode status 4xx atau 5xx untuk objek lebih lama dari yang Anda inginkan, dan objek tersedia lagi, Anda dapat membatalkan kode galat cache dengan menggunakan URL objek yang diminta. Jika asal Anda mengembalikan respons kesalahan untuk beberapa objek, Anda perlu menggugurkan setiap objek secara terpisah. Untuk informasi lebih lanjut tentang objek yang tidak valid, lihat [Membatalkan file untuk menghapus konten](#).

Jika Anda mengaktifkan caching untuk asal bucket S3, dan Anda mengonfigurasi kesalahan caching TTL minimum 0 detik dalam CloudFront distribusi Anda, Anda masih akan melihat kesalahan caching TTL minimum 1 detik untuk kesalahan asal S3. CloudFront melakukan ini untuk melindungi asal Anda dari serangan DDoS. Ini tidak berlaku untuk jenis asal lainnya.

Menambahkan, menghapus, atau mengganti konten yang CloudFront mendistribusikan

Bagian ini menjelaskan cara memastikan CloudFront dapat mengakses konten yang ingin disajikan kepada pemirsa Anda, cara menentukan objek di situs web Anda atau di aplikasi Anda, dan cara menghapus atau mengganti konten.

Topik

- [Menambahkan dan mengakses konten yang CloudFront mendistribusikan](#)
- [Gunakan versi file untuk memperbarui atau menghapus konten dengan distribusi CloudFront](#)
- [Sesuaikan format URL untuk file di CloudFront](#)
- [Tentukan objek root default](#)
- [Membatalkan file untuk menghapus konten](#)
- [Sajikan file terkompresi](#)

Menambahkan dan mengakses konten yang CloudFront mendistribusikan

Ketika Anda CloudFront ingin mendistribusikan konten (objek), Anda menambahkan file ke salah satu asal yang Anda tentukan untuk distribusi, dan Anda mengekspos CloudFront link ke file. Lokasi CloudFront tepi tidak mengambil file baru dari asal sampai lokasi tepi menerima permintaan penampil untuk file tersebut. Untuk informasi selengkapnya, lihat [Bagaimana CloudFront memberikan konten](#).

Saat Anda menambahkan file yang CloudFront ingin Anda distribusikan, pastikan Anda menemukannya ke salah satu bucket Amazon S3 yang ditentukan dalam distribusi Anda atau, untuk asal kustom, ke direktori di domain yang ditentukan. Selain itu, konfirmasi bahwa pola jalur dalam perilaku cache yang berlaku mengirimkan permintaan ke asal yang benar.

Misalnya, anggaplah pola jalur untuk perilaku cache adalah *.html. Jika Anda tidak memiliki perilaku cache lain yang dikonfigurasi untuk meneruskan permintaan ke asal itu, hanya CloudFront akan meneruskan *.html file. Dalam skenario ini, misalnya, CloudFront tidak akan pernah mendistribusikan file .jpg yang Anda unggah ke asalnya, karena Anda belum membuat perilaku cache yang mencakup file .jpg.

CloudFront server tidak menentukan tipe MIME untuk objek yang mereka layani. Saat Anda mengunggah file ke asal Anda, kami sarankan Anda mengatur Content-Type bidang judul untuk itu.

Gunakan versi file untuk memperbarui atau menghapus konten dengan distribusi CloudFront

Untuk memperbarui konten yang CloudFront sudah ada yang disiapkan untuk didistribusikan untuk Anda, sebaiknya gunakan pengenal versi dalam nama file atau nama folder. Ini membantu memberi Anda kendali atas pengelolaan konten yang CloudFront disajikan.

Perbarui file yang ada menggunakan nama file berversi

Ketika Anda memperbarui file yang ada dalam CloudFront distribusi, kami sarankan Anda menyertakan semacam pengenal versi baik dalam nama file Anda atau dalam nama direktori Anda untuk memberi diri Anda kontrol yang lebih baik atas konten Anda. Pengidentifikasi ini dapat berupa stempel waktu tanggal, nomor urut, atau beberapa metode lain untuk membedakan dua versi dari objek yang sama.

Sebagai contoh, alih-alih menamai gambar file grafis.jpg, Anda dapat menyebutnya_1.jpg. Saat Anda ingin mulai menyajikan versi baru file ini, Anda akan menyebutkan gambar file baru_2.jpg, dan Anda akan memperbarui tautan di aplikasi web atau situs web untuk menunjukkan gambar_2.jpg. Sebagai alternatif, Anda dapat menempatkan semua grafik di direktori gambar_v1 dan, ketika Anda ingin mulai menyajikan versi baru dari satu grafik atau lebih, Anda akan membuat direktori gambar_v2 baru, dan Anda akan memperbarui tautan Anda untuk mengarah ke direktori itu. Dengan pembuatan versi, Anda tidak perlu menunggu objek kedaluwarsa sebelum CloudFront mulai menyajikan versi baru, dan Anda tidak perlu membayar untuk pembatalan objek.

Sekalipun Anda sudah membuat versi file, kami masih menyarankan agar Anda mengatur tanggal kedaluwarsa. Untuk informasi selengkapnya, lihat [Mengelola berapa lama konten tetap dalam cache \(kedaluwarsa\)](#).

Note

Menentukan nama file atau nama direktori versi tidak terkait dengan versi objek Amazon S3.

Hapus konten sehingga tidak CloudFront akan mendistribusikannya

Anda dapat menghapus file dari tempat asal Anda bahwa Anda tidak ingin lagi disertakan dalam CloudFront distribusi. Namun, CloudFront akan terus menampilkan konten pemirsa dari cache tepi hingga file kedaluwarsa.

Jika Anda ingin langsung menghapus file, Anda harus melakukan salah satu hal berikut:

- Gunakan versi file. Saat Anda menggunakan versi, versi file yang berbeda memiliki nama berbeda yang dapat Anda gunakan dalam CloudFront distribusi, untuk mengubah file mana yang dikembalikan ke pemirsa. Untuk informasi selengkapnya, lihat [Perbarui file yang ada menggunakan nama file berversi](#).
- Membatalkan file. Untuk informasi selengkapnya, lihat [Membatalkan file untuk menghapus konten](#).

Sesuaikan format URL untuk file di CloudFront

Setelah Anda mengatur asal Anda dengan objek (konten) yang ingin Anda sajikan CloudFront kepada pemirsa Anda, Anda harus menggunakan URL yang benar untuk mereferensikan objek tersebut di situs web atau kode aplikasi Anda sehingga CloudFront dapat menyajikannya.

Nama domain yang Anda gunakan di URL untuk objek di halaman web atau aplikasi web Anda dapat berupa salah satu dari berikut ini:

- Nama domain, seperti `d111111abcdef8.cloudfront.net`, yang CloudFront secara otomatis menetapkan saat Anda membuat distribusi
- Nama domain Anda sendiri, seperti `example.com`

Misalnya, Anda dapat menggunakan salah satu URL berikut untuk mengembalikan file `image.jpg`:

`https://d111111abcdef8.cloudfront.net/images/image.jpg`

`https://example.com/images/image.jpg`

Anda menggunakan format URL yang sama baik Anda menyimpan konten dalam bucket Amazon S3 atau di tempat yang dibuat khusus, seperti salah satu server web Anda sendiri.

Note

Format URL bergantung sebagian pada nilai yang Anda tentukan untuk Jalur Asal dalam distribusi Anda. Nilai ini CloudFront memberikan jalur direktori terbatas untuk objek Anda. Untuk informasi lebih lanjut tentang pengaturan jalur asal ketika Anda membuat distribusi, lihat [Jalur asal](#).

Untuk informasi selengkapnya tentang format URL, lihat bagian berikut.

Gunakan nama domain Anda sendiri (example.com)

Alih-alih menggunakan nama domain default CloudFront yang ditetapkan untuk Anda saat membuat distribusi, Anda dapat [menambahkan nama domain alternatif](#) yang lebih mudah digunakan, seperti `example.com`. Dengan menyiapkan nama domain Anda sendiri CloudFront, Anda dapat menggunakan URL seperti ini untuk objek dalam distribusi Anda:

```
https://example.com/images/image.jpg
```

Jika Anda berencana untuk menggunakan HTTPS di antara pemirsa dan CloudFront, lihat [Gunakan nama domain alternatif dan HTTPS](#).

Gunakan garis miring (/) di URL

Saat Anda menentukan URL untuk direktori dalam CloudFront distribusi Anda, pilih untuk selalu menggunakan garis miring tambahan atau tidak pernah menggunakan garis miring. Misalnya, pilih hanya salah satu format berikut untuk semua URL Anda:

```
https://d111111abcdef8.cloudfront.net/images/
```

```
https://d111111abcdef8.cloudfront.net/images
```

Mengapa itu penting?

Kedua format berfungsi untuk menautkan ke CloudFront objek, tetapi konsisten dapat membantu mencegah masalah saat Anda ingin membatalkan direktori nanti. CloudFront menyimpan URL persis seperti yang ditentukan, termasuk garis miring. Jadi jika format Anda tidak konsisten, Anda harus membatalkan URL direktori dengan dan tanpa garis miring, untuk memastikan bahwa menghapus direktori. CloudFront

Tidak nyaman untuk membuat kedua format URL menjadi invalid, dan ini dapat menyebabkan biaya tambahan. Itu karena jika Anda harus menggandakan ketidakabsahan untuk menutupi kedua jenis URL, Anda dapat melampaui jumlah maksimum tidak berlaku gratis yang diizinkan untuk bulan tersebut. Dan jika hal itu terjadi, Anda harus membayar semua ketidakabsahan, bahkan jika hanya satu format untuk setiap URL direktori yang ada CloudFront.

Buat URL yang ditandatangani untuk konten terbatas

Jika Anda memiliki konten yang ingin dibatasi aksesnya, Anda dapat membuat URL yang ditandatangani. Misalnya, jika Anda ingin mendistribusikan konten hanya kepada pengguna yang telah mengautentikasi, Anda dapat membuat URL yang valid hanya selama periode waktu tertentu atau yang tersedia hanya dari alamat IP tertentu. Untuk informasi selengkapnya, lihat [Sajikan konten pribadi dengan URL yang ditandatangani dan cookie yang ditandatangani](#).

Tentukan objek root default

Anda dapat mengonfigurasi CloudFront untuk mengembalikan objek tertentu (objek root default) ketika pengguna (penampil) meminta URL root untuk distribusi Anda alih-alih meminta objek dalam distribusi Anda. Anda dapat menggunakan objek root default untuk menghindari mengekspos konten distribusi Anda.

Daftar Isi

- [Cara menentukan objek root default](#)
- [Cara kerja objek root default](#)
- [Bagaimana cara CloudFront kerja jika Anda tidak mendefinisikan objek root](#)

Cara menentukan objek root default

Untuk menghindari mengekspos konten distribusi Anda atau mengembalikan kesalahan, tentukan objek root default untuk distribusi Anda. Anda dapat menentukan nama file yang tepat atau jalur ke file. Misalnya, jika objek root Anda adalah `index.html` file, Anda dapat menentukan nama file itu. Jika `index.html` file Anda berada di folder lain, tentukan jalur sebagai gantinya seperti `exampleFolderName/index.html`. Jika Anda menetapkan jalur ke objek root default, permintaan penampil ke URL root distribusi akan mengembalikan file yang ditentukan dari jalur itu. Anda dapat menggunakan jalur file untuk memiliki lebih banyak fleksibilitas untuk mengatur konten Anda di asal, karena objek root default Anda dapat berada di folder alih-alih di tingkat root.

Untuk menentukan objek akar default untuk distribusi Anda

1. Unggah objek akar default ke asal titik distribusi Anda.

File dapat berupa jenis apa pun yang didukung oleh CloudFront. Untuk daftar batasan pada nama file, lihat `DefaultRootObject` elemen dalam [DistributionConfig](#) Referensi Amazon CloudFront API.

Note

Jika nama file dari objek root default terlalu panjang atau berisi karakter yang tidak valid, CloudFront mengembalikan kesalahan. HTTP 400 Bad Request - `InvalidDefaultRootObject` Selain itu, CloudFront cache kode selama 10 detik (secara default) dan menulis hasilnya ke log akses.

2. Konfirmasikan bahwa izin untuk objek memberikan CloudFront setidaknya akses baca.

Untuk informasi selengkapnya tentang izin Amazon S3, lihat [Manajemen identitas dan akses di Amazon S3](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

3. Perbarui distribusi Anda untuk merujuk ke objek root default dengan menggunakan CloudFront konsol atau CloudFront API.

Untuk menentukan objek root default dengan menggunakan CloudFront konsol:

- a. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
- b. Pada daftar distribusi di panel atas, pilih distribusi yang akan diperbarui.
- c. Di panel Pengaturan, pada tab Umum, pilih Edit.
- d. Dalam kotak dialog Edit pengaturan, di bidang objek root default, masukkan nama file atau jalur ke objek root default.

Tip

String Anda tidak dapat dimulai dengan garis miring (/). Hanya tentukan nama objek atau jalur ke objek. Misalnya, gunakan `index.html` atau `exampleFolderName/index.html`. Menentukan `/exampleFolderName/index.html` atau `/index.html` dapat menyebabkan kesalahan [403 Access Denied](#).

- e. Pilih Simpan perubahan.

Untuk memperbarui konfigurasi Anda menggunakan CloudFront API, tentukan nilai untuk `DefaultRootObject` elemen dalam distribusi Anda. Untuk informasi tentang penggunaan CloudFront API untuk menentukan objek root default, lihat [UpdateDistribution](#) di Referensi Amazon CloudFront API.

4. Konfirmasikan bahwa Anda telah mengaktifkan objek akar default dengan meminta URL akar Anda. Jika browser Anda tidak menampilkan objek akar default, lakukan langkah-langkah berikut:
 - a. Konfirmasikan bahwa distribusi Anda sepenuhnya digunakan dengan melihat status distribusi Anda di CloudFront konsol.
 - b. Ulangi langkah 2 dan 3 untuk memverifikasi bahwa Anda memberikan izin yang benar dan bahwa Anda memperbarui konfigurasi distribusi Anda dengan benar untuk menentukan objek akar default.

Cara kerja objek root default

Misalkan, permintaan berikut mengarah ke objek `image.jpg`:

```
https://d111111abcdef8.cloudfront.net/image.jpg
```

Sebaliknya, permintaan berikut menunjuk URL akar dari distribusi yang sama, bukan ke objek tertentu, seperti pada contoh pertama:

```
https://d111111abcdef8.cloudfront.net/
```

Saat Anda menentukan objek akar default, permintaan pengguna akhir yang menyebut akar distribusi Anda akan mengembalikan objek akar default. Misalnya, jika Anda menetapkan file `index.html` sebagai objek akar default Anda, permintaan untuk:

```
https://d111111abcdef8.cloudfront.net/
```

Pengembalian:

```
https://d111111abcdef8.cloudfront.net/index.html
```

 Note

CloudFront tidak menentukan apakah URL dengan beberapa garis miring (`https://d111111abcdef8.cloudfront.net///`) setara dengan `https://d111111abcdef8.cloudfront.net/`. Server asal Anda membuat perbandingan itu.

Jika Anda mendefinisikan objek root default, permintaan pengguna akhir untuk subdirektori distribusi Anda tidak mengembalikan objek root default. Misalnya, misalkan `index.html` adalah objek root default Anda dan yang CloudFront menerima permintaan pengguna akhir untuk `install` direktori di bawah distribusi Anda CloudFront :

`https://d111111abcdef8.cloudfront.net/install/`

CloudFront tidak mengembalikan objek root default bahkan jika salinan `index.html` muncul di `install` direktori. Namun, jika Anda menentukan jalur ke objek root default Anda, (`install/index.html`) CloudFront akan mengembalikan objek root default untuk permintaan pengguna akhir untuk direktori `install`

Jika Anda mengonfigurasi distribusi untuk memungkinkan semua metode HTTP yang CloudFront mendukung, objek akar default berlaku untuk semua metode. Misalnya, jika objek root default Anda adalah `index.php` dan Anda menulis aplikasi Anda untuk mengirimkan `POST` permintaan ke root domain Anda (`https://example.com`), CloudFront kirimkan permintaan ke `https://example.com/index.php`.

Perilaku objek root CloudFront default berbeda dari perilaku dokumen indeks Amazon S3. Saat Anda mengonfigurasi bucket Amazon S3 sebagai situs web dan menentukan dokumen indeks, Amazon S3 mengembalikan dokumen indeks meskipun pengguna meminta subdirektori dalam bucket. (Salinan dokumen indeks harus muncul di setiap subdirektori.) Untuk informasi selengkapnya tentang mengonfigurasi bucket Amazon S3 sebagai situs web dan tentang dokumen indeks, lihat [bagian Situs Web Hosting di Amazon S3 di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon](#).

 Important

Ingat bahwa objek root default hanya berlaku untuk CloudFront distribusi Anda. Anda masih perlu mengelola keamanan untuk negara asal Anda. Misalnya, jika Anda menggunakan asal Amazon S3, Anda masih perlu mengatur ACL bucket Amazon S3 Anda secara tepat untuk memastikan tingkat akses yang Anda inginkan di bucket Anda.

Bagaimana cara CloudFront kerja jika Anda tidak mendefinisikan objek root

Jika Anda tidak menentukan objek akar default, mintalah akar pas distribusi Anda ke server asal Anda. Jika Anda menggunakan asal Amazon S3, salah satu dari hal berikut dapat dikembalikan:

- Daftar isi bucket Amazon S3 Anda — Di bawah salah satu kondisi berikut, konten asal Anda dapat dilihat oleh siapa saja yang menggunakan CloudFront untuk mengakses distribusi Anda:
 - Bucket Anda tidak dikonfigurasi dengan benar.
 - Izin Amazon S3 pada bucket yang terkait dengan distribusi Anda dan objek di bucket memberikan akses ke semua orang.
 - Pengguna akhir mengakses asal Anda menggunakan URL akar asal Anda.
- Daftar konten pribadi asal Anda — Jika Anda mengonfigurasi asal Anda sebagai distribusi pribadi (hanya Anda dan CloudFront memiliki akses), konten bucket Amazon S3 yang terkait dengan distribusi Anda dapat dilihat oleh siapa saja yang memiliki kredensial untuk mengakses distribusi Anda. CloudFront Dalam hal ini, pengguna tidak dapat mengakses konten Anda melalui URL akar asal Anda. Untuk informasi selengkapnya tentang distribusi konten pribadi, lihat [the section called “Batasi konten dengan URL yang ditandatangani dan cookie yang ditandatangani”](#).
- Error 403 Forbidden— CloudFront mengembalikan kesalahan ini jika izin di bucket Amazon S3 yang terkait dengan distribusi Anda atau izin pada objek di bucket tersebut menolak akses CloudFront ke dan ke semua orang.

Membatalkan file untuk menghapus konten

Jika Anda perlu menghapus file dari cache CloudFront tepi sebelum kedaluwarsa, Anda dapat melakukan salah satu hal berikut:

- Validasikan file dari cache edge. Lain kali penampil meminta file, CloudFront kembali ke asal untuk mengambil versi terbaru dari file tersebut.
- Gunakan pembuatan versi file untuk menyajikan versi lain dari file yang memiliki nama berbeda. Untuk informasi selengkapnya, lihat [Perbarui file yang ada menggunakan nama file berversi](#).

Topik

- [Pilih antara membatalkan file dan menggunakan nama file berversi](#)
- [Menentukan jalur mana yang akan dihapus validasinya](#)
- [Yang perlu Anda ketahui saat menghapus validasi jalur](#)

- [Membatalkan konten dengan tag cache](#)
- [Apa yang perlu Anda ketahui saat membatalkan tag](#)
- [Membatalkan file](#)
- [Membayar untuk penghapusan validasi file](#)

Pilih antara membatalkan file dan menggunakan nama file berversi

Untuk mengontrol versi file yang dilayani dari distribusi Anda, Anda dapat menginvalidasi file atau memberikan nama file versi. Jika Anda ingin sering memperbarui file, sebaiknya gunakan versi file untuk alasan berikut:

- Pemutakhiran memungkinkan Anda mengontrol file mana yang mengembalikan permintaan, bahkan ketika pengguna memiliki versi yang disimpan secara lokal atau di belakang proksi caching perusahaan. Jika Anda membuat berkas menjadi tidak valid, pengguna mungkin akan terus melihat versi lama hingga berkas tersebut kedaluwarsa dari cache tersebut.
- CloudFront log akses menyertakan nama-nama file Anda, sehingga pembuatan versi memudahkan untuk menganalisis hasil perubahan file.
- Pembuatan versi memberikan cara untuk menyajikan versi file berbeda ke pengguna berbeda.
- Pembuatan versi menyederhanakan peluncuran maju dan mundur antar revisi file.
- Pembuatan versi lebih murah. Anda masih harus membayar CloudFront untuk mentransfer versi baru file Anda ke lokasi tepi, tetapi Anda tidak perlu membayar untuk membatalkan file.

Untuk informasi lebih lanjut tentang versi file, lihat [Perbarui file yang ada menggunakan nama file berversi](#).

Menentukan jalur mana yang akan dihapus validasinya

Jika Anda ingin menginvalidasi beberapa file seperti semua file dalam direktori atau semua file yang dimulai dengan karakter yang sama, Anda dapat menyertakan * wildcard di akhir jalur invalidasi. Untuk informasi lebih lanjut tentang menggunakan * wildcard, lihat [Invalidation paths](#).

Untuk membuat file tidak valid, Anda dapat menentukan jalur untuk file individu atau jalur yang berakhir dengan * wildcard, yang mungkin berlaku untuk satu file atau untuk banyak file, seperti yang ditunjukkan dalam contoh berikut:

- `/images/image1.jpg`

- `/images/image*`
- `/images/*`

Jika Anda ingin membatalkan file yang dipilih tetapi pengguna tidak selalu mengakses setiap file yang ada di tempat Anda, Anda dapat menentukan file yang diminta oleh penampil CloudFront dan batalkan hanya file tersebut. Untuk menentukan file yang diminta pemirsa, aktifkan pencatatan CloudFront akses. Untuk informasi selengkapnya tentang log akses, lihat [Akses log \(log standar\)](#).

Yang perlu Anda ketahui saat menghapus validasi jalur

Saat Anda menentukan file yang akan dibatalkan, lihat informasi berikut:

Sensitivitas kasus

Jalur pembatalan peka huruf besar/kecil. Misalnya, `/images/image.jpg` dan `/images/Image.jpg` tentukan dua file yang berbeda.

Mengubah URI dengan menggunakan fungsi Lambda

Jika CloudFront distribusi Anda memicu fungsi Lambda pada peristiwa permintaan penampil, dan jika fungsi tersebut mengubah URI file yang diminta, sebaiknya Anda membatalkan kedua URI untuk menghapus file dari cache tepi: CloudFront

- URI dalam permintaan penampil
- URI setelah fungsi mengubahnya

Example Contoh

Misalkan fungsi Lambda Anda mengubah URI untuk file dari:

`https://d111111abcdef8.cloudfront.net/index.html`

Ke URI yang menyertakan direktori bahasa:

`https://d111111abcdef8.cloudfront.net/en/index.html`

Untuk menginvalidasi file, Anda harus menentukan jalur berikut:

- `/index.html`
- `/en/index.html`

Untuk informasi selengkapnya, lihat [Invalidation paths](#).

Objek akar default

Untuk menginvalidasi objek akar default (file), tentukan jalur dengan cara yang sama dengan Anda menentukan jalur untuk file lain. Untuk informasi selengkapnya, lihat [Cara kerja objek root default](#).

Meneruskan cookie

Jika Anda mengonfigurasi CloudFront untuk meneruskan cookie ke asal Anda, cache CloudFront edge mungkin berisi beberapa versi file. Saat Anda membatalkan file, CloudFront membatalkan setiap versi file yang di-cache terlepas dari cookie yang terkait. Anda tidak dapat menginvalidasi beberapa versi secara selektif dan tidak lainnya berdasarkan cookie terkait. Untuk informasi selengkapnya, lihat [Konten cache berdasarkan cookie](#).

Meneruskan header

Jika Anda mengonfigurasi CloudFront untuk meneruskan daftar header ke asal Anda dan ke cache berdasarkan nilai header, cache CloudFront tepi mungkin berisi beberapa versi file. Saat Anda membuat file tidak valid, CloudFront mengvalidasi setiap versi file yang disimpan, terlepas dari nilai header. Anda tidak dapat menginvalidasi beberapa versi secara selektif dan tidak lainnya berdasarkan nilai header. (Jika Anda mengonfigurasi CloudFront untuk meneruskan semua header ke asal Anda, CloudFront jangan cache file Anda.) Untuk informasi selengkapnya, lihat [Konten cache berdasarkan header permintaan](#).

Meneruskan string kueri

Jika Anda mengonfigurasi CloudFront untuk meneruskan string kueri ke asal Anda, Anda harus menyertakan string kueri saat membatalkan file, seperti yang ditunjukkan dalam contoh berikut:

- `/images/image.jpg?parameter1=a`
- `/images/image.jpg?parameter1=b`

Jika permintaan klien mencakup lima string kueri berbeda untuk file yang sama, Anda dapat membuat file tidak valid sebanyak lima kali, satu kali untuk setiap string kueri, atau Anda dapat menggunakan * wildcard di jalur tidak valid, seperti yang ditunjukkan dalam contoh berikut:

```
/images/image.jpg*
```

Untuk informasi lebih lanjut tentang penggunaan wildcard di jalur ketidakvalidan, lihat [Invalidation paths](#).

Untuk informasi lebih lanjut tentang string pencarian, lihat [Konten cache berdasarkan parameter string kueri](#).

Untuk menentukan string kueri mana yang sedang digunakan, Anda dapat mengaktifkan CloudFront log. Untuk informasi selengkapnya, lihat [Akses log \(log standar\)](#).

Maksimum yang diizinkan

Untuk informasi selengkapnya tentang jumlah maksimum pembatalan yang diizinkan, lihat [Kuotas pada invalidasi](#)

File Microsoft Smooth Streaming

Anda tidak dapat membatalkan file media dalam format Microsoft Smooth Streaming ketika Anda telah mengaktifkan Smooth Streaming untuk perilaku cache yang sesuai.

Non-ASCII atau karakter tidak aman di jalur

Jika jalur menyertakan karakter non-ASCII atau karakter tidak aman seperti yang didefinisikan dalam [RFC 1738](#), karakter tersebut. URL-encode Jangan URL-encode ada karakter lain di jalur, atau CloudFront tidak akan membatalkan versi lama file yang diperbarui.

Important

Jangan gunakan ~ karakter di jalan Anda. CloudFront tidak mendukung karakter ini untuk pembatalan, apakah itu atau tidak. URL-encoded

Jalur pembatalan

Jalurnya relatif terhadap distribusi. Misalnya, untuk membatalkan file di `https://d111111abcdef8.cloudfront.net/images/image2.jpg`, Anda akan menentukan `/images/image2.jpg`

Note

Di [CloudFrontkonsol](#), Anda dapat menghilangkan garis miring di jalur, seperti ini: `images/image2.jpg` Saat Anda menggunakan CloudFront API secara langsung, jalur pembatalan harus dimulai dengan garis miring di depan.

Anda juga dapat menginvalidasi beberapa file sekaligus dengan menggunakan * wildcard. *, yang menggantikan 0 karakter atau lebih, harus menjadi karakter terakhir di jalur ketidakvalidan.

⚠ Important

Untuk menggunakan wildcard (*) dalam penghapusan validasi, Anda harus meletakkan wildcard di akhir jalur. Tanda bintang (*) yang disisipkan di posisi lain akan diperlakukan sebagai kecocokan karakter literal, bukan penghapusan validasi wildcard.

Jika Anda menggunakan AWS Command Line Interface (AWS CLI) untuk membatalkan file dan menentukan jalur yang menyertakan * wildcard, Anda harus menggunakan tanda kutip (") di sekitar jalur seperti. `"/*"`

Panjang maksimal sebuah jalur adalah 4.000 karakter.

Example Contoh: Jalur pembatalan

- Untuk membatalkan semua file dalam direktori, semua subdirektornya, dan semua file dalam direktori dan subdirektori:

`/ directory-path /*`

- Untuk membatalkan direktori, semua subdirektornya, dan semua file dalam direktori dan subdirektori:

`/ directory-path *`

- Untuk mengvalidasi semua file yang memiliki nama yang sama tetapi memiliki ekstensi nama file yang berbeda, seperti logo.jpg, logo.png, dan logo.gif:

`/ directory-path / file-name .*`

- Untuk membatalkan semua file dalam direktori di mana nama file dimulai dengan karakter yang sama (seperti semua file untuk video dalam format HLS), terlepas dari ekstensi nama file:

`/ directory-path / initial-characters-in-file-name *`

- Saat Anda mengonfigurasi CloudFront ke cache berdasarkan parameter string kueri dan Anda ingin membatalkan setiap versi file:

`/ directory-path / file-name . file-name-extension *`

- Untuk membatalkan semua file dalam distribusi:

`/*`

Untuk informasi tentang mendevalidasi file jika Anda menggunakan fungsi Lambda untuk mengubah URI, lihat [Changing the URI Using a Lambda Function](#).

Jika alur ketidakabsahan adalah direktori dan jika Anda belum menstandarkan metode untuk menentukan direktori—dengan atau tanpa garis miring yang menyimpang (/)—kami menyarankan Anda untuk menginvalidasi direktori dengan dan tanpa garis miring, misalnya, `/images` dan `/images/`.

URL yang ditandatangani

Jika Anda menggunakan URL yang ditandatangani, batalkan file hanya dengan menyertakan bagian URL sebelum tanda tanya (?).

Membatalkan konten dengan tag cache

Amazon CloudFront mendukung pembatalan tag, yang memungkinkan Anda membatalkan objek yang di-cache berdasarkan tag semantik, bukan jalur URL. Ini memberi Anda kontrol fleksibel atas pembatalan cache tanpa memerlukan struktur URL Anda agar sesuai dengan strategi pembatalan Anda.

Cara kerja pembatalan tag

1. Konfigurasi distribusi Anda: Tambahkan `CacheTagConfig` ke distribusi Anda dengan menentukan nama header HTTP yang digunakan asal Anda untuk mengembalikan tag cache.
2. Tandai objek di asal Anda: Saat mengembalikan objek yang ingin Anda cache dengan tag, konfigurasi asal Anda untuk menyertakan header yang ditentukan dalam respons HTTP dengan nilai tag yang dipisahkan koma.

```
HTTP/1.1 200 OK
Content-Type: text/html
x-amz-meta-cache-tag: product:electronics, category:tv, brand:example
Cache-Control: max-age=3600
```

Note

Untuk asal S3, Anda dapat melampirkan tag cache ke objek S3 Anda sebagai metadata. Anda dapat menambahkan entri metadata dengan kunci pilihan Anda (contoh: `cache-tag`) dan daftar tag yang dipisahkan koma sebagai nilai (contoh:).
`product:electronics, category:tv, brand:example`

S3 memunculkan metadata objek sebagai header respons yang diawali `x-amz-meta-<Key>`, sehingga kunci metadata akan dikembalikan sebagai header. `cache-tag x-amz-meta-cache-tag` Anda dapat mengatur ke `HeaderName` dalam `CacheTagConfig x-amz-meta-cache-tag` untuk mengaktifkan pengiriman invalidasi ke tag ini.

 Note

Atau, Anda dapat melampirkan respons asal fungsi Lambda `@Edge` ke perilaku cache Anda untuk menambahkan header tag cache. Jika menggunakan Lambda `@Edge`, nama header tidak perlu mengikuti format. `x-amz-meta-<Key>`

3. Membatalkan menurut tag: Gunakan `CreateInvalidation` API dengan `#` awalan untuk membatalkan semua objek yang di-cache dengan tag yang ditentukan.

```
aws cloudfront create-invalidation \  
  --distribution-id distribution_ID \  
  --paths "#product:electronics"
```

Ini membatalkan SEMUA objek cache yang berisi tag `product:electronics`, terlepas dari jalur URL-nya.

Persyaratan format tag

Nilai header respons asal:

- Tag dipisahkan koma dalam nilai header.
- Ruang depan dan belakang di sekitar setiap tag dipangkas.
- `tag1, tag2, tag3`Keduanya `tag1, tag2, tag3` valid dan setara.

Nilai tag individu:

- Karakter terlihat ASCII (33—126)
- Tidak ada karakter kontrol, spasi, atau koma
- Tidak peka huruf besar/kecil
- Maksimum 256 karakter per tag

- Maksimal 50 tag per objek (tag tambahan diabaikan)

Mengkonfigurasi distribusi untuk pembatalan tag

Untuk mengaktifkan pembatalan tag, tambahkan `CacheTagConfig` ke konfigurasi distribusi Anda:

```
aws cloudfront create-distribution \  
  --distribution-config '{  
    "CallerReference": "my-distribution",  
    "CacheTagConfig": {  
      "HeaderName": "x-amz-meta-cache-tag"  
    },  
    "DefaultCacheBehavior": {  
      "TargetOriginId": "myOrigin",  
      "ViewerProtocolPolicy": "redirect-to-https",  
      "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e58f6"  
    },  
    "Origins": { ... },  
    "Enabled": true,  
    "Comment": "Distribution with tag support"  
  }'
```

Membuat pembatalan tag

Gunakan `#` awalan dalam `--paths` parameter untuk menentukan tag:

```
# Invalidate all objects tagged with "user1"  
aws cloudfront create-invalidation \  
  --distribution-id distribution_ID \  
  --paths "#user1"  
  
# Invalidate objects matching any of multiple tags (OR logic)  
aws cloudfront create-invalidation \  
  --distribution-id distribution_ID \  
  --paths "#user1" "#product-category:electronics"  
  
# Mix path and tag invalidations in one batch  
aws cloudfront create-invalidation \  
  --distribution-id distribution_ID \  
  --paths "/index.html" "#user1" "/images/*" "#product-category:electronics"
```


Memeriksa status pembatalan

Gunakan `GetInvalidation` untuk memeriksa status. Respons mencakup item path dan tag:

```
aws cloudfront get-invalidation \  
  --distribution-id distribution_ID \  
  --id invalidation_ID
```

Pertimbangan penting

- Diperlukan keikutsertaan: Pembatalan tag hanya berfungsi pada distribusi yang telah dikonfigurasi. `CacheTagConfig` Distribusi tanpa konfigurasi ini mengabaikan header tag cache dari asal.
- Batas pemrosesan tag: CloudFront memproses hingga 50 tag per objek yang di-cache. Jika respons asal berisi lebih dari 50 tag, tag tambahan di luar batas tidak disimpan.
- Mengubah nama header: Pembatalan tag dicari terhadap konfigurasi saat ini. `CacheTagConfig` Jika Anda mengubah `HeaderName` dalam `CacheTagConfig`, pembatalan yang dikeluarkan untuk objek yang di-cache dengan tag di bawah nama header lama tidak akan lagi dievaluasi. Jika Anda perlu mengubah nama header, mulailah mengembalikan header tag cache baru dan lama dengan objek Anda, lalu keluarkan pembatalan jalur (misalnya, `/*`) atau batalkan tag yang ada sebelum mengubah nama header untuk menghindari penyajian konten basi. Setelah selesai, Anda dapat berhenti mengirim header tag cache lama dengan objek Anda.
- Menghapus `CacheTagConfig`: Saat Anda menghapus `CacheTagConfig` dari distribusi, CloudFront berhenti mengekstrak tag dari respons asal. Objek cache yang ada dengan tag disajikan secara normal sampai kedaluwarsa atau tidak valid oleh jalur.
- Kompatibilitas mundur: Pembatalan jalur dan wildcard yang ada terus berfungsi tidak berubah. Pembatalan tag bersifat aditif — Anda dapat menggunakan kedua metode pada distribusi yang sama.
- Penyewa distribusi: Pembatalan tag juga didukung untuk penyewa distribusi melalui API. `CreateInvalidationForDistributionTenant`

Contoh kasus penggunaan

Contoh 1: Katalog produk e-Commerce

Situs e-commerce menandai halaman produk yang di-cache dengan produk, kategori, dan merek:

```
x-amz-meta-cache-tag: category:electronics, brand:acme, product:12345
```

Saat ACME memperbarui branding mereka, batalkan semua produk ACME sekaligus:

```
aws cloudfront create-invalidation \  
  --distribution-id distribution_ID \  
  --paths "#brand:acme"
```

Contoh 2: Platform konten buatan pengguna

Platform menandai konten yang di-cache dengan ID pengguna pemilik:

```
x-amz-meta-cache-tag: user:12345, content-type:image
```

Saat pengguna menutup akun mereka, batalkan semua kontennya setelah Anda menghapusnya dari asal:

```
aws cloudfront create-invalidation \  
  --distribution-id distribution_ID \  
  --paths "#user:12345"
```

Contoh 3: Pembuatan versi konten

Sebuah konten tag CMS dengan pengidentifikasi versi:

```
x-amz-meta-cache-tag: version:v2, template:homepage
```

Saat menerapkan versi baru, batalkan semua konten v2:

```
aws cloudfront create-invalidation \  
  --distribution-id distribution_ID \  
  --paths "#version:v2"
```

Apa yang perlu Anda ketahui saat membatalkan tag

Saat Anda menentukan tag untuk dibatalkan, lihat informasi berikut:

Diperlukan keikutsertaan

Pembatalan tag hanya berfungsi pada distribusi yang telah dikonfigurasi. CacheTagConfig Distribusi tanpa konfigurasi ini mengabaikan header tag cache dari asal. Untuk informasi selengkapnya, lihat [Membatalkan konten dengan tag cache](#).

Awalan pembatalan tag

Item pembatalan tag harus dimulai dengan karakter. # Misalnya, untuk membatalkan semua objek cache yang ditandai dengan `product:electronics`, tentukan `#product:electronics` dalam daftar `Paths.Items`

Sensitivitas kasus

Nilai tag tidak peka huruf besar/kecil. Misalnya, `#Product:Electronics` dan `#product:electronics` lihat tag yang sama.

Format tag

Setiap nilai tag harus berisi hanya karakter ASCII yang terlihat (33-126), tidak termasuk koma. Panjang maksimum tag adalah 256 karakter.

Tidak ada dukungan wildcard

Tidak seperti pembatalan jalur, item pembatalan tag tidak mendukung wildcard. * Setiap item # - prefixed cocok dengan nilai tag yang tepat.

Tag maksimum per objek cache

CloudFront memproses hingga 50 tag per objek yang di-cache. Jika respons asal berisi lebih dari 50 tag di header yang dikonfigurasi, tag tambahan di luar batas tidak disimpan dan tidak dapat digunakan untuk pembatalan.

Meneruskan cookie, header, dan string kueri

Saat Anda membatalkan berdasarkan tag, CloudFront membatalkan setiap varian cache dari setiap objek yang membawa tag yang ditentukan, terlepas dari cookie, header, atau parameter string kueri terkait. Anda tidak dapat secara selektif membatalkan beberapa varian dan bukan yang lain.

Pembatalan jalur dan tag campuran

Anda dapat menyertakan kedua item jalur (misalnya, `/images/`) dan item tag (misalnya, `#brand:acme`) dalam kumpulan pembatalan yang sama dalam permintaan. `CreateInvalidation` Setiap item dihitung sebagai satu jalur pembatalan menuju batas Anda.

Maksimum yang diizinkan

Item pembatalan tag dihitung menuju batas pembatalan bersamaan yang sama dengan item jalur. Setiap item tag dihitung sebagai satu jalur pembatalan. Untuk informasi selengkapnya, lihat [Kuotas pada invalidasi](#).

Mengubah nama header

Jika Anda mengubah `HeaderName` in `CacheTagConfig`, pembatalan yang dikeluarkan terhadap tag yang diekstrak di bawah nama header lama tidak akan dievaluasi lagi. Sebelum mengubah nama header:

1. Mulai kembalikan header tag cache lama dan baru dari asal Anda.
2. Mengeluarkan pembatalan jalur (misalnya, `/`*) atau membatalkan tag yang ada.
3. Perbarui `CacheTagConfig` dengan yang baru `HeaderName`.
4. Berhenti mengembalikan header lama dari asal Anda.

Penyewa distribusi

Pembatalan tag juga didukung untuk penyewa distribusi melalui API.

`CreateInvalidationForDistributionTenant`

Membatalkan file

Anda dapat menggunakan CloudFront konsol untuk membuat dan menjalankan pembatalan, menampilkan daftar pembatalan yang Anda kirimkan sebelumnya, dan menampilkan informasi terperinci tentang pembatalan individual. Anda juga dapat menyalin ketidakvalidan yang sudah ada, mengedit daftar jalur file, dan menjalankan edit ketidakvalidan. Anda tidak dapat menghapus kesalahan dari daftar.

Daftar Isi

- [Membatalkan file](#)
- [Salin, edit, dan jalankan kembali pembatalan yang ada](#)
- [Batalkan pembatalan](#)
- [Daftar pembatalan](#)
- [Menampilkan informasi tentang pembatalan](#)

Membatalkan file

Untuk membatalkan file menggunakan CloudFront konsol, lakukan hal berikut.

Console

Untuk membatalkan file (konsol)

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Pilih distribusi yang ingin Anda batalkan file.
3. Pilih Tidak berlakunya tab.
4. Pilih Buat pembatalan.
5. Untuk file yang ingin Anda batalkan, masukkan satu jalur ketidakvalidan per baris. Untuk informasi tentang menentukan jalur ketidakvalidan, lihat [Yang perlu Anda ketahui saat menghapus validasi jalur](#).

Important

Tentukan jalur file dengan cermat. Anda tidak dapat menginvalidasi permintaan ketidakabsahan setelah Anda memulainya.

6. Pilih Buat pembatalan.

CloudFront API

Untuk mempelajari tentang membatalkan objek dan menampilkan informasi tentang pembatalan, lihat topik berikut di Referensi Amazon API: CloudFront

- [CreateInvalidation](#)
- [ListInvalidations](#)
- [GetInvalidation](#)

Note

Jika Anda menggunakan AWS Command Line Interface (AWS CLI) untuk membatalkan file dan menentukan jalur yang menyertakan * wildcard, Anda harus menggunakan tanda kutip (") di sekitar jalur, seperti contoh berikut:

```
aws cloudfront create-invalidation --distribution-id distribution_ID --paths  
"/"
```

Salin, edit, dan jalankan kembali pembatalan yang ada

Anda dapat menyalin ketidakvalidan yang Anda buat sebelumnya, memperbarui daftar jalur ketidakvalidan, dan menjalankan ketidakabsahan yang diperbarui. Anda tidak dapat menyalin pembatalan yang ada, memperbarui jalur pembatalan, dan kemudian menyimpan pembatalan yang diperbarui tanpa menjalankannya.

Important

Jika Anda menyalin pembatalan yang masih berlangsung, perbarui daftar jalur pembatalan, lalu jalankan pembatalan yang diperbarui, tidak CloudFront akan menghentikan atau menghapus pembatalan yang Anda salin. Jika ada jalur pembatalan muncul di aslinya dan dalam salinan, CloudFront akan mencoba untuk membatalkan file dua kali, dan kedua pembatalan akan dihitung terhadap jumlah maksimum pembatalan gratis untuk bulan tersebut. Jika Anda sudah mencapai jumlah maksimum pembatalan gratis, Anda akan dikenakan biaya untuk kedua pembatalan setiap file. Untuk informasi selengkapnya, lihat [Kuotas pada invalidasi](#).

Untuk menyalin, mengedit, dan menjalankan ulang ketidakabsahan yang ada

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Pilih distribusi yang berisi ketidakvalidan yang ingin Anda salin.
3. Pilih Tidak berlakunya tab.
4. Pilih invalidation (tidak valid) yang ingin Anda salin.

Jika Anda tidak yakin pembatalan mana yang ingin Anda salin, Anda dapat memilih pembatalan dan memilih Lihat detail untuk menampilkan informasi terperinci tentang pembatalan tersebut.

5. Pilih Salin ke yang baru.
6. Perbarui daftar path ketidakvalidan jika berlaku.
7. Pilih Buat pembatalan.

Batalan pembatalan

Saat Anda mengirimkan permintaan pembatalan ke CloudFront, CloudFront teruskan permintaan ke semua lokasi tepi dalam beberapa detik, dan setiap lokasi tepi mulai memproses pembatalan segera. Oleh karena itu, Anda tidak dapat menginvalidasi ketidakvalidan setelah Anda mengirimkannya.

Daftar pembatalan

Anda dapat menampilkan daftar 100 pembatalan terakhir yang telah Anda buat dan jalankan untuk distribusi menggunakan konsol. CloudFront Jika Anda ingin mendapatkan daftar lebih dari 100 pembatalan, gunakan operasi API. `ListInvalidations` Untuk informasi selengkapnya, lihat [ListInvalidations](#) di Referensi Amazon CloudFront API.

Untuk mencantumkan ketidakabsahan

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Pilih distribusi yang ingin Anda tampilkan daftar ketidakvalidan.
3. Pilih Tidak berlakunya tab.

Note

Anda tidak dapat menghapus kesalahan dari daftar.

Menampilkan informasi tentang pembatalan

Anda dapat menampilkan informasi terperinci tentang ketidakabsahan, termasuk ID distribusi, ID ketidakvalidan, status ketidakabsahan, tanggal dan waktu di mana ketidakvalidan dibuat, dan daftar lengkap jalur ketidakvalidan.

Untuk menampilkan informasi tentang ketidakabsahan

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Pilih distribusi yang memuat ketidakvalidan yang ingin Anda tampilkan informasi terperinci.
3. Pilih Tidak berlakunya tab.

4. Pilih ID pembatalan yang berlaku atau pilih ID pembatalan lalu pilih Lihat detail.

Membayar untuk penghapusan validasi file

1.000 jalur ketidakvalidan pertama yang Anda kirim per bulan adalah gratis; Anda membayar setiap alur ketidakabsahan lebih dari 1.000 dalam satu bulan. Alur ketidakvalidan bisa untuk satu file (seperti `/images/logo.jpg`) atau untuk beberapa file (seperti `/images/*`). Jalur yang menyertakan `*` wildcard dihitung sebagai satu jalur meskipun CloudFront menyebabkan ribuan file tidak valid.

Maksimal 1.000 jalur pembatalan gratis per bulan berlaku untuk jumlah total jalur pembatalan di semua distribusi yang Anda buat dengan satu. Akun AWS Misalnya, jika Anda menggunakan Akun AWS `john@example.com` untuk membuat tiga distribusi, dan Anda mengirimkan 600 jalur pembatalan untuk setiap distribusi dalam bulan tertentu (untuk total 1.800 jalur pembatalan), AWS akan menagih Anda untuk perbedaan antara total jalur pembatalan dan batas bebas 1000. Dalam contoh ini, AWS akan menagih Anda untuk 800 jalur pembatalan di bulan itu.

Biaya untuk mengirim alur ketidakvalidan sama terlepas dari jumlah file yang Anda batalkan: file tunggal (`/images/logo.jpg`) atau semua file yang terkait dengan distribusi (`/`). Karena Anda dikenakan biaya per jalur dalam permintaan pembatalan, meskipun Anda menggabungkan beberapa jalur ke dalam satu permintaan, setiap jalur masih dihitung satu per satu untuk tujuan penagihan.

Untuk informasi selengkapnya tentang harga pembatalan, lihat Harga [Amazon CloudFront](#). Untuk informasi lebih lanjut tentang jalur ketidakvalidan, lihat [Invalidation paths](#).

Jalur pembatalan tag dihitung terhadap 1.000 jalur pembatalan gratis yang sama per bulan. Misalnya, jika Anda mengirimkan 500 pembatalan jalur dan 600 pembatalan tag dalam sebulan (dengan total 1.100 jalur pembatalan), Anda akan dikenakan biaya untuk 100 jalur pembatalan. Item pembatalan tag tunggal (misalnya, `#product:electronics`) dihitung sebagai satu jalur terlepas dari berapa banyak objek yang di-cache yang dibatalkan — sama dengan pembatalan jalur wildcard (misalnya,) dihitung sebagai satu jalur. `/*`

Untuk informasi selengkapnya tentang pembatalan tag, lihat. [Membatalkan konten dengan tag cache](#)

Sajikan file terkompresi

Ketika objek yang diminta dikompresi, unduhan bisa lebih cepat karena objek lebih kecil—dalam beberapa kasus, kurang dari seperempat ukuran aslinya. Unduhan yang lebih cepat dapat

menghasilkan rendering halaman web yang lebih cepat untuk pemirsa Anda, terutama untuk JavaScript dan file CSS. Selain itu, biaya transfer CloudFront data didasarkan pada jumlah total data yang disajikan. Melayani benda terkompresi bisa lebih murah daripada menyajikannya tanpa kompresi.

Topik

- [Konfigurasi CloudFront untuk mengompres objek](#)
- [Bagaimana CloudFront kompresi bekerja](#)
- [Kondisi untuk kompresi](#)
- [Jenis file yang CloudFront mengompres](#)
- [ETag konversi header](#)

Konfigurasi CloudFront untuk mengompres objek

Untuk mengkonfigurasi CloudFront untuk mengompres objek, perbarui perilaku cache yang ingin Anda layani objek terkompresi.

Untuk mengkonfigurasi CloudFront untuk mengompres objek (konsol)

1. Masuk ke [konsol CloudFront](#) tersebut.
2. Pilih distribusi Anda dan kemudian pilih Perilaku yang akan diedit.
3. Untuk pengaturan objek Kompres secara otomatis, pilih Ya.
4. Gunakan [kebijakan cache](#) untuk menentukan pengaturan caching, dan aktifkan format kompresi Gzip dan Brotli.

Catatan

- Anda harus menggunakan [kebijakan cache](#) untuk menggunakan kompresi Brotli. Brotli tidak mendukung pengaturan cache lama.
- Untuk mengaktifkan kompresi dengan menggunakan [CloudFormation](#) atau [CloudFront API](#), atur `EnableAcceptEncodingBrotli` parameter `CompressEnableAcceptEncodingGzip`, ke `true`.

Untuk memahami bagaimana CloudFront mengompres objek, lihat bagian berikut.

Bagaimana CloudFront kompresi bekerja

1. Penampil meminta sebuah objek. Penampil menyertakan header `Accept-Encoding` HTTP dalam permintaan, dan nilai header termasuk `gzip`, `br`, atau keduanya. Ini menunjukkan bahwa penampil mendukung objek terkompresi. Saat penampil mendukung Gzip dan Brotli, CloudFront gunakan Brotli.

Note

Browser web Chrome dan Firefox mendukung kompresi Brotli hanya ketika permintaan dikirim menggunakan HTTPS. Mereka tidak mendukung Brotli dengan permintaan HTTP.

2. Di lokasi tepi, CloudFront periksa cache untuk salinan terkompresi dari objek yang diminta.
3. Tergantung apakah objek terkompresi ada dalam cache atau CloudFront tidak, lakukan salah satu hal berikut:
 - Jika objek terkompresi sudah ada di cache, CloudFront kirim objek ke penampil dan lewati langkah-langkah yang tersisa.
 - Jika objek terkompresi tidak ada dalam cache, CloudFront teruskan permintaan ke asal.

Note

Jika salinan objek yang tidak terkompresi sudah ada di cache, CloudFront mungkin mengirimkannya ke penampil tanpa meneruskan permintaan ke asal. Misalnya, ini bisa terjadi ketika CloudFront [sebelumnya dilewati kompresi](#). Ketika ini terjadi, CloudFront cache objek yang tidak dikompresi dan terus melayani sampai objek kedaluwarsa, diusir, atau tidak valid.

4. Jika asal mengembalikan objek terkompresi, (seperti yang ditunjukkan oleh `Content-Encoding` header dalam respons HTTP), CloudFront mengirimkan objek terkompresi ke penampil, menambahkannya ke cache, dan melewati langkah-langkah yang tersisa. CloudFront tidak memampatkan objek lagi.
5. Jika asal mengembalikan objek yang tidak dikompresi ke CloudFront tanpa `Content-Encoding` header dalam respons HTTP, CloudFront kemudian menentukan apakah objek dapat dikompresi. Untuk informasi selengkapnya, lihat [Kondisi untuk kompresi](#).
6. Jika objek dapat dikompresi, CloudFront kompres, kirimkan ke penampil, dan kemudian menambahkannya ke cache.

7. Jika ada permintaan penampil berikutnya untuk objek yang sama, CloudFront mengembalikan versi cache pertama. Misalnya, jika penampil meminta objek cache tertentu yang menggunakan kompresi Gzip, dan penampil menerima format Gzip, permintaan berikutnya ke objek yang sama akan selalu mengembalikan versi Gzip, bahkan jika penampil menerima Brotli dan Gzip.

Beberapa custom origin juga dapat mengompres objek. Asal Anda mungkin dapat mengompres objek yang CloudFront tidak dikompres. Untuk informasi selengkapnya, lihat [Jenis file yang CloudFront mengompres](#).

Kondisi untuk kompresi

Daftar berikut memberikan informasi lebih lanjut tentang skenario di mana CloudFront tidak memampatkan objek.

Permintaan menggunakan HTTP 1.0

Jika permintaan untuk CloudFront menggunakan HTTP 1.0, CloudFront menghapus Accept-Encoding header dan tidak memampatkan objek dalam respons.

Accept-Encoding permintaan header

Jika Accept-Encoding header hilang dari permintaan penampil, atau jika tidak berisi gzip atau br sebagai nilai, CloudFront tidak akan memampatkan objek dalam respons. Jika Accept-Encoding header menyertakan nilai tambahan seperti deflate, CloudFront hapus sebelum meneruskan permintaan ke asal.

Ketika CloudFront [dikonfigurasi untuk mengompres objek](#), itu termasuk Accept-Encoding header di kunci cache dan permintaan asal secara otomatis.

Konten sudah di-cache saat Anda mengonfigurasi CloudFront untuk mengompres objek

CloudFront memampatkan objek ketika mendapatkannya dari asal. Saat Anda mengonfigurasi CloudFront untuk mengompres objek, CloudFront tidak mengompres objek yang sudah di-cache di lokasi tepi. Selain itu, ketika objek yang di-cache kedaluwarsa di lokasi tepi dan CloudFront meneruskan permintaan lain untuk objek ke asal Anda, CloudFront tidak akan memampatkan objek saat asal Anda mengembalikan kode status HTTP 304. Ini berarti bahwa lokasi tepi sudah memiliki versi terbaru dari objek. Jika Anda ingin CloudFront mengompres objek yang sudah di-cache di lokasi tepi, Anda perlu membatalkan objek tersebut. Untuk informasi selengkapnya, lihat [Membatalkan file untuk menghapus konten](#).

Origin sudah dikonfigurasi untuk mengompres objek

Jika Anda mengonfigurasi CloudFront untuk mengompres objek dan asal juga mengompres objek, asal harus menyertakan Content-Encoding header. Header ini menunjukkan CloudFront bahwa objek sudah dikompresi. Ketika respons dari asal menyertakan Content-Encoding header, CloudFront tidak memampatkan objek, terlepas dari nilai header. CloudFront mengirimkan respons ke penampil dan menyimpan objek di lokasi tepi.

Jenis file yang CloudFront mengompres

Untuk daftar lengkap, lihat [Jenis file yang CloudFront mengompres](#).

Ukuran benda yang CloudFront memampatkan

CloudFront memampatkan objek yang berukuran antara 1.000 byte dan 10.000.000 byte.

Header Content-Length

Asal harus menyertakan Content-Length header dalam respons, yang CloudFront digunakan untuk menentukan apakah ukuran objek berada dalam kisaran yang CloudFront memampatkan. Jika Content-Length header hilang, berisi nilai yang tidak valid, atau berisi nilai di luar rentang ukuran yang CloudFront dikompres, CloudFront tidak akan memampatkan objek. Untuk informasi selengkapnya tentang cara CloudFront memproses objek besar yang dapat melebihi rentang ukuran, lihat [Bagaimana CloudFront memproses permintaan sebagian untuk suatu objek \(rentang GETs\)](#).

Kode status HTTP dari respons

CloudFront mengompres objek hanya ketika kode status HTTP dari respon adalah 200, 403, atau 404.

Respon tidak memiliki tubuh

Ketika respons HTTP dari asal tidak memiliki badan, tidak ada yang bisa dikompres. CloudFront

Header ETag

CloudFront terkadang memodifikasi ETag header dalam respons HTTP ketika memampatkan objek. Untuk informasi selengkapnya, lihat [the section called “ETag konversi header”](#).

CloudFront melompati kompresi

CloudFront mengompres objek dengan upaya terbaik. Dalam kasus yang jarang terjadi, CloudFront lompat mengompresi objek saat CloudFront mengalami beban lalu lintas yang tinggi. CloudFront membuat keputusan ini berdasarkan berbagai faktor, termasuk kapasitas host. Jika

CloudFront melewatkan kompresi untuk suatu objek, itu akan menyimpan objek yang tidak dikompresi dan terus menyajikannya kepada pemirsa sampai objek kedaluwarsa, diusir, atau tidak valid.

Jenis file yang CloudFront mengompres

Jika Anda mengonfigurasi CloudFront untuk mengompres objek, CloudFront hanya kompres objek yang memiliki salah satu nilai berikut di header Content-Type respons:

- application/dash+xml
- application/eot
- application/font
- application/font-sfnt
- application/javascript
- application/json
- application/opentype
- application/otf
- application/pdf
- application/pkcs7-mime
- application/protobuf
- application/rss+xml
- application/truetype
- application/ttf
- application/vnd.apple.mpegurl
- application/vnd.mapbox-vector-tile
- application/vnd.ms-fontobject
- application/wasm
- application/xhtml+xml
- application/xml
- application/x-font-opentype
- application/x-font-truetype
- application/x-font-ttf

- application/x-httpd-cgi
- application/x-javascript
- application/x-mpegurl
- application/x-opentype
- application/x-otf
- application/x-perl
- application/x-ttf
- font/eot
- font/opentype
- font/otf
- font/ttf
- image/svg+xml
- text/css
- text/csv
- text/html
- text/javascript
- text/js
- text/plain
- text/richtext
- text/tab-separated-values
- text/xml
- text/x-component
- text/x-java-source
- text/x-script
- vnd.apple.mpegurl

ETagkonversi header

Ketika objek yang tidak dikompresi dari asal menyertakan header ETag HTTP yang valid dan kuat, dan CloudFront mengompres objek, CloudFront juga mengubah nilai ETag header yang kuat menjadi lemahETag, dan mengembalikan nilai lemah ke penampilETag. Penampil dapat menyimpan yang

lemah ETag nilai dan penggunaannya untuk mengirim permintaan bersyarat dengan If-None-Match Header HTTP. Hal ini memungkinkan pemirsa CloudFront,, dan asal untuk memperlakukan versi objek yang dikompresi dan tidak terkompresi sebagai setara secara semantik, yang mengurangi transfer data yang tidak perlu.

Nilai ETag header yang valid dan kuat dimulai dan diakhiri dengan karakter kutipan ganda ("). Untuk mengonversi ETag yang bernilai rendah, CloudFront menambahkan karakter W/ pada awal ETag yang sesuai.

Ketika objek dari asal menyertakan nilai ETag header yang lemah (nilai yang dimulai dengan karakterW/), CloudFront tidak mengubah nilai ini, dan mengembalikannya ke penampil seperti yang diterima dari asal.

Ketika objek dari asal menyertakan nilai ETag header yang tidak valid (nilai tidak dimulai dengan " atau denganW/), CloudFront menghapus ETag header dan mengembalikan objek ke penampil tanpa header ETag respon.

Untuk informasi lebih lanjut, lihat halaman berikut di dokumen web MDN:

- [Arahan](#) (ETag Header HTTP)
- [Validasi lemah](#) (Permintaan bersyarat HTTP)
- [If-None-MatchHeader HTTP](#)

Gunakan AWS WAF perlindungan

Anda dapat menggunakan [AWS WAF](#) untuk melindungi CloudFront distribusi dan server asal Anda. AWS WAF adalah firewall aplikasi web yang membantu mengamankan aplikasi web Anda dan APIs dengan memblokir permintaan sebelum mereka mencapai server Anda. Untuk informasi selengkapnya, lihat [Mempercepat dan melindungi situs web Anda menggunakan CloudFront dan AWS WAF dan Pedoman untuk Menerapkan AWS WAF](#).

Untuk mengaktifkan AWS WAF perlindungan, Anda dapat:

- Gunakan perlindungan satu klik di CloudFront konsol. Perlindungan sekali klik membuat daftar kontrol akses AWS WAF web (web ACL), mengonfigurasi aturan untuk melindungi server Anda dari ancaman web umum, dan melampirkan ACL web ke distribusi untuk Anda. CloudFront Topik di bagian ini mengasumsikan penggunaan perlindungan satu klik.
- Gunakan ACL web yang telah dikonfigurasi sebelumnya (daftar kontrol akses) yang Anda buat di AWS WAF konsol, atau dengan menggunakan AWS WAF APIs Untuk informasi selengkapnya, lihat [Daftar kontrol akses Web \(ACLs\)](#) di Panduan AWS WAF Pengembang dan [AssociateWebACL di Referensi AWS WAF API](#)

Anda dapat mengaktifkan AWS WAF ketika Anda:

- Buat distribusi
- Gunakan dasbor Keamanan untuk mengedit pengaturan keamanan distribusi yang ada

Saat Anda menggunakan perlindungan sekali klik, CloudFront terapkan serangkaian perlindungan yang AWS disarankan yang:

- Blokir alamat IP dari potensi ancaman berdasarkan intelijen ancaman internal Amazon.
- Lindungi dari kerentanan paling umum yang ditemukan dalam aplikasi web seperti yang dijelaskan dalam [OWASP Top 10](#).
- Pertahankan terhadap aktor jahat yang menemukan kerentanan aplikasi.

Important

Anda harus mengaktifkan AWS WAF jika Anda ingin melihat metrik keamanan di dasbor CloudFront Keamanan. Tanpa AWS WAF, diaktifkan, Anda hanya dapat menggunakan

dasbor Keamanan untuk mengaktifkan AWS WAF atau mengonfigurasi batasan CloudFront geografis. Untuk informasi selengkapnya tentang dasbor, lihat [Kelola perlindungan AWS WAF keamanan di dasbor CloudFront keamanan](#), nanti di bagian ini.

Topik

- [Aktifkan AWS WAF untuk distribusi](#)
- [Kelola perlindungan AWS WAF keamanan di dasbor CloudFront keamanan](#)
- [Mengatur pembatasan tarif](#)
- [Nonaktifkan perlindungan AWS WAF keamanan](#)

Aktifkan AWS WAF untuk distribusi

Anda dapat mengaktifkan AWS WAF saat membuat distribusi, atau Anda dapat mengaktifkan perlindungan keamanan untuk daftar kontrol akses (ACL) yang ada.

Jika Anda mengaktifkan AWS WAF CloudFront distribusi Anda, Anda juga dapat mengaktifkan kontrol bot dan mengonfigurasi perlindungan keamanan berdasarkan kategori bot.

Topik

- [AWS WAF Aktifkan distribusi baru](#)
- [Gunakan ACL web yang ada](#)
- [Aktifkan kontrol bot](#)
- [Konfigurasi perlindungan berdasarkan kategori bot](#)

AWS WAF Aktifkan distribusi baru

Prosedur berikut menunjukkan cara mengaktifkan AWS WAF saat Anda membuat CloudFront distribusi baru.

AWS WAF Untuk mengaktifkan distribusi baru

1. Buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Distribusi, lalu pilih Buat distribusi.
3. Sesuai kebutuhan, ikuti langkah-langkahnya [Buat distribusi](#).

4. Di bagian Web Application Firewall, pilih Edit, lalu pilih Aktifkan perlindungan keamanan.
5. Lengkapi bidang-bidang berikut:
 - Gunakan mode monitor — Anda mengaktifkan mode monitor saat Anda ingin mengumpulkan data terlebih dahulu untuk menguji bagaimana perlindungan akan bekerja. Saat Anda mengaktifkan mode monitor, permintaan tidak diblokir jika proteksi aktif. Sebagai gantinya, mode monitor mengumpulkan data tentang permintaan yang akan diblokir jika proteksi aktif. Saat Anda siap untuk mulai memblokir, Anda dapat mengaktifkan pemblokiran di halaman Keamanan.
 - Perlindungan tambahan — Pilih opsi apa pun yang ingin Anda aktifkan. Jika Anda mengaktifkan pembatasan tarif, lihat [the section called “Mengatur pembatasan tarif”](#) untuk informasi selengkapnya.
 - Perkiraan harga — Anda dapat membuka bagian untuk menampilkan bidang di mana Anda memasukkan jumlah permintaan/bulan yang berbeda dan melihat perkiraan baru.
6. Tinjau setelan distribusi yang tersisa, lalu pilih Buat distribusi.

Setelah Anda membuat distribusi, CloudFront buat dasbor Keamanan. Anda dapat menggunakan dasbor ini untuk menonaktifkan atau mengaktifkan AWS WAF. Jika Anda AWS WAF belum mengaktifkan, bagan dan grafik di dasbor tetap kosong.

Gunakan ACL web yang ada

Jika Anda memiliki ACL web yang ada, Anda dapat menggunakannya alih-alih perlindungan yang ditawarkan oleh AWS WAF.

Untuk menggunakan AWS WAF konfigurasi yang ada

1. Buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Lakukan salah satu hal berikut ini:
 - a. Pilih Buat distribusi dan ikuti langkah-langkahnya [Buat distribusi](#), lalu kembali ke topik ini.
 - b. Pilih konfigurasi yang ada, lalu pilih tab Keamanan.
3. Di bagian Web Application Firewall (WAF), pilih Edit, lalu Aktifkan perlindungan keamanan.
4. Pilih Gunakan konfigurasi WAF yang ada. Opsi ini hanya muncul jika Anda telah ACLs mengonfigurasi web.
5. Pilih ACL web yang ada dari tabel Choose a web ACL.

6. Tinjau setelan distribusi yang tersisa, lalu pilih Buat distribusi.

Aktifkan kontrol bot

Jika Anda mengaktifkan AWS WAF CloudFront distribusi, Anda dapat melihat permintaan bot untuk rentang waktu tertentu di bawah dasbor keamanan di CloudFront konsol. Anda juga dapat mengaktifkan atau menonaktifkan kontrol bot di sini.

Anda dikenakan biaya saat mengaktifkan kontrol bot. Dasbor keamanan memberikan perkiraan biaya.

Jika Anda mengaktifkan kontrol bot, dasbor keamanan menampilkan lalu lintas bot berdasarkan setiap jenis dan kategori bot. Jika Anda menonaktifkan kontrol bot, lalu lintas bot ditampilkan berdasarkan pengambilan sampel permintaan.

Untuk mengaktifkan kontrol bot

1. Buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Distribusi, lalu pilih distribusi yang ingin Anda ubah.
3. Pilih tab Security.
4. Gulir ke bawah ke permintaan Bot untuk bagian rentang waktu tertentu dan pilih Aktifkan Kontrol Bot.
5. Dalam kotak dialog Kontrol Bot, di bawah Konfigurasi, pilih kotak centang Aktifkan Kontrol Bot untuk bot umum.
6. Pilih Simpan perubahan.

Konfigurasikan perlindungan berdasarkan kategori bot

Saat Anda mengaktifkan kontrol bot, Anda dapat mengonfigurasi bagaimana setiap bot yang tidak diverifikasi ditangani per kategori bot. Misalnya, Anda dapat mengatur bot perpustakaan HTTP ke mode Monitor dan menetapkan Tantangan ke pemeriksa tautan.

Note

Bot yang dikenal umum dan dapat diverifikasi, seperti crawler mesin pencari yang dikenal, tidak tunduk pada tindakan yang Anda tetapkan di sini. AWS Kontrol bot mengonfirmasi

bahwa bot yang divalidasi berasal dari sumber yang mereka klaim sebelum menandainya sebagai diverifikasi.

Untuk mengkonfigurasi perlindungan untuk kategori bot

1. Buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Distribusi, lalu pilih distribusi yang ingin Anda ubah.
3. Pilih tab Security.
4. Dalam bagan kategori Permintaan menurut bot, arahkan ke salah satu item di kolom Tindakan bot yang tidak diverifikasi dan pilih ikon pensil untuk mengeditnya.
5. Buka daftar yang dihasilkan dan pilih salah satu dari yang berikut ini:
 - Blok
 - Izinkan
 - Modus monitor
 - CAPTCHA
 - Tantangan
6. Pilih tanda centang di sebelah daftar untuk menyimpan perubahan Anda.

Kelola perlindungan AWS WAF keamanan di dasbor CloudFront keamanan

CloudFront membuat dasbor keamanan untuk setiap distribusi Anda. Anda menggunakan dasbor di CloudFront konsol. Dengan dasbor, Anda dapat menggunakan CloudFront dan AWS WAF bersama-sama di satu lokasi untuk memantau dan mengelola perlindungan keamanan umum untuk aplikasi web Anda. Dasbor menyediakan tugas dan data berikut:

- Konfigurasi keamanan — Anda dapat mengaktifkan dan menonaktifkan AWS WAF perlindungan, dan melihat perlindungan khusus aplikasi seperti perlindungan. WordPress
- Tren keamanan — Ini termasuk permintaan yang diizinkan dan diblokir, permintaan tantangan dan CAPTCHA, dan jenis serangan teratas. Anda dapat melihat rasio lalu lintas dan bagaimana mereka berubah dari waktu ke waktu. Misalnya, jika semua permintaan meningkat sebesar 3% tetapi permintaan yang diizinkan meningkat sebesar 14%, itu berarti Anda mengizinkan sebagian besar lalu lintas Anda melalui periode saat ini.

- **Permintaan bot** — Anda dapat melihat berapa banyak lalu lintas yang berasal dari bot, jenis bot mana (terverifikasi vs tidak terverifikasi), dan bagaimana persentase alokasi jenis bot (terverifikasi vs tidak terverifikasi) berubah dari waktu ke waktu. Untuk informasi selengkapnya tentang mengaktifkan kontrol bot, lihat [Aktifkan kontrol bot](#).
- **Permintaan log** — Data log dapat membantu menjawab pertanyaan tentang tren keamanan atau permintaan bot. Anda dapat mencari log tanpa menulis kueri, dan melihat bagan agregat untuk membantu menentukan apakah kumpulan log yang difilter terutama didorong oleh subset metode HTTP, alamat IP, jalur URI, atau negara. Anda dapat mengarahkan kursor ke nilai di bagan dan memblokir alamat IP dan negara. Untuk informasi selengkapnya, lihat [Aktifkan AWS WAF log](#).
- **Manajemen pembatasan geografis** — CloudFront dan AWS WAF menyediakan fitur pembatasan geografis. CloudFront memberikan batasan geografis secara gratis, tetapi metrik untuk pembatasan CloudFront geografis tidak ditampilkan di dasbor keamanan. Untuk melihat metrik permintaan untuk permintaan negara yang diblokir, Anda harus menggunakan batasan AWS WAF geografis. Untuk melakukan ini, arahkan kursor ke bilah negara di dasbor keamanan dan blokir negara. Untuk informasi selengkapnya, lihat [Gunakan CloudFront batasan geografis](#).
- **Opsi Blokir** mungkin tidak tersedia jika sebelumnya Anda membuat AWS WAF aturan khusus di luar CloudFront konsol untuk memblokir negara.

Topik

- [Prasyarat](#)
- [Aktifkan AWS WAF log](#)

Prasyarat

Anda harus mengaktifkan AWS WAF jika Anda ingin melihat metrik keamanan di dasbor CloudFront Keamanan. Jika Anda tidak mengaktifkan AWS WAF, Anda hanya dapat menggunakan dasbor Keamanan untuk mengaktifkan AWS WAF atau mengonfigurasi batasan CloudFront geografis.

Untuk informasi selengkapnya tentang mengaktifkan AWS WAF, lihat [Aktifkan AWS WAF untuk distribusi](#).

Aktifkan AWS WAF log

AWS WAF data log dapat membantu Anda mengisolasi pola lalu lintas tertentu. Misalnya, log dapat menunjukkan kepada Anda dari mana lalu lintas tertentu berasal atau apa fungsinya.

Jika Anda mengaktifkan AWS WAF login ke CloudWatch, dasbor CloudFront keamanan akan melakukan kueri, agregat, dan menampilkan wawasan dari log. CloudWatch Kami tidak mengenakan biaya untuk menggunakan dasbor keamanan, tetapi CloudWatch harga berlaku untuk log yang ditanyakan melalui dasbor. Untuk informasi selengkapnya, lihat [CloudWatch Harga Amazon](#).

Untuk mengaktifkan log

1. Masukkan volume permintaan yang Anda harapkan di kotak Jumlah permintaan/bulan untuk memperkirakan biaya mengaktifkan log.
2. Pilih kotak centang Aktifkan AWS WAF log.
3. Pilih Aktifkan.

CloudFront membuat grup CloudWatch log dan memperbarui AWS WAF konfigurasi Anda untuk mulai masuk CloudWatch. Pada penggunaan pertama, data log dapat memakan waktu beberapa menit untuk muncul. Bagian Permintaan pada bagan mencantumkan setiap permintaan. Di bawah permintaan individual, diagram batang mengumpulkan data dengan metode HTTP, jalur URI teratas, alamat IP teratas, dan negara teratas. Grafik dapat membantu Anda menemukan pola. Misalnya, Anda mungkin melihat volume permintaan yang tidak proporsional dari satu alamat IP, atau data dari negara yang sebelumnya tidak pernah Anda lihat di log Anda. Anda dapat memfilter permintaan berdasarkan Negara, Header Host, dan atribut lainnya untuk membantu menemukan lalu lintas yang tidak diinginkan. Setelah Anda mengidentifikasi lalu lintas tersebut, arahkan kursor ke permintaan individu atau item bagan dan blokir alamat IP atau negara.

Note

Metrik yang ditampilkan didasarkan pada ACL web. Oleh karena itu, jika Anda mengaitkan ACL web yang sama ke beberapa distribusi, Anda akan melihat semua metrik untuk ACL web Anda, tidak hanya AWS WAF permintaan yang diproses untuk distribusi itu.

Mengatur pembatasan tarif

Pembatasan tarif adalah salah satu rekomendasi yang mungkin Anda terima saat mengonfigurasi perlindungan keamanan.

CloudFront selalu memungkinkan pembatasan laju dalam mode monitor. Saat mode monitor diaktifkan, CloudFront tangkap metrik yang memberi tahu Anda apakah tingkat yang Anda konfigurasi di bidang Pembatasan tarif telah terlampaui, seberapa sering, dan berapa banyak.

Setelah Anda menyimpan distribusi, CloudFront mulai mengumpulkan data berdasarkan nomor di bidang Pembatasan tarif.

Anda dapat mengaktifkan atau mengelola pengaturan pembatasan laju di bagian Security - Web Application Firewall (WAF) pada tab Keamanan dari CloudFront distribusi apa pun.

Note

Opsi pembatasan tarif hanya muncul di CloudFront konsol jika Anda menentukan asal kustom non-S3 untuk distribusi Anda. Jika tidak, Anda hanya akan melihat perlindungan Core diaktifkan untuk distribusi. Untuk informasi selengkapnya tentang tipe asal, lihat [Gunakan berbagai asal dengan CloudFront distribusi](#).

Untuk mengatur pembatasan tarif

1. Buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Distribusi, lalu pilih distribusi yang ingin Anda ubah.
3. Pilih tab Security.
4. Di bagian Security — Web Application Firewall (WAF), pilih Edit.
5. Di bawah Perlindungan tambahan, pilih Pembatasan tarif. Anda dapat secara opsional mengubah batas tarif. Jika Anda telah menyempurnakan tarif, pilih Simpan perubahan.
6. Di bagian Security — Web Application Firewall (WAF), di samping Pembatasan tarif, Anda dapat memilih mode Monitor dan kemudian memilih Aktifkan pemblokiran untuk menonaktifkan mode monitor. CloudFront akan mulai memblokir permintaan yang melebihi batas tarif yang ditentukan.

Untuk informasi selengkapnya tentang mengaktifkan AWS WAF dan membatasi tarif, lihat [Introducing CloudFront Security Dashboard](#), [postingan blog Unified CDN](#), dan [Security Experience](#).

Nonaktifkan perlindungan AWS WAF keamanan

Jika distribusi Anda tidak memerlukan perlindungan AWS WAF keamanan, Anda dapat menonaktifkan fitur ini dengan menggunakan CloudFront konsol.

Jika sebelumnya Anda mengaktifkan AWS WAF perlindungan dan tidak memilih konfigurasi WAF yang ada (juga dikenal sebagai perlindungan satu klik), CloudFront secara otomatis membuat ACL web untuk Anda. Untuk web yang ACLs dibuat dengan cara ini, CloudFront konsol akan memisahkan sumber daya dan menghapus ACL web.

Memutuskan hubungan ACL web berbeda dengan menghapusnya. Disassociating menghapus ACL web dari distribusi Anda, tetapi tidak dihapus dari Anda. Akun AWS Untuk informasi selengkapnya, lihat [Mengaitkan atau memisahkan ACL web dengan AWS sumber daya](#) di AWS WAF, AWS Firewall Manager, dan Panduan Pengembang. AWS Shield Advanced

Lihat prosedur berikut untuk menonaktifkan AWS WAF perlindungan dan memisahkan ACL web dari distribusi Anda.

Untuk menonaktifkan perlindungan AWS WAF keamanan di CloudFront

1. Buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Distribusi, lalu pilih distribusi yang ingin Anda ubah.
3. Pilih tab Keamanan dan kemudian pilih Edit.
4. Di bagian Web Application Firewall (WAF), pilih Nonaktifkan AWS WAF perlindungan.
5. Pilih Simpan perubahan.

Catatan

- Jika Anda menonaktifkan perlindungan AWS WAF keamanan dan Anda masih ingin menghapus ACL web dari Anda Akun AWS, Anda dapat menghapusnya secara manual. Ikuti prosedur untuk [menghapus ACL web](#). Di konsol AWS WAF & Shield, untuk ACLs halaman Web, Anda harus memilih daftar Global (CloudFront) untuk menemukan web ACLs.
- Ketika Anda menghapus distribusi dari CloudFront konsol, CloudFront akan mencoba untuk juga menghapus ACL web jika Anda memilih perlindungan satu-klik. Ini adalah upaya terbaik dan tidak selalu dijamin. Untuk informasi selengkapnya, lihat [Menghapus sebuah distribusi](#).

Konfigurasi akses aman dan batasi akses ke konten

CloudFront menyediakan beberapa opsi untuk mengamankan konten yang dikirimkannya. Berikut ini adalah beberapa cara yang dapat Anda gunakan CloudFront untuk mengamankan dan membatasi akses ke konten:

- Konfigurasi koneksi HTTPS
- Cegah pengguna di lokasi geografis tertentu agar tidak mengakses konten
- Mewajibkan pengguna untuk mengakses konten menggunakan URL yang CloudFront tandatangi atau cookie yang tandatangi
- Siapkan enkripsi tingkat lapangan untuk kolom konten khusus
- Gunakan AWS WAF untuk mengontrol akses ke konten Anda

Anda juga harus menerapkan DDoS-resilient arsitektur untuk infrastruktur dan aplikasi Anda. Untuk informasi selengkapnya, lihat [Praktik AWS Terbaik untuk Ketahanan DDoS](#).

Untuk informasi tambahan, lihat hal berikut:

- [Mengamankan pengiriman konten Anda dengan CloudFront](#)
- [SIEM di Layanan Amazon OpenSearch](#)

Topik

- [Gunakan HTTPS dengan CloudFront](#)
- [Gunakan nama domain alternatif dan HTTPS](#)
- [Otentikasi TLS timbal balik dengan CloudFront \(Viewer mTLS\)](#)
- [Asal TLS timbal balik dengan CloudFront](#)
- [Sajikan konten pribadi dengan URL yang tandatangi dan cookie yang tandatangi](#)
- [Membatasi akses ke AWS asal](#)
- [Batasi akses ke Application Load Balancers](#)
- [Batasi distribusi geografis konten Anda](#)
- [Gunakan enkripsi tingkat lapangan untuk membantu melindungi data sensitif](#)

Gunakan HTTPS dengan CloudFront

Anda dapat mengonfigurasi CloudFront agar pemirsa menggunakan HTTPS sehingga koneksi dienkripsi saat CloudFront berkomunikasi dengan pemirsa. Anda juga dapat mengonfigurasi CloudFront untuk menggunakan HTTPS dengan asal Anda sehingga koneksi dienkripsi saat CloudFront berkomunikasi dengan asal Anda.

Jika Anda mengonfigurasi CloudFront untuk mewajibkan HTTPS baik untuk berkomunikasi dengan pemirsa maupun untuk berkomunikasi dengan asal Anda, inilah yang terjadi saat CloudFront menerima permintaan:

1. Penampil mengirimkan permintaan HTTPS ke CloudFront. Ada beberapa SSL/TLS negosiasi di sini antara pemirsa dan CloudFront. Pada akhirnya, penampil mengajukan permintaan dalam format terenkripsi.
2. Jika lokasi CloudFront tepi berisi respons yang di-cache, CloudFront mengenkripsi respons dan mengembalikannya ke penampil, dan penampil mendekripsi.
3. Jika lokasi CloudFront edge tidak berisi respons cache, CloudFront lakukan negosiasi SSL/TLS dengan asal Anda dan, ketika negosiasi selesai, teruskan permintaan ke asal Anda dalam format terenkripsi.
4. Asal Anda mendekripsi permintaan, memprosesnya (menghasilkan respons), mengenkripsi respons, dan mengembalikan respons ke CloudFront
5. CloudFront mendekripsi respons, mengenkripsi ulang, dan meneruskannya ke pemirsa. CloudFront juga menyimpan respons di lokasi tepi sehingga tersedia saat diminta berikutnya.
6. Penampil akan mendekripsi respons.

Proses ini bekerja pada dasarnya dengan cara yang sama apakah asal Anda adalah bucket Amazon S3 MediaStore, atau custom origin seperti server HTTP/S.

Note

Untuk membantu menggagalkan serangan tipe negosiasi ulang SSL, CloudFront tidak mendukung negosiasi ulang untuk permintaan penampil dan asal.

Atau, Anda dapat mengaktifkan otentikasi timbal balik untuk CloudFront distribusi Anda. Untuk informasi selengkapnya, lihat [Otentikasi TLS timbal balik dengan CloudFront \(Viewer mTLS\)](#).

Untuk informasi tentang cara mewajibkan HTTPS antara pemirsa dan CloudFront, CloudFront dan antara dan asal Anda, lihat topik berikut.

Topik

- [Memerlukan HTTPS untuk komunikasi antara pemirsa dan CloudFront](#)
- [Memerlukan HTTPS untuk komunikasi antara CloudFront dan asal kustom Anda](#)
- [Memerlukan HTTPS untuk komunikasi antara CloudFront dan asal Amazon S3 Anda](#)
- [Protokol dan sandi yang didukung antara pemirsa dan CloudFront](#)
- [Protokol dan cipher yang didukung antara dan asal CloudFront](#)

Memerlukan HTTPS untuk komunikasi antara pemirsa dan CloudFront

Anda dapat mengonfigurasi satu atau beberapa perilaku cache dalam CloudFront distribusi Anda agar memerlukan HTTPS untuk komunikasi antara pemirsa dan CloudFront. Anda juga dapat mengonfigurasi satu atau lebih perilaku cache untuk memungkinkan HTTP dan HTTPS, sehingga CloudFront memerlukan HTTPS untuk beberapa objek tetapi tidak untuk yang lain. Langkah-langkah konfigurasi tergantung pada nama domain yang Anda gunakan di objek URLs:

- Jika Anda menggunakan nama domain yang CloudFront ditetapkan ke distribusi Anda, seperti `d111111abcdef8.cloudfront.net`, Anda mengubah setelan Kebijakan Protokol Penampil untuk satu atau beberapa perilaku cache agar memerlukan komunikasi HTTPS. Dalam konfigurasi itu, CloudFront berikan SSL/TLS sertifikat.

Untuk mengubah nilai Kebijakan Protokol Penampil dengan menggunakan CloudFront konsol, lihat prosedurnya nanti di bagian ini.

Untuk informasi tentang cara menggunakan CloudFront API untuk mengubah nilai `ViewerProtocolPolicy` elemen, lihat [UpdateDistribution](#) di Referensi Amazon CloudFront API.

- Jika Anda menggunakan nama domain Anda sendiri, seperti `contoh.com`, Anda perlu mengubah beberapa CloudFront pengaturan. Anda juga perlu menggunakan SSL/TLS sertifikat yang disediakan oleh AWS Certificate Manager (ACM), atau mengimpor sertifikat dari otoritas sertifikat pihak ketiga ke ACM atau toko sertifikat IAM. Untuk informasi selengkapnya, lihat [Gunakan nama domain alternatif dan HTTPS](#).

 Note

Jika Anda ingin memastikan bahwa objek yang didapat pemirsa CloudFront dienkripsi saat CloudFront mendapatkannya dari asal Anda, selalu gunakan HTTPS antara CloudFront dan asal Anda. Jika Anda baru saja mengubah dari HTTP ke HTTPS antara CloudFront dan asal Anda, kami sarankan Anda membatalkan objek di lokasi CloudFront tepi. CloudFront akan mengembalikan objek ke penampil terlepas dari apakah protokol yang digunakan oleh penampil (HTTP atau HTTPS) cocok dengan protokol yang CloudFront digunakan untuk mendapatkan objek. Untuk informasi lebih lanjut tentang menghapus atau mengganti objek dalam distribusi, lihat [Menambahkan, menghapus, atau mengganti konten yang CloudFront mendistribusikan](#).

Memerlukan HTTPS untuk pemirsa

Untuk mewajibkan HTTPS antara CloudFront pemirsa dan untuk satu atau beberapa perilaku cache, lakukan prosedur berikut.

Untuk mengonfigurasi CloudFront agar memerlukan HTTPS antara pemirsa dan CloudFront

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel atas CloudFront konsol, pilih ID untuk distribusi yang ingin Anda perbarui.
3. Pada tab Perilaku, pilih perilaku cache yang ingin Anda perbarui, lalu pilih Edit.
4. Tentukan salah satu nilai berikut untuk kebijakan protokol Viewer:

Arahkan ulang HTTP ke HTTPS

Penampil dapat menggunakan kedua protokol. HTTP GET dan HEAD permintaan secara otomatis dialihkan ke permintaan HTTPS. CloudFront mengembalikan kode status HTTP 301 (Dipindahkan Secara Permanen) bersama dengan URL HTTPS baru. Penampil kemudian mengirimkan kembali permintaan untuk CloudFront menggunakan URL HTTPS.

 Important

Jika Anda mengirim POST, PUT, DELETE, OPTIONS, atau PATCH melalui HTTP dengan perilaku cache HTTP ke HTTPS dan versi protokol permintaan HTTP 1.1 atau lebih tinggi, CloudFront mengalihkan permintaan ke lokasi HTTPS dengan kode status

HTTP 307 (Pengalihan Sementara). Ini menjamin bahwa permintaan dikirim kembali ke lokasi baru menggunakan metode dan muatan tubuh yang sama.

Jika Anda mengirim POST, PUT, DELETE, OPTIONS, atau PATCH permintaan melalui HTTP ke perilaku cache HTTPS dengan versi protokol permintaan di bawah HTTP 1.1, CloudFront mengembalikan kode status HTTP 403 (Terlarang).

Saat penampil membuat permintaan HTTP yang diarahkan ke permintaan HTTPS, CloudFront biaya untuk kedua permintaan. Untuk permintaan HTTP, biaya hanya untuk permintaan dan untuk header yang CloudFront kembali ke penampil. Untuk permintaan HTTPS, biaya adalah untuk permintaan, dan untuk header dan objek yang dikembalikan oleh asal Anda.

HTTPS saja

Penampil dapat mengakses konten Anda hanya jika mereka menggunakan HTTPS. Jika penampil mengirim permintaan HTTP alih-alih permintaan HTTPS, CloudFront mengembalikan kode status HTTP 403 (Terlarang) dan tidak mengembalikan objek.

5. Pilih Simpan perubahan.
6. Ulangi langkah 3 hingga 5 untuk setiap perilaku cache tambahan yang ingin Anda perlukan HTTPS untuk antara pemirsa dan CloudFront.
7. Konfirmasikan hal berikut sebelum menggunakan konfigurasi yang diperbarui dalam lingkungan produksi:
 - Pola jalur dalam setiap perilaku cache hanya berlaku untuk permintaan yang ingin digunakan penampil dengan HTTPS.
 - Perilaku cache tercantum dalam urutan yang CloudFront ingin Anda evaluasi. Untuk informasi selengkapnya, lihat [Pola jalur](#).
 - Perilaku singgahan sedang mengarahkan permintaan ke asal-usul yang benar.

Memerlukan HTTPS untuk komunikasi antara CloudFront dan asal kustom Anda

Anda dapat meminta HTTPS untuk komunikasi antara CloudFront dan asal Anda.

 Note

Jika asal Anda adalah bucket Amazon S3 yang dikonfigurasi sebagai titik akhir situs web, Anda tidak dapat mengonfigurasi CloudFront untuk menggunakan HTTPS dengan asal Anda karena Amazon S3 tidak mendukung HTTPS untuk titik akhir situs web.

Untuk meminta HTTPS antara CloudFront dan asal Anda, ikuti prosedur dalam topik ini untuk melakukan hal berikut:

1. Pada distribusi Anda, ubah pengaturan Kebijakan Protokol Asal untuk asal itu.
2. Instal SSL/TLS sertifikat di server asal Anda (ini tidak diperlukan saat Anda menggunakan asal Amazon S3 atau asal tertentu lainnya AWS).

Topik

- [Memerlukan HTTPS untuk asal kustom](#)
- [Instal SSL/TLS sertifikat pada asal kustom Anda](#)

Memerlukan HTTPS untuk asal kustom

Prosedur berikut menjelaskan cara mengonfigurasi penggunaan HTTPS CloudFront untuk berkomunikasi dengan penyeimbang beban Elastic Load Balancing, instans Amazon EC2, atau custom origin lainnya. Untuk informasi tentang menggunakan CloudFront API untuk memperbarui distribusi, lihat [UpdateDistribution](#) di Referensi CloudFront API Amazon.

Untuk mengonfigurasi CloudFront agar memerlukan HTTPS antara CloudFront dan asal kustom Anda

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel atas CloudFront konsol, pilih ID untuk distribusi yang ingin Anda perbarui.
3. Pada tab Perilaku, pilih asal yang ingin Anda perbarui, lalu pilih Edit.
4. Perbarui pengaturan berikut:

Kebijakan Protokol Asal

Ubah Kebijakan Protokol Asal untuk asal yang berlaku dalam distribusi Anda:

- Hanya HTTPS — hanya CloudFront menggunakan HTTPS untuk berkomunikasi dengan custom origin Anda.
- Match Viewer — CloudFront berkomunikasi dengan custom origin Anda menggunakan HTTP atau HTTPS, tergantung pada protokol permintaan viewer. Misalnya, jika Anda memilih Penampil Pencocokan untuk Kebijakan Protokol Asal dan penampil menggunakan HTTPS untuk meminta objek CloudFront, CloudFront juga menggunakan HTTPS untuk meneruskan permintaan ke asal Anda.

Pilih Penampil Kecocokan hanya jika Anda menentukan Arahkan ulang HTTP ke HTTPS atau HTTPS Saja untuk Kebijakan Protokol Penampil.

CloudFront cache objek hanya sekali meskipun pemirsa membuat permintaan menggunakan protokol HTTP dan HTTPS.

Protokol SSL Asal

Pilih Protokol SSL Asal untuk asal yang berlaku dalam distribusi Anda. SSLv3 Protokolnya kurang aman, jadi kami sarankan Anda memilih SSLv3 hanya jika asal Anda tidak mendukung TLSv1 atau lebih baru. TLSv1 Jabat tangan kompatibel dengan mundur dan maju SSLv3, tetapi TLSv1 .1 dan yang lebih baru tidak. Saat Anda memilih SSLv3, CloudFront hanya mengirim permintaan SSLv3 jabat tangan.

5. Pilih Simpan perubahan.
6. Ulangi langkah 3 hingga 5 untuk setiap asal tambahan yang ingin Anda perlukan HTTPS untuk antara CloudFront dan asal kustom Anda.
7. Konfirmasikan hal berikut sebelum menggunakan konfigurasi yang diperbarui dalam lingkungan produksi:
 - Pola jalur dalam setiap perilaku cache hanya berlaku untuk permintaan yang ingin digunakan penampil dengan HTTPS.
 - Perilaku cache tercantum dalam urutan yang CloudFront ingin Anda evaluasi. Untuk informasi selengkapnya, lihat [Pola jalur](#).
 - Perilaku singgahan sedang mengarahkan permintaan ke asal-usul perubahan Kebijakan Protokol Asal untuk.

Instal SSL/TLS sertifikat pada asal kustom Anda

Anda dapat menggunakan SSL/TLS sertifikat dari sumber berikut di asal kustom Anda:

- Jika asal Anda adalah penyeimbang beban Elastic Load Balancing, Anda bisa menggunakan sertifikat yang disediakan oleh AWS Certificate Manager (ACM). Anda juga dapat menggunakan sertifikat yang ditandatangani oleh otoritas sertifikat pihak ketiga tepercaya dan diimpor ke ACM.
- Untuk asal selain penyeimbang beban Elastic Load Balancing, Anda harus menggunakan sertifikat yang ditandatangani oleh otoritas sertifikat pihak ketiga tepercaya (CA), misalnya, Comodo, atau Symantec. DigiCert

Sertifikat yang dikembalikan dari asal harus menyertakan salah satu nama domain berikut:

- Nama domain di bidang domain Origin asal (DomainNamebidang di CloudFront API).
- Nama domain di Host header, jika perilaku cache dikonfigurasi untuk meneruskan Host header ke asal.

Saat CloudFront menggunakan HTTPS untuk berkomunikasi dengan asal Anda, CloudFront verifikasi bahwa sertifikat dikeluarkan oleh otoritas sertifikat tepercaya. CloudFront mendukung otoritas sertifikat yang sama seperti yang dilakukan Mozilla. Untuk daftar saat ini, lihat [Daftar Sertifikat CA yang Disertakan Mozilla](#). Anda tidak dapat menggunakan sertifikat yang ditandatangani sendiri untuk komunikasi HTTPS antara CloudFront dan asal Anda.

Important

Jika server asal mengembalikan sertifikat kedaluwarsa, sertifikat yang tidak valid, atau sertifikat yang ditandatangani sendiri, atau jika server asal mengembalikan rantai sertifikat dalam urutan yang salah, lepaskan koneksi TCP, CloudFront mengembalikan kode status HTTP 502 (Bad Gateway) ke penampil, dan menyetel header ke `X-Cache Error from cloudfront`. Juga, jika rantai lengkap sertifikat, termasuk sertifikat perantara, tidak ada, lepaskan CloudFront koneksi TCP.

Memerlukan HTTPS untuk komunikasi antara CloudFront dan asal Amazon S3 Anda

Jika asal Anda adalah bucket Amazon S3, opsi Anda untuk menggunakan HTTPS untuk komunikasi CloudFront bergantung pada cara Anda menggunakan bucket. Jika bucket Amazon S3 Anda dikonfigurasi sebagai titik akhir situs web, Anda tidak dapat mengonfigurasi CloudFront untuk

menggunakan HTTPS untuk berkomunikasi dengan asal Anda karena Amazon S3 tidak mendukung koneksi HTTPS dalam konfigurasi tersebut.

Jika asal Anda adalah bucket Amazon S3 yang mendukung komunikasi HTTPS, CloudFront teruskan permintaan ke S3 dengan menggunakan protokol yang digunakan pemirsa untuk mengirimkan permintaan. Pengaturan default untuk [Protokol \(hanya asal kustom\)](#) adalah Penampil Kecocokan dan tidak dapat diubah. Namun, jika Anda mengaktifkan kontrol akses asal (OAC) untuk asal Amazon S3 Anda, komunikasi yang digunakan CloudFront antara dan Amazon S3 tergantung pada pengaturan Anda. Untuk informasi selengkapnya, lihat [Buat kontrol akses asal baru](#).

Jika Anda ingin meminta HTTPS untuk komunikasi antara CloudFront dan Amazon S3, Anda harus mengubah nilai Kebijakan Protokol Penampil untuk Mengalihkan HTTP ke HTTPS atau HTTPS Saja. Prosedur nanti di bagian ini menjelaskan cara menggunakan CloudFront konsol untuk mengubah Kebijakan Protokol Penampil. Untuk informasi tentang penggunaan CloudFront API untuk memperbarui `ViewerProtocolPolicy` elemen untuk distribusi, lihat [UpdateDistribution](#) di Referensi Amazon CloudFront API.

Saat Anda menggunakan HTTPS dengan bucket Amazon S3 yang mendukung komunikasi HTTPS, Amazon S3 menyediakan SSL/TLS sertifikat, jadi Anda tidak perlu melakukannya.

Memerlukan HTTPS untuk asal Amazon S3

Prosedur berikut menunjukkan kepada Anda cara mengonfigurasi CloudFront agar memerlukan HTTPS ke asal Amazon S3 Anda.

Untuk mengonfigurasi CloudFront agar memerlukan HTTPS ke Amazon S3 asal Anda

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel atas CloudFront konsol, pilih ID untuk distribusi yang ingin Anda perbarui.
3. Di Perilaku, pilih perilaku cache yang ingin Anda perbarui, lalu pilih Edit.
4. Tentukan salah satu nilai untuk Kebijakan Protokol Penampil:

Arahkan ulang HTTP ke HTTPS

Pemirsa dapat menggunakan kedua protokol, tetapi permintaan HTTP secara otomatis dialihkan ke permintaan HTTPS. CloudFront mengembalikan kode status HTTP 301 (Dipindahkan Secara Permanen) bersama dengan URL HTTPS baru. Penampil kemudian mengirimkan kembali permintaan untuk CloudFront menggunakan URL HTTPS.

 Important

CloudFront tidak mengalihkan DELETE, OPTIONS, PATCHPOST, atau PUT permintaan dari HTTP ke HTTPS. Jika Anda mengonfigurasi perilaku cache untuk mengarahkan ulang ke HTTPS, CloudFront merespons HTTP, DELETE, OPTIONS PATCHPOST, atau PUT permintaan untuk perilaku cache tersebut dengan kode status HTTP 403 (Terlarang).

Saat penampil membuat permintaan HTTP yang diarahkan ke permintaan HTTPS, CloudFront biaya untuk kedua permintaan. Untuk permintaan HTTP, biaya hanya untuk permintaan dan untuk header yang CloudFront kembali ke penampil. Untuk permintaan HTTPS, biaya adalah untuk permintaan, dan untuk header dan objek yang dikembalikan oleh asal Anda.

HTTPS Saja

Penampil dapat mengakses konten Anda hanya jika mereka menggunakan HTTPS. Jika penampil mengirim permintaan HTTP alih-alih permintaan HTTPS, CloudFront mengembalikan kode status HTTP 403 (Terlarang) dan tidak mengembalikan objek.

5. Pilih Ya, Edit.
6. Ulangi langkah 3 hingga 5 untuk setiap perilaku cache tambahan yang ingin Anda perlukan HTTPS untuk antara pemirsa dan CloudFront, dan antara CloudFront dan S3.
7. Konfirmasikan hal berikut sebelum menggunakan konfigurasi yang diperbarui dalam lingkungan produksi:
 - Pola jalur dalam setiap perilaku cache hanya berlaku untuk permintaan yang ingin digunakan penampil dengan HTTPS.
 - Perilaku cache tercantum dalam urutan yang CloudFront ingin Anda evaluasi. Untuk informasi selengkapnya, lihat [Pola jalur](#).
 - Perilaku singgahan sedang mengarahkan permintaan ke asal-usul yang benar.

Protokol dan sandi yang didukung antara pemirsa dan CloudFront

Jika Anda [memerlukan HTTPS antara pemirsa dan CloudFront distribusi Anda](#), Anda harus memilih [kebijakan keamanan](#), yang menentukan pengaturan berikut:

- SSL/TLS Protokol minimum yang CloudFront digunakan untuk berkomunikasi dengan pemirsa.
- Cipher yang CloudFront dapat digunakan untuk mengenkripsi komunikasi dengan pemirsa.

Untuk memilih kebijakan keamanan, tentukan nilai yang berlaku untuk [Kebijakan keamanan \(SSL/ TLS versi minimum\)](#). Tabel berikut mencantumkan protokol dan cipher yang CloudFront dapat digunakan untuk setiap kebijakan keamanan.

Penampil harus mendukung setidaknya satu dari cipher yang didukung untuk membuat koneksi HTTPS dengan. CloudFront memilih cipher dalam urutan yang terdaftar dari antara cipher yang didukung pemirsa. Lihat juga [Nama cipher OpenSSL, s2n, dan RFC](#).

	Kebijakan keamanan								
	SSLv3	TLSv1	TLSv1_6	TLSv1.016	TLSv1.018	TLSv1.019	TLSv1.021	TLSv1.025	TLSv1.3_2025

SSL/TLS Protokol yang didukung

TLSv1.3	◆	◆	◆	◆	◆	◆	◆	◆	◆
TLSv1.2	◆	◆	◆	◆	◆	◆	◆	◆	
TLSv1.1	◆	◆	◆	◆					
TLSv1	◆	◆	◆						
SSLv3	◆								

Didukung TLSv1 .3 cipher

TLS_AES_128_GCM_SHA256	◆	◆	◆	◆	◆	◆	◆	◆	◆
TLS_AES_256_GCM_SHA384	◆	◆	◆	◆	◆	◆	◆	◆	◆
TLS_ _ _ CHACHA20 POLY1305 SHA256	◆	◆	◆	◆	◆	◆	◆		◆

Cipher ECDSA yang didukung

	Kebijakan keamanan								
	SSLv3	TLSv1	TLSv1.0_6	TLSv1.016	TLSv1.018	TLSv1.019	TLSv1.021	TLSv1.025	TLSv1.3_2025
ECDHE-ECDSA- -GCM- AES128 SHA256	◆	◆	◆	◆	◆	◆	◆	◆	
ECDHE-ECDSA- - AES128 SHA256	◆	◆	◆	◆	◆	◆			
ECDHE-ECDSA- - SHA AES128	◆	◆	◆	◆					
ECDHE-ECDSA- -GCM- AES256 SHA384	◆	◆	◆	◆	◆	◆	◆	◆	
ECDHE-ECD SA- - CHACHA20 POLY1305	◆	◆	◆	◆	◆	◆	◆		
ECDHE-ECDSA- - AES256 SHA384	◆	◆	◆	◆	◆	◆			
ECDHE-ECDSA- - SHA AES256	◆	◆	◆	◆					

Cipher RSA yang didukung

ECDHE-RSA- -GCM- AES128 SHA256	◆	◆	◆	◆	◆	◆	◆	◆	
ECDHE-RSA- - AES128 SHA256	◆	◆	◆	◆	◆	◆			
ECDHE-RSA- -SHA AES128	◆	◆	◆	◆					

	Kebijakan keamanan								
	SSLv3	TLSv1	TLSv1.2	TLSv1.016	TLSv1.018	TLSv1.019	TLSv1.021	TLSv1.025	TLSv1.3_2025
ECDHE-RSA- -GCM-AES256 SHA384	◆	◆	◆	◆	◆	◆	◆	◆	
ECDHE-RSA- - CHACHA20 POLY1305	◆	◆	◆	◆	◆	◆	◆		
ECDHE-RSA- - AES256 SHA384	◆	◆	◆	◆	◆	◆			
ECDHE-RSA- -SHA AES256	◆	◆	◆	◆					
AES128-GCM-SHA256	◆	◆	◆	◆	◆				
AES256-GCM-SHA384	◆	◆	◆	◆	◆				
AES128-SHA256	◆	◆	◆	◆	◆				
AES256-SHA	◆	◆	◆	◆					
AES128-SHA	◆	◆	◆	◆					
DES- CBC3 -SHA	◆	◆							
RC4-MD5	◆								

Nama cipher OpenSSL, s2n, dan RFC

OpenSSL dan [s2n](#) menggunakan nama lain untuk cipher selain penggunaan standar TLS ([RFC 2246](#), [RFC 4346](#), [RFC 5246](#), dan [RFC 8446](#)). Tabel berikut memetakan nama OpenSSL dan s2n ke nama RFC untuk cipher lain.

CloudFront mendukung pertukaran kunci klasik dan kuantum yang aman. Untuk pertukaran kunci klasik menggunakan kurva elips, CloudFront mendukung yang berikut:

- `prime256v1`
- `X25519`
- `secp384r1`

Untuk pertukaran kunci yang aman kuantum, CloudFront mendukung yang berikut:

- `X25519MLKEM768`
- `SecP256r1MLKEM768`

 Note

Pertukaran kunci aman kuantum hanya didukung dengan TLS 1.3. TLS 1.2 dan versi sebelumnya tidak mendukung pertukaran kunci kuantum yang aman.

Untuk informasi selengkapnya, lihat topik berikut:

- [Kriptografi Pasca-Kuantum](#)
- [Algoritma kriptografi dan Layanan AWS](#)
- [Pertukaran kunci hibrida di TLS 1.3](#)

Untuk informasi selengkapnya tentang persyaratan sertifikat CloudFront, lihat [Persyaratan untuk menggunakan SSL/TLS sertifikat dengan CloudFront](#).

Nama cipher OpenSSL dan s2n	Nama cipher RFC
Didukung TLSv1.3 cipher	
TLS_AES_128_GCM_SHA256	TLS_AES_128_GCM_SHA256
TLS_AES_256_GCM_SHA384	TLS_AES_256_GCM_SHA384
TLS_..._CHACHA20_POLY1305_SHA256	TLS_..._CHACHA20_POLY1305_SHA256
Cipher ECDSA yang didukung	

Nama cipher OpenSSL dan s2n	Nama cipher RFC
ECDHE-ECDSA- -GCM- AES128 SHA256	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
ECDHE-ECDSA- - AES128 SHA256	TLS_ECDHE_ECDSA_DENGAN_AES_128_CBC_SHA256
ECDHE-ECDSA- -SHA AES128	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
ECDHE-ECDSA- -GCM- AES256 SHA384	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
ECDHE-ECDSA- - CHACHA20 POLY1305	TLS_ECDHE_ECDSA_DENGAN_ _ _ CHACHA20 POLY1305 SHA256
ECDHE-ECDSA- - AES256 SHA384	TLS_ECDHE_ECDSA_DENGAN_AES_256_CBC_SHA384
ECDHE-ECDSA- -SHA AES256	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
Cipher RSA yang didukung	
ECDHE-RSA- -GCM- AES128 SHA256	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
ECDHE-RSA- - AES128 SHA256	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
ECDHE-RSA- -SHA AES128	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
ECDHE-RSA- -GCM- AES256 SHA384	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
ECDHE-RSA- - CHACHA20 POLY1305	TLS_ECDHE_RSA_DENGAN_ _ _ CHACHA20 POLY1305 SHA256

Nama cipher OpenSSL dan s2n	Nama cipher RFC
ECDHE-RSA- - AES256 SHA384	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
ECDHE-RSA- -SHA AES256	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
AES128-GCM- SHA256	TLS_RSA_WITH_AES_128_GCM_SHA256
AES256-GCM- SHA384	TLS_RSA_WITH_AES_256_GCM_SHA384
AES128-SHA256	TLS_RSA_WITH_AES_128_CBC_SHA256
AES256-SHA	TLS_RSA_WITH_AES_256_CBC_SHA
AES128-SHA	TLS_RSA_WITH_AES_128_CBC_SHA
DES- CBC3 -SHA	TLS_RSA_WITH_3DES_EDE_CBC_SHA
RC4-MD5	RC4TLS_RSA_DENGAN_ _128_ MD5

Skema tanda tangan yang didukung antara pemirsa dan CloudFront

CloudFront mendukung skema tanda tangan berikut untuk koneksi antara pemirsa dan CloudFront.

Skema tanda tangan	Kebijakan keamanan								
	SSLv3	TLSv1	TLSv1.6	TLSv1.016	TLSv1.018	TLSv1.019	TLSv1.021	TLSv1.025	TLSv1.3_2025
TLS_SIGNATURE_RSA_PSS_PSS_SHA256	◆	◆	◆	◆	◆	◆	◆	◆	◆
TLS_SIGNATURE	◆	◆	◆	◆	◆	◆	◆	◆	◆

	Kebijakan keamanan								
Skema tanda tangan	SSLv3	TLSv1	TLSv1.6	TLSv1.016	TLSv1.018	TLSv1.019	TLSv1.021	TLSv1.025	TLSv1.3_2025
ME_RSA_PSS_PSS_SHA384									
TLS_SIGNATURE_SCHEME_ME_RSA_PSS_SHA512	◆	◆	◆	◆	◆	◆	◆	◆	◆
TLS_SIGNATURE_SCHEME_ME_RSA_PSS_RSAE_SHA256	◆	◆	◆	◆	◆	◆	◆	◆	◆
TLS_SIGNATURE_SCHEME_ME_RSA_PSS_RSAE_SHA384	◆	◆	◆	◆	◆	◆	◆	◆	◆
TLS_SIGNATURE_SCHEME_ME_RSA_PSS_RSAE_SHA512	◆	◆	◆	◆	◆	◆	◆	◆	◆
PKCS1TLS_SIGNATURE_SCHEME_RSA_SHA256	◆	◆	◆	◆	◆	◆	◆	◆	◆
PKCS1TLS_SIGNATURE_SCHEME_RSA_SHA384	◆	◆	◆	◆	◆	◆	◆	◆	◆

	Kebijakan keamanan								
Skema tanda tangan	SSLv3	TLSv1	TLSv1.6	TLSv1.016	TLSv1.018	TLSv1.019	TLSv1.021	TLSv1.025	TLSv1.3_2025
PKCS1TLS_SIGNATURE_SCHEME_RSA_SHA512	◆	◆	◆	◆	◆	◆	◆	◆	◆
PKCS1TLS_SIGNATURE_SCHEME_RSA_SHA224	◆	◆	◆	◆	◆	◆	◆		
TLS_SIGNATURE_SCHEME_ECDSA_SHA256	◆	◆	◆	◆	◆	◆	◆	◆	◆
TLS_SIGNATURE_SCHEME_ECDSA_SHA384	◆	◆	◆	◆	◆	◆	◆	◆	◆
TLS_SIGNATURE_SCHEME_ECDSA_SHA512	◆	◆	◆	◆	◆	◆	◆	◆	◆
TLS_SIGNATURE_SCHEME_ECDSA_SHA224	◆	◆	◆	◆	◆	◆	◆		
TLS_SIGNATURE_SCHEME_ECDSA_R1_SECP256 SHA256	◆	◆	◆	◆	◆	◆	◆	◆	◆

	Kebijakan keamanan								
Skema tanda tangan	SSLv3	TLSv1	TLSv1.6	TLSv1.016	TLSv1.018	TLSv1.019	TLSv1.021	TLSv1.025	TLSv1.3_2025
TLS_SIGNATURE_ECDSA_R1_SECP384_SHA384	◆	◆	◆	◆	◆	◆	◆	◆	◆
PKCS1_SIGNATURE_SCHEME_RSA_SHA1	◆	◆	◆	◆					
TLS_SIGNATURE_ECDSA_SHA1	◆	◆	◆	◆					

Protokol dan cipher yang didukung antara dan asal CloudFront

Jika Anda memilih untuk [meminta HTTPS antara CloudFront dan asal Anda](#), Anda dapat memutuskan [SSL/TLS protokol mana yang memungkinkan](#) koneksi aman, dan CloudFront dapat terhubung ke asal menggunakan salah satu sandi ECDSA atau RSA yang tercantum dalam tabel berikut. Asal Anda harus mendukung setidaknya satu dari cipher ini CloudFront untuk membuat koneksi HTTPS ke asal Anda.

OpenSSL dan [s2n](#) menggunakan nama lain untuk cipher selain penggunaan standar TLS ([RFC 2246](#), [RFC 4346](#), [RFC 5246](#), dan [RFC 8446](#)). Tabel berikut mencakup nama OpenSSL dan s2n, dan nama RFC, untuk setiap cipher.

Untuk cipher dengan algoritma pertukaran kunci kurva elips, CloudFront mendukung kurva elips berikut:

- primer256v1
- secp384r1
- X25519

Nama cipher OpenSSL dan s2n	Nama cipher RFC
Cipher ECDSA yang didukung	
ECDHE-ECDSA- -GCM- AES256 SHA384	TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
ECDHE-ECDSA- - AES256 SHA384	TLS_ECDHE_ECDSA_DENGAN_AES_256_CBC_SHA384
ECDHE-ECDSA- -SHA AES256	TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
ECDHE-ECDSA- -GCM- AES128 SHA256	TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
ECDHE-ECDSA- - AES128 SHA256	TLS_ECDHE_ECDSA_DENGAN_AES_128_CBC_SHA256
ECDHE-ECDSA- -SHA AES128	TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
Cipher RSA yang didukung	
ECDHE-RSA- -GCM- AES256 SHA384	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
ECDHE-RSA- - AES256 SHA384	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
ECDHE-RSA- -SHA AES256	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
ECDHE-RSA- -GCM- AES128 SHA256	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
ECDHE-RSA- - AES128 SHA256	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256

Nama cipher OpenSSL dan s2n	Nama cipher RFC
ECDHE-RSA- -SHA AES128	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
AES256-SHA	TLS_RSA_WITH_AES_256_CBC_SHA
AES128-SHA	TLS_RSA_WITH_AES_128_CBC_SHA
DES- CBC3 -SHA	TLS_RSA_WITH_3DES_EDE_CBC_SHA
RC4-MD5	RC4TLS_RSA_DENGAN_ _128_ MD5

Skema tanda tangan yang didukung antara CloudFront dan asal

CloudFront mendukung skema tanda tangan berikut untuk koneksi antara CloudFront dan asal.

- PKCS1TLS_SIGNATURE_SCHEME_RSA_ _ SHA256
- PKCS1TLS_SIGNATURE_SCHEME_RSA_ _ SHA384
- PKCS1TLS_SIGNATURE_SCHEME_RSA_ _ SHA512
- PKCS1TLS_SIGNATURE_SCHEME_RSA_ _ SHA224
- TLS_SIGNATURE_SCHEME_ECDSA_ SHA256
- TLS_SIGNATURE_SCHEME_ECDSA_ SHA384
- TLS_SIGNATURE_SCHEME_ECDSA_ SHA512
- TLS_SIGNATURE_SCHEME_ECDSA_ SHA224
- PKCS1TLS_SIGNATURE_SCHEME_RSA_ _ SHA1
- TLS_SIGNATURE_SCHEME_ECDSA_ SHA1

Gunakan nama domain alternatif dan HTTPS

Jika Anda ingin menggunakan nama domain Anda sendiri di URL untuk file Anda (misalnya, `https://www.example.com/image.jpg`) dan Anda ingin pemirsa menggunakan HTTPS, Anda harus menyelesaikan langkah-langkah dalam topik berikut. (Jika Anda menggunakan nama domain CloudFront distribusi default di URL, misalnya `https://d111111abcdef8.cloudfront.net/image.jpg`, ikuti panduan dalam topik berikut: [Memerlukan HTTPS untuk komunikasi antara pemirsa dan CloudFront.](#))

⚠ Important

Saat Anda menambahkan sertifikat ke distribusi Anda, CloudFront segera menyebarkan sertifikat ke semua lokasi tepinya. Saat lokasi tepi baru tersedia, CloudFront menyebarkan sertifikat ke lokasi tersebut juga. Anda tidak dapat membatasi lokasi tepi yang CloudFront menyebarkan sertifikat ke.

Topik

- [Pilih cara CloudFront melayani permintaan HTTPS](#)
- [Persyaratan untuk menggunakan SSL/TLS sertifikat dengan CloudFront](#)
- [Kuota tentang penggunaan SSL/TLS sertifikat dengan CloudFront \(HTTPS antara pemirsa dan CloudFront hanya\)](#)
- [Konfigurasi nama domain alternatif dan HTTPS](#)
- [Tentukan ukuran kunci publik dalam sertifikat SSL/TLS RSA](#)
- [Tingkatkan kuota untuk sertifikat SSL/TLS](#)
- [Putar SSL/TLS sertifikat](#)
- [Kembalikan dari SSL/TLS sertifikat kustom ke sertifikat default CloudFront](#)
- [Beralih dari SSL/TLS sertifikat khusus dengan alamat IP khusus ke SNI](#)

Pilih cara CloudFront melayani permintaan HTTPS

Jika Anda ingin pemirsa menggunakan HTTPS dan menggunakan nama domain alternatif untuk file Anda, pilih salah satu opsi berikut untuk cara CloudFront melayani permintaan HTTPS:

- Gunakan [Indikasi Nama Server \(SNI\) — Rekomendasi](#)
- Gunakan alamat IP khusus di setiap lokasi tepi

Bagian ini menjelaskan cara kerja setiap opsi.

Gunakan SNI untuk melayani permintaan HTTPS (berfungsi untuk sebagian besar klien)

[Server Name Indication \(SNI\)](#) adalah ekstensi ke protokol TLS yang didukung oleh browser dan klien yang dirilis setelah 2010. Jika Anda mengonfigurasi CloudFront untuk melayani permintaan HTTPS

menggunakan SNI, CloudFront kaitkan nama domain alternatif Anda dengan alamat IP untuk setiap lokasi tepi. Saat penampil mengirimkan permintaan HTTPS untuk konten Anda, DNS mengirimkan permintaan ke alamat IP untuk lokasi tepi yang benar. Alamat IP ke nama domain Anda ditentukan selama negosiasi SSL/TLS jabat tangan; alamat IP tidak didedikasikan untuk distribusi Anda.

SSL/TLS Negosiasi terjadi di awal proses membangun koneksi HTTPS. Jika tidak CloudFront dapat segera menentukan domain mana permintaan itu, itu akan menghentikan koneksi. Saat penampil yang mendukung SNI mengirimkan permintaan HTTPS untuk konten Anda, inilah yang terjadi:

1. Penampil secara otomatis mendapatkan nama domain dari URL permintaan dan menambahkannya ke ekstensi SNI dari pesan halo klien TLS.
2. Saat CloudFront menerima halo klien TLS, ia menggunakan nama domain di ekstensi SNI untuk menemukan CloudFront distribusi yang cocok dan mengirimkan kembali sertifikat TLS terkait.
3. Penampil dan CloudFront melakukan SSL/TLS negosiasi.
4. CloudFront mengembalikan konten yang diminta ke pemirsa.

Untuk daftar terkini dari browser yang mendukung SNI, lihat entri Wikipedia [Indikasi Nama Server](#).

Jika Anda ingin menggunakan SNI tetapi beberapa browser pengguna tidak mendukung SNI, Anda memiliki beberapa opsi:

- Konfigurasi CloudFront untuk melayani permintaan HTTPS dengan menggunakan alamat IP khusus, bukan SNI. Untuk informasi selengkapnya, lihat [Gunakan alamat IP khusus untuk melayani permintaan HTTPS \(berfungsi untuk semua klien\)](#).
- Gunakan CloudFront SSL/TLS sertifikat alih-alih sertifikat khusus. Ini mengharuskan Anda menggunakan nama CloudFront domain untuk distribusi Anda di URL untuk file Anda, misalnya, `https://d1111111abcdef8.cloudfront.net/logo.png`.

Jika Anda menggunakan CloudFront sertifikat default, pemirsa harus mendukung protokol SSL TLSv1 atau yang lebih baru. CloudFront tidak mendukung SSLv3 dengan sertifikat default. CloudFront

Anda juga harus mengubah SSL/TLS sertifikat yang CloudFront digunakan dari sertifikat kustom ke CloudFront sertifikat default:

- Jika Anda belum menggunakan distribusi untuk mendistribusikan konten Anda, Anda dapat mengubah konfigurasinya. Untuk informasi selengkapnya, lihat [Perbarui distribusi](#).

- Jika Anda telah menggunakan distribusi Anda untuk mendistribusikan konten Anda, Anda harus membuat CloudFront distribusi baru dan mengubah URL untuk file Anda untuk mengurangi atau menghilangkan jumlah waktu konten Anda tidak tersedia. Untuk informasi selengkapnya, lihat [Kembalikan dari SSL/TLS sertifikat kustom ke sertifikat default CloudFront](#).
- Jika Anda dapat mengontrol browser yang digunakan pengguna, minta mereka meng-upgrade browser mereka ke browser yang mendukung SNI.
- Gunakan HTTP alih-alih HTTPS.

Gunakan alamat IP khusus untuk melayani permintaan HTTPS (berfungsi untuk semua klien)

Indikasi Nama Server (Ser Server Name Indication atau SNI) adalah salah satu cara untuk mengaitkan permintaan dengan domain. Cara lainnya adalah menggunakan alamat IP khusus. Jika Anda memiliki pengguna yang tidak dapat meng-upgrade ke browser atau klien yang dirilis setelah 2010, Anda dapat menggunakan alamat IP khusus untuk melayani permintaan HTTPS. Untuk daftar terkini dari browser yang mendukung SNI, lihat entri Wikipedia [Indikasi Nama Server](#).

 Important

Jika Anda mengonfigurasi CloudFront untuk melayani permintaan HTTPS menggunakan alamat IP khusus, Anda dikenakan biaya bulanan tambahan. Biaya dimulai ketika Anda mengaitkan SSL/TLS sertifikat Anda dengan distribusi dan Anda mengaktifkan distribusi. Untuk informasi selengkapnya tentang CloudFront harga, lihat [CloudFront Harga Amazon](#). Selain itu, lihat [Using the Same Certificate for Multiple CloudFront Distributions](#).

Saat Anda mengonfigurasi CloudFront untuk melayani permintaan HTTPS dengan menggunakan alamat IP khusus, CloudFront kaitkan sertifikat Anda dengan alamat IP khusus di setiap lokasi CloudFront tepi. Saat penampil mengirimkan permintaan HTTPS untuk konten Anda, inilah yang terjadi:

1. DNS mengarahkan permintaan ke alamat IP untuk distribusi Anda di lokasi edge yang berlaku.
2. Jika permintaan klien menyediakan ekstensi SNI dalam ClientHello pesan, CloudFront cari distribusi yang terkait dengan SNI tersebut.
 - Jika ada kecocokan, CloudFront tanggapinya permintaan dengan SSL/TLS sertifikat.

- Jika tidak ada kecocokan, CloudFront gunakan alamat IP sebagai gantinya untuk mengidentifikasi distribusi Anda dan untuk menentukan SSL/TLS sertifikat mana yang akan dikembalikan ke penampil.
3. Penampil dan CloudFront melakukan SSL/TLS negosiasi menggunakan SSL/TLS sertifikat Anda.
 4. CloudFront mengembalikan konten yang diminta ke pemirsa.

Metode ini berfungsi untuk setiap permintaan HTTPS, terlepas dari browser atau penampil lain yang digunakan pengguna.

 Note

IP khusus bukan IP statis dan dapat berubah seiring waktu. Alamat IP yang dikembalikan untuk lokasi tepi dialokasikan secara dinamis dari rentang alamat IP dari daftar [server CloudFront tepi](#).

Rentang alamat IP untuk server CloudFront tepi dapat berubah. Untuk diberitahu tentang perubahan alamat IP, [berlangganan Perubahan Alamat IP AWS Publik melalui Amazon SNS](#).

Meminta izin untuk menggunakan tiga atau lebih SSL/TLS sertifikat IP khusus

Jika Anda memerlukan izin untuk mengaitkan tiga atau lebih sertifikat IP SSL/TLS khusus secara permanen CloudFront, lakukan prosedur berikut. Untuk detail lebih lanjut tentang permintaan HTTPS, lihat [Pilih cara CloudFront melayani permintaan HTTPS](#).

 Note

Prosedur ini adalah untuk menggunakan tiga atau lebih sertifikat IP khusus di seluruh CloudFront distribusi Anda. Nilai default-nya adalah 2. Harap diingat bahwa Anda tidak dapat mengikat lebih dari satu sertifikat SSL ke distribusi.

Anda hanya dapat mengaitkan satu SSL/TLS sertifikat ke CloudFront distribusi pada satu waktu. Jumlah ini untuk jumlah total sertifikat SSL IP khusus yang dapat Anda gunakan di semua CloudFront distribusi Anda.

Untuk meminta izin untuk menggunakan tiga atau lebih sertifikat dengan CloudFront distribusi

1. Buka [Pusat Dukungan](#) dan membuat kasus.

2. Sebutkan berapa banyak sertifikat yang Anda perlukan izin penggunaannya, dan jelaskan keadaan di dalam permintaan Anda. Kami akan memperbarui akun Anda sesegera mungkin.
3. Lanjutkan dengan prosedur berikutnya.

Persyaratan untuk menggunakan SSL/TLS sertifikat dengan CloudFront

Persyaratan untuk SSL/TLS sertifikat dijelaskan dalam topik ini. Mereka berlaku untuk kedua hal berikut, kecuali sebagaimana disebutkan:

- Sertifikat untuk menggunakan HTTPS antara pemirsa dan CloudFront
- Sertifikat untuk menggunakan HTTPS antara CloudFront dan asal Anda

Topik

- [Penerbit sertifikat](#)
- [Wilayah AWS untuk AWS Certificate Manager](#)
- [Format Sertifikat](#)
- [Sertifikat Menengah](#)
- [Tipe Kunci](#)
- [Kunci privat](#)
- [Izin](#)
- [Ukuran kunci sertifikat](#)
- [Jenis sertifikat yang didukung](#)
- [Tanggal kedaluwarsa sertifikat dan perpanjangan](#)
- [Nama domain dalam CloudFront distribusi dan sertifikat](#)
- [Versi SSL/TLS protokol minimum](#)
- [Versi HTTP yang didukung](#)

Penerbit sertifikat

Kami menyarankan Anda menggunakan sertifikat publik yang dikeluarkan oleh [AWS Certificate Manager \(ACM\)](#). Untuk informasi tentang mendapatkan sertifikat dari ACM, lihat [Panduan AWS Certificate Manager Pengguna](#). Untuk menggunakan sertifikat ACM dengan CloudFront distribusi,

pastikan Anda meminta (atau mengimpor) sertifikat di Wilayah AS Timur (Virginia Utara) (us-east-1).

CloudFront mendukung otoritas sertifikat (CA) yang sama dengan Mozilla, jadi jika Anda tidak menggunakan ACM, gunakan sertifikat yang dikeluarkan oleh CA pada Daftar Sertifikat [CA Termasuk Mozilla](#).

Sertifikat TLS yang digunakan oleh asal yang Anda tentukan untuk CloudFront distribusi Anda juga harus dikeluarkan dari CA pada Daftar Sertifikat CA Termasuk Mozilla.

Untuk informasi lebih lanjut tentang mendapatkan dan menginstal sertifikat, lihat dokumentasi untuk perangkat lunak server HTTP Anda dan ke dokumentasi untuk CA.

Wilayah AWS untuk AWS Certificate Manager

Untuk menggunakan sertifikat di AWS Certificate Manager (ACM) untuk mewajibkan HTTPS antar pemirsa dan CloudFront, pastikan Anda meminta (atau mengimpor) sertifikat di Wilayah AS Timur (Virginia Utara) (us-east-1).

Jika Anda ingin meminta HTTPS antara CloudFront dan asal Anda, dan Anda menggunakan penyeimbang beban di Elastic Load Balancing sebagai asal Anda, Anda dapat meminta atau mengimpor sertifikat di mana pun. Wilayah AWS

Format Sertifikat

Sertifikat harus dalam format X.509 PEM. Ini adalah format default jika Anda menggunakan AWS Certificate Manager.

Sertifikat Menengah

Jika Anda menggunakan otoritas sertifikat pihak ketiga (CA), cantumkan semua sertifikat perantara dalam rantai sertifikat yang ada di .pem file, dimulai dengan sertifikat untuk CA yang menandatangani sertifikat untuk domain Anda. Biasanya, Anda akan menemukan file di situs web CA yang mencantumkan sertifikat perantara dan root dalam urutan rantai yang tepat.

Important

Jangan sertakan yang berikut ini: sertifikat root, sertifikat perantara yang tidak berada di jalur kepercayaan, atau sertifikat kunci publik CA Anda.

Berikut ini contohnya:

```
-----BEGIN CERTIFICATE-----  
Intermediate certificate 2  
-----END CERTIFICATE-----  
-----BEGIN CERTIFICATE-----  
Intermediate certificate 1  
-----END CERTIFICATE-----
```

Tipe Kunci

CloudFront mendukung pasangan kunci publik-pribadi RSA dan ECDSA.

CloudFront mendukung koneksi HTTPS ke pemirsa dan asal menggunakan sertifikat RSA dan ECDSA. Dengan [AWS Certificate Manager \(ACM\)](#), Anda dapat meminta dan mengimpor sertifikat RSA atau ECDSA dan kemudian mengaitkannya dengan distribusi Anda. CloudFront

Untuk daftar sandi RSA dan ECDSA yang didukung oleh CloudFront yang dapat Anda negosiasikan dalam koneksi HTTPS, lihat dan. [the section called “Protokol dan sandi yang didukung antara pemirsa dan CloudFront”](#) [the section called “Protokol dan cipher yang didukung antara dan asal CloudFront”](#)

Kunci privat

Jika Anda menggunakan sertifikat dari otoritas sertifikat pihak ketiga (CA), catat hal-hal berikut:

- Kunci pribadi harus cocok dengan kunci publik yang ada dalam sertifikat.
- Kunci pribadi harus dalam format PEM.
- Kunci pribadi tidak dapat dienkripsi dengan kata sandi.

Jika AWS Certificate Manager (ACM) memberikan sertifikat, ACM tidak melepaskan kunci pribadi. Kunci pribadi disimpan dalam ACM untuk digunakan oleh AWS layanan yang terintegrasi dengan ACM.

Izin

Anda harus memiliki izin untuk menggunakan dan mengimpor SSL/TLS sertifikat. Jika Anda menggunakan AWS Certificate Manager (ACM), sebaiknya gunakan AWS Identity and Access Management izin untuk membatasi akses ke sertifikat. Untuk informasi selengkapnya, lihat [Identitas dan manajemen akses](#) di Panduan AWS Certificate Manager Pengguna.

Ukuran kunci sertifikat

Ukuran kunci sertifikat yang CloudFront mendukung tergantung pada jenis kunci dan sertifikat.

Untuk sertifikat RSA:

CloudFront mendukung kunci RSA 1024-bit, 2048-bit, dan 3072-bit, dan 4096-bit. Panjang kunci maksimum untuk sertifikat RSA yang Anda gunakan CloudFront adalah 4096 bit.

Perhatikan bahwa ACM mengeluarkan sertifikat RSA dengan kunci hingga 2048-bit. Untuk menggunakan sertifikat RSA 3072-bit atau 4096-bit, Anda perlu mendapatkan sertifikat secara eksternal dan mengimpornya ke ACM, setelah itu akan tersedia untuk Anda gunakan. CloudFront

Untuk informasi tentang cara menentukan ukuran kunci RSA, lihat [Tentukan ukuran kunci publik dalam sertifikat SSL/TLS RSA](#).

Untuk sertifikat ECDSA:

CloudFront mendukung kunci 256-bit. Untuk menggunakan sertifikat ECDSA di ACM untuk mewajibkan HTTPS antar pemirsa dan CloudFront, gunakan kurva elips prime256v1.

Jenis sertifikat yang didukung

CloudFront mendukung semua jenis sertifikat yang dikeluarkan oleh otoritas sertifikat tepercaya.

Tanggal kedaluwarsa sertifikat dan perpanjangan

Jika Anda menggunakan sertifikat yang Anda dapatkan dari otoritas sertifikat pihak ketiga (CA), Anda harus memantau tanggal kedaluwarsa sertifikat dan memperbarui sertifikat yang Anda impor ke AWS Certificate Manager (ACM) atau mengunggah ke toko AWS Identity and Access Management sertifikat sebelum kedaluwarsa.

Important

Untuk menghindari masalah kedaluwarsa sertifikat, perbarui atau impor ulang sertifikat Anda setidaknya 24 jam sebelum NotAfter nilai sertifikat Anda saat ini. Jika sertifikat Anda kedaluwarsa dalam waktu 24 jam, mintalah sertifikat baru dari ACM atau impor sertifikat baru ke ACM. Selanjutnya, kaitkan sertifikat baru ke CloudFront distribusi.

CloudFront mungkin terus menggunakan sertifikat sebelumnya saat perpanjangan sertifikat atau impor ulang sedang berlangsung. Ini adalah proses asinkron yang dapat memakan waktu hingga 24 jam sebelum CloudFront menunjukkan perubahan Anda.

Jika Anda menggunakan sertifikat yang disediakan ACM, ACM mengelola perpanjangan sertifikat untuk Anda. Untuk informasi lebih lanjut, lihat [Perpanjangan Terkelola](#) dalam AWS Certificate Manager Panduan Pengguna.

Nama domain dalam CloudFront distribusi dan sertifikat

Saat Anda menggunakan custom origin, SSL/TLS sertifikat di asal Anda menyertakan nama domain di bidang Nama Umum, dan mungkin beberapa lainnya di bidang Nama Alternatif Subjek. (CloudFront mendukung karakter wildcard dalam nama domain sertifikat.)

Salah satu nama domain dalam sertifikat harus sesuai dengan nama domain yang Anda tentukan untuk Nama Domain Asal. Jika tidak ada nama domain yang cocok, CloudFront mengembalikan kode status HTTP 502 (Bad Gateway) ke penampil.

Important

Saat Anda menambahkan nama domain alternatif ke distribusi, CloudFront periksa apakah nama domain alternatif dicakup oleh sertifikat yang telah Anda lampirkan. Sertifikat harus mencakup nama domain alternatif di bidang nama alternatif subjek (SAN) pada sertifikat. Ini berarti bidang SAN harus berisi kecocokan persis untuk nama domain alternatif, atau berisi wildcard pada tingkat yang sama dengan nama domain alternatif yang Anda tambahkan. Untuk informasi selengkapnya, lihat [Persyaratan untuk menggunakan nama domain alternatif](#).

Versi SSL/TLS protokol minimum

Jika Anda menggunakan alamat IP khusus, tetapkan versi SSL/TLS protokol minimum untuk koneksi antara pemirsa dan CloudFront dengan memilih kebijakan keamanan.

Untuk informasi lebih lanjut, lihat [Kebijakan keamanan \(SSL/TLS versi minimum\)](#) dalam topik [Semua referensi pengaturan distribusi](#).

Versi HTTP yang didukung

Jika Anda mengaitkan satu sertifikat dengan lebih dari satu CloudFront distribusi, semua distribusi yang terkait dengan sertifikat harus menggunakan opsi yang sama untuk [Versi HTTP yang didukung](#). Anda menentukan opsi ini saat membuat atau memperbarui CloudFront distribusi.

Kuota tentang penggunaan SSL/TLS sertifikat dengan CloudFront (HTTPS antara pemirsa dan CloudFront hanya)

Perhatikan kuota berikut tentang penggunaan SSL/TLS sertifikat dengan CloudFront. Kuota ini hanya berlaku untuk SSL/TLS sertifikat yang Anda berikan dengan menggunakan AWS Certificate Manager (ACM), yang Anda impor ke ACM, atau unggah ke penyimpanan sertifikat IAM untuk komunikasi HTTPS antara pemirsa dan CloudFront.

Untuk informasi selengkapnya, lihat [Tingkatkan kuota untuk sertifikat SSL/TLS](#).

Jumlah maksimum sertifikat per CloudFront distribusi

Anda dapat mengaitkan maksimal satu SSL/TLS sertifikat dengan setiap CloudFront distribusi.

Jumlah maksimum sertifikat yang dapat Anda impor ke ACM atau unggah ke toko sertifikat IAM

Jika Anda memperoleh SSL/TLS sertifikat dari CA pihak ketiga, Anda harus menyimpan sertifikat di salah satu lokasi berikut:

- AWS Certificate Manager – Untuk kuota saat ini pada jumlah sertifikat ACM, lihat [Kuota](#) di Panduan Pengguna AWS Certificate Manager. Kuota terdaftar adalah total yang mencakup sertifikat yang Anda berikan dengan menggunakan ACM dan sertifikat yang Anda impor ke ACM.
- Penyimpanan sertifikat IAM — Untuk kuota saat ini (sebelumnya dikenal sebagai batas) pada jumlah sertifikat yang dapat Anda unggah ke toko sertifikat IAM untuk sebuah AWS akun, lihat [Batas IAM dan STS di Panduan Pengguna IAM](#). Anda dapat meminta kuota yang lebih tinggi di konsol Service Quotas.

Jumlah maksimum sertifikat per AWS akun (hanya alamat IP khusus)

Jika Anda ingin melayani permintaan HTTPS dengan menggunakan alamat IP khusus, perhatikan hal berikut:

- Secara default, CloudFront memberi Anda izin untuk menggunakan dua sertifikat dengan AWS akun Anda, satu untuk penggunaan sehari-hari dan satu untuk saat Anda perlu memutar sertifikat untuk beberapa distribusi.

- Jika Anda memerlukan lebih dari dua SSL/TLS sertifikat khusus untuk AWS akun Anda, Anda dapat meminta kuota yang lebih tinggi di konsol Service Quotas.

Gunakan sertifikat yang sama untuk CloudFront distribusi yang dibuat dengan menggunakan akun yang berbeda AWS

Jika Anda menggunakan CA pihak ketiga dan ingin menggunakan sertifikat yang sama dengan beberapa CloudFront distribusi yang dibuat menggunakan AWS akun yang berbeda, Anda harus mengimpor sertifikat ke ACM atau mengunggahnya ke penyimpanan sertifikat IAM sekali untuk setiap akun. AWS

Jika Anda menggunakan sertifikat yang disediakan oleh ACM, Anda tidak dapat mengonfigurasi CloudFront untuk menggunakan sertifikat yang dibuat oleh AWS akun lain.

Gunakan sertifikat yang sama untuk CloudFront dan untuk AWS layanan lainnya

Jika Anda membeli sertifikat dari otoritas sertifikat terpercaya seperti Comodo,, atau Symantec DigiCert, Anda dapat menggunakan sertifikat yang sama untuk CloudFront dan untuk layanan lainnya. AWS Jika Anda mengimpor sertifikat ke ACM, Anda perlu mengimpornya hanya satu kali untuk menggunakannya untuk beberapa layanan AWS .

Jika Anda menggunakan sertifikat yang diberikan oleh ACM, sertifikat tersebut disimpan di ACM.

Gunakan sertifikat yang sama untuk beberapa CloudFront distribusi

Anda dapat menggunakan sertifikat yang sama untuk setiap atau semua CloudFront yang Anda gunakan untuk memenuhi permintaan HTTPS. Perhatikan hal-hal berikut:

- Anda dapat menggunakan sertifikat yang sama untuk melayani permintaan menggunakan alamat IP khusus dan untuk melayani permintaan menggunakan SNI.
- Anda hanya dapat mengaitkan satu sertifikat dengan setiap distribusi.
- Setiap distribusi harus menyertakan satu atau beberapa nama domain alternatif yang juga muncul di Nama Umum bidang atau Nama Alternatif Subjek dalam sertifikat.
- Jika Anda melayani permintaan HTTPS menggunakan alamat IP khusus dan Anda membuat semua distribusi Anda dengan menggunakan AWS akun yang sama, Anda dapat secara signifikan mengurangi biaya Anda dengan menggunakan sertifikat yang sama untuk semua distribusi. CloudFront biaya untuk setiap sertifikat, bukan untuk setiap distribusi.

Misalnya, Anda membuat tiga distribusi dengan menggunakan AWS akun yang sama, dan Anda menggunakan sertifikat yang sama untuk ketiga distribusi. Anda hanya akan dikenakan satu biaya untuk menggunakan alamat IP khusus.

Namun, jika Anda melayani permintaan HTTPS menggunakan alamat IP khusus dan menggunakan sertifikat yang sama untuk membuat CloudFront distribusi di AWS akun yang berbeda, setiap akun dikenakan biaya untuk menggunakan alamat IP khusus. Misalnya, jika Anda membuat tiga distribusi dengan menggunakan tiga AWS akun berbeda dan Anda menggunakan sertifikat yang sama untuk ketiga distribusi, setiap akun dikenakan biaya penuh untuk menggunakan alamat IP khusus.

Konfigurasi nama domain alternatif dan HTTPS

Untuk menggunakan nama domain alternatif di URL untuk file Anda dan menggunakan HTTPS antar pemirsa dan CloudFront, lakukan prosedur yang berlaku.

Topik

- [Dapatkan SSL/TLS sertifikat](#)
- [Impor SSL/TLS sertifikat](#)
- [Perbarui CloudFront distribusi Anda](#)

Dapatkan SSL/TLS sertifikat

Dapatkan SSL/TLS sertifikat jika Anda belum memilikinya. Untuk informasi lebih lanjut, lihat dokumentasi yang berlaku:

- Untuk menggunakan sertifikat yang disediakan oleh AWS Certificate Manager (ACM), lihat [Panduan AWS Certificate Manager Pengguna](#). Lalu, langsung ke [Perbarui CloudFront distribusi Anda](#).

Note

Kami menyarankan Anda menggunakan ACM untuk menyediakan, mengelola, dan menyebarkan SSL/TLS sertifikat pada sumber daya AWS terkelola. Anda harus meminta sertifikat ACM di Wilayah Timur AS (N. Virginia).

- Untuk mendapatkan sertifikat dari otoritas sertifikat pihak ketiga (CA), lihat dokumentasi yang diberikan oleh otoritas sertifikat. Jika Anda memiliki sertifikat, lanjutkan dengan prosedur berikutnya.

Impor SSL/TLS sertifikat

Jika Anda mendapatkan sertifikat Anda dari CA pihak ketiga, impor sertifikat ke ACM atau unggah ke toko sertifikat IAM:

ACM (disarankan)

ACM memungkinkan Anda mengimpor sertifikat pihak ketiga dari konsol ACM, dan juga secara terprogram. Untuk informasi tentang mengimpor sertifikat ke ACM, lihat [Mengimpor Sertifikat ke dalam AWS Certificate Manager](#) dalam Panduan Pengguna AWS Certificate Manager . Anda harus mengimpor sertifikat di Wilayah Timur AS (N. Virginia).

Toko sertifikat IAM

(Tidak disarankan) Gunakan AWS CLI perintah berikut untuk mengunggah sertifikat pihak ketiga Anda ke toko sertifikat IAM.

```
aws iam upload-server-certificate \  
  --server-certificate-name CertificateName \  
  --certificate-body file://public_key_certificate_file \  
  --private-key file://privatekey.pem \  
  --certificate-chain file://certificate_chain_file \  
  --path /cloudfront/path/
```

Perhatikan hal-hal berikut:

- AWS akun — Anda harus mengunggah sertifikat ke toko sertifikat IAM menggunakan AWS akun yang sama dengan yang Anda gunakan untuk membuat CloudFront distribusi Anda.
- --parameter batas – Ketika Anda mengunggah sertifikat ke IAM, nilai --path parameter (jalur sertifikat) harus dimulai dengan /cloudfront/, misalnya, /cloudfront/production/ atau /cloudfront/test/. Jalan harus diakhiri dengan /.
- Sertifikat yang ada – Anda harus menentukan nilai untuk --server-certificate-name dan --path parameter yang berbeda dari nilai yang terkait dengan sertifikat yang ada.
- Menggunakan CloudFront konsol — Nilai yang Anda tentukan untuk --server-certificate-name parameter di AWS CLI, misalnyamyServerCertificate, muncul di daftar Sertifikat SSL di CloudFront konsol.
- Menggunakan CloudFront API — Catat string alfanumerik yang AWS CLI dikembalikan, misalnya,. AS1A2M3P4L5E67SIIXR3J Ini adalah nilai yang akan Anda tentukan di elemen IAMCertificateId. Anda tidak perlu IAM ARN, yang juga dikembalikan oleh CLI.

Untuk informasi selengkapnya tentang AWS CLI, lihat [Panduan AWS Command Line Interface Pengguna](#) dan [Referensi AWS CLI Perintah](#).

Perbarui CloudFront distribusi Anda

Untuk memperbarui pengaturan distribusi Anda, lakukan prosedur berikut:

Untuk mengonfigurasi CloudFront distribusi Anda untuk nama domain alternatif

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Pilih ID untuk distribusi yang ingin Anda perbarui.
3. Di Umum pilih, pilih Edit.
4. Perbarui nilai berikut:

Nama domain alternatif (CNAME)

Pilih Tambahkan item untuk menambahkan nama domain alternatif yang berlaku. Pisahkan nama domain dengan koma, atau ketikkan setiap nama domain pada baris baru.

Sertifikat SSL khusus

Pilih sertifikat dari daftar dropdown.

Hingga 100 sertifikat tercantum di sini. Jika Anda memiliki lebih dari 100 sertifikat dan tidak melihat sertifikat yang ingin Anda tambahkan, Anda dapat mengetikkan sertifikat ARN di bidang untuk memilihnya.

Jika Anda mengunggah sertifikat ke penyimpanan sertifikat IAM tetapi tidak tercantum, dan Anda tidak dapat memilihnya dengan mengetikkan nama di bidang, periksa prosedurnya [Impor SSL/TLS sertifikat](#) untuk mengonfirmasi bahwa Anda telah mengunggah sertifikat dengan benar.

Important

Setelah Anda mengaitkan SSL/TLS sertifikat Anda dengan CloudFront distribusi Anda, jangan hapus sertifikat dari ACM atau penyimpanan sertifikat IAM sampai Anda menghapus sertifikat dari semua distribusi dan semua distribusi dikerahkan.

5. Pilih Simpan perubahan.
6. Konfigurasi CloudFront untuk mewajibkan HTTPS antara pemirsa dan CloudFront:
 - a. Di Perilaku pilih perilaku cache yang ingin Anda perbarui, lalu pilih Edit.
 - b. Tentukan salah satu nilai untuk Kebijakan Protokol Penampil:

Arahkan ulang HTTP ke HTTPS

Pemirsa dapat menggunakan kedua protokol, tetapi permintaan HTTP secara otomatis dialihkan ke permintaan HTTPS. CloudFront mengembalikan kode status HTTP 301 (Moved Permanently) bersama dengan URL HTTPS baru. Penampil kemudian mengirimkan kembali permintaan untuk CloudFront menggunakan URL HTTPS.

Important

CloudFront tidak mengalihkan DELETE, OPTIONS, PATCHPOST, atau PUT permintaan dari HTTP ke HTTPS. Jika Anda mengonfigurasi perilaku cache untuk mengarahkan ulang ke HTTPS, CloudFront merespons HTTP DELETE, OPTIONS, PATCHPOST, atau PUT permintaan untuk perilaku cache tersebut dengan kode status HTTP 403 (Forbidden).

Saat penampil membuat permintaan HTTP yang dialihkan ke permintaan HTTPS, CloudFront dikenakan biaya untuk kedua permintaan tersebut. Untuk permintaan HTTP, tagihan hanya untuk permintaan dan header yang CloudFront kembali ke penampil. Untuk permintaan HTTPS, biayanya adalah untuk permintaan, dan untuk header dan file yang dikembalikan oleh asal Anda.

HTTPS Saja

Penampil dapat mengakses konten Anda hanya jika mereka menggunakan HTTPS. Jika penampil mengirim permintaan HTTP bukan permintaan HTTPS, CloudFront mengembalikan kode status HTTP 403 (Forbidden) dan tidak mengembalikan file.

- c. Pilih Ya, Edit.
 - d. Ulangi langkah a hingga c untuk setiap perilaku cache tambahan yang ingin Anda perlukan HTTPS antara pemirsa dan CloudFront.
7. Konfirmasikan hal berikut sebelum menggunakan konfigurasi yang diperbarui dalam lingkungan produksi:

- Pola jalur dalam setiap perilaku cache hanya berlaku untuk permintaan yang ingin digunakan penampil dengan HTTPS.
- Perilaku singgahan dicantumkan sesuai urutan yang Anda inginkan CloudFront untuk mengevaluasinya. Untuk informasi selengkapnya, lihat [Pola jalur](#).
- Perilaku singgahan sedang mengarahkan permintaan ke asal-usul yang benar.

Tentukan ukuran kunci publik dalam sertifikat SSL/TLS RSA

Saat Anda menggunakan nama domain CloudFront alternatif dan HTTPS, ukuran maksimum kunci publik dalam sertifikat SSL/TLS RSA adalah 4096 bit. (Ini adalah ukuran kunci, bukan jumlah karakter dalam kunci publik.) Jika Anda menggunakan AWS Certificate Manager untuk sertifikat Anda, meskipun ACM mendukung kunci RSA yang lebih besar, Anda tidak dapat menggunakan kunci yang lebih besar dengan CloudFront.

Anda dapat menentukan ukuran kunci publik RSA dengan menjalankan perintah OpenSSL berikut:

```
openssl x509 -in path and filename of SSL/TLS certificate -text -noout
```

Di mana:

- `-in` menentukan path dan nama file sertifikat SSL/TLS RSA Anda.
- `-text` menyebabkan OpenSSL menampilkan panjang kunci publik RSA dalam bit.
- `-noout` mencegah OpenSSL menampilkan kunci publik.

Contoh output:

```
Public-Key: (2048 bit)
```

Tingkatkan kuota untuk sertifikat SSL/TLS

Ada kuota pada jumlah SSL/TLS sertifikat yang dapat Anda impor ke AWS Certificate Manager (ACM) atau unggah ke AWS Identity and Access Management (IAM). Ada juga kuota pada jumlah SSL/TLS sertifikat yang dapat Anda gunakan dengan Akun AWS ketika Anda mengonfigurasi CloudFront untuk melayani permintaan HTTPS dengan menggunakan alamat IP khusus. Namun, Anda dapat meminta kuota yang lebih tinggi.

Topik

- [Meningkatkan kuota sertifikat yang diimpor ke ACM](#)
- [Meningkatkan kuota sertifikat yang diunggah ke IAM](#)
- [Meningkatkan kuota pada sertifikat yang digunakan dengan alamat IP khusus](#)

Meningkatkan kuota sertifikat yang diimpor ke ACM

Untuk kuota jumlah sertifikat yang dapat Anda impor menjadi ACM, lihat [Kuota](#) dalam Panduan Pengguna AWS Certificate Manager .

Untuk meminta kuota yang lebih tinggi, gunakan konsol Service Quotas. Untuk informasi selengkapnya, lihat [Meminta peningkatan kuota](#) di Panduan Pengguna Service Quotas.

Meningkatkan kuota sertifikat yang diunggah ke IAM

Untuk kuota (sebelumnya dikenal sebagai batas) pada jumlah sertifikat yang dapat Anda unggah ke IAM, lihat [Batas IAM dan STS](#) dalam Panduan Pengguna IAM.

Untuk meminta kuota yang lebih tinggi, gunakan konsol Service Quotas. Untuk informasi selengkapnya, lihat [Meminta peningkatan kuota](#) di Panduan Pengguna Service Quotas.

Meningkatkan kuota pada sertifikat yang digunakan dengan alamat IP khusus

Untuk kuota jumlah sertifikat SSL yang dapat Anda gunakan untuk setiap Akun AWS saat melayani permintaan HTTPS menggunakan alamat IP khusus, lihat. [Kuota pada sertifikat SSL](#)

Untuk meminta kuota yang lebih tinggi, gunakan konsol Service Quotas. Untuk informasi selengkapnya, lihat [Meminta peningkatan kuota](#) di Panduan Pengguna Service Quotas.

Putar SSL/TLS sertifikat

Ketika SSL/TLS sertifikat Anda hampir kedaluwarsa, Anda perlu memutarnya untuk memastikan keamanan distribusi Anda dan menghindari gangguan layanan bagi pemirsa Anda. Anda dapat memutarnya dengan cara berikut:

- Untuk SSL/TLS sertifikat yang disediakan oleh AWS Certificate Manager (ACM), Anda tidak perlu memutarnya. ACM secara otomatis mengelola perpanjangan sertifikat untuk Anda. Untuk informasi selengkapnya, lihat [Perpanjangan sertifikat terkelola](#) di Panduan AWS Certificate Manager Pengguna.

- Jika Anda menggunakan otoritas sertifikat pihak ketiga dan Anda mengimpor sertifikat ke ACM (disarankan) atau mengunggahnya ke toko sertifikat IAM, Anda harus sesekali mengganti satu sertifikat dengan yang lain.

Important

- ACM tidak mengelola perpanjangan sertifikat untuk sertifikat yang Anda peroleh dari otoritas sertifikat pihak ketiga dan impor ke ACM.
- Jika Anda mengonfigurasi CloudFront untuk melayani permintaan HTTPS dengan menggunakan alamat IP khusus, Anda mungkin dikenakan biaya tambahan pro-rating untuk menggunakan satu atau beberapa sertifikat tambahan saat Anda memutar sertifikat. Kami menyarankan Anda memperbarui distribusi Anda untuk meminimalkan biaya tambahan.

Putar SSL/TLS sertifikat

Untuk memutar sertifikat Anda, lakukan prosedur berikut. Penampil dapat terus mengakses konten Anda saat Anda memutar sertifikat serta setelah proses selesai.

Untuk memutar SSL/TLS sertifikat

1. [Tingkatkan kuota untuk sertifikat SSL/TLS](#) untuk menentukan apakah Anda memerlukan izin untuk menggunakan lebih banyak sertifikat SSL. Jika ya, minta izin dan tunggu sampai izin diberikan sebelum Anda melanjutkan langkah 2.
2. Impor sertifikat baru ke ACM atau unggah ke IAM. Untuk informasi selengkapnya, lihat [Mengimpor SSL/TLS Sertifikat](#) di Panduan CloudFront Pengembang Amazon.
3. (Hanya untuk sertifikat IAM) Perbarui distribusi Anda satu per satu untuk menggunakan sertifikat baru. Untuk informasi selengkapnya, lihat [Perbarui distribusi](#).
4. (Opsional) Hapus sertifikat sebelumnya dari ACM atau IAM.

⚠ Important

Jangan menghapus SSL/TLS sertifikat sampai Anda menghapusnya dari semua distribusi dan hingga status distribusi yang telah Anda perbarui telah berubah.
Deployed

Kembalikan dari SSL/TLS sertifikat kustom ke sertifikat default CloudFront

Jika Anda mengonfigurasi CloudFront untuk menggunakan HTTPS antara penonton dan CloudFront, dan Anda mengonfigurasi CloudFront untuk menggunakan SSL/TLS sertifikat kustom, Anda dapat mengubah konfigurasi untuk menggunakan CloudFront SSL/TLS sertifikat default. Proses tersebut bergantung pada apakah Anda telah menggunakan distribusi Anda untuk mendistribusikan konten:

- Jika Anda belum menggunakan distribusi untuk mendistribusikan konten, Anda dapat mengubah konfigurasi saja. Untuk informasi selengkapnya, lihat [Perbarui distribusi](#).
- Jika Anda telah menggunakan distribusi Anda untuk mendistribusikan konten Anda, Anda harus membuat CloudFront distribusi baru dan mengubah URL untuk file Anda untuk mengurangi atau menghilangkan jumlah waktu konten Anda tidak tersedia. Untuk melakukannya, lakukan prosedur berikut.

Kembalikan ke sertifikat default CloudFront

Prosedur berikut menunjukkan kepada Anda cara mengembalikan dari sertifikat kustom ke SSL/TLS CloudFront sertifikat default.

Untuk kembali ke sertifikat default CloudFront

1. Buat CloudFront distribusi baru dengan konfigurasi yang diinginkan. Untuk Sertifikat SSL, pilih Default CloudFront Sertifikat (*.cloudfront.net).

Untuk informasi selengkapnya, lihat [Buat distribusi](#).

2. Untuk file yang Anda distribusikan CloudFront, perbarui URL di aplikasi Anda untuk menggunakan nama domain yang CloudFront ditetapkan ke distribusi baru. Misalnya, perubahan `https://www.example.com/images/logo.png` ke `https://d111111abcdef8.cloudfront.net/images/logo.png`.

3. Hapus distribusi yang terkait dengan SSL/TLS sertifikat kustom, atau perbarui distribusi untuk mengubah nilai Sertifikat SSL menjadi CloudFront Sertifikat Default (*.cloudfront.net). Untuk informasi selengkapnya, lihat [Perbarui distribusi](#).

 Important

Sampai Anda menyelesaikan langkah ini, AWS terus menagih Anda untuk menggunakan SSL/TLS sertifikat khusus.

4. (Opsional) Hapus SSL/TLS sertifikat kustom Anda.
 - a. Jalankan AWS CLI perintah `list-server-certificates` untuk mendapatkan ID sertifikat yang ingin Anda hapus. Untuk informasi lebih lanjut, lihat [list-server-certificates](#) dalam Referensi Perintah AWS CLI .
 - b. Jalankan AWS CLI perintah `delete-server-certificate` untuk menghapus sertifikat. Untuk informasi selengkapnya, lihat [menghapus-server-sertifikat](#) di Referensi Perintah AWS CLI

Beralih dari SSL/TLS sertifikat khusus dengan alamat IP khusus ke SNI

Jika Anda mengonfigurasi CloudFront untuk menggunakan SSL/TLS sertifikat khusus dengan alamat IP khusus, Anda dapat beralih menggunakan SSL/TLS sertifikat khusus dengan SNI sebagai gantinya dan menghilangkan biaya yang terkait dengan alamat IP khusus.

 Important

Pembaruan CloudFront konfigurasi Anda ini tidak berpengaruh pada pemirsa yang mendukung SNI. Pemirsa dapat mengakses konten Anda sebelum dan sesudah perubahan, serta saat perubahan menyebar ke lokasi CloudFront tepi. Pemirsa yang tidak mendukung SNI tidak dapat mengakses konten Anda setelah perubahan. Untuk informasi selengkapnya, lihat [Pilih cara CloudFront melayani permintaan HTTPS](#).

Beralih dari sertifikat khusus ke SNI

Prosedur berikut menunjukkan kepada Anda cara beralih dari SSL/TLS sertifikat khusus dengan alamat IP khusus ke SNI.

Untuk beralih dari SSL/TLS sertifikat khusus dengan alamat IP khusus ke SNI

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Pilih ID distribusi yang ingin Anda lihat atau perbarui.
3. Pilih Pengaturan Distribusi.
4. Di Umum pilih, pilih Edit.
5. Di bawah sertifikasi SSL Kustom - opsional, batalkan pilihan dukungan klien Legacy.
6. Pilih Ya, Edit.

Otentikasi TLS timbal balik dengan CloudFront (Viewer mTLS)

Mutual TLS Authentication (Mutual Transport Layer Security Authentication — mTLS) adalah protokol keamanan yang memperluas otentikasi TLS standar dengan memerlukan otentikasi berbasis sertifikat dua arah, di mana klien dan server harus membuktikan identitas mereka sebelum membuat koneksi yang aman. Menggunakan TLS timbal balik, Anda dapat memastikan bahwa hanya klien yang menyajikan sertifikat TLS tepercaya yang mendapatkan akses ke distribusi Anda. CloudFront

Cara kerjanya

Dalam jabat tangan TLS standar, hanya server yang menyajikan sertifikat untuk membuktikan identitasnya kepada klien. Dengan TLS timbal balik, proses otentikasi menjadi dua arah. Ketika klien mencoba untuk terhubung ke CloudFront distribusi Anda, CloudFront meminta sertifikat klien selama jabat tangan TLS. Klien harus menunjukkan X.509 sertifikat yang valid yang CloudFront memvalidasi terhadap toko kepercayaan Anda yang dikonfigurasi sebelum membuat koneksi aman.

CloudFront melakukan validasi sertifikat ini di lokasi AWS tepi, menurunkan kompleksitas otentikasi dari server asal Anda sambil mempertahankan CloudFront manfaat kinerja global. Anda dapat mengonfigurasi mTL dalam tiga mode:

- Mode wajib (default) - CloudFront memvalidasi sertifikat klien terhadap toko kepercayaan. Jika validasi gagal atau tidak ada sertifikat yang disajikan, CloudFront tolak koneksi. Gunakan mode wajib ketika setiap klien harus mengautentikasi dengan sertifikat yang valid.
- Mode opsional — CloudFront memvalidasi sertifikat klien jika disajikan, tetapi memungkinkan koneksi tanpa sertifikat. Metadata sertifikat tersedia di Fungsi Koneksi dan header HTTP untuk asal Anda untuk membuat keputusan otorisasi. Gunakan mode opsional saat Anda mendukung klien yang diautentikasi dan tidak diautentikasi.

- Mode passthrough - CloudFront tidak memvalidasi sertifikat klien terhadap toko kepercayaan. CloudFront hanya memvalidasi bahwa klien memiliki kunci pribadi yang sesuai. Ini meneruskan sertifikat ke asal Anda sebagai header HTTP untuk asal Anda untuk melakukan validasi. Tidak diperlukan toko kepercayaan dan tidak ada caching yang terjadi. Gunakan mode passthrough saat Anda memiliki implementasi mTLS yang ada di asal Anda.

Kasus penggunaan

Otentikasi TLS timbal balik dengan CloudFront membahas beberapa skenario keamanan kritis di mana metode otentikasi tradisional tidak mencukupi:

- Otentikasi perangkat dengan caching konten - Anda dapat mengautentikasi konsol game, perangkat IoT, atau perangkat keras perusahaan sebelum mengizinkan akses ke pembaruan firmware, unduhan game, atau sumber daya internal. Setiap perangkat berisi sertifikat unik yang membuktikan keasliannya sambil memanfaatkan kemampuan caching CloudFront.
- API-to-API otentikasi - Anda dapat mengamankan komunikasi mesin-ke-mesin antara mitra bisnis tepercaya, sistem pembayaran, atau layanan mikro. Certificate-based otentikasi menghilangkan kebutuhan akan rahasia bersama atau kunci API sambil memberikan verifikasi identitas yang kuat untuk pertukaran data otomatis.

Topik

- [Toko kepercayaan dan manajemen sertifikat](#)
- [Aktifkan TLS timbal balik untuk distribusi CloudFront](#)
- [Kaitkan Fungsi CloudFront Koneksi](#)
- [Mengkonfigurasi pengaturan tambahan](#)
- [Header mTLS penampil untuk kebijakan cache dan diteruskan ke asal](#)
- [Pencabutan sertifikat](#)
- [Metode pembantu untuk TLS timbal balik](#)
- [Mode validasi tambahan](#)
- [Observabilitas menggunakan log koneksi](#)

Toko kepercayaan dan manajemen sertifikat

Membuat dan mengonfigurasi toko kepercayaan adalah persyaratan wajib untuk menerapkan otentikasi TLS timbal balik dengan CloudFront. CloudFront Trust store berisi sertifikat Certificate Authority (CA) yang CloudFront digunakan untuk memvalidasi sertifikat klien selama proses otentikasi.

Apa itu toko kepercayaan?

Toko kepercayaan adalah repositori sertifikat CA yang CloudFront digunakan untuk memvalidasi sertifikat klien selama otentikasi TLS bersama. Trust store berisi sertifikat CA root dan intermediate yang membentuk rantai kepercayaan untuk mengautentikasi sertifikat klien.

Saat Anda menerapkan TLS timbal balik CloudFront, toko kepercayaan menentukan Otoritas Sertifikat mana yang Anda percayai untuk menerbitkan sertifikat klien yang valid. CloudFront memvalidasi setiap sertifikat klien terhadap toko kepercayaan Anda selama jabat tangan TLS. Hanya klien yang menyajikan sertifikat yang berantai ke salah satu CA di toko kepercayaan Anda yang akan berhasil diautentikasi.

Trust store CloudFront adalah sumber daya tingkat akun yang dapat Anda kaitkan dengan beberapa distribusi. Hal ini memungkinkan Anda untuk mempertahankan kebijakan validasi sertifikat yang konsisten di seluruh CloudFront penerapan Anda sambil menyederhanakan manajemen sertifikat CA.

Dukungan Otoritas Sertifikat

CloudFront mendukung sertifikat yang dikeluarkan oleh Otoritas Sertifikat AWS Pribadi dan Otoritas Sertifikat swasta pihak ketiga. Fleksibilitas ini memungkinkan Anda untuk menggunakan infrastruktur sertifikat yang ada atau memanfaatkan layanan sertifikat AWS terkelola berdasarkan persyaratan organisasi Anda.

- **AWS Otoritas Sertifikat Pribadi:** Anda dapat menggunakan sertifikat yang dikeluarkan oleh AWS Private CA, yang menyediakan layanan otoritas sertifikat swasta yang dikelola. Integrasi ini menyederhanakan manajemen siklus hidup sertifikat dan menyediakan integrasi tanpa batas dengan layanan lain. AWS
- **Third-party Otoritas Sertifikat pribadi:** Anda juga dapat menggunakan sertifikat dari infrastruktur Otoritas Sertifikat pribadi Anda yang ada, termasuk CA perusahaan atau penyedia sertifikat pihak ketiga lainnya. Hal ini memungkinkan Anda untuk mempertahankan proses manajemen sertifikat Anda saat ini sambil menambahkan CloudFront kemampuan mTLS.

Persyaratan dan spesifikasi sertifikat

Toko kepercayaan memiliki persyaratan khusus untuk sertifikat CA yang dikandungnya:

Persyaratan format sertifikat CA

- Format: Format PEM (Privacy Enhanced Mail)
- Batasan konten: Sertifikat harus dilampirkan dalam batas ----- MULAI SERTIFIKAT ----- dan ----- END CERTIFICATE -----
- Komentar: Harus didahului oleh karakter # dan tidak dapat berisi - karakter
- Jeda baris: Tidak ada baris kosong yang diizinkan di antara sertifikat

Spesifikasi sertifikat yang didukung

- Jenis sertifikat: X.509v3
- Jenis kunci publik:
 - RSA 2048, RSA 3072, RSA 4096
 - ECDSA: secp256r1, secp384r1
- Algoritma tanda tangan:
 - SHA256, SHA384, SHA512 dengan RSA
 - SHA256, SHA384, SHA512 dengan EC
 - SHA256, SHA384, SHA512 dengan dengan MGF1 RSASSA-PSS

Contoh format bundel sertifikat

Beberapa sertifikat (PEM-encoded):

```
# Root CA Certificate
-----BEGIN CERTIFICATE-----
MIIDXTCCAkWgAwIBAgIJAKoK/0vD/XqiMA0GCSqGSIb3DQEBCwUAMEUxCzAJBgNV
BAYTAKFVMRMwEQYDVQQIDApTb211LVN0YXR1MSEwHwYDVQQKDBhJbnR1cm5ldCBX
aWRnaXRzIFB0eSBMdGQwHhcNMTCwNzEyMTU0NzQ4WWhcNMjcwNzEwMTU0NzQ4WjBF
MQswCQYDVQQGEwJBVTETMBEGA1UECAwKU29tZS1TdGF0ZTEhMB8GA1UECgwYSW50
ZXJuZXQgV2lkZ2l0cyBqdHkgTHRkMIIBIjANBgkqhkiG9w0BAQEFAA0CAQ8AMIIB
CgKCAQEAAuuExKvY1xzHFylsHiuowqpmzs7rEcuuy10uEszpFp+BtXh0ZuEtts9LP
-----END CERTIFICATE-----
# Intermediate CA Certificate
-----BEGIN CERTIFICATE-----
```

```
MIIDXTCCAkwGAwIBAgIJAKoK/OvD/XqjMA0GCSqGSIb3DQEBCwUAMEUxCzAJBgNV
BAYTAKFVMRMwEQYDVQQIDApTb211LVN0YXR1MSEwHwYDVQQKDBhJbnRlcm5ldCBX
aWRnaXRzIFB0eSBMdGQwHhcNMTcwNzEyMTU0NzQ4WhcNMjcwNzEwMTU0NzQ4WjBF
MQswCQYDVQQGEwJBVETETMBEGA1UECAwKU29tZS1TdGF0ZTEhMB8GA1UECgwYSW50
ZXJuZXQgV21kZ210cyBQdHkgTHRkMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB
CgKCAQEAAuExKvY1xzHFylsHiuowqpmzs7rEcuuy10uEszpFp+BtXh0ZuEtts9LP
-----END CERTIFICATE-----
```

Buat toko kepercayaan

Sebelum membuat toko kepercayaan, Anda harus mengunggah bundel sertifikat CA Anda dalam format PEM ke bucket Amazon S3. Bundel sertifikat harus berisi semua sertifikat CA root dan perantara tepercaya yang diperlukan untuk memvalidasi sertifikat klien Anda.

Bundel sertifikat CA hanya dibaca sekali dari S3 saat membuat toko kepercayaan. Jika perubahan future dilakukan pada bundel sertifikat CA, maka trust store harus diperbarui secara manual. Tidak ada sinkronisasi yang dipertahankan antara toko kepercayaan dan bundel sertifikat S3 CA.

Prasyarat

- Paket sertifikat dari Otoritas Sertifikat (CA) yang diunggah ke bucket Amazon S3
- Izin yang diperlukan untuk membuat sumber daya CloudFront

Untuk membuat toko kepercayaan (Konsol)

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Trust store.
3. Pilih Buat toko kepercayaan.
4. Untuk nama toko Trust, masukkan nama untuk toko kepercayaan Anda.
5. Untuk paket otoritas Sertifikat (CA), masukkan jalur Amazon S3 ke bundel sertifikat PEM-format CA Anda.
6. Pilih Buat toko kepercayaan.

Untuk membuat toko kepercayaan (AWS CLI)

```
aws cloudfront create-trust-store \  
  --name MyTrustStore \  
  --cert-bundle s3://my-bucket/cert-bundle.pem
```

```
--ca-certificates-bundle-source '{"CaCertificatesBundleS3Location":{"Bucket":"my-bucket","Key":"ca-bundle.pem","Region":"bucket-region"}}' \
--tags Items=[{Key=Environment,Value=Production}]
```

Kaitkan toko kepercayaan dengan distribusi

Setelah membuat toko kepercayaan, Anda harus mengaitkannya dengan CloudFront distribusi untuk mengaktifkan otentikasi TLS timbal balik.

Prasyarat

- CloudFront Distribusi yang ada dengan kebijakan protokol HTTPS-only penampil diaktifkan dan dukungan HTTP3 dinonaktifkan.

Untuk mengaitkan toko kepercayaan (Konsol)

Ada dua cara untuk mengaitkan toko kepercayaan di dalam CloudFront konsol: melalui halaman detail toko kepercayaan atau melalui halaman pengaturan distribusi.

Mengaitkan toko kepercayaan melalui halaman detail toko kepercayaan:

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Trust store.
3. Pilih nama toko kepercayaan yang ingin Anda kaitkan.
4. Pilih Associate to Distribution.
5. Konfigurasi opsi mTLS Viewer yang tersedia:
 - Mode validasi sertifikat klien: Pilih antara mode Diperlukan dan Opsional. Dalam mode wajib, semua klien diharuskan untuk menunjukkan sertifikat. Dalam mode opsional, klien yang menyajikan sertifikat divalidasi, sementara klien yang tidak menunjukkan sertifikat diizinkan mengakses.
 - Iklankan nama CA toko kepercayaan: Pilih apakah akan mengiklankan nama CA di toko kepercayaan Anda kepada klien selama jabat tangan TLS.
 - Abaikan tanggal kedaluwarsa sertifikat: Pilih apakah akan mengizinkan koneksi dengan sertifikat kedaluwarsa (kriteria validasi lainnya masih berlaku).
 - Fungsi Koneksi: Fungsi Koneksi opsional dapat dikaitkan dengan allow/deny koneksi berdasarkan kriteria khusus lainnya.

6. Pilih satu atau beberapa distribusi untuk dikaitkan dengan toko kepercayaan. Hanya distribusi dengan HTTP3 yang dinonaktifkan dan dengan perilaku HTTPS-only cache yang dapat mendukung mTL Penampil.
7. Pilih Kaitkan.

Mengaitkan toko kepercayaan melalui halaman pengaturan distribusi:

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Pilih distribusi yang ingin Anda kaitkan
3. Di bawah tab Umum, di dalam wadah Pengaturan, pilih Edit di sudut kanan atas
4. Gulir ke bawah ke bagian bawah halaman, di dalam wadah Konektivitas, aktifkan sakelar Viewer mTLS
5. Konfigurasi opsi mTLS Viewer yang tersedia:
 - Mode validasi sertifikat klien: Pilih antara mode Diperlukan dan Opsional. Dalam mode wajib, semua klien diharuskan untuk menunjukkan sertifikat. Dalam mode opsional, klien yang menyajikan sertifikat divalidasi, sementara klien yang tidak menunjukkan sertifikat diizinkan mengakses.
 - Iklankan nama CA toko kepercayaan: Pilih apakah akan mengiklankan nama CA di toko kepercayaan Anda kepada klien selama jabat tangan TLS.
 - Abaikan tanggal kedaluwarsa sertifikat: Pilih apakah akan mengizinkan koneksi dengan sertifikat kedaluwarsa (kriteria validasi lainnya masih berlaku).
 - Fungsi Koneksi: Fungsi Koneksi opsional dapat dikaitkan dengan allow/deny koneksi berdasarkan kriteria khusus lainnya.
6. Pilih Simpan perubahan di sudut kanan bawah.

Untuk mengaitkan toko kepercayaan (AWS CLI)

Toko kepercayaan dapat dikaitkan dengan distribusi melalui `DistributionConfig.ViewerMtlsConfig` properti. Ini berarti pertama-tama kita perlu mengambil konfigurasi distribusi dan kemudian memberikan permintaan `ViewerMtlsConfig` berikutnya `UpdateDistribution`.

```
// First fetch the distribution
aws cloudfront get-distribution {DISTRIBUTION_ID}
```



```
// Update the distribution config, for example:
Distribution config, file://distConf.json:
{
  ...other fields,
  ViewerMtlsConfig: {
    Mode: 'required',
    TrustStoreConfig: {
      AdvertiseTrustStoreCaNames: false,
      IgnoreCertificateExpiry: true,
      TrustStoreId: {TRUST_STORE_ID}
    }
  }
}

aws cloudfront update-distribution \
  --id {DISTRIBUTION_ID} \
  --if-match {ETAG} \
  --distribution-config file://distConf.json
```

Kelola toko kepercayaan

Lihat detail toko kepercayaan

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Trust store.
3. Pilih nama toko kepercayaan untuk melihat halaman detailnya.

Halaman detail menunjukkan:

- Nama dan ID toko kepercayaan
- Jumlah sertifikat CA
- Tanggal pembuatan dan tanggal modifikasi terakhir
- Distribusi terkait
- Tag

Ubah toko kepercayaan

Untuk mengganti bundel sertifikat CA:

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Trust store.
3. Pilih nama toko kepercayaan.
4. Pilih Tindakan, lalu Edit.
5. Untuk bundel otoritas Sertifikat (CA), masukkan lokasi Amazon S3 dari file PEM bundel CA yang diperbarui.
6. Pilih Perbarui toko kepercayaan.

Hapus toko kepercayaan

Prasyarat: Anda harus terlebih dahulu memisahkan toko kepercayaan dari semua distribusi CloudFront

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Trust store.
3. Pilih nama toko kepercayaan.
4. Pilih Hapus toko kepercayaan.
5. Pilih Hapus untuk mengonfirmasi.

Langkah selanjutnya

Setelah membuat dan mengaitkan toko kepercayaan Anda dengan CloudFront distribusi, Anda dapat melanjutkan untuk mengaktifkan otentikasi TLS timbal balik pada distribusi Anda dan mengonfigurasi pengaturan tambahan seperti meneruskan header sertifikat ke asal Anda. Untuk petunjuk terperinci tentang mengaktifkan MTL pada distribusi Anda, lihat. [Aktifkan TLS timbal balik untuk distribusi CloudFront](#)

Aktifkan TLS timbal balik untuk distribusi CloudFront

Prasyarat dan persyaratan

CloudFrontMode verifikasi TLS timbal balik mengharuskan semua klien untuk menunjukkan sertifikat yang valid selama jabat tangan TLS dan menolak koneksi tanpa sertifikat yang valid. Sebelum mengaktifkan TLS timbal balik pada CloudFront distribusi, pastikan Anda memiliki:

- Membuat toko kepercayaan dengan sertifikat Otoritas Sertifikat Anda
- Mengaitkan toko kepercayaan dengan CloudFront distribusi Anda
- Memastikan semua perilaku cache distribusi menggunakan kebijakan protokol HTTPS-only penampil
- Memastikan distribusi Anda menggunakan HTTP/2 (pengaturan default, mTL Viewer tidak didukung) HTTP/3

Note

Otentikasi TLS timbal balik memerlukan koneksi HTTPS antara pemirsa dan CloudFront. Anda tidak dapat mengaktifkan mTL pada distribusi dengan perilaku cache apa pun yang mendukung koneksi HTTP.

Aktifkan TLS timbal balik (Konsol)

Untuk distribusi baru

MTL penampil tidak dapat dikonfigurasi dalam proses pembuatan distribusi baru di CloudFront konsol. Pertama buat distribusi dengan cara apa pun (konsol, CLI, API), lalu edit pengaturan distribusi untuk mengaktifkan mTL. Penampil sesuai instruksi distribusi yang ada di bawah ini.

Untuk distribusi yang ada

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Dari daftar distribusi, pilih distribusi yang ingin Anda ubah.
3. Pastikan kebijakan protokol Viewer disetel ke Redirect HTTP ke HTTPS atau HTTPS Only untuk semua perilaku cache. (Anda dapat memilih tab Perilaku cache untuk melihat dan memperbarui perilaku cache apa pun dengan kebijakan protokol HTTP.)
4. Pilih tab Umum.
5. Di bagian Pengaturan, pilih Edit.
6. Di bagian Konektivitas, temukan otentikasi bersama Viewer (mTLS).
7. Alihkan Aktifkan otentikasi timbal balik ke Aktif.
8. Untuk mode validasi sertifikat Klien, pilih Diperlukan (semua klien harus menunjukkan sertifikat) atau Opsional (klien dapat menyajikan sertifikat secara opsional).

9. Untuk toko Trust, pilih toko kepercayaan yang Anda buat sebelumnya.
10. (Opsional) Alihkan Iklan nama CA toko kepercayaan jika Anda ingin mengirim nama CA CloudFront ke klien selama jabat tangan TLS.
11. (Opsional) Alihkan Abaikan tanggal kedaluwarsa sertifikat jika Anda ingin mengizinkan koneksi dengan sertifikat kedaluwarsa.
12. Pilih Simpan perubahan.

Aktifkan TLS timbal balik (AWS CLI)

Untuk distribusi baru

Contoh berikut menunjukkan cara membuat file konfigurasi distribusi (distribution-config.json) yang menyertakan pengaturan mTLS:

```
{
  "CallerReference": "cli-example-1",
  "Origins": {
    "Quantity": 1,
    "Items": [
      {
        "Id": "my-origin",
        "DomainName": "example.com",
        "CustomOriginConfig": {
          "HTTPPort": 80,
          "HTTPSPort": 443,
          "OriginProtocolPolicy": "https-only"
        }
      }
    ]
  },
  "DefaultCacheBehavior": {
    "TargetOriginId": "my-origin",
    "ViewerProtocolPolicy": "https-only",
    "MinTTL": 0,
    "ForwardedValues": {
      "QueryString": false,
      "Cookies": {
        "Forward": "none"
      }
    }
  }
},
```

```
"ViewerCertificate": {
  "CloudFrontDefaultCertificate": true
},
"ViewerMtlsConfig": {
  "Mode": "required",
  "TrustStoreConfig": {
    "TrustStoreId": {TRUST_STORE_ID},
    "AdvertiseTrustStoreCaNames": true,
    "IgnoreCertificateExpiry": true
  }
},
"Enabled": true
}
```

Buat distribusi dengan mTL diaktifkan menggunakan perintah contoh berikut:

```
aws cloudfront create-distribution --distribution-config file://distribution-
config.json
```

Untuk distribusi yang ada

Dapatkan konfigurasi distribusi saat ini menggunakan perintah contoh berikut:

```
aws cloudfront get-distribution-config --id E1A2B3C4D5E6F7 --output json > dist-
config.json
```

Edit file untuk menambahkan pengaturan mTLS. Tambahkan bagian contoh berikut ke konfigurasi distribusi Anda:

```
"ViewerMtlsConfig": {
  "Mode": "required",
  "TrustStoreConfig": {
    "TrustStoreId": {TRUST_STORE_ID},
    "AdvertiseTrustStoreCaNames": true,
    "IgnoreCertificateExpiry": true
  }
}
```

Hapus bidang ETag dari file tetapi simpan nilainya secara terpisah.

Perbarui distribusi dengan konfigurasi baru menggunakan perintah contoh berikut:

```
aws cloudfront update-distribution \  
  --id E1A2B3C4D5E6F7 \  
  --if-match YOUR-ETAG-VALUE \  
  --distribution-config file://dist-config.json
```

Kebijakan protokol penampil

Saat menggunakan TLS timbal balik, semua perilaku cache distribusi harus dikonfigurasi dengan kebijakan protokol HTTPS-only penampil:

- Redirect HTTP ke HTTPS - Mengalihkan permintaan HTTP ke HTTPS sebelum melakukan validasi sertifikat.
- Hanya HTTPS - Hanya menerima permintaan HTTPS dan melakukan validasi sertifikat.

Note

Kebijakan protokol penampil HTTP dan HTTPS tidak didukung dengan TLS timbal balik karena koneksi HTTP tidak dapat melakukan validasi sertifikat.

Langkah selanjutnya

Setelah mengaktifkan Viewer TLS pada CloudFront distribusi Anda, Anda dapat mengaitkan Fungsi Koneksi untuk menerapkan logika validasi sertifikat kustom. Fungsi Koneksi memungkinkan Anda untuk memperluas kemampuan otentikasi mTLS bawaan dengan aturan validasi kustom, pemeriksaan pencabutan sertifikat, dan pencatatan. Untuk detail tentang membuat dan mengaitkan Fungsi Koneksi, lihat [Kaitkan Fungsi CloudFront Koneksi](#).

Kaitkan Fungsi CloudFront Koneksi

CloudFront Fungsi Koneksi memungkinkan Anda menerapkan logika validasi sertifikat kustom selama jabat tangan TLS, menyediakan ekstensi ke kemampuan otentikasi mTLS bawaan.

Apa itu Fungsi Koneksi?

Fungsi Koneksi adalah JavaScript fungsi yang berjalan selama jabat tangan TLS setelah sertifikat klien divalidasi. Sertifikat klien yang divalidasi diteruskan ke Fungsi Koneksi di mana Fungsi Koneksi dapat membuat penentuan tambahan apakah akan memberikan akses atau tidak. Untuk informasi rinci tentang Fungsi Koneksi, lihat [Sesuaikan di tepi dengan CloudFront Fungsi](#).

Bagaimana Fungsi Koneksi bekerja dengan mTL

Ketika klien mencoba membuat koneksi mTLS ke CloudFront distribusi Anda, urutan berikut terjadi:

1. Klien memulai jabat tangan TLS dengan CloudFront lokasi tepi.
2. CloudFront meminta dan menerima sertifikat klien.
3. CloudFront melakukan validasi sertifikat standar terhadap trust store.
4. Jika sertifikat melewati validasi standar, CloudFront memanggil Fungsi Koneksi Anda. Jika `IgnoreCertificateExpiry` diaktifkan di dalam `ViewerMtlsConfig`, maka sertifikat Anda yang kedaluwarsa—tetapi valid—juga diteruskan ke Fungsi Sambungan. Jika sertifikat klien tidak valid, Fungsi Koneksi tidak akan dipanggil.
5. Fungsi Koneksi Anda menerima informasi sertifikat dan detail koneksi yang diuraikan.
6. Fungsi Anda membuat allow/deny keputusan berdasarkan logika kustom.
7. CloudFront menyelesaikan atau mengakhiri koneksi TLS berdasarkan keputusan Anda.

Fungsi Koneksi dipanggil untuk mode verifikasi dan mode opsional (saat klien menyajikan sertifikat).

Buat Fungsi Koneksi

Anda dapat membuat Fungsi Koneksi menggunakan CloudFront konsol atau AWS CLI.

Untuk membuat Fungsi Koneksi (Konsol)

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Fungsi.
3. Pilih tab Connection Functions dan pilih Create Connection Function.
4. Masukkan nama fungsi yang unik di dalam AWS akun Anda.
5. Pilih Lanjutkan.
6. Di editor fungsi, tulis JavaScript kode Anda untuk validasi sertifikat. Penangan fungsi harus memanggil `allow` atau `deny`.
7. Opsional: Key/Value toko dapat dikaitkan dengan Fungsi Koneksi untuk mengimplementasikan kontrol pencabutan.
8. Pilih Simpan perubahan.

Untuk membuat Fungsi Koneksi (AWS CLI)

Contoh berikut menunjukkan cara membuat Fungsi Koneksi:

Tulis kode fungsi Anda dalam file terpisah, misalnya `code.js`:

```
function connectionHandler(connection) {  
    connection.allow();  
}
```

```
aws cloudfront create-connection-function \  
  --name "certificate-validator" \  
  --connection-function-config '{  
    "Comment": "Client certificate validation function",  
    "Runtime": "cloudfront-js-2.0"  
  }' \  
  --connection-function-code fileb://code.js
```

Struktur kode Fungsi Koneksi

Fungsi Koneksi mengimplementasikan fungsi `ConnectionHandler` yang menerima objek koneksi yang berisi sertifikat dan informasi koneksi. Fungsi Anda harus menggunakan salah satu `connection.allow()` atau `connection.deny()` untuk membuat keputusan tentang koneksi.

Contoh Fungsi Koneksi Dasar

Contoh berikut menunjukkan Fungsi Koneksi sederhana yang memverifikasi bidang subjek sertifikat klien:

```
function connectionHandler(connection) {  
    // Only process if a certificate was presented  
    if (!connection.clientCertificate) {  
        console.log("No certificate presented");  
        connection.deny();  
    }  
  
    // Check the subject field for specific organization  
    const subject = connection.clientCertificate.certificates.leaf.subject;  
    if (!subject.includes("O=ExampleCorp")) {  
        console.log("Certificate not from authorized organization");  
        connection.deny();  
    } else {
```



```
// All checks passed
console.log("Certificate validation passed");
connection.allow();
}
}
```

Spesifikasi lengkap properti sertifikat klien yang tersedia pada objek koneksi tersedia di sini:

```
{
  "connectionId": "Fdb-Eb7L9gVn2cFakz7wWyBJIDAD4-oN06g8r3vXDV132BtnIVtqDA==", // Unique
  identifier for this TLS connection
  "clientId": "203.0.113.42", // IP address of the connecting client (IPv4 or IPv6)
  "clientCertificate": {
    "certificates": {
      "leaf": {
        "subject": "CN=client.example.com,O=Example Corp,C=US", // Distinguished Name
        (DN) of the certificate holder
        "issuer": "CN=Example Corp Intermediate CA,O=Example Corp,C=US", //
        Distinguished Name (DN) of the certificate authority that issued this certificate
        "serialNumber": "4a:3f:5c:92:d1:e8:7b:6c", // Unique serial number assigned by
        the issuing CA (hexadecimal)
        "validity": {
          "notBefore": "2024-01-15T00:00:00Z", // Certificate validity start date (ISO
          8601 format)
          "notAfter": "2025-01-14T23:59:59Z" // Certificate expiration date (ISO 8601
          format)
        },
        "sha256Fingerprint": "a1b2c3d4e5f6...abc123def456", // SHA-256 hash of the
        certificate (64 hex characters)
      },
    },
  },
}
```

Kaitkan Fungsi Koneksi

Setelah membuat Fungsi Koneksi Anda, Anda harus mempublikasikannya ke tahap LIVE dan mengaitkannya dengan distribusi Anda.

Untuk memublikasikan dan mengaitkan Fungsi Koneksi (Konsol)

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.

2. Di panel navigasi, pilih Fungsi
3. Pilih tab Fungsi Koneksi dan pilih Fungsi Koneksi Anda.
4. Pilih Publikasikan untuk memindahkannya ke panggung LIVE.
5. Pilih Tambahkan asosiasi dalam tabel distribusi terkait di bawah bagian penerbitan.
6. Pilih distribusi dengan mTL Penampil yang diaktifkan yang ingin Anda kaitkan.

Fungsi Koneksi yang dipublikasikan secara alternatif juga dapat dikaitkan dari halaman detail distribusi.

1. Arahkan ke halaman beranda konsol tempat semua distribusi Anda terdaftar.
2. Pilih distribusi yang ingin Anda kaitkan.
3. Pilih tab Umum.
4. Di bagian Pengaturan, pilih Edit.
5. Di bagian Konektivitas, temukan otentikasi bersama Viewer (mTLS).
6. Untuk Fungsi Koneksi, pilih fungsi Anda.
7. Pilih Simpan perubahan.

Untuk mengaitkan Fungsi Koneksi (AWS CLI)

Contoh berikut menunjukkan bagaimana mengaitkan Fungsi Koneksi dengan distribusi:

```
// DistributionConfig:
{
  ...other settings,
  "ConnectionFunctionAssociation": {
    "Id": "cf_30c2CV2e1HwCoInb3LtcaUJkZeD"
  }
}
```

Kasus penggunaan untuk Fungsi Koneksi

Fungsi Koneksi memungkinkan beberapa kasus penggunaan mTL tingkat lanjut:

- Validasi atribut sertifikat - Verifikasi bidang tertentu dalam sertifikat klien seperti persyaratan unit organisasi atau pola nama alternatif subjek.

- Pemeriksaan pencabutan sertifikat - Menerapkan pemeriksaan pencabutan sertifikat khusus menggunakan KeyValueCollection untuk menyimpan nomor seri sertifikat yang dicabut.
- IP-based kebijakan sertifikat - Menerapkan kebijakan sertifikat yang berbeda berdasarkan alamat IP klien atau batasan geografis.
- Multi-tenant validasi - Menerapkan aturan validasi khusus penyewa di mana persyaratan sertifikat yang berbeda berlaku berdasarkan nama host atau atribut sertifikat.

Note

Fungsi Koneksi berjalan sekali per koneksi klien selama jabat tangan TLS.

Fungsi Koneksi hanya dapat mengizinkan atau menolak koneksi, bukan memodifikasi HTTP requests/responses.

Hanya fungsi panggung LIVE (diterbitkan) yang dapat dikaitkan dengan distribusi.

Setiap distribusi dapat memiliki paling banyak satu Fungsi Koneksi.

Langkah selanjutnya

Setelah mengaitkan Fungsi Koneksi dengan CloudFront distribusi Anda, Anda dapat mengonfigurasi pengaturan opsional untuk menyesuaikan perilaku implementasi mTLS Anda. Untuk petunjuk terperinci tentang mengonfigurasi pengaturan tambahan seperti mode validasi sertifikat klien opsional, lihat. [Mengkonfigurasi pengaturan tambahan](#)

Mengkonfigurasi pengaturan tambahan

Setelah mengaktifkan otentikasi TLS timbal balik dasar, Anda dapat mengonfigurasi pengaturan tambahan untuk menyesuaikan perilaku otentikasi untuk kasus dan persyaratan penggunaan tertentu.

Iklan Otoritas Sertifikat

AdvertiseTrustStoreCaNames Bidang mengontrol apakah CloudFront mengirimkan daftar nama CA tepercaya ke klien selama jabat tangan TLS, membantu klien memilih sertifikat yang sesuai.

Untuk mengonfigurasi iklan CA (Konsol)

1. Di pengaturan distribusi Anda, navigasikan ke tab Umum, pilih Edit.

2. Gulir ke bagian Authentication Mutual Authentication (mTLS) Viewer dalam wadah Konektivitas.
3. Pilih atau hapus centang kotak centang Iklankan nama CA toko kepercayaan.
4. Pilih Simpan perubahan.

Untuk mengkonfigurasi iklan CA (AWS CLI)

Contoh berikut menunjukkan cara mengaktifkan iklan CA:

```
"ViewerMtlsConfig": {
  "Mode": "required", // or "optional"
  "TrustStoreConfig": {
    "AdvertiseTrustStoreCaNames": true,
    ...other settings
  }
}
```

Penanganan kedaluwarsa sertifikat

IgnoreCertificateExpiry Properti menentukan bagaimana CloudFront menanggapi sertifikat klien yang kedaluwarsa. Secara default, CloudFront menolak sertifikat klien yang kedaluwarsa, tetapi Anda dapat mengonfigurasinya untuk menerimanya bila perlu. Ini biasanya diaktifkan untuk perangkat dengan sertifikat kedaluwarsa yang tidak dapat diperbarui dengan mudah.

Untuk mengonfigurasi penanganan kedaluwarsa sertifikat (Konsol)

1. Di pengaturan distribusi Anda, navigasikan ke tab Umum, pilih Edit.
2. Gulir ke bagian Viewer mutual authentication (mTLS) dari Connectivity container.
3. Pilih atau batalkan pilihan kotak centang Abaikan tanggal kedaluwarsa sertifikat.
4. Pilih Simpan perubahan.

Untuk mengonfigurasi penanganan kedaluwarsa sertifikat (AWS CLI)

Contoh berikut menunjukkan cara mengabaikan kedaluwarsa sertifikat:

```
"ViewerMtlsConfig": {
  "Mode": "required", // or "optional"
  "TrustStoreConfig": {
    "IgnoreCertificateExpiry": false,
```

```
    ...other settings
  }
}
```

Note

IgnoreCertificateExpiry hanya berlaku untuk tanggal validitas sertifikat. Semua pemeriksaan validasi sertifikat lainnya masih berlaku (rantai kepercayaan, validasi tanda tangan).

Langkah selanjutnya

Setelah mengonfigurasi pengaturan tambahan, Anda dapat mengatur penerusan header untuk meneruskan informasi sertifikat ke asal Anda, menerapkan pencabutan sertifikat menggunakan Fungsi Koneksi dan KeyValueStore, dan mengaktifkan log koneksi untuk pemantauan. Untuk detail tentang meneruskan informasi sertifikat ke asal, lihat [Teruskan Header ke asal](#).

Header mTLS penampil untuk kebijakan cache dan diteruskan ke asal

Saat menggunakan otentikasi TLS timbal balik, CloudFront dapat mengekstrak informasi dari sertifikat klien dan meneruskannya ke asal Anda sebagai header HTTP. Ini memungkinkan server asal Anda untuk mengakses detail sertifikat tanpa menerapkan logika validasi sertifikat.

Header berikut tersedia untuk membuat perilaku cache:

Nama header	Deskripsi	Nilai contoh	Mode yang Didukung
CloudFront-Viewer-Cert-Serial-Number	Representasi heksadesimal dari nomor seri sertifikat	4a: 3f: 5c: 92: d1: e8: 7 b: 6c	Diperlukan, Opsional, Passthrough
CloudFront-Viewer-Cert-Issuer	Representasi string RFC2253 dari nama terhormat penerbit (DN)	cn = Rootcamtl s.com, OU = Rootca, O = MTLS, L = Seattle, ST = Washington, C = AS	Diperlukan, Opsional, Passthrough

Nama header	Deskripsi	Nilai contoh	Mode yang Didukung
CloudFront-Viewer-Cert-Subject	Representasi string RFC2253 dari nama subjek yang dibedakan (DN)	cn=client_.com, OU = klien-3, O = MTLS, ST = Washington, C = AS	Diperlukan, Opsional, Passthrough
CloudFront-Viewer-Cert-Present	Baik 1 (sekarang) atau 0 (tidak ada) yang menunjukkan apakah sertifikat tersebut ada. Nilai ini selalu 1 dalam mode Wajib.	1	Diperlukan, Opsional, Passthrough
CloudFront-Viewer-Cert-Sha256	Hash SHA256 dari sertifikat klien	01bf94fe f55697534 20c349f49 adbfd80af 527537781 6e3ab1fb3 71b29cb586	Diperlukan, Opsional, Passthrough

Untuk permintaan asal, dua header tambahan disediakan, selain header di atas yang tersedia untuk perilaku cache. Karena potensi ukuran header, CloudFront-Viewer-Cert-Pem header tidak terkena fungsi tepi (Lambda @Edge atau CloudFront Functions) dan hanya diteruskan ke asal.

Nama header	Deskripsi	Nilai contoh	Mode yang Didukung
CloudFront-Viewer-Cert-Validity	Format ISO8601 dari tanggal NotBefore dan NotAfter	CloudFront-Viewer-Cert-Validity: NotBefore = 2023-09-21; = 2024-09-20 T01:50:17Z	Diperlukan, Opsional, Passthrough

Nama header	Deskripsi	Nilai contoh	Mode yang Didukung
		NotAfter T01:50:17 Z	
CloudFront-Viewer-Cert-Pem	URL-encoded Format PEM dari sertifikat daun	CloudFront-Viewer-Cert-Pem: ----- MULAI% 20 SERTIFIKAT ----- %0AMIIG<.. .. dikurangi... > NmrUlw %0A ----- AKHIR%20S ERTIFIKAT ----- %0A	Diperlukan, Opsional
Client-Cert	Sertifikat daun	Client-Cert:: MIIG<... berkurang ... >NmrUlw:	Passthrough
Client-Cert-Chain	Non-leaf rantai sertifikat	Client-Cert-Chain:: MIIG<... berkurang ... >NmrUlw:	Passthrough

Konfigurasi penerusan header

Konsol

Dalam mode verifikasi, CloudFront secara otomatis menambahkan header CloudFront-Viewer-Cert -* ke semua permintaan penampil. Untuk meneruskan header ini ke asal Anda:

1. Dari halaman distribusi daftar utama, pilih distribusi Anda dengan mTLS penampil diaktifkan dan buka tab Perilaku
2. Pilih perilaku cache dan pilih Edit
3. Di bagian Kebijakan permintaan asal, pilih Buat kebijakan atau pilih kebijakan yang ada
4. Pastikan header berikut disertakan dalam kebijakan permintaan asal:

- CloudFront-Viewer-Cert-Serial-Number
 - CloudFront-Viewer-Cert-Issuer
 - CloudFront-Viewer-Cert-Subject
 - CloudFront-Viewer-Cert-Present
 - Cloudfront-Viewer-Cert-Sha256
 - CloudFront-Viewer-Cert-Validity
 - CloudFront-Viewer-Cert-Pem
5. Pilih Buat (untuk kebijakan baru) atau Simpan perubahan (untuk kebijakan yang ada)
 6. Pilih kebijakan dalam perilaku cache Anda dan simpan perubahan

Penggunaan AWS CLI

Contoh berikut menunjukkan cara membuat kebijakan permintaan asal yang menyertakan header mTLS untuk mode verifikasi:

```
aws cloudfront create-origin-request-policy \  
--origin-request-policy-config '{  
  "Name": "MTLSHeadersPolicy",  
  "HeadersConfig": {  
    "HeaderBehavior": "whitelist",  
    "Headers": {  
      "Quantity": 5,  
      "Items": [  
        "CloudFront-Viewer-Cert-Serial-Number",  
        "CloudFront-Viewer-Cert-Issuer",  
        "CloudFront-Viewer-Cert-Subject",  
        "CloudFront-Viewer-Cert-Validity",  
        "CloudFront-Viewer-Cert-Pem"  
      ]  
    }  
  },  
  "CookiesConfig": {  
    "CookieBehavior": "none"  
  },  
  "QueryStringConfig": {  
    "QueryStringBehavior": "none"  
  }  
}'
```


Pertimbangan pemrosesan header

Saat bekerja dengan header sertifikat, pertimbangkan praktik terbaik ini:

- Validasi header: Verifikasi nilai header sertifikat di asal Anda sebagai langkah keamanan tambahan
- Batas ukuran header: Header sertifikat PEM bisa besar, pastikan server asal Anda dapat menanganinya
- Pertimbangan cache: Menggunakan header sertifikat di kunci cache Anda meningkatkan fragmentasi cache
- Cross-origin permintaan: Jika aplikasi Anda menggunakan CORS, maka Anda mungkin perlu mengonfigurasinya untuk mengizinkan header sertifikat

Langkah selanjutnya

Setelah mengonfigurasi penerusan header, Anda dapat menerapkan pemeriksaan pencabutan sertifikat menggunakan Fungsi Koneksi dan CloudFront KeyValueCollection. Untuk detail tentang penerapan pemeriksaan pencabutan, lihat [Pencabutan sertifikat](#)

Pencabutan sertifikat

Ketika sertifikat perlu dicabut — karena kompromi, perubahan kebijakan, atau hubungan yang dihentikan — Anda memerlukan mekanisme untuk menolak sertifikat tersebut selama jabat tangan mTLS. CloudFront menyediakan dua pendekatan asli untuk pencabutan sertifikat, dan Anda dapat menggabungkannya untuk kontrol berlapis.

- OCSP (Online Certificate Status Protocol) — CloudFront menanyakan responden OCSP Otoritas Sertifikat Anda secara real time untuk memeriksa apakah sertifikat klien telah dicabut. Aktifkan OCSP di toko kepercayaan Anda dan CloudFront tangani validasi secara otomatis selama jabat tangan TLS. Hasil OCSP juga terlihat di Fungsi Koneksi, memberi Anda akses terprogram ke status pencabutan untuk pengambilan keputusan khusus.
- CloudFront Fungsi dan KeyValueCollection — Anda menyimpan daftar nomor seri sertifikat yang dicabut di file. CloudFront KeyValueCollection Fungsi Koneksi menanyakan KeyValueCollection selama jabat tangan TLS dan memungkinkan atau menyangkal koneksi. Ini memberi Anda kontrol penuh atas data pencabutan, waktu pembaruan, dan logika khusus seperti masa tenggang atau IP-based pengecualian.

Perbandingan: OCSP vs Fungsi CloudFront dengan KeyValueTypeStore

	OCSP	CloudFront Fungsi + KeyValueTypeStore
Sumber data	Responden OCSP Otoritas Sertifikat	Anda mengelola daftar pencabutan
Mekanisme pembaruan	Real-time query ke CA	Anda mendorong pembaruan ke KeyValueTypeStore
Logika khusus	Tersedia melalui Fungsi Koneksi	Dibangun ke dalam kode fungsi Anda
Ketergantungan eksternal	Membutuhkan ketersediaan responden CA OCSP	Tidak ada ketergantungan eksternal
Terbaik untuk	CA yang mempertahankan responden OCSP; status waktu nyata CA-authoritative	Self-managed pencabutan; kebijakan khusus; CA tanpa dukungan OCSP

Anda dapat menggunakan kedua pendekatan bersama-sama. Aktifkan OCSP untuk pemeriksaan CA-authoritative pencabutan, lalu gunakan Fungsi Koneksi untuk melapisi logika tambahan di atas hasil OCSP — misalnya, mengizinkan sertifikat yang dicabut dari rentang IP tepercaya selama masa tenggang.

OCSP (Protokol Status Sertifikat Online)

OCSP adalah protokol real-time yang memeriksa status pencabutan sertifikat secara langsung dengan Certificate Authority (CA). Selama proses penandatanganan sertifikat, CA menyematkan URL responden OCSP dalam sertifikat. Ketika klien menunjukkan sertifikat mereka selama jabat tangan mTLS, CloudFront mengirimkan permintaan OCSP ke URL responder yang disematkan dan bertindak berdasarkan respons: sertifikat yang valid dilanjutkan, sertifikat yang dicabut dihentikan.

CloudFront memvalidasi seluruh rantai sertifikat — sertifikat daun dan hingga tiga sertifikat perantara — masing-masing terhadap URL responden OCSP masing-masing. Sertifikat root di trust store dikecualikan dari validasi OCSP.

Aktifkan OCSP

Aktifkan OCSP di toko kepercayaan Anda. Ketika diaktifkan, CloudFront secara otomatis melakukan validasi OCSP untuk setiap sertifikat klien yang berisi URL responder OCSP dalam ekstensi Authority Information Access (AIA). Setelah OCSP diaktifkan, seluruh rantai sertifikat klien harus memiliki URL


```

    "chainValidity": "Valid",           // "Valid" | "Invalid" | "Unknown"
    "certificates": {
      "leaf": {
        "method": "OCSP",              // "OCSP"
        "status": "Good",              // "Good" | "Revoked" | "Unknown" | "Error"
        "serialNumber": "00:a7:30:9e:73:7b:3e:63:bd:b7:c0:7e:bf:d5:c9:86"
      },
      "intermediates": [
        {
          "method": "OCSP",            // "OCSP"
          "status": "Error",           // "Good" | "Revoked" | "Unknown" | "Error"
          "errorType": "InternalError", // "InternalError" | "OCSP response
verification failed: {Type}"
          "serialNumber": "00:a7:30:9e:73:7b:3e:63:bd:b7:c0:7e:bf:d5:c9:86"
        }
      ]
    }
  },
  "clientId": "127.0.0.1",
  "endpoint": "d123.cloudfront.net",
  "distributionId": "E1NXS4MQZH501R",
  "connectionId": "xdzQ6lJUDUt8b70uqOD8lmzOC9HcMaXPmhH5ZdzLCZpKxqzfCPpR4A=="
}

```

chainValidityBidang bisaValid,Invalid, atauUnknown. statusNilai sertifikat individu dapat berupaGood,Revoked,Unknown, atauError. errorTypeBidang berisi InternalError atau OCSP response verification failed: {Type} kapan statusnyaError.

Fungsi Koneksi dapat mengganti hasil OCSP - misalnya, mengizinkan sertifikat yang dicabut dari rentang IP terpercaya, atau menolak sertifikat yang dilaporkan OCSP sebagai “baik” berdasarkan logika bisnis tambahan.

Note

errorTypeBidang hanya akan hadir jika statusnyaError.

Contoh: Penanganan OCSP khusus dengan pengecualian IP terpercaya

```

function connectionHandler(connection) {
  var revocationStatus = connection.clientCertificate.revocationStatus;

```

```

var trustedIP = (connection.clientIp === "[IP_ADDRESS]");

if (revocationStatus.chainValidity === "Invalid") {
    if (trustedIP) {
        connection.allow();
    } else {
        connection.deny();
    }
} else if (revocationStatus.certificates.leaf.status === "Error") {
    console.log(revocationStatus.certificates.leaf.errorType);
    connection.deny();
} else {
    connection.allow();
}
}

```

Perilaku kegagalan OCSP

Ketika CloudFront tidak dapat menentukan status pencabutan sertifikat — karena responden OCSP tidak dapat dijangkau, mengembalikan kesalahan, atau mengembalikan “tidak dikenal” — menolak koneksi secara default. CloudFront Untuk mengimplementasikan perilaku soft-fail, gunakan Fungsi Koneksi. Fungsi Koneksi tidak akan dijalankan jika OCSP diaktifkan dan URL OCSP tidak ada dalam sertifikat klien. Hasil OCSP tersedia di objek koneksi, dan fungsi Anda dapat mengizinkan koneksi ketika status OCSP tidak ditentukan:

```

async function connectionHandler(connection) {
    var revocationStatus = connection.clientCertificate.revocationStatus;

    if (revocationStatus.certificates.leaf.status === "Error" ||
        revocationStatus.certificates.leaf.status === "Unknown") {
        // OCSP responder unreachable – allow connection (soft-fail)
        connection.logCustomData(`OCSP_SOFT_FAIL:
${revocationStatus.certificates.leaf.errorType}`);
        return connection.allow();
    }

    if (revocationStatus.chainValidity === "Invalid") {
        return connection.deny();
    }

    return connection.allow();
}

```

Pencabutan sertifikat dengan CloudFront Fungsi dan KeyValueCollection

Anda dapat menggunakan Fungsi CloudFront Koneksi dengan KeyValueCollection untuk mengimplementasikan pemeriksaan pencabutan sertifikat tanpa ketergantungan eksternal. Anda menyimpan daftar nomor seri sertifikat yang dicabut di KeyValueCollection, dan Fungsi Koneksi Anda memeriksa setiap sertifikat klien terhadap daftar ini selama jabat tangan TLS.

Proses pencabutan sertifikat berfungsi sebagai berikut:

1. Simpan nomor seri sertifikat yang telah dicabut di file. CloudFront KeyValueCollection
2. Ketika klien menyajikan sertifikat, Fungsi Koneksi Anda dipanggil.
3. Fungsi memeriksa nomor seri sertifikat terhadap KeyValueCollection.
4. Jika nomor seri ditemukan di toko, sertifikat dicabut.
5. Fungsi Anda menolak koneksi untuk sertifikat yang dicabut.

Pendekatan ini memberikan pengecekan pencabutan hampir real-time di seluruh jaringan edge global CloudFront.

Untuk menerapkan pendekatan ini, Anda perlu:

- Distribusi yang dikonfigurasi dengan mTLS penampil
- Sebuah KeyValueCollection berisi nomor seri sertifikat yang dicabut
- Fungsi Koneksi yang menanyakan status sertifikat KeyValueCollection untuk memeriksa

Saat klien terhubung, CloudFront validasi sertifikat terhadap trust store, lalu jalankan Fungsi Koneksi Anda. Fungsi Anda memeriksa nomor seri sertifikat terhadap KeyValueCollection dan mengizinkan atau menolak koneksi.

Langkah 1: Buat sertifikat KeyValueCollection yang dicabut

Siapkan nomor seri sertifikat yang telah dicabut dalam format JSON:

```
{
  "data": [
    { "key": "ABC123DEF456", "value": "" },
    { "key": "789XYZ012GHI", "value": "" }
  ]
}
```

```
}
```

Unggah file JSON ini ke bucket S3, lalu buat: KeyValueStore

```
aws s3 cp revoked-serials.json s3://your-bucket-name/revoked-serials.json

aws cloudfront create-key-value-store \
  --name revoked-serials-kvs \
  --import-source '{
    "SourceType": "S3",
    "SourceARN": "arn:aws:s3:::your-bucket-name/revoked-serials.json"
  }'
```

Tunggu KeyValueStore hingga penyediaan selesai. Periksa statusnya dengan:

```
aws cloudfront get-key-value-store --name "revoked-serials-kvs"
```

Langkah 2: Buat Fungsi Koneksi Pencabutan

Buat Fungsi Koneksi yang memeriksa nomor seri sertifikat terhadap KeyValueStore:

```
aws cloudfront create-connection-function \
  --name "revocation-control" \
  --connection-function-config file://connection-function-config.json \
  --connection-function-code file://connection-function-code.txt
```

File konfigurasi menentukan KeyValueStore asosiasi:

```
{
  "Runtime": "cloudfront-js-2.0",
  "Comment": "A function that implements revocation control via KVS",
  "KeyValueStoreAssociations": {
    "Quantity": 1,
    "Items": [
      {
        "KeyValueStoreArn": "arn:aws:cloudfront::account-id:key-value-store/kvs-id"
      }
    ]
  }
}
```

Contoh kode Fungsi Koneksi:

```
import cf from 'cloudfront';

async function connectionHandler(connection) {
  const kvsHandle = cf.kvs();

  // Get client serial number from client certificate
  const clientSerialNumber =
    connection.clientCertificate.certificates[0].serialNumber;

  // Check KVS to see if serial number exists as a key
  // Remove : from the clientSerialNumber if KVS entries dont have it
  const serialNumberExistsInKvs =
    await kvsHandle.exists(clientSerialNumber.replaceAll(":", ""));

  // Deny connection if serial number exists in KVS
  if (serialNumberExistsInKvs) {
    console.log("Connection denied – certificate revoked");
    connection.logCustomData("Connection denied – certificate revoked");
    return connection.deny();
  }

  // Allow connections that don't exist in KVS
  console.log("Connection allowed");
  return connection.allow();
}
```

Langkah 3: Uji fungsi pencabutan Anda

Gunakan CloudFront konsol untuk menguji Fungsi Koneksi Anda dengan sertifikat sampel. Arahkan ke Fungsi Koneksi di konsol dan gunakan tab Uji.

- Tempelkan sertifikat sampel dalam format PEM ke antarmuka pengujian.
- Secara opsional menentukan alamat IP klien untuk IP-based logika pengujian.
- Pilih fungsi Uji untuk melihat hasil eksekusi.
- Tinjau log eksekusi untuk memverifikasi logika fungsi Anda.

Uji dengan sertifikat yang valid dan dicabut untuk memastikan fungsi Anda menangani kedua skenario dengan benar.

Langkah 4: Kaitkan fungsi ke distribusi Anda

Setelah Anda mempublikasikan Fungsi Koneksi, kaitkan dengan TLS-enabled distribusi m Anda untuk mengaktifkan pemeriksaan pencabutan sertifikat. Arahkan ke pengaturan distribusi Anda, gulir ke bagian “Viewer mutual authentication (mTLS)”, pilih Fungsi Koneksi Anda, dan simpan perubahannya.

Strategi pencabutan lanjutan

Gabungkan OCSP dengan logika Fungsi Koneksi

Anda dapat mengaktifkan OCSP untuk pemeriksaan CA-authoritative pencabutan dan melapisi Fungsi Koneksi di atas untuk kebijakan khusus. Fungsi Koneksi menerima hasil OCSP dan dapat menerapkan logika tambahan:

- Masa tenggang - Izinkan sertifikat yang dicabut dari jaringan internal untuk jangka waktu tertentu selama rotasi sertifikat.
- Akses darurat — Izinkan koneksi dari IP tertentu bahkan ketika OCSP melaporkan status dicabut.
- Logika penolakan khusus - Blokir sertifikat yang dilaporkan OCSP sebagai “baik” berdasarkan data pencabutan Anda sendiri. KeyValueCollection

Example Pencabutan gabungan OCSP+KVS

```
import cf from 'cloudfront';

async function connectionHandler(connection) {
  var kvsHandle = cf.kvs();
  var revocationStatus = connection.clientCertificate.revocationStatus;
  var serialNumber = connection.clientCertificate.certificates.leaf.serialNumber;

  // Check your own revocation list first (immediate revocation, no cache delay)
  var inKvs = await kvsHandle.exists(serialNumber.replaceAll(":", ""));
  if (inKvs) {
    connection.logCustomData("KVS_REVOKED:" + serialNumber);
    return connection.deny();
  }

  // Then check OCSP result
  if (revocationStatus.chainValidity === 'Valid' &&
    revocationStatus.certificates.leaf.status === "Revoked") {
```

```
// OCSP says revoked – allow grace period from trusted IPs
if (connection.clientIp.startsWith("10.0.")) {
    connection.logCustomData("GRACE_PERIOD:" + serialNumber + ":" +
connection.clientIp);
    return connection.allow();
}
connection.logCustomData("OCSP_REVOKED:" + serialNumber);
return connection.deny();
}

connection.allow();
}
```

Pola ini memberi Anda pencabutan langsung melalui KVS (tanpa penundaan cache) ditambah CA-authoritative pencabutan melalui OCSP, dengan penanganan pengecualian khusus di antaranya.

Metode pembantu untuk TLS timbal balik

Gunakan fungsi `cf.mtls` pembantu dalam fungsi permintaan penampil untuk mengganti nama, memformat ulang, atau menggabungkan header sertifikat klien sebelum meneruskan permintaan ke asal Anda. Lihat perinciannya di [the section called “Metode pembantu untuk TLS timbal balik”](#).

Mode validasi tambahan

Mode yang diperlukan adalah default untuk TLS CloudFront timbal balik - CloudFront memvalidasi setiap sertifikat klien dan menyangkal koneksi yang gagal. Namun, untuk mendukung kasus penggunaan tambahan - seperti aplikasi yang melayani campuran m TLS-authenticated dan klien yang tidak diautentikasi, atau lingkungan di mana server asal melakukan validasi mTLS sendiri - CloudFront mendukung dua mode tambahan.

Mode opsional dirancang untuk aplikasi dengan populasi klien campuran. Misalnya, portal yang menyajikan konten publik ke browser yang tidak diautentikasi sambil membatasi titik akhir API ke klien yang menampilkan sertifikat yang valid. Atau skenario migrasi di mana Anda secara bertahap mengarahkan klien ke mTL dan perlu mengizinkan koneksi yang diautentikasi dan tidak diautentikasi selama transisi.

Mode passthrough dirancang untuk pelanggan yang server asalnya sudah melakukan validasi mTLS. Misalnya, layanan yang bermigrasi dari proxy terbalik lokal, CDN lain, atau Application Load Balancer yang sudah menangani verifikasi sertifikat. Mode passthrough memungkinkan pelanggan ini untuk menempatkan CloudFront di depan aplikasi mereka tanpa menerapkan kembali logika validasi di tepi.

Mode opsional

Validasi sertifikat klien Mode opsional

CloudFront menawarkan alternatif mode validasi sertifikat klien opsional yang memvalidasi sertifikat klien yang disajikan tetapi memungkinkan akses ke klien yang tidak menunjukkan sertifikat.

Perilaku mode opsional

- Memberikan koneksi ke klien dengan sertifikat yang valid (sertifikat tidak valid ditolak).
- Memungkinkan koneksi ke klien tanpa sertifikat.
- Memungkinkan skenario otentikasi klien campuran melalui distribusi tunggal.

Mode opsional sangat ideal untuk migrasi bertahap ke otentikasi mTLS, mendukung klien dengan sertifikat dan klien tanpa sertifikat, atau mempertahankan kompatibilitas mundur dengan klien lama.

Note

Dalam mode opsional, Fungsi Koneksi masih dipanggil bahkan ketika klien tidak menunjukkan sertifikat. Hal ini memungkinkan Anda untuk menerapkan logika kustom seperti mencatat alamat IP klien atau menerapkan kebijakan yang berbeda berdasarkan apakah sertifikat disajikan.

Untuk mengkonfigurasi mode opsional (Konsol)

1. Di pengaturan distribusi Anda, navigasikan ke tab Umum, pilih Edit.
2. Gulir ke bagian otentikasi bersama Viewer (mTLS) dalam wadah Konektivitas.
3. Untuk mode validasi sertifikat Klien, pilih Opsional.
4. Simpan perubahan.

Untuk mengkonfigurasi mode opsional (AWS CLI)

Contoh berikut menunjukkan cara mengkonfigurasi mode opsional:

```
"ViewerMtlsConfig": {  
  "Mode": "optional",
```

```
...other settings  
}
```

Sesuaikan header sertifikat

Gunakan fungsi pembantu mTLS di CloudFront Fungsi permintaan pemirsa untuk mengganti nama, memformat ulang, atau menggabungkan header sertifikat sebelum mencapai asal Anda. Ini berguna saat bermigrasi dari layanan lain yang menggunakan nama header atau format pengkodean sertifikat yang berbeda.

Mode passthrough

Mode passthrough memungkinkan pelanggan dengan implementasi mTL yang ada di asal mereka untuk digunakan. CloudFront mengakhiri koneksi TLS dan meneruskan sertifikat klien ke asal Anda sebagai header HTTP. Asal Anda melakukan semua validasi sertifikat — termasuk verifikasi rantai, pemeriksaan pencabutan, dan penegakan kebijakan khusus.

Cara kerja mode passthrough

1. Klien terhubung ke CloudFront dan menyajikan sertifikat klien selama jabat tangan TLS.
2. CloudFront menyelesaikan jabat tangan TLS tanpa memvalidasi sertifikat terhadap toko kepercayaan.
3. CloudFront menambahkan sertifikat klien dan rantai sertifikat sebagai header HTTP pada permintaan.
4. Permintaan diteruskan ke asal Anda. Tidak ada konten yang di-cache.
5. Asal Anda memvalidasi sertifikat dan memproses permintaan.

Klien juga dapat terhubung tanpa menunjukkan sertifikat. Fungsi asal atau Koneksi Anda menangani skenario sertifikat kosong.

Note

Dalam mode passthrough, Fungsi Koneksi masih dipanggil bahkan ketika klien tidak menunjukkan sertifikat. Hal ini memungkinkan Anda untuk menerapkan logika kustom seperti mencatat alamat IP klien atau menerapkan kebijakan yang berbeda berdasarkan apakah sertifikat disajikan.

Persyaratan konfigurasi

- Tidak ada toko kepercayaan - Distribusi harus memiliki asosiasi toko tanpa kepercayaan.
- Caching dinonaktifkan - Semua perilaku cache harus menggunakan kebijakan `CachingDisabled` cache terkelola.
- Origin Shield dilarang — Origin Shield tidak dapat diaktifkan.
- Lambda @Edge dilarang — Asosiasi fungsi Lambda @Edge tidak diizinkan.
- Kebijakan permintaan asal diperlukan — Anda harus mengizinkan daftar `Client-Cert` dan `Client-Cert-Chain` header dalam kebijakan permintaan asal Anda agar asal dapat menerimanya.
- Certificate-chain-depth— CloudFront memungkinkan kedalaman maksimum 4 untuk rantai sertifikat klien untuk diteruskan ke asal.

Aktifkan mode passthrough

Konsol

1. Perbarui semua perilaku cache untuk menggunakan kebijakan `CachingDisabled` cache terkelola.
2. Buka CloudFront konsol dan pilih distribusi Anda.
3. Pilih tab Umum, lalu pilih Edit di bawah Pengaturan.
4. Di bawah Authentication mutual Viewer (mTLS), pilih Passthrough.
5. Pilih Simpan perubahan.

AWS CLI

Pastikan semua perilaku cache mereferensikan `CachingDisabled` kebijakan terkelola, lalu perbarui konfigurasi distribusi:

```
{
  "ViewerMtlsConfig": {
    "Mode": "passthrough"
  }
}
```

Header diteruskan ke asal

CloudFront menambahkan header berikut ke permintaan yang dikirim ke asal Anda:

- `Client-Cert`—Sertifikat entitas akhir (daun) yang disajikan oleh klien, dikodekan base64.
- `Client-Cert-Chain`—Rantai sertifikat (tidak termasuk daun), sebagai daftar bidang terstruktur. Setiap sertifikat dikodekan base64. `Client-Cert-Chain` adalah header tipe List. Ini mungkin muncul beberapa kali dalam permintaan. Menggabungkan semua nilai mempertahankan urutan rantai asli. `Client-Cert-Chain` dihilangkan ketika klien hanya menyajikan satu sertifikat.

CloudFront menjatuhkan semua yang masuk `Client-Cert` atau `Client-Cert-Chain` header dari permintaan klien sebelum menambahkan data sertifikat yang sebenarnya. Ini mencegah spoofing header.

Sesuaikan header sertifikat

Gunakan fungsi pembantu mTLS di CloudFront Fungsi permintaan pemirsa untuk mengganti nama, memformat ulang, atau menggabungkan header sertifikat sebelum mencapai asal Anda. Ini berguna saat bermigrasi dari layanan lain yang menggunakan nama header atau format pengkodean sertifikat yang berbeda.

Observabilitas menggunakan log koneksi

CloudFront log koneksi memberikan visibilitas terperinci ke dalam peristiwa otentikasi TLS timbal balik, memungkinkan Anda memantau validasi sertifikat, melacak upaya koneksi, dan memecahkan masalah otentikasi.

Apa itu log koneksi?

Log koneksi menangkap informasi terperinci tentang jabat tangan TLS dan validasi sertifikat untuk distribusi bersama. TLS-enabled Tidak seperti log akses standar yang merekam informasi permintaan HTTP, log koneksi berfokus secara khusus pada fase pembentukan koneksi TLS, termasuk:

- Status koneksi (success/failure)
- Detail sertifikat klien
- Protokol TLS dan informasi sandi
- Metrik waktu koneksi

- Data kustom dari Fungsi Koneksi

Log ini memberikan visibilitas komprehensif ke dalam peristiwa otentikasi berbasis sertifikat, membantu Anda memantau keamanan, memecahkan masalah, dan memenuhi persyaratan kepatuhan.

Aktifkan log koneksi

Log koneksi hanya tersedia untuk distribusi dengan otentikasi TLS timbal balik diaktifkan. Anda dapat mengirim log koneksi ke beberapa tujuan termasuk CloudWatch Log, Amazon Data Firehose, dan Amazon S3.

Prasyarat

Sebelum mengaktifkan log koneksi:

- Konfigurasi TLS timbal balik untuk distribusi Anda CloudFront
- Aktifkan log koneksi untuk CloudFront distribusi Anda
- Pastikan Anda memiliki izin yang diperlukan untuk tujuan pencatatan yang Anda pilih
- Untuk pengiriman lintas akun, konfigurasi kebijakan IAM yang sesuai

Untuk mengaktifkan log koneksi (Konsol)

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Dari daftar distribusi, pilih TLS-enabled distribusi m Anda.
3. Pilih tab Logging.
4. Pilih Tambahkan.
5. Pilih layanan untuk menerima log Anda:
 - CloudWatch Log
 - Firehose
 - Amazon S3
6. Untuk Tujuan, pilih sumber daya untuk layanan yang Anda pilih:
 - Untuk CloudWatch Log, masukkan nama grup Log
 - Untuk Firehose, pilih aliran pengiriman Firehose

- Untuk Amazon S3, masukkan nama Bucket (opsional dengan awalan)
7. (Opsional) Konfigurasi pengaturan tambahan:
 - Pemilihan bidang: Pilih bidang log tertentu untuk disertakan.
 - Format keluaran: Pilih dari JSON, Plain, w3c, Raw, atau Parquet (hanya S3).
 - Pembatas bidang: Tentukan cara memisahkan bidang log.
 8. Pilih Save changes (Simpan perubahan)

Untuk mengaktifkan log koneksi (AWS CLI)

Contoh berikut menunjukkan cara mengaktifkan log koneksi menggunakan CloudWatch API:

```
# Step 1: Create a delivery source
aws logs put-delivery-source \
  --name "cf-mtls-connection-logs" \
  --resource-arn "arn:aws:cloudfront::123456789012:distribution/E1A2B3C4D5E6F7" \
  --log-type CONNECTION_LOGS

# Step 2: Create a delivery destination
aws logs put-delivery-destination \
  --name "s3-destination" \
  --delivery-destination-configuration \
    "destinationResourceArn=arn:aws:s3:::amzn-s3-demo-bucket1"

# Step 3: Create the delivery
aws logs create-delivery \
  --delivery-source-name "cf-mtls-connection-logs" \
  --delivery-destination-arn "arn:aws:logs:us-east-1:123456789012:delivery-destination:s3-destination"
```

Note

Saat menggunakan CloudWatch API, Anda harus menentukan Wilayah AS Timur (Virginia N.) (us-east-1) bahkan saat mengirimkan log ke wilayah lain.

Bidang log koneksi

Log koneksi mencakup informasi terperinci tentang setiap upaya koneksi TLS:

Bidang	Deskripsi	Contoh
eventTimestamp	Stempel waktu ISO 8601 saat koneksi dibuat atau gagal	1731620046814
connectionId	Pengidentifikasi unik untuk koneksi TLS	oLHiEKbQSn8lkvJfA3 D4gFowK3_iZ0g4i5nM UjE1Akod8TuAzn5nzg==
connectionStatus	Status upaya koneksi mTLS.	Success atau Failed
clientIp	Alamat IP dari klien penghubung	2001:0db8:85a3:000 0:0000:8a2e:0370:7334
clientPort	Port yang digunakan oleh klien	12137
serverIp	Alamat IP server CloudFront tepi	99.84.71.136
distributionId	CloudFront ID distribusi	E2DX1SLDPK0123
distributionTenantId	CloudFront ID penyewa distribusi (bila berlaku)	dt_2te1Ura9X3R2iCGNjW123
tlsProtocol	Versi protokol TLS digunakan	TLSv1.3
tlsCipher	TLS cipher suite digunakan untuk koneksi	TLS_AES_128_GCM_SHA256
tlsHandshakeDuration	Durasi jabat tangan TLS dalam milidetik	153
tlsSni	Nilai Indikasi Nama Server dari jabat tangan TLS	d111111abcdef8.cloudfront.net

Bidang	Deskripsi	Contoh
<code>clientLeafCertSerialNumber</code>	Nomor seri sertifikat klien	<code>00:b1:43:ed:93:d2:d8:f3:9d</code>
<code>clientLeafCertSubject</code>	Bidang subjek sertifikat klien	<code>C=US, ST=WA, L=Seattle, O=Amazon.com, OU=CloudFront, CN=client.test.mtls.net</code>
<code>clientLeafCertIssuer</code>	Bidang penerbit sertifikat klien	<code>C=US, ST=WA, L=Seattle, O=Amazon.com, OU=CloudFront, CN=test.mtls.net</code>
<code>clientLeafCertValidity</code>	Masa berlaku sertifikat klien	<code>NotBefore=2025-06-05T23:28:21Z;NotAfter=2125-05-12T23:28:21Z</code>
<code>connectionLogCustomData</code>	Data kustom ditambahkan melalui Fungsi Koneksi	<code>REVOKED:00:b1:43:ed:93:d2:d8:f3:9d</code>

Kode kesalahan koneksi

```

Failed:ClientCertMaxChainDepthExceeded
Failed:ClientCertMaxSizeExceeded
Failed:ClientCertUntrusted
Failed:ClientCertNotYetValid
Failed:ClientCertExpired
Failed:ClientCertTypeUnsupported
Failed:ClientCertInvalid
Failed:ClientCertIntentInvalid
Failed:ClientCertRejected
Failed:ClientCertMissing
Failed:ClientCertOcspProcessingError
Failed:ClientCertOcspDenied
Failed:TcpError
Failed:TcpTimeout

```

```
Failed:ConnectionFunctionError
Failed:ConnectionFunctionDenied
Failed:Internal
Failed:UnmappedConnectionError
```

Ketika koneksi gagal, CloudFront mencatat kode alasan tertentu:

Kode	Deskripsi
ClientCertMaxChainDepthExceeded	Kedalaman rantai sertifikat maksimum terlampaui
ClientCertMaxSizeExceeded	Ukuran sertifikat maksimum terlampaui
ClientCertUntrusted	Sertifikat tidak dipercaya
ClientCertNotYetValid	Sertifikat belum valid
ClientCertExpired	Sertifikat kedaluwarsa
ClientCertTypeUnsupported	Jenis sertifikat tidak didukung
ClientCertInvalid	Sertifikat tidak valid
ClientCertIntentInvalid	Maksud sertifikat tidak valid
ClientCertRejected	Sertifikat ditolak oleh validasi kustom
ClientCertMissing	Sertifikat tidak ada
TcpError	Terjadi kesalahan saat mencoba membuat koneksi
TcpTimeout	Koneksi tidak dapat dibuat dalam periode batas waktu
ConnectionFunctionError	Pengecualian yang tidak tertangkap dilemparkan selama eksekusi Fungsi Koneksi
Internal	Terjadi kesalahan layanan internal
UnmappedConnectionError	Terjadi kesalahan yang tidak termasuk dalam kategori lainnya

Asal TLS timbal balik dengan CloudFront

Mutual TLS Authentication (Mutual Transport Layer Security Authentication — mTLS) adalah protokol keamanan yang memperluas otentikasi TLS standar dengan memerlukan otentikasi berbasis sertifikat dua arah, di mana klien dan server harus membuktikan identitas mereka sebelum membuat koneksi yang aman.

Penampil mTL vs mTL Asal

Otentikasi timbal balik (mTL) dapat diaktifkan antara pemirsa dan CloudFront distribusi Anda (mTL penampil) and/or juga antara CloudFront distribusi Anda dan asal (mTL asal). Dokumentasi ini berkaitan dengan konfigurasi mTLS asal. Untuk konfigurasi mTLS penampil lihat: [Otentikasi TLS timbal balik dengan CloudFront \(Viewer mTLS\)](#).

Origin mTLS memungkinkan CloudFront untuk mengautentikasi dirinya ke server asal Anda menggunakan sertifikat klien. Dengan mTL asal, Anda dapat memastikan bahwa hanya CloudFront distribusi resmi Anda yang dapat membuat koneksi dengan server aplikasi Anda, membantu melindungi dari upaya akses yang tidak sah.

Note

Dalam koneksi mTLS asal, CloudFront bertindak sebagai klien dan menyajikan sertifikat kliennya ke server asal Anda selama jabat tangan TLS. CloudFront tidak melakukan validasi validitas atau status pencabutan sertifikat klien — ini adalah tanggung jawab server asal Anda. Infrastruktur asal Anda harus dikonfigurasi untuk memvalidasi sertifikat klien terhadap trust store, memeriksa kedaluwarsa sertifikat, dan melakukan pemeriksaan pencabutan (seperti validasi CRL atau OCSP) sesuai dengan persyaratan keamanan Anda. CloudFront Peran terbatas pada penyajian sertifikat; semua logika validasi sertifikat dan kebijakan keamanan diberlakukan oleh server asal Anda.

Cara kerjanya

Dalam jabat tangan TLS standar antara CloudFront dan asal, hanya server asal yang menyajikan sertifikat untuk membuktikan identitasnya. CloudFront Dengan mTL asal, proses otentikasi menjadi dua arah. Saat CloudFront mencoba terhubung ke server asal Anda, berikan CloudFront sertifikat klien selama jabat tangan TLS. Server asal Anda memvalidasi sertifikat ini terhadap trust store sebelum membuat koneksi aman.

Kasus penggunaan

Origin mTLS membahas beberapa skenario keamanan kritis di mana metode otentikasi tradisional membuat overhead operasional:

- Keamanan hybrid dan multi-cloud - Anda dapat mengamankan koneksi antara CloudFront dan asal-usul yang dihosting di luar AWS atau asal publik. AWS Ini menghilangkan kebutuhan untuk mengelola daftar izin IP atau solusi header khusus, menyediakan otentikasi berbasis sertifikat yang konsisten di seluruh AWS, pusat data lokal, dan penyedia pihak ketiga. Perusahaan media, pengecer, dan perusahaan yang mengoperasikan infrastruktur terdistribusi mendapat manfaat dari kontrol keamanan standar di seluruh infrastruktur mereka.
- B2B API dan keamanan backend - Anda dapat melindungi API backend dan layanan mikro Anda dari upaya akses langsung sambil mempertahankan manfaat kinerja. CloudFront Platform SaaS, sistem pemrosesan pembayaran, dan aplikasi perusahaan dengan persyaratan otentikasi yang ketat dapat memverifikasi bahwa permintaan API hanya berasal dari CloudFront distribusi resmi, mencegah serangan man-in-the-middle dan upaya akses yang tidak sah.

Penting: Persyaratan Server Asal

Origin mTLS mengharuskan server asal Anda dikonfigurasi untuk mendukung otentikasi TLS timbal balik. Infrastruktur asal Anda harus mampu:

- Meminta dan memvalidasi sertifikat klien selama jabat tangan TLS
- Mempertahankan toko kepercayaan dengan sertifikat Otoritas Sertifikat yang mengeluarkan sertifikat CloudFront klien
- Pencatatan dan pemantauan peristiwa koneksi TLS timbal balik
- Mengelola kebijakan validasi sertifikat dan menangani kegagalan otentikasi

CloudFront menangani presentasi sertifikat sisi klien, tetapi server asal Anda bertanggung jawab untuk memvalidasi sertifikat ini dan mengelola koneksi TLS bersama. Pastikan infrastruktur asal Anda dikonfigurasi dengan benar sebelum mengaktifkan mTL asal. CloudFront

Memulai

Untuk mengimplementasikan mTL asal CloudFront, Anda harus mengimpor sertifikat klien di AWS Certificate Manager, mengonfigurasi server asal Anda agar memerlukan TLS bersama, dan

mengaktifkan mTLS asal pada distribusi Anda. CloudFront Bagian berikut memberikan petunjuk langkah demi langkah untuk setiap tugas konfigurasi.

Topik

- [Manajemen sertifikat dengan AWS Certificate Manager](#)
- [Aktifkan TLS timbal balik asal untuk distribusi CloudFront](#)
- [Menggunakan CloudFront Fungsi dengan TLS timbal balik asal](#)

Manajemen sertifikat dengan AWS Certificate Manager

[AWS Certificate Manager \(ACM\)](#) menyimpan sertifikat klien yang CloudFront ditampilkan ke server asal Anda selama otentikasi TLS timbal balik asal.

Dukungan Otoritas Sertifikat

CloudFront mTLS asal memerlukan sertifikat klien dengan Extended Key Usage (EKU) untuk Otentikasi Klien TLS. Karena persyaratan ini, Anda harus mengeluarkan sertifikat dari Otoritas Sertifikat Anda dan mengimpornya ke AWS Certificate Manager. Fitur penyediaan dan pembaruan sertifikat otomatis ACM tidak tersedia untuk sertifikat klien mTLS asal. CloudFront mTLS asal mendukung sertifikat klien dari dua sumber:

- **AWS Otoritas Sertifikat Pribadi:** Anda dapat menerbitkan sertifikat dari CA AWS Pribadi menggunakan templat sertifikat yang menyertakan Otentikasi Klien TLS di bidang Penggunaan Kunci yang Diperpanjang (seperti EndEntityClientAuthCertificate templat). Setelah mengeluarkan sertifikat dari AWS Private CA, Anda harus mengimpornya ke ACM di Wilayah AS Timur (Virginia N.) (us-east-1). Pendekatan ini memberikan manfaat keamanan dari AWS Private CA sekaligus memberi Anda kendali atas manajemen siklus hidup sertifikat.
- **Third-party Otoritas Sertifikat pribadi:** Anda juga dapat menerbitkan sertifikat dari infrastruktur Otoritas Sertifikat pribadi Anda yang ada dan mengimpornya ke ACM. Ini memungkinkan Anda untuk mempertahankan proses manajemen sertifikat Anda saat ini sambil memanfaatkan kemampuan CloudFront mTLS asal. Sertifikat harus menyertakan Otentikasi Klien TLS di bidang Penggunaan Kunci Diperpanjang dan harus dalam format PEM dengan sertifikat, kunci pribadi, dan rantai sertifikat.

 Important

Untuk CA AWS Pribadi dan CA pihak ketiga, Anda bertanggung jawab untuk memantau tanggal kedaluwarsa sertifikat dan mengimpor sertifikat yang diperbarui ke ACM sebelum kedaluwarsa. Fitur perpanjangan otomatis ACM tidak berlaku untuk sertifikat impor yang digunakan untuk mTL asal.

Persyaratan dan spesifikasi sertifikat

Persyaratan sertifikat klien

- Format: Format PEM (Privacy Enhanced Mail)
- Komponen: Sertifikat, kunci pribadi, dan rantai sertifikat
- Kedalaman rantai sertifikat maksimum: 3 (sertifikat daun+sertifikat perantara+sertifikat akar)
- Ukuran rantai sertifikat maksimum: 64 KB
- Ukuran sertifikat: Tidak boleh melebihi 96 KB
- Ukuran kunci pribadi maksimum: 5 KB (pembatasan ACM)
- ARN sertifikat mTL asal unik maksimum yang dapat ditambahkan atau dimodifikasi per CloudFront distribusi membuat atau memperbarui panggilan API: 5
- Wilayah: Sertifikat harus disimpan di ACM di Wilayah AS Timur (Virginia N.) (us-east-1)

Spesifikasi sertifikat yang didukung

- Jenis sertifikat: X.509v3
- Algoritma kunci publik:
 - RSA: 2048-bit
 - ECDSA: P-256
- Algoritma tanda tangan:
 - SHA256, SHA384, SHA512 dengan RSA
 - SHA256, SHA384, SHA512 dengan ECDSA
 - SHA256, SHA384, SHA512 dengan dengan MGF1 RSASSA-PSS
- Penggunaan Kunci yang Diperpanjang (wajib): Sertifikat memerlukan ekstensi Extended Key Usage (EKU) yang disetel ke Otentikasi Klien TLS, memastikannya diizinkan untuk tujuan mTLS

Persyaratan sertifikat server

Server asal Anda harus menunjukkan sertifikat dari Otoritas Sertifikat yang dipercaya publik selama jabat tangan TLS bersama. Untuk detail lengkap tentang persyaratan sertifikat server asal, lihat [Persyaratan untuk menggunakan SSL/TLS sertifikat dengan CloudFront](#).

Meminta atau mengimpor sertifikat

Sebelum mengaktifkan mTL asal, Anda harus memiliki sertifikat klien yang tersedia di ACM.

Meminta dan mengimpor sertifikat dari AWS CA Privat

Prasyarat:

- Otoritas Sertifikat AWS Pribadi yang dikonfigurasi di akun Anda
- Izin untuk menerbitkan sertifikat dari AWS Private CA
- Izin untuk mengimpor sertifikat ke ACM
- [Templat sertifikat](#) ARN Extended key usage:TLS web client authentication yang sesuai dengan kasus penggunaan Anda
- Instal OpenSSL AWS , CLI, dan jq (untuk parsing JSON).

Untuk meminta sertifikat dari PCA dan impor ke ACM (AWS CLI)

1. Atur ARN CA Pribadi Anda dalam variabel agar lebih mudah digunakan kembali.

```
PCA_ARN="arn:aws:acm-pca:region:account:certificate-authority/12345678..."
```

2. Gunakan OpenSSL untuk menghasilkan kunci pribadi P-256 ECDSA (kurva prime256v1) dan Certificate Signing Request (CSR), memastikan flag -nodes digunakan untuk menjaga kunci pribadi tidak terenkripsi seperti yang diperlukan untuk impor ACM.

```
openssl req -new -newkey ec -pkeyopt ec_paramgen_curve:prime256v1 -nodes \  
-keyout private.key \  
-out request.csr \  
-subj "/CN=client.example.com"
```

3. Kirimkan CSR ke CA AWS Pribadi Anda untuk menerbitkan sertifikat, yang mengembalikan ARN dari sertifikat yang baru diterbitkan.

```
CERT_ARN=$(aws acm-pca issue-certificate \  

```



```
--certificate-authority-arn "$PCA_ARN" \
--csr fileb://request.csr \
--signing-algorithm "SHA256WITHECDSA" \
--validity Value=365,Type="DAYS" \
--template-arn arn:aws:acm-pca:::template/EndEntityCertificate/V1 \
--query 'CertificateArn' --output text)
```

4. Ambil bundel sertifikat dari AWS PCA menggunakan perintah `get-certificate`, yang mengembalikan sertifikat daun dan rantai, lalu gunakan `jq` untuk memisahkannya menjadi file yang berbeda.

```
# Retrieve the full certificate bundle in JSON format
aws acm-pca get-certificate \
  --certificate-authority-arn "$PCA_ARN" \
  --certificate-arn "$CERT_ARN" \
  --output json > full_cert.json

# Split into Leaf and Chain
jq -r '.Certificate' full_cert.json > leaf_cert.pem
jq -r '.CertificateChain' full_cert.json > cert_chain.pem
```

5. Impor kunci pribadi yang tidak terenkripsi, sertifikat daun, dan rantai sertifikat ke AWS ACM, menggunakan protokol `fileb://` untuk menangani data file biner dengan benar di CLI.

```
aws acm import-certificate \
  --certificate fileb://leaf_cert.pem \
  --private-key fileb://private.key \
  --certificate-chain fileb://cert_chain.pem \
  --region us-east-1 \
  --query 'CertificateArn' \
  --output text
```

Impor sertifikat dari CA pihak ketiga

Prasyarat:

- Sertifikat, kunci pribadi yang tidak terenkripsi, dan rantai sertifikat dari Otoritas Sertifikat Anda dalam format PEM
- Sertifikat harus menyertakan Penggunaan Kunci yang Diperpanjang untuk Otentikasi Klien TLS
- Izin untuk mengimpor sertifikat ke ACM

Untuk mengimpor sertifikat ke ACM (AWS CLI)

```
aws acm import-certificate \  
  --certificate fileb://certificate.pem \  
  --private-key fileb://private-key.pem \  
  --certificate-chain fileb://certificate-chain.pem \  
  --region us-east-1 \  
  --query 'CertificateArn' \  
  --output text
```

Langkah selanjutnya

Setelah memperoleh atau mengimpor sertifikat klien Anda di ACM, Anda dapat mengonfigurasi server asal Anda untuk meminta otentikasi TLS bersama dan mengaktifkan mTL asal pada distribusi Anda. CloudFront Untuk petunjuk tentang mengaktifkan mTL asal CloudFront, lihat bagian selanjutnya “Aktifkan TLS timbal balik asal untuk CloudFront distribusi.”

Aktifkan TLS timbal balik asal untuk distribusi CloudFront

Setelah mendapatkan sertifikat klien melalui AWS Certificate Manager dan mengonfigurasi server asal Anda untuk meminta TLS bersama, Anda dapat mengaktifkan mTL asal pada distribusi Anda. CloudFront

Prasyarat dan persyaratan

Sebelum mengaktifkan mTL asal pada CloudFront distribusi, pastikan Anda memiliki:

- Sertifikat klien yang disimpan di AWS Certificate Manager di Wilayah AS Timur (Virginia N.) (us-east-1)
- Server asal dikonfigurasi untuk memerlukan otentikasi TLS timbal balik dan memvalidasi sertifikat klien
- Server asal yang menyajikan sertifikat dari Otoritas Sertifikat yang dipercaya publik
- Izin untuk memodifikasi distribusi CloudFront
- Origin mTLS hanya tersedia pada paket Bisnis, Premium, atau Paket harga Pay as you go.

 Note

MTL Asal dapat dikonfigurasi untuk asal kustom (termasuk asal yang dihosting di luar AWS) dan AWS asal yang mendukung TLS timbal balik seperti Application Load Balancer dan API Gateway.

 Important

CloudFront Fitur-fitur berikut tidak didukung dengan mTL asal:

- lalu lintas gRPC: protokol gRPC tidak didukung untuk asal dengan mTL asal diaktifkan
- WebSocket koneksi: WebSocket protokol tidak didukung untuk asal dengan mTL asal diaktifkan
- Asal VPC: mTL asal tidak dapat digunakan dengan asal VPC
- Permintaan asal dan pemicu respons asal dengan Lambda @Edge: Fungsi Lambda @Edge dalam permintaan asal dan posisi respons asal tidak didukung dengan mTL asal
- POP yang disematkan: mTL asal tidak didukung untuk POP yang disematkan

Aktifkan mTL asal

Per-origin konfigurasi memungkinkan Anda untuk menentukan sertifikat klien yang berbeda untuk asal yang berbeda dalam distribusi yang sama. Pendekatan ini memberikan fleksibilitas maksimum ketika asal Anda memiliki persyaratan otentikasi yang berbeda.

Untuk distribusi baru (Konsol)

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Pilih Buat distribusi
3. Pilih paket harga: Pilih Bisnis atau Premium atau Bayar Saat Anda Pergi (MTL Asal tidak tersedia pada paket Gratis)
4. Di bagian Pengaturan asal, pilih Jenis Asal sebagai Lainnya
5. Di bagian Pengaturan asal, pilih Sesuaikan pengaturan asal
6. Konfigurasi asal pertama Anda (nama domain, protokol, dll.)

7. Dalam konfigurasi asal, temukan mTL
8. Alihkan mTLS ke On
9. Untuk sertifikat Klien, pilih sertifikat Anda dari AWS Certificate Manager
10. (Opsional) Tambahkan asal tambahan dengan konfigurasi mTL asal mereka sendiri
11. Lengkapi setelan distribusi yang tersisa dan pilih Buat distribusi

Untuk distribusi yang ada (Konsol)

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Dari daftar distribusi, pilih distribusi yang ingin Anda ubah. (Catatan: Pastikan distribusi Anda menggunakan paket harga Pro atau Premium atau Pay As You Go. Jika tidak, Anda harus meningkatkan paket harga Anda sebelum mengaktifkan mTL asal)
3. Pilih tab Origins
4. Pilih asal yang ingin Anda konfigurasikan dan pilih Edit
5. Di pengaturan asal, temukan mTL
6. Alihkan mTLS ke On
7. Untuk sertifikat Klien, pilih sertifikat Anda dari AWS Certificate Manager. (Catatan: Hanya sertifikat klien dengan properti ECU (Extended Key Usage) yang disetel ke "Otentikasi Klien TLS" yang akan dicantumkan)
8. Pilih Save changes (Simpan perubahan)
9. Ulangi untuk asal tambahan sesuai kebutuhan

Penggunaan AWS CLI

Untuk konfigurasi per asal, tentukan pengaturan mTLS asal dalam konfigurasi masing-masing asal:

```
{
  "Origins": {
    "Quantity": 2,
    "Items": [
      {
        "Id": "origin-1",
        "DomainName": "api.example.com",
        "CustomOriginConfig": {
          "HTTPSPort": 443,
```

```

      "OriginProtocolPolicy": "https-only"
    },
    "OriginMtlsConfig": {
      "ClientCertificateArn": "arn:aws:acm:us-east-1:123456789012:certificate/
cert-1"
    }
  },
  {
    "Id": "origin-2",
    "DomainName": "backend.example.com",
    "CustomOriginConfig": {
      "HTTPSPort": 443,
      "OriginProtocolPolicy": "https-only"
    },
    "OriginMtlsConfig": {
      "CertificateArn": "arn:aws:acm:us-east-1:123456789012:certificate/cert-2"
    }
  }
]
}
}

```

Note

CloudFront tidak akan memberikan sertifikat klien jika server tidak memintanya, memungkinkan koneksi berjalan normal.

Langkah selanjutnya

Setelah mengaktifkan mTL asal pada CloudFront distribusi Anda, Anda dapat memantau peristiwa otentikasi menggunakan CloudFront log akses.

Menggunakan CloudFront Fungsi dengan TLS timbal balik asal

CloudFront Fungsi menyediakan komputasi ringan tanpa server di tepi untuk menyesuaikan pengiriman konten. Saat menggunakan TLS timbal balik asal dengan CloudFront Fungsi, ada perilaku dan batasan khusus yang harus diperhatikan mengenai pemilihan dan manipulasi asal.

Operasi CloudFront Fungsi yang Didukung

CloudFront Fungsi dapat berinteraksi dengan TLS-enabled asal-usul m dengan cara berikut:

perbarui RequestOrigin ()

Fungsi update RequestOrigin () mendukung modifikasi terbatas saat bekerja dengan m TLS-enabled origin:

- Beralih di antara asal mTLS asal: Anda dapat memperbarui permintaan untuk merutekan ke asal berbeda yang menggunakan mTL asal, asalkan kedua asal menggunakan sertifikat klien yang sama. Hal ini memungkinkan Anda untuk menerapkan logika routing kustom sambil mempertahankan otentikasi TLS timbal balik. Beralih antar asal yang menggunakan sertifikat yang berbeda didukung melalui createRequestOriginGroup() API selectRequestOriginById() dan.
- Menonaktifkan mTL asal: Anda dapat beralih dari TLS-enabled asal m ke asal non-MTLS dengan mengatur fungsi. mTLSConfig: 'off' Ini memberikan fleksibilitas untuk menonaktifkan otentikasi TLS timbal balik secara kondisional berdasarkan karakteristik permintaan.

Contoh: Beralih antara asal mTLS asal dengan sertifikat yang sama

```
import cf from 'cloudfront';

function handler(event) {
    var request = event.request;

    // Route to different origin based on request path
    if (request.uri.startsWith('/api/v2')) {
        cf.updateRequestOrigin({
            "domainName": "api-v2.example.com",
            "mTLSConfig": "inherit",
            // If no value is provided for mTLSConfig, it defaults to inherit
            // Both origins must use the same certificate
        });
    }

    return request;
}
```

Contoh: Menonaktifkan mTL asal secara kondisional

```
import cf from 'cloudfront';

function handler(event) {
```

```
var request = event.request;

// Disable mTLS for specific paths
if (request.uri.startsWith('/public')) {
    cf.updateRequestOrigin({
        "domainName": "public-origin.example.com",
        "mTLSConfig": "off"
    });
}

return request;
}
```

pilih RequestOriginById ()

`selectRequestOriginById()` Fungsi ini mendukung pemilihan asal yang mengaktifkan TLS (asal) timbal balik. Anda dapat menggunakan fungsi ini untuk merutekan permintaan secara dinamis ke m TLS-enabled origin yang dikonfigurasi dalam distribusi Anda. Saat memilih TLS timbal balik (asal) yang diaktifkan dengan ID, CloudFront gunakan sertifikat klien yang dikonfigurasi untuk asal tersebut dalam pengaturan distribusi.

Contoh: Memilih TLS timbal balik (asal) yang diaktifkan dengan ID

```
import cf from 'cloudfront';

function handler(event) {
    var request = event.request;

    // Select mTLS-enabled origin based on request characteristics
    if (request.uri.startsWith('/secure-api')) {
        cf.selectRequestOriginById("mtls-origin-1");
    }

    return request;
}
```

buat RequestOriginGroup ()

`createRequestOriginGroup()` Fungsi ini mendukung pembuatan grup asal yang menyertakan asal-usul yang diaktifkan TLS (asal) timbal balik. Anda dapat secara dinamis membuat grup asal dengan m TLS-enabled origin untuk skenario failover dalam CloudFront Functions.

Contoh: Membuat grup asal dengan asal-usul yang diaktifkan TLS (origin) timbal balik

```
import cf from 'cloudfront';

function handler(event) {
  // Create origin group with mTLS-enabled primary and failover origins
  cf.createRequestOriginGroup({
    "originIds": ["mtls-origin-primary", "mtls-origin-failover"],
    "failoverCriteria": {
      "statusCodes": [500, 502, 503, 504]
    }
  });

  return event.request;
}
```

Sajikan konten pribadi dengan URL yang ditandatangani dan cookie yang ditandatangani

Banyak perusahaan yang mendistribusikan konten melalui internet ingin membatasi akses ke dokumen, data bisnis, aliran media, atau konten yang ditujukan untuk pengguna tertentu, misalnya, pengguna yang telah membayar biaya. Untuk menyajikan konten pribadi ini dengan aman CloudFront, Anda dapat melakukan hal berikut:

- Mengharuskan pengguna Anda mengakses konten pribadi Anda dengan menggunakan URL khusus yang CloudFront ditandatangani atau cookie yang ditandatangani.
- Mengharuskan pengguna mengakses konten Anda dengan menggunakan CloudFront URL, bukan URL yang mengakses konten langsung di server asal (misalnya, Amazon S3 atau server HTTP pribadi). Memerlukan CloudFront URL tidak diperlukan, tetapi kami merekomendasikannya untuk mencegah pengguna melewati batasan yang Anda tentukan di URL yang ditandatangani atau cookie yang ditandatangani.

Untuk informasi selengkapnya, lihat [Batasi akses ke file](#).

Cara menyajikan konten pribadi

Untuk mengonfigurasi CloudFront untuk menyajikan konten pribadi, lakukan tugas-tugas berikut:

1. (Opsional tetapi disarankan) Minta pengguna Anda untuk mengakses konten Anda hanya melalui CloudFront. Metode yang Anda gunakan tergantung pada apakah Anda menggunakan Amazon S3 atau asal kustom:

- Amazon S3 – Lihat [the section called “Batasi akses ke asal Amazon S3”](#).
- Asal yang disesuaikan – Lihat [Batasi akses ke file pada asal kustom](#).

Asal kustom termasuk Amazon EC2, bucket Amazon S3 yang dikonfigurasi sebagai titik akhir situs web, Elastic Load Balancing, dan server web HTTP Anda sendiri.

2. Tentukan kelompok kunci tepercaya atau penanda tangan tepercaya yang ingin Anda gunakan untuk membuat URL yang ditandatangani atau cookie yang ditandatangani. Kami menyarankan agar Anda menggunakan grup kunci tepercaya. Untuk informasi selengkapnya, lihat [Tentukan penandatanganan yang dapat membuat URL yang ditandatangani dan cookie yang ditandatangani](#).
3. Tulis aplikasi Anda untuk merespons permintaan dari pengguna yang sah baik dengan URL yang ditandatangani atau dengan header Set-Cookie yang mengatur cookie yang ditandatangani. Ikuti langkah-langkah di salah satu topik berikut:
 - [Gunakan URL yang ditandatangani](#)
 - [Gunakan cookie yang ditandatangani](#)

Jika Anda tidak yakin metode mana yang harus digunakan, lihat [Memutuskan untuk menggunakan URL yang ditandatangani atau cookie yang ditandatangani](#).

Topik

- [Batasi akses ke file](#)
- [Tentukan penandatanganan yang dapat membuat URL yang ditandatangani dan cookie yang ditandatangani](#)
- [Memutuskan untuk menggunakan URL yang ditandatangani atau cookie yang ditandatangani](#)
- [Gunakan URL yang ditandatangani](#)
- [Gunakan cookie yang ditandatangani](#)
- [Perintah Linux dan OpenSSL untuk pengkodean dan enkripsi base64](#)
- [Contoh kode untuk membuat tanda tangan untuk URL yang ditandatangani](#)

Batasi akses ke file

Anda dapat mengontrol akses pengguna ke konten pribadi Anda dengan dua cara:

- [Batasi akses ke file dalam CloudFront cache.](#)
- Batasi akses ke file yang ada di tempat Anda dengan melakukan salah satu hal berikut:
 - [Siapkan kontrol akses asal \(OAC\) untuk bucket Amazon S3 Anda.](#)
 - [Konfigurasi header khusus untuk server HTTP privat \(asal khusus\).](#)

Batasi akses ke file dalam cache CloudFront

Anda dapat mengonfigurasi CloudFront agar pengguna mengakses file Anda menggunakan URL yang ditandatangani atau cookie yang ditandatangani. Kemudian, Anda mengembangkan aplikasi Anda untuk membuat dan mendistribusikan untuk pengguna terotentikasi atau untuk mengirim header Set-Cookie yang menetapkan cookie yang ditandatangani untuk pengguna yang diautentikasi. (Untuk memberi beberapa pengguna akses jangka panjang ke sejumlah kecil file, Anda juga dapat membuat URL secara manual.)

Saat Anda membuat URL atau cookie yang ditandatangani untuk mengendalikan akses ke file Anda, Anda dapat menentukan batasan berikut:

- Tanggal dan waktu akhir, yang setelahnya URL tidak lagi valid.
- (Opsional) Tanggal dan waktu URL menjadi valid.
- (Opsional) Alamat IP atau berbagai alamat komputer yang dapat digunakan untuk mengakses konten Anda.

Salah satu bagian dari URL yang ditandatangani atau cookie yang ditandatangani di-hash dan ditandatangani menggunakan kunci pribadi dari pasangan kunci publik–swasta. Ketika seseorang menggunakan URL yang ditandatangani atau cookie yang ditandatangani untuk mengakses file, CloudFront bandingkan bagian URL atau cookie yang ditandatangani dan tidak ditandatangani. Jika mereka tidak cocok, CloudFront tidak melayani file.

Anda harus menggunakan kunci pribadi RSA 2048 atau ECDSA 256 untuk menandatangani URL atau cookie.

Batasi akses ke file di bucket Amazon S3

Anda dapat mengamankan konten secara opsional di bucket Amazon S3 Anda sehingga pengguna dapat mengaksesnya melalui distribusi yang CloudFront ditentukan tetapi tidak dapat mengaksesnya secara langsung dengan menggunakan URL Amazon S3. Ini mencegah seseorang melewati CloudFront dan menggunakan URL Amazon S3 untuk mendapatkan konten yang ingin Anda batasi aksesnya. Langkah ini tidak diwajibkan untuk menggunakan URL yang ditandatangani, tetapi kami merekomendasikannya.

Untuk mengharuskan pengguna mengakses konten Anda melalui CloudFront URL, Anda melakukan tugas-tugas berikut:

- Berikan izin kontrol akses CloudFront asal untuk membaca file di bucket S3.
- Buat kontrol akses asal dan kaitkan dengan CloudFront distribusi Anda.
- Hapus izin orang lain untuk menggunakan URL Amazon S3 untuk membaca file.

Untuk informasi selengkapnya, lihat [the section called “Batasi akses ke asal Amazon S3”](#).

Batasi akses ke file pada asal kustom

Jika Anda menggunakan asal kustom, Anda dapat secara opsional mengatur header khusus untuk membatasi akses. CloudFront Untuk mendapatkan file Anda dari asal kustom, file harus dapat diakses dengan CloudFront menggunakan permintaan HTTP (atau HTTPS) standar. Tetapi dengan menggunakan header khusus, Anda dapat lebih membatasi akses ke konten Anda sehingga pengguna dapat mengaksesnya hanya melalui CloudFront, tidak secara langsung. Langkah ini tidak diwajibkan untuk menggunakan URL yang ditandatangani, tetapi kami merekomendasikannya.

Untuk mengharuskan pengguna mengakses konten CloudFront, ubah pengaturan berikut di CloudFront distribusi Anda:

Header Kustom Asal

Konfigurasi CloudFront untuk meneruskan header khusus ke asal Anda. Lihat [Konfigurasi CloudFront untuk menambahkan header khusus ke permintaan asal](#).

Kebijakan Protokol Penampil

Konfigurasi distribusi Anda agar pemirsa menggunakan HTTPS untuk mengakses CloudFront. Lihat [Kebijakan protokol penampil](#).

Kebijakan Protokol Asal

Konfigurasi distribusi Anda CloudFront agar perlu menggunakan protokol yang sama dengan pemirsa untuk meneruskan permintaan ke asal. Lihat [Protokol \(hanya asal kustom\)](#).

Setelah Anda membuat perubahan ini, perbarui aplikasi Anda di asal kustom Anda untuk hanya menerima permintaan yang menyertakan header khusus yang telah Anda konfigurasi CloudFront untuk dikirim.

Kombinasi dari Kebijakan Protokol Penampil dan Kebijakan Protokol Asal memastikan bahwa header kustom dienkripsi saat transit. Namun, kami menyarankan Anda melakukan hal berikut secara berkala untuk memutar header khusus yang CloudFront diteruskan ke asal Anda:

1. Perbarui CloudFront distribusi Anda untuk mulai meneruskan header baru ke asal kustom Anda.
2. Perbarui aplikasi Anda untuk menerima header baru sebagai konfirmasi bahwa permintaan tersebut berasal CloudFront.
3. Ketika permintaan tidak lagi menyertakan header yang Anda ganti, perbarui aplikasi Anda agar tidak lagi menerima header lama sebagai konfirmasi bahwa permintaan tersebut berasal CloudFront.

Tentukan penandatanganan yang dapat membuat URL yang ditandatangani dan cookie yang ditandatangani

Topik

- [Pilih antara grup kunci tepercaya \(disarankan\) dan Akun AWS](#)
- [Buat pasangan kunci untuk penandatanganan Anda](#)
- [Memformat ulang kunci pribadi \(hanya .NET dan Java\)](#)
- [Menambahkan tanda tangan ke distribusi](#)
- [Pasangan kunci berputar](#)

Untuk membuat tanda tangan atau cookie yang ditandatangani, Anda perlu signer. Penandatanganan adalah grup kunci tepercaya yang Anda buat CloudFront, atau AWS akun yang berisi CloudFront key pair. Kami sarankan Anda menggunakan grup kunci tepercaya dengan URL dan cookie yang ditandatangani. Untuk informasi selengkapnya, lihat [Pilih antara grup kunci tepercaya \(disarankan\) dan Akun AWS](#).

Signer memiliki dua tujuan:

- Segera setelah Anda menambahkan tanda tangan ke distribusi Anda, CloudFront mulai mengharuskan pemirsa menggunakan URL yang ditandatangani atau cookie yang ditandatangani untuk mengakses file Anda.
- Saat Anda membuat URL atau cookie yang ditandatangani, Anda menggunakan kunci privat dari pasangan kunci signer untuk menandatangani sebagian URL atau cookie. Ketika seseorang meminta file terbatas, CloudFront membandingkan tanda tangan di URL atau cookie dengan URL atau cookie yang tidak ditandatangani, untuk memverifikasi bahwa itu belum dirusak. CloudFront juga memverifikasi bahwa URL atau cookie valid, artinya, misalnya, bahwa tanggal kedaluwarsa dan waktu belum berlalu.

Saat Anda menentukan signer, Anda juga secara tidak langsung menentukan file yang memerlukan URL atau cookie yang ditandatangani dengan menambahkan signer ke perilaku cache. Jika distribusi Anda hanya memiliki satu perilaku tembolok, penampil harus menggunakan URL atau cookie yang ditandatangani untuk mengakses setiap file dalam distribusi. Jika Anda membuat beberapa perilaku cache dan menambahkan signer ke beberapa perilaku cache dan tidak pada yang lain, Anda dapat meminta penampil untuk menggunakan tanda tangan URL atau cookie yang ditandatangani untuk mengakses beberapa file dan bukan file lainnya.

Untuk menentukan tanda tangan (kunci pribadi) yang diizinkan untuk membuat URL yang ditandatangani atau cookie yang ditandatangani, dan untuk menambahkan tanda tangan ke CloudFront distribusi Anda, lakukan tugas-tugas berikut:

1. Putuskan apakah akan menggunakan grup kunci tepercaya atau Akun AWS sebagai penandatanganan. Kami merekomendasikan penggunaan grup kunci tepercaya. Untuk informasi selengkapnya, lihat [Pilih antara grup kunci tepercaya \(disarankan\) dan Akun AWS](#).
2. Untuk signer yang Anda pilih pada langkah 1, buat pasangan kunci publik–pribadi. Untuk informasi selengkapnya, lihat [Buat pasangan kunci untuk penandatanganan Anda](#).
3. Jika Anda menggunakan .NET atau Java untuk membuat tanda tangan atau cookie yang ditandatangani, format ulang kunci privat. Untuk informasi selengkapnya, lihat [Memformat ulang kunci pribadi \(hanya .NET dan Java\)](#).
4. Dalam distribusi tempat Anda membuat URL atau cookie yang ditandatangani, tentukan signernya. Untuk informasi selengkapnya, lihat [Menambahkan tanda tangan ke distribusi](#).

Pilih antara grup kunci tepercaya (disarankan) dan Akun AWS

Menggunakan tanda tangan atau cookie yang ditandatangani, Anda perlu signer. Penandatanganan adalah grup kunci tepercaya yang Anda buat CloudFront, atau Akun AWS yang berisi CloudFront key pair. Kami sarankan Anda menggunakan grup kunci tepercaya, karena alasan berikut:

- Dengan grup CloudFront kunci, Anda tidak perlu menggunakan pengguna root AWS akun untuk mengelola kunci publik untuk URL yang CloudFront ditandatangani dan cookie yang ditandatangani. [AWS praktik terbaik](#) merekomendasikan agar Anda tidak menggunakan pengguna root saat Anda tidak perlu melakukannya.
- Dengan grup CloudFront kunci, Anda dapat mengelola kunci publik, grup kunci, dan penandatanganan tepercaya menggunakan CloudFront API. Anda dapat menggunakan API untuk mengotomatiskan pembuatan kunci dan rotasi utama. Saat Anda menggunakan pengguna AWS root, Anda harus menggunakan Konsol Manajemen AWS untuk mengelola pasangan CloudFront kunci, sehingga Anda tidak dapat mengotomatiskan prosesnya.
- Karena Anda dapat mengelola grup kunci dengan CloudFront API, Anda juga dapat menggunakan kebijakan izin AWS Identity and Access Management (IAM) untuk membatasi apa yang diizinkan dilakukan oleh pengguna yang berbeda. Misalnya, Anda dapat mengizinkan pengguna mengunggah kunci publik, tetapi tidak dapat menghapusnya. Atau Anda dapat mengizinkan pengguna untuk menghapus kunci publik, tetapi hanya jika kondisi tertentu terpenuhi, seperti menggunakan autentikasi multifaktor, mengirim permintaan dari jaringan tertentu, atau mengirim permintaan dalam rentang tanggal dan waktu tertentu.
- Dengan grup CloudFront kunci, Anda dapat mengaitkan jumlah kunci publik yang lebih tinggi dengan CloudFront distribusi Anda, memberi Anda lebih banyak fleksibilitas dalam cara Anda menggunakan dan mengelola kunci publik. Secara default, Anda dapat mengaitkan hingga empat kelompok utama dengan satu distribusi, dan Anda dapat memiliki hingga lima kunci publik dalam grup utama.

Saat Anda menggunakan pengguna root AWS akun untuk mengelola pasangan CloudFront kunci, Anda hanya dapat memiliki hingga dua pasangan CloudFront kunci aktif per AWS akun.

Buat pasangan kunci untuk penandatanganan Anda

Setiap penandatanganan yang Anda gunakan untuk membuat URL yang CloudFront ditandatangani atau cookie yang ditandatangani harus memiliki key pair publik-pribadi. Penandatanganan menggunakan kunci pribadinya untuk menandatangani URL atau cookie, dan CloudFront menggunakan kunci publik untuk memverifikasi tanda tangan.

Cara Anda membuat key pair tergantung pada apakah Anda menggunakan grup kunci tepercaya sebagai penandatanganan (recommended), atau CloudFront key pair. Untuk informasi selengkapnya, silakan lihat bagian-bagian berikut ini. Pasangan kunci yang Anda buat harus memenuhi persyaratan berikut:

- Ini harus berupa key pair SSH-2 RSA 2048 atau ECDSA 256.
- Informasi tersebut harus dalam format PEM yang dikodekan base64.

Untuk membantu mengamankan aplikasi Anda, kami sarankan Anda memutar pasangan kunci secara berkala. Untuk informasi selengkapnya, lihat [Pasangan kunci berputar](#).

Buat pasangan kunci untuk kelompok kunci yang dipercaya (disarankan)

Untuk membuat pasangan kunci untuk grup kunci tepercaya, lakukan langkah-langkah berikut:

1. Ciptakan pasangan kunci publik–pribadi.
2. Unggah kunci publik ke CloudFront.
3. Tambahkan kunci publik ke grup CloudFront kunci.

Untuk informasi selengkapnya, lihat prosedur berikut.

Untuk membuat pasangan kunci

Note

Langkah-langkah berikut menggunakan OpenSSL sebagai contoh dari salah satu metode untuk membuat key pair. Ada banyak cara lain untuk membuat key pair RSA atau ECDSA.

1. Jalankan salah satu perintah contoh berikut:

- Contoh perintah berikut menggunakan OpenSSL untuk membuat pasangan kunci RSA dengan panjang 2048 bit dan menyimpan ke file dengan nama `private_key.pem`.

```
openssl genrsa -out private_key.pem 2048
```

- Contoh perintah berikut menggunakan OpenSSL untuk menghasilkan key pair ECDSA `prime256v1` dengan kurva dan simpan ke file bernama `private_key.pem`

```
openssl ecparam -name prime256v1 -genkey -noout -out privatekey.pem
```

2. Berkas yang dihasilkan berisi baik publik maupun kunci pribadi. Contoh perintah berikut mengekstrak kunci publik dari file yang diberi nama `private_key.pem`.

```
openssl rsa -pubout -in private_key.pem -out public_key.pem
```

Anda mengunggah kunci publik (di `public_key.pem` file) nanti, dalam prosedur berikut.

Untuk mengunggah kunci publik ke CloudFront

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Dalam menu navigasi, pilih Kunci publik.
3. Pilih Buat kunci publik.
4. Di jendela Create public key, lakukan hal berikut:
 - a. Untuk Nama kunci, ketik nama untuk mengidentifikasi kunci publik.
 - b. Untuk Nilai utama, merekatkan kunci publik. Jika Anda mengikuti langkah-langkah dalam prosedur sebelumnya, kunci publik ada dalam file dengan nama `public_key.pem`. Untuk menyalin dan menempelkan isi kunci publik, Anda dapat:
 - Gunakan perintah cat pada baris perintah macOS atau Linux, seperti ini:

```
cat public_key.pem
```

Salin hasil dari perintah tersebut, kemudian rekatkan ke Nilai utama bidang.

- Buka `public_key.pem` file dengan editor teks biasa seperti Notepad (di Windows) atau (di macOS). TextEdit Salin konten file, lalu tempelkan ke Nilai utama bidang.
- c. (Opsional) Untuk Komentar, tambahkan komentar untuk menggambarkan kunci publik.

Setelah selesai, pilih Tambahkan.

5. Catat ID kunci publik. Anda menggunakannya nanti saat membuat URL atau cookie yang ditandatangani, sebagai nilai bidang `Key-Pair-Id`.

Untuk menambahkan kunci publik ke kelompok utama

1. Buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Dalam menu navigasi, pilih Kelompok utama.
3. Pilih Tambahkan kelompok kunci.
4. Di Buat grup utama, lakukan hal berikut:
 - a. Untuk Nama grup utama, ketikkan nama untuk mengidentifikasi kelompok kunci.
 - b. (Opsional) Untuk Komentar, ketik komentar untuk mendeskripsikan kelompok utama.
 - c. Untuk Kunci publik, pilih kunci publik untuk ditambahkan ke kelompok utama, lalu pilih Tambahkan. Ulangi langkah ini untuk setiap kunci publik yang ingin Anda tambahkan ke grup utama.
5. Pilih Buat grup utama.
6. Catat nama kelompok kunci. Anda menggunakannya nanti untuk mengaitkan grup kunci dengan perilaku cache dalam CloudFront distribusi. (Di CloudFront API, Anda menggunakan ID grup kunci untuk mengaitkan grup kunci dengan perilaku cache.)

Buat CloudFront key pair (tidak disarankan, membutuhkan Akun AWS pengguna root)

 Important

Kami sarankan Anda membuat kunci publik untuk grup kunci yang dipercaya, bukan mengikuti langkah-langkah ini. Dengan cara yang direkomendasikan untuk membuat kunci publik untuk URL dan cookie yang ditandatangani, lihat [Buat pasangan kunci untuk kelompok kunci yang dipercaya \(disarankan\)](#).

Anda dapat membuat CloudFront key pair dengan cara berikut:

- Buat key pair di Konsol Manajemen AWS dan unduh kunci privat. Lihat prosedur berikut.
- Buat pasangan kunci RSA dengan menggunakan aplikasi seperti OpenSSL, lalu unggah kunci publik ke Konsol Manajemen AWS. Untuk informasi lebih lanjut tentang membuat pasangan kunci RSA, lihat [Buat pasangan kunci untuk kelompok kunci yang dipercaya \(disarankan\)](#).

Untuk membuat pasangan CloudFront kunci di Konsol Manajemen AWS

1. Masuk ke Konsol Manajemen AWS menggunakan kredensial pengguna root AWS akun.

Important

Pengguna IAM tidak dapat membuat pasangan CloudFront kunci. Anda harus masuk menggunakan kredensial pengguna akar untuk membuat pasangan kunci.

2. Pilih nama akun Anda, lalu pilih Kredensial Keamanan Saya.
3. Pilih CloudFront pasangan kunci.
4. Konfirmasikan bahwa Anda tidak memiliki lebih dari satu pasangan kunci aktif. Anda tidak dapat membuat pasangan kunci jika Anda sudah memiliki dua pasangan kunci aktif.
5. Pilih Buat Pasangan Kunci Baru.

Note

Anda juga dapat memilih untuk membuat key pair Anda sendiri dan mengunggah kunci publik. CloudFront pasangan kunci mendukung kunci 1024, 2048, atau 4096-bit.

6. Di Buat Pasangan Utama kotak dialog, pilih Unduh File Kunci Pribadi, lalu simpan file di komputer Anda.

Important

Simpan kunci pribadi untuk CloudFront key pair Anda di lokasi yang aman, dan atur izin pada file sehingga hanya administrator yang diinginkan yang dapat membacanya. Jika seseorang mendapatkan kunci pribadi Anda, mereka dapat membuat URL valid yang ditandatangani dan cookie yang ditandatangani serta mengunduh konten Anda. Anda tidak bisa mendapatkan kunci pribadi lagi, jadi jika Anda kehilangan atau menghapusnya, Anda harus membuat CloudFront key pair baru.

7. Catat ID pasangan kunci untuk pasangan kunci Anda. (Dalam Konsol Manajemen AWS, ini disebut ID Kunci Akses.) Anda akan menggunakannya saat membuat URL atau cookie yang ditandatangani.

Memformat ulang kunci pribadi (hanya .NET dan Java)

Jika Anda menggunakan .NET atau Java untuk membuat tanda tangan atau cookie yang telah ditandatangani, Anda tidak dapat menggunakan kunci privat dari pasangan kunci Anda dalam format PEM default untuk membuat tanda tangan. Sebaliknya, lakukan hal berikut:

- Kerangka kerja .NET – Konversikan kunci pribadi ke format XML yang digunakan kerangka kerja .NET. Tersedia beberapa alat.
- Jawa – Mengonversi kunci pribadi menjadi format DER. Salah satu cara untuk melakukannya adalah dengan perintah OpenSSL. Dalam perintah berikut, `private_key.pem` adalah nama file yang berisi kunci PEM-formatted pribadi, dan `private_key.der` merupakan nama file yang berisi kunci DER-formatted pribadi setelah Anda menjalankan perintah.

```
openssl pkcs8 -topk8 -nocrypt -in private_key.pem -inform PEM -out private_key.der -  
outform DER
```

Untuk memastikan bahwa enkoder bekerja dengan benar, tambahkan JAR untuk kriptografi API Bouncy Castle Java ke proyek Anda, lalu tambahkan penyedia Bouncy Castle.

Menambahkan tanda tangan ke distribusi

Penandatanganan adalah grup kunci tepercaya (recommended) atau CloudFront key pair yang dapat membuat URL yang ditandatangani dan cookie yang ditandatangani untuk distribusi. Untuk menggunakan URL yang ditandatangani atau cookie yang ditandatangani dengan CloudFront distribusi, Anda harus menentukan tanda tangan.

Signer dikaitkan dengan perilaku cache. Hal ini memungkinkan Anda untuk memerlukan URL yang ditandatangani atau cookie yang ditandatangani untuk beberapa file dan bukan untuk file lain dalam distribusi yang sama. Distribusi perlu URL atau cookie yang ditandatangani hanya untuk file yang terkait dengan perilaku cache yang sesuai.

Demikian pula, signer hanya dapat menandatangani URL atau cookie untuk file yang terkait dengan perilaku cache yang sesuai. Misalnya, jika Anda memiliki satu signer untuk perilaku cache dan signer yang berbeda untuk perilaku cache yang berbeda, kedua signer tidak dapat membuat URL atau cookie yang ditandatangani untuk file yang terkait dengan perilaku cache lainnya.

 Important

Sebelum Anda menambahkan signer ke distribusi Anda, lakukan hal berikut:

- Tentukan pola jalur dalam perilaku cache dan urutan perilaku cache secara saksama sehingga Anda tidak memberi pengguna akses yang tidak diinginkan ke konten Anda atau mencegah mereka mengakses konten yang ingin tersedia bagi semua orang.

Misalnya, anggaplah permintaan tersebut sesuai dengan pola jalur untuk perilaku cache. Perilaku singgahan pertama tidak memerlukan URL atau cookie yang ditandatangani dan perilaku cache kedua. Pengguna akan dapat mengakses file tanpa menggunakan URL yang ditandatangani atau cookie yang ditandatangani karena CloudFront memproses perilaku cache yang terkait dengan kecocokan pertama.

Untuk informasi lebih lanjut tentang pola jalur, lihat [Pola jalur](#).

- Untuk distribusi yang sudah Anda gunakan untuk mendistribusikan konten, pastikan Anda siap untuk mulai membuat URL dan cookie yang ditandatangani sebelum Anda menambahkan signer. Saat Anda menambahkan tanda tangan, CloudFront tolak permintaan yang tidak menyertakan URL bertanda tangan yang valid atau cookie yang ditandatangani.

Anda dapat menambahkan tanda tangan ke distribusi menggunakan CloudFront konsol atau CloudFront API.

Console

Langkah-langkah berikut menunjukkan cara menambahkan kelompok kunci tepercaya sebagai signer. Anda juga dapat menambahkan Akun AWS sebagai penandatanganan tepercaya, tetapi tidak disarankan.

Untuk menambahkan signer ke distribusi menggunakan konsol

1. Catat ID kelompok utama dari kelompok kunci yang ingin Anda gunakan sebagai signer tepercaya. Untuk informasi selengkapnya, lihat [Buat pasangan kunci untuk kelompok kunci yang dipercaya \(disarankan\)](#).
2. Buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.

3. Pilih distribusi yang file-filenya ingin Anda lindungi dengan URL atau cookie yang ditandatangani.

 Note

Untuk menambahkan signer ke distribusi baru, Anda menentukan pengaturan yang sama yang dijelaskan di langkah 6 saat Anda membuat distribusi.

4. Pilih Perilaku tab.
5. Pilih perilaku cache yang pola jalurnya sesuai dengan file yang ingin Anda lindungi dengan URL atau cookie yang ditandatangani, lalu pilih Edit.
6. Di Edit Perilaku , lakukan hal berikut:
 - a. Untuk Batasi Akses Penampil (Gunakan URL atau Cookie yang Ditandatangani), pilih Ya.
 - b. Untuk Grup Kunci Tepercaya atau Signer Tepercaya, pilih Grup Utama yang Dipercaya.
 - c. Untuk Grup Utama yang Dipercaya, pilih grup utama untuk ditambahkan, lalu pilih Tambahkan. Ulangi jika Anda ingin menambahkan lebih dari satu grup kunci.
7. Pilih Ya, Edit untuk memperbarui perilaku cache.

API

Anda dapat menggunakan CloudFront API untuk menambahkan grup kunci tepercaya sebagai penandatanganan. Anda dapat menambahkan signer ke distribusi yang ada atau ke distribusi baru. Dalam kedua kasus, tentukan nilai dalam `TrustedKeyGroups` elemen lainnya.

Anda juga dapat menambahkan Akun AWS sebagai penandatanganan tepercaya, tetapi tidak disarankan.

Lihat topik berikut di Referensi Amazon CloudFront API:

- Perbarui distribusi yang ada — [UpdateDistribution](#)
- Buat distribusi baru — [CreateDistribution](#)

Pasangan kunci berputar

Kami menyarankan agar Anda secara berkala memutar (mengubah) pasangan kunci Anda untuk URL dan cookie yang ditandatangani. Untuk memutar pasangan kunci yang Anda gunakan untuk membuat URL atau cookie yang ditandatangani tanpa membatalkan URL atau cookie yang belum kedaluwarsa, lakukan tugas berikut:

1. Buat pasangan kunci baru, dan tambahkan kunci publik ke kelompok kunci. Untuk informasi selengkapnya, lihat [Buat pasangan kunci untuk kelompok kunci yang dipercaya \(disarankan\)](#).
2. Jika Anda membuat grup kunci baru di langkah sebelumnya, [menambahkan grup utama ke dalam distribusi sebagai penanda tangan](#).

Important

Jangan hapus kunci publik yang ada dari grup utama, atau grup utama mana pun dari distribusi. Hanya tambahkan yang baru.

3. Perbarui aplikasi Anda untuk membuat tanda tangan menggunakan kunci pribadi dari pasangan kunci baru. Konfirmasikan bahwa URL atau cookie bertanda tangan yang ditandatangani dengan kunci privat baru berfungsi baik.
4. Tunggu hingga tanggal kedaluwarsa telah meneruskan URL atau cookie yang ditandatangani menggunakan kunci privat sebelumnya. Kemudian, hapus kunci publik lama dari kelompok utama. Jika Anda membuat grup kunci baru di langkah 2, hapus grup kunci lama dari distribusi Anda.

Memutuskan untuk menggunakan URL yang ditandatangani atau cookie yang ditandatangani

CloudFront URL yang ditandatangani dan cookie yang ditandatangani menyediakan fungsionalitas dasar yang sama: mereka memungkinkan Anda untuk mengontrol siapa yang dapat mengakses konten Anda. Jika Anda ingin menyajikan konten pribadi CloudFront dan Anda mencoba memutuskan apakah akan menggunakan URL yang ditandatangani atau cookie yang ditandatangani, pertimbangkan hal berikut.

Gunakan URL yang ditandatangani dalam kasus berikut:

- Anda ingin membatasi akses ke file individual, misalnya, unduhan penginstalan untuk aplikasi Anda.

- Pengguna Anda menggunakan klien (misalnya, klien HTTP kustom) yang tidak mendukung cookie.

Gunakan cookie yang ditandatangani dalam kasus berikut:

- Anda ingin memberikan akses ke beberapa file terbatas, misalnya, semua file untuk video dalam format HLS atau semua file dalam area pelanggan di situs web.
- Anda tidak ingin mengubah URL saat ini.

Jika saat ini Anda tidak menggunakan URL yang ditandatangani, dan jika URL (yang belum ditandatangani) mengandung salah satu parameter string kueri berikut, Anda tidak dapat menggunakan URL atau cookie yang ditandatangani:

- Expires
- Policy
- Signature
- Key-Pair-Id
- Hash-Algorithm

CloudFront mengasumsikan bahwa URL yang berisi salah satu parameter string kueri tersebut adalah URL yang ditandatangani, dan karenanya tidak akan melihat cookie yang ditandatangani.

Gunakan URL yang ditandatangani dan cookie yang ditandatangani

URL yang ditandatangani lebih diutamakan dibandingkan cookie yang ditandatangani. Jika Anda menggunakan URL yang ditandatangani dan cookie yang ditandatangani untuk mengontrol akses ke file yang sama dan penampil menggunakan URL yang ditandatangani untuk meminta file, CloudFront tentukan apakah akan mengembalikan file ke penampil hanya berdasarkan URL yang ditandatangani.

Gunakan URL yang ditandatangani

URL yang ditandatangani mencakup informasi tambahan, misalnya, tanggal dan waktu kedaluwarsa, yang memberi Anda lebih banyak kendali atas akses ke konten Anda. Informasi tambahan ini muncul dalam pernyataan kebijakan, yang didasarkan pada kebijakan terekam atau kebijakan pabeian. Perbedaan antara kebijakan terekam dan kustom dijelaskan dalam dua bagian berikutnya.

 Note

Anda dapat membuat beberapa URL yang ditandatangani menggunakan kebijakan terekam dan membuat beberapa URL menggunakan kebijakan khusus untuk distribusi yang sama.

Topik

- [Memutuskan untuk menggunakan kebijakan kalengan atau kustom untuk URL yang ditandatangani](#)
- [Cara kerja URL yang ditandatangani](#)
- [Tentukan berapa lama URL yang ditandatangani valid](#)
- [Saat CloudFront memeriksa tanggal dan waktu kedaluwarsa di URL yang ditandatangani](#)
- [Kode contoh dan alat pihak ketiga](#)
- [Membuat URL yang ditandatangani menggunakan kebijakan kalengan](#)
- [Membuat URL yang ditandatangani menggunakan kebijakan khusus](#)

Memutuskan untuk menggunakan kebijakan kalengan atau kustom untuk URL yang ditandatangani

Saat Anda membuat URL yang ditandatangani, Anda menulis pernyataan kebijakan dalam format JSON yang menetapkan batasan pada URL yang ditandatangani, misalnya, berapa lama URL tersebut valid. Anda dapat menggunakan kebijakan terekam atau kebijakan bea cukai. Berikut ini adalah perbandingan kebijakan yang dapat disesuaikan dan disesuaikan:

Deskripsi	Kebijakan kalengan	Kebijakan khusus
Anda dapat menggunakan kembali pernyataan kebijakan untuk beberapa file. Untuk menggunakan kembali pernyataan kebijakan, Anda harus menggunakan karakter wildcard dalam Resource objek. Untuk informasi lebih lanjut, lihat Nilai yang Anda sebutkan dalam pernyataan kebijakan untuk URL yang ditandatangani menggunakan kebijakan khusus.)	Tidak	Ya

Deskripsi	Kebijakan kalengan	Kebijakan khusus
Anda dapat menentukan tanggal dan waktu pengguna dapat mulai mengakses konten Anda.	Tidak	Ya (opsional)
Anda dapat menentukan tanggal dan waktu saat pengguna tidak lagi dapat mengakses konten Anda.	Ya	Ya
Anda dapat menentukan alamat IP atau berbagai alamat IP pengguna yang dapat mengakses konten Anda.	Tidak	Ya (opsional)
URL yang ditandatangani mencakup versi kebijakan yang dikodekan base64, yang menghasilkan URL yang lebih panjang.	Tidak	Ya

Untuk informasi tentang pembuatan URL yang ditandatangani menggunakan kebijakan terekam, lihat [Membuat URL yang ditandatangani menggunakan kebijakan kalengan](#).

Untuk informasi tentang pembuatan URL yang ditandatangani menggunakan kebijakan terekam, lihat [Membuat URL yang ditandatangani menggunakan kebijakan khusus](#).

Cara kerja URL yang ditandatangani

Berikut adalah ikhtisar tentang cara Anda mengonfigurasi CloudFront dan Amazon S3 untuk URL yang ditandatangani dan cara CloudFront merespons saat pengguna menggunakan URL yang ditandatangani untuk meminta file.

1. Dalam CloudFront distribusi Anda, tentukan satu atau beberapa grup kunci tepercaya, yang berisi kunci publik yang CloudFront dapat digunakan untuk memverifikasi tanda tangan URL. Anda menggunakan kunci privat yang sesuai untuk menandatangani URL.

CloudFront mendukung URL yang ditandatangani dengan tanda tangan kunci RSA 2048 dan ECDSA 256.

Untuk informasi selengkapnya, lihat [Tentukan penandatanganan yang dapat membuat URL yang ditandatangani dan cookie yang ditandatangani](#).

2. Kembangkan aplikasi Anda untuk menentukan apakah pengguna harus memiliki akses ke konten Anda dan untuk membuat URL yang ditandatangani untuk file atau bagian aplikasi yang ingin dibatasi aksesnya. Untuk informasi selengkapnya, lihat topik berikut:
 - [Membuat URL yang ditandatangani menggunakan kebijakan kalengan](#)
 - [Membuat URL yang ditandatangani menggunakan kebijakan khusus](#)
3. Seorang pengguna meminta file yang memerlukan URL yang ditandatangani.
4. Aplikasi Anda memverifikasi bahwa pengguna berhak mengakses file: mereka telah masuk, mereka telah membayar akses ke konten, atau mereka telah memenuhi beberapa persyaratan lain untuk akses.
5. Aplikasi Anda membuat dan mengembalikan URL yang ditandatangani ke pengguna.
6. URL yang ditandatangani memungkinkan pengguna mengunduh atau men-streaming konten.

Langkah ini bersifat otomatis; pengguna biasanya tidak perlu melakukan tindakan tambahan apa pun untuk mengakses konten. Misalnya, jika pengguna mengakses konten Anda di peramban web, aplikasi akan mengembalikan URL yang ditandatangani ke peramban. Browser segera menggunakan URL yang ditandatangani untuk mengakses file di cache CloudFront tepi tanpa campur tangan dari pengguna.

7. CloudFront menggunakan kunci publik untuk memvalidasi tanda tangan dan mengonfirmasi bahwa URL belum dirusak. Jika tanda tangan tidak valid, permintaan ditolak.

Jika tanda tangan valid, CloudFront lihat pernyataan kebijakan di URL (atau buat jika Anda menggunakan kebijakan kalengan) untuk mengonfirmasi bahwa permintaan tersebut masih valid. Misalnya, jika Anda menentukan tanggal dan waktu awal dan akhir untuk URL, CloudFront konfirmasikan bahwa pengguna mencoba mengakses konten Anda selama periode waktu yang ingin Anda izinkan akses.

Jika permintaan memenuhi persyaratan dalam pernyataan kebijakan, CloudFront lakukan operasi standar: menentukan apakah file sudah ada di cache tepi, meneruskan permintaan ke asal jika perlu, dan mengembalikan file ke pengguna.

Note

Jika URL yang belum ditandatangani memuat parameter string kueri, pastikan Anda menyertakannya di bagian URL yang Anda tanda tangani. Jika Anda menambahkan string

kueri ke URL yang ditandatangani setelah menandatanganinya, URL akan mengembalikan status HTTP 403.

Tentukan berapa lama URL yang ditandatangani valid

Anda dapat mendistribusikan konten pribadi menggunakan URL bertanda tangan yang hanya berlaku sebentar—mungkin hanya selama beberapa menit. URL yang ditandatangani yang berlaku untuk periode waktu singkat tersebut baik untuk mendistribusikan konten saat ini kepada pengguna untuk tujuan tertentu, seperti mendistribusikan sewa film atau unduhan musik kepada pelanggan sesuai permintaan. Jika URL yang ditandatangani Anda hanya akan berlaku untuk waktu singkat, Anda mungkin ingin membuatnya secara otomatis menggunakan aplikasi yang Anda kembangkan. Saat pengguna mulai mengunduh file atau mulai memutar file media, CloudFront bandingkan waktu kedaluwarsa di URL dengan waktu saat ini untuk menentukan apakah URL tersebut masih valid.

Anda juga dapat mendistribusikan konten pribadi menggunakan URL bertanda tangan yang valid untuk waktu yang lebih lama, mungkin selama bertahun-tahun. URL yang ditandatangani yang valid untuk periode yang lebih lama berguna untuk mendistribusikan konten privat kepada pengguna yang dikenal, seperti mendistribusikan rencana bisnis kepada investor atau mendistribusikan materi pelatihan kepada karyawan. Anda dapat mengembangkan aplikasi untuk membuat URL yang ditandatangani jangka panjang untuk Anda.

Saat CloudFront memeriksa tanggal dan waktu kedaluwarsa di URL yang ditandatangani

CloudFront memeriksa tanggal kedaluwarsa dan waktu dalam URL yang ditandatangani pada saat permintaan HTTP. Jika klien mulai mengunduh file besar segera sebelum waktu kedaluwarsa, pengunduhan harus selesai meskipun waktu kedaluwarsa sudah lewat selama pengunduhan. Jika koneksi TCP menurun dan klien mencoba memulai ulang unduhan setelah waktu kedaluwarsa berlalu, pengunduhan akan gagal.

Jika klien menggunakan Range GETs untuk mendapatkan file dalam potongan yang lebih kecil, setiap permintaan GET yang terjadi setelah waktu kedaluwarsa akan gagal. Untuk informasi lebih lanjut tentang Range GET, lihat [Bagaimana CloudFront memproses permintaan sebagian untuk suatu objek \(rentang GETs\)](#).

Kode contoh dan alat pihak ketiga

Misalnya, kode yang membuat bagian yang di-hash dan ditandatangani dari URL yang ditandatangani, lihat topik berikut:

- [Buat tanda tangan URL menggunakan Perl](#)
- [Buat tanda tangan URL menggunakan PHP](#)
- [Buat tanda tangan URL menggunakan C# dan .NET Framework](#)
- [Buat tanda tangan URL menggunakan Java](#)

Membuat URL yang ditandatangani menggunakan kebijakan kalengan

Untuk membuat URL yang ditandatangani menggunakan kebijakan terekam, lakukan langkah-langkah berikut.

Untuk membuat URL yang ditandatangani menggunakan kebijakan terekam

1. Jika Anda menggunakan .NET atau Java untuk membuat URL yang ditandatangani, dan jika Anda belum memformat ulang kunci privat untuk pasangan kunci dari format .pemiksa default ke format yang kompatibel dengan .NET atau Java, lakukan sekarang. Untuk informasi selengkapnya, lihat [Memformat ulang kunci pribadi \(hanya .NET dan Java\)](#).
2. Menggabungkan nilai-nilai berikut. Anda dapat menggunakan format dalam contoh URL yang ditandatangani ini.

```
https://d1111111abcdef8.cloudfront.net/  
image.jpg?color=red&size=medium&Expires=1767290400&Signature=nitfHRCrtziw02HwPfWw~yYDhUF5Ew  
j19DzZrvDh6hQ73LDx~-ar3UocvvRQVw6EkC~GdpGQyy0SKQim-  
TxAnW7d8F5Kkai9HVx0FIu-5jcQb0UEmatEXAMPLE3ReXySpLSMj0yCd3ZAB4UcBCAqEijkytL6f3fVYNGQI6&Key-  
Pair-Id=K2JCJMDEHXQW5F&Hash-Algorithm=SHA256
```

Hapus semua spasi kosong (termasuk tab dan karakter baris baru). Anda mungkin harus memasukkan karakter escape dalam string di kode aplikasi. Semua nilai memiliki tipeString.

1. **Base URL for the file**

URL dasar adalah CloudFront URL yang akan Anda gunakan untuk mengakses file jika Anda tidak menggunakan URL yang ditandatangani, termasuk parameter string kueri Anda sendiri, jika ada. Pada contoh sebelumnya, URL dasar adalah. `https://`

d111111abcdef8.cloudfront.net/image.jpg Untuk informasi lebih lanjut tentang format URL untuk distribusi, lihat [Sesuaikan format URL untuk file di CloudFront](#).

- CloudFront URL berikut adalah untuk file gambar dalam distribusi (menggunakan nama CloudFront domain). Perhatikan bahwa `image.jpg` dalam `images` direktori. Jalur menuju file dalam URL harus sesuai dengan alur menuju file pada server HTTP Anda atau pada bucket Amazon S3.

```
https://d111111abcdef8.cloudfront.net/images/image.jpg
```

- CloudFront URL berikut mencakup string kueri:

```
https://d111111abcdef8.cloudfront.net/images/image.jpg?size=large
```

- CloudFront URL berikut adalah untuk file gambar dalam distribusi. Keduanya menggunakan nama domain alternatif. Yang kedua mencakup string kueri:

```
https://www.example.com/images/image.jpg
```

```
https://www.example.com/images/image.jpg?color=red
```

- CloudFront URL berikut adalah untuk file gambar dalam distribusi yang menggunakan nama domain alternatif dan protokol HTTPS:

```
https://www.example.com/images/image.jpg
```

2. ?

?Ini menunjukkan bahwa parameter kueri mengikuti URL dasar. Sertakan ? bahkan jika Anda tidak menentukan parameter kueri apa pun.

Note

Anda dapat menentukan parameter kueri berikut dalam urutan apa pun.

3. *Your query string parameters, if any*&

(Opsional) Anda dapat memasukkan parameter string kueri Anda sendiri. Untuk melakukannya, tambahkan ampersand (&) di antara masing-masing, seperti. `color=red&size=medium` Anda dapat menentukan parameter string kueri dalam urutan apa pun dalam URL.

⚠ Important

Parameter string kueri Anda tidak dapat diberi nama `Expires`, `Signature`, `Key-Pair-Id`, atau `Hash-Algorithm`.

4. `Expires`=*date and time in Unix time format (in seconds) and Coordinated Universal Time (UTC)*

Tanggal dan waktu Anda ingin URL berhenti memungkinkan akses ke file.

Tentukan tanggal dan waktu kedaluwarsa dalam format waktu Unix (dalam detik) dan Waktu Universal Terkoordinasi (UTC). Misalnya, 1 Januari 2026 10:00 UTC dikonversi ke 1767290400 dalam format waktu Unix, seperti yang ditunjukkan pada contoh di awal topik ini.

Untuk menggunakan waktu epoch, tentukan bilangan bulat 64-bit untuk tanggal yang paling lambat 9223372036854775807 (Jumat, 11 April 2262 pukul 23:47:16.854 UTC).

Untuk informasi tentang UTC, lihat [RFC 3339, Tanggal dan Waktu di Internet: Stempel Waktu](#).

5. `&Signature`=*hashed and signed version of the policy statement*

Versi yang di-hash, ditandatangani, dan dikodekan base64 dari pernyataan kebijakan JSON. Untuk informasi selengkapnya, lihat [Membuat tanda tangan untuk URL yang ditandatangani yang menggunakan kebijakan kalengan](#).

6. `&Key-Pair-Id`=*public key ID for the CloudFront public key whose corresponding private key you're using to generate the signature*

ID untuk kunci CloudFront publik, misalnya, K2JCJMDEHXQW5F. ID kunci publik memberi tahu kunci publik CloudFront mana yang akan digunakan untuk memvalidasi URL yang ditandatangani. CloudFront membandingkan informasi dalam tanda tangan dengan informasi dalam pernyataan kebijakan untuk memverifikasi bahwa URL belum dirusak.

Kunci publik ini harus dimiliki oleh kelompok kunci yang merupakan signer tepercaya dalam distribusi. Untuk informasi selengkapnya, lihat [Tentukan penandatanganan yang dapat membuat URL yang ditandatangani dan cookie yang ditandatangani](#).

7. &Hash-Algorithm=**SHA1 or SHA256**

(Opsional) Algoritma hash yang digunakan untuk membuat tanda tangan. Nilai yang didukung adalah SHA1 dan SHA256. Jika Anda tidak menentukan parameter ini, CloudFront defaultnya. SHA1

Membuat tanda tangan untuk URL yang ditandatangani yang menggunakan kebijakan kalengan

Untuk membuat tanda tangan untuk URL yang ditandatangani yang menggunakan kebijakan kalengan, selesaikan prosedur berikut.

Topik

- [Membuat pernyataan kebijakan untuk URL yang ditandatangani yang menggunakan kebijakan kalengan](#)
- [Membuat tanda tangan untuk URL yang ditandatangani yang menggunakan kebijakan kalengan](#)

Membuat pernyataan kebijakan untuk URL yang ditandatangani yang menggunakan kebijakan kalengan

Saat Anda membuat URL yang ditandatangani menggunakan kebijakan terekam, Signature parameter adalah versi dokumen pernyataan kebijakan yang di-hash dan ditandatangani. Untuk URL yang ditandatangani yang menggunakan kebijakan terekam, Anda tidak menyertakan pernyataan kebijakan di URL, seperti yang Anda lakukan untuk URL yang ditandatangani dengan kebijakan khusus. Untuk membuat pernyataan kebijakan, lakukan prosedur berikut.

Untuk membuat pernyataan kebijakan untuk URL yang ditandatangani menggunakan kebijakan terekam

1. Membangun pernyataan kebijakan menggunakan format JSON berikut dan menggunakan pengkodean UTF-8 karakter. Sertakan semua tanda baca dan nilai literal lainnya persis seperti yang ditentukan. Untuk informasi tentang Resource dan DateLessThan parameter, lihat [Nilai yang Anda sebutkan dalam pernyataan kebijakan untuk URL yang ditandatangani dengan menggunakan kebijakan terekam](#).

```
{
  "Statement": [
    {
      "Resource": "base URL or stream name",
```

```

        "Condition": {
            "DateLessThan": {
                "AWS:EpochTime": ending date and time in Unix time format and
                UTC
            }
        }
    ]
}

```

2. Hapus semua spasi kosong (termasuk tab dan karakter baris baru) dari pernyataan kebijakan. Anda mungkin harus memasukkan karakter escape dalam string di kode aplikasi.

Nilai yang Anda sebutkan dalam pernyataan kebijakan untuk URL yang ditandatangani dengan menggunakan kebijakan terekam

Ketika Anda membuat pernyataan kebijakan untuk kebijakan terekam, Anda menentukan nilai-nilai berikut.

Sumber Daya

Note

Anda hanya dapat menentukan satu nilai untuk Resource.

URL dasar termasuk string kueri Anda, jika ada, tetapi tidak termasuk CloudFront Expires,, SignatureKey-Pair-Id, dan Hash-Algorithm parameter, misalnya:

```
https://d111111abcdef8.cloudfront.net/images/horizon.jpg?
size=large&license=yes
```

Perhatikan hal-hal berikut:

- Protokol – Nilai harus dimulai dengan `http://` atau `https://`.
- Parameter string kueri – Jika Anda tidak memiliki parameter string pencarian, hapus tanda tanya.
- Nama domain alternatif – Jika Anda menentukan nama domain alternatif (CNAME) di URL, Anda harus menentukan nama domain alternatif saat merujuk file di halaman web atau aplikasi Anda. Jangan menentukan URL Amazon S3 untuk objek tersebut.

DateLessThan

Tanggal dan waktu kedaluwarsa untuk URL dalam format waktu Unix (dalam detik) dan Waktu Universal Terkoordinasi (UTC). Misalnya, 1 Januari 2026 10:00 UTC mengkonversi ke 1767290400 dalam format waktu Unix.

Nilai ini harus cocok dengan nilai Expires parameter string kueri dalam URL yang ditandatangani. Jangan melampirkan nilai dalam tanda petik.

Untuk informasi selengkapnya, lihat [Saat CloudFront memeriksa tanggal dan waktu kedaluwarsa di URL yang ditandatangani](#).

Contoh pernyataan kebijakan untuk URL yang ditandatangani yang menggunakan kebijakan terekam

Saat Anda menggunakan contoh pernyataan kebijakan berikut di URL yang ditandatangani, pengguna dapat mengakses file tersebut `https://d111111abcdef8.cloudfront.net/horizon.jpg` hingga 1 Januari 2026 10:00 UTC:

```
{
  "Statement": [
    {
      "Resource": "https://d111111abcdef8.cloudfront.net/horizon.jpg?size=large&license=yes",
      "Condition": {
        "DateLessThan": {
          "AWS:EpochTime": 1767290400
        }
      }
    }
  ]
}
```

Membuat tanda tangan untuk URL yang ditandatangani yang menggunakan kebijakan kalengan

Untuk membuat nilai untuk Signature parameter dalam URL yang ditandatangani, Anda telah dan menandatangani pernyataan kebijakan yang Anda buat di [Membuat pernyataan kebijakan untuk URL yang ditandatangani yang menggunakan kebijakan kalengan](#).

Untuk informasi tambahan dan contoh cara membuat, menandatangani, dan mengkode pernyataan kebijakan, lihat:

- [Perintah Linux dan OpenSSL untuk pengkodean dan enkripsi base64](#)
- [Contoh kode untuk membuat tanda tangan untuk URL yang ditandatangani](#)

 Note

Contoh yang ditautkan digunakan secara SHA-1 default. Untuk menggunakannya, SHA-256 ganti sha1 dengan sha256 perintah OpenSSL dan sertakan parameter kueri Hash-Algorithm=SHA256 di URL yang ditandatangani.

Opsi 1: Untuk membuat tanda tangan dengan menggunakan kebijakan terekam

1. Gunakan fungsi SHA-1 atau SHA-256 hash dan kunci pribadi RSA atau ECDSA yang dihasilkan untuk hash dan tandatangani pernyataan kebijakan yang Anda buat dalam prosedur. [Untuk membuat pernyataan kebijakan untuk URL yang ditandatangani menggunakan kebijakan terekam](#) Gunakan versi pernyataan kebijakan yang tidak lagi menyertakan spasi kosong.

Jika Anda menggunakan SHA-256, Anda harus menyertakan &Hash-Algorithm=SHA256 dalam URL yang ditandatangani.

Untuk kunci privat yang diperlukan oleh fungsi hash, gunakan kunci pribadi yang kunci publiknya berada dalam grup kunci yang dipercaya aktif untuk distribusi.

 Note

Metode yang Anda gunakan untuk men-emuk dan menandatangani pernyataan kebijakan tergantung pada bahasa pemrograman dan platform Anda. Untuk kode sampel, lihat [Contoh kode untuk membuat tanda tangan untuk URL yang ditandatangani](#).

2. Hapus spasi kosong (termasuk tab dan karakter baris baru) dari string hash dan ditandatangani.
3. Base64-encode string menggunakan pengkodean MIME base64. Untuk informasi selengkapnya, lihat [Bagian 6.8, Base64 Content-Transfer-Encoding](#) di RFC 2045, MIME (Ekstensi Surat Internet Serbaguna) Bagian Satu: Format Badan Pesan Internet.
4. Ganti karakter yang tidak valid dalam string kueri URL dengan karakter yang valid. Tabel berikut mencantumkan karakter yang tidak valid dan valid.

Ganti karakter tidak valid ini	Dengan karakter valid ini
+	- (tanda hubung)
=	_ (garis bawah)
/	~ (tilde)

5. Tambahkan nilai yang dihasilkan ke URL Anda yang ditandatangani setelah `&Signature=`, dan kembali ke [Untuk membuat URL yang ditandatangani menggunakan kebijakan terekam](#) untuk menyelesaikan penyatuan bagian URL yang Anda tanda tangani.

Membuat URL yang ditandatangani menggunakan kebijakan khusus

Untuk membuat URL yang ditandatangani menggunakan kebijakan kustom, selesaikan prosedur berikut.

Untuk membuat URL yang ditandatangani menggunakan kebijakan kustom

1. Jika Anda menggunakan .NET atau Java untuk membuat URL yang ditandatangani, dan jika Anda belum memformat ulang kunci privat untuk pasangan kunci dari format .pemiksa default ke format yang kompatibel dengan .NET atau Java, lakukan sekarang. Untuk informasi selengkapnya, lihat [Memformat ulang kunci pribadi \(hanya .NET dan Java\)](#).
2. Menggabungkan nilai-nilai berikut. Anda dapat menggunakan format dalam contoh URL yang ditandatangani ini.

```
https://d111111abcdef8.cloudfront.net/
image.jpg?color=red&size=medium&Policy=eyJANCIAGICEXAMPLEW1lbnQiOiBbeyANCiAgICAgICJSZXNvdXJj
j19DzZrvDh6hQ73lDx~-ar3UocvvRQVw6EkC~GdpGQyy0SKQim-
TxAnW7d8F5Kkai9HVx0FIu-5jcQb0UEmatEXAMPLE3ReXySpLSMj0yCd3ZAB4UcBCAqEijkytL6f3fVYNGQI6&Key-
Pair-Id=K2JCJMDEHXQW5F&Hash-Algorithm=SHA256
```

Hapus semua spasi kosong (termasuk tab dan karakter baris baru). Anda mungkin harus memasukkan karakter escape dalam string di kode aplikasi. Semua nilai memiliki tipe `String`.

1. *Base URL for the file*

URL dasar adalah CloudFront URL yang akan Anda gunakan untuk mengakses file jika Anda tidak menggunakan URL yang ditandatangani, termasuk parameter string kueri Anda sendiri, jika ada. Pada contoh sebelumnya, URL dasar adalah. `https://d111111abcdef8.cloudfront.net/image.jpg` Untuk informasi lebih lanjut tentang format URL untuk distribusi, lihat [Sesuaikan format URL untuk file di CloudFront](#).

Contoh berikut menunjukkan nilai yang Anda tentukan untuk distribusi.

- CloudFront URL berikut adalah untuk file gambar dalam distribusi (menggunakan nama CloudFront domain). Perhatikan bahwa `image.jpg` dalam `images` direktori. Jalur menuju file dalam URL harus sesuai dengan alur menuju file pada server HTTP Anda atau pada bucket Amazon S3.

```
https://d111111abcdef8.cloudfront.net/images/image.jpg
```

- CloudFront URL berikut mencakup string kueri:

```
https://d111111abcdef8.cloudfront.net/images/image.jpg?size=large
```

- CloudFront URL berikut adalah untuk file gambar dalam distribusi. Keduanya menggunakan nama domain alternatif; yang kedua menyertakan string kueri:

```
https://www.example.com/images/image.jpg
```

```
https://www.example.com/images/image.jpg?color=red
```

- CloudFront URL berikut adalah untuk file gambar dalam distribusi yang menggunakan nama domain alternatif dan protokol HTTPS:

```
https://www.example.com/images/image.jpg
```

2. ?

?Ini menunjukkan bahwa parameter string kueri mengikuti URL dasar. Sertakan ? bahkan jika Anda tidak menentukan parameter kueri apa pun.

Note

Anda dapat menentukan parameter kueri berikut dalam urutan apa pun.

3. ***Your query string parameters, if any***&

(Opsional) Anda dapat memasukkan parameter string kueri Anda sendiri. Untuk melakukannya, tambahkan ampersand (&) di antara masing-masing, seperti. `color=red&size=medium` Anda dapat menentukan parameter string kueri dalam urutan apa pun dalam URL.

Important

Parameter string kueri Anda tidak dapat diberi nama `Policy`, `Signature`, `Key-Pair-Id`, atau `Hash-Algorithm`.

Jika Anda menambahkan parameter Anda sendiri, tambahkan & setelah masing-masing, termasuk yang terakhir.

4. ***Policy=base64 encoded version of policy statement***

Pernyataan kebijakan Anda dalam format JSON, dengan spasi kosong dihapus, lalu base64 dikodekan. Untuk informasi selengkapnya, lihat [Membuat pernyataan kebijakan untuk URL yang ditandatangani yang menggunakan kebijakan kustom](#).

Pernyataan kebijakan mengontrol akses yang diberikan oleh URL yang ditandatangani kepada pengguna. Ini mencakup URL file, tanggal dan waktu kedaluwarsa, tanggal dan waktu opsional di mana URL menjadi valid, dan alamat IP opsional atau rentang alamat IP yang diizinkan untuk mengakses file.

5. ***&Signature=hashed and signed version of the policy statement***

Versi yang di-hash, ditandatangani, dan dikodekan base64 dari pernyataan kebijakan JSON. Untuk informasi selengkapnya, lihat [Membuat tanda tangan untuk URL bertanda tangan yang menggunakan kebijakan kustom](#).

6. ***&Key-Pair-Id=public key ID for the CloudFront public key whose corresponding private key you're using to generate the signature***

ID untuk kunci CloudFront publik, misalnya, `K2JCJMDEHXQW5F`. ID kunci publik memberi tahu kunci publik CloudFront mana yang akan digunakan untuk memvalidasi URL yang

ditandatangani. CloudFront membandingkan informasi dalam tanda tangan dengan informasi dalam pernyataan kebijakan untuk memverifikasi bahwa URL belum dirusak.

Kunci publik ini harus dimiliki oleh kelompok kunci yang merupakan signer tepercaya dalam distribusi. Untuk informasi selengkapnya, lihat [Tentukan penandatanganan yang dapat membuat URL yang ditandatangani dan cookie yang ditandatangani](#).

7. &Hash-Algorithm=**SHA1 or SHA256**

(Opsional) Algoritma hash yang digunakan untuk membuat tanda tangan. Nilai yang didukung adalah SHA1 dan SHA256. Jika Anda tidak menentukan parameter ini, CloudFront defaultnya. SHA1

Membuat pernyataan kebijakan untuk URL yang ditandatangani yang menggunakan kebijakan kustom

Selesaikan langkah-langkah berikut untuk membuat pernyataan kebijakan untuk URL yang ditandatangani yang menggunakan kebijakan khusus.

Misalnya pernyataan kebijakan yang mengontrol akses ke file dalam berbagai cara, lihat [the section called “Contoh pernyataan kebijakan untuk URL yang ditandatangani menggunakan kebijakan kustom”](#).

Untuk membuat pernyataan kebijakan untuk URL yang ditandatangani menggunakan kebijakan kustom

1. Buat pernyataan kebijakan dengan menggunakan format JSON berikut. Ganti simbol kurang dari (<) dan lebih besar dari (>), dan deskripsi di dalamnya, dengan nilai Anda sendiri. Untuk informasi selengkapnya, lihat [the section called “Nilai yang Anda sebutkan dalam pernyataan kebijakan untuk URL yang ditandatangani menggunakan kebijakan khusus”](#).

```
{
  "Statement": [
    {
      "Resource": "<Optional but recommended: URL of the file>",
      "Condition": {
        "DateLessThan": {
          "AWS:EpochTime": <Required: ending date and time in Unix time
format and UTC>
        },
        "DateGreaterThan": {
```

```

        "AWS:EpochTime": <Optional: beginning date and time in Unix time
format and UTC>
    },
    "IpAddress": {
        "AWS:SourceIp": "<Optional: IP address>"
    }
}
]
}

```

Perhatikan hal-hal berikut:

- Anda hanya dapat memasukkan satu pernyataan dalam kebijakan.
 - Gunakan pengkodean UTF-8 karakter.
 - Sertakan semua nama tanda baca dan parameter persis seperti yang ditentukan. Singkatan untuk nama parameter tidak diterima.
 - Urutan parameter di Condition tidak masalah.
 - Untuk informasi tentang nilai untuk Resource, DateLessThan, DateGreaterThan, dan IpAddress, lihat [the section called “Nilai yang Anda sebutkan dalam pernyataan kebijakan untuk URL yang ditandatangani menggunakan kebijakan khusus”](#).
2. Hapus semua spasi kosong (termasuk tab dan karakter baris baru) dari pernyataan kebijakan. Anda mungkin harus memasukkan karakter escape dalam string di kode aplikasi.
 3. Base64-encode pernyataan kebijakan menggunakan pengkodean MIME base64. Untuk informasi selengkapnya, lihat [Bagian 6.8, Base64 Content-Transfer-Encoding](#) di RFC 2045, MIME (Ekstensi Surat Internet Serbaguna) Bagian Satu: Format Badan Pesan Internet.
 4. Ganti karakter yang tidak valid dalam string kueri URL dengan karakter yang valid. Tabel berikut mencantumkan karakter yang tidak valid dan valid.

Ganti karakter tidak valid ini	Dengan karakter valid ini
+	- (tanda hubung)
=	_ (garis bawah)
/	~ (tilde)

5. Tambahkan nilai yang dihasilkan ke URL Anda yang ditandatangani setelah Policy=.

6. Buat tanda tangan untuk URL yang ditandatangani dengan melakukan, menandatangani, dan memberikan kode dasar64 untuk pernyataan kebijakan. Untuk informasi selengkapnya, lihat [the section called “Membuat tanda tangan untuk URL bertanda tangan yang menggunakan kebijakan kustom”](#).

Nilai yang Anda sebutkan dalam pernyataan kebijakan untuk URL yang ditandatangani menggunakan kebijakan khusus

Saat Anda membuat pernyataan kebijakan untuk kebijakan kustom, Anda menentukan nilai berikut.

Sumber daya

URL, termasuk string kueri apa pun, tetapi tidak termasuk CloudFront Policy,, SignatureKey-Pair-Id, dan Hash-Algorithm parameter. Contoh:

```
https://d1111111abcdef8.cloudfront.net/images/horizon.jpg?size=large&license=yes
```

Anda hanya dapat menentukan satu nilai URL untuk Resource.

 Important

Anda dapat menghilangkan Resource parameter dalam kebijakan, tetapi melakukannya berarti siapa pun yang memiliki URL yang ditandatangani dapat mengakses semua file dalam distribusi apa pun yang terkait dengan key pair yang Anda gunakan untuk membuat URL yang ditandatangani.

Perhatikan hal-hal berikut:

- Protokol – Nilai harus dimulai dengan http://, https://, atau *://.
- Parameter string kueri - Jika URL memiliki parameter string kueri, jangan gunakan karakter garis miring terbalik (\) untuk menghindari karakter tanda tanya (?) yang memulai string kueri. Contoh:

```
https://d1111111abcdef8.cloudfront.net/images/horizon.jpg?size=large&license=yes
```

- Karakter wildcard — Anda dapat menggunakan karakter wildcard di URL dalam kebijakan. Karakter wildcard berikut didukung:

- asterisk (*), yang cocok dengan nol atau lebih karakter
- tanda tanya (?), yang cocok persis dengan satu karakter

Saat CloudFront mencocokkan URL dalam kebijakan dengan URL dalam permintaan HTTP, URL dalam kebijakan dibagi menjadi empat bagian—protokol, domain, jalur, dan string kueri—sebagai berikut:

`[protocol]://[domain]/[path]\?[query string]`

Bila Anda menggunakan karakter wildcard di URL dalam kebijakan, pencocokan wildcard hanya berlaku dalam batas bagian yang berisi wildcard. Misalnya, pertimbangkan URL ini dalam kebijakan:

`https://www.example.com/hello*world`

Dalam contoh ini, wildcard asterisk (*) hanya berlaku di bagian jalur, sehingga cocok dengan URL `https://www.example.com/helloworld` dan `https://www.example.com/hello-world`, tetapi tidak cocok dengan URL `https://www.example.net/hello?world`

Pengecualian berikut berlaku untuk batas bagian untuk pencocokan wildcard:

- Tanda bintang tertinggal di bagian jalur menyiratkan tanda bintang di bagian string kueri. Misalnya, `http://example.com/hello*` setara dengan `http://example.com/hello*\?*`.
- Tanda bintang di bagian domain menyiratkan tanda bintang di bagian jalur dan string kueri. Misalnya, `http://example.com*` setara dengan `http://example.com/*\?*`.
- URL dalam kebijakan dapat menghilangkan bagian protokol dan memulai dengan tanda bintang di bagian domain. Dalam hal ini, bagian protokol secara implisit diatur ke tanda bintang. Misalnya, URL `*example.com` dalam kebijakan setara dengan `*://example.com/`.
- Tanda bintang dengan sendirinya ("Resource": "*") cocok dengan URL apa pun.

Misalnya, nilai `https://d111111abcdef8.cloudfront.net/*game_download.zip*` dalam kebijakan cocok dengan semua URL berikut:

- `https://d111111abcdef8.cloudfront.net/game_download.zip`
- `https://d111111abcdef8.cloudfront.net/example_game_download.zip?license=yes`

- `https://d111111abcdef8.cloudfront.net/test_game_download.zip?license=temp`
- Nama domain alternatif — Jika Anda menentukan nama domain alternatif (CNAME) di URL dalam kebijakan, permintaan HTTP harus menggunakan nama domain alternatif di halaman web atau aplikasi Anda. Jangan tentukan URL Amazon S3 untuk file dalam kebijakan.

DateLessThan

Tanggal dan waktu kedaluwarsa untuk URL dalam format waktu Unix (dalam detik) dan Waktu Universal Terkoordinasi (UTC). Dalam kebijakan, jangan lampirkan nilai dalam tanda kutip. Untuk informasi tentang UTC, lihat [Tanggal dan Waktu di Internet: Stempel Waktu](#).

Misalnya, 31 Januari 2023 10:00 UTC dikonversi ke 1675159200 dalam format waktu Unix.

Ini adalah satu-satunya parameter yang diperlukan di `Condition` bagian ini. CloudFront memerlukan nilai ini untuk mencegah pengguna memiliki akses permanen ke konten pribadi Anda.

Untuk informasi selengkapnya, lihat [the section called “Saat CloudFront memeriksa tanggal dan waktu kedaluwarsa di URL yang ditandatangani”](#)

DateGreaterThan (Opsional)

Tanggal dan waktu mulai opsional untuk URL dalam format waktu Unix (dalam detik) dan Waktu Universal Terkoordinasi (UTC). Pengguna tidak diizinkan untuk mengakses file pada atau sebelum tanggal dan waktu yang ditentukan. Jangan melampirkan nilai dalam tanda petik.

IpAddress (Opsional)

Alamat IP klien membuat permintaan HTTP. Perhatikan hal-hal berikut:

- Untuk mengizinkan alamat IP mengakses file, hapus `IpAddress` parameter.
- Anda dapat menentukan salah satu alamat IP atau satu rentang alamat IP. Anda tidak dapat menggunakan kebijakan untuk mengizinkan akses jika alamat IP klien berada di salah satu dari dua rentang terpisah.
- Untuk memungkinkan akses dari satu alamat IP, Anda menentukan:

`"IPv4 IP address/32"`

- Anda harus menentukan rentang alamat IP dalam format IPv4 CIDR standar (misalnya, `192.0.2.0/24`). Untuk informasi selengkapnya, lihat [Classless Inter-domain Routing \(CIDR\): Rencana Penetapan dan Agregasi Alamat Internet](#).

 Important

Alamat IP dalam format IPv6, seperti 2001:0 db 8:85 a3: :8a2e: 0370:7334, tidak didukung.

Jika Anda menggunakan kebijakan khusus yang mencakup `IpAddress`, jangan mengaktifkan IPv6 untuk distribusi. Jika Anda ingin membatasi akses ke sebagian konten dengan alamat IP dan mendukung permintaan IPv6 untuk konten lain, Anda dapat membuat dua distribusi. Untuk informasi lebih lanjut, lihat [the section called “Aktifkan IPv6 \(permintaan penampil\)”](#) dalam topik [the section called “Semua pengaturan distribusi”](#).

Contoh pernyataan kebijakan untuk URL yang ditandatangani menggunakan kebijakan kustom

Contoh pernyataan kebijakan berikut menunjukkan cara mengontrol akses ke file tertentu, semua file di direktori, atau semua file yang terkait dengan ID pasangan kunci. Contoh ini juga menunjukkan cara mengontrol akses dari alamat IP individu atau serangkaian alamat IP, dan cara mencegah pengguna menggunakan URL yang ditandatangani setelah tanggal dan waktu yang ditentukan.

Jika Anda menyalin dan menempelkan salah satu contoh ini, hapus spasi kosong (termasuk tab dan karakter baris baru), ganti nilai dengan nilai Anda sendiri, dan sertakan karakter baris baru setelah tanda kurung kurung penutup `()`. `}`

Untuk informasi selengkapnya, lihat [the section called “Nilai yang Anda sebutkan dalam pernyataan kebijakan untuk URL yang ditandatangani menggunakan kebijakan khusus”](#).

Topik

- [Contoh pernyataan kebijakan: Akses satu file dari berbagai alamat IP](#)
- [Contoh pernyataan kebijakan: Akses semua file dalam direktori dari berbagai alamat IP](#)
- [Contoh pernyataan kebijakan: Akses semua file yang terkait dengan ID key pair dari satu alamat IP](#)

Contoh pernyataan kebijakan: Akses satu file dari berbagai alamat IP

Contoh kebijakan kustom berikut dalam URL yang ditandatangani menetapkan bahwa pengguna dapat mengakses file `https://d111111abcdef8.cloudfront.net/game_download.zip` dari alamat IP dalam rentang `192.0.2.0/24` hingga 31 Januari 2023 10:00 UTC:

```
{
  "Statement": [
    {
      "Resource": "https://d1111111abcdef8.cloudfront.net/game_download.zip",
      "Condition": {
        "IpAddress": {
          "AWS:SourceIp": "192.0.2.0/24"
        },
        "DateLessThan": {
          "AWS:EpochTime": 1675159200
        }
      }
    }
  ]
}
```

Contoh pernyataan kebijakan: Akses semua file dalam direktori dari berbagai alamat IP

Contoh kebijakan kustom berikut memungkinkan Anda membuat URL yang ditandatangani untuk file apa pun di `training` direktori, seperti yang ditunjukkan oleh karakter wildcard asterisk (*) dalam parameter. Pengguna dapat mengakses file dari alamat IP dalam kisaran `192.0.2.0/24` hingga 31 Januari 2023 10:00 UTC:

```
{
  "Statement": [
    {
      "Resource": "https://d1111111abcdef8.cloudfront.net/training/*",
      "Condition": {
        "IpAddress": {
          "AWS:SourceIp": "192.0.2.0/24"
        },
        "DateLessThan": {
          "AWS:EpochTime": 1675159200
        }
      }
    }
  ]
}
```

Setiap URL yang ditandatangani tempat Anda menggunakan kebijakan ini memiliki URL yang mengidentifikasi file tertentu, misalnya:

`https://d1111111abcdef8.cloudfront.net/training/orientation.pdf`

Contoh pernyataan kebijakan: Akses semua file yang terkait dengan ID key pair dari satu alamat IP

Contoh kebijakan kustom berikut memungkinkan Anda membuat URL yang ditandatangani untuk file apa pun yang terkait dengan distribusi apa pun, seperti yang ditunjukkan oleh karakter wildcard asterisk (*) dalam parameter. Resource URL yang ditandatangani harus menggunakan `https://` protokol, bukan `http://`. Pengguna harus menggunakan alamat IP `192.0.2.10/32`. (Nilai `192.0.2.10/32` dalam notasi CIDR mengacu pada alamat IP tunggal, `192.0.2.10`.) File hanya tersedia mulai 31 Januari 2023 10:00 UTC hingga 2 Februari 2023 10:00 UTC:

```
{
  "Statement": [
    {
      "Resource": "https://*",
      "Condition": {
        "IpAddress": {
          "AWS:SourceIp": "192.0.2.10/32"
        },
        "DateGreaterThan": {
          "AWS:EpochTime": 1675159200
        },
        "DateLessThan": {
          "AWS:EpochTime": 1675332000
        }
      }
    }
  ]
}
```

Setiap URL bertanda tangan yang Anda gunakan kebijakan ini memiliki URL yang mengidentifikasi file tertentu dalam CloudFront distribusi tertentu, misalnya:

`https://d1111111abcdef8.cloudfront.net/training/orientation.pdf`

URL yang ditandatangani juga menyertakan ID key pair, yang harus dikaitkan dengan grup kunci tepercaya dalam distribusi (`d1111111abcdef8.cloudfront.net`) yang Anda tentukan di URL.

Membuat tanda tangan untuk URL bertanda tangan yang menggunakan kebijakan kustom

Tanda tangan untuk URL bertanda tangan yang menggunakan kebijakan kustom adalah versi salinan dokumen, ditandatangani, dan dikodekan base64 dari pernyataan kebijakan. Untuk membuat tanda tangan untuk kebijakan kustom, selesaikan langkah berikut.

Untuk informasi tambahan dan contoh cara membuat, menandatangani, dan mengkode pernyataan kebijakan, lihat:

- [Perintah Linux dan OpenSSL untuk pengkodean dan enkripsi base64](#)
- [Contoh kode untuk membuat tanda tangan untuk URL yang ditandatangani](#)

Note

Contoh yang ditautkan digunakan secara SHA-1 default. Untuk menggunakannya, SHA-256 ganti sha1 dengan sha256 perintah OpenSSL dan sertakan parameter kueri Hash-Algorithm=SHA256 di URL yang ditandatangani.

Opsi 1: Untuk membuat tanda tangan dengan menggunakan kebijakan khusus

1. Gunakan fungsi SHA-1 atau SHA-256 hash dan kunci pribadi RSA atau ECDSA yang dihasilkan untuk hash dan tandatangani pernyataan kebijakan JSON yang Anda buat dalam prosedur. [Untuk membuat pernyataan kebijakan untuk URL yang ditandatangani menggunakan kebijakan kustom](#) Gunakan versi pernyataan kebijakan yang tidak lagi menyertakan spasi kosong tetapi belum dikodekan base64.

Jika Anda menggunakan SHA-256, Anda harus menyertakan &Hash-Algorithm=SHA256 dalam URL yang ditandatangani.

Untuk kunci privat yang diperlukan oleh fungsi hash, gunakan kunci pribadi yang kunci publiknya berada dalam grup kunci yang dipercaya aktif untuk distribusi.

Note

Metode yang Anda gunakan untuk men-emuk dan menandatangani pernyataan kebijakan tergantung pada bahasa pemrograman dan platform Anda. Untuk kode sampel, lihat [Contoh kode untuk membuat tanda tangan untuk URL yang ditandatangani](#).

2. Hapus spasi kosong (termasuk tab dan karakter baris baru) dari string hash dan ditandatangani.
3. Base64-encode string menggunakan pengkodean MIME base64. Untuk informasi selengkapnya, lihat [Bagian 6.8, Base64 Content-Transfer-Encoding](#) di RFC 2045, MIME (Ekstensi Surat Internet Serbaguna) Bagian Satu: Format Badan Pesan Internet.
4. Ganti karakter yang tidak valid dalam string kueri URL dengan karakter yang valid. Tabel berikut mencantumkan karakter yang tidak valid dan valid.

Ganti karakter tidak valid ini	Dengan karakter valid ini
+	- (tanda hubung)
=	_ (garis bawah)
/	~ (tilde)

5. Tambahkan nilai yang dihasilkan ke URL Anda yang ditandatangani setelah `&Signature=`, dan kembali ke [Untuk membuat URL yang ditandatangani menggunakan kebijakan kustom](#) untuk menyelesaikan penyatuan bagian URL yang Anda tanda tangani.

Gunakan cookie yang ditandatangani

CloudFront Cookie yang ditandatangani memungkinkan Anda untuk mengontrol siapa yang dapat mengakses konten Anda ketika Anda tidak ingin mengubah URL Anda saat ini atau ketika Anda ingin memberikan akses ke beberapa file terbatas, misalnya, semua file di area pelanggan situs web. Topik ini menjelaskan pertimbangan saat menggunakan cookie yang ditandatangani dan menjelaskan cara mengatur cookie yang ditandatangani menggunakan kebijakan terekam dan kustom.

Topik

- [Memutuskan untuk menggunakan kebijakan kalengan atau kustom untuk cookie yang ditandatangani](#)
- [Cara kerja cookie yang ditandatangani](#)
- [Mencegah penyalahgunaan cookie yang ditandatangani](#)
- [Saat CloudFront memeriksa tanggal dan waktu kedaluwarsa dalam cookie yang ditandatangani](#)
- [Kode sampel dan alat pihak ketiga](#)
- [Tetapkan cookie yang ditandatangani menggunakan kebijakan kalengan](#)

- [Tetapkan cookie yang ditandatangani menggunakan kebijakan khusus](#)
- [Buat cookie yang ditandatangani menggunakan PHP](#)

Memutuskan untuk menggunakan kebijakan kalengan atau kustom untuk cookie yang ditandatangani

Saat Anda membuat cookie bertanda tangan, Anda menulis pernyataan kebijakan dalam format JSON yang menetapkan batasan pada cookie yang ditandatangani, misalnya, berapa lama cookie valid. Anda dapat menggunakan kebijakan terekam atau kebijakan khusus. Tabel berikut membandingkan kebijakan terekam dan kustom:

Deskripsi	Kebijakan kalengan	Kebijakan khusus
Anda dapat menggunakan kembali pernyataan kebijakan untuk beberapa file. Untuk menggunakan kembali pernyataan kebijakan, Anda harus menggunakan karakter wildcard dalam Resource objek. Untuk informasi lebih lanjut, lihat Nilai yang Anda sebutkan dalam pernyataan kebijakan untuk kebijakan kustom untuk cookie yang ditandatangani.)	Tidak	Ya
Anda dapat menentukan tanggal dan waktu pengguna dapat mulai mengakses konten Anda	Tidak	Ya (opsional)
Anda dapat menentukan tanggal dan waktu saat pengguna tidak lagi dapat mengakses konten Anda	Ya	Ya
Anda dapat menentukan alamat IP atau berbagai alamat IP pengguna yang dapat mengakses konten Anda	Tidak	Ya (opsional)

Untuk informasi tentang membuat cookie yang ditandatangani menggunakan kebijakan terekam, lihat [Tetapkan cookie yang ditandatangani menggunakan kebijakan kalengan.](#)

Untuk informasi tentang membuat cookie yang ditandatangani menggunakan kebijakan kustom, lihat [Tetapkan cookie yang ditandatangani menggunakan kebijakan khusus.](#)

Cara kerja cookie yang ditandatangani

Berikut adalah ikhtisar tentang cara Anda mengonfigurasi CloudFront cookie yang ditandatangani dan bagaimana CloudFront merespons ketika pengguna mengirimkan permintaan yang berisi cookie yang ditandatangani.

1. Dalam CloudFront distribusi Anda, tentukan satu atau beberapa grup kunci tepercaya, yang berisi kunci publik yang CloudFront dapat digunakan untuk memverifikasi tanda tangan URL. Anda menggunakan kunci privat yang sesuai untuk menandatangani URL.

Untuk informasi selengkapnya, lihat [Tentukan penandatanganan yang dapat membuat URL yang ditandatangani dan cookie yang ditandatangani](#).

2. Anda mengembangkan aplikasi untuk menentukan apakah pengguna harus memiliki akses ke konten Anda dan, jika demikian, untuk mengirim tiga Set-Cookie judul ke penampil. (Setiap Set-Cookie header hanya dapat berisi satu pasangan nama-nilai, dan cookie yang CloudFront ditandatangani memerlukan tiga pasangan nama-nilai.) Anda harus mengirim header Set-Cookie ke penampil sebelum penonton meminta konten privat Anda. Jika Anda mengatur waktu kedaluwarsa singkat pada cookie, Anda mungkin juga ingin mengirim tiga lagi Set-Cookie header dalam menanggapi permintaan berikutnya, sehingga pengguna terus memiliki akses.

Biasanya, CloudFront distribusi Anda akan memiliki setidaknya dua perilaku cache, satu yang tidak memerlukan otentikasi dan satu lagi yang melakukannya. Laman kesalahan untuk bagian situs yang aman mencakup pengarah atau tautan ke halaman login.

Jika Anda mengonfigurasi distribusi Anda ke file cache berdasarkan cookie, CloudFront tidak menyimpan file terpisah berdasarkan atribut dalam cookie yang ditandatangani.

3. Pengguna masuk ke situs web Anda dan membayar konten atau memenuhi beberapa persyaratan lain untuk akses.
4. Aplikasi Anda mengembalikan Set-Cookie judul di respons, dan penampil menyimpan pasangan nilai.
5. Pengguna meminta file.

Browser pengguna atau penampil lain mendapatkan pasangan nilai dari langkah 4 dan menambahkannya ke permintaan dalam Cookie header. Ini adalah cookie yang ditandatangani.

6. CloudFront menggunakan kunci publik untuk memvalidasi tanda tangan dalam cookie yang ditandatangani dan untuk mengonfirmasi bahwa cookie belum dirusak. Jika tanda tangan tidak valid, permintaan ditolak.

Jika tanda tangan dalam cookie valid, CloudFront lihat pernyataan kebijakan di cookie (atau buat jika Anda menggunakan kebijakan kalengan) untuk mengonfirmasi bahwa permintaan tersebut masih valid. Misalnya, jika Anda menentukan tanggal dan waktu awal dan akhir untuk cookie, CloudFront konfirmasikan bahwa pengguna mencoba mengakses konten Anda selama periode waktu yang ingin Anda izinkan akses.

Jika permintaan memenuhi persyaratan dalam pernyataan kebijakan, CloudFront menyajikan konten Anda seperti halnya untuk konten yang tidak dibatasi: ini menentukan apakah file sudah berada di cache tepi, meneruskan permintaan ke asal jika perlu, dan mengembalikan file ke pengguna.

Mencegah penyalahgunaan cookie yang ditandatangani

Jika Anda menentukan `Domain` parameter dalam `Set-Cookie` tajuk, menentukan nilai paling tepat yang memungkinkan untuk mengurangi potensi akses oleh seseorang dengan nama domain akar yang sama. Misalnya, `aplikasi.contoh.com` lebih disukai ke `contoh.com`, terutama ketika Anda tidak mengontrol `contoh.com`. Ini membantu mencegah seseorang mengakses konten Anda dari `www.example.com`.

Untuk membantu mencegah jenis serangan ini, lakukan hal berikut:

- Kecualikan `Expires` dan `Max-Age` sehingga `Set-Cookie` judul membuat cookie sesi. Cookie sesi secara otomatis dihapus ketika pengguna menutup peramban, yang mengurangi kemungkinan seseorang mendapatkan akses tanpa izin ke konten Anda.
- Sertakan `Secure` sehingga cookie dienkripsi ketika penampil menyertakannya dalam permintaan.
- Jika memungkinkan, gunakan kebijakan khusus dan sertakan alamat IP penampil.
- Di `CloudFront-Expires` menentukan waktu kedaluwarsa yang paling pendek yang masuk akal berdasarkan berapa lama Anda ingin pengguna mengakses konten Anda.

Saat CloudFront memeriksa tanggal dan waktu kedaluwarsa dalam cookie yang ditandatangani

Untuk menentukan apakah cookie yang ditandatangani masih valid, CloudFront periksa tanggal kedaluwarsa dan waktu dalam cookie pada saat permintaan HTTP. Jika klien mulai mengunduh file besar segera sebelum waktu kedaluwarsa, pengunduhan harus selesai meskipun waktu kedaluwarsa

sudah lewat selama pengunduhan. Jika koneksi TCP menurun dan klien mencoba memulai ulang unduhan setelah waktu kedaluwarsa berlalu, pengunduhan akan gagal.

Jika klien menggunakan Range GETs untuk mendapatkan file dalam potongan yang lebih kecil, setiap permintaan GET yang terjadi setelah waktu kedaluwarsa akan gagal. Untuk informasi lebih lanjut tentang Range GET, lihat [Bagaimana CloudFront memproses permintaan sebagian untuk suatu objek \(rentang GETs\)](#).

Kode sampel dan alat pihak ketiga

Kode contoh untuk konten privat hanya menampilkan cara membuat tanda tangan untuk URL yang ditandatangani. Namun, proses untuk membuat tanda tangan untuk cookie yang ditandatangani sangat mirip, sehingga sebagian besar kode sampel masih relevan. Untuk informasi selengkapnya, lihat topik berikut:

- [Buat tanda tangan URL menggunakan Perl](#)
- [Buat tanda tangan URL menggunakan PHP](#)
- [Buat tanda tangan URL menggunakan C# dan .NET Framework](#)
- [Buat tanda tangan URL menggunakan Java](#)

Tetapkan cookie yang ditandatangani menggunakan kebijakan kalengan

Untuk mengatur cookie bertanda tangan dengan menggunakan kebijakan terekam, selesaikan langkah berikut. Untuk membuat tanda tangan, lihat [Membuat tanda tangan untuk cookie yang ditandatangani yang menggunakan kebijakan kalengan](#).

Untuk mengatur cookie bertanda tangan menggunakan kebijakan terekam

1. Jika Anda menggunakan .NET atau Java untuk membuat cookie yang telah ditandatangani, dan jika Anda belum memformat ulang kunci pribadi untuk pasangan kunci Anda dari format .pem default ke format yang kompatibel dengan .NET atau dengan Java, lakukan sekarang. Untuk informasi selengkapnya, lihat [Memformat ulang kunci pribadi \(hanya .NET dan Java\)](#).
2. Program aplikasi Anda untuk mengirim tiga Set-Cookie header ke pemirsa yang disetujui (atau empat, jika Anda ingin menentukan algoritma hash). Anda perlu tiga Set-Cookie karena tiap Set-Cookie header hanya dapat berisi satu pasangan dengan nilai nama, dan CloudFront cookie yang ditandatangani memerlukan tiga pasangan nilai. Pasangan nama-nilai adalah: CloudFront-Expires, CloudFront-Signature, dan CloudFront-Key-Pair-Id. Anda dapat secara opsional menyertakan pasangan nama-nilai keempat, CloudFront-Hash-

Algorithm, untuk menentukan algoritma hash yang digunakan untuk tanda tangan. Nilai harus ada di penampil sebelum pengguna membuat permintaan pertama untuk file yang ingin Anda kontrol akses.

 Note

Secara umum, kami sarankan Anda tidak memasukkan Expires dan Max-Age atribut. Kecuali atribut tersebut akan menyebabkan peramban menghapus cookie saat pengguna menutup peramban, yang mengurangi kemungkinan seseorang mendapatkan akses tanpa izin ke konten Anda. Untuk informasi selengkapnya, lihat [Mencegah penyalahgunaan cookie yang ditandatangani](#).

Nama atribut cookie peka huruf besar-kecil.

Pemutusan jalur hanya disertakan untuk membuat atribut lebih mudah dibaca.

```
Set-Cookie:
CloudFront-Expires=date and time in Unix time format (in seconds) and Coordinated
Universal Time (UTC);
Domain=optional domain name;
Path=/optional directory path;
Secure;
HttpOnly

Set-Cookie:
CloudFront-Signature=hashed and signed version of the policy statement;
Domain=optional domain name;
Path=/optional directory path;
Secure;
HttpOnly

Set-Cookie:
CloudFront-Key-Pair-Id=public key ID for the CloudFront public key whose
corresponding private key you're using to generate the signature;
Domain=optional domain name;
Path=/optional directory path;
Secure;
HttpOnly

Set-Cookie:
```

```
CloudFront-Hash-Algorithm=SHA1 or SHA256;  
Domain=optional domain name;  
Path=/optional directory path;  
Secure;  
HttpOnly
```

(Opsional) **Domain**

Nama domain untuk file yang diminta. Jika Anda tidak menyebutkan `Domain` atribut, nilai default adalah nama domain di URL, dan hanya berlaku untuk nama domain tertentu, bukan subdomain. Jika Anda menentukan `Domain`, ini juga berlaku untuk subdomain. Titik utama pada nama domain (misalnya, `Domain=.example.com`) bersifat opsional. Selain itu, jika Anda menentukan `Domain`, nama domain di URL dan nilai `Domain` atribut harus cocok.

Anda dapat menentukan nama domain yang CloudFront ditetapkan untuk distribusi Anda, misalnya, `d111111abcdef8.cloudfront.net`, tetapi Anda tidak dapat menentukan `*.cloudfront.net` untuk nama domain.

Jika Anda ingin menggunakan nama domain alternatif seperti `contoh.com` di URL, Anda harus menambahkan nama domain alternatif ke distribusi Anda terlepas dari apakah Anda menetapkan atribut `Domain`. Untuk informasi lebih lanjut, lihat [Nama domain alternatif \(CNames\)](#) dalam [Semua referensi pengaturan distribusi](#) topik.

(Opsional) **Path**

Jalur untuk file yang diminta. Jika Anda tidak menyebutkan `Path` atribut, nilai default adalah alur di URL.

Secure

Meminta pemirsa mengenkripsi cookie sebelum mengirim permintaan. Kami menyarankan agar Anda mengirimkan `Set-Cookie` header melalui koneksi HTTPS untuk memastikan bahwa atribut cookie dilindungi dari serangan manusia dalam menengah.

HttpOnly

Mendefinisikan bagaimana browser (jika didukung) berinteraksi dengan nilai cookie. Dengan `HttpOnly`, nilai cookie tidak dapat diakses JavaScript. Tindakan pencegahan ini dapat membantu mengurangi serangan cross-site scripting (XSS). Untuk informasi selengkapnya, lihat [Menggunakan cookie HTTP](#).

CloudFront-Expires

Tentukan tanggal dan waktu kedaluwarsa dalam format waktu Unix (dalam detik) dan Waktu Universal Terkoordinasi (UTC). Misalnya, 1 Januari 2026 10:00 UTC mengkonversi ke 1767290400 dalam format waktu Unix.

Untuk menggunakan waktu epoch, tentukan bilangan bulat 64-bit untuk tanggal yang paling lambat 9223372036854775807 (Jumat, 11 April 2262 pukul 23:47:16.854 UTC).

Untuk informasi tentang UTC, lihat RFC 3339, Tanggal dan Waktu di Internet: Stempel Waktu, <https://tools.ietf.org/html/rfc3339>.

CloudFront-Signature

Versi yang di-hash, ditandatangani, dan dikodekan base64 dari pernyataan kebijakan JSON. Untuk informasi selengkapnya, lihat [Membuat tanda tangan untuk cookie yang ditandatangani yang menggunakan kebijakan kalengan](#).

CloudFront-Key-Pair-Id

ID untuk kunci CloudFront publik, misalnya, K2JCJMDEHXQW5F. ID kunci publik memberi tahu kunci publik CloudFront mana yang akan digunakan untuk memvalidasi URL yang ditandatangani. CloudFront membandingkan informasi dalam tanda tangan dengan informasi dalam pernyataan kebijakan untuk memverifikasi bahwa URL belum dirusak.

Kunci publik ini harus dimiliki oleh kelompok kunci yang merupakan signer tepercaya dalam distribusi. Untuk informasi selengkapnya, lihat [Tentukan penandatanganan yang dapat membuat URL yang ditandatangani dan cookie yang ditandatangani](#).

CloudFront-Hash-Algorithm

(Opsional) Algoritma hash yang digunakan untuk membuat tanda tangan. Nilai yang didukung adalah SHA1 dan SHA256. Jika Anda tidak menyertakan cookie ini, CloudFront defaultnya. SHA1

Contoh berikut menunjukkan header Set-Cookie untuk satu cookie yang ditandatangani saat Anda menggunakan nama domain yang terkait dengan distribusi Anda di untuk file Anda:

```
Set-Cookie: CloudFront-Expires=1426500000; Domain=d111111abcdef8.cloudfront.net; Path=/images/*; Secure; HttpOnly
```

```
Set-Cookie: CloudFront-Signature=yXrSIgyQoeE4FBI4eMKF6ho~CA8_  
Domain=d111111abcdef8.cloudfront.net; Path=/images/*; Secure; HttpOnly  
Set-Cookie: CloudFront-Key-Pair-Id=K2JCJMDEHXQW5F;  
Domain=d111111abcdef8.cloudfront.net; Path=/images/*; Secure; HttpOnly  
Set-Cookie: CloudFront-Hash-Algorithm=SHA256; Domain=d111111abcdef8.cloudfront.net;  
Path=/images/*; Secure; HttpOnly
```

Contoh berikut menunjukkan header Set-Cookie untuk satu cookie yang ditandatangani saat Anda menggunakan contoh nama domain alternatif.org di untuk file Anda:

```
Set-Cookie: CloudFront-Expires=1426500000; Domain=example.org; Path=/images/*; Secure;  
HttpOnly  
Set-Cookie: CloudFront-Signature=yXrSIgyQoeE4FBI4eMKF6ho~CA8_  
Domain=example.org;  
Path=/images/*; Secure; HttpOnly  
Set-Cookie: CloudFront-Key-Pair-Id=K2JCJMDEHXQW5F; Domain=example.org; Path=/images/*;  
Secure; HttpOnly  
Set-Cookie: CloudFront-Hash-Algorithm=SHA256; Domain=example.org; Path=/images/*;  
Secure; HttpOnly
```

Jika Anda ingin menggunakan nama domain alternatif seperti contoh.com di URL, Anda harus menambahkan nama domain alternatif ke distribusi Anda terlepas dari apakah Anda menetapkan atribut Domain. Untuk informasi lebih lanjut, lihat [Nama domain alternatif \(CNames\)](#) dalam [Semua referensi pengaturan distribusi](#) topik.

Membuat tanda tangan untuk cookie yang ditandatangani yang menggunakan kebijakan kalengan

Untuk membuat tanda tangan untuk cookie yang ditandatangani yang menggunakan kebijakan kalengan, selesaikan prosedur berikut.

Topik

- [Membuat pernyataan kebijakan untuk cookie yang ditandatangani yang menggunakan kebijakan kalengan](#)
- [Menandatangani pernyataan kebijakan untuk membuat tanda tangan untuk cookie yang ditandatangani yang menggunakan kebijakan kalengan](#)

Membuat pernyataan kebijakan untuk cookie yang ditandatangani yang menggunakan kebijakan kalengan

Saat Anda mengatur cookie bertanda tangan yang menggunakan kebijakan terekam, CloudFront-Signature atribut adalah versi yang di- hashed dan ditandatangani dari pernyataan kebijakan.

Untuk cookie bertanda tangan yang menggunakan kebijakan terekam, Anda tidak menyertakan pernyataan kebijakan di Set-Cookie seperti yang Anda lakukan untuk cookie bertanda tangan yang menggunakan kebijakan kustom. Untuk membuat pernyataan kebijakan, selesaikan langkah berikut.

Untuk membuat pernyataan kebijakan untuk cookie bertanda tangan yang menggunakan kebijakan terekam

1. Membangun pernyataan kebijakan menggunakan format JSON berikut dan menggunakan pengkodean UTF-8 karakter. Sertakan semua tanda baca dan nilai literal lainnya persis seperti yang ditentukan. Untuk informasi tentang Resource dan DateLessThan parameter, lihat [Nilai yang Anda sebutkan dalam pernyataan kebijakan untuk kebijakan terekam untuk cookie yang ditandatangani](#).

```
{
  "Statement": [
    {
      "Resource": "base URL or stream name",
      "Condition": {
        "DateLessThan": {
          "AWS:EpochTime": ending date and time in Unix time format and
          UTC
        }
      }
    }
  ]
}
```

2. Hapus semua spasi kosong (termasuk tab dan karakter baris baru) dari pernyataan kebijakan. Anda mungkin harus memasukkan karakter escape dalam string di kode aplikasi.

Nilai yang Anda sebutkan dalam pernyataan kebijakan untuk kebijakan terekam untuk cookie yang ditandatangani

Ketika Anda membuat pernyataan kebijakan untuk kebijakan terekam, Anda menentukan nilai-nilai berikut:

Sumber Daya

URL dasar termasuk string pencarian Anda, jika ada, misalnya:


```
https://d111111abcdef8.cloudfront.net/images/horizon.jpg?  
size=large&license=yes
```

Anda hanya dapat menentukan satu nilai untuk Resource.

Perhatikan hal-hal berikut:

- Protokol – Nilai harus dimulai dengan `http://` atau `https://`.
- Parameter string kueri – Jika Anda tidak memiliki parameter string pencarian, hapus tanda tanya.
- Nama domain alternatif – Jika Anda menentukan nama domain alternatif (CNAME) di URL, Anda harus menentukan nama domain alternatif saat merujuk file di halaman web atau aplikasi Anda. Jangan tentukan URL Amazon S3 untuk file tersebut.

DateLessThan

Tanggal dan waktu kedaluwarsa untuk URL dalam format waktu Unix (dalam detik) dan Waktu Universal Terkoordinasi (UTC). Jangan melampirkan nilai dalam tanda petik.

Misalnya, 16 Maret 2015 10.00 UTC dikonversi menjadi 1426500000 dalam format waktu Unix.

Nilai ini harus cocok dengan nilai `CloudFront-Expires` dalam `Set-Cookie` header. Jangan melampirkan nilai dalam tanda petik.

Untuk informasi selengkapnya, lihat [Saat CloudFront memeriksa tanggal dan waktu kedaluwarsa dalam cookie yang ditandatangani](#).

Contoh pernyataan kebijakan untuk kebijakan terekam

Saat Anda menggunakan contoh pernyataan kebijakan berikut dalam cookie yang ditandatangani, pengguna dapat mengakses file `https://d111111abcdef8.cloudfront.net/horizon.jpg` hingga 16 Maret 2015 10.00 UTC:

```
{  
  "Statement": [  
    {  
      "Resource": "https://d111111abcdef8.cloudfront.net/horizon.jpg?  
size=large&license=yes",  
      "Condition": {  
        "DateLessThan": {
```

```

    "AWS:EpochTime": 1426500000
  }
}
]
}

```

Menandatangani pernyataan kebijakan untuk membuat tanda tangan untuk cookie yang ditandatangani yang menggunakan kebijakan kalengan

Untuk membuat nilai untuk CloudFront-Signature atribut dalam Set-Cookie tajuk, Anda memiliki dan menandatangani pernyataan kebijakan yang Anda buat di [Untuk membuat pernyataan kebijakan untuk cookie bertanda tangan yang menggunakan kebijakan terekam](#).

Untuk informasi tambahan dan contoh cara membuat, menandatangani, dan mengodekan pernyataan kebijakan, lihat topik berikut:

- [Perintah Linux dan OpenSSL untuk pengkodean dan enkripsi base64](#)
- [Contoh kode untuk membuat tanda tangan untuk URL yang ditandatangani](#)

Note

Contoh yang ditautkan digunakan secara SHA-1 default. Untuk menggunakannya, SHA-256 ganti sha1 dengan sha256 perintah OpenSSL dan sertakan cookie dengan CloudFront-Hash-Algorithm nilai. SHA256

Untuk membuat tanda tangan untuk cookie yang ditandatangani menggunakan kebijakan terekam

1. Gunakan fungsi SHA-1 or SHA-256 hash dan RSA untuk hash dan tandatangani pernyataan kebijakan yang Anda buat dalam prosedur. [Untuk membuat pernyataan kebijakan untuk cookie bertanda tangan yang menggunakan kebijakan terekam](#) Gunakan versi pernyataan kebijakan yang tidak lagi menyertakan spasi kosong.

Jika Anda menggunakannya SHA-256, Anda harus menyertakan CloudFront-Hash-Algorithm cookie dengan nilaiSHA256.

Untuk kunci privat yang diperlukan oleh fungsi hash, gunakan kunci pribadi yang kunci publiknya berada dalam grup kunci yang dipercaya aktif untuk distribusi.

Note

Metode yang Anda gunakan untuk men-emuk dan menandatangani pernyataan kebijakan tergantung pada bahasa pemrograman dan platform Anda. Untuk kode sampel, lihat [Contoh kode untuk membuat tanda tangan untuk URL yang ditandatangani](#).

2. Hapus spasi kosong (termasuk tab dan karakter baris baru) dari string hash dan ditandatangani.
3. Base64-encode string menggunakan pengkodean MIME base64. Untuk informasi selengkapnya, lihat [Bagian 6.8, Base64 Content-Transfer-Encoding](#) di RFC 2045, MIME (Ekstensi Surat Internet Serbaguna) Bagian Satu: Format Badan Pesan Internet.
4. Ganti karakter yang tidak valid dalam string kueri URL dengan karakter yang valid. Tabel berikut mencantumkan karakter yang tidak valid dan valid.

Ganti karakter tidak valid ini	Dengan karakter valid ini
+	- (tanda hubung)
=	_ (garis bawah)
/	~ (tilde)

5. Sertakan nilai yang dihasilkan dalam Set-Cookie header untuk CloudFront-Signature pasangan yang bernilai. Lalu kembali ke [Untuk mengatur cookie bertanda tangan menggunakan kebijakan terekam](#) tambahkan Set-Cookie header untuk CloudFront-Key-Pair-Id.

Tetapkan cookie yang ditandatangani menggunakan kebijakan khusus

Untuk mengatur cookie bertanda tangan yang menggunakan kebijakan kustom, selesaikan langkah berikut.

Untuk mengatur cookie bertanda tangan menggunakan kebijakan kustom

1. Jika Anda menggunakan .NET atau Java untuk membuat URL yang ditandatangani, dan jika Anda belum memformat ulang kunci privat untuk pasangan kunci dari format .pemiksa default ke format yang kompatibel dengan .NET atau Java, lakukan sekarang. Untuk informasi selengkapnya, lihat [Memformat ulang kunci pribadi \(hanya .NET dan Java\)](#).

2. Program aplikasi Anda untuk mengirim tiga Set-Cookie header ke pemirsa yang disetujui (atau empat, jika Anda ingin menentukan algoritma hash). Anda memerlukan tiga Set-Cookie header karena setiap Set-Cookie header hanya dapat berisi satu pasangan nama-nilai, dan cookie yang CloudFront ditandatangani memerlukan tiga pasangan nama-nilai. Pasangan nama-nilai adalah: CloudFront-Policy, CloudFront-Signature, dan CloudFront-Key-Pair-Id. Anda dapat secara opsional menyertakan pasangan nama-nilai keempat, CloudFront-Hash-Algorithm, untuk menentukan algoritma hash yang digunakan untuk tanda tangan. Nilai harus ada di penampil sebelum pengguna membuat permintaan pertama untuk file yang ingin Anda kontrol akses.

Note

Secara umum, kami sarankan Anda tidak memasukkan Expires dan Max-Age atribut. Ini menyebabkan browser menghapus cookie saat pengguna menutup browser, yang mengurangi kemungkinan seseorang mendapatkan akses tidak sah ke konten Anda. Untuk informasi selengkapnya, lihat [Mencegah penyalahgunaan cookie yang ditandatangani](#).

Nama atribut cookie peka huruf besar-kecil.

Pemutusan jalur hanya disertakan untuk membuat atribut lebih mudah dibaca.

```
Set-Cookie:
CloudFront-Policy=base64 encoded version of the policy statement;
Domain=optional domain name;
Path=/optional directory path;
Secure;
HttpOnly

Set-Cookie:
CloudFront-Signature=hashed and signed version of the policy statement;
Domain=optional domain name;
Path=/optional directory path;
Secure;
HttpOnly

Set-Cookie:
```

```
CloudFront-Key-Pair-Id=public key ID for the CloudFront public key whose  
corresponding private key you're using to generate the signature;  
Domain=optional domain name;  
Path=/optional directory path;  
Secure;  
HttpOnly  
  
Set-Cookie:  
CloudFront-Hash-Algorithm=SHA1 or SHA256;  
Domain=optional domain name;  
Path=/optional directory path;  
Secure;  
HttpOnly
```

(Optional) **Domain**

Nama domain untuk file yang diminta. Jika Anda tidak menyebutkan Domain atribut, nilai default adalah nama domain di URL, dan hanya berlaku untuk nama domain tertentu, bukan subdomain. Jika Anda menentukan Domain, ini juga berlaku untuk subdomain. Titik utama pada nama domain (misalnya, Domain=.example.com) bersifat opsional. Selain itu, jika Anda menentukan Domain, nama domain di URL dan nilai Domain atribut harus cocok.

Anda dapat menentukan nama domain yang CloudFront ditetapkan untuk distribusi Anda, misalnya, d111111abcdef8.cloudfront.net, tetapi Anda tidak dapat menentukan *.cloudfront.net untuk nama domain.

Jika Anda ingin menggunakan nama domain alternatif seperti contoh.com di URL, Anda harus menambahkan nama domain alternatif ke distribusi Anda terlepas dari apakah Anda menetapkan atribut Domain. Untuk informasi lebih lanjut, lihat [Nama domain alternatif \(CNames\)](#) dalam [Semua referensi pengaturan distribusi](#) topik.

(Optional) **Path**

Jalur untuk file yang diminta. Jika Anda tidak menyebutkan Path atribut, nilai default adalah jalur di URL.

Secure

Meminta pemirsa mengenkripsi cookie sebelum mengirim permintaan. Kami menyarankan agar Anda mengirimkan Set-Cookie header melalui koneksi HTTPS untuk memastikan bahwa atribut cookie dilindungi dari serangan manusia dalam menengah.

HttpOnly

Mengharuskan penampil mengirimkan cookie hanya dalam permintaan HTTP atau HTTPS.

CloudFront-Policy

Pernyataan kebijakan Anda dalam format JSON, dengan spasi kosong dihapus, lalu base64 dikodekan. Untuk informasi selengkapnya, lihat [Membuat tanda tangan untuk cookie yang ditandatangani yang menggunakan kebijakan kustom](#).

Pernyataan kebijakan mengendalikan akses yang diberikan oleh cookie yang ditandatangani kepada pengguna. Ini mencakup file yang dapat diakses pengguna, tanggal dan waktu kedaluwarsa, tanggal dan waktu opsional URL menjadi valid, dan alamat IP opsional atau rentang alamat IP yang diizinkan untuk mengakses file.

CloudFront-Signature

Versi yang di-hash, ditandatangani, dan dikodekan base64 dari pernyataan kebijakan JSON. Untuk informasi selengkapnya, lihat [Membuat tanda tangan untuk cookie yang ditandatangani yang menggunakan kebijakan kustom](#).

CloudFront-Key-Pair-Id

ID untuk kunci CloudFront publik, misalnya, K2JCJMDEHXQW5F. ID kunci publik memberi tahu kunci publik CloudFront mana yang akan digunakan untuk memvalidasi URL yang ditandatangani. CloudFront membandingkan informasi dalam tanda tangan dengan informasi dalam pernyataan kebijakan untuk memverifikasi bahwa URL belum dirusak.

Kunci publik ini harus dimiliki oleh kelompok kunci yang merupakan signer tepercaya dalam distribusi. Untuk informasi selengkapnya, lihat [Tentukan penandatanganan yang dapat membuat URL yang ditandatangani dan cookie yang ditandatangani](#).

CloudFront-Hash-Algorithm

(Opsional) Algoritma hash yang digunakan untuk membuat tanda tangan. Nilai yang didukung adalah SHA1 dan SHA256. Jika Anda tidak menyertakan cookie ini, CloudFront defaultnya. SHA1

Contoh **Set-Cookie** header untuk kebijakan kustom

Lihat contoh pasangan Set-Cookie header berikut.

Jika Anda ingin menggunakan nama domain alternatif seperti `example.org` di URL, Anda harus menambahkan nama domain alternatif ke distribusi Anda terlepas dari apakah Anda menentukan atribut. Domain Untuk informasi selengkapnya, lihat [Nama domain alternatif \(CNames\)](#) dalam topik [Semua referensi pengaturan distribusi](#).

Example Contoh 1

Anda dapat menggunakan Set-Cookie header untuk satu cookie yang ditandatangani saat Anda menggunakan nama domain yang terkait dengan distribusi Anda di URL untuk file Anda.

```
Set-Cookie: CloudFront-  
Policy=eyJTdGF0ZWl1bnQiOi0t7IlJlc291cmNlIjoiaHR0cDovL2QxMTExMTFhYmNkZWY4LmNsb3VkZnJvbnQubmV0L2dh  
Domain=d111111abcdef8.cloudfront.net; Path=/; Secure; HttpOnly  
Set-Cookie: CloudFront-Signature=dtKhpJ3aUYxqDIwepczPiDb9NXQ_  
Domain=d111111abcdef8.cloudfront.net; Path=/; Secure; HttpOnly  
Set-Cookie: CloudFront-Key-Pair-Id=K2JCJMDEHXQW5F;  
Domain=d111111abcdef8.cloudfront.net; Path=/; Secure; HttpOnly  
Set-Cookie: CloudFront-Hash-Algorithm=SHA256; Domain=d111111abcdef8.cloudfront.net;  
Path=/; Secure; HttpOnly
```

Example Contoh 2

Anda dapat menggunakan Set-Cookie header untuk satu cookie yang ditandatangani saat Anda menggunakan nama domain alternatif (`example.org`) di URL untuk file Anda.

```
Set-Cookie: CloudFront-  
Policy=eyJTdGF0ZWl1bnQiOi0t7IlJlc291cmNlIjoiaHR0cDovL2QxMTExMTFhYmNkZWY4LmNsb3VkZnJvbnQubmV0L2dh  
Domain=example.org; Path=/; Secure; HttpOnly  
Set-Cookie: CloudFront-Signature=dtKhpJ3aUYxqDIwepczPiDb9NXQ_; Domain=example.org;  
Path=/; Secure; HttpOnly  
Set-Cookie: CloudFront-Key-Pair-Id=K2JCJMDEHXQW5F; Domain=example.org; Path=/; Secure;  
HttpOnly  
Set-Cookie: CloudFront-Hash-Algorithm=SHA256; Domain=example.org; Path=/; Secure;  
HttpOnly
```

Example Contoh 3

Anda dapat menggunakan pasangan Set-Cookie header untuk permintaan yang ditandatangani saat Anda menggunakan nama domain yang terkait dengan distribusi Anda di URL untuk file Anda.

```
Set-Cookie: CloudFront-
Policy=eyJTdGF0ZWl1bnQiOi0t7IlJlcl291cmNlIjoiaHR0cDovL2QxMTExMTFhYmNkZWY4LmNsb3VkZnJvbnQubmV0L2dh
Domain=d111111abcdef8.cloudfront.net; Path=/; Secure; HttpOnly
Set-Cookie: CloudFront-Signature=dtKhpJ3aUYxqDIwepczPiDb9NXQ_;
Domain=d111111abcdef8.cloudfront.net; Path=/; Secure; HttpOnly
Set-Cookie: CloudFront-Key-Pair-Id=K2JCJMDEHXQW5F;
Domain=dd111111abcdef8.cloudfront.net; Path=/; Secure; HttpOnly
Set-Cookie: CloudFront-Hash-Algorithm=SHA256; Domain=d111111abcdef8.cloudfront.net;
Path=/; Secure; HttpOnly
```

Example Contoh 4

Anda dapat menggunakan pasangan Set-Cookie header untuk satu permintaan yang ditandatangani saat Anda menggunakan nama domain alternatif (example.org) yang terkait dengan distribusi Anda di URL untuk file Anda.

```
Set-Cookie: CloudFront-
Policy=eyJTdGF0ZWl1bnQiOi0t7IlJlcl291cmNlIjoiaHR0cDovL2QxMTExMTFhYmNkZWY4LmNsb3VkZnJvbnQubmV0L2dh
Domain=example.org; Path=/; Secure; HttpOnly
Set-Cookie: CloudFront-Signature=dtKhpJ3aUYxqDIwepczPiDb9NXQ_; Domain=example.org;
Path=/; Secure; HttpOnly
Set-Cookie: CloudFront-Key-Pair-Id=K2JCJMDEHXQW5F; Domain=example.org; Path=/; Secure;
HttpOnly
Set-Cookie: CloudFront-Hash-Algorithm=SHA256; Domain=example.org; Path=/; Secure;
HttpOnly
```

Membuat pernyataan kebijakan untuk cookie yang ditandatangani yang menggunakan kebijakan khusus

Untuk membuat pernyataan kebijakan untuk kebijakan kustom, selesaikan langkah berikut. Untuk beberapa contoh pernyataan kebijakan yang mengendalikan akses ke file dalam berbagai cara, lihat [Contoh pernyataan kebijakan untuk cookie bertanda tangan yang menggunakan kebijakan kustom](#).

Untuk membuat pernyataan kebijakan untuk cookie bertanda tangan yang menggunakan kebijakan khusus

1. Buat pernyataan kebijakan dengan menggunakan format JSON berikut.

```
{
  "Statement": [
    {
```



```

    "Resource": "URL of the file",
    "Condition": {
        "DateLessThan": {
            "AWS:EpochTime": required ending date and time in Unix time
format and UTC
        },
        "DateGreaterThan": {
            "AWS:EpochTime": optional beginning date and time in Unix time
format and UTC
        },
        "IpAddress": {
            "AWS:SourceIp": "optional IP address"
        }
    }
}
]
}

```

Perhatikan hal-hal berikut:

- Anda dapat menyertakan hanya satu pernyataan.
 - Gunakan pengkodean UTF-8 karakter.
 - Sertakan semua nama tanda baca dan parameter persis seperti yang ditentukan. Singkatan untuk nama parameter tidak diterima.
 - Urutan parameter di `Condition` tidak masalah.
 - Untuk informasi tentang nilai untuk `Resource`, `DateLessThan`, `DateGreaterThan`, dan `IpAddress`, lihat [Nilai yang Anda sebutkan dalam pernyataan kebijakan untuk kebijakan kustom untuk cookie yang ditandatangani](#).
2. Hapus semua spasi kosong (termasuk tab dan karakter baris baru) dari pernyataan kebijakan. Anda mungkin harus memasukkan karakter escape dalam string di kode aplikasi.
 3. Base64-encode pernyataan kebijakan menggunakan pengkodean MIME base64. Untuk informasi selengkapnya, lihat [Bagian 6.8, Base64 Content-Transfer-Encoding](#) di RFC 2045, MIME (Ekstensi Surat Internet Serbaguna) Bagian Satu: Format Badan Pesan Internet.
 4. Ganti karakter yang tidak valid dalam string kueri URL dengan karakter yang valid. Tabel berikut mencantumkan karakter yang tidak valid dan valid.

Ganti karakter tidak valid ini

Dengan karakter valid ini

Ganti karakter tidak valid ini	Dengan karakter valid ini
+	- (tanda hubung)
=	_ (garis bawah)
/	~ (tilde)

- Sertakan nilai yang dihasilkan dalam Set-Cookie setelah CloudFront-Policy=.
- Buat tanda tangan untuk Set-Cookie header untuk CloudFront-Signature dengan mengadakan, menandatangani, dan memberikan kode dasar64 pada pernyataan kebijakan. Untuk informasi selengkapnya, lihat [Membuat tanda tangan untuk cookie yang ditandatangani yang menggunakan kebijakan kustom](#).

Nilai yang Anda sebutkan dalam pernyataan kebijakan untuk kebijakan kustom untuk cookie yang ditandatangani

Saat Anda membuat pernyataan kebijakan untuk kebijakan kustom, Anda menentukan nilai berikut.

Sumber Daya

URL dasar termasuk string pencarian Anda, jika ada:

```
https://d111111abcdef8.cloudfront.net/images/horizon.jpg?
size=large&license=yes
```

Important

Jika Anda menghilangkan Resource parameter, pengguna dapat mengakses semua file yang terkait dengan distribusi apa pun yang terkait dengan pasangan kunci yang Anda gunakan untuk membuat URL yang ditandatangani.

Anda hanya dapat menentukan satu nilai untuk Resource.

Perhatikan hal-hal berikut:

- Protokol – Nilai harus dimulai dengan http:// atau https://.

- Parameter string kueri – Jika Anda tidak memiliki parameter string pencarian, hapus tanda tanya.
- Wildcard – Anda dapat menggunakan karakter wildcard yang sesuai dengan nol karakter atau lebih (*) atau karakter wildcard yang persis sesuai dengan satu karakter (?) di mana pun dalam string. Misalnya, nilai:

`https://d111111abcdef8.cloudfront.net/*game_download.zip*`

akan mencakup (misalnya) file berikut:

- `https://d111111abcdef8.cloudfront.net/game_download.zip`
- `https://d111111abcdef8.cloudfront.net/example_game_download.zip?license=yes`
- `https://d111111abcdef8.cloudfront.net/test_game_download.zip?license=temp`
- Nama domain alternatif – Jika Anda menentukan nama domain alternatif (CNAME) di URL, Anda harus menentukan nama domain alternatif saat merujuk file di halaman web atau aplikasi Anda. Jangan tentukan URL Amazon S3 untuk file tersebut.

DateLessThan

Tanggal dan waktu kedaluwarsa untuk URL dalam format waktu Unix (dalam detik) dan Waktu Universal Terkoordinasi (UTC). Jangan melampirkan nilai dalam tanda petik.

Misalnya, 16 Maret 2015 10.00 UTC dikonversi menjadi 1426500000 dalam format waktu Unix.

Untuk informasi selengkapnya, lihat [Saat CloudFront memeriksa tanggal dan waktu kedaluwarsa dalam cookie yang ditandatangani](#).

DateGreaterThan (Opsional)

Tanggal dan waktu mulai opsional untuk URL dalam format waktu Unix (dalam detik) dan Waktu Universal Terkoordinasi (UTC). Pengguna tidak diizinkan untuk mengakses file pada atau sebelum tanggal dan waktu yang ditentukan. Jangan melampirkan nilai dalam tanda petik.

IpAddress (Opsional)

Alamat IP klien yang membuat permintaan GET. Perhatikan hal-hal berikut:

- Untuk mengizinkan alamat IP mengakses file, hapus `IpAddress` parameter.

- Anda dapat menentukan salah satu alamat IP atau satu rentang alamat IP. Misalnya, Anda tidak dapat mengatur kebijakan untuk memungkinkan akses jika alamat IP klien berada dalam satu dari dua rentang yang berbeda.
- Untuk memungkinkan akses dari satu alamat IP, Anda menentukan:

"IPv4 IP address/32"

- Anda harus menentukan rentang alamat IP dalam format IPv4 CIDR standar (misalnya, 192.0.2.0/24). Untuk informasi lebih lanjut, buka RFC 4632, Classless Inter-domain Routing (CIDR): Rencana Penugasan dan Agregasi Alamat Internet,. <https://tools.ietf.org/html/rfc4632>



Important

Alamat IP dalam format IPv6, seperti 2001:0 db 8:85 a3: :8a2e: 0370:7334, tidak didukung.

Jika Anda menggunakan kebijakan khusus yang mencakup `IpAddress`, jangan mengaktifkan IPv6 untuk distribusi. Jika Anda ingin membatasi akses ke sebagian konten dengan alamat IP dan mendukung permintaan IPv6 untuk konten lain, Anda dapat membuat dua distribusi. Untuk informasi lebih lanjut, lihat [Aktifkan IPv6 \(permintaan penampil\)](#) dalam topik [Semua referensi pengaturan distribusi](#).

Contoh pernyataan kebijakan untuk cookie bertanda tangan yang menggunakan kebijakan kustom

Contoh pernyataan kebijakan berikut menunjukkan cara mengontrol akses ke file tertentu, semua file di direktori, atau semua file yang terkait dengan ID pasangan kunci. Contoh ini juga menunjukkan cara mengontrol akses dari alamat IP individu atau serangkaian alamat IP, dan cara mencegah pengguna menggunakan cookie yang ditandatangani setelah tanggal dan waktu yang ditentukan.

Jika Anda menyalin dan menempelkan salah satu contoh ini, hapus spasi kosong (termasuk tab dan karakter baris baru), ganti nilai dengan nilai Anda sendiri, dan sertakan karakter baris baru setelah tanda kurung kurung penutup (`}`).

Untuk informasi selengkapnya, lihat [Nilai yang Anda sebutkan dalam pernyataan kebijakan untuk kebijakan kustom untuk cookie yang ditandatangani](#).

Topik

- [Contoh pernyataan kebijakan: Akses satu file dari berbagai alamat IP](#)

- [Contoh pernyataan kebijakan: Akses semua file dalam direktori dari berbagai alamat IP](#)
- [Contoh pernyataan kebijakan: Akses semua file yang terkait dengan ID key pair dari satu alamat IP](#)

Contoh pernyataan kebijakan: Akses satu file dari berbagai alamat IP

Contoh kebijakan kustom berikut dalam cookie yang ditandatangani menetapkan bahwa pengguna dapat mengakses file `https://d111111abcdef8.cloudfront.net/game_download.zip` dari alamat IP dalam rentang `192.0.2.0/24` hingga 1 Januari 2023 10:00 UTC:

```
{
  "Statement": [
    {
      "Resource": "https://d111111abcdef8.cloudfront.net/game_download.zip",
      "Condition": {
        "IpAddress": {
          "AWS:SourceIp": "192.0.2.0/24"
        },
        "DateLessThan": {
          "AWS:EpochTime": 1767290400
        }
      }
    }
  ]
}
```

Contoh pernyataan kebijakan: Akses semua file dalam direktori dari berbagai alamat IP

Contoh kebijakan khusus berikut memungkinkan Anda membuat cookie yang ditandatangani untuk setiap file dalam `training` direktori, sebagaimana ditunjukkan oleh `*` karakter wildcard di `Resource` parameter. Pengguna dapat mengakses file dari alamat IP di rentang `192.0.2.0/24` hingga 1 Januari 2013 pukul 10.00 UTC:

```
{
  "Statement": [
    {
      "Resource": "https://d111111abcdef8.cloudfront.net/training/*",
      "Condition": {
        "IpAddress": {
          "AWS:SourceIp": "192.0.2.0/24"
        },
        "DateLessThan": {
```

```

    "AWS:EpochTime": 1767290400
  }
}
]
}

```

Setiap cookie yang ditandatangani di mana Anda menggunakan kebijakan ini mencakup URL dasar yang mengidentifikasi file tertentu, misalnya:

<https://d1111111abcdef8.cloudfront.net/training/orientation.pdf>

Contoh pernyataan kebijakan: Akses semua file yang terkait dengan ID key pair dari satu alamat IP

Kebijakan khusus sampel berikut memungkinkan Anda untuk mengatur cookie yang ditandatangani untuk setiap file yang terkait dengan distribusi apa pun, sebagaimana ditunjukkan oleh * karakter wildcard di Resource parameter. Pengguna harus menggunakan alamat IP 192.0.2.10/32. (Nilai 192.0.2.10/32 dalam notasi CIDR mengacu pada alamat IP tunggal, 192.0.2.10.) File hanya tersedia dari 1 Januari 2013 10.00 UTC hingga 2 Januari 2013 10.00 UTC:

```

{
  "Statement": [
    {
      "Resource": "https://*",
      "Condition": {
        "IpAddress": {
          "AWS:SourceIp": "192.0.2.10/32"
        },
        "DateGreaterThan": {
          "AWS:EpochTime": 1767290400
        },
        "DateLessThan": {
          "AWS:EpochTime": 1767376800
        }
      }
    }
  ]
}

```

Setiap cookie yang ditandatangani di mana Anda menggunakan kebijakan ini mencakup URL dasar yang mengidentifikasi file tertentu dalam CloudFront distribusi tertentu, misalnya:

`https://d111111abcdef8.cloudfront.net/training/orientation.pdf`

Cookie yang ditandatangani juga mencakup ID pasangan kunci, yang harus dikaitkan dengan grup kunci tepercaya dalam distribusi (`d111111abcdef8.cloudfront.net`) yang Anda tentukan di URL dasar.

Membuat tanda tangan untuk cookie yang ditandatangani yang menggunakan kebijakan kustom

Tanda tangan untuk cookie bertanda tangan yang menggunakan kebijakan kustom merupakan versi pernyataan kebijakan yang di-hash, ditandatangani, dan dikodekan base64.

Untuk informasi tambahan dan contoh cara membuat, menandatangani, dan mengkode pernyataan kebijakan, lihat:

- [Perintah Linux dan OpenSSL untuk pengkodean dan enkripsi base64](#)
- [Contoh kode untuk membuat tanda tangan untuk URL yang ditandatangani](#)

 Note

Contoh yang ditautkan digunakan secara SHA-1 default. Untuk menggunakannya, SHA-256 ganti sha1 dengan sha256 perintah OpenSSL dan sertakan cookie dengan CloudFront-Hash-Algorithm nilai. SHA256

Untuk membuat tanda tangan untuk cookie yang ditandatangani dengan menggunakan kebijakan kustom

1. Gunakan fungsi SHA-1 or SHA-256 hash dan RSA untuk hash dan tandatangi pernyataan kebijakan JSON yang Anda buat dalam prosedur. [Untuk membuat pernyataan kebijakan untuk URL yang ditandatangani menggunakan kebijakan kustom](#) Gunakan versi pernyataan kebijakan yang tidak lagi menyertakan spasi kosong tetapi belum dikodekan base64.

Jika Anda menggunakannya SHA-256, Anda harus menyertakan CloudFront-Hash-Algorithm cookie dengan nilai SHA256.

Untuk kunci privat yang diperlukan oleh fungsi hash, gunakan kunci pribadi yang kunci publiknya berada dalam grup kunci yang dipercaya aktif untuk distribusi.

Note

Metode yang Anda gunakan untuk men-emuk dan menandatangani pernyataan kebijakan tergantung pada bahasa pemrograman dan platform Anda. Untuk kode sampel, lihat [Contoh kode untuk membuat tanda tangan untuk URL yang ditandatangani](#).

2. Hapus spasi kosong (termasuk tab dan karakter baris baru) dari string hash dan ditandatangani.
3. Base64-encode string menggunakan pengkodean MIME base64. Untuk informasi selengkapnya, lihat [Bagian 6.8, Base64 Content-Transfer-Encoding](#) di RFC 2045, MIME (Ekstensi Surat Internet Serbaguna) Bagian Satu: Format Badan Pesan Internet.
4. Ganti karakter yang tidak valid dalam string kueri URL dengan karakter yang valid. Tabel berikut mencantumkan karakter yang tidak valid dan valid.

Ganti karakter tidak valid ini	Dengan karakter valid ini
+	- (tanda hubung)
=	_ (garis bawah)
/	~ (tilde)

5. Sertakan nilai yang dihasilkan dalam Set-Cookie header untuk CloudFront-Signature=nama-nilai, dan kembali ke [Untuk mengatur cookie bertanda tangan menggunakan kebijakan kustom](#) untuk menambahkan Set-Cookie header untuk CloudFront-Key-Pair-Id.

Buat cookie yang ditandatangani menggunakan PHP

Contoh kode berikut mirip dengan contoh [Buat tanda tangan URL menggunakan PHP](#) di mana ia membuat tautan ke video. Namun, alih-alih menandatangani URL dalam kode, contoh ini menandatangani cookie dengan `create_signed_cookies()` fungsi tersebut. Pemain sisi klien menggunakan cookie untuk mengautentikasi setiap permintaan ke distribusi. CloudFront

Pendekatan ini berguna untuk streaming konten, seperti HTTP Live Streaming (HLS) atau Dynamic Adaptive Streaming melalui HTTP (DASH), di mana klien perlu membuat beberapa permintaan untuk mengambil manifes, segmen, dan aset pemutaran terkait. Dengan menggunakan cookie yang

ditandatangani, klien dapat mengautentikasi setiap permintaan tanpa perlu membuat URL baru yang ditandatangani untuk setiap segmen.

Note

- Membuat tanda tangan URL hanyalah salah satu bagian dari proses penyajian konten pribadi menggunakan cookie yang ditandatangani. Untuk informasi selengkapnya, lihat [Gunakan cookie yang ditandatangani](#).

Topik

- [Buat RSA SHA-1 atau tanda tangan SHA-256](#)
- [Buat cookie yang ditandatangani](#)
- [Kode lengkap](#)

Bagian berikut memecah contoh kode menjadi bagian-bagian individual. Anda dapat menemukan [contoh kode](#) lengkap di bawah ini.

Buat RSA SHA-1 atau tanda tangan SHA-256

Contoh kode ini melakukan hal berikut:

1. Fungsi melakukan `rsa_sha1_sign` hash dan menandatangani pernyataan kebijakan menggunakan SHA-1. Untuk menggunakannya, SHA-256 gunakan fungsi `rsa_sha256_sign` yang ditunjukkan di bawah ini. Argumen yang diperlukan adalah pernyataan kebijakan dan kunci pribadi yang sesuai dengan kunci publik yang ada dalam grup kunci tepercaya untuk distribusi Anda.
2. Selanjutnya, `url_safe_base64_encode` fungsi membuat URL-safe versi tanda tangan.

```
function rsa_sha1_sign($policy, $private_key_filename) {  
    $signature = "";  
    $fp = fopen($private_key_filename, "r");  
    $priv_key = fread($fp, 8192);  
    fclose($fp);  
    $pkeyid = openssl_get_privatekey($priv_key);  
    openssl_sign($policy, $signature, $pkeyid);  
    openssl_free_key($pkeyid);  
    return $signature;  
}
```

```

}

function url_safe_base64_encode($value) {
    $encoded = base64_encode($value);
    return str_replace(
        array('+', '=', '/'),
        array('-', '_', '~'),
        $encoded);
}

```

Fungsi berikut menggunakan SHA-256 alih-alih SHA-1:

```

function rsa_sha256_sign($policy, $private_key_filename) {
    $signature = "";
    $fp = fopen($private_key_filename, "r");
    $priv_key = fread($fp, 8192);
    fclose($fp);
    $pkeyid = openssl_get_privatekey($priv_key);
    openssl_sign($policy, $signature, $pkeyid, OPENSSL_ALGO_SHA256);
    openssl_free_key($pkeyid);
    return $signature;
}

```

`rsa_sha256_sign` Fungsinya sama dengan `rsa_sha1_sign`, kecuali bahwa ia lolos `OPENSSL_ALGO_SHA256` ke `openssl_sign`. Saat Anda menggunakan SHA-256, sertakan `CloudFront-Hash-Algorithm` cookie dengan nilai `SHA256`.

Buat cookie yang ditandatangani

Kode berikut membangun membuat cookie yang ditandatangani, menggunakan atribut cookie berikut: `CloudFront-Expires`, `CloudFront-Signature`, `CloudFront-Key-Pair-Id` dan `CloudFront-Hash-Algorithm`. Kode menggunakan kebijakan khusus.

```

function create_signed_cookies($resource, $private_key_filename, $key_pair_id,
    $expires, $client_ip = null, $hash_algorithm = 'SHA1') {
    $policy = array(
        'Statement' => array(
            array(
                'Resource' => $resource,
                'Condition' => array(
                    'DateLessThan' => array('AWS:EpochTime' => $expires)
                )
            )
        )
    );
}

```

```

        )
    )
);

if ($client_ip) {
    $policy['Statement'][0]['Condition']['IpAddress'] = array('AWS:SourceIp' =>
$client_ip . '/32');
}

$policy = json_encode($policy);
$encoded_policy = url_safe_base64_encode($policy);
if ($hash_algorithm === 'SHA256') {
    $signature = rsa_sha256_sign($policy, $private_key_filename);
} else {
    $signature = rsa_sha1_sign($policy, $private_key_filename);
}
$encoded_signature = url_safe_base64_encode($signature);

$cookies = array(
    'CloudFront-Policy' => $encoded_policy,
    'CloudFront-Signature' => $encoded_signature,
    'CloudFront-Key-Pair-Id' => $key_pair_id
);

if ($hash_algorithm === 'SHA256') {
    $cookies['CloudFront-Hash-Algorithm'] = 'SHA256';
}

return $cookies;
}

```

Untuk informasi selengkapnya, lihat [Tetapkan cookie yang ditandatangani menggunakan kebijakan khusus](#).

Kode lengkap

Kode contoh berikut memberikan demonstrasi lengkap untuk membuat cookie yang CloudFront ditandatangani dengan PHP. Anda dapat mengunduh contoh lengkap dari file [demo-php.zip](#).

Dalam contoh berikut, Anda dapat memodifikasi \$policy Condition elemen untuk memungkinkan rentang alamat IPv4 dan IPv6. Sebagai contoh, lihat [Menggunakan alamat IPv6 dalam kebijakan IAM](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

```
<?php

function rsa_sha1_sign($policy, $private_key_filename) {
    $signature = "";
    $fp = fopen($private_key_filename, "r");
    $priv_key = fread($fp, 8192);
    fclose($fp);
    $pkeyid = openssl_get_privatekey($priv_key);
    openssl_sign($policy, $signature, $pkeyid);
    openssl_free_key($pkeyid);
    return $signature;
}

function url_safe_base64_encode($value) {
    $encoded = base64_encode($value);
    return str_replace(
        array('+', '=', '/'),
        array('-', '_', '~'),
        $encoded);
}

function rsa_sha256_sign($policy, $private_key_filename) {
    $signature = "";
    $fp = fopen($private_key_filename, "r");
    $priv_key = fread($fp, 8192);
    fclose($fp);
    $pkeyid = openssl_get_privatekey($priv_key);
    openssl_sign($policy, $signature, $pkeyid, OPENSSL_ALGO_SHA256);
    openssl_free_key($pkeyid);
    return $signature;
}

function create_signed_cookies($resource, $private_key_filename, $key_pair_id,
    $expires, $client_ip = null, $hash_algorithm = 'SHA1') {
    $policy = array(
        'Statement' => array(
            array(
                'Resource' => $resource,
                'Condition' => array(
                    'DateLessThan' => array('AWS:EpochTime' => $expires)
                )
            )
        )
    )
}
```

```

    );

    if ($client_ip) {
        $policy['Statement'][0]['Condition']['IpAddress'] = array('AWS:SourceIp' =>
$client_ip . '/32');
    }

    $policy = json_encode($policy);
    $encoded_policy = url_safe_base64_encode($policy);
    if ($hash_algorithm === 'SHA256') {
        $signature = rsa_sha256_sign($policy, $private_key_filename);
    } else {
        $signature = rsa_sha1_sign($policy, $private_key_filename);
    }
    $encoded_signature = url_safe_base64_encode($signature);

    $cookies = array(
        'CloudFront-Policy' => $encoded_policy,
        'CloudFront-Signature' => $encoded_signature,
        'CloudFront-Key-Pair-Id' => $key_pair_id
    );

    if ($hash_algorithm === 'SHA256') {
        $cookies['CloudFront-Hash-Algorithm'] = 'SHA256';
    }

    return $cookies;
}

$private_key_filename = '/home/test/secure/example-priv-key.pem';
$key_pair_id = 'K2JCJMDEHXQW5F';
$base_url = 'https://d1234.cloudfront.net';

$expires = time() + 3600; // 1 hour from now

// Get the viewer real IP from the x-forward-for header as $_SERVER['REMOTE_ADDR']
will return viewer facing IP. An alternative option is to use CloudFront-Viewer-
Address header. Note that this header is a trusted CloudFront immutable header. Example
format: IP:PORT ("CloudFront-Viewer-Address": "1.2.3.4:12345")
$client_ip = $_SERVER['HTTP_X_FORWARDED_FOR'];

```

```
// For HLS manifest and segments (using wildcard)
$hls_resource = $base_url . '/sign/*';
$signed_cookies = create_signed_cookies($hls_resource, $private_key_filename,
    $key_pair_id, $expires, $client_ip, 'SHA256');

// Set the cookies
$cookie_domain = parse_url($base_url, PHP_URL_HOST);
foreach ($signed_cookies as $name => $value) {
    setcookie($name, $value, $expires, '/', $cookie_domain, true, true);
}

?>

<!DOCTYPE html>
<html>
<head>
    <title>CloudFront Signed HLS Stream with Cookies</title>
</head>
<body>
    <h1>Amazon CloudFront Signed HLS Stream with Cookies</h1>
    <h2>Expires at <?php echo gmdate('Y-m-d H:i:s T', $expires); ?> only viewable by IP
    <?php echo $client_ip; ?></h2>

    <div id='hls-video'>
        <video id="video" width="640" height="360" controls></video>
    </div>

    <script src="https://cdn.jsdelivr.net/npm/hls.js@latest"></script>
    <script>
        var video = document.getElementById('video');
        var manifestUrl = '<?php echo $base_url; ?>/sign/manifest.m3u8';

        if (Hls.isSupported()) {
            var hls = new Hls();
            hls.loadSource(manifestUrl);
            hls.attachMedia(video);
        }
        else if (video.canPlayType('application/vnd.apple.mpegurl')) {
            video.src = manifestUrl;
        }
    </script>
</body>
</html>
```

Alih-alih menggunakan cookie yang ditandatangani, Anda dapat menggunakan URL yang ditandatangani. Untuk informasi selengkapnya, lihat [Buat tanda tangan URL menggunakan PHP](#).

Perintah Linux dan OpenSSL untuk pengkodean dan enkripsi base64

Anda bisa menggunakan baris perintah Linux berikut dan OpenSSL untuk membubuhkan dan menandatangani pernyataan kebijakan, mengodekan tanda tangan dengan base64, dan mengganti karakter yang tidak valid dalam parameter string kueri URL dengan karakter yang valid.

Untuk informasi tentang OpenSSL, kunjungi <https://www.openssl.org>.

SHA-1 (default):

```
cat policy | tr -d "\n" | tr -d " \t\n\r" | openssl sha1 -sign private_key.pem |  
openssl base64 -A | tr -- '+=/' '-_~'
```

SHA-256:

```
cat policy | tr -d "\n" | tr -d " \t\n\r" | openssl sha256 -sign private_key.pem |  
openssl base64 -A | tr -- '+=/' '-_~'
```

Dalam perintah sebelumnya:

- `cat` membaca `policy` file Anda.
- `tr -d "\n" | tr -d " \t\n\r"` menghapus spasi kosong dan karakter baris baru yang ditambahkan oleh `cat`.
- OpenSSL melakukan hash file SHA-1 menggunakan (SHA-256 atau) dan menandatanganinya menggunakan file kunci pribadi. `private_key.pem` Tanda tangan kunci pribadi dapat berupa RSA 2048 atau ECDSA 256. Jika Anda menggunakan SHA-256, sertakan parameter `Hash-Algorithm=SHA256` kueri di URL yang ditandatangani, atau `CloudFront-Hash-Algorithm=SHA256` cookie untuk cookie yang ditandatangani.
- OpenSSL `base64`-mengkodekan pernyataan kebijakan yang di-hash dan ditandatangani.
- `tr` mengganti karakter yang tidak valid dalam parameter string kueri URL dengan karakter yang valid.

Untuk contoh kode lainnya yang menunjukkan pembuatan tanda tangan, lihat [Contoh kode untuk membuat tanda tangan untuk URL yang ditandatangani](#).

Contoh kode untuk membuat tanda tangan untuk URL yang ditandatangani

Bagian ini menyertakan contoh aplikasi yang dapat diunduh yang mendemonstrasikan cara membuat tanda tangan untuk URL yang ditandatangani. Contoh tersedia di Perl, PHP, C #, dan Java. Anda dapat menggunakan salah satu contoh untuk membuat URL yang ditandatangani. Skrip Perl berjalan pada platform Linux dan MacOS. Contoh PHP akan bekerja pada setiap server yang menjalankan PHP. Contoh C# menggunakan Kerangka Kerja .NET.

Contoh di bagian ini digunakan SHA-1 untuk hash dan menandatangani pernyataan kebijakan. Anda juga bisa menggunakan SHA-256. Untuk menggunakannya SHA-256, perbarui algoritma hash dalam fungsi penandatanganan (misalnya, ganti sha1 dengan sha256 panggilan OpenSSL, atau gunakan konstanta yang SHA-256 setara di pustaka kriptografi bahasa Anda). Saat Anda menggunakan SHA-256, sertakan parameter Hash-Algorithm=SHA256 kueri di URL yang ditandatangani.

Misalnya kode di JavaScript (Node.js), lihat [Membuat URL yang CloudFront Ditandatangani Amazon Node.js di Blog AWS Pengembang](#).

[Misalnya kode dengan Python, lihat Menghasilkan URL yang ditandatangani untuk Amazon CloudFront di AWS SDK for Python \(Boto3\) Referensi API dan kode contoh ini di repositori Boto3.](#)

GitHub

Topik

- [Buat tanda tangan URL menggunakan Perl](#)
- [Buat tanda tangan URL menggunakan PHP](#)
- [Buat tanda tangan URL menggunakan C# dan .NET Framework](#)
- [Buat tanda tangan URL menggunakan Java](#)

Buat tanda tangan URL menggunakan Perl

Bagian ini mencakup skrip Perl untuk Linux/Mac platform yang dapat Anda gunakan untuk membuat tanda tangan untuk konten pribadi. Untuk membuat tanda tangan, jalankan skrip dengan argumen baris perintah yang menentukan CloudFront URL, jalur ke kunci pribadi penandatanganan, ID kunci, dan tanggal kedaluwarsa URL. Alat ini juga dapat menguraikan URL yang ditandatangani.

Catatan

- Membuat tanda tangan URL hanyalah satu bagian dari proses menyajikan konten pribadi menggunakan URL yang ditandatangani. Untuk informasi selengkapnya tentang proses menyeluruh, lihat [Gunakan URL yang ditandatangani](#).
- Dalam perintah penandatanganan, perhatikan yang sha1 dapat diganti dengan sha256 openssl dgst panggilan.

Topik

- [Source untuk skrip Perl untuk membuat URL yang ditandatangani](#)

Source untuk skrip Perl untuk membuat URL yang ditandatangani

Kode sumber Perl berikut dapat digunakan untuk membuat URL yang ditandatangani untuk CloudFront. Komentar dalam kode mencakup informasi tentang saklar baris perintah dan fitur alat.

```
#!/usr/bin/perl -w

# Copyright 2008 Amazon Technologies, Inc. Licensed under the Apache License, Version
# 2.0 (the "License");
# you may not use this file except in compliance with the License. You may obtain a
# copy of the License at:
#
# https://aws.amazon.com/apache2.0
#
# This file is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
# KIND, either express or implied.
# See the License for the specific language governing permissions and limitations under
# the License.

=head1 cfsign.pl

cfsign.pl - A tool to generate and verify Amazon CloudFront signed URLs

=head1 SYNOPSIS

This script uses an existing RSA key pair to sign and verify Amazon CloudFront signed
URLs
```

View the script source for details as to which CPAN packages are required beforehand.

For help, try:

```
cfsign.pl --help
```

URL signing examples:

```
cfsign.pl --action encode --url https://images.my-website.com/gallery1.zip --policy
sample_policy.json --private-key privkey.pem --key-pair-id mykey
```

```
cfsign.pl --action encode --url https://images.my-website.com/gallery1.zip --expires
1257439868 --private-key privkey.pem --key-pair-id mykey
```

URL decode example:

```
cfsign.pl --action decode --url "http://mydist.cloudfront.net/?Signature=AG0-
PgXkYo99MkJFHvjfGXjG1QDEXeaDb4Qtzmy85wqyJjK7eKojQWa4BCRcow__&Policy=eyJTdGF0ZWl1bnQiOi0t7I1Jlc29
Pair-Id=mykey"
```

To generate an RSA key pair, you can use openssl and the following commands:

```
# Generate a 2048 bit key pair
openssl genrsa -out private-key.pem 2048
openssl rsa -in private-key.pem -pubout -out public-key.pem
```

```
=head1 OPTIONS
```

```
=over 8
```

```
=item B<--help>
```

Print a help message and exits.

```
=item B<--action> [action]
```

The action to execute. action can be one of:

- encode - Generate a signed URL (using a canned policy or a user policy)
- decode - Decode a signed URL

```
=item B<--url>
```

The URL to en/decode

```
=item B<--stream>
```

The stream to en/decode

```
=item B<--private-key>
```

The path to your private key.

```
=item B<--key-pair-id>
```

The key pair identifier.

```
=item B<--policy>
```

The CloudFront policy document.

```
=item B<--expires>
```

The Unix epoch time when the URL is to expire. If both this option and the --policy option are specified, --policy will be used. Otherwise, this option alone will use a canned policy.

```
=back
```

```
=cut
```

```
use strict;  
use warnings;
```

```
# you might need to use CPAN to get these modules.  
# run perl -MCPAN -e "install <module>" to get them.  
# The openssl command line will also need to be in your $PATH.  
use File::Temp qw/tempfile/;  
use File::Slurp;  
use Getopt::Long;  
use IPC::Open2;  
use MIME::Base64 qw(encode_base64 decode_base64);  
use Pod::Usage;  
use URI;
```

```
my $CANNED_POLICY
```

```

    = '{"Statement":[{"Resource":"<RESOURCE>","Condition":{"DateLessThan":
{"AWS:EpochTime":<EXPIRES>}}}}]';

my $POLICY_PARAM      = "Policy";
my $EXPIRES_PARAM     = "Expires";
my $SIGNATURE_PARAM   = "Signature";
my $KEY_PAIR_ID_PARAM = "Key-Pair-Id";

my $verbose = 0;
my $policy_filename = "";
my $expires_epoch = 0;
my $action = "";
my $help = 0;
my $key_pair_id = "";
my $url = "";
my $stream = "";
my $private_key_filename = "";

my $result = GetOptions("action=s"      => \$action,
                        "policy=s"      => \$policy_filename,
                        "expires=i"     => \$expires_epoch,
                        "private-key=s" => \$private_key_filename,
                        "key-pair-id=s" => \$key_pair_id,
                        "verbose"       => \$verbose,
                        "help"          => \$help,
                        "url=s"         => \$url,
                        "stream=s"      => \$stream,
                        );

if ($help or !$result) {
    pod2usage(1);
    exit;
}

if ($url eq "" and $stream eq "") {
    print STDERR "Must include a stream or a URL to encode or decode with the --stream
or --url option\n";
    exit;
}

if ($url ne "" and $stream ne "") {
    print STDERR "Only one of --url and --stream may be specified\n";
    exit;
}

```

```

if ($url ne "" and !is_url_valid($url)) {
    exit;
}

if ($stream ne "") {
    exit unless is_stream_valid($stream);

    # The signing mechanism is identical, so from here on just pretend we're
    # dealing with a URL
    $url = $stream;
}

if ($action eq "encode") {
    # The encode action will generate a private content URL given a base URL,
    # a policy file (or an expires timestamp) and a key pair id parameter
    my $private_key;
    my $public_key;
    my $public_key_file;

    my $policy;
    if ($policy_filename eq "") {
        if ($expires_epoch == 0) {
            print STDERR "Must include policy filename with --policy argument or an
expires" .
                "time using --expires\n";
        }

        $policy = $CANNED_POLICY;
        $policy =~ s/<EXPIRES>/$expires_epoch/g;
        $policy =~ s/<RESOURCE>/$url/g;
    } else {
        if (! -e $policy_filename) {
            print STDERR "Policy file $policy_filename does not exist\n";
            exit;
        }
        $expires_epoch = 0; # ignore if set
        $policy = read_file($policy_filename);
    }

    if ($private_key_filename eq "") {
        print STDERR "You must specific the path to your private key file with --
private-key\n";
        exit;
    }

```

```
}

if (! -e $private_key_filename) {
    print STDERR "Private key file $private_key_filename does not exist\n";
    exit;
}

if ($key_pair_id eq "") {
    print STDERR "You must specify a key pair id with --key-pair-id\n";
    exit;
}

my $encoded_policy = url_safe_base64_encode($policy);
my $signature = rsa_sha1_sign($policy, $private_key_filename);
my $encoded_signature = url_safe_base64_encode($signature);

my $generated_url = create_url($url, $encoded_policy, $encoded_signature,
$key_pair_id, $expires_epoch);

if ($stream ne "") {
    print "Encoded stream (for use within a swf):\n" . $generated_url . "\n";
    print "Encoded and escaped stream (for use on a webpage):\n" .
escape_url_for_webpage($generated_url) . "\n";
} else {
    print "Encoded URL:\n" . $generated_url . "\n";
}
} elsif ($action eq "decode") {
    my $decoded = decode_url($url);
    if (!$decoded) {
        print STDERR "Improperly formed URL\n";
        exit;
    }

    print_decoded_url($decoded);
} else {
    # No action specified, print help. But only if this is run as a program (caller
will be empty)
    pod2usage(1) unless caller();
}

# Decode a private content URL into its component parts
sub decode_url {
    my $url = shift;
```

```

if ($url =~ /(.*?)\?(.*)/) {
    my $base_url = $1;
    my $params = $2;

    my @unparsed_params = split(/&/, $params);
    my %params = ();
    foreach my $param (@unparsed_params) {
        my ($key, $val) = split(/=/, $param);
        $params{$key} = $val;
    }

    my $encoded_signature = "";
    if (exists $params{$SIGNATURE_PARAM}) {
        $encoded_signature = $params{"Signature"};
    } else {
        print STDERR "Missing Signature URL parameter\n";
        return 0;
    }

    my $encoded_policy = "";
    if (exists $params{$POLICY_PARAM}) {
        $encoded_policy = $params{$POLICY_PARAM};
    } else {
        if (!exists $params{$EXPIRES_PARAM}) {
            print STDERR "Either the Policy or Expires URL parameter needs to be
specified\n";
            return 0;
        }

        my $expires = $params{$EXPIRES_PARAM};

        my $policy = $CANNED_POLICY;
        $policy =~ s/<EXPIRES>/ $expires /g;

        my $url_without_cf_params = $url;
        $url_without_cf_params =~ s/$SIGNATURE_PARAM=[^&]*&?//g;
        $url_without_cf_params =~ s/$POLICY_PARAM=[^&]*&?//g;
        $url_without_cf_params =~ s/$EXPIRES_PARAM=[^&]*&?//g;
        $url_without_cf_params =~ s/$KEY_PAIR_ID_PARAM=[^&]*&?//g;

        if ($url_without_cf_params =~ /(.*?)\?$/) {
            $url_without_cf_params = $1;
        }
    }
}

```

```

        $policy =~ s/<RESOURCE>/$url_without_cf_params/g;

        $encoded_policy = url_safe_base64_encode($policy);
    }

    my $key = "";
    if (exists $params{$KEY_PAIR_ID_PARAM}) {
        $key = $params{$KEY_PAIR_ID_PARAM};
    } else {
        print STDERR "Missing $KEY_PAIR_ID_PARAM parameter\n";
        return 0;
    }

    my $policy = url_safe_base64_decode($encoded_policy);

    my %ret = ();
    $ret{"base_url"} = $base_url;
    $ret{"policy"} = $policy;
    $ret{"key"} = $key;

    return \%ret;
} else {
    return 0;
}
}

# Print a decoded URL out
sub print_decoded_url {
    my $decoded = shift;

    print "Base URL: \n" . $decoded->{"base_url"} . "\n";
    print "Policy: \n" . $decoded->{"policy"} . "\n";
    print "Key: \n" . $decoded->{"key"} . "\n";
}

# Encode a string with base 64 encoding and replace some invalid URL characters
sub url_safe_base64_encode {
    my ($value) = @_ ;

    my $result = encode_base64($value);
    $result =~ tr|+|=/_~|;

    return $result;
}

```



```

}

# Decode a string with base 64 encoding.  URL-decode the string first
# followed by reversing any special character ("+=/") translation.
sub url_safe_base64_decode {
    my ($value) = @_;

    $value =~ s/%([0-9A-Fa-f]{2})/chr(hex($1))/eg;
    $value =~ tr|_|_|+=/|;

    my $result = decode_base64($value);

    return $result;
}

# Create a private content URL
sub create_url {
    my ($path, $policy, $signature, $key_pair_id, $expires) = @_;

    my $result;
    my $separator = $path =~ /\?/ ? '&' : '?';
    if ($expires) {
        $result = "$path$separator$EXPIRES_PARAM=$expires&$SIGNATURE_PARAM=$signature&
$KEY_PAIR_ID_PARAM=$key_pair_id";
    } else {
        $result = "$path$separator$POLICY_PARAM=$policy&$SIGNATURE_PARAM=$signature&
$KEY_PAIR_ID_PARAM=$key_pair_id";
    }
    $result =~ s/\n//g;

    return $result;
}

# Sign a document with given private key file.
# The first argument is the document to sign
# The second argument is the name of the private key file
sub rsa_sha1_sign {
    my ($to_sign, $pvkFile) = @_;
    print "openssl sha1 -sign $pvkFile $to_sign\n";

    return write_to_program($pvkFile, $to_sign);
}

# Helper function to write data to a program

```

```
sub write_to_program {
my ($keyfile, $data) = @_;
unlink "temp_policy.dat" if (-e "temp_policy.dat");
unlink "temp_sign.dat" if (-e "temp_sign.dat");

write_file("temp_policy.dat", $data);

system("openssl dgst -sha1 -sign \"\$keyfile\" -out temp_sign.dat temp_policy.dat");

my $output = read_file("temp_sign.dat");

    return $output;
}

# Read a file into a string and return the string
sub read_file {
    my ($file) = @_;

    open(INFILE, "<$file") or die("Failed to open $file: $!");
    my $str = join('', <INFILE>);
    close INFILE;

    return $str;
}

sub is_url_valid {
    my ($url) = @_;

    # HTTP distributions start with http[s]:// and are the correct thing to sign
    if ($url =~ /^https?:\\\/\\\/) {
        return 1;
    } else {
        print STDERR "CloudFront requires absolute URLs for HTTP distributions\\n";
        return 0;
    }
}

sub is_stream_valid {
    my ($stream) = @_;

    if ($stream =~ /^rtmp:\\\/\\\/ or $stream =~ /^\\/?.cfx\\\/st/) {
        print STDERR "Streaming distributions require that only the stream name is
signed.\\n";
    }
}
```

```

        print STDERR "The stream name is everything after, but not including, cfx/st/
\n";
        return 0;
    } else {
        return 1;
    }
}

# flash requires that the query parameters in the stream name are url
# encoded when passed in through javascript, etc. This sub handles the minimal
# required url encoding.
sub escape_url_for_webpage {
    my ($url) = @_;

    $url =~ s/\?/%3F/g;
    $url =~ s/=/%3D/g;
    $url =~ s/&/%26/g;

    return $url;
}

1;

```

Buat tanda tangan URL menggunakan PHP

Setiap server web yang menjalankan PHP dapat menggunakan kode contoh PHP ini untuk membuat pernyataan kebijakan dan tanda tangan untuk distribusi pribadi CloudFront. Contoh lengkap menciptakan halaman web yang berfungsi dengan tautan URL yang ditandatangani yang memutar streaming video menggunakan CloudFront streaming. Anda dapat mengunduh contoh lengkap dari file [demo-php.zip](#).

Catatan

- Membuat tanda tangan URL hanyalah satu bagian dari proses menyajikan konten pribadi menggunakan URL yang ditandatangani. Untuk informasi selengkapnya tentang seluruh proses, lihat [Gunakan URL yang ditandatangani](#).
- Anda juga dapat membuat URL yang ditandatangani dengan menggunakan kelas AWS SDK untuk PHP di `UrlSigner`. Untuk informasi selengkapnya, lihat [Kelas UrlSigner](#) di Referensi AWS SDK untuk PHP API.

- Dalam `openssl_sign` panggilan tersebut, perhatikan bahwa passing `OPENSSL_ALGO_SHA256` sebagai argumen keempat beralih ke SHA-256. (Lihat juga [Buat cookie yang ditandatangani menggunakan PHP](#) untuk contoh lengkap.)

Topik

- [Buat tanda tangan RSA SHA-1](#)
- [Buat kebijakan kalengan](#)
- [Buat kebijakan kustom](#)
- [Contoh kode lengkap](#)

Bagian berikut memecah contoh kode menjadi bagian-bagian individual. Anda dapat menemukan di [Contoh kode lengkap](#) bawah ini.

Buat tanda tangan RSA SHA-1

Contoh kode ini melakukan hal berikut:

- Fungsi melakukan `rsa_sha1_sign` hash dan menandatangani pernyataan kebijakan. Argumen yang diperlukan adalah pernyataan kebijakan dan kunci pribadi yang sesuai dengan kunci publik yang ada dalam grup kunci tepercaya untuk distribusi Anda.
- Selanjutnya, `url_safe_base64_encode` fungsi membuat URL-safe versi tanda tangan.

```
function rsa_sha1_sign($policy, $private_key_filename) {  
    $signature = "";  
  
    // load the private key  
    $fp = fopen($private_key_filename, "r");  
    $priv_key = fread($fp, 8192);  
    fclose($fp);  
    $pkeyid = openssl_get_privatekey($priv_key);  
  
    // compute signature  
    openssl_sign($policy, $signature, $pkeyid);  
  
    // free the key from memory  
    openssl_free_key($pkeyid);  
}
```

```

    return $signature;
}

function url_safe_base64_encode($value) {
    $encoded = base64_encode($value);
    // replace unsafe characters +, = and / with
    // the safe characters -, _ and ~
    return str_replace(
        array('+', '=', '/'),
        array('-', '_', '~'),
        $encoded);
}

```

Cuplikan kode berikut menggunakan fungsi `get_canned_policy_stream_name()` dan `get_custom_policy_stream_name()` untuk membuat kebijakan kalengan dan kustom. CloudFront menggunakan kebijakan untuk membuat URL untuk streaming video, termasuk menentukan waktu kedaluwarsa.

Anda kemudian dapat menggunakan kebijakan kalengan atau kebijakan khusus untuk menentukan cara mengelola akses ke konten Anda. Untuk informasi selengkapnya tentang mana yang harus dipilih, lihat [Memutuskan untuk menggunakan kebijakan kalengan atau kustom untuk URL yang ditandatangani](#) bagian.

Buat kebijakan kalengan

Contoh kode berikut membangun kalengan pernyataan kebijakan untuk tanda tangan.

Note

`$expiresVariabel` adalah date/time stempel yang harus berupa bilangan bulat, bukan string.

```

function get_canned_policy_stream_name($video_path, $private_key_filename,
    $key_pair_id, $expires) {
    // this policy is well known by CloudFront, but you still need to sign it, since it
    contains your parameters
    $canned_policy = '{"Statement":[{"Resource":"' . $video_path . '", "Condition":
{"DateLessThan":{"AWS:EpochTime":"' . $expires . '"} }]}';
    // the policy contains characters that cannot be part of a URL, so we base64 encode
    it
    $encoded_policy = url_safe_base64_encode($canned_policy);
}

```

```
// sign the original policy, not the encoded version
$signature = rsa_sha1_sign($canned_policy, $private_key_filename);
// make the signature safe to be included in a URL
$encoded_signature = url_safe_base64_encode($signature);

// combine the above into a stream name
$stream_name = create_stream_name($video_path, null, $encoded_signature,
$key_pair_id, $expires);
// URL-encode the query string characters
return $stream_name;
}
```

Untuk informasi selengkapnya tentang kebijakan terekam, lihat [Membuat URL yang ditandatangani menggunakan kebijakan kalengan](#).

Buat kebijakan kustom

Contoh kode berikut membangun khusus pernyataan kebijakan untuk tanda tangan.

```
function get_custom_policy_stream_name($video_path, $private_key_filename,
$key_pair_id, $policy) {
    // the policy contains characters that cannot be part of a URL, so we base64 encode
    it
    $encoded_policy = url_safe_base64_encode($policy);
    // sign the original policy, not the encoded version
    $signature = rsa_sha1_sign($policy, $private_key_filename);
    // make the signature safe to be included in a URL
    $encoded_signature = url_safe_base64_encode($signature);

    // combine the above into a stream name
    $stream_name = create_stream_name($video_path, $encoded_policy, $encoded_signature,
$key_pair_id, null);
    // URL-encode the query string characters
    return $stream_name;
}
```

Untuk informasi selengkapnya tentang kebijakan khusus, lihat [Membuat URL yang ditandatangani menggunakan kebijakan khusus](#).

Contoh kode lengkap

Kode contoh berikut memberikan demonstrasi lengkap untuk membuat URL yang CloudFront ditandatangani dengan PHP. Anda dapat mengunduh contoh lengkap dari file [demo-php.zip](#).

Dalam contoh berikut, Anda dapat memodifikasi `$policy` Condition elemen untuk memungkinkan rentang alamat IPv4 dan IPv6. Sebagai contoh, lihat [Menggunakan alamat IPv6 dalam kebijakan IAM](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

```
<?php

function rsa_sha1_sign($policy, $private_key_filename) {
    $signature = "";

    // load the private key
    $fp = fopen($private_key_filename, "r");
    $priv_key = fread($fp, 8192);
    fclose($fp);
    $pkeyid = openssl_get_privatekey($priv_key);

    // compute signature
    openssl_sign($policy, $signature, $pkeyid);

    // free the key from memory
    openssl_free_key($pkeyid);

    return $signature;
}

function url_safe_base64_encode($value) {
    $encoded = base64_encode($value);
    // replace unsafe characters +, = and / with the safe characters -, _ and ~
    return str_replace(
        array('+', '=', '/'),
        array('-', '_', '~'),
        $encoded);
}

function create_stream_name($stream, $policy, $signature, $key_pair_id, $expires) {
    $result = $stream;
    // if the stream already contains query parameters, attach the new query parameters
    // to the end
    // otherwise, add the query parameters
    $separator = strpos($stream, '?') == FALSE ? '?' : '&';
    // the presence of an expires time means we're using a canned policy
    if($expires) {
        $result .= $separator . "Expires=" . $expires . "&Signature=" . $signature .
        "&Key-Pair-Id=" . $key_pair_id;
    }
}
```

```

    }
    // not using a canned policy, include the policy itself in the stream name
    else {
        $result .= $separator . "Policy=" . $policy . "&Signature=" . $signature .
"&Key-Pair-Id=" . $key_pair_id;
    }

    // new lines would break us, so remove them
    return str_replace('\n', '', $result);
}

function get_canned_policy_stream_name($video_path, $private_key_filename,
$key_pair_id, $expires) {
    // this policy is well known by CloudFront, but you still need to sign it, since it
contains your parameters
    $canned_policy = '{"Statement":[{"Resource":"' . $video_path . '", "Condition":
{"DateLessThan":{"AWS:EpochTime":'. $expires . '}}]}]';
    // the policy contains characters that cannot be part of a URL, so we base64 encode
it
    $encoded_policy = url_safe_base64_encode($canned_policy);
    // sign the original policy, not the encoded version
    $signature = rsa_sha1_sign($canned_policy, $private_key_filename);
    // make the signature safe to be included in a URL
    $encoded_signature = url_safe_base64_encode($signature);

    // combine the above into a stream name
    $stream_name = create_stream_name($video_path, null, $encoded_signature,
$key_pair_id, $expires);
    // URL-encode the query string characters
    return $stream_name;
}

function get_custom_policy_stream_name($video_path, $private_key_filename,
$key_pair_id, $policy) {
    // the policy contains characters that cannot be part of a URL, so we base64 encode
it
    $encoded_policy = url_safe_base64_encode($policy);
    // sign the original policy, not the encoded version
    $signature = rsa_sha1_sign($policy, $private_key_filename);
    // make the signature safe to be included in a URL
    $encoded_signature = url_safe_base64_encode($signature);

    // combine the above into a stream name

```



```

    $stream_name = create_stream_name($video_path, $encoded_policy, $encoded_signature,
    $key_pair_id, null);
    // URL-encode the query string characters
    return $stream_name;
}

// Path to your private key. Be very careful that this file is not accessible
// from the web!

$private_key_filename = '/home/test/secure/example-priv-key.pem';
$key_pair_id = 'K2JCJMDEHXQW5F';

// Make sure you have "Restrict viewer access" enabled on this path behaviour and using
the above Trusted key groups (recommended).
$video_path = 'https://example.com/secure/example.mp4';

$expires = time() + 300; // 5 min from now
$canned_policy_stream_name = get_canned_policy_stream_name($video_path,
    $private_key_filename, $key_pair_id, $expires);

// Get the viewer real IP from the x-forward-for header as $_SERVER['REMOTE_ADDR']
will return viewer facing IP. An alternative option is to use CloudFront-Viewer-
Address header. Note that this header is a trusted CloudFront immutable header. Example
format: IP:PORT ("CloudFront-Viewer-Address": "1.2.3.4:12345")
$client_ip = $_SERVER['HTTP_X_FORWARDED_FOR'];
$policy =
'{'.
    '"Statement":['.
        '{'.
            '"Resource":' . $video_path . ','.
            '"Condition":{'.
                '"IpAddress":{"AWS:SourceIp":"' . $client_ip . '/32"},'.
                '"DateLessThan":{"AWS:EpochTime":"' . $expires . '}'.
            '}'.
        '}'.
    ']' .
    '}' ;

$custom_policy_stream_name = get_custom_policy_stream_name($video_path,
    $private_key_filename, $key_pair_id, $policy);

?>

<html>

```

```
<head>
  <title>CloudFront</title>
</head>

<body>
  <h1>Amazon CloudFront</h1>
  <h2>Canned Policy</h2>
  <h3>Expires at <?php echo gmdate('Y-m-d H:i:s T', $expires); ?></h3>
  <br />

  <div id='canned'>The canned policy video will be here: <br>

    <video width="640" height="360" autoplay muted controls>
      <source src="<?php echo $canned_policy_stream_name; ?>" type="video/mp4">
      Your browser does not support the video tag.
    </video>
  </div>

  <h2>Custom Policy</h2>
  <h3>Expires at <?php echo gmdate('Y-m-d H:i:s T', $expires); ?> only viewable by IP
  <?php echo $client_ip; ?></h3>
  <div id='custom'>The custom policy video will be here: <br>

    <video width="640" height="360" autoplay muted controls>
      <source src="<?php echo $custom_policy_stream_name; ?>" type="video/mp4">
      Your browser does not support the video tag.
    </video>
  </div>

</body>

</html>
```

Untuk contoh tanda tangan URL tambahan, lihat topik berikut:

- [Buat tanda tangan URL menggunakan Perl](#)
- [Buat tanda tangan URL menggunakan C# dan .NET Framework](#)
- [Buat tanda tangan URL menggunakan Java](#)

Alih-alih menggunakan URL yang ditandatangani untuk membuat tanda tangan, Anda dapat menggunakan cookie yang ditandatangani. Untuk informasi selengkapnya, lihat [Buat cookie yang ditandatangani menggunakan PHP](#).

Buat tanda tangan URL menggunakan C# dan .NET Framework

Contoh C# di bagian ini mengimplementasikan contoh aplikasi yang menunjukkan cara membuat tanda tangan untuk distribusi CloudFront pribadi menggunakan pernyataan kebijakan kalengan dan kustom. Contoh termasuk fungsi utilitas berdasarkan [AWS SDK untuk .NET](#) yang dapat berguna dalam aplikasi .NET.

Anda juga dapat membuat URL dan cookie yang ditandatangani dengan menggunakan SDK untuk .NET. Di Referensi API SDK untuk .NET, lihat topik berikut:

- URL yang ditandatangani - [AmazonCloudFrontUrlSigner](#)
- Cookie yang ditandatangani — [AmazonCloudFrontCookieSigner](#)

Untuk mengunduh kode, kunjungi [Kode Tanda Tangan dalam C#](#).

Catatan

- `AmazonCloudFrontCookieSigner` dan `AmazonCloudFrontUrlSigner` telah pindah ke paket terpisah. Untuk informasi selengkapnya tentang menggunakannya, lihat [CookieSigner dan UrlSigner](#) di Panduan Pengembang AWS SDK untuk .NET (V4).
- Membuat tanda tangan URL hanyalah satu bagian dari proses menyajikan konten pribadi menggunakan URL yang ditandatangani. Untuk informasi selengkapnya, lihat [Gunakan URL yang ditandatangani](#). Untuk informasi selengkapnya tentang penggunaan cookie yang ditandatangani, lihat [Gunakan cookie yang ditandatangani](#).
- Dalam panggilan penandatanganan RSA, perhatikan bahwa SHA1 dapat diganti dengan SHA256 parameter algoritma hash.

Gunakan kunci RSA di .NET Framework

Untuk menggunakan kunci RSA di .NET Framework, Anda harus mengonversi file.pem yang AWS sediakan ke format XML.NET Framework yang digunakan.NET Framework.

Setelah konversi, file kunci privat RSA memiliki format berikut:

Example: Kunci pribadi RSA dalam format XML.NET Framework

```
<RSAKeyValue>
  <Modulus>
    w05IvYCP5UcoCKDo1dcspoMehWBZcyfs9QEzGi60e5y+ewGr1oW+vB2GPB
    ANBiVPcUHTFWhwaIBd3oglmF0lGQljP/j0fmXHUK2kUUnLnJp+o0BL2NiuFtqcW6h/L5lIpD8Yq+NRHg
    Ty4zDsyr2880MvXv88yEFURckqEXAMPLE=
  </Modulus>
  <Exponent>AQAB</Exponent>
  <P>
    5bmKDaTz
    npENGvqz4Cea8XPH+sxt+2VaAwYnsarVUoSBeVt8WLl0VuZGG9IZYmH5KteXEu7fZveYd9UEXAMPLE==
  </P>
  <Q>
    1v9l/WN1a1N3r0K4VGoCokx7kR2SyTMSbZgF9IWJN0ugR/WZw7HTnjip03c9dy1Ms9pUKwUF4
    6d7049EXAMPLE==
  </Q>
  <DP>
    RgrSKuLWXMyBH+/l1Dx/I4tXuAJIrlPyo+Vmi0c7b5NzHptkSHEPfR9s1
    0K0VqjknclqCJ3Ig860MEtEXAMPLE==
  </DP>
  <DQ>
    pjPjvSFw+RoaTu0pgCA/jwW/FGyfN6iim1RFbkT4
    z49DZb2IM885f3vf35eLTaEYRYUHqgZtChNEV0TEXAMPLE==
  </DQ>
  <InverseQ>
    nkV0JTg5QtGNgWb9i
    cVtzrL/1pFE0HbJXwEJdU99N+7sMK+1066DL/HSBUCD63qD4USpnf0myc24in0EXAMPLE==</InverseQ>
  <D>
    Bc7mp7XYHynuPZxChjWNJZIq+A73gm0ASDv6At7F8Vi9r0xUlQe/v0AQs3ycN8QlyR4XMbzMLYk
    3yJxFDXo4ZKQt0GzLGteCU2srANiLv26/imXA8FVidZftTAtLviWQZBVPTeYIA69ATUYPEq0a5u5wjGy
    U0ij90WyuEXAMPLE=
  </D>
</RSAKeyValue>
```

Metode penandatanganan kebijakan kalengan di C

Kode C# berikut membuat URL yang ditandatangani yang menggunakan kebijakan terekam dengan melakukan hal berikut:

- Membuat pernyataan kebijakan.

- Menanamkan pernyataan kebijakan dengan menggunakan SHA1, dan menandatangani hasil menggunakan RSA dan kunci pribadi yang kunci publik terkaitnya berada dalam kelompok kunci yang dipercaya.
- Base64-encodes pernyataan kebijakan yang di-hash dan ditandatangani dan menggantikan karakter khusus untuk membuat string aman digunakan sebagai parameter permintaan URL.
- Menyusun nilai.

Untuk implementasi lengkap, lihat contoh di [Kode Tanda Tangan dalam C#](#).

Note

keyIdIni dikembalikan saat Anda mengunggah kunci publik ke CloudFront. Untuk informasi lebih lanjut, lihat



[& Key-Pair-Id](#).

Example: Metode penandatanganan kebijakan kalengan di C #

```
public static string ToUrlSafeBase64String(byte[] bytes)
{
    return System.Convert.ToBase64String(bytes)
        .Replace('+', '-')
        .Replace('=', '~')
        .Replace('/', '~');
}

public static string CreateCannedPrivateURL(string urlString,
    string durationUnits, string durationNumber, string pathToPolicyStmnt,
    string pathToPrivateKey, string keyId)
{
    // args[] 0-thisMethod, 1-resourceUrl, 2-seconds-minutes-hours-days
    // to expiration, 3-numberOfPreviousUnits, 4-pathToPolicyStmnt,
    // 5-pathToPrivateKey, 6-keyId

    TimeSpan timeSpanInterval = GetDuration(durationUnits, durationNumber);

    // Create the policy statement.
    string strPolicy = CreatePolicyStatement(pathToPolicyStmnt,
        urlString,
```

```

        DateTime.Now,
        DateTime.Now.Add(TimeSpanInterval),
        "0.0.0.0/0");
if ("Error!" == strPolicy) return "Invalid time frame." +
    "Start time cannot be greater than end time.";

// Copy the expiration time defined by policy statement.
string strExpiration = CopyExpirationTimeFromPolicy(strPolicy);

// Read the policy into a byte buffer.
byte[] bufferPolicy = Encoding.ASCII.GetBytes(strPolicy);

// Initialize the SHA1CryptoServiceProvider object and hash the policy data.
using (SHA1CryptoServiceProvider
    cryptoSHA1 = new SHA1CryptoServiceProvider())
{
    bufferPolicy = cryptoSHA1.ComputeHash(bufferPolicy);

    // Initialize the RSACryptoServiceProvider object.
    RSACryptoServiceProvider providerRSA = new RSACryptoServiceProvider();
    XmlDocument xmlPrivateKey = new XmlDocument();

    // Load your private key, which you created by converting your
    // .pem file to the XML format that the .NET framework uses.
    // Several tools are available.
    xmlPrivateKey.Load(pathToPrivateKey);

    // Format the RSACryptoServiceProvider providerRSA and
    // create the signature.
    providerRSA.FromXmlString(xmlPrivateKey.InnerXml);
    RSAPKCS1SignatureFormatter rsaFormatter =
        new RSAPKCS1SignatureFormatter(providerRSA);
    rsaFormatter.SetHashAlgorithm("SHA1");
    byte[] signedPolicyHash = rsaFormatter.CreateSignature(bufferPolicy);

    // Convert the signed policy to URL-safe base64 encoding and
    // replace unsafe characters + = / with the safe characters - _ ~
    string strSignedPolicy = ToUrlSafeBase64String(signedPolicyHash);

    // Concatenate the URL, the timestamp, the signature,
    // and the key pair ID to form the signed URL.
    return urlString +
        "?Expires=" +
        strExpiration +

```

```
        "&Signature=" +  
        strSignedPolicy +  
        "&Key-Pair-Id=" +  
        keyId;  
    }  
}
```

Metode penandatanganan kebijakan khusus di C

Kode C# berikut membuat URL yang ditandatangani menggunakan kebijakan kustom dengan melakukan hal berikut:

1. Membuat pernyataan kebijakan.
2. Base64-encodes pernyataan kebijakan dan menggantikan karakter khusus untuk membuat string aman digunakan sebagai parameter permintaan URL.
3. Menanamkan pernyataan kebijakan menggunakan SHA1, dan mengenkripsi hasil menggunakan RSA dan kunci pribadi yang kunci publik terkaitnya berada dalam grup kunci yang dipercaya.
4. Base64-encodes pernyataan kebijakan yang di-hash dan mengganti karakter khusus untuk membuat string aman digunakan sebagai parameter permintaan URL.
5. Menyusun nilai.

Untuk implementasi lengkap, lihat contoh di [Kode Tanda Tangan dalam C#](#).

Note

keyIdIni dikembalikan saat Anda mengunggah kunci publik ke CloudFront. Untuk informasi lebih lanjut, lihat



[& Key-Pair-Id](#).

Example: Metode penandatanganan kebijakan khusus di C

```
public static string ToUrlSafeBase64String(byte[] bytes)  
{  
    return System.Convert.ToBase64String(bytes)  
        .Replace('+', '-')  
        .Replace('=', '_')  
        .Replace('/', '~');
```

```

}

public static string CreateCustomPrivateURL(string urlString,
    string durationUnits, string durationNumber, string startIntervalFromNow,
    string ipaddress, string pathToPolicyStmnt, string pathToPrivateKey,
    string keyId)
{
    // args[] 0-thisMethod, 1-resourceUrl, 2-seconds-minutes-hours-days
    // to expiration, 3-numberOfPreviousUnits, 4-starttimeFromNow,
    // 5-ip_address, 6-pathToPolicyStmnt, 7-pathToPrivateKey, 8-keyId

    TimeSpan timeSpanInterval = GetDuration(durationUnits, durationNumber);
    TimeSpan timeSpanToStart = GetDurationByUnits(durationUnits,
        startIntervalFromNow);
    if (null == timeSpanToStart)
        return "Invalid duration units." +
            "Valid options: seconds, minutes, hours, or days";

    string strPolicy = CreatePolicyStatement(
        pathToPolicyStmnt, urlString, DateTime.Now.Add(timeSpanToStart),
        DateTime.Now.Add(timeSpanInterval), ipaddress);

    // Read the policy into a byte buffer.
    byte[] bufferPolicy = Encoding.ASCII.GetBytes(strPolicy);

    // Convert the policy statement to URL-safe base64 encoding and
    // replace unsafe characters + = / with the safe characters - _ ~

    string urlSafePolicy = ToUrlSafeBase64String(bufferPolicy);

    // Initialize the SHA1CryptoServiceProvider object and hash the policy data.
    byte[] bufferPolicyHash;
    using (SHA1CryptoServiceProvider cryptoSHA1 =
        new SHA1CryptoServiceProvider())
    {
        bufferPolicyHash = cryptoSHA1.ComputeHash(bufferPolicy);

        // Initialize the RSACryptoServiceProvider object.
        RSACryptoServiceProvider providerRSA = new RSACryptoServiceProvider();
        XmlDocument xmlPrivateKey = new XmlDocument();

        // Load your private key, which you created by converting your
        // .pem file to the XML format that the .NET framework uses.
        // Several tools are available.

```



```

        xmlPrivateKey.Load(pathToPrivateKey);

        // Format the RSACryptoServiceProvider providerRSA
        // and create the signature.
        providerRSA.FromXmlString(xmlPrivateKey.InnerXml);
        RSAPKCS1SignatureFormatter RSAFormatter =
            new RSAPKCS1SignatureFormatter(providerRSA);
        RSAFormatter.SetHashAlgorithm("SHA1");
        byte[] signedHash = RSAFormatter.CreateSignature(bufferPolicyHash);

        // Convert the signed policy to URL-safe base64 encoding and
        // replace unsafe characters + = / with the safe characters - _ ~
        string strSignedPolicy = ToUrlSafeBase64String(signedHash);

        return urlString +
            "?Policy=" +
            urlSafePolicy +
            "&Signature=" +
            strSignedPolicy +
            "&Key-Pair-Id=" +
            keyId;
    }
}

```

Metode utilitas untuk pembuatan tanda tangan

Metode berikut ini mendapatkan pernyataan kebijakan dari interval waktu file dan parse untuk pembuatan tanda tangan.

Example: Metode utilitas untuk pembuatan tanda tangan

```

public static string CreatePolicyStatement(string policyStmnt,
    string resourceUrl,
    DateTime startTime,
    DateTime endTime,
    string ipAddress)

{
    // Create the policy statement.
    FileStream streamPolicy = new FileStream(policyStmnt, FileMode.Open,
    FileAccess.Read);
    using (StreamReader reader = new StreamReader(streamPolicy))
    {
        string strPolicy = reader.ReadToEnd();
    }
}

```

```

    TimeSpan startTimeSpanFromNow = (startTime - DateTime.Now);
    TimeSpan endTimeSpanFromNow = (endTime - DateTime.Now);
    TimeSpan intervalStart =
        (DateTime.UtcNow.Add(startTimeSpanFromNow)) -
        new DateTime(1970, 1, 1, 0, 0, 0, DateTimeKind.Utc);
    TimeSpan intervalEnd =
        (DateTime.UtcNow.Add(endTimeSpanFromNow)) -
        new DateTime(1970, 1, 1, 0, 0, 0, DateTimeKind.Utc);

    int startTimestamp = (int)intervalStart.TotalSeconds; // START_TIME
    int endTimestamp = (int)intervalEnd.TotalSeconds; // END_TIME

    if (startTimestamp > endTimestamp)
        return "Error!";

    // Replace variables in the policy statement.
    strPolicy = strPolicy.Replace("RESOURCE", resourceUrl);
    strPolicy = strPolicy.Replace("START_TIME", startTimestamp.ToString());
    strPolicy = strPolicy.Replace("END_TIME", endTimestamp.ToString());
    strPolicy = strPolicy.Replace("IP_ADDRESS", ipAddress);
    strPolicy = strPolicy.Replace("EXPIRES", endTimestamp.ToString());
    return strPolicy;
}
}

public static TimeSpan GetDuration(string units, string numUnits)
{
    TimeSpan timeSpanInterval = new TimeSpan();
    switch (units)
    {
        case "seconds":
            timeSpanInterval = new TimeSpan(0, 0, 0, int.Parse(numUnits));
            break;
        case "minutes":
            timeSpanInterval = new TimeSpan(0, 0, int.Parse(numUnits), 0);
            break;
        case "hours":
            timeSpanInterval = new TimeSpan(0, int.Parse(numUnits), 0, 0);
            break;
        case "days":
            timeSpanInterval = new TimeSpan(int.Parse(numUnits), 0, 0, 0);
            break;
        default:

```

```
        Console.WriteLine("Invalid time units;" +
            "use seconds, minutes, hours, or days");
        break;
    }
    return timeSpanInterval;
}

private static TimeSpan GetDurationByUnits(string durationUnits,
    string startIntervalFromNow)
{
    switch (durationUnits)
    {
        case "seconds":
            return new TimeSpan(0, 0, int.Parse(startIntervalFromNow));
        case "minutes":
            return new TimeSpan(0, int.Parse(startIntervalFromNow), 0);
        case "hours":
            return new TimeSpan(int.Parse(startIntervalFromNow), 0, 0);
        case "days":
            return new TimeSpan(int.Parse(startIntervalFromNow), 0, 0, 0);
        default:
            return new TimeSpan(0, 0, 0, 0);
    }
}

public static string CopyExpirationTimeFromPolicy(string policyStatement)
{
    int startExpiration = policyStatement.IndexOf("EpochTime");
    string strExpirationRough = policyStatement.Substring(startExpiration +
        "EpochTime".Length);
    char[] digits = { '0', '1', '2', '3', '4', '5', '6', '7', '8', '9' };

    List<char> listDigits = new List<char>(digits);
    StringBuilder buildExpiration = new StringBuilder(20);

    foreach (char c in strExpirationRough)
    {
        if (listDigits.Contains(c))
            buildExpiration.Append(c);
    }
    return buildExpiration.ToString();
}
```

Lihat juga

- [Buat tanda tangan URL menggunakan Perl](#)
- [Buat tanda tangan URL menggunakan PHP](#)
- [Buat tanda tangan URL menggunakan Java](#)

Buat tanda tangan URL menggunakan Java

Selain contoh kode berikut, Anda dapat menggunakan [kelas CloudFrontUrlSigner utilitas di AWS SDK untuk Java \(versi 1\)](#) untuk membuat [URL yang CloudFront ditandatangani](#).

Untuk contoh selengkapnya, lihat [Membuat URL dan cookie yang ditandatangani menggunakan AWS SDK](#) di Perpustakaan Kode Contoh Kode AWS SDK.

Catatan

- Membuat URL yang ditandatangani hanyalah salah satu bagian dari proses [penyajian konten pribadi CloudFront](#). Untuk informasi selengkapnya tentang seluruh proses, lihat [Gunakan URL yang ditandatangani](#).
- Dalam `Signature.getInstance` panggilan, perhatikan yang `SHA1withRSA` dapat diganti dengan `SHA256withRSA`.

Example Kebijakan Java dan metode enkripsi tanda tangan

```
package org.example;

import java.time.Instant;
import java.time.temporal.ChronoUnit;
import software.amazon.awssdk.services.cloudfront.CloudFrontUtilities;
import software.amazon.awssdk.services.cloudfront.model.CannedSignerRequest;
import software.amazon.awssdk.services.cloudfront.url.SignedUrl;

public class Main {

    public static void main(String[] args) throws Exception {
        CloudFrontUtilities cloudFrontUtilities = CloudFrontUtilities.create();
        Instant expirationDate = Instant.now().plus(7, ChronoUnit.DAYS);
        String resourceUrl = "https://a1b2c3d4e5f6g7.cloudfront.net";
```

```

String keyPairId = "K1UA3WV15I7JSD";
CannedSignerRequest cannedRequest = CannedSignerRequest.builder()
    .resourceUrl(resourceUrl)
    .privateKey(new java.io.File("/path/to/private_key.pem").toPath())
    .keyPairId(keyPairId)
    .expirationDate(expirationDate)
    .build();

SignedUrl signedUrl =
cloudFrontUtilities.getSignedUrlWithCannedPolicy(cannedRequest);
String url = signedUrl.url();
System.out.println(url);

}
}

```

Example Contoh Penandatanganan Kebijakan Kalengan dengan SHA256 di Jawa

```

package org.example;

import java.io.File;
import java.nio.file.Files;
import java.security.KeyFactory;
import java.security.PrivateKey;
import java.security.Signature;
import java.security.spec.PKCS8EncodedKeySpec;
import java.time.Instant;
import java.time.temporal.ChronoUnit;
import java.util.Base64;

public class Main {

    public static void main(String[] args) throws Exception {
        String resourceUrl = "https://a1b2c3d4e5f6g7.cloudfront.net/myfile.html";
        String keyPairId = "K1UA3WV15I7JSD";
        Instant expiration = Instant.now().plus(7, ChronoUnit.DAYS);
        PrivateKey privateKey = loadPrivateKey("/path/to/private_key.der");

        System.out.println(createSignedUrl(resourceUrl, keyPairId, privateKey,
expiration, "SHA1"));
        System.out.println(createSignedUrl(resourceUrl, keyPairId, privateKey,
expiration, "SHA256"));
    }
}

```

```

static String createSignedUrl(String resourceUrl, String keyPairId,
                             PrivateKey privateKey, Instant expiration,
                             String hashAlgorithm) throws Exception {
    long epochSeconds = expiration.getEpochSecond();

    String policy = "{\"Statement\":[{\"Resource\":\"\" + resourceUrl
        + "\",\"Condition\":{\"DateLessThan\":{\"AWS:EpochTime\":\" +
epochSeconds + \"}}}}]\"";

    String jcaAlgorithm = hashAlgorithm.equals("SHA256") ? "SHA256withRSA" :
"SHA1withRSA";

    Signature sig = Signature.getInstance(jcaAlgorithm);
    sig.initSign(privateKey);
    sig.update(policy.getBytes("UTF-8"));
    String signature = base64UrlEncode(sig.sign());

    String url = resourceUrl
        + (resourceUrl.contains("?") ? "&" : "?")
        + "Expires=" + epochSeconds
        + "&Signature=" + signature
        + "&Key-Pair-Id=" + keyPairId;

    if (hashAlgorithm.equals("SHA256")) {
        url += "&Hash-Algorithm=SHA256";
    }

    return url;
}

static String base64UrlEncode(byte[] bytes) {
    return Base64.getEncoder().encodeToString(bytes)
        .replace('+', '-')
        .replace('=', '_')
        .replace('/', '~');
}

static PrivateKey loadPrivateKey(String path) throws Exception {
    byte[] keyBytes = Files.readAllBytes(new File(path).toPath());
    return KeyFactory.getInstance("RSA")
        .generatePrivate(new PKCS8EncodedKeySpec(keyBytes));
}
}

```

Lihat juga:

- [Buat tanda tangan URL menggunakan Perl](#)
- [Buat tanda tangan URL menggunakan PHP](#)
- [Buat tanda tangan URL menggunakan C# dan .NET Framework](#)

Membatasi akses ke AWS asal

Anda dapat mengonfigurasi CloudFront dan beberapa AWS asal dengan cara yang memberikan manfaat berikut:

- Membatasi akses ke AWS asal sehingga tidak dapat diakses publik.
- Memastikan bahwa pemirsa (pengguna) dapat mengakses konten di AWS asal hanya melalui CloudFront distribusi yang ditentukan. Ini mencegah pemirsa mengakses konten langsung dari asal, atau melalui distribusi yang tidak diinginkan CloudFront .

Untuk melakukan ini, konfigurasi CloudFront untuk mengirim permintaan yang diautentikasi ke AWS asal Anda, dan konfigurasi AWS asal untuk hanya mengizinkan akses ke permintaan yang diautentikasi dari CloudFront. Untuk informasi selengkapnya, lihat topik berikut untuk jenis AWS asal yang kompatibel.

Topik

- [Batasi akses ke asal AWS Elemental MediaPackage v2](#)
- [Membatasi akses ke AWS Elemental MediaStore asal](#)
- [Batasi akses ke asal URL AWS Lambda fungsi](#)
- [Batasi akses ke asal Amazon S3](#)
- [Batasi akses dengan asal VPC](#)

Batasi akses ke asal AWS Elemental MediaPackage v2

CloudFront menyediakan kontrol akses asal (OAC) untuk membatasi akses ke asal MediaPackage v2.

 Note

CloudFront OAC hanya mendukung MediaPackage v2. MediaPackage v1 tidak didukung.

Topik

- [Membuat OAC baru](#)
- [Pengaturan lanjutan untuk kontrol akses asal](#)

Membuat OAC baru

Selesaikan langkah-langkah yang dijelaskan dalam topik berikut untuk menyiapkan OAC baru. CloudFront

Topik

- [Prasyarat](#)
- [Berikan CloudFront izin untuk mengakses asal MediaPackage v2](#)
- [Menciptakan OAC](#)

Prasyarat

Sebelum Anda membuat dan mengatur OAC, Anda harus memiliki CloudFront distribusi dengan asal MediaPackage v2. Untuk informasi selengkapnya, lihat [Gunakan MediaStore wadah atau MediaPackage saluran](#).

Berikan CloudFront izin untuk mengakses asal MediaPackage v2

Sebelum Anda membuat OAC atau mengaturnya dalam CloudFront distribusi, pastikan bahwa CloudFront memiliki izin untuk mengakses asal MediaPackage v2. Lakukan ini setelah Anda membuat CloudFront distribusi, tetapi sebelum Anda menambahkan OAC ke asal MediaPackage v2 dalam konfigurasi distribusi.

Gunakan kebijakan IAM untuk mengizinkan prinsipal CloudFront layanan (`ccloudfront.amazonaws.com`) mengakses asal. `ConditionElement` dalam kebijakan memungkinkan CloudFront untuk mengakses asal MediaPackage v2 hanya jika permintaan atas nama CloudFront distribusi yang berisi asal MediaPackage v2. Ini adalah distribusi dengan asal MediaPackage v2 yang ingin Anda tambahkan OAC.

Example: Kebijakan IAM yang memungkinkan akses hanya-baca untuk CloudFront distribusi dengan OAC diaktifkan

Kebijakan berikut memungkinkan akses CloudFront distribusi (**E1PDK09ESKHJWT**) ke asal MediaPackage v2. Asal adalah ARN yang ditentukan untuk elemen Resource

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCloudFrontServicePrincipal",
      "Effect": "Allow",
      "Principal": {"Service": "cloudfront.amazonaws.com"},
      "Action": "mediapackagev2:GetObject",
      "Resource": "arn:aws:mediapackagev2:us-east-1:123456789012:channelGroup/channel-group-name/channel/channel-name/originEndpoint/origin_endpoint_name",
      "Condition": {
        "StringEquals": {"AWS:SourceArn": "arn:aws:cloudfront::123456789012:distribution/E1PDK09ESKHJWT"}
      }
    }
  ]
}
```

Catatan

- Jika Anda mengaktifkan fitur MQAR dan kontrol akses asal (OAC), tambahkan `mediapackagev2:GetHeadObject` tindakan ke kebijakan IAM. MQAR memerlukan izin ini untuk mengirim HEAD permintaan ke asal MediaPackage v2. Untuk informasi lebih lanjut tentang MQAR, lihat [Ketahanan sadar kualitas media](#)
- Jika Anda membuat distribusi yang tidak memiliki izin ke asal MediaPackage v2 Anda, Anda dapat memilih Salin kebijakan dari CloudFront konsol dan kemudian memilih Perbarui izin titik akhir. Anda kemudian dapat melampirkan izin yang disalin ke titik akhir. Untuk informasi selengkapnya, lihat [bidang kebijakan titik akhir](#) di Panduan AWS Elemental MediaPackage Pengguna.

Menciptakan OAC

Untuk membuat OAC, Anda dapat menggunakan Konsol Manajemen AWS CloudFormation, AWS CLI, atau CloudFront API.

Console

Untuk membuat OAC

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Akses asal.
3. Pilih Buat pengaturan kontrol.
4. Pada formulir Create new OAC, lakukan hal berikut:
 - a. Masukkan Nama dan (opsional) Deskripsi untuk OAC.
 - b. Untuk perilaku Penandatanganan, sebaiknya Anda meninggalkan setelan default (Permintaan tanda tangan (disarankan)). Untuk informasi selengkapnya, lihat [the section called “Pengaturan lanjutan untuk kontrol akses asal”](#).
5. Untuk tipe Origin, pilih MediaPackage V2.
6. Pilih Buat.

Tip

Setelah Anda membuat OAC, catat Nama. Anda membutuhkan ini dalam prosedur berikut.

Untuk menambahkan OAC ke asal MediaPackage v2 dalam distribusi

1. Buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Pilih distribusi dengan asal MediaPackage V2 yang ingin Anda tambahkan OAC, lalu pilih tab Origins.
3. Pilih asal MediaPackage v2 yang ingin Anda tambahkan OAC, lalu pilih Edit.
4. Pilih HTTPS hanya untuk Protokol asal Anda.
5. Dari menu tarik-turun kontrol akses Origin, pilih nama OAC yang ingin Anda gunakan.
6. Pilih Simpan perubahan.

Distribusi mulai menyebar ke semua lokasi CloudFront tepi. Ketika lokasi tepi menerima konfigurasi baru, ia menandatangani semua permintaan yang dikirim ke asal MediaPackage v2.

CloudFormation

Untuk membuat OAC dengan CloudFormation, gunakan jenis `AWS::CloudFront::OriginAccessControl` sumber daya. Contoh berikut menunjukkan sintaks CloudFormation template, dalam format YAMAL, untuk membuat OAC.

```
Type: AWS::CloudFront::OriginAccessControl
Properties:
  OriginAccessControlConfig:
    Description: An optional description for the origin access control
    Name: ExampleOAC
    OriginAccessControlOriginType: mediapackagev2
    SigningBehavior: always
    SigningProtocol: sigv4
```

Untuk informasi selengkapnya, lihat [AWS::CloudFront::OriginAccessControl](#) di Panduan AWS CloudFormation Pengguna.

CLI

Untuk membuat kontrol akses asal dengan AWS Command Line Interface (AWS CLI), gunakan `aws cloudfront create-origin-access-control` perintah. Anda dapat menggunakan file input untuk memberikan parameter input untuk perintah, daripada menentukan setiap parameter individu sebagai input baris perintah.

Untuk membuat kontrol akses asal (CLI dengan file input)

1. Gunakan perintah berikut untuk membuat file yang diberi nama `origin-access-control.yaml`. File ini berisi semua parameter input untuk `create-origin-access-control` perintah.

```
aws cloudfront create-origin-access-control --generate-cli-skeleton yaml-input >
origin-access-control.yaml
```

2. Buka `origin-access-control.yaml` file yang baru saja Anda buat. Edit file untuk menambahkan nama untuk OAC, deskripsi (opsional), dan ubah `SigningBehavior` ke `always`. Kemudian simpan filenya.

Untuk informasi tentang pengaturan OAC lainnya, lihat [the section called “Pengaturan lanjutan untuk kontrol akses asal”](#).

- Gunakan perintah berikut untuk membuat kontrol akses asal menggunakan parameter input dari `origin-access-control.yaml` file.

```
aws cloudfront create-origin-access-control --cli-input-yaml file://origin-access-control.yaml
```

Catat Id nilai dalam output perintah. Anda membutuhkannya untuk menambahkan OAC ke asal MediaPackage v2 dalam CloudFront distribusi.

Untuk melampirkan OAC ke asal MediaPackage v2 dalam distribusi yang ada (CLI dengan file input)

- Gunakan perintah berikut untuk menyimpan konfigurasi distribusi untuk CloudFront distribusi yang ingin Anda tambahkan OAC. Distribusi harus memiliki asal MediaPackage v2.

```
aws cloudfront get-distribution-config --id <CloudFront distribution ID> --output yaml > dist-config.yaml
```

- Buka file yang diberi nama `dist-config.yaml` yang baru saja Anda buat. Edit file akan membuat perubahan berikut:
 - Di `Origins` objek, tambahkan ID OAC ke bidang yang diberi `OriginAccessControlId` nama.
 - Hapus nilai dari bidang yang diberi nama `OriginAccessIdentity`, jika ada.
 - Ubah nama `ETag` bidang menjadi `IfMatch`, tetapi jangan ubah nilai bidang.

Simpan file setelah selesai.

- Gunakan perintah berikut untuk memperbarui distribusi untuk menggunakan kontrol akses asal.

```
aws cloudfront update-distribution --id <CloudFront distribution ID> --cli-input-yaml file://dist-config.yaml
```

Distribusi mulai menyebar ke semua lokasi CloudFront tepi. Ketika lokasi tepi menerima konfigurasi baru, ia menandatangani semua permintaan yang dikirim ke asal MediaPackage v2.

API

Untuk membuat OAC dengan CloudFront API, gunakan [CreateOriginAccessControl](#). Untuk informasi selengkapnya tentang bidang yang Anda tentukan dalam panggilan API ini, lihat dokumentasi referensi API untuk AWS SDK atau klien API lainnya.

Setelah membuat OAC, Anda dapat melampirkannya ke asal MediaPackage v2 dalam distribusi, menggunakan salah satu panggilan API berikut:

- Untuk melampirkannya ke distribusi yang ada, gunakan [UpdateDistribution](#).
- Untuk melampirkannya ke distribusi baru, gunakan [CreateDistribution](#).

Untuk kedua panggilan API ini, berikan ID OAC di `OriginAccessControlId` bidang, di dalam asal. Untuk informasi selengkapnya tentang bidang lain yang Anda tentukan dalam panggilan API ini, lihat [Semua referensi pengaturan distribusi](#) dan dokumentasi referensi API untuk AWS SDK atau klien API lainnya.

Pengaturan lanjutan untuk kontrol akses asal

Fitur CloudFront OAC mencakup pengaturan lanjutan yang ditujukan hanya untuk kasus penggunaan tertentu. Gunakan pengaturan yang disarankan kecuali Anda memiliki kebutuhan khusus untuk pengaturan lanjutan.

OAC berisi setelan bernama Perilaku penandatanganan (di konsol), atau `SigningBehavior` (di API, CLI, CloudFormation dan). Pengaturan ini menyediakan opsi berikut:

Selalu tandatangi permintaan asal (pengaturan yang disarankan)

Sebaiknya gunakan pengaturan ini, bernama `Permintaan tanda` (disarankan) di konsol, atau `always` di API, CLI, dan CloudFormation. Dengan pengaturan ini, CloudFront selalu tandatangi semua permintaan yang dikirimkan ke asal MediaPackage v2.

Jangan pernah menandatangani permintaan asal

Pengaturan ini diberi nama Jangan menandatangani permintaan di konsol, atau never di API, CLI, dan. CloudFormation Gunakan pengaturan ini untuk mematikan OAC untuk semua asal di semua distribusi yang menggunakan OAC ini. Ini dapat menghemat waktu dan tenaga dibandingkan dengan menghapus OAC dari semua asal dan distribusi yang menggunakannya, satu per satu. Dengan pengaturan ini, CloudFront tidak menandatangani permintaan apa pun yang dikirimkan ke asal MediaPackage v2.

Warning

Untuk menggunakan pengaturan ini, asal MediaPackage v2 harus dapat diakses publik. Jika Anda menggunakan setelan ini dengan asal MediaPackage v2 yang tidak dapat diakses publik, tidak CloudFront dapat mengakses asal. Asal MediaPackage v2 mengembalikan kesalahan ke CloudFront dan CloudFront meneruskan kesalahan tersebut ke pemirsa. Untuk informasi selengkapnya, lihat contoh kebijakan MediaPackage v2 untuk [Kebijakan dan Izin MediaPackage](#) di Panduan AWS Elemental MediaPackage Pengguna.

Jangan mengganti header penampil (klien) **Authorization**

Pengaturan ini diberi nama Jangan timpa header otorisasi di konsol, atau no-override di API, CLI, dan. CloudFormation Gunakan setelan ini saat Anda CloudFront ingin menandatangani permintaan asal hanya jika permintaan penampil yang sesuai tidak menyertakan Authorization header. Dengan pengaturan ini, CloudFront meneruskan Authorization header dari permintaan penampil saat ada, tetapi menandatangani permintaan asal (menambahkan tajuknya sendiri Authorization) saat permintaan penampil tidak menyertakan Authorization header.

Warning

Untuk meneruskan Authorization header dari permintaan penampil, Anda harus menambahkan Authorization header ke [kebijakan cache](#) untuk semua perilaku cache yang menggunakan asal MediaPackage v2 yang terkait dengan kontrol akses asal ini.

Membatasi akses ke AWS Elemental MediaStore asal

CloudFront menyediakan kontrol akses asal (OAC) untuk membatasi akses ke asal AWS Elemental MediaStore .

Topik

- [Buat kontrol akses asal baru](#)
- [Pengaturan lanjutan untuk kontrol akses asal](#)

Buat kontrol akses asal baru

Selesaikan langkah-langkah yang dijelaskan dalam topik berikut untuk menyiapkan kontrol akses asal baru CloudFront.

Topik

- [Prasyarat](#)
- [Berikan CloudFront izin untuk mengakses MediaStore asal](#)
- [Buat kontrol akses asal](#)

Prasyarat

Sebelum Anda membuat dan mengatur kontrol akses asal, Anda harus memiliki CloudFront distribusi dengan MediaStore asal.

Berikan CloudFront izin untuk mengakses MediaStore asal

Sebelum Anda membuat kontrol akses asal atau mengaturnya dalam CloudFront distribusi, pastikan bahwa CloudFront memiliki izin untuk mengakses MediaStore asal. Lakukan ini setelah membuat CloudFront distribusi, tetapi sebelum menambahkan OAC ke MediaStore asal dalam konfigurasi distribusi.

Gunakan kebijakan MediaStore kontainer untuk mengizinkan CloudFront service principal (`ccloudfront.amazonaws.com`) mengakses origin. Gunakan `Condition` elemen dalam kebijakan CloudFront untuk mengizinkan akses MediaStore penampung hanya jika permintaan atas nama CloudFront distribusi yang berisi MediaStore asal. Ini adalah distribusi dengan MediaStore asal yang ingin Anda tambahkan OAC.

Berikut ini adalah contoh kebijakan MediaStore kontainer yang memungkinkan CloudFront distribusi mengakses MediaStore asal.

Example MediaStore kebijakan kontainer yang memungkinkan akses hanya-baca untuk CloudFront distribusi dengan OAC diaktifkan

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCloudFrontServicePrincipalReadOnly",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudfront.amazonaws.com"
      },
      "Action": [
        "mediastore:GetObject"
      ],
      "Resource": "arn:aws:mediastore:us-east-1:111122223333:container/<container name>/*",
      "Condition": {
        "StringEquals": {
          "AWS:SourceArn":
            "arn:aws:cloudfront::111122223333:distribution/CloudFront-distribution-ID"
        },
        "Bool": {
          "aws:SecureTransport": "true"
        }
      }
    }
  ]
}
```


Example MediaStore kebijakan kontainer yang memungkinkan akses baca dan tulis untuk CloudFront distribusi dengan OAC diaktifkan

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCloudFrontServicePrincipalReadWrite",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudfront.amazonaws.com"
      },
      "Action": [
        "mediastore:GetObject",
        "mediastore:PutObject"
      ],
      "Resource": "arn:aws:mediastore:us-east-1:111122223333:container/container-name/*",
      "Condition": {
        "StringEquals": {
          "AWS:SourceArn":
            "arn:aws:cloudfront::111122223333:distribution/CloudFront-distribution-ID"
        },
        "Bool": {
          "aws:SecureTransport": "true"
        }
      }
    }
  ]
}
```

Note

Untuk mengizinkan akses tulis, Anda harus mengonfigurasi metode HTTP yang Diizinkan untuk disertakan PUT dalam pengaturan perilaku CloudFront distribusi Anda.

Buat kontrol akses asal

Untuk membuat OAC, Anda dapat menggunakan Konsol Manajemen AWS CloudFormation, AWS CLI, atau CloudFront API.

Console

Untuk membuat kontrol akses asal

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Akses asal.
3. Pilih Buat pengaturan kontrol.
4. Pada formulir pengaturan Create control, lakukan hal berikut:
 - a. Di panel Detail, masukkan Nama dan (opsional) Deskripsi untuk kontrol akses asal.
 - b. Di panel Pengaturan, kami sarankan Anda meninggalkan pengaturan default (Permintaan tanda tangan (disarankan)). Untuk informasi selengkapnya, lihat [the section called “Pengaturan lanjutan untuk kontrol akses asal”](#).
5. Pilih MediaStore dari dropdown tipe Origin.
6. Pilih Buat.

Setelah OAC dibuat, catat Nama. Anda membutuhkan ini dalam prosedur berikut.

Untuk menambahkan kontrol akses asal ke MediaStore asal dalam distribusi

1. Buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Pilih distribusi dengan MediaStore asal yang ingin Anda tambahkan OAC, lalu pilih tab Origins.
3. Pilih MediaStore asal yang ingin Anda tambahkan OAC, lalu pilih Edit.
4. Pilih HTTPS hanya untuk Protokol asal Anda.
5. Dari menu tarik-turun kontrol akses Origin, pilih OAC yang ingin Anda gunakan.
6. Pilih Simpan perubahan.

Distribusi mulai menyebar ke semua lokasi CloudFront tepi. Saat lokasi tepi menerima konfigurasi baru, ia akan menandatangani semua permintaan yang dikirimkan ke asal MediaStore bucket.

CloudFormation

Untuk membuat kontrol akses asal (OAC) dengan CloudFormation, gunakan jenis `AWS::CloudFront::OriginAccessControl` sumber daya. Contoh berikut menunjukkan sintaks CloudFormation template, dalam format YAMM, untuk membuat kontrol akses asal.

```
Type: AWS::CloudFront::OriginAccessControl
Properties:
  OriginAccessControlConfig:
    Description: An optional description for the origin access control
    Name: ExampleOAC
    OriginAccessControlOriginType: mediastore
    SigningBehavior: always
    SigningProtocol: sigv4
```

Untuk informasi selengkapnya, lihat [AWS::CloudFront:: OriginAccessControl](#) di Panduan AWS CloudFormation Pengguna.

CLI

Untuk membuat kontrol akses asal dengan AWS Command Line Interface (AWS CLI), gunakan `aws cloudfront create-origin-access-control` perintah. Anda dapat menggunakan file input untuk memberikan parameter input untuk perintah, daripada menentukan setiap parameter individu sebagai input baris perintah.

Untuk membuat kontrol akses asal (CLI dengan file input)

1. Gunakan perintah berikut untuk membuat file yang diberi nama `origin-access-control.yaml`. File ini berisi semua parameter input untuk `create-origin-access-control` perintah.

```
aws cloudfront create-origin-access-control --generate-cli-skeleton yml-input >
origin-access-control.yaml
```

2. Buka `origin-access-control.yaml` file yang baru saja Anda buat. Edit file untuk menambahkan nama untuk OAC, deskripsi (opsional), dan ubah `SigningBehavior` ke `always`. Kemudian simpan filenya.

Untuk informasi tentang pengaturan OAC lainnya, lihat [the section called “Pengaturan lanjutan untuk kontrol akses asal”](#).

- Gunakan perintah berikut untuk membuat kontrol akses asal menggunakan parameter input dari `origin-access-control.yaml` file.

```
aws cloudfront create-origin-access-control --cli-input-yaml file://origin-access-control.yaml
```

Catat Id nilai dalam output perintah. Anda membutuhkannya untuk menambahkan OAC ke MediaStore asal dalam CloudFront distribusi.

Untuk melampirkan OAC ke MediaStore asal dalam distribusi yang ada (CLI dengan file input)

- Gunakan perintah berikut untuk menyimpan konfigurasi distribusi untuk CloudFront distribusi yang ingin Anda tambahkan OAC. Distribusi harus memiliki MediaStore asal.

```
aws cloudfront get-distribution-config --id <CloudFront distribution ID> --output yaml > dist-config.yaml
```

- Buka file yang diberi nama `dist-config.yaml` yang baru saja Anda buat. Edit file akan membuat perubahan berikut:
 - Di `Origins` objek, tambahkan ID OAC ke bidang yang diberi `OriginAccessControlId` nama.
 - Hapus nilai dari bidang yang diberi nama `OriginAccessIdentity`, jika ada.
 - Ubah nama `ETag` bidang menjadi `IfMatch`, tetapi jangan ubah nilai bidang.

Simpan file setelah selesai.

- Gunakan perintah berikut untuk memperbarui distribusi untuk menggunakan kontrol akses asal.

```
aws cloudfront update-distribution --id <CloudFront distribution ID> --cli-input-yaml file://dist-config.yaml
```

Distribusi mulai menyebar ke semua lokasi CloudFront tepi. Ketika lokasi tepi menerima konfigurasi baru, ia menandatangani semua permintaan yang dikirim ke MediaStore asal.

API

Untuk membuat kontrol akses asal dengan CloudFront API, gunakan [CreateOriginAccessControl](#). Untuk informasi selengkapnya tentang bidang yang Anda tentukan dalam panggilan API ini, lihat dokumentasi referensi API untuk AWS SDK atau klien API lainnya.

Setelah membuat kontrol akses asal, Anda dapat melampirkannya ke MediaStore asal dalam distribusi, menggunakan salah satu panggilan API berikut:

- Untuk melampirkannya ke distribusi yang ada, gunakan [UpdateDistribution](#).
- Untuk melampirkannya ke distribusi baru, gunakan [CreateDistribution](#).

Untuk kedua panggilan API ini, berikan ID kontrol akses asal di `OriginAccessControlId` bidang, di dalam asal. Untuk informasi selengkapnya tentang bidang lain yang Anda tentukan dalam panggilan API ini, lihat [Semua referensi pengaturan distribusi](#) dan dokumentasi referensi API untuk AWS SDK atau klien API lainnya.

Pengaturan lanjutan untuk kontrol akses asal

Fitur kontrol akses CloudFront asal mencakup pengaturan lanjutan yang ditujukan hanya untuk kasus penggunaan tertentu. Gunakan pengaturan yang disarankan kecuali Anda memiliki kebutuhan khusus untuk pengaturan lanjutan.

Kontrol akses asal berisi setelan bernama Perilaku penandatanganan (di konsol), atau `SigningBehavior` (di API, CLI, dan CloudFormation). Pengaturan ini menyediakan opsi berikut:

Selalu tandatangani permintaan asal (pengaturan yang disarankan)

Sebaiknya gunakan pengaturan ini, bernama Permintaan tanda (disarankan) di konsol, atau `always` di API, CLI, dan CloudFormation. Dengan pengaturan ini, CloudFront selalu tandatangani semua permintaan yang dikirimkan ke MediaStore asal.

Jangan pernah menandatangani permintaan asal

Pengaturan ini diberi nama Jangan menandatangani permintaan di konsol, atau `never` di API, CLI, dan CloudFormation. Gunakan pengaturan ini untuk menonaktifkan kontrol akses asal untuk semua asal di semua distribusi yang menggunakan kontrol akses asal ini. Ini dapat menghemat waktu dan tenaga dibandingkan dengan menghapus kontrol akses asal dari semua asal dan distribusi yang menggunakannya, satu per satu. Dengan pengaturan ini, CloudFront tidak menandatangani permintaan apa pun yang dikirimkan ke MediaStore asal.

 Warning

Untuk menggunakan pengaturan ini, MediaStore asal harus dapat diakses publik. Jika Anda menggunakan setelan ini dengan MediaStore asal yang tidak dapat diakses publik, CloudFront tidak dapat mengakses asal. MediaStore Asal mengembalikan kesalahan ke CloudFront dan CloudFront meneruskan kesalahan tersebut ke pemirsa. Untuk informasi selengkapnya, lihat contoh kebijakan MediaStore kontainer untuk [akses baca Publik melalui HTTPS](#).

Jangan mengganti header penampil (klien) **Authorization**

Pengaturan ini diberi nama Jangan timpa header otorisasi di konsol, atau no-override di API, CLI, dan. CloudFormation Gunakan setelan ini saat Anda CloudFront ingin menandatangani permintaan asal hanya jika permintaan penampil yang sesuai tidak menyertakan Authorization header. Dengan pengaturan ini, CloudFront meneruskan Authorization header dari permintaan penampil saat ada, tetapi menandatangani permintaan asal (menambahkan tajuknya sendiri Authorization) saat permintaan penampil tidak menyertakan Authorization header.

 Warning

Untuk meneruskan Authorization header dari permintaan penampil, Anda harus menambahkan Authorization header ke [kebijakan cache](#) untuk semua perilaku cache yang menggunakan MediaStore asal yang terkait dengan kontrol akses asal ini.

Batasi akses ke asal URL AWS Lambda fungsi

CloudFront menyediakan kontrol akses asal (OAC) untuk membatasi akses ke asal URL fungsi Lambda.

Topik

- [Buat OAC baru](#)
- [Pengaturan lanjutan untuk kontrol akses asal](#)
- [Contoh kode template](#)

Buat OAC baru

Selesaikan langkah-langkah yang dijelaskan dalam topik berikut untuk menyiapkan OAC baru. CloudFront

Important

Jika Anda menggunakan PUT atau POST metode dengan URL fungsi Lambda Anda, pengguna Anda harus menghitung isi dan menyertakan nilai hash payload dari badan permintaan di `x-amz-content-sha256` header saat mengirim permintaan ke. SHA256 CloudFront Lambda tidak mendukung muatan yang tidak ditandatangani.

Topik

- [Prasyarat](#)
- [Berikan CloudFront izin untuk mengakses URL fungsi Lambda](#)
- [Buat OAC](#)

Prasyarat

Sebelum Anda membuat dan mengatur OAC, Anda harus memiliki CloudFront distribusi dengan URL fungsi Lambda sebagai asal. Untuk menggunakan OAC, Anda harus menentukan `AWS_IAM` sebagai nilai untuk `AuthType` parameter. Untuk informasi selengkapnya, lihat [Gunakan URL fungsi Lambda](#).

Berikan CloudFront izin untuk mengakses URL fungsi Lambda

Sebelum Anda membuat OAC atau mengaturnya dalam CloudFront distribusi, pastikan bahwa CloudFront memiliki izin untuk mengakses URL fungsi Lambda. Lakukan ini setelah Anda membuat CloudFront distribusi, tetapi sebelum Anda menambahkan OAC ke URL fungsi Lambda dalam konfigurasi distribusi.

Note

Untuk memperbarui kebijakan IAM untuk URL fungsi Lambda, Anda harus menggunakan () AWS Command Line Interface .AWS CLI Mengedit kebijakan IAM di konsol Lambda tidak didukung saat ini.

AWS CLI Perintah berikut memberikan akses CloudFront service principal (`cloudfront.amazonaws.com`) ke URL fungsi Lambda Anda. ConditionElement dalam kebijakan memungkinkan CloudFront untuk mengakses Lambda hanya jika permintaan atas nama CloudFront distribusi yang berisi URL fungsi Lambda. Ini adalah distribusi dengan asal URL fungsi Lambda yang ingin Anda tambahkan OAC.

Example: AWS CLI perintah untuk memperbarui kebijakan untuk mengizinkan akses hanya-baca untuk CloudFront distribusi dengan OAC diaktifkan

AWS CLI Perintah berikut memungkinkan CloudFront distribusi (*E1PDK09ESKHJWT*) akses ke Lambda *FUNCTION_URL_NAME* Anda.

```
aws lambda add-permission \
--statement-id "AllowCloudFrontServicePrincipal" \
--action "lambda:InvokeFunctionUrl" \
--principal "cloudfront.amazonaws.com" \
--source-arn "arn:aws:cloudfront::123456789012:distribution/E1PDK09ESKHJWT" \
--function-name FUNCTION_URL_NAME
```

```
aws lambda add-permission \
--statement-id "AllowCloudFrontServicePrincipalInvokeFunction" \
--action "lambda:InvokeFunction" \
--principal "cloudfront.amazonaws.com" \
--source-arn "arn:aws:cloudfront::123456789012:distribution/E1PDK09ESKHJWT" \
--function-name FUNCTION_URL_NAME
```

Note

Jika Anda membuat distribusi dan tidak memiliki izin ke URL fungsi Lambda Anda, Anda dapat memilih Salin perintah CLI dari CloudFront konsol, lalu masukkan perintah ini dari terminal baris perintah Anda. Untuk informasi selengkapnya, lihat [Memberikan akses fungsi Layanan AWS](#) di Panduan AWS Lambda Pengembang.

Buat OAC

Untuk membuat OAC, Anda dapat menggunakan Konsol Manajemen AWS CloudFormation, AWS CLI, atau CloudFront API.

Console

Untuk membuat OAC

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Akses asal.
3. Pilih Buat pengaturan kontrol.
4. Pada formulir Create new OAC, lakukan hal berikut:
 - a. Masukkan Nama dan (opsional) Deskripsi untuk OAC.
 - b. Untuk perilaku Penandatanganan, sebaiknya Anda meninggalkan pengaturan default (Permintaan tanda tangan (disarankan)). Untuk informasi selengkapnya, lihat [the section called “Pengaturan lanjutan untuk kontrol akses asal”](#).
5. Untuk tipe Origin, pilih Lambda.
6. Pilih Buat.

Tip

Setelah Anda membuat OAC, catat Nama. Anda membutuhkan ini dalam prosedur berikut.

Untuk menambahkan kontrol akses asal ke URL fungsi Lambda dalam distribusi

1. Buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Pilih distribusi dengan URL fungsi Lambda yang ingin Anda tambahkan OAC, lalu pilih tab Origins.
3. Pilih URL fungsi Lambda yang ingin Anda tambahkan OAC, lalu pilih Edit.
4. Pilih HTTPS hanya untuk Protokol asal Anda.
5. Dari menu tarik-turun kontrol akses Origin, pilih nama OAC yang ingin Anda gunakan.
6. Pilih Simpan perubahan.

Distribusi mulai menyebar ke semua lokasi CloudFront tepi. Ketika lokasi tepi menerima konfigurasi baru, ia menandatangani semua permintaan yang dikirim ke URL fungsi Lambda.

CloudFormation

Untuk membuat OAC dengan CloudFormation, gunakan jenis `AWS::CloudFront::OriginAccessControl` sumber daya. Contoh berikut menunjukkan sintaks CloudFormation template, dalam format YAMAL, untuk membuat OAC.

```
Type: AWS::CloudFront::OriginAccessControl
Properties:
  OriginAccessControlConfig:
    Description: An optional description for the origin access control
    Name: ExampleOAC
    OriginAccessControlOriginType: lambda
    SigningBehavior: always
    SigningProtocol: sigv4
```

Untuk informasi selengkapnya, lihat [AWS::CloudFront::OriginAccessControl](#) di Panduan AWS CloudFormation Pengguna.

CLI

Untuk membuat kontrol akses asal dengan AWS Command Line Interface (AWS CLI), gunakan `aws cloudfront create-origin-access-control` perintah. Anda dapat menggunakan file input untuk memberikan parameter input untuk perintah, daripada menentukan setiap parameter individu sebagai input baris perintah.

Untuk membuat kontrol akses asal (CLI dengan file input)

1. Gunakan perintah berikut untuk membuat file yang diberi nama `origin-access-control.yaml`. File ini berisi semua parameter input untuk `create-origin-access-control` perintah.

```
aws cloudfront create-origin-access-control --generate-cli-skeleton yml-input >
origin-access-control.yaml
```

2. Buka `origin-access-control.yaml` file yang baru saja Anda buat. Edit file untuk menambahkan nama untuk OAC, deskripsi (opsional), dan ubah `SigningBehavior` ke `always`. Kemudian simpan filenya.

Untuk informasi tentang pengaturan OAC lainnya, lihat [the section called “Pengaturan lanjutan untuk kontrol akses asal”](#).

- Gunakan perintah berikut untuk membuat kontrol akses asal menggunakan parameter input dari `origin-access-control.yaml` file.

```
aws cloudfront create-origin-access-control --cli-input-yaml file://origin-access-control.yaml
```

Catat Id nilai dalam output perintah. Anda membutuhkannya untuk menambahkan OAC ke URL fungsi Lambda dalam CloudFront distribusi.

Untuk melampirkan OAC ke URL fungsi Lambda dalam distribusi yang ada (CLI dengan file input)

- Gunakan perintah berikut untuk menyimpan konfigurasi distribusi untuk CloudFront distribusi yang ingin Anda tambahkan OAC. Distribusi harus memiliki URL fungsi Lambda sebagai asal.

```
aws cloudfront get-distribution-config --id <CloudFront distribution ID> --output yaml > dist-config.yaml
```

- Buka file yang diberi nama `dist-config.yaml` yang baru saja Anda buat. Edit file akan membuat perubahan berikut:
 - Di `Origins` objek, tambahkan ID OAC ke bidang yang diberi `OriginAccessControlId` nama.
 - Hapus nilai dari bidang yang diberi nama `OriginAccessIdentity`, jika ada.
 - Ubah nama `ETag` bidang menjadi `IfMatch`, tetapi jangan ubah nilai bidang.

Simpan file setelah selesai.

- Gunakan perintah berikut untuk memperbarui distribusi untuk menggunakan kontrol akses asal.

```
aws cloudfront update-distribution --id <CloudFront distribution ID> --cli-input-yaml file://dist-config.yaml
```

Distribusi mulai menyebar ke semua lokasi CloudFront tepi. Ketika lokasi tepi menerima konfigurasi baru, ia menandatangani semua permintaan yang dikirim ke URL fungsi Lambda.

API

Untuk membuat OAC dengan CloudFront API, gunakan [CreateOriginAccessControl](#). Untuk informasi selengkapnya tentang bidang yang Anda tentukan dalam panggilan API ini, lihat dokumentasi referensi API untuk AWS SDK atau klien API lainnya.

Setelah membuat OAC, Anda dapat melampirkannya ke URL fungsi Lambda dalam distribusi, menggunakan salah satu panggilan API berikut:

- Untuk melampirkannya ke distribusi yang ada, gunakan [UpdateDistribution](#).
- Untuk melampirkannya ke distribusi baru, gunakan [CreateDistribution](#).

Untuk kedua panggilan API ini, berikan ID OAC di `OriginAccessControlId` bidang, di dalam asal. Untuk informasi selengkapnya tentang bidang lain yang Anda tentukan dalam panggilan API ini, lihat dan dokumentasi referensi API untuk AWS SDK atau klien API lainnya.

Pengaturan lanjutan untuk kontrol akses asal

Fitur CloudFront OAC mencakup pengaturan lanjutan yang ditujukan hanya untuk kasus penggunaan tertentu. Gunakan pengaturan yang disarankan kecuali Anda memiliki kebutuhan khusus untuk pengaturan lanjutan.

OAC berisi setelan bernama Perilaku penandatanganan (di konsol), atau `SigningBehavior` (di API, CLI, CloudFormation dan). Pengaturan ini menyediakan opsi berikut:

Selalu tandatangani permintaan asal (pengaturan yang disarankan)

Sebaiknya gunakan pengaturan ini, bernama `Permintaan tanda` (disarankan) di konsol, atau `always` di API, CLI, dan. CloudFormation Dengan pengaturan ini, CloudFront selalu tandatangani semua permintaan yang dikirimkan ke URL fungsi Lambda.

Jangan pernah menandatangani permintaan asal

Pengaturan ini diberi nama `Jangan menandatangani permintaan` di konsol, atau `never` di API, CLI, dan. CloudFormation Gunakan pengaturan ini untuk mematikan OAC untuk semua asal di semua distribusi yang menggunakan OAC ini. Ini dapat menghemat waktu dan tenaga dibandingkan dengan menghapus OAC dari semua asal dan distribusi yang menggunakannya, satu per satu. Dengan pengaturan ini, CloudFront tidak menandatangani permintaan apa pun yang dikirimkan ke URL fungsi Lambda.

 Warning

Untuk menggunakan pengaturan ini, URL fungsi Lambda harus dapat diakses publik. Jika Anda menggunakan setelan ini dengan URL fungsi Lambda yang tidak dapat diakses publik, tidak CloudFront dapat mengakses asal. URL fungsi Lambda mengembalikan kesalahan ke CloudFront dan CloudFront meneruskan kesalahan tersebut ke pemirsa. Untuk informasi selengkapnya, lihat [Model keamanan dan autentikasi untuk URLs fungsi Lambda](#) di Panduan Pengguna AWS Lambda .

Jangan mengganti header penampil (klien) **Authorization**

Pengaturan ini diberi nama Jangan timpa header otorisasi di konsol, atau no-override di API, CLI, dan. CloudFormation Gunakan setelan ini saat Anda CloudFront ingin menandatangani permintaan asal hanya jika permintaan penampil yang sesuai tidak menyertakan `Authorization` header. Dengan pengaturan ini, CloudFront meneruskan `Authorization` header dari permintaan penampil saat ada, tetapi menandatangani permintaan asal (menambahkan tajuknya sendiri `Authorization`) saat permintaan penampil tidak menyertakan `Authorization` header.

 Warning

- Jika Anda menggunakan setelan ini, Anda harus menentukan penandatanganan Tanda Tangan Versi 4 untuk URL fungsi Lambda, bukan nama CloudFront distribusi atau CNAME. Saat CloudFront meneruskan `Authorization` header dari permintaan penampil ke URL fungsi Lambda, Lambda akan memvalidasi tanda tangan terhadap host domain URL Lambda. Jika tanda tangan tidak didasarkan pada domain URL Lambda, host dalam tanda tangan tidak akan cocok dengan host yang digunakan oleh asal URL Lambda. Ini berarti permintaan akan gagal, menghasilkan kesalahan validasi tanda tangan.
- Untuk meneruskan `Authorization` header dari permintaan penampil, Anda harus menambahkan `Authorization` header ke [kebijakan cache](#) untuk semua perilaku cache yang menggunakan fungsi Lambda yang URLs terkait dengan kontrol akses asal ini.

Contoh kode template

Jika CloudFront asal Anda adalah URL fungsi Lambda yang terkait dengan OAC, Anda dapat menggunakan skrip Python berikut untuk mengunggah file ke fungsi Lambda dengan metode tersebut. POST

Kode ini mengasumsikan bahwa Anda mengonfigurasi OAC dengan perilaku penandatanganan default yang disetel ke Selalu menandatangani permintaan asal dan Anda tidak memilih setelan header Jangan timpa otorisasi.

Konfigurasi ini memungkinkan OAC untuk mengelola otorisasi SiGv4 dengan benar dengan Lambda dengan menggunakan nama host Lambda. Payload ditandatangani dengan menggunakan Sigv4 dari identitas IAM yang diotorisasi untuk URL fungsi Lambda, yang ditetapkan sebagai tipe. IAM_AUTH

Template menunjukkan cara menangani nilai hash payload yang ditandatangani di x-amz-content-sha256 header untuk POST permintaan dari sisi klien. Secara khusus, template ini dirancang untuk mengelola muatan data formulir. Template memungkinkan unggahan file aman ke URL fungsi Lambda CloudFront melalui, dan AWS menggunakan mekanisme otentikasi untuk memastikan bahwa hanya permintaan resmi yang dapat mengakses fungsi Lambda.

 Kode ini mencakup fungsionalitas berikut:

- Memenuhi persyaratan untuk memasukkan hash payload di header x-amz-content-sha256
- Menggunakan otentikasi SiGv4 untuk akses aman Layanan AWS
- Mendukung unggahan file dengan menggunakan data formulir multi-bagian
- Termasuk penanganan kesalahan untuk pengecualian permintaan

```
import boto3
from botocore.auth import SigV4Auth
from botocore.awsrequest import AWSRequest
import requests
import hashlib
import os

def calculate_body_hash(body):
    return hashlib.sha256(body).hexdigest()
```

```
def sign_request(request, credentials, region, service):
    sigv4 = SigV4Auth(credentials, service, region)
    sigv4.add_auth(request)

def upload_file_to_lambda(cloudfront_url, file_path, region):
    # AWS credentials
    session = boto3.Session()
    credentials = session.get_credentials()

    # Prepare the multipart form-data
    boundary = "-----boundary"

    # Read file content
    with open(file_path, 'rb') as file:
        file_content = file.read()

    # Get the filename from the path
    filename = os.path.basename(file_path)

    # Prepare the multipart body
    body = (
        f'--{boundary}\r\n'
        f'Content-Disposition: form-data; name="file"; filename="{filename}"\r\n'
        f'Content-Type: application/octet-stream\r\n\r\n'
    ).encode('utf-8')
    body += file_content
    body += f'\r\n--{boundary}--\r\n'.encode('utf-8')

    # Calculate SHA256 hash of the entire body
    body_hash = calculate_body_hash(body)

    # Prepare headers
    headers = {
        'Content-Type': f'multipart/form-data; boundary={boundary}',
        'x-amz-content-sha256': body_hash
    }

    # Create the request
    request = AWSRequest(
        method='POST',
        url=cloudfront_url,
        data=body,
```

```
        headers=headers
    )

    # Sign the request
    sign_request(request, credentials, region, 'lambda')

    # Get the signed headers
    signed_headers = dict(request.headers)

    # Print request headers before sending
    print("Request Headers:")
    for header, value in signed_headers.items():
        print(f"{header}: {value}")

    try:
        # Send POST request with signed headers
        response = requests.post(
            cloudfront_url,
            data=body,
            headers=signed_headers
        )

        # Print response status and content
        print(f"\nStatus code: {response.status_code}")
        print("Response:", response.text)

        # Print response headers
        print("\nResponse Headers:")
        for header, value in response.headers.items():
            print(f"{header}: {value}")

    except requests.exceptions.RequestException as e:
        print(f"An error occurred: {e}")

# Usage
cloudfront_url = "https://d1111111abcdef8.cloudfront.net"
file_path = r"filepath"
region = "us-east-1" # example: "us-west-2"

upload_file_to_lambda(cloudfront_url, file_path, region)
```


Batasi akses ke asal Amazon S3

CloudFront menyediakan dua cara untuk mengirim permintaan yang diautentikasi ke asal Amazon S3: kontrol akses asal (OAC) dan identitas akses asal (OAI). OAC membantu Anda mengamankan asal Anda, seperti Amazon S3.

Kami menyarankan Anda menggunakan OAC sebagai gantinya karena mendukung fitur-fitur berikut:

- Semua bucket Amazon S3 secara keseluruhan Wilayah AWS, termasuk Wilayah keikutsertaan diluncurkan setelah Desember 2022
- Enkripsi sisi [server Amazon S3 dengan \(\) AWS KMS](#) SSE-KMS
- Permintaan dinamis (PUT dan DELETE) ke Amazon S3

OAI tidak mendukung fitur-fitur ini atau memerlukan solusi tambahan dalam skenario tersebut. Jika Anda sudah menggunakan OAI dan ingin bermigrasi, lihat [the section called “Migrasi dari Origin Access Identity \(OAI\) ke Origin Access Control \(OAC\)”](#)

Catatan

- Saat menggunakan CloudFront OAC dengan asal bucket Amazon S3, Anda harus menyetel Kepemilikan Objek Amazon S3 ke pemilik Bucket yang diberlakukan, default untuk bucket Amazon S3 baru. Jika Anda memerlukan ACL, gunakan setelan pilihan pemilik Bucket untuk mempertahankan kontrol atas objek yang diunggah melalui CloudFront
- Jika asal Anda adalah bucket Amazon S3 yang dikonfigurasi sebagai [titik akhir situs web](#), Anda harus mengaturnya CloudFront sebagai asal khusus. Itu berarti Anda tidak dapat menggunakan OAC (atau OAI). OAC tidak mendukung pengalihan asal dengan menggunakan Lambda @Edge.

Topik berikut menjelaskan cara menggunakan OAC dengan asal Amazon S3.

Topik

- [the section called “Buat kontrol akses asal baru”](#)
- [the section called “Hapus distribusi dengan OAC yang terpasang pada bucket S3”](#)
- [the section called “Migrasi dari Origin Access Identity \(OAI\) ke Origin Access Control \(OAC\)”](#)

- [the section called “Pengaturan lanjutan untuk kontrol akses asal”](#)

Buat kontrol akses asal baru

Selesaikan langkah-langkah yang dijelaskan dalam topik berikut untuk menyiapkan kontrol akses asal baru CloudFront.

Topik

- [Prasyarat](#)
- [Berikan CloudFront izin untuk mengakses bucket S3](#)
- [Buat kontrol akses asal](#)

Prasyarat

Sebelum membuat dan mengatur kontrol akses asal (OAC), Anda harus memiliki CloudFront distribusi dengan asal bucket Amazon S3. Asal ini harus berupa bucket S3 biasa, bukan bucket yang dikonfigurasi sebagai titik [akhir situs web](#). Untuk informasi selengkapnya tentang menyiapkan CloudFront distribusi dengan asal bucket S3, lihat [the section called “Memulai dengan distribusi standar”](#).

Important

Saat Anda menggunakan OAC untuk mengamankan asal Amazon S3 Anda, komunikasi CloudFront antara dan Amazon S3 selalu melalui HTTPS, tetapi hanya ketika Anda memilih untuk selalu menandatangani permintaan. Anda harus memilih Permintaan tanda tangan (disarankan) di konsol atau tentukan `always` di CloudFront API, AWS CLI, atau CloudFormation.

Jika Anda memilih opsi Jangan menandatangani permintaan atau Jangan timpa header otorisasi, CloudFront gunakan protokol koneksi yang Anda tentukan dalam kebijakan berikut:

- [Kebijakan protokol penampil](#)
- [Kebijakan protokol asal](#) (hanya asal kustom)

[Misalnya, jika Anda memilih Jangan timpa header otorisasi dan ingin menggunakan HTTPS antara CloudFront dan asal Amazon S3 Anda, gunakan Redirect HTTP ke HTTPS atau HTTPS hanya untuk kebijakan protokol penampil.](#)

Berikan CloudFront izin untuk mengakses bucket S3

Sebelum Anda membuat kontrol akses asal (OAC) atau mengaturnya dalam CloudFront distribusi, pastikan bahwa CloudFront memiliki izin untuk mengakses asal bucket S3. Lakukan ini setelah membuat CloudFront distribusi, tetapi sebelum menambahkan OAC ke asal S3 dalam konfigurasi distribusi.

Gunakan [kebijakan bucket](#) S3 untuk mengizinkan CloudFront service principal (`cloudfront.amazonaws.com`) mengakses bucket. Gunakan `Condition` elemen dalam kebijakan CloudFront untuk mengizinkan akses bucket hanya jika permintaan tersebut atas nama CloudFront distribusi yang berisi asal S3. Ini adalah distribusi dengan asal S3 yang ingin Anda tambahkan OAC.

Untuk informasi tentang menambahkan atau memodifikasi kebijakan bucket, lihat [Menambahkan kebijakan bucket menggunakan konsol Amazon S3](#) di Panduan Pengguna Amazon S3.

Berikut ini adalah contoh kebijakan bucket S3 yang memungkinkan CloudFront distribusi dengan akses berkemampuan OAC ke asal S3.

Example Kebijakan bucket S3 yang memungkinkan akses hanya-baca untuk CloudFront distribusi dengan OAC diaktifkan

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCloudFrontServicePrincipalReadOnly",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudfront.amazonaws.com"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
      "Condition": {
        "StringEquals": {
          "AWS:SourceArn":
            "arn:aws:cloudfront::111122223333:distribution/<CloudFront distribution ID>"
        }
      }
    }
  ]
}
```

```

    }
  ]
}
```

Example Kebijakan bucket S3 yang memungkinkan akses baca dan tulis untuk CloudFront distribusi dengan OAC diaktifkan

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCloudFrontServicePrincipalReadWrite",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudfront.amazonaws.com"
      },
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket/*",
      "Condition": {
        "StringEquals": {
          "AWS:SourceArn":
            "arn:aws:cloudfront::111122223333:distribution/CloudFront-distribution-ID>"
        }
      }
    }
  ]
}
```

SSE-KMS

Jika objek dalam asal bucket S3 dienkripsi menggunakan [enkripsi sisi server dengan AWS Key Management Service \(SSE-KMS\)](#), Anda harus memastikan bahwa CloudFront distribusi memiliki izin untuk menggunakan kunci. AWS KMS Untuk memberikan izin CloudFront distribusi untuk menggunakan kunci KMS, tambahkan pernyataan ke kebijakan [kunci KMS](#). Untuk informasi tentang

cara mengubah kebijakan kunci, lihat [Mengubah kebijakan kunci](#) di Panduan AWS Key Management Service Pengembang.

Example Pernyataan kebijakan kunci KMS

Contoh berikut menunjukkan pernyataan AWS KMS kebijakan yang memungkinkan CloudFront distribusi dengan OAC untuk mengakses kunci KMS. SSE-KMS

```
{
  "Sid": "AllowCloudFrontServicePrincipalSSE-KMS",
  "Effect": "Allow",
  "Principal": {
    "Service": [
      "cloudfront.amazonaws.com"
    ]
  },
  "Action": [
    "kms:Decrypt",
    "kms:Encrypt",
    "kms:GenerateDataKey*"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {
      "AWS:SourceArn":
        "arn:aws:cloudfront::111122223333:distribution/<CloudFront distribution ID>"
    }
  }
}
```

Buat kontrol akses asal

Untuk membuat kontrol akses asal (OAC), Anda dapat menggunakan Konsol Manajemen AWS CloudFormation, AWS CLI, atau CloudFront API.

Console

Untuk membuat kontrol akses asal

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Akses asal.

3. Pilih Buat pengaturan kontrol.
4. Pada formulir pengaturan Create control, lakukan hal berikut:
 - a. Di panel Detail, masukkan Nama dan (opsional) Deskripsi untuk kontrol akses asal.
 - b. Di panel Pengaturan, kami sarankan Anda meninggalkan pengaturan default (Permintaan tanda tangan (disarankan)). Untuk informasi selengkapnya, lihat [the section called “Pengaturan lanjutan untuk kontrol akses asal”](#).
5. Pilih S3 dari dropdown tipe Origin.
6. Pilih Buat.

Setelah OAC dibuat, catat Nama. Anda membutuhkan ini dalam prosedur berikut.

Untuk menambahkan kontrol akses asal ke asal S3 dalam distribusi

1. Buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Pilih distribusi dengan asal S3 yang ingin Anda tambahkan OAC, lalu pilih tab Origins.
3. Pilih asal S3 yang ingin Anda tambahkan OAC, lalu pilih Edit.
4. Untuk akses Origin, pilih Pengaturan kontrol akses Origin (disarankan).
5. Dari menu tarik-turun kontrol akses Origin, pilih OAC yang ingin Anda gunakan.
6. Pilih Simpan perubahan.

Distribusi mulai menyebar ke semua lokasi CloudFront tepi. Saat lokasi edge menerima konfigurasi baru, ia akan menandatangani semua permintaan yang dikirimkan ke bucket origin S3.

CloudFormation

Untuk membuat kontrol akses asal (OAC) dengan CloudFormation, gunakan jenis `AWS::CloudFront::OriginAccessControl` sumber daya. Contoh berikut menunjukkan sintaks CloudFormation template, dalam format YAMM, untuk membuat kontrol akses asal.

```
Type: AWS::CloudFront::OriginAccessControl
Properties:
  OriginAccessControlConfig:
    Description: An optional description for the origin access control
    Name: ExampleOAC
    OriginAccessControlOriginType: s3
    SigningBehavior: always
```

```
SigningProtocol: sigv4
```

Untuk informasi selengkapnya, lihat [AWS::CloudFront:: OriginAccessControl](#) di Panduan AWS CloudFormation Pengguna.

CLI

Untuk membuat kontrol akses asal dengan AWS Command Line Interface (AWS CLI), gunakan `aws cloudfront create-origin-access-control` perintah. Anda dapat menggunakan file input untuk memberikan parameter input untuk perintah, daripada menentukan setiap parameter individu sebagai input baris perintah.

Untuk membuat kontrol akses asal (CLI dengan file input)

1. Gunakan perintah berikut untuk membuat file yang diberi nama `origin-access-control.yaml`. File ini berisi semua parameter input untuk `create-origin-access-control` perintah.

```
aws cloudfront create-origin-access-control --generate-cli-skeleton yaml-input >
origin-access-control.yaml
```

2. Buka `origin-access-control.yaml` file yang baru saja Anda buat. Edit file untuk menambahkan nama untuk OAC, deskripsi (opsional), dan ubah `SigningBehavior` ke `always`. Kemudian simpan filenya.

Untuk informasi tentang pengaturan OAC lainnya, lihat [the section called “Pengaturan lanjutan untuk kontrol akses asal”](#).

3. Gunakan perintah berikut untuk membuat kontrol akses asal menggunakan parameter input dari `origin-access-control.yaml` file.

```
aws cloudfront create-origin-access-control --cli-input-yaml file://origin-
access-control.yaml
```

Catat Id nilai dalam output perintah. Anda membutuhkannya untuk menambahkan OAC ke asal bucket S3 dalam distribusi. CloudFront

Untuk melampirkan OAC ke asal bucket S3 dalam distribusi yang ada (CLI dengan file input)

1. Gunakan perintah berikut untuk menyimpan konfigurasi distribusi untuk CloudFront distribusi yang ingin Anda tambahkan OAC. Distribusi harus memiliki asal bucket S3.

```
aws cloudfront get-distribution-config --id <CloudFront distribution ID> --  
output yaml > dist-config.yaml
```

2. Buka file yang diberi nama `dist-config.yaml` yang baru saja Anda buat. Edit file akan membuat perubahan berikut:
 - Di `Origins` objek, tambahkan ID OAC ke bidang yang diberi `OriginAccessControlId` nama.
 - Hapus nilai dari bidang yang diberi nama `OriginAccessIdentity`, jika ada.
 - Ubah nama `ETag` bidang menjadi `IfMatch`, tetapi jangan ubah nilai bidang.

Simpan file setelah selesai.

3. Gunakan perintah berikut untuk memperbarui distribusi untuk menggunakan kontrol akses asal.

```
aws cloudfront update-distribution --id <CloudFront distribution ID> --cli-  
input-yaml file://dist-config.yaml
```

Distribusi mulai menyebar ke semua lokasi CloudFront tepi. Saat lokasi edge menerima konfigurasi baru, ia akan menandatangani semua permintaan yang dikirimkan ke bucket origin S3.

API

Untuk membuat kontrol akses asal dengan CloudFront API, gunakan [CreateOriginAccessControl](#). Untuk informasi selengkapnya tentang bidang yang Anda tentukan dalam panggilan API ini, lihat dokumentasi referensi API untuk AWS SDK atau klien API lainnya.

Setelah membuat kontrol akses asal, Anda dapat melampirkannya ke bucket origin S3 dalam distribusi, menggunakan salah satu panggilan API berikut:

- Untuk melampirkannya ke distribusi yang ada, gunakan [UpdateDistribution](#).
- Untuk melampirkannya ke distribusi baru, gunakan [CreateDistribution](#).

Untuk kedua panggilan API ini, berikan ID kontrol akses asal di `OriginAccessControlId` bidang, di dalam asal. Untuk informasi selengkapnya tentang bidang lain yang Anda tentukan dalam panggilan API ini, lihat [Semua referensi pengaturan distribusi](#) dan dokumentasi referensi API untuk AWS SDK atau klien API lainnya.

Hapus distribusi dengan OAC yang terpasang pada bucket S3

Jika Anda perlu menghapus distribusi dengan OAC yang terpasang pada bucket S3, Anda harus menghapus distribusi sebelum menghapus asal bucket S3. Atau, sertakan Wilayah dalam nama domain asal. Jika ini tidak memungkinkan, Anda dapat menghapus OAC dari distribusi dengan beralih ke publik sebelum dihapus. Untuk informasi selengkapnya, lihat [Menghapus sebuah distribusi](#).

Migrasi dari Origin Access Identity (OAI) ke Origin Access Control (OAC)

Untuk bermigrasi dari identitas akses asal lama (OAI) ke kontrol akses asal (OAC), perbarui terlebih dahulu asal bucket S3 untuk memungkinkan OAI dan distribusi dengan OAC diaktifkan untuk mengakses konten bucket. Ini memastikan bahwa CloudFront tidak pernah kehilangan akses ke bucket selama transisi. Untuk memungkinkan OAI dan distribusi dengan OAC diaktifkan mengakses bucket S3, perbarui [kebijakan bucket](#) untuk menyertakan dua pernyataan, satu untuk setiap jenis prinsipal.

Contoh kebijakan bucket S3 berikut memungkinkan OAI dan distribusi dengan OAC diaktifkan untuk mengakses asal S3.

Example Kebijakan bucket S3 yang memungkinkan akses hanya-baca untuk OAI dan distribusi dengan OAC diaktifkan CloudFront

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCloudFrontServicePrincipalReadOnly",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudfront.amazonaws.com"
      },
      "Action": "s3:GetObject",
```

```

        "Resource": "arn:aws:s3:::<S3 bucket name>/*",
        "Condition": {
            "StringEquals": {
                "AWS:SourceArn":
"arn:aws:cloudfront::111122223333:distribution/<CloudFront distribution ID>"
            }
        },
        {
            "Sid": "AllowLegacyOAIReadOnly",
            "Effect": "Allow",
            "Principal": {
                "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access
Identity <origin access identity ID>"
            },
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::<S3 bucket name>/*"
        }
    ]
}

```

Setelah memperbarui kebijakan bucket asal S3 untuk mengizinkan akses ke OAI dan OAC, Anda dapat memperbarui konfigurasi distribusi untuk menggunakan OAC, bukan OAI. Untuk informasi selengkapnya, lihat [the section called “Buat kontrol akses asal baru”](#).

Setelah distribusi sepenuhnya diterapkan, Anda dapat menghapus pernyataan dalam kebijakan bucket yang memungkinkan akses ke OAI. Untuk informasi selengkapnya, lihat [the section called “Berikan CloudFront izin untuk mengakses bucket S3”](#).

Pengaturan lanjutan untuk kontrol akses asal

Fitur kontrol akses CloudFront asal mencakup pengaturan lanjutan yang ditujukan hanya untuk kasus penggunaan tertentu. Gunakan pengaturan yang disarankan kecuali Anda memiliki kebutuhan khusus untuk pengaturan lanjutan.

Kontrol akses asal berisi setelan bernama Perilaku penandatanganan (di konsol), atau SigningBehavior (di API, CLI, dan CloudFormation). Pengaturan ini menyediakan opsi berikut:

Selalu tandatangani permintaan asal (pengaturan yang disarankan)

Sebaiknya gunakan pengaturan ini, bernama **Permintaan tanda** (disarankan) di konsol, atau **always** di API, CLI, dan. CloudFormation Dengan pengaturan ini, CloudFront selalu tandatangani semua permintaan yang dikirimkan ke asal bucket S3.

Jangan pernah menandatangani permintaan asal

Pengaturan ini diberi nama **Jangan menandatangani permintaan** di konsol, atau **never** di API, CLI, dan. CloudFormation Gunakan pengaturan ini untuk menonaktifkan kontrol akses asal untuk semua asal di semua distribusi yang menggunakan kontrol akses asal ini. Ini dapat menghemat waktu dan tenaga dibandingkan dengan menghapus kontrol akses asal dari semua asal dan distribusi yang menggunakannya, satu per satu. Dengan pengaturan ini, CloudFront tidak menandatangani permintaan apa pun yang dikirimkan ke asal bucket S3.

Warning

Untuk menggunakan pengaturan ini, asal bucket S3 harus dapat diakses publik. Jika Anda menggunakan setelan ini dengan asal bucket S3 yang tidak dapat diakses publik, CloudFront tidak dapat mengakses asal. Asal bucket S3 mengembalikan kesalahan ke CloudFront dan CloudFront meneruskan kesalahan tersebut ke pemirsa.

Jangan mengganti header penampil (klien) **Authorization**

Pengaturan ini diberi nama **Jangan timpa header otorisasi** di konsol, atau **no-override** di API, CLI, dan. CloudFormation Gunakan setelan ini saat Anda CloudFront ingin menandatangani permintaan asal hanya jika permintaan penampil yang sesuai tidak menyertakan **Authorization** header. Dengan pengaturan ini, CloudFront meneruskan **Authorization** header dari permintaan penampil saat ada, tetapi menandatangani permintaan asal (menambahkan tajuknya sendiri **Authorization**) saat permintaan penampil tidak menyertakan **Authorization** header.

Warning

Untuk meneruskan **Authorization** header dari permintaan penampil, Anda harus menambahkan **Authorization** header ke [kebijakan cache](#) untuk semua perilaku cache yang menggunakan asal bucket S3 yang terkait dengan kontrol akses asal ini.

Gunakan identitas akses asal (warisan, tidak disarankan)

Ikhtisar identitas akses asal

CloudFront origin access identity (OAI) menyediakan fungsionalitas yang mirip dengan origin access control (OAC), tetapi tidak berfungsi untuk semua skenario. Secara khusus, OAI tidak mendukung:

- Bucket Amazon S3 di semua, termasuk Wilayah Wilayah AWS keikutsertaan
- Enkripsi sisi [server Amazon S3 dengan \(\) AWS KMS](#) SSE-KMS
- Permintaan dinamis (PUTPOST,, atauDELETE) ke Amazon S3
- Baru Wilayah AWS diluncurkan setelah Januari 2023

Tip

Kami menyarankan Anda menggunakan OAC sebagai gantinya. Untuk mengatur OAC, lihat [Buat kontrol akses asal baru](#). Untuk informasi tentang cara bermigrasi dari OAI ke OAC, lihat [the section called “Migrasi dari Origin Access Identity \(OAI\) ke Origin Access Control \(OAC\)”](#)

Berikan izin identitas akses asal untuk membaca file di bucket Amazon S3

Saat membuat OAI atau menambahkannya ke distribusi dengan CloudFront konsol, Anda dapat memperbarui kebijakan bucket Amazon S3 secara otomatis untuk memberikan izin kepada OAI untuk mengakses bucket Anda. Atau, Anda dapat memilih untuk membuat atau memperbarui kebijakan bucket secara manual. Metode apa pun yang Anda gunakan, Anda masih harus meninjau izin untuk memastikan bahwa:

- CloudFront OAI Anda dapat mengakses file dalam ember atas nama pemirsa yang memintanya. CloudFront
- Pemirsa tidak dapat menggunakan URL Amazon S3 untuk mengakses file Anda di luar. CloudFront

Important

Jika Anda mengonfigurasi CloudFront untuk menerima dan meneruskan semua metode HTTP yang CloudFront mendukung, pastikan Anda memberikan CloudFront OAI izin yang diinginkan. Misalnya, jika Anda mengonfigurasi CloudFront untuk menerima dan meneruskan

permintaan yang menggunakan DELETE metode ini, konfigurasi kebijakan bucket Anda untuk menangani DELETE permintaan dengan tepat sehingga penonton hanya dapat menghapus file yang Anda inginkan.

Menggunakan kebijakan bucket Amazon S3

Anda dapat memberikan akses CloudFront OAI ke file di bucket Amazon S3 dengan membuat atau memperbarui kebijakan bucket dengan cara berikut:

- [Menggunakan tab Izin bucket Amazon S3 di konsol Amazon S3](#).
- Menggunakan [PutBucketPolicy](#) di API Amazon S3.
- Menggunakan [CloudFront konsol](#). Saat menambahkan OAI ke setelan asal di CloudFront konsol, Anda dapat memilih Ya, perbarui kebijakan bucket untuk memberi tahu CloudFront agar kebijakan bucket diperbarui atas nama Anda.

Jika Anda memperbarui kebijakan keranjang secara manual, pastikan bahwa Anda:

- Tentukan OAI yang tepat sebagai `Principal` dalam kebijakan.
- Berikan izin yang diperlukan OAI untuk mengakses objek atas nama penampil.

Untuk informasi selengkapnya, silakan lihat bagian-bagian berikut ini.

Tentukan OAI sebagai **Principal** dalam kebijakan bucket

Untuk menentukan OAI sebagai kebijakan bucket Amazon S3, gunakan Nama Sumber Daya Amazon (ARN) OAI, yang menyertakan ID OAI. `Principal` Contoh:

```
"Principal": {
  "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access Identity <origin
access identity ID>"
}
```

Temukan ID OAI di CloudFront konsol di bawah Security, Origin access, Identities (legacy). Atau, gunakan [ListCloudFrontOriginAccessIdentities](#) di CloudFront API.

Berikan izin ke OAI

Untuk memberikan izin kepada OAI untuk mengakses objek di bucket Amazon S3 Anda, gunakan tindakan dalam kebijakan yang terkait dengan operasi API Amazon S3 tertentu. Misalnya, `s3:GetObject` tindakan memungkinkan OAI untuk membaca objek di ember. Untuk informasi selengkapnya, lihat contoh di bagian berikut, atau lihat [tindakan Amazon S3](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Contoh kebijakan bucket Amazon S3

Contoh berikut menunjukkan kebijakan bucket Amazon S3 yang memungkinkan CloudFront OAI mengakses bucket S3.

Temukan ID OAI di CloudFront konsol di bawah Security, Origin access, Identities (legacy). Atau, gunakan [ListCloudFrontOriginAccessIdentities](#) di CloudFront API.

Example Kebijakan buket Amazon S3 yang memberikan akses baca OAI

Contoh berikut memungkinkan OAI untuk membaca objek dalam buket yang ditentukan (`s3:GetObject`).

JSON

```
{
  "Version": "2012-10-17",
  "Id": "PolicyForCloudFrontPrivateContent",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access
Identity <origin access identity ID>"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::<S3 bucket name>/*"
    }
  ]
}
```

Example Kebijakan bucket Amazon S3 yang memberikan akses baca dan tulis OAI

Contoh berikut memungkinkan OAI untuk membaca dan menulis objek dalam bucket yang ditentukan (s3:GetObject dan s3:PutObject). Ini memungkinkan pemirsa untuk mengunggah file ke bucket Amazon S3 Anda. CloudFront

JSON

```
{
  "Version": "2012-10-17",
  "Id": "PolicyForCloudFrontPrivateContent",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::cloudfront:user/CloudFront Origin Access
Identity <origin access identity ID>"
      },
      "Action": [
        "s3:GetObject",
        "s3:PutObject"
      ],
      "Resource": "arn:aws:s3:::<S3 bucket name>/*"
    }
  ]
}
```

Gunakan ACL objek Amazon S3 (tidak disarankan)

Important

Sebaiknya [gunakan kebijakan bucket Amazon S3](#) untuk memberikan akses OAI ke bucket S3. Anda dapat menggunakan daftar kontrol akses (ACL) seperti yang dijelaskan di bagian ini, tetapi kami tidak merekomendasikannya.

Amazon S3 merekomendasikan untuk menyetel [Kepemilikan Objek S3](#) ke pemilik bucket yang diberlakukan, yang berarti ACL dinonaktifkan untuk bucket dan objek di dalamnya. Saat menerapkan pengaturan ini untuk Kepemilikan Objek, Anda harus menggunakan kebijakan bucket untuk memberikan akses ke OAI (lihat bagian sebelumnya).

Bagian berikut ini hanya untuk kasus penggunaan lama yang memerlukan ACL.

Anda dapat memberikan akses CloudFront OAI ke file di bucket Amazon S3 dengan membuat atau memperbarui ACL file dengan cara berikut:

- [Menggunakan tab Izin objek Amazon S3 di konsol Amazon S3.](#)
- Menggunakan [PutObjectAcl](#) di API Amazon S3.

Ketika Anda memberikan akses ke OAI menggunakan ACL, Anda harus menentukan OAI menggunakan ID pengguna kanonik Amazon S3. Di CloudFront konsol, Anda dapat menemukan ID ini di bawah Keamanan, akses Asal, Identitas (warisan). Jika Anda menggunakan CloudFront API, gunakan nilai `S3CanonicalUserId` elemen yang dikembalikan saat Anda membuat OAI, atau panggil [ListCloudFrontOriginAccessIdentities](#) di CloudFront API.

Menggunakan identitas akses asal di wilayah Amazon S3 yang hanya mendukung otentikasi tanda tangan versi 4

Wilayah Amazon S3 yang lebih baru mengharuskan Anda menggunakan Signature Version 4 untuk permintaan yang diautentikasi. (Untuk versi tanda tangan yang didukung di setiap Wilayah Amazon S3, lihat [titik akhir dan kuota Amazon Simple Storage Service](#) di.) Referensi Umum AWS Jika Anda menggunakan identitas akses asal dan jika bucket Anda berada di salah satu Wilayah yang memerlukan Tanda Tangan Versi 4, perhatikan hal berikut:

- DELETE, GET, HEAD, OPTIONS, dan PATCH permintaan didukung tanpa kualifikasi.
- POST permintaan tidak didukung.

Batasi akses dengan asal VPC

Anda dapat menggunakannya CloudFront untuk mengirimkan konten dari aplikasi yang di-host di subnet pribadi virtual private cloud (VPC) Anda. Anda dapat menggunakan Application Load Balancers (ALB), Network Load Balancers (NLBs), dan instans EC2 di subnet pribadi sebagai asal VPC.

Berikut adalah beberapa alasan mengapa Anda mungkin ingin menggunakan asal VPC:

- Keamanan - Asal VPC dirancang untuk meningkatkan postur keamanan aplikasi Anda dengan menempatkan penyeimbang beban dan instans EC2 Anda di subnet pribadi, menjadikan titik masuk tunggal. CloudFront Permintaan pengguna beralih dari CloudFront ke asal VPC melalui koneksi pribadi dan aman, memberikan keamanan tambahan untuk aplikasi Anda.

- **Manajemen** — Asal VPC mengurangi overhead operasional yang diperlukan untuk konektivitas yang aman antara CloudFront dan asal. Anda dapat memindahkan asal Anda ke subnet pribadi tanpa akses publik, dan Anda tidak perlu menerapkan daftar kontrol akses (ACL) atau mekanisme lain untuk membatasi akses ke asal Anda. Dengan cara ini, Anda tidak perlu berinvestasi dalam pekerjaan pengembangan yang tidak terdiferensiasi untuk mengamankan aplikasi web Anda. CloudFront
- **Skalabilitas dan kinerja** - Asal VPC membantu Anda mengamankan aplikasi web Anda, membebaskan waktu untuk fokus pada pengembangan aplikasi bisnis penting Anda sambil meningkatkan keamanan dan mempertahankan kinerja tinggi dan skalabilitas global dengan CloudFront. Asal VPC merampingkan manajemen keamanan dan mengurangi kompleksitas operasional sehingga Anda dapat menggunakannya CloudFront sebagai titik masuk tunggal untuk aplikasi Anda.

Tip

CloudFront mendukung berbagi asal-usul VPC di seluruh Akun AWS, apakah mereka berada di organisasi Anda atau tidak. Anda dapat membagikan asal VPC dari CloudFront konsol atau menggunakan AWS Resource Access Manager (AWS RAM). Untuk informasi selengkapnya, lihat [Bekerja dengan sumber daya bersama di CloudFront](#).

Prasyarat

Sebelum Anda membuat asal VPC untuk CloudFront distribusi Anda, Anda harus menyelesaikan yang berikut ini:

Konfigurasi VPC

Buat virtual private cloud (VPC) di Amazon VPC di salah satu Wilayah AWS yang didukung untuk asal VPC. Untuk informasi tentang membuat VPC, lihat [Membuat VPC plus sumber daya VPC lainnya di Panduan Pengguna Amazon VPC](#). Untuk mengetahui daftar Wilayah yang didukung, lihat [Didukung Wilayah AWS untuk asal VPC](#).

VPC Anda harus menyertakan hal-hal berikut:

- **Internet gateway** — Anda perlu menambahkan gateway internet ke VPC yang memiliki sumber daya asal VPC Anda. Gateway internet diperlukan untuk menunjukkan bahwa VPC dapat

menerima lalu lintas dari internet. Gateway internet tidak digunakan untuk merutekan lalu lintas ke asal di dalam subnet, dan Anda tidak perlu memperbarui kebijakan perutean.

- Subnet pribadi dengan setidaknya satu alamat IPv4 yang tersedia — CloudFront rute ke subnet Anda dengan menggunakan service-managed elastic network interface (ENI CloudFront) yang dibuat setelah Anda menentukan sumber daya asal VPC Anda. CloudFront Anda harus memiliki setidaknya satu alamat IPv4 yang tersedia di subnet pribadi Anda sehingga proses pembuatan ENI dapat berhasil. Alamat IPv4 dapat bersifat pribadi, dan tidak ada biaya tambahan untuk itu. IPv6-only subnet tidak didukung.

Sumber Asal

Di subnet pribadi, luncurkan Application Load Balancer, Network Load Balancer, atau instans EC2 untuk digunakan sebagai asal Anda. Sumber daya yang Anda luncurkan harus sepenuhnya digunakan dan dalam status Aktif sebelum Anda dapat menggunakannya untuk asal VPC.

Pembatasan asal:

- Gateway Load Balancers tidak dapat ditambahkan sebagai asal
- Dual-stack Network Load Balancers tidak dapat ditambahkan sebagai asal
- Network Load Balancers dengan pendengar TLS tidak dapat ditambahkan sebagai asal
- Untuk digunakan sebagai asal VPC, Network Load Balancer harus memiliki grup keamanan yang melekat padanya

Konfigurasi Grup Keamanan

Sumber daya asal VPC Anda (Application Load Balancer, Network Load Balancer, atau instans EC2) harus memiliki grup keamanan yang terpasang. Saat Anda membuat asal VPC, CloudFront secara otomatis membuat grup keamanan yang dikelola layanan dengan pola penamaan. CloudFront-VPCorigins-Service-SG Grup keamanan ini sepenuhnya dikelola oleh AWS, dan tidak boleh diedit.

Untuk mengizinkan lalu lintas dari CloudFront untuk mencapai asal VPC Anda, perbarui grup keamanan yang dilampirkan ke sumber daya asal Anda (instans ALB, NLB, atau EC2) untuk mengizinkan lalu lintas masuk menggunakan salah satu metode berikut:

- Opsi 1: Izinkan lalu lintas dari daftar awalan CloudFront terkelola. Untuk informasi selengkapnya, lihat [Gunakan daftar awalan CloudFront terkelola](#). Ini dapat dilakukan sebelum asal VPC dibuat juga.

- Opsi 2: Izinkan lalu lintas dari grup keamanan CloudFront yang dikelola layanan ()CloudFront-VPCOrigins-Service-SG. Ini dapat dilakukan hanya setelah asal VPC dibuat dan grup keamanan yang dikelola layanan dibuat. Konfigurasi ini lebih ketat karena membatasi lalu lintas hanya untuk distribusi Anda CloudFront .

Important

Jangan membuat grup keamanan Anda sendiri dengan nama yang dimulai denganCloudFront-VPCOrigins-Service-SG. Ini adalah pola penamaan yang AWS dicadangkan untuk grup keamanan yang dikelola layanan. Untuk informasi selengkapnya, lihat [Membuat grup keamanan](#).

Pembatasan Protokol dan Fitur

Asal VPC tidak mendukung hal berikut:

- lalu lintas gRPC
- Permintaan asal dan respons asal dipicu dengan Lambda @Edge
- Network Access Control Lists (NACLs): Subnet-level mengizinkan dan menolak aturan tidak dievaluasi untuk lalu lintas ini.

Buat asal VPC (distribusi baru)

Prosedur berikut menunjukkan cara membuat asal VPC untuk CloudFront distribusi baru Anda di CloudFront konsol. Atau, Anda dapat menggunakan operasi [CreateVpcOrigin](#) dan [CreateDistribution](#) API dengan AWS CLI atau AWS SDK.

Untuk membuat asal VPC untuk distribusi baru CloudFront

1. Buka CloudFront konsol di<https://console.aws.amazon.com/cloudfront/v4/home>.
2. Pilih asal VPC, Buat asal VPC.
3. Isi bidang yang diperlukan. Untuk Origin ARN, pilih ARN Application Load Balancer, Network Load Balancer, atau instans EC2 Anda. Jika Anda tidak melihat ARN, Anda dapat menyalin ARN sumber daya spesifik Anda dan menempelkannya di sini.
4. Pilih Buat asal VPC.

5. Tunggu status asal VPC Anda berubah menjadi Deployed. Ini bisa memakan waktu hingga 15 menit.
6. Pilih Distribusi, Buat distribusi.
7. Untuk domain Origin, pilih sumber daya asal VPC Anda dari daftar tarik-turun.

Jika asal VPC Anda adalah instans EC2, salin dan tempel nama DNS IP Pribadi instance ke bidang domain Origin.

8. Selesai membuat distribusi Anda. Untuk informasi selengkapnya, lihat [Buat CloudFront distribusi di konsol](#).

Buat asal VPC (distribusi yang ada)

Prosedur berikut menunjukkan kepada Anda cara membuat asal VPC untuk CloudFront distribusi yang ada di CloudFront konsol, yang membantu memastikan ketersediaan aplikasi Anda secara berkelanjutan. Atau, Anda dapat menggunakan operasi [CreateVpcOrigin](#) dan [UpdateDistributionWithStagingConfig](#) API dengan AWS CLI atau AWS SDK.

Secara opsional, Anda dapat memilih untuk menambahkan asal VPC Anda ke distribusi yang ada tanpa membuat distribusi pementasan.

Untuk membuat asal VPC untuk distribusi yang ada CloudFront

1. Buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Pilih asal VPC, Buat asal VPC.
3. Isi bidang yang diperlukan. Untuk Origin ARN, pilih ARN Application Load Balancer, Network Load Balancer, atau instans EC2 Anda. Jika Anda tidak melihat ARN, Anda dapat menyalin ARN sumber daya spesifik Anda dan menempelkannya di sini.
4. Pilih Buat asal VPC.
5. Tunggu status asal VPC Anda berubah menjadi Deployed. Ini bisa memakan waktu hingga 15 menit.
6. Di panel navigasi, pilih Distribusi.
7. Pilih ID distribusi Anda.
8. Pada tab Umum, di bawah Penerapan berkelanjutan, pilih Buat distribusi pementasan. Untuk informasi selengkapnya, lihat [Gunakan penerapan CloudFront berkelanjutan untuk menguji perubahan konfigurasi CDN dengan aman](#).

9. Ikuti langkah-langkah di wizard Buat distribusi pementasan untuk membuat distribusi pementasan. Sertakan langkah-langkah berikut:
 - Untuk Origins, pilih Create Origin.
 - Untuk domain Origin, pilih sumber daya asal VPC Anda dari menu tarik-turun.

Jika asal VPC Anda adalah instans EC2, salin dan tempel nama DNS IP Pribadi instance ke bidang domain Origin.
 - Pilih Buat asal.
10. Dalam distribusi pementasan Anda, uji asal VPC.
11. Promosikan konfigurasi distribusi pementasan ke distribusi utama Anda. Untuk informasi selengkapnya, lihat [Mempromosikan konfigurasi distribusi pementasan](#).
12. Hapus akses publik ke asal VPC Anda dengan membuat subnet pribadi. Setelah Anda melakukan ini, asal VPC tidak akan dapat ditemukan melalui internet, tetapi masih CloudFront akan memiliki akses pribadi ke sana. Untuk informasi selengkapnya, lihat [Mengaitkan atau memisahkan subnet dengan tabel rute di Panduan](#) Pengguna Amazon VPC.

Perbarui asal VPC

Prosedur berikut menunjukkan cara memperbarui asal VPC untuk CloudFront distribusi Anda di konsol. CloudFront Atau, Anda dapat menggunakan operasi [UpdateDistribution](#) dan [UpdateVpcOrigin](#) API dengan AWS CLI atau AWS SDK.

Untuk memperbarui asal VPC yang ada untuk distribusi Anda CloudFront

1. Buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Distribusi.
3. Pilih ID distribusi Anda.
4. Pilih Perilaku tab.
5. Pastikan bahwa asal VPC bukan asal default untuk perilaku cache Anda.
6. Pilih tab Origins.
7. Pilih asal VPC yang akan Anda perbarui dan pilih Hapus. Ini memisahkan asal VPC dari distribusi Anda. Ulangi langkah 2-7 untuk memisahkan asal VPC dari distribusi lainnya.
8. Pilih asal VPC.
9. Pilih asal VPC dan pilih Edit.

10. Buat pembaruan Anda dan pilih Perbarui asal VPC.
11. Tunggu status asal VPC Anda berubah menjadi Deployed. Ini bisa memakan waktu hingga 15 menit.
12. Di panel navigasi, pilih Distribusi.
13. Pilih ID distribusi Anda.
14. Pilih tab Origins.
15. Pilih Buat asal.
16. Untuk domain Origin, pilih sumber daya asal VPC Anda dari menu tarik-turun.

Jika asal VPC Anda adalah instans EC2, salin dan tempel nama DNS IP Pribadi instance ke bidang domain Origin.

17. Pilih Buat asal. Ini mengaitkan asal VPC dengan distribusi Anda lagi. Ulangi langkah 12-17 untuk mengaitkan asal VPC yang diperbarui dengan distribusi lainnya.

Didukung Wilayah AWS untuk asal VPC

Asal VPC saat ini didukung dalam iklan berikut. Wilayah AWS Pengecualian Availability Zone (AZ) dicatat.

Nama Wilayah	Wilayah
AS Timur (Ohio)	us-east-2
AS Timur (Virginia Utara)	us-east-1 (except AZ use1-az3)
AS Barat (California Utara)	us-west-1 (except AZ usw1-az2)
AS Barat (Oregon)	us-west-2
Afrika (Cape Town)	af-south-1
Asia Pasifik (Hong Kong)	ap-east-1
Asia Pasifik (Mumbai)	ap-south-1
Asia Pasifik (Hyderabad)	ap-south-2
Asia Pasifik (Jakarta)	ap-southeast-3

Nama Wilayah	Wilayah
Asia Pacific (Melbourne)	ap-southeast-4
Asia Pasifik (Osaka)	ap-northeast-3
Asia Pasifik (Singapura)	ap-southeast-1
Asia Pasifik (Sydney)	ap-southeast-2
Asia Pasifik (Tokyo)	ap-northeast-1 (except AZ apne1-az3)
Asia Pasifik (Seoul)	ap-northeast-2
Asia Pasifik (Thailand)	ap-southeast-7
Asia Pasifik (Malaysia)	ap-southeast-5
Asia Pasifik (Selandia Baru)	ap-southeast-6
Asia Pasifik (Taipei)	ap-east-2
Kanada (Pusat)	ca-central-1 (except AZ cac1-az3)
Kanada Barat (Calgary)	ca-west-1
Eropa (Frankfurt)	eu-central-1
Eropa (Irlandia)	eu-west-1
Eropa (London)	eu-west-2
Eropa (Milan)	eu-south-1
Eropa (Paris)	eu-west-3
Eropa (Spanyol)	eu-south-2
Eropa (Stockholm)	eu-north-1
Eropa (Zürich)	eu-central-2

Nama Wilayah	Wilayah
Israel (Tel Aviv)	<code>il-central-1</code>
Timur Tengah (Bahrain)	<code>me-south-1</code>
Timur Tengah (UEA)	<code>me-central-1</code>
Amerika Selatan (Sao Paulo)	<code>sa-east-1</code>
Meksiko (Tengah)	<code>mx-central-1</code>

Batasi akses ke Application Load Balancers

Anda dapat menggunakan Application Load Balancer internal dan internet dengan Amazon CloudFront. Anda dapat menggunakan Application Load Balancer internal di dalam subnet pribadi dengan CloudFront menggunakan asal VPC. CloudFront Asal VPC memungkinkan Anda untuk menyajikan konten dari aplikasi yang dihosting di subnet VPC pribadi tanpa memaparkannya ke internet publik. Untuk informasi selengkapnya, lihat [Batasi akses dengan asal VPC](#).

Jika Anda menggunakan Application Load Balancer CloudFront yang menghadap ke internet, Anda dapat menggunakan mitigasi keamanan berikut untuk mencegah pengguna mengakses Application Load Balancer secara langsung, dan mengizinkan akses hanya melalui CloudFront

1. Konfigurasi CloudFront untuk menambahkan header HTTP kustom ke permintaan yang dikirim ke Application Load Balancer.
2. Mengonfigurasi Application Load Balancer untuk hanya meneruskan permintaan yang berisi header HTTP kustom.
3. Memerlukan HTTPS untuk meningkatkan keamanan solusi ini.

CloudFront juga dapat membantu mengurangi latensi dan bahkan menyerap beberapa serangan penolakan layanan (DDoS) terdistribusi.

Jika kasus penggunaan Anda memerlukan akses ganda ke aplikasi web dari keduanya CloudFront dan Application Load Balancer langsung melalui internet, pertimbangkan untuk memisahkan aplikasi APIs web Anda sebagai berikut:

- APIs yang harus dilalui CloudFront. Dalam hal ini, pertimbangkan untuk menggunakan Application Load Balancer pribadi terpisah sebagai asal.
- APIs yang membutuhkan akses melalui Application Load Balancer. Dalam hal ini, Anda melewati CloudFront.

Atau, untuk aplikasi web atau konten lain yang disajikan oleh Application Load Balancer yang menghadap ke internet di Elastic Load Balancing CloudFront, dapat menyimpan objek dan menyajikannya langsung ke pengguna (pemirsa), mengurangi beban pada Application Load Balancer Anda. Penyeimbang beban yang menghadap ke internet memiliki nama DNS yang dapat diselesaikan secara publik dan mengarahkan permintaan dari klien ke target melalui internet.

Untuk informasi selengkapnya, lihat topik berikut. Setelah Anda menyelesaikan langkah-langkah ini, pengguna hanya dapat mengakses Application Load Balancer Anda melalui CloudFront

Topik

- [Konfigurasi CloudFront untuk menambahkan header HTTP kustom ke permintaan](#)
- [Konfigurasi Application Load Balancer untuk hanya meneruskan permintaan yang berisi header tertentu](#)
- [\(Opsional\) tingkatkan keamanan solusi ini.](#)
- [\(Opsional\) Batasi akses ke asal dengan menggunakan daftar awalan AWS-managed untuk CloudFront](#)

Konfigurasi CloudFront untuk menambahkan header HTTP kustom ke permintaan

Anda dapat mengonfigurasi CloudFront untuk menambahkan header HTTP kustom ke permintaan yang dikirim ke asal Anda (dalam hal ini, Application Load Balancer).

Important

Kasus penggunaan ini bergantung pada menjaga nama header kustom dan rahasia nilai. Jika nama header dan nilai tidak rahasia, klien HTTP lain berpotensi memasukkannya dalam permintaan yang mereka kirim langsung ke Application Load Balancer. Hal ini dapat menyebabkan Application Load Balancer berperilaku seolah-olah permintaan berasal dari CloudFront saat tidak. Untuk mencegah hal ini, rahasiakan nama header kustom dan nilai.

Anda dapat mengonfigurasi CloudFront untuk menambahkan header HTTP kustom ke permintaan asal dengan CloudFront konsol CloudFormation, atau CloudFront API.

Untuk menambahkan header HTTP kustom (CloudFront konsol)

Di CloudFront konsol, gunakan pengaturan Header Kustom Asal di Pengaturan Asal. Masukkan Nama Header dan Nilainya.

Note

Dalam produksi, gunakan nama dan nilai header yang dihasilkan secara acak. Perlakukan nama dan nilai header sebagai kredensial aman, seperti nama pengguna dan kata sandi.

Anda dapat mengedit setelan Header Kustom Asal saat membuat atau mengedit asal untuk CloudFront distribusi yang ada, dan saat Anda membuat distribusi baru. Untuk informasi selengkapnya, lihat [Perbarui distribusi](#) dan [Buat distribusi](#).

Untuk menambah header HTTP kustom (CloudFormation)

Dalam CloudFormation template, gunakan `OriginCustomHeaders` properti, seperti yang ditunjukkan pada contoh berikut.

Note

Nama header dan nilai dalam contoh ini hanya untuk demonstrasi. Dalam produksi, gunakan nilai yang dihasilkan secara acak. Perlakukan nama header dan nilai sebagai kredensial aman, seperti nama pengguna dan kata sandi.

```
AWSTemplateFormatVersion: '2010-09-09'
Resources:
  TestDistribution:
    Type: 'AWS::CloudFront::Distribution'
    Properties:
      DistributionConfig:
        Origins:
          - DomainName: app-load-balancer.example.com
            Id: Example-ALB
            CustomOriginConfig:
              OriginProtocolPolicy: https-only
```

```
OriginSSLProtocols:
  - TLSv1.2
OriginCustomHeaders:
  - HeaderName: X-Custom-Header
    HeaderValue: random-value-1234567890
Enabled: 'true'
DefaultCacheBehavior:
  TargetOriginId: Example-ALB
  ViewerProtocolPolicy: allow-all
  CachePolicyId: 658327ea-f89d-4fab-a63d-7e88639e58f6
  PriceClass: PriceClass_All
  ViewerCertificate:
    CloudFrontDefaultCertificate: 'true'
```

Untuk informasi selengkapnya, lihat [Asal](#) dan [OriginCustomHeader](#) properti di Panduan AWS CloudFormation Pengguna.

Untuk menambahkan header HTTP kustom (CloudFront API)

Di CloudFront API, gunakan `CustomHeaders` objek di dalamnya `Origin`. Untuk informasi selengkapnya, lihat [CreateDistribution](#) dan [UpdateDistribution](#) di Referensi Amazon CloudFront API, dan dokumentasi untuk SDK Anda atau klien API lainnya.

Ada beberapa nama header yang Anda tidak dapat tentukan sebagai header kustom asal. Untuk informasi selengkapnya, lihat [Header khusus yang tidak CloudFront dapat ditambahkan ke permintaan asal](#).

Konfigurasi Application Load Balancer untuk hanya meneruskan permintaan yang berisi header tertentu

Setelah Anda mengonfigurasi CloudFront untuk menambahkan header HTTP kustom ke permintaan yang dikirimkan ke Application Load Balancer Anda (lihat [bagian sebelumnya](#)), Anda dapat mengonfigurasi penyeimbang beban untuk hanya meneruskan permintaan yang berisi header kustom ini. Anda melakukan ini dengan menambah aturan baru dan memodifikasi aturan default dalam listener penyeimbang beban Anda.

Prasyarat

Untuk menggunakan prosedur berikut, Anda memerlukan Application Load Balancer dengan setidaknya satu listener. Jika Anda belum membuatnya, lihat [Membuat Application Load Balancer](#) di Panduan Pengguna untuk Application Load Balancers.

Prosedur berikut ini memodifikasi listener HTTPS. Anda dapat menggunakan proses yang sama untuk memodifikasi listener HTTP.

Untuk memperbarui aturan dalam listener Application Load Balancer

1. Tambahkan aturan baru. Gunakan instruksi dari [Tambahkan aturan](#), dengan modifikasi berikut:
 - Tambahkan aturan ke penyeimbang beban yang merupakan asal CloudFront distribusi Anda.
 - Untuk kondisi Tambah, pilih header Http. Tentukan nama header HTTP dan nilai yang Anda tambahkan sebagai header kustom asal CloudFront.
 - Untuk menambahkan tindakan, pilih Teruskan ke. Pilih grup target tempat Anda ingin meneruskan permintaan.
2. Edit aturan default di pendengar penyeimbang beban Anda. Gunakan instruksi dari [Edit aturan](#), dengan modifikasi berikut:
 - Edit aturan default penyeimbang beban yang merupakan asal CloudFront distribusi Anda.
 - Hapus tindakan default, lalu untuk Add action, pilih Return fixed response.
 - Untuk Kode respons, masukkan **403**.
 - Untuk Isi respons, masukkan **Access denied**.

Setelah Anda menyelesaikan langkah-langkah ini, pendengar penyeimbang beban Anda memiliki dua aturan. Satu aturan meneruskan permintaan yang berisi header HTTP (permintaan yang berasal dari CloudFront). Aturan lain mengirimkan respons tetap ke semua permintaan lainnya (permintaan yang tidak berasal CloudFront).

Anda dapat memverifikasi bahwa solusi berfungsi dengan mengirimkan permintaan ke CloudFront distribusi Anda dan satu ke Application Load Balancer Anda. Permintaan untuk CloudFront mengembalikan aplikasi web atau konten Anda, dan yang dikirim langsung ke Application Load Balancer Anda mengembalikan 403 respons dengan pesan teks biasa. Access denied

(Opsional) tingkatkan keamanan solusi ini.

Untuk meningkatkan keamanan solusi ini, Anda dapat mengonfigurasi CloudFront distribusi Anda agar selalu menggunakan HTTPS saat mengirim permintaan ke Application Load Balancer Anda. Ingat, solusi ini hanya berfungsi jika Anda menyimpan nama header kustom dan rahasia nilai. Menggunakan HTTPS dapat membantu mencegah penyadap menemukan nama dan nilai header. Kami juga merekomendasikan merotasi nama header dan nilai secara berkala.

Menggunakan HTTPS untuk permintaan asal

CloudFront Untuk mengonfigurasi penggunaan HTTPS untuk permintaan asal, setel pengaturan Kebijakan Protokol Asal ke HTTPS Saja. Pengaturan ini tersedia di CloudFront konsol, CloudFormation, dan CloudFront API. Untuk informasi selengkapnya, lihat [Protokol \(hanya asal kustom\)](#).

Berikut ini juga berlaku saat Anda mengonfigurasi CloudFront untuk menggunakan HTTPS untuk permintaan asal:

- Anda harus mengonfigurasi CloudFront untuk meneruskan Host header ke asal dengan kebijakan permintaan asal. Anda dapat menggunakan [kebijakan permintaan asal AllViewer terkelola](#).
- Pastikan Application Load Balancer Anda memiliki listener HTTPS (seperti yang ditunjukkan pada [bagian sebelumnya](#)). Untuk informasi lebih lanjut, lihat [Buat listener HTTPS](#) di Panduan pengguna untuk Application Load Balancers. Menggunakan pendengar HTTPS mengharuskan Anda memiliki SSL/TLS sertifikat yang cocok dengan nama domain yang dirutekan ke Application Load Balancer Anda.
- Sertifikat SSL/TLS untuk hanya CloudFront dapat diminta (atau diimpor) di in (ACM). us-east-1 Wilayah AWS AWS Certificate Manager Karena CloudFront merupakan layanan global, secara otomatis mendistribusikan sertifikat dari us-east-1 Wilayah ke semua Wilayah yang terkait dengan CloudFront distribusi Anda.
- Misalnya, jika Anda memiliki Application Load Balancer (ALB) di ap-southeast-2 Wilayah, Anda harus mengonfigurasi SSL/TLS sertifikat di ap-southeast-2 Wilayah (untuk menggunakan HTTPS antara CloudFront dan asal ALB) dan us-east-1 Wilayah (untuk menggunakan HTTPS antara pemirsa dan). CloudFront Kedua sertifikat harus sesuai dengan nama domain yang dirutekan ke Application Load Balancer Anda. Untuk informasi selengkapnya, lihat [Wilayah AWS untuk AWS Certificate Manager](#).
- Jika pengguna akhir (juga dikenal sebagai pemirsa, atau klien) aplikasi web Anda dapat menggunakan HTTPS, Anda juga dapat mengonfigurasi CloudFront untuk memilih (atau bahkan memerlukan) koneksi HTTPS dari pengguna akhir. Untuk melakukannya, gunakan pengaturan Kebijakan Protokol Penampil. Anda dapat mengaturnya untuk mengarahkan pengguna akhir dari HTTP ke HTTPS, atau untuk menolak permintaan yang menggunakan HTTP. Pengaturan ini tersedia di CloudFront konsol, CloudFormation, dan CloudFront API. Untuk informasi selengkapnya, lihat [Kebijakan protokol penampil](#).

Memutar nama header dan nilai

Selain menggunakan HTTPS, kami juga merekomendasikan merotasi nama header dan nilai secara berkala. Langkah-langkah tingkat tinggi untuk melakukan ini adalah sebagai berikut:

1. Konfigurasi CloudFront untuk menambahkan header HTTP kustom tambahan ke permintaan yang dikirim ke Application Load Balancer.
2. Perbarui aturan listener Application Load Balancer untuk meneruskan permintaan yang berisi header HTTP kustom tambahan ini.
3. Konfigurasi CloudFront untuk berhenti menambahkan header HTTP kustom asli ke permintaan yang dikirim ke Application Load Balancer.
4. Perbarui aturan listener Application Load Balancer untuk menghentikan penerusan permintaan yang berisi header HTTP kustom tambahan ini.

Untuk informasi lebih lanjut tentang pencapaian langkah-langkah ini, lihat bagian sebelumnya.

(Opsional) Batasi akses ke asal dengan menggunakan daftar awalan AWS-managed untuk CloudFront

Untuk lebih membatasi akses ke Application Load Balancer, Anda dapat mengonfigurasi grup keamanan yang terkait dengan Application Load Balancer sehingga hanya menerima CloudFront lalu lintas dari saat layanan AWS menggunakan daftar awalan -managed. Ini mencegah lalu lintas yang tidak berasal CloudFront dari mencapai Application Load Balancer Anda di lapisan jaringan (layer 3) atau lapisan transport (layer 4).

Untuk informasi selengkapnya, lihat [Batasi akses ke asal Anda menggunakan daftar awalan AWS-managed untuk CloudFront posting blog Amazon](#).

Batasi distribusi geografis konten Anda

Anda dapat menggunakan pembatasan geografis, kadang-kadang dikenal sebagai pemblokiran geografis, untuk mencegah pengguna di lokasi geografis tertentu mengakses konten yang Anda distribusikan melalui distribusi Amazon CloudFront. Untuk menggunakan batasan geografis, Anda memiliki dua opsi:

- Gunakan fitur pembatasan CloudFront geografis. Gunakan opsi ini untuk membatasi akses ke semua file yang terkait dengan distribusi dan untuk membatasi akses di tingkat negara.

- Gunakan layanan geolokasi pihak ketiga. Gunakan opsi ini untuk membatasi akses ke subset file yang terkait dengan distribusi atau untuk membatasi akses pada granularitas yang lebih baik dari tingkat negara.

Topik

- [Gunakan CloudFront batasan geografis](#)
- [Gunakan layanan geolokasi pihak ketiga](#)

Gunakan CloudFront batasan geografis

Ketika pengguna meminta konten Anda, CloudFront biasanya menyajikan konten yang diminta di mana pun pengguna berada. Jika Anda perlu mencegah pengguna di negara tertentu mengakses konten Anda, Anda dapat menggunakan fitur pembatasan CloudFront geografis untuk melakukan salah satu hal berikut:

- Berikan izin kepada pengguna Anda untuk mengakses konten Anda hanya jika mereka berada di salah satu negara yang disetujui di daftar izin Anda.
- Cegah pengguna Anda mengakses konten Anda jika mereka berada di salah satu negara terlarang di denylist Anda.

Misalnya, jika permintaan berasal dari negara di mana Anda tidak berwenang untuk mendistribusikan konten Anda, Anda dapat menggunakan batasan CloudFront geografis untuk memblokir permintaan tersebut.

Note

CloudFront menentukan lokasi pengguna Anda dengan menggunakan database pihak ketiga. Keakuratan pemetaan antara alamat IP dan negara berbeda-beda berdasarkan Wilayah. Berdasarkan pengujian terbaru, keseluruhan keakuratannya adalah 99,8%. Jika tidak CloudFront dapat menentukan lokasi pengguna, CloudFront menyajikan konten yang diminta pengguna.

Begini cara kerja pembatasan geografis:

1. Misalkan Anda memiliki hak untuk mendistribusikan konten Anda hanya di Liechtenstein. Anda memperbarui CloudFront distribusi Anda untuk menambahkan daftar izin yang hanya berisi Liechtenstein. (Atau, Anda dapat menambahkan denlist yang berisi setiap negara kecuali Liechtenstein.)
2. Seorang pengguna di Monako meminta konten Anda, dan DNS merutekan permintaan ke lokasi CloudFront tepi di Milan, Italia.
3. Lokasi edge di Milan mencari distribusi Anda dan menentukan bahwa pengguna di Monako tidak memiliki izin untuk mengunduh konten Anda.
4. CloudFront mengembalikan kode status HTTP 403 (Forbidden) ke pengguna.

Note

Pembatasan geografis tidak berlaku untuk permintaan validasi sertifikat untuk sertifikat CloudFront terkelola. Bila Anda menggunakan sertifikat CloudFront terkelola, permintaan ke `/.well-known/pki-validation/` jalur dikecualikan dari aturan pembatasan geografis. Ini memastikan bahwa Otoritas Sertifikat dapat memvalidasi sertifikat Anda dari berbagai perspektif jaringan yang didistribusikan secara geografis yang diperlukan untuk penerbitan dan pembaruan, terlepas dari konfigurasi pembatasan geografis Anda.

Jika Anda menggunakan AWS WAF distribusi Anda, pastikan bahwa AWS WAF aturan Anda (termasuk aturan kontrol bot) tidak memblokir permintaan ke `/.well-known/pki-validation/` jalur. Memblokir permintaan ini dapat mencegah validasi sertifikat berhasil diselesaikan.

Anda dapat mengonfigurasi secara opsional CloudFront untuk mengembalikan pesan kesalahan khusus kepada pengguna, dan Anda dapat menentukan berapa lama Anda CloudFront ingin menyimpan respons kesalahan untuk file yang diminta. Nilai default adalah 10 detik. Untuk informasi selengkapnya, lihat [Buat halaman kesalahan khusus untuk kode status HTTP tertentu](#).

Pembatasan geografis berlaku untuk seluruh distribusi. Jika Anda perlu menerapkan satu batasan pada bagian konten Anda dan pembatasan yang berbeda (atau tidak ada batasan) ke bagian lain dari konten Anda, Anda harus membuat CloudFront distribusi terpisah atau [menggunakan](#) layanan geolokasi pihak ketiga.

Jika Anda mengaktifkan [log CloudFront standar](#) (log akses), Anda dapat mengidentifikasi permintaan yang CloudFront ditolak dengan mencari entri log di mana nilai `sc-status` (kode status HTTP)

berada403. Namun, hanya dengan menggunakan log standar, Anda tidak dapat membedakan permintaan yang CloudFront ditolak berdasarkan lokasi pengguna dari permintaan yang CloudFront ditolak karena pengguna tidak memiliki izin untuk mengakses file karena alasan lain. Jika Anda memiliki layanan geolokasi pihak ketiga seperti Elemen Digital atau MaxMind, Anda dapat mengidentifikasi lokasi permintaan berdasarkan alamat IP di kolom `c-ip` (IP klien) di log akses. Untuk informasi selengkapnya tentang log CloudFront standar, lihat [Akses log \(log standar\)](#).

Prosedur berikut menjelaskan cara menggunakan CloudFront konsol untuk menambahkan batasan geografis ke distribusi yang ada. Untuk informasi tentang cara menggunakan konsol untuk membuat distribusi, lihat [Buat distribusi](#).

Untuk menambahkan batasan geografis ke distribusi CloudFront web Anda (konsol)

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Distribusi, lalu pilih distribusi yang ingin Anda perbarui.
3. Pilih tab Keamanan, lalu pilih Pembatasan geografis.
4. Pilih Edit.
5. Pilih Izinkan daftar untuk membuat daftar negara yang diizinkan, atau Blokir daftar untuk membuat daftar negara yang diblokir.
6. Tambahkan negara yang diinginkan ke daftar, lalu pilih Simpan perubahan.

Gunakan layanan geolokasi pihak ketiga

Dengan fitur pembatasan CloudFront geografis, Anda mengontrol distribusi konten Anda di tingkat negara untuk semua file yang Anda distribusikan dengan distribusi web tertentu. Jika Anda memiliki kasus penggunaan untuk pembatasan geografis di mana pembatasan tidak mengikuti batas negara, atau jika Anda ingin membatasi akses hanya ke beberapa file yang Anda layani oleh distribusi tertentu, Anda dapat menggabungkan CloudFront dengan layanan geolokasi pihak ketiga. Ini memberi Anda kontrol atas konten Anda tidak hanya berdasarkan negara tetapi juga berdasarkan kota, ZIP, atau kode pos, atau bahkan garis lintang dan bujur.

Saat menggunakan layanan geolokasi pihak ketiga, sebaiknya gunakan URL yang CloudFront ditandatangani, yang dengannya Anda dapat menentukan tanggal dan waktu kedaluwarsa setelah URL tidak lagi valid. Selain itu, kami menyarankan Anda menggunakan bucket Amazon S3 sebagai asal Anda karena Anda kemudian dapat menggunakan [kontrol akses CloudFront asal](#) untuk

mencegah pengguna mengakses konten Anda langsung dari asal. Untuk informasi selengkapnya tentang URL yang ditandatangani dan kontrol akses asal, lihat [Sajikan konten pribadi dengan URL yang ditandatangani dan cookie yang ditandatangani](#).

Langkah-langkah berikut menjelaskan cara mengontrol akses ke file Anda dengan menggunakan layanan geolokasi pihak ketiga.

Untuk menggunakan layanan geolokasi pihak ketiga untuk membatasi akses ke file dalam distribusi CloudFront

1. Dapatkan akun dengan layanan geolokasi.
2. Unggah konten Anda ke keranjang Amazon S3.
3. Konfigurasi Amazon CloudFront dan Amazon S3 untuk menyajikan konten pribadi. Untuk informasi selengkapnya, lihat [Sajikan konten pribadi dengan URL yang ditandatangani dan cookie yang ditandatangani](#).
4. Tulis aplikasi web Anda untuk melakukan hal berikut:
 - Kirim alamat IP untuk setiap permintaan pengguna ke layanan geolokasi.
 - Evaluasi nilai pengembalian dari layanan geolokasi untuk menentukan apakah pengguna berada di lokasi tempat Anda CloudFront ingin mendistribusikan konten Anda.
 - Jika Anda ingin mendistribusikan konten Anda ke lokasi pengguna, buat URL yang ditandatangani untuk CloudFront konten Anda. Jika Anda tidak ingin mendistribusikan konten ke lokasi tersebut, kembalikan kode status HTTP 403 (Forbidden) ke pengguna. Atau, Anda dapat mengonfigurasi CloudFront untuk mengembalikan pesan kesalahan khusus. Untuk informasi selengkapnya, lihat [the section called “Buat halaman kesalahan khusus untuk kode status HTTP tertentu”](#).

Untuk informasi selengkapnya, lihat dokumentasi untuk layanan geolokasi yang Anda gunakan.

Anda dapat menggunakan variabel server web untuk mendapatkan alamat IP pengguna yang mengunjungi situs web Anda. Perhatikan peringatan berikut ini:

- Jika server web Anda tidak terhubung ke internet melalui neraca beban, Anda dapat menggunakan variabel server web untuk mendapatkan alamat IP jarak jauh. Namun, alamat IP ini tidak selalu alamat IP pengguna. Ini juga dapat berupa alamat IP dari server proksi, bergantung pada bagaimana pengguna terhubung ke internet.

- Jika server web Anda terhubung ke internet melalui neraca beban, variabel server web mungkin berisi alamat IP dari neraca beban, bukan alamat IP pengguna. Dalam konfigurasi ini, kami menyarankan Anda menggunakan alamat IP terakhir dalam X-Forwarded-For Header HTTP. Header ini biasanya berisi lebih dari satu alamat IP, sebagian besar untuk proksik atau timbangan beban. Alamat IP terakhir dalam daftar adalah yang paling mungkin dikaitkan dengan lokasi geografis pengguna.

Jika server web Anda tidak terhubung ke neraca beban, sebaiknya gunakan variabel server web, bukan X-Forwarded-For untuk menghindari spoofing alamat IP.

Gunakan enkripsi tingkat lapangan untuk membantu melindungi data sensitif

Dengan Amazon CloudFront, Anda dapat menerapkan koneksi end-to-end yang aman ke server asal dengan menggunakan HTTPS. Field-level enkripsi menambahkan lapisan keamanan tambahan yang memungkinkan Anda melindungi data tertentu selama pemrosesan sistem sehingga hanya aplikasi tertentu yang dapat melihatnya.

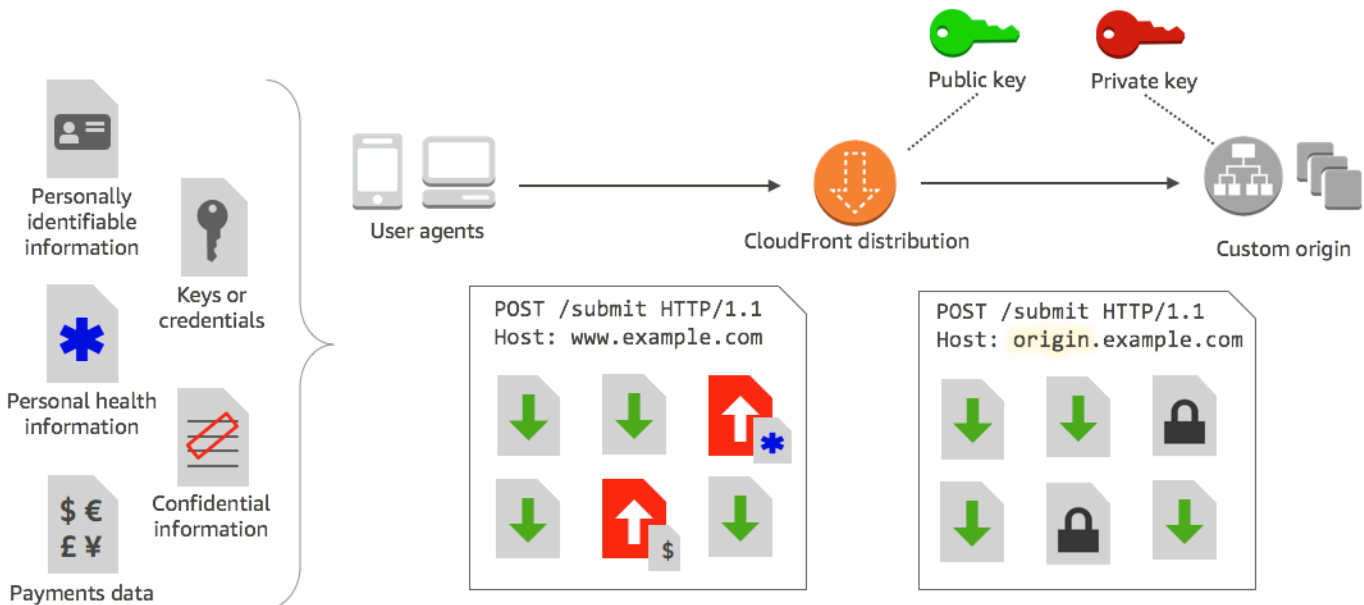
Field-level enkripsi memungkinkan Anda untuk memungkinkan pengguna Anda untuk mengunggah informasi sensitif dengan aman ke server web Anda. Informasi sensitif yang diberikan oleh pengguna Anda dienkripsi di edge, dekat dengan pengguna, dan tetap dienkripsi di seluruh tumpukan aplikasi Anda. Enkripsi ini memastikan bahwa hanya aplikasi yang memerlukan data—dan memiliki kredensial untuk mendekripsinya—dapat melakukannya.

Untuk menggunakan enkripsi tingkat bidang, saat Anda mengonfigurasi CloudFront distribusi, tentukan kumpulan bidang dalam permintaan POST yang ingin dienkripsi, dan kunci publik yang akan digunakan untuk mengenkripsi mereka. Anda dapat mengenkripsi hingga 10 kolom data dalam permintaan. (Anda tidak dapat mengenkripsi semua data dalam permintaan dengan enkripsi tingkat lapangan; Anda harus menentukan bidang individu untuk mengenkripsi.)

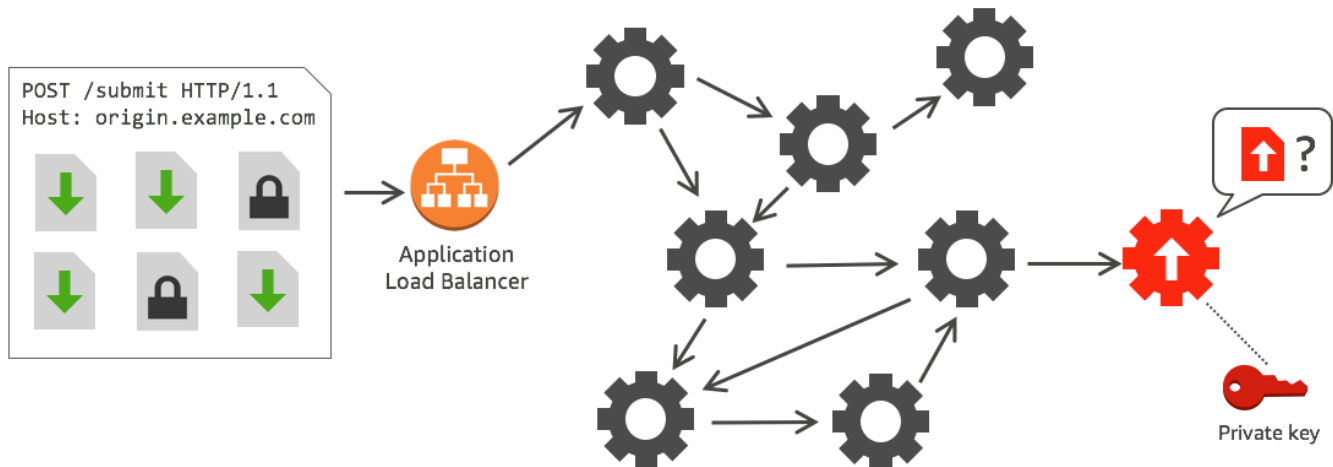
Ketika permintaan HTTPS dengan enkripsi tingkat lapangan diteruskan ke asal, dan permintaan diarahkan ke seluruh aplikasi atau subsistem asal Anda, data sensitif masih dienkripsi, sehingga mengurangi risiko pelanggaran data atau kehilangan data data sensitif yang tidak disengaja. Komponen yang membutuhkan akses ke data sensitif untuk alasan bisnis, seperti sistem pemrosesan pembayaran yang memerlukan akses ke nomor kredit, dapat menggunakan kunci pribadi yang sesuai untuk mendekripsi dan mengakses data.

Note

Untuk menggunakan enkripsi tingkat-lapangan, asal Anda harus mendukung pengkodean yang disusun (chunked encoding).



CloudFront enkripsi tingkat lapangan menggunakan enkripsi asimetris, juga dikenal sebagai enkripsi kunci publik. Anda memberikan kunci publik CloudFront, dan semua data sensitif yang Anda tentukan dienkripsi secara otomatis. Kunci yang Anda berikan CloudFront tidak dapat digunakan untuk mendekripsi nilai terenkripsi; hanya kunci pribadi Anda yang dapat melakukannya.



Topik

- [Ikhtisar enkripsi tingkat lapangan](#)
- [Siapkan enkripsi tingkat lapangan](#)
- [Dekripsi bidang data di tempat asal Anda](#)

Ikhtisar enkripsi tingkat lapangan

Langkah-langkah berikut memberikan ikhtisar pengaturan enkripsi tingkat lapangan. Untuk langkah spesifik, lihat [Siapkan enkripsi tingkat lapangan](#).

1. Dapatkan public key-private key pair. Anda harus mendapatkan dan menambahkan kunci publik sebelum memulai pengaturan enkripsi tingkat lapangan di CloudFront.
2. Buat profil enkripsi tingkat lapangan. Field-level profil enkripsi, yang Anda buat CloudFront, menentukan bidang yang ingin Anda enkripsi.
3. Buat konfigurasi enkripsi tingkat lapangan. Konfigurasi menentukan profil yang akan digunakan, berdasarkan jenis permintaan atau argumen kueri, untuk mengenkripsi kolom data spesifik. Anda juga dapat memilih opsi perilaku permintaan-penerusan yang Anda inginkan untuk skenario yang berbeda. Misalnya, Anda dapat mengatur perilaku saat nama profil yang ditentukan oleh argumen kueri di URL permintaan tidak ada CloudFront.
4. Tautan ke perilaku cache. Menautkan konfigurasi ke perilaku cache untuk distribusi, untuk menentukan kapan CloudFront harus mengenkripsi data.

Siapkan enkripsi tingkat lapangan

Ikuti langkah-langkah ini untuk mulai menggunakan enkripsi tingkat lapangan. Untuk mempelajari tentang kuota (sebelumnya dikenal sebagai batas) pada enkripsi tingkat lapangan, lihat [Kuota](#).

- [Langkah 1: Buat key pair RSA](#)
- [Langkah 2: Tambahkan kunci publik Anda CloudFront](#)
- [Langkah 3: Buat profil untuk enkripsi tingkat lapangan](#)
- [Langkah 4: Buat konfigurasi](#)
- [Langkah 5: Tambahkan konfigurasi ke perilaku cache](#)

Langkah 1: Buat key pair RSA

Untuk memulai, Anda harus membuat pasangan kunci RSA yang mencakup kunci publik dan kunci pribadi. Kunci publik memungkinkan CloudFront untuk mengenkripsi data, dan kunci pribadi memungkinkan komponen di asal Anda untuk mendekripsi bidang yang telah dienkripsi. Anda dapat menggunakan OpenSSL atau alat lain untuk membuat pasangan kunci. Ukuran kunci harus 2048 bit.

Misalnya, jika Anda menggunakan OpenSSL, Anda dapat menggunakan perintah berikut untuk membuat pasangan kunci dengan panjang 2048 bit dan menyimpannya dalam file `private_key.pem`:

```
openssl genrsa -out private_key.pem 2048
```

Berkas yang dihasilkan berisi baik publik maupun kunci pribadi. Untuk mengekstrak kunci publik dari file tersebut, jalankan perintah berikut:

```
openssl rsa -pubout -in private_key.pem -out public_key.pem
```

File kunci publik (`public_key.pem`) berisi nilai kunci terkode yang Anda tempelkan pada langkah berikut.

Langkah 2: Tambahkan kunci publik Anda CloudFront

Setelah Anda mendapatkan key pair RSA Anda, tambahkan kunci publik Anda ke CloudFront.

Untuk menambahkan kunci publik Anda ke CloudFront (konsol)

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Kunci publik.
3. Pilih Tambahkan kunci publik.
4. Untuk Nama kunci, ketikkan nama unik untuk kunci. Nama tidak boleh memiliki spasi dan hanya dapat menyertakan karakter alfanumerik, garis bawah (_), dan tanda hubung (-). Jumlah karakter maksimum adalah 128.
5. Untuk Nilai utama, tempelkan nilai utama yang disandikan untuk kunci publik Anda, termasuk -----BEGIN PUBLIC KEY----- dan -----END PUBLIC KEY----- yang tepat.
6. Untuk Komentar, tambahkan komentar opsional. Misalnya, Anda dapat menyertakan tanggal kedaluwarsa kunci publik.
7. Pilih Tambahkan.

Anda dapat menambahkan lebih banyak kunci untuk digunakan CloudFront dengan mengulangi langkah-langkah dalam prosedur.

Langkah 3: Buat profil untuk enkripsi tingkat lapangan

Setelah Anda menambahkan setidaknya satu kunci publik CloudFront, buat profil yang memberi tahu bidang CloudFront mana yang akan dienkripsi.

Untuk membuat profil enkripsi tingkat lapangan (konsole)

1. Di panel navigasi, pilih Field-level enkripsi.
2. Pilih Buat profil.
3. Isi kolom berikut:

Nama profil

Ketikkan nama unik untuk profil. Nama tidak boleh memiliki spasi dan hanya dapat menyertakan karakter alfanumerik, garis bawah (_), dan tanda hubung (-). Jumlah karakter maksimum adalah 128.

Nama kunci publik

Dalam daftar drop-down, pilih nama kunci publik yang Anda tambahkan CloudFront pada langkah 2. CloudFront menggunakan kunci untuk mengenkripsi bidang yang Anda tentukan di profil ini.

Nama penyedia

Ketikkan frasa untuk membantu mengidentifikasi kunci, seperti penyedia tempat Anda mendapatkan pasangan kunci. Informasi ini, bersama dengan kunci pribadi, diperlukan ketika aplikasi mendekripsi bidang data. Nama penyedia tidak boleh memiliki spasi dan hanya dapat menyertakan karakter alfanumerik, usus besar (:), garis bawah (_), dan tanda hubung (-). Jumlah karakter maksimum adalah 128.

Pola nama bidang agar cocok

Ketik nama bidang data, atau pola yang mengidentifikasi nama bidang data dalam permintaan, yang Anda inginkan CloudFront untuk mengenkripsi. Pilih opsi + untuk menambahkan semua kolom yang ingin Anda enkripsi dengan kunci ini.

Untuk pola nama bidang, Anda dapat mengetikkan seluruh nama bidang data, seperti DateOfBirth, atau hanya bagian pertama dari nama dengan karakter wildcard (*), CreditCard seperti*. Pola nama bidang hanya boleh menyertakan karakter alfanumerik, tanda kurung persegi ([dan]), periode (.), garis bawah (_), dan tanda hubung (-), selain karakter wildcard opsional (*).

Pastikan Anda tidak menggunakan karakter yang tumpang tindih untuk pola nama bidang yang berbeda. Misalnya, jika Anda memiliki pola nama kolom ABC*, Anda tidak dapat menambahkan pola nama bidang lain yang merupakan AB*. Selain itu, nama bidang bersifat peka huruf besar dan jumlah maksimal karakter yang dapat Anda gunakan adalah 128.

Komentar

(Opsional) Ketik komentar tentang profil ini. Jumlah maksimal karakter yang dapat Anda gunakan adalah 128.

4. Setelah mengisi kolom, pilih Buat profil.
5. Jika Anda ingin menambahkan profil lagi, pilih Tambahkan profil.

Langkah 4: Buat konfigurasi

Setelah Anda membuat satu atau beberapa profil enkripsi tingkat lapangan, buat konfigurasi yang menentukan jenis konten permintaan yang menyertakan data yang akan dienkrpsi, profil yang akan digunakan untuk enkripsi, dan opsi lain yang menentukan cara Anda ingin menangani enkripsi. CloudFront

Misalnya, ketika tidak CloudFront dapat mengenkripsi data, Anda dapat menentukan apakah CloudFront harus memblokir atau meneruskan permintaan ke asal Anda dalam skenario berikut:

- Jika jenis konten permintaan tidak ada dalam konfigurasi — Jika Anda belum menambahkan tipe konten ke konfigurasi, Anda dapat menentukan apakah CloudFront harus meneruskan permintaan dengan tipe konten tersebut ke asal tanpa mengenkripsi bidang data, atau memblokir permintaan dan mengembalikan kesalahan.

Note

Jika Anda menambahkan jenis konten ke konfigurasi tetapi belum menentukan profil untuk digunakan dengan tipe tersebut, CloudFront selalu teruskan permintaan dengan jenis konten tersebut ke asal.

- Bila nama profil yang disediakan dalam argumen kueri tidak diketahui - Bila Anda menentukan argumen `fle-profile` kueri dengan nama profil yang tidak ada untuk distribusi Anda, Anda dapat menentukan apakah CloudFront harus mengirim permintaan ke asal tanpa mengenkripsi bidang data, atau memblokir permintaan dan mengembalikan kesalahan.

Dalam konfigurasi, Anda juga dapat menentukan apakah memberikan profil sebagai argumen kueri dalam URL membatalkan profil yang telah dipetakan ke jenis konten untuk kueri tersebut. Secara default, CloudFront gunakan profil yang telah Anda petakan ke jenis konten, jika Anda menentukannya. Ini memungkinkan Anda memiliki profil yang digunakan secara default tetapi memutuskan untuk permintaan tertentu yang ingin Anda gunakan profil berbeda.

Jadi, misalnya, Anda dapat menentukan (dalam konfigurasi Anda) **SampleProfile** sebagai profil argumen kueri untuk digunakan. Kemudian Anda dapat menggunakan URL `https://d1234.cloudfront.net?fle-profile=SampleProfile` alih-alih `https://d1234.cloudfront.net`, untuk CloudFront digunakan **SampleProfile** untuk permintaan ini, alih-alih profil yang akan Anda siapkan untuk jenis konten permintaan.

Anda dapat membuat hingga 10 konfigurasi untuk satu akun, lalu mengaitkan salah satu konfigurasi ke perilaku cache dari setiap distribusi untuk akun.

Untuk membuat konfigurasi enkripsi tingkat lapangan (konsole)

1. Pada halaman Field-level enkripsi, pilih Buat konfigurasi.

Catatan: Jika Anda belum membuat setidaknya satu profil, Anda tidak akan melihat opsi untuk membuat konfigurasi.

2. Isi kolom berikut untuk menentukan profil yang akan digunakan. (Beberapa kolom tidak dapat diubah.)

Jenis konten (tidak dapat diubah)

Jenis konten diatur ke `application/x-www-form-urlencoded` dan tidak dapat diubah.

ID profil default (opsional)

Di daftar menurun, pilih profil yang ingin Anda petakan ke jenis konten di Jenis konten bidang.

Format konten (tidak dapat diubah)

Format konten diatur ke `URLencoded` dan tidak dapat diubah.

3. Jika Anda ingin mengubah perilaku CloudFront default untuk opsi berikut, pilih kotak centang yang sesuai.

Teruskan permintaan ke asal ketika jenis konten permintaan tidak dikonfigurasi

Pilih kotak centang jika Anda ingin mengizinkan permintaan untuk pergi ke asal Anda jika Anda belum menentukan profil yang akan digunakan untuk jenis permintaan konten tersebut.

Ubah profil untuk jenis konten dengan argumen kueri yang diberikan

Centang kotak jika Anda ingin mengizinkan profil yang diberikan dalam argumen kueri menimpa profil yang telah Anda tentukan untuk jenis konten.

4. Jika Anda memilih kotak centang untuk mengizinkan argumen kueri menimpa profil default, Anda harus melengkapi kolom tambahan berikut untuk konfigurasi. Anda dapat membuat hingga lima pemetaan argumen kueri ini untuk digunakan dengan kueri.

Argumen pertanyaan

Ketik nilai yang ingin Anda masukkan dalam URL untuk argumen kueri `fle-profile`. Nilai ini memberitahu CloudFront untuk menggunakan ID profil (yang Anda tentukan di bidang berikutnya) yang terkait dengan argumen kueri ini untuk enkripsi tingkat bidang untuk kueri ini.

Jumlah maksimal karakter yang dapat Anda gunakan adalah 128. Nilai tidak boleh menyertakan spasi, dan hanya boleh menggunakan karakter alfanumerik atau karakter berikut: tanda hubung (-), periode (.), garis bawah (_), tanda bintang (*), tanda tambah (+), persen (%).

ID Profil

Di daftar menurun, pilih profil yang ingin Anda kaitkan dengan nilai yang Anda masukkan Argumen pertanyaan.

Teruskan permintaan ke asal ketika profil yang ditentukan dalam argumen kueri tidak ada

Pilih kotak centang jika Anda ingin mengizinkan permintaan untuk pergi ke asal Anda jika profil yang ditentukan dalam argumen kueri tidak didefinisikan dalam CloudFront.

Langkah 5: Tambahkan konfigurasi ke perilaku cache

Untuk menggunakan enkripsi tingkat lapangan, tautkan konfigurasi ke perilaku cache untuk distribusi dengan menambahkan ID konfigurasi sebagai nilai untuk distribusi Anda.

Important

Untuk menautkan konfigurasi enkripsi tingkat lapangan ke perilaku cache, distribusi harus dikonfigurasi untuk selalu menggunakan HTTPS, dan untuk menerima HTTP POST dan PUT permintaan dari penampil. Yaitu, hal-hal berikut harus benar:

- Perilaku singgahan Kebijakan Protokol Penampil harus diatur menjadi Arahkan ulang HTTP ke HTTPS atau HTTPS Saja. (Dalam CloudFormation atau CloudFront API, `ViewerProtocolPolicy` harus disetel ke `redirect-to-https` atau `https-only`.)
- Perilaku singgahan Metode HTTP yang Diizinkan harus ditetapkan ke DAPATKAN, KEPALA, OPSI, PUT, POST, PATCH, DELETE. (Dalam CloudFormation atau CloudFront

API, AllowedMethods harus disetel ke GET, HEAD, OPTIONS, PUT, POST, PATCH, DELETE. Ini dapat ditentukan dalam urutan apa pun.)

- Pengaturan asal Kebijakan Protokol Asal harus diatur menjadi Penampil Kecocokan atau HTTPS Saja. (Dalam CloudFormation atau CloudFront API, OriginProtocolPolicy harus disetel ke match-viewer atau https-only.)

Untuk informasi selengkapnya, lihat [Semua referensi pengaturan distribusi](#).

Dekripsi bidang data di tempat asal Anda

CloudFront mengenkripsi bidang data dengan menggunakan file. [AWS Encryption SDK](#) Data tetap terenkripsi di seluruh tumpukan aplikasi Anda dan hanya dapat diakses oleh aplikasi yang memiliki kredensial untuk mendekripsinya.

Setelah enkripsi, ciphertext adalah dasar64 yang dikodekan. Ketika aplikasi Anda mendekripsi teks pada awalnya, aplikasi harus menguraikan ciphertext, lalu menggunakan AWS Encryption SDK untuk mendekripsi data.

Contoh kode berikut mengcitarkan bagaimana aplikasi dapat mendekripsi data di tempat asal Anda. Perhatikan hal-hal berikut:

- Untuk menyederhanakan contoh, sampel ini memuat kunci publik dan privat (dalam format DER) dari file di direktori kerja. Dalam praktiknya, Anda akan menyimpan kunci pribadi di lokasi offline yang aman, seperti modul keamanan perangkat keras offline, dan mendistribusikan kunci publik ke tim pengembangan Anda.
- CloudFront menggunakan informasi spesifik saat mengenkripsi data, dan set parameter yang sama harus digunakan di tempat asal untuk mendekripsi data. Parameter yang CloudFront digunakan saat menginisialisasi MasterKey meliputi yang berikut:
 - PROVIDER_NAME: Anda menentukan nilai ini saat membuat profil enkripsi tingkat lapangan. Gunakan nilai yang sama di sini.
 - KEY_NAME: Anda membuat nama untuk kunci publik Anda ketika Anda mengunggahnya CloudFront, dan kemudian menentukan nama kunci di profil. Gunakan nilai yang sama di sini.
 - ALGORITHM: CloudFront digunakan RSA/ECB/OAEPWithSHA-256AndMGF1Padding sebagai algoritma untuk mengenkripsi, jadi Anda harus menggunakan algoritma yang sama untuk mendekripsi data.

- Jika Anda menjalankan program sampel berikut dengan ciphertext sebagai input, data yang didekripsi adalah output untuk konsol Anda. Untuk informasi selengkapnya, lihat [Kode Contoh Java](#) di SDK AWS Enkripsi.

Kode sampel

```
import java.nio.file.Files;
import java.nio.file.Paths;
import java.security.KeyFactory;
import java.security.PrivateKey;
import java.security.PublicKey;
import java.security.spec.PKCS8EncodedKeySpec;
import java.security.spec.X509EncodedKeySpec;

import org.apache.commons.codec.binary.Base64;

import com.amazonaws.encryptionsdk.AwsCrypto;
import com.amazonaws.encryptionsdk.CryptoResult;
import com.amazonaws.encryptionsdk.jce.JceMasterKey;

/**
 * Sample example of decrypting data that has been encrypted by CloudFront field-level
 * encryption.
 */
public class DecryptExample {

    private static final String PRIVATE_KEY_FILENAME = "private_key.der";
    private static final String PUBLIC_KEY_FILENAME = "public_key.der";
    private static PublicKey publicKey;
    private static PrivateKey privateKey;

    // CloudFront uses the following values to encrypt data, and your origin must use
    // same values to decrypt it.
    // In your own code, for PROVIDER_NAME, use the provider name that you specified
    // when you created your field-level
    // encryption profile. This sample uses 'DEMO' for the value.
    private static final String PROVIDER_NAME = "DEMO";
    // In your own code, use the key name that you specified when you added your public
    // key to CloudFront. This sample
    // uses 'DEMOKEY' for the key name.
    private static final String KEY_NAME = "DEMOKEY";
```

```
// CloudFront uses this algorithm when encrypting data.
private static final String ALGORITHM = "RSA/ECB/OAEPWithSHA-256AndMGF1Padding";

public static void main(final String[] args) throws Exception {

    final String dataToDecrypt = args[0];

    // This sample uses files to get public and private keys.
    // In practice, you should distribute the public key and save the private key
in secure storage.
    populateKeyPair();

    System.out.println(decrypt(debase64(dataToDecrypt)));
}

private static String decrypt(final byte[] bytesToDecrypt) throws Exception {
    // You can decrypt the stream only by using the private key.

    // 1. Instantiate the SDK
    final AwsCrypto crypto = new AwsCrypto();

    // 2. Instantiate a JCE master key
    final JceMasterKey masterKey = JceMasterKey.getInstance(
        publicKey,
        privateKey,
        PROVIDER_NAME,
        KEY_NAME,
        ALGORITHM);

    // 3. Decrypt the data
    final CryptoResult <byte[], ? > result = crypto.decryptData(masterKey,
bytesToDecrypt);
    return new String(result.getResult());
}

// Function to decode base64 cipher text.
private static byte[] debase64(final String value) {
    return Base64.decodeBase64(value.getBytes());
}

private static void populateKeyPair() throws Exception {
    final byte[] PublicKeyBytes =
Files.readAllBytes(Paths.get(PUBLIC_KEY_FILENAME));
```

```
        final byte[] privateKeyBytes =  
Files.readAllBytes(Paths.get(PRIVATE_KEY_FILENAME));  
        publicKey = KeyFactory.getInstance("RSA").generatePublic(new  
X509EncodedKeySpec(PublicKeyBytes));  
        privateKey = KeyFactory.getInstance("RSA").generatePrivate(new  
PKCS8EncodedKeySpec(privateKeyBytes));  
    }  
}
```

Video sesuai permintaan dan video streaming langsung dengan CloudFront

Anda dapat menggunakan CloudFront untuk mengirimkan video on demand (VOD) atau video streaming langsung dengan menggunakan asal HTTP apa pun. Salah satu cara Anda dapat mengatur alur kerja video di cloud adalah dengan menggunakan CloudFront bersama dengan [Layanan AWS Media](#).

Topik

- [Tentang streaming video](#)
- [Mengirimkan video sesuai permintaan dengan CloudFront](#)
- [Memberikan streaming video dengan CloudFront dan Layanan AWS Media](#)
- [Ketahanan sadar kualitas media](#)

Tentang streaming video

Anda harus menggunakan encoder untuk mengemas konten video sebelum CloudFront dapat mendistribusikan konten. Proses pengemasan membuat segmen yang berisi konten audio, video, dan keterangan Anda. Ini juga menghasilkan file manifes, yang menjelaskan secara spesifik segmen mana yang akan diputar dan kapan. Format paket umum adalah MPEG DASH, Apple HLS, Microsoft Smooth Streaming, dan CMAF.

Streaming VOD

Untuk streaming VOD, konten video Anda disimpan di server dan pemirsa dapat menontonnya kapan saja. Untuk membuat aset yang dapat dialirkan pemirsa, gunakan encoder, seperti [AWS Elemental MediaConvert](#), untuk memformat dan mengemas file media Anda.

Setelah video dikemas ke dalam format yang tepat, Anda dapat menyimpannya di server atau di bucket Amazon S3, lalu mengirimkannya CloudFront sesuai permintaan pemirsa.

Streaming video langsung

Untuk streaming video langsung, konten video Anda dialirkan secara real time saat acara langsung berlangsung, atau diatur sebagai saluran langsung 24x7. Untuk membuat output langsung untuk siaran dan pengiriman streaming, gunakan encoder seperti AWS Elemental MediaLive, untuk mengompres video dan memformatnya untuk melihat perangkat.

Setelah video Anda dikodekan, Anda dapat menyimpannya AWS Elemental MediaStore atau mengubahnya menjadi format pengiriman yang berbeda dengan menggunakan AWS Elemental MediaPackage. Gunakan salah satu dari asal-usul ini untuk mengatur CloudFront distribusi untuk mengirimkan konten. Untuk langkah dan panduan spesifik dalam membuat distribusi yang bekerja bersama dengan layanan ini, lihat [Sajikan video dengan menggunakan AWS Elemental MediaStore sebagai asal](#) dan [Sajikan video langsung yang diformat dengan AWS Elemental MediaPackage](#).

Wowza dan Unified Streaming juga menyediakan alat yang dapat Anda gunakan untuk streaming video. CloudFront Untuk informasi lebih lanjut tentang menggunakan Wowza dengan CloudFront, lihat [Membawa lisensi Wowza Streaming Engine Anda ke streaming HTTP CloudFront langsung](#) di situs web dokumentasi Wowza. Untuk informasi tentang penggunaan Streaming Terpadu dengan CloudFront streaming VOD, lihat [CloudFront](#) di situs web dokumentasi Streaming Terpadu.

Mengirimkan video sesuai permintaan dengan CloudFront

Untuk mengirimkan streaming video on demand (VOD) CloudFront, gunakan layanan berikut:

- Amazon S3 untuk menyimpan konten dalam format aslinya dan untuk menyimpan video yang di-transcode.
- Encoder (seperti AWS Elemental MediaConvert) untuk mentranskode video ke dalam format streaming.
- CloudFront untuk mengirimkan video yang ditranskode ke pemirsa. Untuk Microsoft Smooth Streaming, lihat [Konfigurasi video sesuai permintaan untuk Microsoft Smooth Streaming](#).

Untuk membuat solusi VOD dengan CloudFront

1. Unggah konten Anda ke keranjang Amazon S3. Untuk mempelajari selengkapnya tentang bekerja dengan Amazon S3, lihat [Panduan Pengguna Layanan Penyimpanan Sederhana Amazon](#).
2. Transkode konten Anda dengan menggunakan MediaConvert pekerjaan. Tugas mengubah video menjadi format yang diperlukan oleh pemain yang digunakan oleh pemirsa Anda. Anda juga dapat menggunakan pekerjaan untuk membuat aset yang berbeda-beda dalam resolusi dan bitrate. Aset ini digunakan untuk streaming bitrate adaptif (ABR), yang menyesuaikan kualitas tampilan tergantung pada bandwidth pemirsa yang tersedia. MediaConvert menyimpan video yang ditranskode dalam ember S3.

3. Kirimkan konten Anda yang dikonversi dengan menggunakan CloudFront distribusi. Penampil dapat menonton konten di perangkat apa pun, kapan saja.

Konfigurasi video sesuai permintaan untuk Microsoft Smooth Streaming

Anda memiliki opsi berikut untuk digunakan CloudFront untuk mendistribusikan konten video on demand (VOD) yang telah Anda transkode ke dalam format Microsoft Smooth Streaming:

- Tentukan server web yang menjalankan Microsoft IIS dan mendukung Streaming Halus sebagai asal distribusi Anda.
- Aktifkan Smooth Streaming dalam perilaku cache CloudFront distribusi. Karena Anda dapat menggunakan beberapa perilaku cache dalam distribusi, Anda dapat menggunakan satu distribusi untuk file media Streaming Halus serta konten lainnya.

Important

Jika Anda menentukan server web yang menjalankan Microsoft IIS sebagai asal Anda, jangan aktifkan Smooth Streaming dalam perilaku cache CloudFront distribusi Anda. CloudFront tidak dapat menggunakan server Microsoft IIS sebagai asal jika Anda mengaktifkan Smooth Streaming sebagai perilaku cache.

Jika Anda mengaktifkan Streaming Mulus dalam perilaku cache (yaitu, Anda tidak memiliki server yang menjalankan Microsoft IIS), perhatikan hal berikut:

- Anda masih dapat mendistribusikan konten lain menggunakan perilaku cache yang sama jika konten sesuai dengan nilai Pola Jalan untuk perilaku singgahan itu.
- CloudFront dapat menggunakan bucket Amazon S3 atau custom origin untuk file media Smooth Streaming. CloudFront tidak dapat menggunakan Microsoft IIS Server sebagai asal jika Anda mengaktifkan Smooth Streaming untuk perilaku cache.
- Anda tidak bisa memvalidasi file media dalam format Streaming Halus. Jika Anda ingin memperbarui file sebelum kedaluwarsa, Anda harus mengganti namanya. Untuk informasi selengkapnya, lihat [Menambahkan, menghapus, atau mengganti konten yang CloudFront mendistribusikan](#).

Untuk informasi tentang klien Smooth Streaming, lihat [Smooth Streaming](#) di situs web dokumentasi Microsoft.

Untuk digunakan CloudFront untuk mendistribusikan file Smooth Streaming ketika server web Microsoft IIS bukan asalnya

1. Transkode file media Anda ke dalam format Smooth Streaming terfragmentasi MP4.
2. Lakukan salah satu tindakan berikut:
 - Jika Anda menggunakan CloudFront konsol: Saat membuat atau memperbarui distribusi, aktifkan Smooth Streaming di satu atau beberapa perilaku cache distribusi.
 - Jika Anda menggunakan CloudFront API: Tambahkan SmoothStreaming elemen ke tipe `DistributionConfig` kompleks untuk satu atau beberapa perilaku cache distribusi.
3. Unggah file Smooth Streaming ke asal Anda.
4. Buat file `clientaccesspolicy.xml` atau `crossdomainpolicy.xml`, dan tambahkan ke lokasi yang dapat diakses dari akar distribusi Anda, misalnya, `https://d111111abcdef8.cloudfront.net/clientaccesspolicy.xml`. Berikut ini adalah contoh kebijakan.

```
<?xml version="1.0" encoding="utf-8"?>
<access-policy>
  <cross-domain-access>
    <policy>
      <allow-from http-request-headers="*">
        <domain uri="*" />
      </allow-from>
      <grant-to>
        <resource path="/" include-subpaths="true" />
      </grant-to>
    </policy>
  </cross-domain-access>
</access-policy>
```

Untuk informasi lebih lanjut, lihat [Menyediakan Layanan di Seluruh Batas Domain](#) di situs web Microsoft Developer Network.

5. Untuk tautan di aplikasi Anda (misalnya, pemutar media), tentukan URL untuk file media dengan format berikut:

`https://d111111abcdef8.cloudfront.net/video/presentation.ism/Manifest`

Memberikan streaming video dengan CloudFront dan Layanan AWS Media

Untuk menggunakan Layanan AWS Media CloudFront untuk mengirimkan konten langsung ke audiens global, lihat panduan berikut.

Gunakan [AWS Elemental MediaLive](#) untuk menyandikan streaming video langsung secara real time. Untuk menyandikan aliran video besar, MediaLive kompres ke dalam versi yang lebih kecil (encode) yang dapat didistribusikan ke pemirsa Anda.

Setelah Anda mengompresi streaming video langsung, Anda dapat menggunakan salah satu dari dua opsi utama berikut untuk menyiapkan dan menyajikan konten:

- Konversikan konten Anda menjadi format yang diperlukan, lalu sajikan — Jika Anda memerlukan konten dalam berbagai format, gunakan [AWS Elemental MediaPackage](#) untuk mengemas konten untuk berbagai jenis perangkat. Saat Anda mengemas konten, Anda juga dapat menerapkan fitur ekstra dan menambahkan manajemen hak digital (DRM) untuk mencegah penggunaan konten yang tidak sah. Untuk step-by-step petunjuk penggunaan CloudFront untuk menyajikan konten yang MediaPackage diformat, lihat [Sajikan video langsung yang diformat dengan AWS Elemental MediaPackage](#).
- Simpan dan sajikan konten Anda menggunakan asal yang dapat diskalakan — Jika konten MediaLive yang disandikan dalam format yang diperlukan oleh semua perangkat yang digunakan pemirsa Anda, gunakan sumber yang sangat skalabel seperti [AWS Elemental MediaStore](#) untuk menyajikan konten. Untuk step-by-step petunjuk penggunaan CloudFront untuk menyajikan konten yang disimpan dalam MediaStore wadah, lihat [Sajikan video dengan menggunakan AWS Elemental MediaStore sebagai asal](#).

Setelah Anda mengatur asal usul dengan menggunakan salah satu opsi ini, Anda dapat mendistribusikan video streaming langsung ke penampil dengan menggunakan CloudFront.

Tip

Anda dapat mempelajari tentang AWS solusi yang secara otomatis menyebarkan layanan untuk membangun pengalaman menonton real-time yang sangat tersedia. Untuk melihat langkah-langkah untuk menerapkan solusi ini secara otomatis, lihat [Penyebaran Otomatis Streaming Langsung](#).

Topik

- [Sajikan video dengan menggunakan AWS Elemental MediaStore sebagai asal](#)
- [Sajikan video langsung yang diformat dengan AWS Elemental MediaPackage](#)
- [Sajikan video-on-demand konten dengan AWS Elemental MediaPackage](#)

Sajikan video dengan menggunakan AWS Elemental MediaStore sebagai asal

Jika Anda memiliki video yang disimpan dalam [AWS Elemental MediaStore](#) wadah, Anda dapat membuat CloudFront distribusi untuk menyajikan konten.

Untuk memulai, Anda memberikan CloudFront akses ke MediaStore wadah Anda. Kemudian Anda membuat CloudFront distribusi dan mengkonfigurasinya agar berfungsi MediaStore.

Untuk menyajikan konten dari AWS Elemental MediaStore wadah

1. Ikuti prosedur di [Mengizinkan Amazon CloudFront mengakses AWS Elemental MediaStore penampung Anda](#), lalu kembali ke langkah-langkah ini untuk membuat distribusi Anda.
2. Buat distribusi dengan pengaturan berikut:
 - a. Domain asal — Titik akhir data yang ditetapkan ke MediaStore wadah Anda. Dari daftar dropdown, pilih MediaStore wadah untuk video langsung Anda.
 - b. Jalur asal - Struktur folder dalam MediaStore wadah tempat objek Anda disimpan. Untuk informasi selengkapnya, lihat [the section called “Jalur asal”](#).
 - c. Tambahkan header kustom - Tambahkan nama header dan nilai-nilai jika Anda CloudFront ingin menambahkan header kustom ketika meneruskan permintaan ke asal Anda.
 - d. Kebijakan protokol penampil - Pilih Redirect HTTP ke HTTPS. Untuk informasi selengkapnya, lihat [the section called “Kebijakan protokol penampil”](#).
 - e. Kebijakan cache dan kebijakan permintaan Origin
 - Untuk kebijakan Cache, pilih Buat kebijakan, lalu buat kebijakan cache yang sesuai untuk kebutuhan caching dan durasi segmen Anda. Setelah membuat kebijakan, segarkan daftar kebijakan cache dan pilih kebijakan yang baru saja dibuat.
 - Untuk kebijakan permintaan Origin, pilih CORS- CustomOrigin dari daftar tarik-turun.

Untuk pengaturan lainnya, Anda dapat menetapkan nilai spesifik berdasarkan persyaratan teknis lain atau kebutuhan bisnis Anda. Untuk daftar semua opsi distribusi dan informasi tentang pengaturannya, lihat [the section called “Semua pengaturan distribusi”](#).

3. Untuk tautan dalam aplikasi Anda (misalnya, pemutar media), tentukan nama file media dalam format yang sama yang Anda gunakan untuk objek lain yang Anda distribusikan. CloudFront

Sajikan video langsung yang diformat dengan AWS Elemental MediaPackage

Jika Anda memformat streaming langsung dengan menggunakan AWS Elemental MediaPackage, Anda dapat membuat CloudFront distribusi dan mengonfigurasi perilaku cache untuk menyajikan streaming langsung. Proses berikut ini mengasumsikan bahwa Anda telah [membuat saluran](#) dan [titik akhir tambahan](#) untuk video langsung Anda menggunakan MediaPackage.

Untuk membuat CloudFront distribusi secara MediaPackage manual, ikuti langkah-langkah berikut:

Langkah 1: Buat dan konfigurasi CloudFront distribusi

Selesaikan prosedur berikut untuk menyiapkan CloudFront distribusi saluran video langsung yang Anda buat MediaPackage.

Untuk membuat distribusi untuk saluran video langsung Anda

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Pilih Buat Distribusi.
3. Pilih pengaturan untuk distribusi, termasuk berikut ini:

Domain asal

Asal tempat saluran video MediaPackage langsung dan titik akhir Anda berada. Pilih bidang teks, lalu dari daftar dropdown, pilih domain MediaPackage asal untuk video langsung Anda. Anda dapat memetakan satu domain ke beberapa titik akhir asal.

Jika Anda membuat domain asal menggunakan AWS akun lain, ketikkan nilai URL asal ke dalam bidang. Asal harus berupa URL HTTPS.

Misalnya, untuk titik akhir HLS seperti `https://3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com/out/v1/abc123/index.m3u8`, domain asal adalah `3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com`

Untuk informasi selengkapnya, lihat [the section called “Domain asal”](#).

Jalur asal

Jalur ke MediaPackage titik akhir dari mana konten disajikan.

Untuk informasi selengkapnya tentang cara kerja jalur asal, lihat [the section called “Jalur asal”](#).

Important

Jalur wildcard * diperlukan untuk merutekan suatu tempat dalam CloudFront distribusi. Untuk mencegah permintaan tidak cocok dengan jalur eksplisit dari perutean ke asal sebenarnya, buat asal “dummy” untuk jalur wildcard tersebut.

Example: Membuat asal “dummy”

Dalam contoh berikut, titik akhir `abc123` dan `def456` rute ke asal “nyata”, tetapi meminta rute konten video titik akhir lainnya `mediapackage.us-west-2.amazonaws.com` tanpa subdomain yang tepat, yang menghasilkan kesalahan HTTP. 404

MediaPackage titik akhir:

```
https://3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com/out/v1/abc123/index.m3u8
https://3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com/out/v1/def456/index.m3u8
```

CloudFront Asal A:

```
Domain: 3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com
Path: None
```

CloudFront Asal B:

```
Domain: mediapackage.us-west-2.amazonaws.com
Path: None
```

CloudFront perilaku cache:

1. Path: /out/v1/abc123/* forward to Origin A
2. Path: /out/v1/def456/* forward to Origin A
3. Path: * forward to Origin B

Untuk pengaturan distribusi lainnya, tetapkan nilai spesifik berdasarkan persyaratan teknis lain atau kebutuhan bisnis Anda. Untuk daftar semua opsi distribusi dan informasi tentang pengaturannya, lihat [the section called “Semua pengaturan distribusi”](#).

Setelah Anda selesai memilih pengaturan distribusi lainnya, pilih Buat distribusi.

4. Pilih distribusi yang baru saja Anda buat, lalu pilih Behaviors.
5. Pilih perilaku cache default, lalu pilih Edit. Tentukan pengaturan perilaku cache yang benar untuk saluran yang Anda pilih untuk asal. Kemudian, Anda akan menambahkan satu atau beberapa pengaturan perilaku cache tambahan dan mengeditnya.
6. Buka [halaman CloudFront distribusi](#).
7. Tunggu hingga nilai kolom Terakhir yang dimodifikasi untuk distribusi Anda telah berubah dari Deploying ke tanggal dan waktu, yang menunjukkan bahwa CloudFront telah membuat distribusi Anda.

Langkah 2: Tambahkan Origins untuk domain endpoint Anda MediaPackage

Ulangi langkah-langkah di sini untuk menambahkan setiap titik akhir MediaPackage saluran Anda ke distribusi Anda, dengan mengingat perlunya membuat asal “dummy”.

Untuk menambahkan titik akhir lain sebagai asal

1. Di CloudFront konsol, pilih distribusi yang Anda buat untuk saluran Anda.
2. Pilih Origins, lalu pilih Create origin.
3. Untuk domain Origin, di daftar tarik-turun, pilih MediaPackage titik akhir untuk saluran Anda.
4. Untuk pengaturan lainnya, tetapkan nilai berdasarkan persyaratan teknis lain atau kebutuhan bisnis Anda. Untuk informasi selengkapnya, lihat [the section called “Pengaturan asal”](#).
5. Pilih Buat asal.

Langkah 3 : Konfigurasi perilaku cache untuk semua titik akhir

Untuk setiap titik akhir, Anda harus mengonfigurasi perilaku cache untuk menambahkan pola jalur yang memenuhi permintaan rute dengan benar. Pola jalur yang Anda tentukan tergantung pada format video yang sedang Anda sajikan. Prosedur berikut mencakup informasi pola jalur yang dapat digunakan untuk format Apple HLS, CMAF, DASH, dan Microsoft Smooth Streaming.

Anda biasanya menyiapkan dua perilaku cache untuk setiap endpoint:

- Manifest induk, yang merupakan indeks untuk file Anda.
- Segmen, yang merupakan file konten video.

Untuk membuat perilaku cache untuk titik akhir

1. Di CloudFront konsol, pilih distribusi yang Anda buat untuk saluran Anda.
2. Pilih Perilaku, lalu pilih Buat perilaku.
3. Untuk pola Path, gunakan MediaPackage OriginEndpoint GUID tertentu sebagai awalan jalur.

Pola jalur

Untuk titik akhir HLS seperti `https://3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com/out/v1/abc123/index.m3u8`, buat dua perilaku cache berikut:

- Untuk manifest orang tua dan anak, gunakan `/out/v1/abc123/*.m3u8`.
- Untuk segmen konten, gunakan `/out/v1/abc123/*.ts`.

Untuk titik akhir CMAF seperti `https://3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com/out/v1/abc123/index.m3u8`, buat dua perilaku cache berikut:

- Untuk manifest orang tua dan anak, gunakan `/out/v1/abc123/*.m3u8`.
- Untuk segmen konten, gunakan `/out/v1/abc123/*.mp4`.

Untuk endpoint DASH seperti `https://3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com/out/v1/abc123/index.mpd`, buat dua perilaku cache berikut:

- Untuk manifest orang tua, gunakan `/out/v1/abc123/*.mpd`.
- Untuk segmen konten, gunakan `/out/v1/abc123/*.mp4`.

Untuk titik akhir Microsoft Smooth Streaming seperti `https://3ae97e9482b0d011.mediapackage.us-west-2.amazonaws.com/out/v1/abc123/index.ism`, hanya manifes yang disajikan, jadi Anda hanya membuat satu perilaku cache: `out/v1/abc123/index.ism/*`.

4. Untuk setiap perilaku cache, tentukan nilai untuk pengaturan berikut:

Kebijakan protokol penampil

Pilih Arahkan ulang HTTP ke HTTPS.

Kebijakan cache dan kebijakan permintaan asal

Untuk kebijakan Cache, pilih Buat kebijakan. Untuk kebijakan cache baru Anda, tentukan pengaturan berikut:

TTL Minimum

Atur hingga 5 detik atau kurang, untuk membantu mencegah agar konten basi tidak tertahan.

String pertanyaan

Untuk string Kueri (dalam pengaturan kunci Cache), pilih Sertakan string kueri yang ditentukan. Untuk Izinkan, tambahkan nilai berikut dengan mengetiknya lalu pilih Tambah item:

- Tambahkan `m` sebagai parameter string kueri yang CloudFront ingin Anda gunakan sebagai dasar untuk caching. MediaPackage Respons selalu menyertakan tag `?m=###` untuk menangkap waktu yang dimodifikasi dari titik akhir. Jika konten sudah di-cache dengan nilai berbeda untuk tag ini, CloudFront minta manifes baru alih-alih menyajikan versi cache.
- Jika Anda menggunakan fungsionalitas tampilan bergeser waktu MediaPackage, tentukan `start` dan `end` sebagai parameter string kueri tambahan pada perilaku cache untuk permintaan manifes (`*.m3u8,* .mpd`, dan `index.ism/*`). Dengan cara ini, konten disajikan khusus untuk periode waktu yang diminta dalam permintaan manifes. Untuk informasi selengkapnya tentang tampilan dan pemformatan parameter permintaan awal dan akhir konten yang diubah waktu, lihat [Tampilan bergeser waktu](#) di Panduan Pengguna.AWS Elemental MediaPackage
- Jika Anda menggunakan fitur pemfilteran manifes di MediaPackage, tentukan `aws.manifestfilter` sebagai parameter string kueri tambahan untuk kebijakan

cache yang Anda gunakan dengan perilaku cache untuk permintaan manifes (*.m3u8,*.mpd,danindex.ism/*). Ini mengonfigurasi distribusi Anda untuk meneruskan string `aws.manifestfilter` ke MediaPackage asal Anda, yang diperlukan agar fitur pemfilteran manifes berfungsi. Untuk informasi selengkapnya, lihat [Pemfilteran manifes](#) di Panduan AWS Elemental MediaPackage Pengguna.

- Jika Anda menggunakan HLS latensi rendah (LL-HLS), tentukan `_HLS_msn` dan `_HLS_part` sebagai parameter string kueri tambahan untuk kebijakan cache yang Anda gunakan dengan perilaku cache untuk permintaan manifes (). *.m3u8 Ini mengonfigurasi distribusi Anda untuk meneruskan string `_HLS_msn` dan `_HLS_part` ke MediaPackage asal Anda, yang diperlukan agar fitur permintaan daftar putar pemblokiran LL-HLS berfungsi.

5. Pilih Buat.
6. Setelah Anda membuat kebijakan cache, kembali ke alur kerja pembuatan perilaku cache. Segarkan daftar kebijakan cache, dan pilih kebijakan yang baru saja Anda buat.
7. Pilih Buat perilaku.
8. Jika titik akhir Anda bukan titik akhir Microsoft Smooth Streaming, ulangi langkah-langkah ini untuk membuat perilaku cache kedua.

Langkah 4: Aktifkan Otorisasi CDN berbasis header MediaPackage

Sebaiknya aktifkan Otorisasi MediaPackage CDN berbasis header antara MediaPackage titik akhir dan distribusi. CloudFront Untuk informasi selengkapnya, lihat [Mengaktifkan otorisasi CDN MediaPackage di AWS Elemental MediaPackage](#) Panduan Pengguna.

Langkah 5: Gunakan CloudFront untuk melayani saluran streaming langsung

Setelah Anda membuat distribusi, menambahkan asal, membuat perilaku cache, dan mengaktifkan otorisasi CDN berbasis header, Anda dapat menayangkan saluran streaming langsung menggunakan CloudFront. CloudFront merutekan permintaan dari pemirsa ke MediaPackage titik akhir yang benar berdasarkan pengaturan yang Anda konfigurasikan untuk perilaku cache.

Untuk tautan dalam aplikasi Anda (misalnya, pemutar media), tentukan URL untuk file media dalam format standar untuk CloudFront URLs. Untuk informasi selengkapnya, lihat [the section called "Kustomisasi URL file"](#).

Sajikan video-on-demand konten dengan AWS Elemental MediaPackage

Jika Anda membuat konten video-on-demand (VOD) dari AWS Elemental MediaPackage asal, Anda dapat membuat CloudFront distribusi dan mengonfigurasi perilaku cache yang dioptimalkan untuk menyajikan konten VOD kepada pemirsa. Proses berikut mengasumsikan bahwa Anda telah [membuat grup pengemasan](#) dengan [konfigurasi kemasan](#) dan [menelan aset](#). MediaPackage

Untuk membuat CloudFront distribusi secara MediaPackage manual, ikuti langkah-langkah berikut:

Langkah 1: Buat dan konfigurasi CloudFront distribusi

Selesaikan prosedur berikut untuk menyiapkan CloudFront distribusi untuk grup kemasan yang Anda buat MediaPackage.

Untuk membuat distribusi untuk konten VOD Anda

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Pilih Buat Distribusi.
3. Pilih pengaturan untuk distribusi, termasuk berikut ini:

Domain asal

Asal untuk grup MediaPackage kemasan Anda. Ketik nilai URL asal ke dalam bidang teks. Asal harus berupa URL HTTPS.

Misalnya, untuk titik akhir HLS

seperti `https://3ae97e9482b0d011.egress.mediapackage-vod.us-west-2.amazonaws.com/out/v1/abc123/def456/ghi789/index.m3u8`, domain asal adalah `3ae97e9482b0d011.egress.mediapackage-vod.us-west-2.amazonaws.com`

Untuk informasi selengkapnya, lihat [the section called “Domain asal”](#).

Jalur asal

Jalur dari mana konten disajikan.

Untuk informasi selengkapnya tentang cara kerja jalur asal, lihat [the section called “Jalur asal”](#).

⚠ Important

Jalur wildcard * diperlukan untuk merutekan suatu tempat dalam CloudFront distribusi. Untuk mencegah permintaan tidak cocok dengan jalur eksplisit dari perutean ke asal sebenarnya, buat asal “dummy” untuk jalur wildcard tersebut.

Example: Membuat asal “dummy”

Dalam contoh berikut, konfigurasi kemasan def456 dan 321xyz rute ke asal “nyata”, tetapi meminta rute konten video lainnya `mediapackage-vod.us-west-2.amazonaws.com` tanpa subdomain yang tepat, yang menghasilkan kesalahan HTTP404.

MediaPackage konten URLs untuk satu aset untuk grup pengemasan dengan dua konfigurasi kemasan:

```
https://3ae97e9482b0d011.egress.mediapackage-vod.us-west-2.amazonaws.com/out/v1/
abc123/def456/ghi789/index.m3u8
https://3ae97e9482b0d011.egress.mediapackage-vod.us-west-2.amazonaws.com/out/v1/
abc123/321xyz/654uvw/index.m3u8
```

CloudFront Asal A:

```
Domain: 3ae97e9482b0d011.egress.mediapackage-vod.us-west-2.amazonaws.com
Path: None
```

CloudFront Asal B:

```
Domain: mediapackage-vod.us-west-2.amazonaws.com
Path: None
```

CloudFront perilaku cache:

1. Path: `/out/v1/*/def456/*` forward to Origin A
2. Path: `/out/v1/*/321xyz/*` forward to Origin A
3. Path: `*` forward to Origin B

Untuk pengaturan distribusi lainnya, tetapkan nilai spesifik berdasarkan persyaratan teknis lain atau kebutuhan bisnis Anda. Untuk daftar semua opsi distribusi dan informasi tentang pengaturannya, lihat [the section called “Semua pengaturan distribusi”](#).

Setelah Anda selesai memilih pengaturan distribusi lainnya, pilih **Buat distribusi**.

4. Pilih distribusi yang baru saja Anda buat, lalu pilih **Behaviors**.
5. Pilih perilaku cache default, lalu pilih **Edit**. Tentukan pengaturan perilaku cache yang benar untuk konfigurasi kemasan yang Anda pilih untuk asal. Nanti, Anda akan menambahkan satu atau lebih asal tambahan dan mengedit pengaturan perilaku cache untuk mereka.
6. Buka [halaman CloudFront distribusi](#).
7. Tunggu hingga nilai kolom **Terakhir** yang dimodifikasi untuk distribusi Anda telah berubah dari **Deploying** ke tanggal dan waktu, yang menunjukkan bahwa CloudFront telah membuat distribusi Anda.

Langkah 2: Tambahkan Origins untuk domain grup MediaPackage kemasan Anda

Ulangi langkah-langkah di sini untuk menambahkan setiap grup MediaPackage kemasan Anda ke distribusi Anda, dengan mengingat kebutuhan untuk membuat asal “dummy”.

Untuk menambahkan grup kemasan lain sebagai asal

1. Di CloudFront konsol, pilih distribusi yang Anda buat untuk saluran Anda.
2. Pilih **Origins**, lalu pilih **Create origin**.
3. Untuk domain Origin, ketik URL untuk grup MediaPackage kemasan.
4. Untuk pengaturan lainnya, tetapkan nilai berdasarkan persyaratan teknis lain atau kebutuhan bisnis Anda. Untuk informasi selengkapnya, lihat [the section called “Pengaturan asal”](#).
5. Pilih **Buat asal**.

Langkah 3: Konfigurasikan perilaku cache untuk semua konfigurasi kemasan

Untuk setiap konfigurasi kemasan, Anda harus mengonfigurasi perilaku cache untuk menambahkan pola jalur yang merutekan permintaan dengan benar. Pola jalur yang Anda tentukan tergantung pada format video yang sedang Anda sajikan. Prosedur berikut mencakup informasi pola jalur yang dapat digunakan untuk format Apple HLS, CMAF, DASH, dan Microsoft Smooth Streaming.

Anda biasanya mengatur beberapa perilaku cache untuk setiap konfigurasi kemasan:

- Manifest induk, yang merupakan indeks untuk file Anda.
- Segmen, yang merupakan file konten video. Format mungkin menggunakan lebih dari satu ekstensi untuk konten, tergantung pada konfigurasi Anda. Perilaku cache diperlukan untuk setiap ekstensi.

Untuk membuat perilaku cache untuk konfigurasi kemasan

1. Di CloudFront konsol, pilih distribusi yang Anda buat untuk saluran Anda.
2. Pilih Perilaku, lalu pilih Buat perilaku.
3. Untuk pola Path, gunakan GUID konfigurasi kemasan MediaPackage VOD tertentu sebagai awalan jalur. Ini adalah GUID kedua di jalur MediaPackage VOD.

Pola jalur

Untuk konten HLS seperti `https://3ae97e9482b0d011.egress.mediapackage-vod.us-west-2.amazonaws.com/out/v1/abc123/def456/ghi789/index.m3u8`, buat perilaku cache berikut:

- Untuk manifes orang tua dan anak, gunakan `/out/v1/*/def456/*.m3u8`.
- Untuk segmen konten, gunakan `/out/v1/*/def456/*.ts` dan ulangi untuk semua ekstensi segmen yang diperlukan.

Untuk konten CMAF seperti `https://3ae97e9482b0d011.egress.mediapackage-vod.us-west-2.amazonaws.com/out/v1/abc123/def456/ghi789/index.m3u8`, buat perilaku cache berikut:

- Untuk manifes orang tua dan anak, gunakan `/out/v1/*/def456/*.m3u8`.
- Untuk segmen konten, gunakan `/out/v1/*/def456/*.mp4` dan ulangi untuk semua ekstensi segmen yang diperlukan.

Untuk konten DASH seperti `https://3ae97e9482b0d011.egress.mediapackage-vod.us-west-2.amazonaws.com/out/v1/abc123/def456/ghi789/index.mpd`, buat perilaku cache berikut:

- Untuk manifest orang tua, gunakan `/out/v1/*/def456/*.mpd`.
- Untuk segmen konten, gunakan `/out/v1/*/def456/*.mp4`.

Untuk titik akhir Microsoft Smooth Streaming

seperti `https://3ae97e9482b0d011.egress.mediapackage-vod.us-`

`west-2.amazonaws.com/out/v1/abc123/def456/ghi789/index.ism/Manifest`, hanya manifes yang disajikan, jadi Anda hanya membuat satu perilaku cache:`out/v1/*/def456/*/index.ism/*`.

4. Untuk setiap perilaku cache, tentukan nilai untuk pengaturan berikut:

Kebijakan protokol penampil

Pilih Arahkan ulang HTTP ke HTTPS.

Kebijakan cache dan kebijakan permintaan asal

Untuk kebijakan Cache, pilih Buat kebijakan. Untuk kebijakan cache baru Anda, tentukan pengaturan berikut:

TTL Minimum

Atur hingga 5 detik atau kurang, untuk membantu mencegah agar konten basi tidak tertahan.

String pertanyaan

Untuk string Kueri (dalam pengaturan kunci Cache), pilih Sertakan string kueri yang ditentukan. Untuk Izinkan, tambahkan nilai berikut dengan mengetiknya lalu pilih Tambah item:

- Jika Anda menggunakan fitur pemfilteran manifes di MediaPackage, tentukan `aws.manifestfilter` sebagai parameter string kueri tambahan untuk kebijakan cache yang Anda gunakan dengan perilaku cache untuk permintaan manifes (`*.m3u8,* .mpd, dan index.ism/*`). Ini mengonfigurasi distribusi Anda untuk meneruskan string `aws.manifestfilter` kueri ke MediaPackage asal Anda, yang diperlukan agar fitur pemfilteran manifes berfungsi. Untuk informasi selengkapnya, lihat [Pemfilteran manifes](#) di Panduan AWS Elemental MediaPackage Pengguna.

5. Pilih Buat.
6. Setelah Anda membuat kebijakan cache, kembali ke alur kerja pembuatan perilaku cache. Segarkan daftar kebijakan cache, dan pilih kebijakan yang baru saja Anda buat.
7. Pilih Buat perilaku.
8. Jika titik akhir Anda bukan titik akhir Microsoft Smooth Streaming, ulangi langkah-langkah ini untuk membuat perilaku cache kedua.

Langkah 4: Aktifkan Otorisasi CDN berbasis header MediaPackage

Sebaiknya aktifkan Otorisasi MediaPackage CDN berbasis header antara konten MediaPackage VOD dan distribusi. CloudFront Untuk informasi selengkapnya, lihat [Mengaktifkan otorisasi CDN MediaPackage di AWS Elemental MediaPackage](#) Panduan Pengguna.

Langkah 5: Gunakan CloudFront untuk menyajikan konten VOD

Setelah Anda membuat distribusi, menambahkan asal, membuat perilaku cache, dan mengaktifkan otorisasi CDN berbasis header, Anda dapat menyajikan konten VOD menggunakan CloudFront. CloudFront merutekan permintaan dari pemirsa ke konten MediaPackage VOD yang benar berdasarkan pengaturan yang Anda konfigurasi untuk perilaku cache.

Untuk tautan dalam aplikasi Anda (misalnya, pemutar media), tentukan URL untuk file media dalam format standar untuk CloudFront URLs. Untuk informasi selengkapnya, lihat [the section called “Kustomisasi URL file”](#).

Ketahanan sadar kualitas media

[Ketahanan sadar kualitas media \(MQAR\) adalah kemampuan terintegrasi antara Amazon dan Layanan Media. CloudFront AWS](#) MQAR menyediakan pemilihan asal lintas wilayah otomatis berdasarkan Skor Keyakinan Kualitas Media (MQCS). MQCS disintesis AWS Elemental MediaLive berdasarkan parameter yang memengaruhi pengalaman kualitas media yang dirasakan oleh pemirsa. Anda dapat mengonfigurasi CloudFront dan Layanan AWS Media untuk menghadirkan streaming acara langsung Anda dengan ketahanan tinggi dengan menggunakan beberapa opsi yang dapat Anda tentukan dalam kriteria failover grup CloudFront asal.

Ketika Anda mengaktifkan fitur MQAR untuk distribusi Anda, Anda berwenang CloudFront untuk secara otomatis memilih asal yang dianggap memiliki skor kualitas tertinggi.

Skor kualitas mewakili masalah kualitas streaming media yang dirasakan dari asal Anda, seperti bingkai hitam, bingkai beku atau jatuh, atau bingkai berulang. Misalnya, jika asal AWS Elemental MediaPackage v2 Anda digunakan dalam dua yang berbeda Wilayah AWS, dan satu melaporkan skor kualitas media yang lebih tinggi daripada yang lain, secara otomatis CloudFront akan beralih ke asal yang melaporkan skor yang lebih tinggi.

Untuk mencapai ini, CloudFront lakukan hal berikut:

1. CloudFront meneruskan GET permintaan ke MediaPackage asal primer, dan juga memulai HEAD permintaan ke MediaPackage asal sekunder pada saat yang sama. CloudFront menerima skor kualitas media di header respons dari masing-masing asal.
2. Selanjutnya, CloudFront lacak skor untuk setiap asal dan gunakan informasi ini untuk menentukan asal dengan skor yang lebih tinggi ketika permintaan baru tiba.

Skor kualitas media untuk asal Anda dapat berubah secara real time. CloudFront menentukan hal ini dengan mengonsumsi perubahan MQCS, dan beralih antar asal untuk memastikan bahwa pemirsa melihat konten berkualitas media yang lebih tinggi. Untuk informasi selengkapnya, lihat [Memanfaatkan skor kualitas media dengan MediaPackage](#) di Panduan Pengguna AWS Elemental MediaPackage V2.

MQAR membantu CloudFront menentukan, sedini mungkin, apakah ada masalah yang berpotensi berdampak pada pelanggan. Misalnya, masalah seperti koneksi jaringan, pemrosesan video, kehilangan atau penurunan audio, masalah kecepatan encoder dapat memengaruhi skor kualitas media untuk pemirsa Anda.

MQAR menyediakan peralihan antar asal tanpa batas, sehingga Anda dapat menerapkan alur kerja pengiriman end-to-end media lintas wilayah yang tangguh, dan menyediakan konten berkualitas untuk pemirsa AWS Anda.

 Note

Saat ini, fitur ini hanya mendukung asal MediaPackage v2.

Untuk mengaktifkan fitur ini untuk distribusi Anda, selesaikan langkah-langkah berikut:

1. Jika Anda belum melakukannya, buat asal MediaPackage v2 Anda dan aktifkan fitur ini di konfigurasi titik akhir Anda. Untuk penyebaran lintas wilayah, buat saluran sekunder dalam pengaturan yang berbeda Wilayah AWS dengan pengaturan yang sama. Untuk informasi selengkapnya, lihat topik berikut di Panduan Pengguna AWS Elemental MediaPackage V2:
 - [Buat saluran dan titik akhir](#)
 - [Aktifkan skor kualitas media](#)
2. Untuk menggunakan asal MediaPackage v2 Anda CloudFront, buat atau perbarui CloudFront distribusi. Lihat [Buat distribusi](#) dan [Perbarui distribusi](#).

3. Buat grup asal, dan pilih dua asal Anda sebagai primer dan sekunder. Di grup asal Anda, aktifkan opsi Skor kualitas Media. Untuk informasi selengkapnya, lihat [Buat grup asal](#).
4. Dalam perilaku cache untuk distribusi Anda, pilih [grup asal](#) yang Anda buat. Kami menyarankan agar perilaku cache cocok dengan pola jalur saluran.

Jika CloudFront menentukan bahwa kedua asal MediaPackage v2 memiliki skor yang sama, maka itu meneruskan permintaan ke asal primer seperti yang tercantum dalam grup asal. Jika asal yang dipilih awalnya merespons dengan kode kesalahan yang cocok dengan kriteria failover yang Anda tentukan di grup asal Anda, CloudFront coba ulang permintaan ke asal alternatif di grup asal Anda terlepas dari skor kualitas medianya.

Catatan

- CloudFront melacak skor kualitas untuk setiap perilaku cache yang menggunakan grup asal yang diaktifkan untuk skor kualitas media. Jika grup asal yang sama digunakan untuk beberapa saluran yang mengeluarkan skor kualitas media, buat perilaku cache terpisah untuk pola jalur setiap saluran untuk menghindari pencampuran skornya. Untuk informasi lebih lanjut tentang kuota grup asal, lihat [Kuota umum di distribusi](#).
- Saat ini, MQAR tidak tersedia saat Anda menggunakan fungsi [Lambda @Edge](#) di pemicu yang menghadap asal (permintaan asal dan respons asal) yang terkait dengan perilaku cache distribusi Anda. Untuk informasi selengkapnya, lihat [Pengaturan perilaku cache](#).
- Jika Anda mengaktifkan fitur MQAR dan kontrol akses asal (OAC), tambahkan `mediapackagev2:GetHeadObject` tindakan ke kebijakan IAM. MQAR memerlukan izin ini untuk mengirim HEAD permintaan ke asal MediaPackage v2. Untuk informasi lebih lanjut tentang OAC, lihat [Batasi akses ke asal AWS Elemental MediaPackage v2](#).

Bidang log MQAR

CloudFront menyediakan bidang berikut dalam log akses real-time untuk mencerminkan skor kualitas dan asal yang dipilih. Anda dapat mengaktifkan bidang ini di log akses CloudFront real-time Anda:

- `r-host`
- `sr-reason`
- `x-edge-mqcs`

Untuk informasi lebih lanjut, lihat [Bidang](#) 65-67.

Sesuaikan di tepi dengan fungsi

Dengan Amazon CloudFront, Anda dapat menulis kode Anda sendiri untuk menyesuaikan bagaimana CloudFront distribusi Anda memproses permintaan dan tanggapan HTTP. Kode berjalan dekat dengan pemirsa Anda (pengguna) untuk meminimalkan latensi, dan Anda tidak perlu mengelola server atau infrastruktur lainnya. Anda dapat menulis kode untuk memanipulasi permintaan dan tanggapan yang mengalir CloudFront, melakukan otentikasi dan otorisasi dasar, menghasilkan respons HTTP di tepi, dan banyak lagi.

Kode yang Anda tulis dan lampirkan ke CloudFront distribusi Anda disebut fungsi tepi. CloudFront menyediakan dua cara untuk menulis dan mengelola fungsi tepi:

CloudFront Fungsi

Anda dapat menulis fungsi ringan JavaScript untuk penyesuaian CDN skala tinggi yang sensitif terhadap latensi. Lingkungan runtime CloudFront Functions menawarkan waktu startup submilidetik, skala segera untuk menangani jutaan permintaan per detik, dan sangat aman. CloudFront Fungsi adalah fitur asli CloudFront, yang berarti Anda dapat membangun, menguji, dan menyebarkan kode Anda sepenuhnya di dalamnya CloudFront.

Lambda@Edge

Lambda @Edge adalah perpanjangan yang menawarkan komputasi [AWS Lambda](#) yang kuat dan fleksibel untuk fungsi kompleks dan logika aplikasi lengkap yang lebih dekat dengan pemirsa Anda, dan sangat aman. Fungsi Lambda@Edge berjalan di lingkungan waktu aktif Node.js atau Python. Anda mempublikasikannya ke satu Wilayah AWS, tetapi ketika Anda mengaitkan fungsi dengan CloudFront distribusi, Lambda @Edge secara otomatis mereplikasi kode Anda di seluruh dunia.

Jika Anda menjalankan AWS WAF CloudFront, Anda dapat menggunakan header yang AWS WAF disisipkan untuk CloudFront Fungsi dan Lambda @Edge. Ini berfungsi untuk permintaan dan tanggapan penampil dan asal.

Topik

- [Perbedaan antara CloudFront Fungsi dan Lambda @Edge](#)
- [Sesuaikan di tepi dengan CloudFront Fungsi](#)
- [Sesuaikan dengan Fungsi CloudFront Koneksi](#)
- [Sesuaikan di tepi dengan Lambda @Edge](#)

- [Pembatasan pada fungsi edge](#)

Perbedaan antara CloudFront Fungsi dan Lambda @Edge

CloudFront Fungsi dan Lambda @Edge keduanya menyediakan cara untuk menjalankan kode sebagai respons terhadap CloudFront peristiwa.

CloudFront Fungsi sangat ideal untuk fungsi ringan dan berjalan pendek untuk kasus penggunaan berikut:

- Normalisasi kunci cache - Ubah atribut permintaan HTTP (header, string kueri, cookie, dan bahkan jalur URL) untuk membuat [kunci cache](#) yang optimal, yang dapat meningkatkan rasio hit cache Anda.
- Manipulasi header - Menyisipkan, memodifikasi, atau menghapus header HTTP dalam permintaan atau tanggapan. Misalnya, Anda dapat menambah header `True-Client-IP` untuk setiap permintaan.
- Pengalihan atau penulisan ulang URL — Mengarahkan pemirsa ke halaman lain berdasarkan informasi dalam permintaan, atau menulis ulang semua permintaan dari satu jalur ke jalur lainnya.
- Meminta otorisasi — Validasi token otorisasi hash, seperti token web JSON (JWT), dengan memeriksa header otorisasi atau metadata permintaan lainnya.

Untuk memulai dengan CloudFront Functions, lihat [Sesuaikan di tepi dengan CloudFront Fungsi](#).

Lambda @Edge sangat ideal untuk kasus penggunaan berikut:

- Fungsi yang membutuhkan beberapa milidetik atau lebih untuk diselesaikan
- Fungsi yang membutuhkan CPU atau memori yang dapat disesuaikan
- Fungsi yang bergantung pada pustaka pihak ketiga (termasuk AWS SDK, untuk integrasi dengan yang lain) Layanan AWS
- Fungsi yang memerlukan akses jaringan untuk menggunakan layanan eksternal untuk diproses
- Fungsi yang memerlukan akses sistem file atau akses ke badan permintaan HTTP

Untuk memulai dengan Lambda@Edge, lihat [Sesuaikan di tepi dengan Lambda @Edge](#).

Untuk membantu Anda memilih opsi untuk kasus penggunaan Anda, gunakan tabel berikut untuk memahami perbedaan antara CloudFront Fungsi dan Lambda @Edge. Untuk informasi tentang

perbedaan yang berlaku untuk metode pembantu modifikasi asal, lihat [Pilih antara CloudFront Fungsi dan Lambda @Edge](#).

	CloudFront Fungsi	Lambda@Edge
Bahasa pemrograman	JavaScript (ECMAScript 5.1 sesuai)	Node.js dan Python
Sumber peristiwa	• Permintaan penampil • Respons penampil	• Permintaan penampil • Respons penampil • Permintaan asal • Respons asal
Mendukung Amazon CloudFront KeyStore	Ya CloudFront KeyStore hanya mendukung JavaScript runtime 2.0	Tidak
Penskalaan	Hingga jutaan permintaan per detik	Hingga 10.000 permintaan per detik per Wilayah
Durasi fungsi	Submilidetik	Hingga 30 detik (permintaan penampil dan respons penampil) Hingga 30 detik (permintaan asal dan respons asal)
Ukuran memori fungsi maksimum	2 MB	128 MB (permintaan penampil dan respons penampil) 10.240 MB (10 GB) (permintaan asal dan respons asal) Untuk informasi selengkapnya, lihat

	CloudFront Fungsi	Lambda@Edge
		Kuotas di Lambda@Edge.
Ukuran maksimum dari kode fungsi dan termasuk pustaka	10 KB	50 MB (permintaan penampil dan respons penampil) 50 MB (permintaan asal dan respons asal)
Akses jaringan	Tidak	Ya
Akses sistem file	Tidak	Ya
Akses ke isi permintaan	Tidak	Ya
Akses ke data geolokasi dan perangkat	Ya	Tidak (permintaan penampil dan respons penampil) Ya (permintaan asal dan respons asal)
Dapat membangun dan menguji sepenuhnya di dalam CloudFront	Ya	Tidak
Pencatatan dan metrik fungsi	Ya	Ya

Sesuaikan di tepi dengan CloudFront Fungsi

Dengan CloudFront Functions, Anda dapat menulis fungsi ringan JavaScript untuk penyesuaian CDN skala tinggi yang sensitif terhadap latensi. Fungsi Anda dapat memanipulasi permintaan dan respons yang mengalir CloudFront, melakukan otentikasi dan otorisasi dasar, menghasilkan respons HTTP di tepi, dan banyak lagi. Lingkungan runtime CloudFront Functions menawarkan waktu startup submilidetik, skala segera untuk menangani jutaan permintaan per detik, dan sangat aman. CloudFront Fungsi adalah fitur asli CloudFront, yang berarti Anda dapat membangun, menguji, dan menyebarkan kode Anda sepenuhnya di dalamnya CloudFront.

Saat Anda mengaitkan CloudFront fungsi dengan CloudFront distribusi, CloudFront mencegat permintaan dan respons di lokasi CloudFront tepi dan meneruskannya ke fungsi Anda. Anda dapat memanggil CloudFront Fungsi saat peristiwa berikut terjadi:

- Saat CloudFront menerima permintaan dari penampil (permintaan penampil)
- Sebelum CloudFront mengembalikan respons ke penampil (respons penampil)
- Selama pembentukan koneksi TLS (permintaan koneksi) - saat ini tersedia untuk koneksi TLS (mTLS) bersama

Untuk informasi selengkapnya tentang CloudFront Fungsi, lihat topik berikut:

Topik

- [Tutorial: Buat fungsi sederhana dengan CloudFront Fungsi](#)
- [Tutorial: Buat CloudFront fungsi yang mencakup nilai-nilai kunci](#)
- [Tulis kode fungsi](#)
- [Buat fungsi](#)
- [Fungsi uji](#)
- [Perbarui fungsi](#)
- [Publikasikan fungsi](#)
- [Mengaitkan fungsi dengan distribusi](#)
- [Amazon CloudFront KeyValueCollection](#)

Tutorial: Buat fungsi sederhana dengan CloudFront Fungsi

Tutorial ini menunjukkan kepada Anda bagaimana memulai dengan CloudFront Functions. Anda dapat membuat fungsi sederhana yang mengarahkan penampil ke URL yang berbeda, dan itu juga mengembalikan header respons khusus.

Daftar Isi

- [Prasyarat](#)
- [Buat fungsi](#)
- [Verifikasi fungsinya](#)

Prasyarat

Untuk menggunakan CloudFront Fungsi, Anda memerlukan CloudFront distribusi. Jika Anda tidak memilikinya, lihat [Memulai dengan distribusi CloudFront standar](#).

Buat fungsi

Anda dapat menggunakan CloudFront konsol untuk membuat fungsi sederhana yang mengarahkan penampil ke URL yang berbeda, dan juga mengembalikan header respons khusus.

Untuk membuat CloudFront fungsi

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Fungsi, lalu pilih Buat fungsi.
3. Pada halaman Create function, untuk Name, masukkan nama fungsi seperti *MyFunctionName*.
4. (Opsional) Untuk Deskripsi, masukkan deskripsi untuk fungsi seperti **Simple test function**.
5. Untuk Runtime, pertahankan JavaScript versi default yang dipilih.
6. Pilih Buat fungsi.
7. Salin kode fungsi berikut. Kode fungsi ini mengalihkan penampil ke URL yang berbeda dan menampilkan header respons kustom.

```
function handler(event) {  
    // NOTE: This example function is for a viewer request event trigger.  
    // Choose viewer request for event trigger when you associate this function  
    with a distribution.  
    var response = {  
        statusCode: 302,  
        statusDescription: 'Found',  
        headers: {  
            'cloudfront-functions': { value: 'generated-by-CloudFront-Functions' },  
            'location': { value: 'https://aws.amazon.com/cloudfront/' }  
        }  
    };  
    return response;  
}
```

8. Untuk kode Fungsi, tempelkan kode ke editor kode untuk mengganti kode default.
9. Pilih Simpan perubahan.

10. (Opsional) Anda dapat menguji fungsi sebelum Anda mempublikasikannya. Tutorial ini tidak menjelaskan cara menguji suatu fungsi. Untuk informasi selengkapnya, lihat [Fungsi uji](#).
11. Pilih tab Publish dan kemudian pilih fungsi Publish. Anda harus mempublikasikan fungsi sebelum Anda dapat mengaitkannya dengan CloudFront distribusi Anda.
12. Selanjutnya, Anda dapat mengaitkan fungsi dengan perilaku distribusi atau cache. Pada *MyFunctionName* halaman, pilih tab Publikasikan.

Warning

Pada langkah-langkah berikut, pilih distribusi atau perilaku cache yang digunakan untuk pengujian. Jangan kaitkan fungsi pengujian ini dengan perilaku distribusi atau cache yang digunakan dalam produksi.

13. Pilih Tambah asosiasi.
14. Pada kotak dialog Associate, pilih distribusi dan/atau perilaku cache. Untuk jenis Event, pertahankan nilai default.
15. Pilih Tambah asosiasi.

Tabel distribusi terkait menunjukkan distribusi terkait.

16. Tunggu beberapa menit sampai distribusi terkait dapat menyelesaikan penyebaran. Untuk memeriksa status distribusi, pilih distribusi di tabel Distribusi terkait, lalu pilih Lihat distribusi.

Ketika status distribusi Diterapkan, Anda siap memverifikasi bahwa fungsi tersebut dapat digunakan.

Verifikasi fungsinya

Setelah Anda menerapkan fungsi, Anda dapat memverifikasi bahwa itu berfungsi untuk distribusi Anda.

Untuk memverifikasi fungsi

1. Di browser web Anda, navigasikan ke nama domain distribusi Anda (misalnya, `https://d111111abcdef8.cloudfront.net`).

Fungsi mengembalikan pengalihan ke browser, sehingga browser secara otomatis masuk ke `https://aws.amazon.com/cloudfront/`.

2. Di jendela baris perintah, Anda dapat menggunakan alat seperti curl mengirim permintaan ke nama domain distribusi Anda.

```
curl -v https://d111111abcdef8.cloudfront.net/
```

Dalam respons, Anda melihat respon pengalihan (302 Found) dan header respons khusus yang ditambahkan fungsi tersebut. Tanggapan Anda mungkin terlihat seperti contoh berikut.

Example

```
curl -v https://d111111abcdef8.cloudfront.net/
> GET / HTTP/1.1
> Host: d111111abcdef8.cloudfront.net
> User-Agent: curl/7.64.1
> Accept: */*
>
< HTTP/1.1 302 Found
< Server: CloudFront
< Date: Tue, 16 Mar 2021 18:50:48 GMT
< Content-Length: 0
< Connection: keep-alive
< Location: https://aws.amazon.com/cloudfront/
< Cloudfront-Functions: generated-by-CloudFront-Functions
< X-Cache: FunctionGeneratedResponse from cloudfront
< Via: 1.1 3035b31bddaf14eded329f8d22cf188c.cloudfront.net (CloudFront)
< X-Amz-Cf-Pop: PHX50-C2
< X-Amz-Cf-Id: ULZdIz6j43uGB1Xyob_JctF9x7CCbwpNniiMlmNbmwzH1YWP9FsEHg==
```

Tutorial: Buat CloudFront fungsi yang mencakup nilai-nilai kunci

Tutorial ini menunjukkan kepada Anda bagaimana untuk memasukkan nilai-nilai kunci dengan CloudFront fungsi. Nilai kunci adalah bagian dari pasangan kunci-nilai. Anda menyertakan nama (dari pasangan kunci-nilai) dalam kode fungsi. Ketika fungsi berjalan, CloudFront menggantikan nama dengan nilai.

Pasangan kunci-nilai adalah variabel yang disimpan dalam penyimpanan nilai kunci. Bila Anda menggunakan kunci dalam fungsi Anda (bukan nilai hard-code), fungsi Anda lebih fleksibel. Anda dapat mengubah nilai kunci tanpa harus menerapkan perubahan kode. Pasangan nilai kunci juga dapat mengurangi ukuran fungsi Anda. Untuk informasi selengkapnya, lihat [???](#).

Daftar Isi

- [Prasyarat](#)
- [Buat toko nilai kunci](#)
- [Tambahkan pasangan kunci-nilai ke penyimpanan nilai kunci](#)
- [Kaitkan penyimpanan nilai kunci dengan fungsi](#)
- [Uji dan publikasikan kode fungsi](#)

Prasyarat

Jika Anda baru mengenal CloudFront fungsi Fungsi dan penyimpanan nilai kunci, kami sarankan Anda mengikuti tutorial di [the section called “Tutorial: Buat CloudFront fungsi sederhana”](#).

Setelah Anda menyelesaikan tutorial itu, Anda dapat mengikuti tutorial ini untuk memperluas fungsi yang Anda buat. Untuk tutorial ini, kami sarankan Anda membuat penyimpanan nilai kunci terlebih dahulu.

Buat toko nilai kunci

Pertama, buat penyimpanan nilai kunci yang akan digunakan untuk fungsi Anda.

Untuk membuat penyimpanan nilai kunci

1. Rencanakan pasangan kunci-nilai yang ingin Anda sertakan dalam fungsi. Catat nama-nama kunci. Pasangan kunci-nilai yang ingin Anda gunakan dalam suatu fungsi harus berada dalam penyimpanan nilai kunci tunggal.
2. Putuskan tentang urutan pekerjaan. Ada dua cara untuk melanjutkan:
 - Buat penyimpanan nilai kunci, dan tambahkan pasangan kunci-nilai ke toko. Kemudian buat (atau modifikasi) fungsi dan masukkan nama-nama kunci.
 - Atau, buat (atau modifikasi) fungsi dan sertakan nama kunci yang ingin Anda gunakan. Kemudian buat penyimpanan nilai kunci, dan tambahkan pasangan kunci-nilai.
3. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
4. Di panel navigasi, pilih Fungsi, lalu pilih KeyValueStorestab.
5. Pilih Buat KeyValueStore dan masukkan bidang berikut:
 - Masukkan nama dan deskripsi (opsional) untuk toko.

- Biarkan URI S3 kosong. Dalam tutorial ini Anda akan memasukkan pasangan kunci-nilai secara manual.
6. Pilih Buat. Halaman detail untuk penyimpanan nilai kunci baru muncul. Halaman ini mencakup bagian Pasangan nilai kunci yang saat ini kosong.

Tambahkan pasangan kunci-nilai ke penyimpanan nilai kunci

Selanjutnya, tambahkan daftar pasangan kunci-nilai secara manual ke penyimpanan nilai kunci yang sebelumnya Anda buat.

Untuk menambahkan pasangan kunci-nilai ke penyimpanan nilai kunci

1. Di bagian Pasangan nilai kunci, pilih Tambahkan pasangan nilai kunci.
2. Pilih Tambah pasangan dan kemudian masukkan kunci dan nilai. Pilih tanda centang untuk mengonfirmasi perubahan Anda dan ulangi langkah ini untuk menambahkan lebih banyak.
3. Setelah selesai, pilih Simpan perubahan untuk menyimpan pasangan nilai kunci di penyimpanan nilai kunci. Pada dialog konfirmasi, pilih Selesai.

Anda sekarang memiliki penyimpanan nilai kunci yang berisi sekelompok pasangan kunci-nilai.

Kaitkan penyimpanan nilai kunci dengan fungsi

Anda sekarang telah membuat toko nilai kunci. Dan Anda telah membuat atau memodifikasi fungsi yang menyertakan nama kunci dari penyimpanan nilai kunci. Anda sekarang dapat mengaitkan penyimpanan nilai kunci dan fungsinya. Anda membuat asosiasi itu dari dalam fungsi.

Untuk mengaitkan penyimpanan nilai kunci dengan fungsi

1. Di panel navigasi, pilih Fungsi. Tab Fungsi muncul di atas, secara default.
2. Pilih nama fungsi dan di KeyValueCollection bagian Terkait, pilih Associate Existing KeyValueCollection.
3. Pilih toko nilai kunci dan pilih Associate KeyValueCollection.

Note

Anda dapat mengaitkan hanya satu penyimpanan nilai kunci dengan setiap fungsi.

Uji dan publikasikan kode fungsi

Setelah Anda mengaitkan penyimpanan nilai kunci dengan fungsi Anda, Anda dapat menguji dan mempublikasikan kode fungsi. Anda harus selalu menguji kode fungsi setiap kali Anda memodifikasinya, termasuk ketika Anda melakukan hal berikut:

- Kaitkan penyimpanan nilai kunci dengan fungsi.
- Ubah fungsi dan penyimpanan nilai kuncinya untuk menyertakan pasangan kunci-nilai baru.
- Ubah nilai pasangan kunci-nilai.

Untuk menguji dan mempublikasikan kode fungsi

1. Untuk informasi tentang cara menguji fungsi, lihat [the section called “Fungsi uji”](#). Pastikan Anda memilih untuk menguji fungsi di DEVELOPMENT panggung.
2. Publikasikan fungsi saat Anda siap menggunakan fungsi (dengan pasangan nilai kunci baru atau yang direvisi) di LIVE lingkungan.

Saat Anda CloudFront mempublikasikan, salin versi fungsi dari DEVELOPMENT panggung ke panggung langsung. Fungsi ini memiliki kode baru dan dikaitkan dengan penyimpanan nilai kunci. (Tidak perlu melakukan asosiasi lagi, di panggung langsung.)

Untuk informasi tentang cara mempublikasikan fungsi, lihat [the section called “Publikasikan fungsi”](#).

Tulis kode fungsi

Anda dapat menggunakan CloudFront Fungsi untuk menulis fungsi ringan untuk penyesuaian JavaScript CDN skala tinggi yang sensitif terhadap latensi. Kode fungsi Anda dapat memanipulasi permintaan dan respons yang mengalir CloudFront, melakukan otentikasi dan otorisasi dasar, menghasilkan respons HTTP di tepi, dan banyak lagi.

Untuk membantu Anda menulis kode fungsi untuk CloudFront Fungsi, lihat topik berikut. Untuk contoh kode, lihat [CloudFront Contoh fungsi untuk CloudFront](#) dan repositori [amazon-cloudfront-functions](#) aktif. GitHub

Topik

- [Tentukan tujuan fungsi](#)

- [CloudFront Fungsi struktur acara](#)
- [JavaScript fitur runtime untuk Fungsi CloudFront](#)
- [Metode pembantu untuk penyimpanan nilai kunci](#)
- [Metode pembantu untuk modifikasi asal](#)
- [Metode pembantu untuk properti CloudFront SaaS Manager](#)
- [Gunakan asinkron and menunggu](#)
- [Dukungan CWT untuk Fungsi CloudFront](#)
- [Metode pembantu umum](#)
- [Metode pembantu untuk TLS timbal balik](#)

Tentukan tujuan fungsi

Sebelum Anda menulis kode fungsi Anda, tentukan tujuan fungsi Anda. Sebagian besar CloudFront fungsi dalam Fungsi memiliki salah satu tujuan berikut.

Topik

- [Ubah permintaan HTTP dalam jenis acara permintaan penampil](#)
- [Hasilkan respons HTTP dalam jenis acara permintaan penampil](#)
- [Ubah respons HTTP dalam jenis peristiwa respons penampil](#)
- [Validasi koneksi mTLS dalam jenis acara permintaan koneksi](#)
- [Informasi Terkait](#)

Terlepas dari tujuan fungsi Anda, `handler` adalah titik masuk untuk fungsi apa pun. Dibutuhkan argumen tunggal yang disebut `event`, yang diteruskan ke fungsi oleh CloudFront. `event` adalah objek JSON yang berisi representasi dari permintaan HTTP (dan respons, jika fungsi Anda memodifikasi respons HTTP).

Ubah permintaan HTTP dalam jenis acara permintaan penampil

Fungsi Anda dapat memodifikasi permintaan HTTP yang CloudFront menerima dari penampil (klien), dan mengembalikan permintaan yang dimodifikasi ke CloudFront pemrosesan lanjutan. Misalnya, kode fungsi Anda mungkin menormalkan [kunci cache](#) atau mengubah header permintaan.

Setelah Anda membuat dan memublikasikan fungsi yang mengubah permintaan HTTP, pastikan untuk menambahkan asosiasi untuk jenis peristiwa permintaan penampil. Untuk informasi

selengkapnya, lihat [Buat fungsi](#). Ini membuat fungsi berjalan setiap kali CloudFront menerima permintaan dari penampil, sebelum memeriksa untuk melihat apakah objek yang diminta ada dalam CloudFront cache.

Example Contoh

Pseudocode berikut menunjukkan struktur fungsi yang mengubah permintaan HTTP.

```
function handler(event) {  
    var request = event.request;  
  
    // Modify the request object here.  
  
    return request;  
}
```

Fungsi mengembalikan request objek yang dimodifikasi ke CloudFront. CloudFront terus memproses permintaan yang dikembalikan dengan memeriksa CloudFront cache untuk mendapatkan cache, dan mengirim permintaan ke asal jika perlu.

Hasilkan respons HTTP dalam jenis acara permintaan penampil

Fungsi Anda dapat menghasilkan respons HTTP di tepi dan mengembalikannya langsung ke penampil (klien) tanpa memeriksa respons yang di-cache atau pemrosesan lebih lanjut oleh CloudFront. Misalnya, kode fungsi Anda mungkin mengalihkan permintaan ke URL baru, atau memeriksa otorisasi dan mengembalikan respons 401 atau 403 terhadap permintaan yang tidak sah.

Saat Anda membuat fungsi yang menghasilkan respons HTTP, pastikan untuk memilih jenis peristiwa permintaan penampil. Ini berarti bahwa fungsi berjalan setiap kali CloudFront menerima permintaan dari penampil, sebelum CloudFront melakukan pemrosesan permintaan lebih lanjut.

Example Contoh

Pseudocode berikut menunjukkan struktur fungsi yang menghasilkan respons HTTP.

```
function handler(event) {  
    var request = event.request;  
  
    var response = ...; // Create the response object here,  
                        // using the request properties if needed.
```

```
    return response;
}
```

Fungsi mengembalikan `response` objek ke CloudFront, yang CloudFront segera kembali ke penampil tanpa memeriksa CloudFront cache atau mengirim permintaan ke asal.

Ubah respons HTTP dalam jenis peristiwa respons penampil

Fungsi Anda dapat memodifikasi respons HTTP sebelum CloudFront mengirimkannya ke penampil (klien), terlepas dari apakah respons berasal dari CloudFront cache atau asal. Misalnya, kode fungsi Anda mungkin menambahkan atau memodifikasi header respons, kode status, dan isi isi.

Saat Anda membuat fungsi yang mengubah respons HTTP, pastikan untuk memilih jenis peristiwa respons penampil. Ini berarti bahwa fungsi berjalan sebelum CloudFront mengembalikan respons ke penampil, terlepas dari apakah respons berasal dari CloudFront cache atau asal.

Example Contoh

Pseudocode berikut menunjukkan struktur fungsi yang mengubah respons HTTP.

```
function handler(event) {
    var request = event.request;
    var response = event.response;

    // Modify the response object here,
    // using the request properties if needed.

    return response;
}
```

Fungsi mengembalikan `response` objek yang dimodifikasi ke CloudFront, yang CloudFront segera kembali ke penampil.

Validasi koneksi mTLS dalam jenis acara permintaan koneksi

Fungsi koneksi adalah jenis CloudFront Fungsi yang berjalan selama koneksi TLS untuk memberikan validasi kustom dan logika otentikasi. Fungsi koneksi saat ini tersedia untuk koneksi TLS (mTLS) bersama, di mana Anda dapat memvalidasi sertifikat klien dan menerapkan logika otentikasi khusus di luar validasi sertifikat standar. Fungsi koneksi berjalan selama proses jabat tangan TLS dan dapat mengizinkan atau menolak koneksi berdasarkan properti sertifikat, alamat IP klien, atau kriteria lainnya.

Setelah Anda membuat dan menerbitkan fungsi koneksi, pastikan untuk menambahkan asosiasi untuk jenis peristiwa permintaan koneksi dengan TLS-enabled distribusi m. Hal ini membuat fungsi berjalan setiap kali klien mencoba untuk membuat koneksi mTLS dengan CloudFront.

Example

Pseudocode berikut menunjukkan struktur fungsi koneksi:

```
function connectionHandler(connection) {  
    // Validate certificate and connection properties here.  
  
    if (/* validation passes */) {  
        connection.allow();  
    } else {  
        connection.deny();  
    }  
}
```

Fungsi ini menggunakan metode pembantu untuk menentukan apakah akan mengizinkan atau menolak koneksi. Tidak seperti permintaan penampil dan fungsi respons penampil, fungsi koneksi tidak dapat mengubah permintaan atau tanggapan HTTP.

Informasi Terkait

Untuk informasi selengkapnya tentang bekerja dengan CloudFront Fungsi, lihat topik berikut:

- [Struktur peristiwa](#)
- [JavaScript fitur runtime](#)
- [CloudFront Contoh fungsi](#)
- [Pembatasan pada fungsi edge](#)

CloudFront Fungsi struktur acara

CloudFront Fungsi meneruskan event objek ke kode fungsi Anda sebagai input ketika menjalankan fungsi. Saat Anda [menguji fungsi](#), Anda membuat objek event dan meneruskannya ke fungsi Anda. Saat Anda membuat objek event untuk menguji fungsi, Anda dapat menghilangkan kolom `distributionDomainName`, `distributionId`, dan `requestId` dalam objek context. Pastikan bahwa nama header adalah huruf kecil, yang selalu terjadi pada event objek yang diteruskan CloudFront Functions ke fungsi Anda dalam produksi.

Berikut ini menunjukkan gambaran umum struktur objek peristiwa ini.

```
{
  "version": "1.0",
  "context": {
    <context object>
  },
  "viewer": {
    <viewer object>
  },
  "request": {
    <request object>
  },
  "response": {
    <response object>
  }
}
```

Untuk informasi selengkapnya, lihat topik berikut:

Topik

- [Bidang versi](#)
- [Objek konteks](#)
- [Struktur acara koneksi](#)
- [Objek penampil](#)
- [Permintaan objek](#)
- [Objek respons](#)
- [Kode status dan badan](#)
- [Struktur untuk string kueri, header, atau cookie](#)
- [Contoh objek respon](#)
- [Contoh objek acara](#)

Bidang versi

`version` Bidang berisi string yang menentukan versi objek acara CloudFront Functions. Versi saat ini adalah `1.0`.

Objek konteks

Objek `context` berisi informasi kontekstual tentang peristiwa tersebut. Ini mencakup kolom-kolom berikut:

`distributionDomainName`

Nama CloudFront domain (misalnya, `d111111abcdef8.cloudfront.net`) dari distribusi standar yang terkait dengan acara tersebut.

`distributionDomainNameBidang` hanya muncul ketika fungsi Anda dipanggil untuk distribusi standar.

`endpoint`

Nama CloudFront domain (misalnya, `d111111abcdef8.cloudfront.net`) dari grup koneksi yang terkait dengan acara tersebut.

`endpointBidang` hanya muncul ketika fungsi Anda dipanggil untuk distribusi multi-penyewa.

`distributionId`

ID distribusi (misalnya, `EDFDVBD6 CONTOH`) yang terkait dengan acara.

`eventType`

Jenis peristiwa, `viewer-request` atau `viewer-response`.

`requestId`

String yang secara unik mengidentifikasi CloudFront permintaan (dan respons terkait).

Struktur acara koneksi

Fungsi koneksi menerima struktur peristiwa yang berbeda dari fungsi penampil. Untuk informasi rinci tentang struktur acara koneksi dan format respons, lihat [Kaitkan Fungsi CloudFront Koneksi](#).

Objek penampil

Objek `viewer` berisi kolom `ip` yang nilainya adalah alamat IP penampil (klien) yang mengirim permintaan. Jika permintaan penampil melewati proksi HTTP atau penyeimbang beban, nilainya adalah alamat IP proksi atau load balancer.

Permintaan objek

`requestObjek` berisi representasi dari permintaan viewer-to-CloudFront HTTP. Dalam event objek yang diteruskan ke fungsi Anda, `request` objek mewakili permintaan aktual yang CloudFront diterima dari penampil.

Jika kode fungsi Anda mengembalikan `request` objek ke CloudFront, itu harus menggunakan struktur yang sama.

Objek `request` berisi kolom-kolom berikut:

method

Metode HTTP permintaan. Jika kode fungsi Anda mengembalikan `arequest`, itu tidak dapat memodifikasi bidang ini. Ini adalah satu-satunya kolom hanya-baca di objek `request`.

uri

Jalur relatif objek yang diminta.

Note

Jika fungsi Anda memodifikasi `uri` nilai, berikut ini berlaku:

- Nilai `uri` baru harus dimulai dengan garis miring ke depan (/).
- Saat fungsi mengubah nilai `uri`, fungsi tersebut mengubah objek yang diminta oleh penampil.
- Saat fungsi mengubah `uri` nilainya, fungsi tersebut tidak mengubah perilaku cache untuk permintaan atau asal tempat permintaan asal dikirim.

querystring

Sebuah objek yang mewakili string kueri dalam permintaan. Jika permintaan tidak menyertakan string kueri, `request` objek masih menyertakan `querystring` objek kosong.

Objek `querystring` berisi satu kolom untuk setiap parameter string kueri dalam permintaan.

headers

Sebuah objek yang mewakili string kueri dalam permintaan. Jika permintaan berisi header `Cookie`, header tersebut bukan bagian dari objek `headers`. `Cookie` diwakili secara terpisah dalam objek `cookies`.

Objek `headers` berisi satu kolom untuk setiap header dalam permintaan. Nama header dikonversi ke ASCII-huruf kecil di objek acara, dan nama header harus ASCII-huruf kecil ketika ditambahkan oleh kode fungsi Anda. Ketika CloudFront Fungsi mengubah objek acara kembali ke permintaan HTTP, huruf pertama dari setiap kata dalam nama header dikapitalisasi, jika itu adalah huruf ASCII. CloudFront Fungsi tidak menerapkan perubahan apa pun pada simbol non-ASCII dalam nama header. Misalnya, `TÈst-header` akan menjadi `tÈst-header` di dalam fungsi. Simbol `È` non-ASCII tidak berubah.

Kata-kata dipisahkan oleh tanda hubung (`-`). Misalnya, jika kode fungsi Anda menambahkan header bernama `example-header-name`, CloudFront konversi ini ke `Example-Header-Name` dalam permintaan HTTP.

cookies

Sebuah objek yang mewakili cookie dalam permintaan (header `Cookie`).

Objek `cookies` berisi satu kolom untuk setiap cookie dalam permintaan.

Untuk informasi selengkapnya tentang struktur string kueri, header, dan cookie, lihat [Struktur untuk string kueri, header, atau cookie](#).

Misalnya, objek event, lihat [Contoh objek acara](#).

Objek respons

`response` Objek berisi representasi dari respon CloudFront-to-viewer HTTP. Dalam event objek yang diteruskan ke fungsi Anda, `response` objek mewakili respons CloudFront aktual terhadap permintaan penampil.

Jika kode fungsi Anda mengembalikan objek `response`, kode tersebut harus menggunakan struktur yang sama.

Objek `response` berisi kolom-kolom berikut:

statusCode

Kode status HTTP dari respons. Nilai ini adalah bilangan bulat, bukan string.

Fungsi Anda dapat menghasilkan atau memodifikasi `statusCode`.

statusDescription

Deskripsi status HTTP untuk respons. Jika kode fungsi Anda menghasilkan respons, kolom ini menjadi opsional.

headers

Sebuah objek yang merepresentasikan header HTTP dalam respons. Jika respons berisi header `Set-Cookie`, header tersebut bukan bagian dari objek `headers`. Cookie diwakili secara terpisah dalam objek `cookies`.

Objek `headers` berisi satu kolom untuk setiap header dalam respons. Nama header dikonversi ke huruf kecil di objek acara, dan nama header harus huruf kecil ketika ditambahkan oleh kode fungsi Anda. Ketika CloudFront Fungsi mengubah objek acara kembali menjadi respons HTTP, huruf pertama dari setiap kata dalam nama header dikapitalisasi. Kata-kata dipisahkan oleh tanda hubung (`-`). Misalnya, jika kode fungsi Anda menambahkan header bernama `example-header-name`, CloudFront konversi ini ke `Example-Header-Name` dalam respon HTTP.

cookies

Sebuah objek yang mewakili cookie dalam respons (header `Set-Cookie`).

Objek `cookies` berisi satu kolom untuk setiap cookie dalam respons.

body

Menambahkan body bidang adalah opsional, dan itu tidak akan ada di `response` objek kecuali Anda menentukannya dalam fungsi Anda. Fungsi Anda tidak memiliki akses ke badan asli yang dikembalikan oleh CloudFront cache atau asal. Jika Anda tidak menentukan body bidang dalam fungsi respons penampil, isi asli yang dikembalikan oleh CloudFront cache atau asal akan dikembalikan ke penampil.

Jika Anda CloudFront ingin mengembalikan badan khusus ke penampil, tentukan konten isi di data bidang, dan pengkodean badan di `encoding` bidang. Anda dapat menentukan encoding sebagai plain text (`"encoding": "text"`) atau sebagai Base64-encoded content (`"encoding": "base64"`).

Sebagai pintasan, Anda juga dapat menentukan isi isi langsung di body bidang (`"body": "<specify the body content here>"`). Ketika Anda melakukan ini, hilangkan encoding bidang data dan. CloudFront memperlakukan tubuh sebagai teks biasa dalam kasus ini.

encoding

Pengkodean untuk body konten (data bidang). Satu-satunya pengodean yang valid adalah text dan base64.

Jika Anda menentukan encoding sebagai base64 tetapi tubuh tidak valid base64, CloudFront mengembalikan kesalahan.

data

bodyKonten.

Untuk informasi selengkapnya tentang kode status dan isi yang dimodifikasi, lihat [Kode status dan badan](#).

Untuk informasi selengkapnya tentang struktur header dan cookie, lihat [Struktur untuk string kueri, header, atau cookie](#).

Misalnya, objek response, lihat [Contoh objek respon](#).

Kode status dan badan

Dengan CloudFront Fungsi, Anda dapat memperbarui kode status respons penampil, mengganti seluruh badan respons dengan yang baru, atau menghapus badan respons. Beberapa skenario umum untuk memperbarui respons penampil setelah mengevaluasi aspek respons dari CloudFront cache atau asal termasuk yang berikut:

- Mengubah status untuk menyetel kode status HTTP 200 dan membuat konten badan statis untuk kembali ke penampil.
- Mengubah status untuk menetapkan kode status HTTP 301 atau 302 untuk mengarahkan pengguna ke situs web lain.
- Memutuskan apakah akan melayani atau menjatuhkan tubuh respons pemirsa.

Note

Jika asal mengembalikan kesalahan HTTP 400 ke atas, CloudFront Fungsi tidak akan berjalan. Untuk mengetahui informasi selengkapnya, lihat [Pembatasan pada semua fungsi edge](#).

Saat Anda bekerja dengan respons HTTP, CloudFront Functions tidak memiliki akses ke badan respons. Anda dapat mengganti isi dengan mengaturnya ke nilai yang diinginkan, atau Anda dapat menghapus tubuh dengan mengatur nilai menjadi kosong. Jika Anda tidak memperbarui bidang isi dalam fungsi Anda, isi asli yang dikembalikan oleh CloudFront cache atau asal dikembalikan ke penampil.

Tip

Saat menggunakan CloudFront Functions untuk mengganti body, pastikan untuk menyelaraskan header yang sesuai, seperti `content-encoding`, atau `content-type` `content-length`, ke isi isi baru.

Misalnya, jika CloudFront asal atau cache kembali `content-encoding: gzip` tetapi fungsi respons penampil menyetel isi teks biasa, fungsi tersebut juga perlu mengubah `content-encoding` dan `content-type` header yang sesuai.

Jika CloudFront Fungsi Anda dikonfigurasi untuk mengembalikan kesalahan HTTP 400 atau lebih tinggi, penampil Anda tidak akan melihat [halaman kesalahan kustom](#) yang telah Anda tentukan untuk kode status yang sama.

Struktur untuk string kueri, header, atau cookie

String kueri, header, dan cookie berbagi struktur yang sama. String kueri dapat muncul dalam permintaan. Header muncul dalam permintaan dan tanggapan. Cookie muncul dalam permintaan dan tanggapan.

Setiap string kueri, header, atau cookie adalah kolom yang unik dalam objek `queryString`, `headers`, atau `cookies` induk. Nama bidang adalah nama string kueri, header, atau cookie. Setiap kolom berisi properti `value` dengan nilai string kueri, header, atau cookie.

Daftar Isi

- [Nilai string kueri atau objek string kueri](#)
- [Pertimbangan khusus untuk header](#)
- [Duplikasi string kueri, header, dan cookie \(multiValue susunan\)](#)
- [Atribut cookie](#)

Nilai string kueri atau objek string kueri

Sebuah fungsi dapat mengembalikan nilai string query selain objek string query. Nilai string kueri dapat digunakan untuk mengatur parameter string kueri dalam urutan kustom apa pun.

Example Contoh

Untuk memodifikasi string kueri dalam kode fungsi Anda, gunakan kode seperti berikut ini.

```
var request = event.request;
request.querystring =
  'ID=42&Exp=1619740800&TTL=1440&NoValue=&querymv=val1&querymv=val2,val3';
```

Pertimbangan khusus untuk header

Hanya untuk header, nama header dikonversi ke huruf kecil di objek acara, dan nama header harus huruf kecil ketika ditambahkan oleh kode fungsi Anda. Ketika CloudFront Fungsi mengubah objek acara kembali menjadi permintaan atau respons HTTP, huruf pertama dari setiap kata dalam nama header dikapitalisasi. Kata-kata dipisahkan oleh tanda hubung (-). Misalnya, jika kode fungsi Anda menambahkan header bernama `example-header-name`, CloudFront mengonversinya menjadi `Example-Header-Name` permintaan atau respons HTTP.

Example Contoh

Pertimbangkan Host header berikut dalam permintaan HTTP.

```
Host: video.example.com
```

Header ini direpresentasikan sebagai berikut dalam objek request:

```
"headers": {
  "host": {
    "value": "video.example.com"
  }
}
```

Untuk mengakses header Host dalam kode fungsi Anda, gunakan kode seperti berikut:

```
var request = event.request;
var host = request.headers.host.value;
```

Untuk menambah atau memodifikasi header dalam kode fungsi Anda, gunakan kode seperti berikut (kode ini menambahkan header bernama X-Custom-Header dengan nilai `example value`):

```
var request = event.request;
request.headers['x-custom-header'] = {value: 'example value'};
```

Duplikasi string kueri, header, dan cookie (**multiValue** susunan)

Permintaan atau respons HTTP dapat berisi lebih dari satu string kueri, header, atau cookie dengan nama yang sama. Dalam hal ini, string kueri, header, atau cookie duplikat dikumpulkan dalam satu kolom di objek `request` atau `response`, tetapi kolom ini berisi properti tambahan bernama `multiValue`. Properti `multiValue` berisi susunan dengan nilai-nilai masing-masing string kueri, header, atau cookie duplikat.

Example Contoh

Pertimbangkan permintaan HTTP dengan Accept header berikut.

```
Accept: application/json
Accept: application/xml
Accept: text/html
```

Header ini direpresentasikan sebagai berikut dalam `request` objek.

```
"headers": {
  "accept": {
    "value": "application/json",
    "multiValue": [
      {
        "value": "application/json"
      },
      {
        "value": "application/xml"
      },
      {
        "value": "text/html"
      }
    ]
  }
}
```

 Note

Nilai header pertama (dalam hal ini, `application/json`) diulang di kedua `multiValue` properti `value` dan. Ini memungkinkan Anda untuk mengakses semua nilai dengan perulangan melalui `multiValue` susunan.

Jika kode fungsi Anda memodifikasi string kueri, header, atau cookie yang memiliki `multiValue` array, CloudFront Fungsi menggunakan aturan berikut untuk menerapkan perubahan:

1. Jika `multiValue` susunan ada dan memiliki modifikasi, modifikasi tersebut akan diterapkan. Elemen pertama dalam properti `value` diabaikan.
2. Jika tidak, modifikasi apa pun pada properti `value` diterapkan, dan nilai berikutnya (jika ada) tetap sama.

Properti `multiValue` digunakan hanya ketika permintaan HTTP atau respons berisi string kueri, header, atau cookie duplikat dengan nama yang sama, seperti yang ditunjukkan dalam contoh sebelumnya. Namun, jika ada beberapa nilai dalam satu string kueri, header, atau cookie, properti `multiValue` tidak akan digunakan.

Example Contoh

Pertimbangkan permintaan dengan satu `Accept` header yang berisi tiga nilai.

```
Accept: application/json, application/xml, text/html
```

Header ini direpresentasikan sebagai berikut dalam `request` objek.

```
"headers": {  
  "accept": {  
    "value": "application/json, application/xml, text/html"  
  }  
}
```

Atribut cookie

Dalam header `Set-Cookie` dalam respons HTTP, header berisi pasangan nama-nilai untuk cookie dan opsi satu set atribut dipisahkan oleh titik koma.

Example Contoh

```
Set-Cookie: cookie1=val1; Secure; Path=/; Domain=example.com; Expires=Wed, 05 Apr 2021
07:28:00 GMT
```

Di objek response, atribut ini direpresentasikan dalam properti `attributes` dari kolom cookie. Sebagai contoh, header Set-Cookie sebelumnya direpresentasikan sebagai berikut:

```
"cookie1": {
  "value": "val1",
  "attributes": "Secure; Path=/; Domain=example.com; Expires=Wed, 05 Apr 2021
07:28:00 GMT"
}
```

Contoh objek respon

Contoh berikut menunjukkan response objek - output dari fungsi respons penampil - di mana tubuh telah digantikan oleh fungsi respons penampil.

```
{
  "response": {
    "statusCode": 200,
    "statusDescription": "OK",
    "headers": {
      "date": {
        "value": "Mon, 04 Apr 2021 18:57:56 GMT"
      },
      "server": {
        "value": "unicorn/19.9.0"
      },
      "access-control-allow-origin": {
        "value": "*"
      },
      "access-control-allow-credentials": {
        "value": "true"
      },
      "content-type": {
        "value": "text/html"
      },
      "content-length": {
        "value": "86"
      }
    }
  }
}
```

```

    }
  },
  "cookies": {
    "ID": {
      "value": "id1234",
      "attributes": "Expires=Wed, 05 Apr 2021 07:28:00 GMT"
    },
    "Cookie1": {
      "value": "val1",
      "attributes": "Secure; Path=/; Domain=example.com; Expires=Wed, 05 Apr 2021
07:28:00 GMT",
      "multiValue": [
        {
          "value": "val1",
          "attributes": "Secure; Path=/; Domain=example.com; Expires=Wed, 05 Apr 2021
07:28:00 GMT"
        },
        {
          "value": "val2",
          "attributes": "Path=/cat; Domain=example.com; Expires=Wed, 10 Jan 2021
07:28:00 GMT"
        }
      ]
    }
  },
  // Adding the body field is optional and it will not be present in the response
  // object
  // unless you specify it in your function.
  // Your function does not have access to the original body returned by the
  // CloudFront
  // cache or origin.
  // If you don't specify the body field in your viewer response function, the
  // original
  // body returned by the CloudFront cache or origin is returned to viewer.

  "body": {
    "encoding": "text",
    "data": "<!DOCTYPE html><html><body><p>Here is your custom content.</p></body></
html>"
  }
}

```

Contoh objek acara

Contoh berikut menunjukkan objek event lengkap. Ini adalah contoh pemanggilan untuk distribusi standar, dan bukan untuk distribusi multi-penyewa. Untuk distribusi multi-penyewa, endpoint bidang digunakan sebagai ganti Nilai dari `distributionDomainName` adalah nama CloudFront domain (misalnya, `d111111abcdef8.cloudfront.net`) dari endpoint grup koneksi yang terkait dengan acara tersebut.

Note

Objek event adalah masukan untuk fungsi Anda. Fungsi Anda hanya mengembalikan hanya objek request atau response, bukan objek event lengkap.

```
{
  "version": "1.0",
  "context": {
    "distributionDomainName": "d111111abcdef8.cloudfront.net",
    "distributionId": "EDFDVBD6EXAMPLE",
    "eventType": "viewer-response",
    "requestId": "EXAMPLEntjQpEXAMPLE_SG5Z-EXAMPLEPmPfEXAMPLEu3EqEXAMPLE=="
  },
  "viewer": {"ip": "198.51.100.11"},
  "request": {
    "method": "GET",
    "uri": "/media/index.mpd",
    "queryString": {
      "ID": {"value": "42"},
      "Exp": {"value": "1619740800"},
      "TTL": {"value": "1440"},
      "NoValue": {"value": ""},
      "querymv": {
        "value": "val1",
        "multiValue": [
          {"value": "val1"},
          {"value": "val2,val3"}
        ]
      }
    }
  },
  "headers": {
    "host": {"value": "video.example.com"},
  }
}
```



```

    "user-agent": {"value": "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:83.0)
    Gecko/20100101 Firefox/83.0"},
    "accept": {
      "value": "application/json",
      "multiValue": [
        {"value": "application/json"},
        {"value": "application/xml"},
        {"value": "text/html"}
      ]
    },
    "accept-language": {"value": "en-GB,en;q=0.5"},
    "accept-encoding": {"value": "gzip, deflate, br"},
    "origin": {"value": "https://website.example.com"},
    "referer": {"value": "https://website.example.com/videos/12345678?
action=play"},
    "cloudfront-viewer-country": {"value": "GB"}
  },
  "cookies": {
    "Cookie1": {"value": "value1"},
    "Cookie2": {"value": "value2"},
    "cookie_consent": {"value": "true"},
    "cookiemv": {
      "value": "value3",
      "multiValue": [
        {"value": "value3"},
        {"value": "value4"}
      ]
    }
  }
},
"response": {
  "statusCode": 200,
  "statusDescription": "OK",
  "headers": {
    "date": {"value": "Mon, 04 Apr 2021 18:57:56 GMT"},
    "server": {"value": "unicorn/19.9.0"},
    "access-control-allow-origin": {"value": "*"},
    "access-control-allow-credentials": {"value": "true"},
    "content-type": {"value": "application/json"},
    "content-length": {"value": "701"}
  },
  "cookies": {
    "ID": {
      "value": "id1234",

```

```

        "attributes": "Expires=Wed, 05 Apr 2021 07:28:00 GMT"
    },
    "Cookie1": {
        "value": "val1",
        "attributes": "Secure; Path=/; Domain=example.com; Expires=Wed, 05 Apr
2021 07:28:00 GMT",
        "multiValue": [
            {
                "value": "val1",
                "attributes": "Secure; Path=/; Domain=example.com; Expires=Wed,
05 Apr 2021 07:28:00 GMT"
            },
            {
                "value": "val2",
                "attributes": "Path=/cat; Domain=example.com; Expires=Wed, 10
Jan 2021 07:28:00 GMT"
            }
        ]
    }
}
}
}
}
}

```

JavaScript fitur runtime untuk Fungsi CloudFront

Lingkungan JavaScript runtime CloudFront Functions sesuai dengan [ECMAScript \(ES\) versi 5.1](#) dan juga mendukung beberapa fitur ES versi 6 hingga 12.

Untuk sebagian besar up-to-date fitur, kami menyarankan Anda menggunakan JavaScript runtime 2.0.

Fitur JavaScript runtime 2.0 memiliki perubahan berikut dibandingkan dengan 1.0:

- Metode modul buffer tersedia
- Metode prototipe string non-standar berikut tidak tersedia:
 - `String.prototype.bytesFrom()`
 - `String.prototype.fromBytes()`
 - `String.prototype.fromUTF8()`
 - `String.prototype.toBytes()`
 - `String.prototype.toUTF8()`

- Modul kriptografi memiliki perubahan berikut:
 - `hash.digest()`— Jenis pengembalian diubah menjadi `Buffer` jika tidak ada pengkodean yang disediakan
 - `hmac.digest()`— Jenis pengembalian diubah menjadi `Buffer` jika tidak ada pengkodean yang disediakan
- Untuk informasi selengkapnya tentang fitur baru tambahan, lihat [JavaScript fitur runtime 2.0 untuk Fungsi CloudFront](#).

Topik

- [JavaScript fitur runtime 1.0 untuk Fungsi CloudFront](#)
- [JavaScript fitur runtime 2.0 untuk Fungsi CloudFront](#)

JavaScript fitur runtime 1.0 untuk Fungsi CloudFront

Lingkungan JavaScript runtime CloudFront Functions sesuai dengan [ECMAScript \(ES\) versi 5.1](#) dan juga mendukung beberapa fitur ES versi 6 hingga 9. Lingkungan ini juga menyediakan beberapa metode standar yang bukan bagian dari spesifikasi ES.

Topik berikut mencantumkan semua fitur bahasa yang didukung.

Topik

- [Fitur inti](#)
- [Objek primitif](#)
- [Objek bawaan](#)
- [Jenis kesalahan](#)
- [Global](#)
- [Modul bawaan](#)
- [Fitur yang dibatasi](#)

Fitur inti

Mendukung fitur inti ES berikut.

Jenis

Mendukung semua jenis ES 5.1. Ini termasuk nilai, angka, string, objek, susunan, fungsi, konstruktor fungsi, dan ekspresi reguler Boolean.

Operator

Mendukung Semua operator ES 5.1 didukung.

Mendukung operator eksponensial ES 7 (**).

Pernyataan

Note

Tidak mendukung pernyataan `const` dan `let`.

Mendukung pernyataan ES 5.1 berikut:

- `break`
- `catch`
- `continue`
- `do-while`
- `else`
- `finally`
- `for`
- `for-in`
- `if`
- `return`
- `switch`
- `throw`
- `try`
- `var`
- `while`
- Pernyataan berlabel

Literal

Literal template ES 6 didukung: string multiline, interpolasi ekspresi, dan templat bersarang.

Fungsi

Mendukung semua fitur fungsi ES 5.1.

Mendukung fungsi panah ES 6 didukung, dan sintaks parameter ES 6 sintaks parameter istirahat.

Unicode

Sumber teks dan literal string dapat berisi karakter Unicode yang dikodekan. Juga mendukung unicode code point escape sequence enam karakter (misalnya, `\uXXXX`).

Mode ketat

Fungsi beroperasi dalam mode ketat secara default, sehingga Anda tidak perlu menambahkan pernyataan `use strict` dalam kode fungsi Anda. Ini tidak dapat diubah.

Objek primitif

Mendukung objek primitif ES berikut.

Objek

Mendukung metode ES 5.1 berikut pada objek:

- `create` (tanpa daftar properti)
- `defineProperties`
- `defineProperty`
- `freeze`
- `getOwnPropertyDescriptor`
- `getOwnPropertyNames`
- `getPrototypeOf`
- `hasOwnProperty`
- `isExtensible`
- `isFrozen`
- `prototype.isPrototypeOf`

- `isSealed`
- `keys`
- `preventExtensions`
- `prototype.propertyIsEnumerable`
- `seal`
- `prototype.toString`
- `prototype.valueOf`

Mendukung metode ES 6 berikut pada objek:

- `assign`
- `is`
- `prototype.setPrototypeOf`

Mendukung metode ES 8 berikut pada objek:

- `entries`
- `values`

String

Mendukung metode ES 5.1 berikut pada string:

- `fromCharCode`
- `prototype.charAt`
- `prototype.concat`
- `prototype.indexOf`
- `prototype.lastIndexOf`
- `prototype.match`
- `prototype.replace`
- `prototype.search`
- `prototype.slice`
- `prototype.split`
- `prototype.substr`
- `prototype.substring`
- `prototype.toLowerCase`

- `prototype.trim`
- `prototype.toUpperCase`

Mendukung metode ES 6 berikut pada string:

- `fromCodePoint`
- `prototype.codePointAt`
- `prototype.endsWith`
- `prototype.includes`
- `prototype.repeat`
- `prototype.startsWith`

Mendukung metode ES 8 berikut pada string:

- `prototype.padStart`
- `prototype.padEnd`

Mendukung metode ES 9 berikut pada string:

- `prototype.trimStart`
- `prototype.trimEnd`

Mendukung metode tidak standar berikut pada string:

- `prototype.bytesFrom(array | string, encoding)`

Menciptakan string byte dari susunan oktet atau string yang dikodekan. Opsi pengkodean string adalah hex, base64, dan base64url.

- `prototype.fromBytes(start[, end])`

Menciptakan string Unicode dari string byte di mana setiap byte diganti dengan titik kode Unicode yang sesuai.

- `prototype.fromUTF8(start[, end])`

Menciptakan string Unicode dari UTF-8 string byte yang dikodekan. Jika pengkodean salah, akan muncul `null`.

- `prototype.toBytes(start[, end])`

Menciptakan string byte dari string Unicode. Semua karakter harus dalam rentang [0,255]. Jika tidak, akan muncul `null`.

- `prototype.toUTF8(start[, end])`

Menciptakan UTF-8 string byte yang dikodekan dari string Unicode.

Nomor

Mendukung semua metode ES 5.1 pada nomor.

Mendukung metode ES 6 berikut pada nomor:

- `isFinite`
- `isInteger`
- `isNaN`
- `isSafeInteger`
- `parseFloat`
- `parseInt`
- `prototype.toExponential`
- `prototype.toFixed`
- `prototype.toPrecision`
- `EPSILON`
- `MAX_SAFE_INTEGER`
- `MAX_VALUE`
- `MIN_SAFE_INTEGER`
- `MIN_VALUE`
- `NEGATIVE_INFINITY`
- `NaN`
- `POSITIVE_INFINITY`

Objek bawaan

Mendukung objek ES bawaan berikut.

Matematika

Mendukung semua metode matematika ES 5.1.

 Note

Di lingkungan runtime CloudFront Functions, `Math.random()` implementasi menggunakan `arc4random` OpenBSD yang diunggulkan dengan stempel waktu saat fungsi berjalan.

Mendukung metode matematika ES 6 berikut:

- `acosh`
- `asinh`
- `atanh`
- `cbrt`
- `clz32`
- `cosh`
- `expm1`
- `fround`
- `hypot`
- `imul`
- `log10`
- `log1p`
- `log2`
- `sign`
- `sinh`
- `tanh`
- `trunc`
- `E`
- `LN10`
- `LN2`
- `LOG10E`
- `LOG2E`
- `PI`
- `SQRT1_2`

- SQRT2

Tanggal

Mendukung semua fitur ES 5.1 Date.

Note

Untuk alasan keamanan, Date selalu mengembalikan nilai yang sama—waktu mulai fungsi ini—saat menjalankan fungsi tunggal. Untuk informasi selengkapnya, lihat [Fitur yang dibatasi](#).

Fungsi

Mendukung metode apply, bind, dan call.

Tidak mendukung konstruktor fungsi.

Ekspresi reguler

Mendukung semua fitur ekspresi reguler ES 5.1. Bahasa ekspresi reguler adalah kompatibel dengan Perl. Mendukung kelompok penangkap bernama ES 9.

JSON

Mendukung semua fitur JSON ES 5.1 JSON, termasuk parse dan stringify.

Susunan

Mendukung metode ES 5.1 berikut pada susunan:

- isArray
- prototype.concat
- prototype.every
- prototype.filter
- prototype.forEach
- prototype.indexOf
- prototype.join
- prototype.lastIndexOf
- prototype.map
- prototype.pop

- `prototype.push`
- `prototype.reduce`
- `prototype.reduceRight`
- `prototype.reverse`
- `prototype.shift`
- `prototype.slice`
- `prototype.some`
- `prototype.sort`
- `prototype.splice`
- `prototype.unshift`

Mendukung metode ES 6 berikut pada susunan:

- `of`
- `prototype.copyWithIn`
- `prototype.fill`
- `prototype.find`
- `prototype.findIndex`

Mendukung metode ES 7 berikut pada susunan:

- `prototype.includes`

Susunan yang dijeniskan

Mendukung susunan yang diketik ES 6 berikut:

- `Int8Array`
- `Uint8Array`
- `Uint8ClampedArray`
- `Int16Array`
- `Uint16Array`
- `Int32Array`
- `Uint32Array`
- `Float32Array`
- `Float64Array`

- `prototype.copyWithIn`
- `prototype.fill`
- `prototype.join`
- `prototype.set`
- `prototype.slice`
- `prototype.subarray`
- `prototype.toString`

ArrayBuffer

Mendukung metode pada `ArrayBuffer` berikut:

- `prototype.isView`
- `prototype.slice`

Janji

Mendukung metode janji berikut:

- `reject`
- `resolve`
- `prototype.catch`
- `prototype.finally`
- `prototype.then`

Kripto

Modul kriptografi menyediakan hashing standar dan pembantu kode autentikasi pesan berbasis hash (HMAC). Anda dapat memuat modul menggunakan `require('crypto')`. Modul ini memperlihatkan metode berikut yang berperilaku persis seperti rekan-rekan Node.js mereka:

- `createHash(algorithm)`
- `hash.update(data)`
- `hash.digest([encoding])`
- `createHmac(algorithm, secret key)`
- `hmac.update(data)`
- `hmac.digest([encoding])`

Untuk informasi lebih lanjut, lihat [Kripto \(hash dan HMAC\)](#) di bagian modul bawaan.

Konsol

Ini adalah objek pembantu untuk debugging. Ini hanya mendukung metode `log()`, untuk merekam pesan log.

Note

CloudFront Fungsi tidak mendukung sintaks koma, seperti `console.log('a', 'b')`. Sebagai gantinya, gunakan `console.log('a' + ' ' + 'b')` formatnya.

Jenis kesalahan

Mendukung objek kesalahan berikut:

- `Error`
- `EvalError`
- `InternalError`
- `MemoryError`
- `RangeError`
- `ReferenceError`
- `SyntaxError`
- `TypeError`
- `URIError`

Global

Mendukung objek `globalThis`.

Mendukung fungsi global ES 5.1 berikut:

- `decodeURI`
- `decodeURIComponent`
- `encodeURI`
- `encodeURIComponent`
- `isFinite`

- `isNaN`
- `parseFloat`
- `parseInt`

Mendukung konstanta global berikut:

- `NaN`
- `Infinity`
- `undefined`

Modul bawaan

Mendukung modul bawaan berikut.

Modul

- [Kripto \(hash dan HMAC\)](#)
- [String kueri](#)

Kripto (hash dan HMAC)

Modul kriptografi (`crypto`) menyediakan hashing standar dan pembantu kode autentikasi pesan berbasis hash (HMAC). Anda dapat memuat modul menggunakan `require('crypto')`. Modul ini menyediakan metode berikut yang berperilaku persis seperti rekan-rekan Node.js mereka.

Metode hashing

`crypto.createHash(algorithm)`

Menciptakan dan mengembalikan objek hash yang dapat digunakan untuk menghasilkan hash digests menggunakan algoritme yang diberikan: `md5`, `sha1`, atau `sha256`.

`hash.update(data)`

Update konten hash dengan data yang tersedia.

`hash.digest([encoding])`

Menghitung digest dari semua data yang diteruskan menggunakan `hash.update()`. Pengkodean dapat berupa `hex`, `base64`, atau `base64url`.

Metode HMAC

```
crypto.createHmac(algorithm, secret key)
```

Menciptakan dan mengembalikan objek HMAC yang menggunakan `algorithm` dan `secret key` yang tersedia. Algoritma dapat berupa md5, sha1, atau sha256.

```
hmac.update(data)
```

Memperbarui konten HMAC dengan data yang tersedia.

```
hmac.digest([encoding])
```

Menghitung digest dari semua data yang diteruskan menggunakan `hmac.update()`. Pengkodean dapat berupa hex, base64, atau base64url.

String kueri

Note

[Objek acara CloudFront Functions](#) secara otomatis mem-parsing string kueri URL untuk Anda. Itu berarti bahwa dalam kebanyakan kasus Anda tidak perlu menggunakan modul ini.

Modul string kueri (`querystring`) menyediakan metode untuk mengurai dan memformat string kueri URL. Anda dapat memuat modul menggunakan `require('querystring')`. Modul ini menyediakan metode berikut.

```
querystring.escape(string)
```

Mengkodekan URL string yang tersedia, mengembalikan string kueri yang lolos. Metode ini digunakan oleh `querystring.stringify()` dan tidak boleh digunakan secara langsung.

```
querystring.parse(string[, separator[, equal[, options]]])
```

Mengurai string kueri (`string`) dan mengembalikan objek.

Parameter `separator` adalah substring untuk membatasi pasangan kunci dan nilai dalam string kueri. Secara default, itu adalah `&`.

Parameter `equal` adalah substring untuk membatasi kunci dan nilai dalam string kueri. Secara default, itu adalah `=`.

Parameter options adalah objek dengan kunci berikut:

`decodeURIComponent` *function*

Sebuah fungsi untuk memecahkan kode karakter persentase dikodekan dalam string kueri.

Secara default, itu adalah `querystring.unescape()`.

`maxKeys` *number*

Jumlah kunci maksimum untuk diurai. Secara default, itu adalah 1000. Gunakan nilai 0 untuk menghapus pembatasan untuk menghitung kunci.

Secara default, karakter persentase dikodekan dalam string kueri diasumsikan menggunakan pengkodean UTF-8. Urutan UTF-8 tidak valid diganti dengan karakter pengganti U+FFFD.

Misalnya, untuk string kueri berikut:

```
'name=value&abc=xyz&abc=123'
```

Nilai `querystring.parse()` yang dikembalikan adalah:

```
{
  name: 'value',
  abc: ['xyz', '123']
}
```

`querystring.decode()` adalah alias untuk `querystring.parse()`.

`querystring.stringify(object[, separator[, equal[, options]])`

Menyerialisasi object dan mengembalikan string kueri.

Parameter `separator` adalah substring untuk membatasi pasangan kunci dan nilai dalam string kueri. Secara default, itu adalah `&`.

Parameter `equal` adalah substring untuk membatasi kunci dan nilai dalam string kueri. Secara default, itu adalah `=`.

Parameter options adalah objek dengan kunci berikut:

`encodeURIComponent` *function*

Fungsi yang digunakan untuk mengonversi karakter URL yang tidak aman untuk pengkodean persentase dalam string kueri. Secara default, itu adalah `querystring.escape()`.

Secara default, karakter yang memerlukan pengkodean persentase dalam string kueri dikodekan sebagai UTF-8. Untuk menggunakan pengkodean yang berbeda, tentukan opsi `encodeURIComponent`.

Misalnya, gunakan kode berikut:

```
querystring.stringify({ name: 'value', abc: ['xyz', '123'], anotherName: '' });
```

Nilai yang dikembalikan adalah:

```
'name=value&abc=xyz&abc=123&anotherName='
```

`querystring.encode()` adalah alias untuk `querystring.stringify()`.

`querystring.unescape(string)`

Mendekode karakter yang dikodekan persentase URL dalam `string` yang tersedia, mengembalikan string kueri yang tidak lolos. Metode ini digunakan oleh `querystring.parse()` dan tidak boleh digunakan secara langsung.

Fitur yang dibatasi

Fitur JavaScript bahasa berikut tidak didukung atau dibatasi karena masalah keamanan.

Evaluasi kode dinamis

Evaluasi kode dinamis tidak didukung. Konstruktor `eval()` dan `Function` mengalami kesalahan jika dicoba. Misalnya, `const sum = new Function('a', 'b', 'return a + b')` mengalami kesalahan.

Timer

Fungsi `setTimeout()`, `setImmediate()`, dan `clearTimeout()` tidak didukung. Tidak ada ketentuan untuk menunda atau menghasilkan dalam fungsi jalankan. Fungsi Anda harus dijalankan secara serentak sampai selesai.

Tanggal dan stempel waktu

Untuk alasan keamanan, tidak ada akses ke timer beresolusi tinggi. Semua metode `Date` untuk meminta waktu saat ini selalu mengembalikan nilai yang sama saat menjalankan fungsi tunggal. Stempel waktu yang dikembalikan adalah waktu fungsi mulai berjalan. Akibatnya, Anda tidak dapat mengukur waktu berlalu dalam fungsi Anda.

Akses sistem file

Tidak ada akses sistem file. Misalnya, tidak ada modul `fs` untuk akses sistem file seperti yang ada di Node.js.

Akses proses

Tidak ada akses proses. Misalnya, tidak ada objek `process` global untuk memproses akses informasi seperti yang ada di Node.js.

Variabel-variabel lingkungan

Tidak ada akses ke variabel lingkungan.

Sebagai gantinya, Anda dapat menggunakan CloudFront KeyValueCollection untuk membuat datastore terpusat dari pasangan kunci-nilai untuk Fungsi Anda. CloudFront CloudFront KeyValueCollection memungkinkan pembaruan dinamis ke data konfigurasi Anda tanpa perlu menerapkan perubahan kode. Anda harus menggunakan [JavaScript runtime 2.0](#) untuk menggunakannya CloudFront KeyValueCollection. Untuk informasi selengkapnya, lihat [Amazon CloudFront KeyValueCollection](#).

Akses jaringan

Tidak ada dukungan untuk panggilan jaringan. Misalnya, XHR, HTTP (S), dan soket tidak didukung.

JavaScript fitur runtime 2.0 untuk Fungsi CloudFront

Lingkungan JavaScript runtime CloudFront Functions sesuai dengan [ECMAScript \(ES\) versi 5.1](#) dan juga mendukung beberapa fitur ES versi 6 hingga 12. Lingkungan ini juga menyediakan beberapa metode standar yang bukan bagian dari spesifikasi ES. Topik berikut mencantumkan semua fitur yang didukung dalam runtime ini.

Topik

- [Fitur inti](#)
- [Objek primitif](#)
- [Objek bawaan](#)
- [Jenis kesalahan](#)
- [Global](#)
- [Modul bawaan](#)

- [Fitur yang dibatasi](#)

Fitur inti

Mendukung fitur inti ES berikut.

Jenis

Mendukung semua jenis ES 5.1. Ini termasuk nilai boolean, angka, string, objek, array, fungsi, dan ekspresi reguler.

Operator

Mendukung Semua operator ES 5.1 didukung.

Mendukung operator eksponensial ES 7 (**).

Pernyataan

Mendukung pernyataan ES 5.1 berikut:

- `break`
- `catch`
- `continue`
- `do-while`
- `else`
- `finally`
- `for`
- `for-in`
- `if`
- `label`
- `return`
- `switch`
- `throw`
- `try`
- `var`
- `while`

Pernyataan ES 6 berikut didukung:

- `const`
- `let`

Pernyataan ES 8 berikut didukung:

- `async`
- `await`

 Note

`async`, `await`, `const`, dan `let` didukung dalam JavaScript runtime 2.0. `await` hanya dapat digunakan di dalam `async` fungsi. `async` argumen dan penutupan tidak didukung.

Literal

Literal template ES 6 didukung: string multiline, interpolasi ekspresi, dan templat bersarang.

Fungsi

Mendukung semua fitur fungsi ES 5.1.

Mendukung fungsi panah ES 6 didukung, dan sintaks parameter ES 6 sintaks parameter istirahat.

Unicode

Sumber teks dan literal string dapat berisi karakter Unicode yang dikodekan. Juga mendukung unicode code point escape sequence enam karakter (misalnya, `\uXXXX`).

Mode ketat

Fungsi beroperasi dalam mode ketat secara default, sehingga Anda tidak perlu menambahkan pernyataan `use strict` dalam kode fungsi Anda. Ini tidak dapat diubah.

Objek primitif

Mendukung objek primitif ES berikut.

Objek

Mendukung metode ES 5.1 berikut pada objek:

- `Object.create()` (tanpa daftar properti)
- `Object.defineProperties()`
- `Object.defineProperty()`
- `Object.freeze()`
- `Object.getOwnPropertyDescriptor()`
- `Object.getOwnPropertyDescriptors()`
- `Object.getOwnPropertyNames()`
- `Object.getPrototypeOf()`
- `Object.isExtensible()`
- `Object.isFrozen()`
- `Object.isSealed()`
- `Object.keys()`
- `Object.preventExtensions()`
- `Object.seal()`

Mendukung metode ES 6 berikut pada objek:

- `Object.assign()`

Mendukung metode ES 8 berikut pada objek:

- `Object.entries()`
- `Object.values()`

Metode prototipe ES 5.1 berikut pada objek didukung:

- `Object.prototype.hasOwnProperty()`
- `Object.prototype.isPrototypeOf()`
- `Object.prototype.propertyIsEnumerable()`
- `Object.prototype.toString()`
- `Object.prototype.valueOf()`

Metode prototipe ES 6 berikut pada objek didukung:

- `Object.prototype.is()`
- `Object.prototype.setPrototypeOf()`

String

Mendukung metode ES 5.1 berikut pada string:

- `String.fromCharCode()`

Mendukung metode ES 6 berikut pada string:

- `String.fromCodePoint()`

Metode prototipe ES 5.1 berikut pada string didukung:

- `String.prototype.charAt()`
- `String.prototype.concat()`
- `String.prototype.indexOf()`
- `String.prototype.lastIndexOf()`
- `String.prototype.match()`
- `String.prototype.replace()`
- `String.prototype.search()`
- `String.prototype.slice()`
- `String.prototype.split()`
- `String.prototype.substr()`
- `String.prototype.substring()`
- `String.prototype.toLowerCase()`
- `String.prototype.trim()`
- `String.prototype.toUpperCase()`

Metode prototipe ES 6 berikut pada string didukung:

- `String.prototype.codePointAt()`
- `String.prototype.endsWith()`
- `String.prototype.includes()`
- `String.prototype.repeat()`
- `String.prototype.startsWith()`

Metode prototipe ES 8 berikut pada string didukung:

- `String.prototype.padStart()`

- `String.prototype.padEnd()`

Metode prototipe ES 9 berikut pada string didukung:

- `String.prototype.trimStart()`
- `String.prototype.trimEnd()`

Metode prototipe ES 12 berikut pada string didukung:

- `String.prototype.replaceAll()`



Note

`String.prototype.replaceAll()` baru di JavaScript runtime 2.0.

Bilangan

SEMUA nomor ES 5 didukung.

Properti ES 6 berikut pada angka didukung:

- `Number.EPSILON`
- `Number.MAX_SAFE_INTEGER`
- `Number.MIN_SAFE_INTEGER`
- `Number.MAX_VALUE`
- `Number.MIN_VALUE`
- `Number.NaN`
- `Number.NEGATIVE_INFINITY`
- `Number.POSITIVE_INFINITY`

Mendukung metode ES 6 berikut pada nomor:

- `Number.isFinite()`
- `Number.isInteger()`
- `Number.isNaN()`
- `Number.isSafeInteger()`
- `Number.parseInt()`
- `Number.parseFloat()`

Metode prototipe ES 5.1 berikut pada angka didukung:

- `Number.prototype.toExponential()`
- `Number.prototype.toFixed()`
- `Number.prototype.toPrecision()`

Pemisah numerik ES 12 didukung.

 Note

Pemisah numerik ES 12 baru di JavaScript runtime 2.0.

Objek bawaan

Mendukung objek ES bawaan berikut.

Matematika

Mendukung semua metode matematika ES 5.1.

 Note

Di lingkungan runtime CloudFront Functions, `Math.random()` implementasi menggunakan `arc4random` OpenBSD yang diunggulkan dengan stempel waktu saat fungsi berjalan.

Properti matematika ES 6 berikut didukung:

- `Math.E`
- `Math.LN10`
- `Math.LN2`
- `Math.LOG10E`
- `Math.LOG2E`
- `Math.PI`
- `Math.SQRT1_2`
- `Math.SQRT2`

Mendukung metode matematika ES 6 berikut:

- `Math.abs()`
- `Math.acos()`
- `Math.acosh()`
- `Math.asin()`
- `Math.asinh()`
- `Math.atan()`
- `Math.atan2()`
- `Math.atanh()`
- `Math.cbrt()`
- `Math.ceil()`
- `Math.clz32()`
- `Math.cos()`
- `Math.cosh()`
- `Math.exp()`
- `Math.expm1()`
- `Math.floor()`
- `Math.fround()`
- `Math.hypot()`
- `Math.imul()`
- `Math.log()`
- `Math.log1p()`
- `Math.log2()`
- `Math.log10()`
- `Math.max()`
- `Math.min()`
- `Math.pow()`
- `Math.random()`
- `Math.round()`
- `Math.sign()`
- `Math.sinh()`

- `Math.sin()`
- `Math.sqrt()`
- `Math.tan()`
- `Math.tanh()`
- `Math.trunc()`

Tanggal

Mendukung semua fitur ES 5.1 Date.

Note

Untuk alasan keamanan, Date selalu mengembalikan nilai yang sama—waktu mulai fungsi ini—saat menjalankan fungsi tunggal. Untuk informasi selengkapnya, lihat [Fitur yang dibatasi](#).

Fungsi

Metode prototipe ES 5.1 berikut didukung:

- `Function.prototype.apply()`
- `Function.prototype.bind()`
- `Function.prototype.call()`

Tidak mendukung konstruktor fungsi.

Ekspresi reguler

Mendukung semua fitur ekspresi reguler ES 5.1. Bahasa ekspresi reguler adalah kompatibel dengan Perl.

Properti pengakses prototipe ES 5.1 berikut didukung:

- `RegExp.prototype.global`
- `RegExp.prototype.ignoreCase`
- `RegExp.prototype.multiline`
- `RegExp.prototype.source`
- `RegExp.prototype.sticky`

- `RegExp.prototype.flags`

 Note

`RegExp.prototype.sticky` dan `RegExp.prototype.flags` baru di JavaScript runtime 2.0.

Metode prototype ES 5.1 berikut didukung:

- `RegExp.prototype.exec()`
- `RegExp.prototype.test()`
- `RegExp.prototype.toString()`
- `RegExp.prototype[@@replace]()`
- `RegExp.prototype[@@split]()`

 Note

`RegExp.prototype[@@split]()` baru di JavaScript runtime 2.0.

Properti instans ES 5.1 berikut didukung:

- `lastIndex`

Mendukung kelompok penangkap bernama ES 9.

JSON

Metode ES 5.1 berikut didukung:

- `JSON.parse()`
- `JSON.stringify()`

Array

Mendukung metode ES 5.1 berikut pada susunan:

- `Array.isArray()`

Mendukung metode ES 6 berikut pada susunan:

- `Array.of()`

Metode prototype ES 5.1 berikut didukung:

- `Array.prototype.concat()`
- `Array.prototype.every()`
- `Array.prototype.filter()`
- `Array.prototype.forEach()`
- `Array.prototype.indexOf()`
- `Array.prototype.join()`
- `Array.prototype.lastIndexOf()`
- `Array.prototype.map()`
- `Array.prototype.pop()`
- `Array.prototype.push()`
- `Array.prototype.reduce()`
- `Array.prototype.reduceRight()`
- `Array.prototype.reverse()`
- `Array.prototype.shift()`
- `Array.prototype.slice()`
- `Array.prototype.some()`
- `Array.prototype.sort()`
- `Array.prototype.splice()`
- `Array.prototype.unshift()`

Metode prototipe ES 6 berikut didukung

- `Array.prototype.copyWithin()`
- `Array.prototype.fill()`
- `Array.prototype.find()`
- `Array.prototype.findIndex()`

Metode prototipe ES 7 berikut didukung:

- `Array.prototype.includes()`

Susunan yang dijeniskan

Konstruktor array yang diketik ES 6 berikut didukung:

- `Float32Array`

- `Float64Array`
- `Int8Array`
- `Int16Array`
- `Int32Array`
- `Uint8Array`
- `Uint8ClampedArray`
- `Uint16Array`
- `Uint32Array`

Metode ES 6 berikut didukung:

- `TypedArray.from()`
- `TypedArray.of()`



Note

`TypedArray.from()` dan `TypedArray.of()` baru di JavaScript runtime 2.0.

Metode prototipe ES 6 berikut didukung:

- `TypedArray.prototype.copyWithin()`
- `TypedArray.prototype.every()`
- `TypedArray.prototype.fill()`
- `TypedArray.prototype.filter()`
- `TypedArray.prototype.find()`
- `TypedArray.prototype.findIndex()`
- `TypedArray.prototype.forEach()`
- `TypedArray.prototype.includes()`
- `TypedArray.prototype.indexOf()`
- `TypedArray.prototype.join()`
- `TypedArray.prototype.lastIndexOf()`
- `TypedArray.prototype.map()`
- `TypedArray.prototype.reduce()`
- `TypedArray.prototype.reduceRight()`

- `TypedArray.prototype.reverse()`
- `TypedArray.prototype.some()`
- `TypedArray.prototype.set()`
- `TypedArray.prototype.slice()`
- `TypedArray.prototype.sort()`
- `TypedArray.prototype.subarray()`
- `TypedArray.prototype.toString()`

 Note

`TypedArray.prototype.every()`, `TypedArray.prototype.fill()`, `TypedArray.prototype` dan `TypedArray.prototype.some()` baru di JavaScript runtime 2.0.

ArrayBuffer

Metode ES 6 berikut `ArrayBuffer` didukung:

- `isView()`

Metode prototipe ES 6 berikut `ArrayBuffer` didukung:

- `ArrayBuffer.prototype.slice()`

Janji

Metode ES 6 berikut pada janji didukung:

- `Promise.all()`
- `Promise.allSettled()`
- `Promise.any()`
- `Promise.reject()`
- `Promise.resolve()`
- `Promise.race()`

 Note

`Promise.all()`, `Promise.allSettled()`, `Promise.any()`, dan `Promise.race()` baru di JavaScript runtime 2.0.

Metode prototipe ES 6 berikut pada janji didukung:

- `Promise.prototype.catch()`
- `Promise.prototype.finally()`
- `Promise.prototype.then()`

DataView

Metode prototipe ES 6 berikut didukung:

- `DataView.prototype.getFloat32()`
- `DataView.prototype.getFloat64()`
- `DataView.prototype.getInt16()`
- `DataView.prototype.getInt32()`
- `DataView.prototype.getInt8()`
- `DataView.prototype.getUint16()`
- `DataView.prototype.getUint32()`
- `DataView.prototype.getUint8()`
- `DataView.prototype.setFloat32()`
- `DataView.prototype.setFloat64()`
- `DataView.prototype.setInt16()`
- `DataView.prototype.setInt32()`
- `DataView.prototype.setInt8()`
- `DataView.prototype.setUint16()`
- `DataView.prototype.setUint32()`
- `DataView.prototype.setUint8()`



Note

Semua metode prototipe DataView ES 6 baru di JavaScript runtime 2.0.

Simbol

Metode ES 6 berikut didukung:

- `Symbol.for()`

- `Symbol.keyfor()`

 Note

Semua metode Symbol ES 6 baru di JavaScript runtime 2.0.

Teks Decoder

Metode prototipe berikut didukung:

- `TextDecoder.prototype.decode()`

Properti pengakses prototipe berikut didukung:

- `TextDecoder.prototype.encoding`
- `TextDecoder.prototype.fatal`
- `TextDecoder.prototype.ignoreBOM`

Encoder Teks

Metode prototipe berikut didukung:

- `TextEncoder.prototype.encode()`
- `TextEncoder.prototype.encodeInto()`

Konsol

Ini adalah objek pembantu untuk debugging. Ini hanya mendukung metode `log()`, untuk merekam pesan log.

 Note

CloudFront Fungsi tidak mendukung sintaks koma, seperti `console.log('a', 'b')`. Sebagai gantinya, gunakan `console.log('a' + ' ' + 'b')` formatnya.

Jenis kesalahan

Mendukung objek kesalahan berikut:

- `Error`
- `EvalError`

- `InternalError`
- `RangeError`
- `ReferenceError`
- `SyntaxError`
- `TypeError`
- `URIError`

Global

Mendukung objek `globalThis`.

Mendukung fungsi global ES 5.1 berikut:

- `decodeURI()`
- `decodeURIComponent()`
- `encodeURI()`
- `encodeURIComponent()`
- `isFinite()`
- `isNaN()`
- `parseFloat()`
- `parseInt()`

Fungsi global ES 6 berikut didukung:

- `atob()`
- `btoa()`

Note

`atob()` dan `btoa()` baru di JavaScript runtime 2.0.

Mendukung konstanta global berikut:

- `NaN`

- Infinity
- undefined
- arguments

Modul bawaan

Mendukung modul bawaan berikut.

Modul

- [Penyangga](#)
- [String kueri](#)
- [Kripto](#)

Penyangga

Modul ini menyediakan metode berikut:

- `Buffer.alloc(size[, fill[, encoding]])`

Alokasikan a. Buffer

- `size`: Ukuran penyangga. Masukkan bilangan bulat.
- `fill`: Opsional. Masukkan string, Buffer, Uint8Array, atau integer. Default-nya adalah 0.
- `encoding`: Opsional. fillKapan string, masukkan salah satu dari berikut ini: utf8, hex, base64, base64url. Default-nya adalah utf8.
- `Buffer.allocUnsafe(size)`

Alokasikan Buffer non-inisialisasi.

- `size`: Masukkan bilangan bulat.
- `Buffer.byteLength(value[, encoding])`

Kembalikan panjang nilai, dalam byte.

- `value`: Sebuah string,,Buffer, DataView TypedArray, atau ArrayBuffer.
- `encoding`: Opsional. valueKapan string, masukkan salah satu dari berikut ini: utf8, hex, base64, base64url. Default-nya adalah utf8.
- `Buffer.compare(buffer1, buffer2)`

Bandingkan dua `Buffer` s untuk membantu mengurutkan array. Mengembalikan `0` jika mereka sama, `-1` jika `buffer1` datang lebih dulu, atau `1` jika `buffer2` datang lebih dulu.

- `buffer1`: Masukkan `aBuffer`.
- `buffer2`: Masukkan yang berbeda `Buffer`.
- `Buffer.concat(list[, totalLength])`

Gabungkan beberapa s. `Buffer` Kembali `0` jika tidak ada. Kembali hingga `totalLength`.

- `list`: Masukkan daftar `Buffer` s. Perhatikan ini akan dipotong menjadi. `totalLength`
- `totalLength`: Opsional. Masukkan bilangan bulat yang tidak ditandatangani. Gunakan jumlah `Buffer` instance dalam daftar jika kosong.
- `Buffer.from(array)`

Buat `Buffer` dari array.

- `array`: Masukkan array byte dari `0` ke `255`.
- `Buffer.from(arrayBuffer, byteOffset[, length])`

Buat tampilan dari `arrayBuffer`, mulai dari offset `byteOffset` dengan panjang `length`.

- `arrayBuffer`: Masukkan `Buffer` array.
- `byteOffset`: Masukkan bilangan bulat.
- `length`: Opsional. Masukkan bilangan bulat.
- `Buffer.from(buffer)`

Buat salinan dari `Buffer`.

- `buffer`: Masukkan `aBuffer`.
- `Buffer.from(object[, offsetOrEncoding[, length]])`

Buat `Buffer` dari objek. Mengembalikan `Buffer.from(object.valueOf(), offsetOrEncoding, length)` jika `valueOf()` tidak sama dengan objek.

- `object`: Masukkan objek.
- `offsetOrEncoding`: Opsional. Masukkan integer atau string encoding.
- `length`: Opsional. Masukkan bilangan bulat.
- `Buffer.from(string[, encoding])`

Buat `Buffer` dari string.

- `string`: Masukkan string.
- `encoding`: Opsional. Masukkan salah satu dari berikut ini: `utf8`, `hex`, `base64`, `base64url`. Default-nya adalah `utf8`.
- `Buffer.isBuffer(object)`

Periksa apakah `object` itu `Buffer`. Pengembalian `true` atau `false`.

- `object`: Masukkan objek.
- `Buffer.isEncoding(encoding)`

Periksa `encoding` apakah didukung. Pengembalian `true` atau `false`.

- `encoding`: Opsional. Masukkan salah satu dari berikut ini: `utf8`, `hex`, `base64`, `base64url`. Default-nya adalah `utf8`.

Modul ini menyediakan metode prototipe `buffer` berikut:

- `Buffer.prototype.compare(target[, targetStart[, targetEnd[, sourceStart[, sourceEnd]]]])`

Bandingkan `Buffer` dengan `target`. Mengembalikan 0 jika mereka sama, 1 jika `buffer` datang lebih dulu, atau -1 jika `target` datang lebih dulu.

- `target`: Masukkan `aBuffer`.
- `targetStart`: Opsional. Masukkan bilangan bulat. Default-nya adalah 0.
- `targetEnd`: Opsional. Masukkan bilangan bulat. Default adalah `target` panjang.
- `sourceStart`: Opsional. Masukkan bilangan bulat. Default-nya adalah 0.
- `sourceEnd`: Opsional. Masukkan bilangan bulat. Default adalah `Buffer` panjang.
- `Buffer.prototype.copy(target[, targetStart[, sourceStart[, sourceEnd]]])`

Salin `buffer` ke `target`.

- `target`: Masukkan `a Buffer` atau `Uint8Array`.
- `targetStart`: Opsional. Masukkan bilangan bulat. Default-nya adalah 0.
- `sourceStart`: Opsional. Masukkan bilangan bulat. Default-nya adalah 0.
- `sourceEnd`: Opsional. Masukkan bilangan bulat. Default adalah `Buffer` panjang.
- `Buffer.prototype.equals(otherBuffer)`

Bandingkan `Buffer` dengan `anotherBuffer`. Pengembalian `true` atau `false`.

- `otherBuffer`: Masukkan string.
- `Buffer.prototype.fill(value[, offset[, end][, encoding]])`

Isi Buffer dengan `value`.

- `value`: Masukkan string, Buffer, atau bilangan bulat.
- `offset`: Opsional. Masukkan bilangan bulat.
- `end`: Opsional. Masukkan bilangan bulat.
- `encoding`: Opsional. Masukkan salah satu dari berikut ini: `utf8`, `hex`, `base64`, `base64url`. Default-nya adalah `utf8`.
- `Buffer.prototype.includes(value[, byteOffset][, encoding])`

Cari `value` di Buffer. Pengembalian `true` atau `false`.

- `value`: Masukkan string, `Buffer`, `Uint8Array`, atau bilangan bulat.
- `byteOffset`: Opsional. Masukkan bilangan bulat.
- `encoding`: Opsional. Masukkan salah satu dari berikut ini: `utf8`, `hex`, `base64`, `base64url`. Default-nya adalah `utf8`.
- `Buffer.prototype.indexOf(value[, byteOffset][, encoding])`

Cari yang pertama `value` di Buffer. Mengembalikan `index` jika ditemukan; kembali `-1` jika tidak ditemukan.

- `value`: Masukkan string, Buffer, `Unit8Array`, atau bilangan bulat dari 0 hingga 255.
- `byteOffset`: Opsional. Masukkan bilangan bulat.
- `encoding`: Opsional. Masukkan salah satu dari berikut jika `value` adalah string: `utf8`, `hex`, `base64`, `base64url`. Default-nya adalah `utf8`.
- `Buffer.prototype.lastIndexOf(value[, byteOffset][, encoding])`

Cari yang terakhir `value` di Buffer. Mengembalikan `index` jika ditemukan; kembali `-1` jika tidak ditemukan.

- `value`: Masukkan string, Buffer, `Unit8Array`, atau bilangan bulat dari 0 hingga 255.
- `byteOffset`: Opsional. Masukkan bilangan bulat.
- `encoding`: Opsional. Masukkan salah satu dari berikut jika `value` adalah string: `utf8`, `hex`, `base64`, `base64url`. Default-nya adalah `utf8`.

- `Buffer.prototype.readInt8(offset)`

Tulis kode fungsi

Baca Int8 di offset dari Buffer.

- offset: Masukkan bilangan bulat.
- `Buffer.prototype.readIntBE(offset, byteLength)`

Baca Int sebagai big-endian di from. offset Buffer

- offset: Masukkan bilangan bulat.
- byteLength: Opsional. Masukkan bilangan bulat dari 1 ke6.
- `Buffer.prototype.readInt16BE(offset)`

Baca Int16 sebagai big-endian di from. offset Buffer

- offset: Masukkan bilangan bulat.
- `Buffer.prototype.readInt32BE(offset)`

Baca Int32 sebagai big-endian di from. offset Buffer

- offset: Masukkan bilangan bulat.
- `Buffer.prototype.readIntLE(offset, byteLength)`

Baca Int sebagai endian kecil di dari. offset Buffer

- offset: Masukkan bilangan bulat.
- byteLength: Masukkan bilangan bulat dari 1 ke6.
- `Buffer.prototype.readInt16LE(offset)`

Baca Int16 sebagai endian kecil di dari. offset Buffer

- offset: Masukkan bilangan bulat.
- `Buffer.prototype.readInt32LE(offset)`

Baca Int32 sebagai endian kecil di dari. offset Buffer

- offset: Masukkan bilangan bulat.
- `Buffer.prototype.readUInt8(offset)`

Baca UInt8 di offset dari Buffer.

- offset: Masukkan bilangan bulat.
- `Buffer.prototype.readUIntBE(offset, byteLength)`

Baca UInt sebagai big-endian di from. offset Buffer

- `offset`: Masukkan bilangan bulat.
- `byteLength`: Masukkan bilangan bulat dari 1 ke6.
- `Buffer.prototype.readUInt16BE(offset)`

Baca `UInt16` sebagai big-endian di from. `offset` `Buffer`

- `offset`: Masukkan bilangan bulat.
- `Buffer.prototype.readUInt32BE(offset)`

Baca `UInt32` sebagai big-endian di from. `offset` `Buffer`

- `offset`: Masukkan bilangan bulat.
- `Buffer.prototype.readUIntLE(offset, byteLength)`

Baca `UInt` sebagai endian kecil di dari. `offset` `Buffer`

- `offset`: Masukkan bilangan bulat.
- `byteLength`: Masukkan bilangan bulat dari 1 ke6.
- `Buffer.prototype.readUInt16LE(offset)`

Baca `UInt16` sebagai endian kecil di dari. `offset` `Buffer`

- `offset`: Masukkan bilangan bulat.
- `Buffer.prototype.readUInt32LE(offset)`

Baca `UInt32` sebagai endian kecil di dari. `offset` `Buffer`

- `offset`: Masukkan bilangan bulat.
- `Buffer.prototype.readDoubleBE([offset])`

Baca 64-bit ganda sebagai big-endian di from. `offset` `Buffer`

- `offset`: Opsional. Masukkan bilangan bulat.
- `Buffer.prototype.readDoubleLE([offset])`

Baca 64-bit ganda sebagai endian kecil di from. `offset` `Buffer`

- `offset`: Opsional. Masukkan bilangan bulat.
- `Buffer.prototype.readFloatBE([offset])`

Baca float 32-bit sebagai big-endian di from. `offset` `Buffer`

- `offset`: Opsional. Masukkan bilangan bulat.

- `Buffer.prototype.readFloatLE([offset])`

Baca float 32-bit sebagai endian kecil di from. offset Buffer

- `offset`: Opsional. Masukkan bilangan bulat.

- `Buffer.prototype.subarray([start[, end]])`

Mengembalikan salinan Buffer yang diimbangi dan dipotong dengan yang baru `start` dan `end`

- `start`: Opsional. Masukkan bilangan bulat. Default-nya adalah 0.
- `end`: Opsional. Masukkan bilangan bulat. Default adalah panjang buffer.

- `Buffer.prototype.swap16()`

Tukar urutan byte Buffer array, memperlakukannya sebagai array angka 16-bit. Bufferpanjangnya harus habis dibagi 2, atau Anda akan menerima kesalahan.

- `Buffer.prototype.swap32()`

Tukar urutan byte Buffer array, memperlakukannya sebagai array angka 32-bit. Bufferpanjangnya harus habis dibagi 4, atau Anda akan menerima kesalahan.

- `Buffer.prototype.swap64()`

Tukar urutan byte Buffer array, memperlakukannya sebagai array angka 64-bit. Bufferpanjangnya harus habis dibagi 8, atau Anda akan menerima kesalahan.

- `Buffer.prototype.toJSON()`

Kembali Buffer sebagai JSON.

- `Buffer.prototype.toString([encoding[, start[, end]])`

Konversi Buffer, dari `start` ke `end`, ke string yang dikodekan.

- `encoding`: Opsional. Masukkan salah satu dari berikut ini: `utf8`, `hex`, `base64`, atau `base64url`. Default-nya adalah `utf8`.
- `start`: Opsional. Masukkan bilangan bulat. Default-nya adalah 0.
- `end`: Opsional. Masukkan bilangan bulat. Default adalah panjang buffer.

- `Buffer.prototype.write(string[, offset[, length]][, encoding])`

Tulis dikodekan `string` ke Buffer jika ada spasi, atau terpotong `string` jika tidak ada cukup ruang.

- `string`: Masukkan string.

- `offset`: Opsional. Masukkan bilangan bulat. Default-nya adalah 0.
- `length`: Opsional. Masukkan bilangan bulat. Default adalah panjang string.
- `encoding`: Opsional. Secara opsional masukkan salah satu dari berikut ini: `utf8`, `hex`, `base64`, atau `base64url`. Default-nya adalah `utf8`.
- `Buffer.prototype.writeInt8(value, offset, byteLength)`

Menulis `Int8` value dari `byteLength` at `offset` ke `Buffer`.

- `value`: Masukkan bilangan bulat.
- `offset`: Masukkan bilangan bulat
- `byteLength`: Masukkan bilangan bulat dari 1 ke 6.
- `Buffer.prototype.writeIntBE(value, offset, byteLength)`

Menulis `value` di `offset` ke `Buffer`, menggunakan big-endian.

- `value`: Masukkan bilangan bulat.
- `offset`: Masukkan bilangan bulat
- `byteLength`: Masukkan bilangan bulat dari 1 ke 6.
- `Buffer.prototype.writeInt16BE(value, offset, byteLength)`

Menulis `value` di `offset` ke `Buffer`, menggunakan big-endian.

- `value`: Masukkan bilangan bulat.
- `offset`: Masukkan bilangan bulat
- `byteLength`: Masukkan bilangan bulat dari 1 ke 6.
- `Buffer.prototype.writeInt32BE(value, offset, byteLength)`

Menulis `value` di `offset` ke `Buffer`, menggunakan big-endian.

- `value`: Masukkan bilangan bulat.
- `offset`: Masukkan bilangan bulat
- `byteLength`: Masukkan bilangan bulat dari 1 ke 6.
- `Buffer.prototype.writeIntLE(offset, byteLength)`

Menulis `value` di `offset` ke `Buffer`, menggunakan little-endian.

- `offset`: Masukkan bilangan bulat.

- `byteLength`: Masukkan bilangan bulat dari 1 ke 6.

- `Buffer.prototype.writeInt16LE(offset, byteLength)`

Menulis value di offset keBuffer, menggunakan little-endian.

- `offset`: Masukkan bilangan bulat.
 - `byteLength`: Masukkan bilangan bulat dari 1 ke6.
- `Buffer.prototype.writeInt32LE(offset, byteLength)`

Menulis value di offset keBuffer, menggunakan little-endian.

- `offset`: Masukkan bilangan bulat.
 - `byteLength`: Masukkan bilangan bulat dari 1 ke6.
- `Buffer.prototype.writeUInt8(value, offset, byteLength)`

Menulis UInt8 value dari byteLength at offset keBuffer.

- `value`: Masukkan bilangan bulat.
 - `offset`: Masukkan bilangan bulat
 - `byteLength`: Masukkan bilangan bulat dari 1 ke6.
- `Buffer.prototype.writeUIntBE(value, offset, byteLength)`

Menulis value di offset keBuffer, menggunakan big-endian.

- `value`: Masukkan bilangan bulat.
 - `offset`: Masukkan bilangan bulat
 - `byteLength`: Masukkan bilangan bulat dari 1 ke6.
- `Buffer.prototype.writeUInt16BE(value, offset, byteLength)`

Menulis value di offset keBuffer, menggunakan big-endian.

- `value`: Masukkan bilangan bulat.
 - `offset`: Masukkan bilangan bulat
 - `byteLength`: Masukkan bilangan bulat dari 1 ke6.
- `Buffer.prototype.writeUInt32BE(value, offset, byteLength)`

Menulis value di offset keBuffer, menggunakan big-endian.

- `value`: Masukkan bilangan bulat.
- `offset`: Masukkan bilangan bulat
- `byteLength`: Masukkan bilangan bulat dari 1 ke6.

- `Buffer.prototype.writeUIntLE(value, offset, byteLength)`

Menulis `value` di `offset` ke `Buffer`, menggunakan little-endian.

- `value`: Masukkan bilangan bulat.
 - `offset`: Masukkan bilangan bulat
 - `byteLength`: Masukkan bilangan bulat dari 1 ke 6.
- `Buffer.prototype.writeUInt16LE(value, offset, byteLength)`

Menulis `value` di `offset` ke `Buffer`, menggunakan little-endian.

- `value`: Masukkan bilangan bulat.
 - `offset`: Masukkan bilangan bulat
 - `byteLength`: Masukkan bilangan bulat dari 1 ke 6.
- `Buffer.prototype.writeUInt32LE(value, offset, byteLength)`

Menulis `value` di `offset` ke `Buffer`, menggunakan little-endian.

- `value`: Masukkan bilangan bulat.
 - `offset`: Masukkan bilangan bulat
 - `byteLength`: Masukkan bilangan bulat dari 1 ke 6.
- `Buffer.prototype.writeDoubleBE(value, [offset])`

Menulis `value` di `offset` ke `Buffer`, menggunakan big-endian.

- `value`: Masukkan bilangan bulat.
 - `offset`: Opsional. Masukkan bilangan bulat. Default-nya adalah 0.
- `Buffer.prototype.writeDoubleLE(value, [offset])`

Menulis `value` di `offset` ke `Buffer`, menggunakan little-endian.

- `value`: Masukkan bilangan bulat.
 - `offset`: Opsional. Masukkan bilangan bulat. Default-nya adalah 0.
- `Buffer.prototype.writeFloatBE(value, [offset])`

Menulis `value` di `offset` ke `Buffer`, menggunakan big-endian.

- `value`: Masukkan bilangan bulat.
- `offset`: Opsional. Masukkan bilangan bulat. Default-nya adalah 0.

- `Buffer.prototype.writeFloatLE(value, [offset])`

Menulis `value` di `offset` ke `buffer`, menggunakan little-endian.

- `value`: Masukkan bilangan bulat.
- `offset`: Opsional. Masukkan bilangan bulat. Default-nya adalah 0.

Metode contoh berikut didukung:

- `buffer[index]`

Dapatkan dan atur oktet (byte) di `index` dalam `Buffer`

- Dapatkan nomor dari 0 ke 255. Atau tetapkan angka dari 0 ke 255.

Properti contoh berikut didukung:

- `buffer`

Dapatkan `ArrayBuffer` objek untuk `buffer`.

- `byteOffset`

Dapatkan `Arraybuffer` objek `buffer.byteOffset`

- `length`

Dapatkan jumlah byte `buffer`.

Note

Semua metode modul `Buffer` baru di JavaScript runtime 2.0.

String kueri

Note

[Objek acara CloudFront Functions](#) secara otomatis mem-parsing string kueri URL untuk Anda. Itu berarti bahwa dalam kebanyakan kasus Anda tidak perlu menggunakan modul ini.

Modul `string` kueri (`querystring`) menyediakan metode untuk mengurai dan memformat string kueri URL. Anda dapat memuat modul menggunakan `require('querystring')`. Modul ini menyediakan metode berikut.

`querystring.escape(string)`

Mengkodekan URL `string` yang tersedia, mengembalikan string kueri yang lolos. Metode ini digunakan oleh `querystring.stringify()` dan tidak boleh digunakan secara langsung.

`querystring.parse(string[, separator[, equal[, options]])`

Mengurai string kueri (`string`) dan mengembalikan objek.

Parameter `separator` adalah substring untuk membatasi pasangan kunci dan nilai dalam string kueri. Secara default, itu adalah `&`.

Parameter `equal` adalah substring untuk membatasi kunci dan nilai dalam string kueri. Secara default, itu adalah `=`.

Parameter `options` adalah objek dengan kunci berikut:

`decodeURIComponent` *function*

Sebuah fungsi untuk memecahkan kode karakter persentase dikodekan dalam string kueri. Secara default, itu adalah `querystring.unescape()`.

`maxKeys` *number*

Jumlah kunci maksimum untuk diurai. Secara default, itu adalah `1000`. Gunakan nilai `0` untuk menghapus pembatasan untuk menghitung kunci.

Secara default, karakter persentase dikodekan dalam string kueri diasumsikan menggunakan pengkodean UTF-8. Urutan UTF-8 tidak valid diganti dengan karakter pengganti `U+FFFD`.

Misalnya, untuk string kueri berikut:

```
'name=value&abc=xyz&abc=123'
```

Nilai `querystring.parse()` yang dikembalikan adalah:

```
{
  name: 'value',
  abc: ['xyz', '123']
}
```

`querystring.decode()` adalah alias untuk `querystring.parse()`.

`querystring.stringify(object[, separator[, equal[, options]])`

Menyerialisasi `object` dan mengembalikan string kueri.

Parameter `separator` adalah substring untuk membatasi pasangan kunci dan nilai dalam string kueri. Secara default, itu adalah `&`.

Parameter `equal` adalah substring untuk membatasi kunci dan nilai dalam string kueri. Secara default, itu adalah `=`.

Parameter `options` adalah objek dengan kunci berikut:

`encodeURIComponent` *function*

Fungsi yang digunakan untuk mengonversi karakter URL yang tidak aman untuk pengkodean persentase dalam string kueri. Secara default, itu adalah `querystring.escape()`.

Secara default, karakter yang memerlukan pengkodean persentase dalam string kueri dikodekan sebagai UTF-8. Untuk menggunakan pengkodean yang berbeda, tentukan opsi `encodeURIComponent`.

Misalnya, gunakan kode berikut:

```
querystring.stringify({ name: 'value', abc: ['xyz', '123'], anotherName: '' });
```

Nilai yang dikembalikan adalah:

```
'name=value&abc=xyz&abc=123&anotherName='
```

`querystring.encode()` adalah alias untuk `querystring.stringify()`.

`querystring.unescape(string)`

Mendekode karakter yang dikodekan persentase URL dalam string yang tersedia, mengembalikan string kueri yang tidak lolos. Metode ini digunakan oleh `querystring.parse()` dan tidak boleh digunakan secara langsung.

Kripto

Modul kriptografi (`crypto`) menyediakan hashing standar dan pembantu kode autentikasi pesan berbasis hash (HMAC). Anda dapat memuat modul menggunakan `require('crypto')`.

Metode hashing

`crypto.createHash(algorithm)`

Menciptakan dan mengembalikan objek hash yang dapat digunakan untuk menghasilkan hash digests menggunakan algoritme yang diberikan: md5, sha1, atau sha256.

`hash.update(data)`

Update konten hash dengan data yang tersedia.

`hash.digest([encoding])`

Menghitung digest dari semua data yang diteruskan menggunakan `hash.update()`. Pengkodean dapat berupa hex, base64, atau base64url.

Metode HMAC

`crypto.createHmac(algorithm, secret key)`

Menciptakan dan mengembalikan objek HMAC yang menggunakan `algorithm` dan `secret key` yang tersedia. Algoritma dapat berupa md5, sha1, atau sha256.

`hmac.update(data)`

Memperbarui konten HMAC dengan data yang tersedia.

`hmac.digest([encoding])`

Menghitung digest dari semua data yang diteruskan menggunakan `hmac.update()`. Pengkodean dapat berupa hex, base64, atau base64url.

Fitur yang dibatasi

Fitur JavaScript bahasa berikut tidak didukung atau dibatasi karena masalah keamanan.

Evaluasi kode dinamis

Evaluasi kode dinamis tidak didukung. Konstruktor `eval()` dan `Function` mengalami kesalahan jika dicoba. Misalnya, `const sum = new Function('a', 'b', 'return a + b')` mengalami kesalahan.

Timer

Fungsi `setTimeout()`, `setImmediate()`, dan `clearTimeout()` tidak didukung. Tidak ada ketentuan untuk menunda atau menghasilkan dalam fungsi jalankan. Fungsi Anda harus dijalankan secara serentak sampai selesai.

Tanggal dan stempel waktu

Untuk alasan keamanan, tidak ada akses ke timer beresolusi tinggi. Semua metode `Date` untuk meminta waktu saat ini selalu mengembalikan nilai yang sama saat menjalankan fungsi tunggal. Stempel waktu yang dikembalikan adalah waktu fungsi mulai berjalan. Akibatnya, Anda tidak dapat mengukur waktu berlalu dalam fungsi Anda.

Akses sistem file

Tidak ada akses sistem file. Misalnya, tidak ada modul `fs` untuk akses sistem file seperti yang ada di `Node.js`.

Akses proses

Tidak ada akses proses. Misalnya, tidak ada objek `process` global untuk memproses akses informasi seperti yang ada di `Node.js`.

Variabel-variabel lingkungan

Tidak ada akses ke variabel lingkungan. Sebagai gantinya, Anda dapat menggunakan CloudFront Key/ValueStore untuk membuat datastore terpusat dari pasangan kunci-nilai untuk Fungsi Anda. CloudFront Key/ValueStore memungkinkan pembaruan dinamis ke data konfigurasi Anda tanpa perlu menerapkan perubahan kode. Untuk informasi selengkapnya, lihat [Amazon CloudFront Key/ValueStore](#).

Akses jaringan

Tidak ada dukungan untuk panggilan jaringan. Misalnya, XHR, HTTP (S), dan soket tidak didukung.

Metode pembantu untuk penyimpanan nilai kunci

Note

Panggilan metode pembantu penyimpanan nilai kunci dari CloudFront Fungsi tidak memicu peristiwa AWS CloudTrail data. Peristiwa ini tidak dicatat dalam riwayat CloudTrail peristiwa.

Untuk informasi selengkapnya, lihat [Pencatatan panggilan CloudFront API Amazon menggunakan AWS CloudTrail](#).

Bagian ini berlaku jika Anda menggunakan [CloudFront Key Value Store](#) untuk menyertakan nilai kunci dalam fungsi yang Anda buat. CloudFront Fungsi memiliki modul yang menyediakan tiga metode pembantu untuk membaca nilai dari penyimpanan nilai kunci.

Untuk menggunakan modul ini dalam kode fungsi, pastikan bahwa Anda telah [mengaitkan penyimpanan nilai kunci](#) dengan fungsi tersebut.

Selanjutnya, sertakan pernyataan berikut di baris pertama kode fungsi:

```
import cf from 'cloudfront';  
const kvsHandle = cf.kvs();
```

get () metode

Gunakan metode ini untuk mengembalikan nilai kunci untuk nama kunci yang Anda tentukan.

Permintaan

```
get("key", options);
```

- **key**: Nama kunci yang nilainya perlu diambil
- **options**: Ada satu pilihan, format. Ini memastikan bahwa fungsi mem-parsing data dengan benar. Kemungkinan nilai:
 - **string**: (Default) UTF8 dikodekan
 - **json**
 - **bytes**: Buffer data biner mentah

Minta contoh

```
const value = await kvsHandle.get("myFunctionKey", { format: "string"});
```

Respons

Responsnya adalah `promise` yang menyelesaikan nilai dalam format yang diminta dengan menggunakan `options`. Secara default, nilai dikembalikan sebagai string.

Penanganan kesalahan

`get()` Metode ini akan mengembalikan kesalahan ketika kunci yang Anda minta tidak ada di penyimpanan nilai kunci terkait. Untuk mengelola kasus penggunaan ini, Anda dapat menambahkan `try` dan `catch` memblokir kode Anda.

Warning

Menggunakan kombinator janji (misalnya, `Promise.all`, `Promise.any`, dan metode rantai janji (misalnya, `then` dan `catch`) dapat memerlukan penggunaan memori fungsi tinggi. Jika fungsi Anda melebihi kuota [memori fungsi maksimum](#), itu akan gagal dijalankan. Untuk menghindari kesalahan ini, kami sarankan Anda menggunakan `await` sintaks secara berurutan atau dalam loop untuk meminta beberapa nilai.

Contoh

```
var value1 = await kvs.get('key1');  
var value2 = await kvs.get('key2');
```

Saat ini, menggunakan kombinator janji untuk mendapatkan beberapa nilai tidak akan meningkatkan kinerja, seperti contoh berikut.

```
var values = await Promise.all([kvs.get('key1'), kvs.get('key2'),]);
```

ada() metode

Gunakan metode ini untuk mengidentifikasi apakah kunci ada di penyimpanan nilai kunci atau tidak.

Permintaan

```
exists("key");
```

Minta contoh

```
const exist = await kvsHandle.exists("myFunctionkey");
```

Respons

Responsnya adalah `promise` yang mengembalikan Boolean (`true` atau `false`). Nilai ini menentukan apakah kunci ada atau tidak di penyimpanan nilai kunci.

`meta ()` metode

Gunakan metode ini untuk mengembalikan metadata tentang penyimpanan nilai kunci.

Permintaan

```
meta();
```

Minta contoh

```
const meta = await kvsHandle.meta();
```

Respons

Responsnya adalah `promise` yang menyelesaikan objek dengan properti berikut:

- `creationDateTime`: Tanggal dan waktu penyimpanan nilai kunci dibuat, dalam format ISO 8601.
- `lastUpdatedDateTime`: Tanggal dan waktu penyimpanan nilai kunci terakhir disinkronkan dari sumber, dalam format ISO 8601. Nilainya tidak termasuk waktu propagasi ke tepi.
- `keyCount`: Jumlah total kunci di KVS setelah sinkronisasi terakhir dari sumber.

Contoh respons

```
{keyCount:3,creationDateTime:2023-11-30T23:07:55.765Z,lastUpdatedDateTime:2023-12-15T03:57:52.4
```

Metode pembantu untuk modifikasi asal

Bagian ini berlaku jika Anda secara dinamis memperbarui atau mengubah asal yang digunakan pada permintaan di dalam kode CloudFront Fungsi Anda. Anda dapat memperbarui asal hanya pada CloudFront fungsi permintaan penampil. CloudFront Fungsi memiliki modul yang menyediakan metode pembantu untuk memperbarui atau mengubah asal secara dinamis.

Untuk menggunakan modul ini, buat CloudFront fungsi menggunakan JavaScript runtime 2.0 dan sertakan pernyataan berikut di baris pertama kode fungsi:

```
import cf from 'cloudfront';
```

Untuk informasi selengkapnya, lihat [JavaScript fitur runtime 2.0 untuk Fungsi CloudFront](#).

 Note

Halaman Test API dan Test console tidak menguji apakah modifikasi asal telah terjadi. Namun, pengujian memastikan bahwa kode fungsi dijalankan tanpa kesalahan.

Pilih antara CloudFront Fungsi dan Lambda @Edge

Anda dapat memperbarui asal Anda dengan menggunakan CloudFront Fungsi atau Lambda @Edge.

Saat menggunakan CloudFront Fungsi untuk memperbarui asal, Anda menggunakan pemicu peristiwa permintaan penampil, yang berarti logika ini akan berjalan pada setiap permintaan saat fungsi ini digunakan. Saat menggunakan Lambda @Edge, kemampuan pembaruan asal ada di pemicu peristiwa permintaan asal, yang berarti logika ini hanya berjalan pada kesalahan cache.

Pilihan Anda sangat bergantung pada beban kerja Anda dan penggunaan CloudFront Fungsi dan Lambda @Edge yang ada pada distribusi Anda. Pertimbangan berikut dapat membantu Anda memutuskan apakah akan menggunakan CloudFront Fungsi atau Lambda @Edge untuk memperbarui asal Anda.

CloudFront Fungsi paling berguna dalam situasi berikut:

- Ketika permintaan Anda dinamis (artinya mereka tidak dapat di-cache) dan akan selalu pergi ke asal. CloudFront Fungsi memberikan kinerja yang lebih baik dan biaya keseluruhan yang lebih rendah.
- Ketika Anda sudah memiliki CloudFront fungsi permintaan penampil yang ada yang akan berjalan pada setiap permintaan, Anda dapat menambahkan logika pembaruan asal ke dalam fungsi yang ada.

Untuk menggunakan CloudFront Fungsi untuk memperbarui asal, lihat metode pembantu dalam topik berikut.

Lambda @Edge paling berguna dalam situasi berikut:

- Ketika Anda memiliki konten yang sangat dapat di-cache, Lambda @Edge bisa lebih hemat biaya karena hanya berjalan pada cache yang gagal, sementara Fungsi berjalan pada setiap permintaan. CloudFront
- Ketika Anda sudah memiliki fungsi Lambda @Edge permintaan asal yang ada, Anda dapat menambahkan logika pembaruan asal ke fungsi yang ada.
- Bila logika pembaruan asal Anda memerlukan pengambilan data dari sumber data pihak ketiga, seperti Amazon DynamoDB atau Amazon S3.

Untuk informasi selengkapnya tentang Lambda @Edge, lihat. [Sesuaikan di tepi dengan Lambda @Edge](#)

metode `updateRequestOrigin()`

Gunakan `updateRequestOrigin()` metode ini untuk memperbarui pengaturan asal untuk permintaan. Anda dapat menggunakan metode ini untuk memperbarui properti asal yang ada untuk asal yang sudah ditentukan dalam distribusi Anda, atau untuk menentukan asal baru untuk permintaan tersebut. Untuk melakukannya, tentukan properti yang ingin Anda ubah.

 Important

Pengaturan apa pun yang tidak Anda tentukan `updateRequestOrigin()` akan mewarisi pengaturan yang sama dari konfigurasi asal yang ada.

Asal yang ditetapkan oleh `updateRequestOrigin()` metode dapat berupa titik akhir HTTP apa pun dan tidak perlu menjadi asal yang ada dalam CloudFront distribusi Anda.

 Catatan

- Jika Anda memperbarui asal yang merupakan bagian dari grup asal, hanya asal utama grup asal yang diperbarui. Asal sekunder tetap tidak berubah. Kode respons apa pun dari asal yang dimodifikasi yang cocok dengan kriteria failover akan memicu failover ke asal sekunder.
- Jika Anda mengubah tipe asal dan mengaktifkan OAC, pastikan tipe asal `originAccessControlConfig` cocok dengan tipe asal baru.

- Anda tidak dapat menggunakan `updateRequestOrigin()` metode ini untuk memperbarui asal [VPC](#). Permintaan akan gagal.

Permintaan

```
updateRequestOrigin({origin properties})
```

`origin properties` Dapat berisi yang berikut:

`domainName` (opsional)

Nama domain asal. Jika ini tidak disediakan, nama domain dari asal yang ditetapkan digunakan sebagai gantinya.

Untuk asal-usul khusus

Tentukan nama domain DNS, seperti `www.example.com`. Nama domain tidak dapat menyertakan titik dua (:) dan tidak bisa menjadi alamat IP. Nama domain dapat terdiri dari hingga 253 karakter.

Untuk asal-usul S3

Tentukan nama domain DNS bucket Amazon S3, seperti `amzn-s3-demo-bucket.s3.eu-west-1.amazonaws.com`. Nama bisa sampai 128 karakter dan harus berupa huruf kecil.

`Hostheader` (opsional, untuk asal-usul kustom non-S3)

Header host untuk digunakan saat membuat permintaan ke asal. Jika ini tidak disediakan, nilai dari parameter `domainName` digunakan. Jika header host atau parameter nama domain tidak disediakan, nama domain dari asal yang ditetapkan akan digunakan atau header host dari permintaan masuk jika kebijakan penerusan ke asal (FTO) menyertakan host. Header host tidak dapat menyertakan titik dua (:) dan tidak bisa menjadi alamat IP. Header host dapat mencapai 253 karakter.

`OriginPath` (opsional)

Jalur direktori di tempat asal permintaan harus menemukan konten. Jalur harus dimulai dengan garis miring (/) tetapi tidak boleh diakhiri dengan garis miring. Misalnya, seharusnya tidak berakhir dengan `example-path/`. Jika ini tidak disediakan, maka jalur asal dari asal yang ditetapkan digunakan.

Untuk asal-usul khusus

Jalur harus dikodekan URL dan memiliki panjang maksimum 255 karakter.

CustomHeaders (opsional)

Anda dapat menyertakan judul kustom dengan permintaan dengan menyebutkan nama header dan pasangan nilai untuk masing-masing header khusus. Formatnya berbeda dari header permintaan dan respons dalam struktur acara. Gunakan sintaks pasangan kunci-nilai berikut:

```
{"key1": "value1", "key2": "value2", ...}
```

Anda tidak dapat menambahkan header yang tidak diizinkan, dan header dengan nama yang sama juga tidak dapat hadir dalam permintaan masuk. headers Nama header harus huruf kecil dalam kode fungsi Anda. Ketika CloudFront Fungsi mengubah objek acara kembali ke permintaan HTTP, huruf pertama dari setiap kata dalam nama header dikapitalisasi, dan kata-kata dipisahkan oleh tanda hubung.

Misalnya, jika kode fungsi Anda menambahkan header bernama `example-header-name`, CloudFront mengkonversi ini ke `Example-Header-Name` dalam permintaan HTTP. Untuk informasi selengkapnya, lihat [Header khusus yang tidak CloudFront dapat ditambahkan ke permintaan asal](#) dan [Pembatasan pada fungsi edge](#).

Jika ini tidak disediakan, maka header khusus apa pun dari asal yang ditetapkan akan digunakan.

ConnectionAtsts (opsional)

Berapa kali yang CloudFront mencoba untuk terhubung ke asal. Minimal adalah 1 dan maksimum adalah 3. Jika ini tidak disediakan, upaya koneksi dari asal yang ditetapkan digunakan.

OriginShield (opsional)

Ini memungkinkan atau memperbarui CloudFront Origin Shield. Menggunakan Shield Asal dapat membantu mengurangi beban pada asal Anda. Untuk informasi selengkapnya, lihat [Gunakan Amazon CloudFront Origin Shield](#). Jika ini tidak disediakan, pengaturan Origin Shield dari asal yang ditetapkan akan digunakan.

diaktifkan (wajib)

Ekspresi Boolean untuk mengaktifkan atau menonaktifkan Origin Shield. Menerima `false` nilai `true` atau.

wilayah (diperlukan saat diaktifkan)

Wilayah AWS Untuk Origin Shield Tentukan Wilayah AWS yang memiliki latensi terendah pada asal Anda. Gunakan kode Region, bukan nama Region. Misalnya, gunakan `us-east-2` untuk menentukan Wilayah Timur AS (Ohio).

Ketika Anda mengaktifkan CloudFront Origin Shield, Anda harus menentukan Wilayah AWS untuk itu. Untuk daftar yang tersedia Wilayah AWS dan membantu memilih Wilayah terbaik untuk asal Anda, lihat [Pilih AWS Wilayah untuk Origin Shield](#).

asal AccessControlConfig (opsional)

Pengidentifikasi unik dari kontrol akses asal (OAC) untuk asal ini. Ini hanya digunakan ketika asal mendukung CloudFront OAC, seperti Amazon S3, URL MediaStore fungsi Lambda, dan V2. MediaPackage Jika ini tidak disediakan, maka pengaturan OAC dari asal yang ditetapkan digunakan.

Ini tidak mendukung identitas akses asal warisan (OAI). Untuk informasi selengkapnya, lihat [Membatasi akses ke AWS asal](#).

diaktifkan (wajib)

Ekspresi Boolean untuk mengaktifkan atau menonaktifkan OAC. Menerima `false` nilai `true` atau.

SigningBehavior (diperlukan saat diaktifkan)

Menentukan CloudFront tanda permintaan mana (menambahkan informasi otentikasi ke). Tentukan `always` untuk kasus penggunaan yang paling umum. Untuk informasi selengkapnya, lihat [Pengaturan lanjutan untuk kontrol akses asal](#).

Bidang ini dapat memiliki salah satu nilai berikut:

- `always`— CloudFront menandatangani semua permintaan asal, menimpa `Authorization` header dari permintaan penampil jika ada.
- `never`— CloudFront tidak menandatangani permintaan asal apa pun. Nilai ini mematikan kontrol akses asal untuk asal.
- `no-override`— Jika permintaan penampil tidak berisi `Authorization` header, maka CloudFront tandatangani permintaan asal. Jika permintaan penampil berisi `Authorization` header, maka CloudFront tidak menandatangani permintaan asal dan sebagai gantinya meneruskan `Authorization` header dari permintaan penampil.

 Warning

Untuk meneruskan `Authorization` header dari permintaan penampil, Anda harus menambahkannya ke kebijakan permintaan asal untuk semua perilaku cache yang menggunakan asal yang terkait dengan kontrol akses asal ini. Untuk informasi selengkapnya, lihat [Kontrol permintaan asal dengan kebijakan](#).

`SigningProtocol` (diperlukan saat diaktifkan)

Protokol penandatanganan OAC, yang menentukan bagaimana CloudFront tanda (mengotentikasi) permintaan. Satu-satunya nilai yang valid adalah `sigv4`.

`OriginType` (diperlukan saat diaktifkan)

Jenis asal untuk OAC ini. Nilai yang valid termasuk `s3`, `mediapackagev2`, `mediastore`, dan `lambda`.

batas waktu (opsional)

Timeout yang dapat Anda tentukan untuk berapa lama CloudFront harus mencoba menunggu asal merespons atau mengirim data. Jika ini tidak disediakan, maka pengaturan batas waktu dari asal yang ditetapkan digunakan.

 Note

Kecuali ditentukan, batas waktu ini mendukung asal kustom dan asal Amazon S3.

`ReadTimeout` (opsional)

`readTimeout` ini berlaku untuk kedua nilai berikut:

- Berapa lama (dalam detik) CloudFront menunggu respons setelah meneruskan permintaan ke asal.
- Berapa lama (dalam detik) CloudFront menunggu setelah menerima paket respons dari asal dan sebelum menerima paket berikutnya.

Batas waktu minimum adalah 1 detik dan maksimum 120 detik. Untuk informasi selengkapnya, lihat [Batas waktu respons](#).

respon CompletionTimeout (opsional)

Waktu (dalam detik) permintaan dari CloudFront ke asal dapat tetap terbuka dan menunggu tanggapan. Jika respons lengkap tidak diterima dari asal saat ini, CloudFront akhiri koneksi.

Nilai untuk `responseCompletionTimeout` harus sama dengan atau lebih besar dari nilai `untukreadTimeout`. Untuk informasi selengkapnya, lihat [Batas waktu penyelesaian respons](#).

simpan AliveTimeout (opsional)

Batas waktu ini hanya berlaku untuk asal kustom, bukan asal Amazon S3. (Konfigurasi asal S3 akan mengabaikan pengaturan ini.)

`keepAliveTimeout` Menentukan berapa lama CloudFront harus mencoba untuk mempertahankan koneksi ke asal setelah menerima paket terakhir dari respon. Batas waktu minimum adalah 1 detik dan maksimum 120 detik. Untuk informasi selengkapnya, lihat [Keep-alive batas waktu \(khusus dan hanya asal VPC\)](#).

ConnectionTimeout (opsional)

Jumlah detik yang CloudFront menunggu ketika mencoba membuat koneksi ke asal. Batas waktu minimum adalah 1 detik dan maksimum 10 detik. Untuk informasi selengkapnya, lihat [Batas waktu koneksi](#).

kustom OriginConfig (opsional)

Gunakan `customOriginConfig` untuk menentukan setelan koneksi untuk asal yang bukan bucket Amazon S3. Ada satu pengecualian: Anda dapat menentukan pengaturan ini jika bucket S3 dikonfigurasi dengan hosting situs web statis. (Jenis konfigurasi bucket S3 lainnya akan mengabaikan pengaturan ini.) Jika tidak `customOriginConfig` disediakan, maka pengaturan dari asal yang ditetapkan digunakan.

port (diperlukan)

Port HTTP yang CloudFront digunakan untuk terhubung ke asal. Tentukan port HTTP yang didengar oleh asal.

protokol (wajib)

Menentukan protokol (HTTP atau HTTPS) yang CloudFront digunakan untuk terhubung ke asal. Nilai yang valid adalah sebagai berikut:

- `http`— CloudFront selalu menggunakan HTTP untuk terhubung ke asal
- `https`— CloudFront selalu menggunakan HTTPS untuk terhubung ke asal

SSLProtocols (wajib)

Daftar yang menentukan SSL/TLS protokol minimum yang CloudFront digunakan saat menghubungkan ke asal Anda melalui HTTPS. Nilai yang valid termasuk SSLv3, TLSv1, TLSv1.1, dan TLSv1.2. Untuk informasi selengkapnya, lihat [Protokol SSL asal minimum](#).

ip AddressType (opsional)

Menentukan jenis alamat IP yang CloudFront digunakan untuk terhubung ke asal. Nilai yang valid termasuk `ipv4`, `ipv6`, dan `dualstack`. Perubahan `ipAddressType` hanya didukung ketika `domainName` properti juga sedang diubah.

sni (opsional, untuk asal kustom non-S3)

Server Name Indication (SNI) adalah ekstensi ke protokol Transport Layer Security (TLS) dimana klien menunjukkan nama host mana yang ingin disambungkan pada awal proses jabat tangan TLS. Nilai ini harus cocok dengan nama umum pada sertifikat TLS di server asal Anda. Jika tidak, server asal Anda mungkin akan menimbulkan kesalahan.

Jika ini tidak disediakan, nilai dari `hostHeader` parameter digunakan. Jika header host tidak disediakan, nilai dari `domainName` parameter digunakan.

Jika header host atau parameter nama domain tidak disediakan, nama domain dari asal yang ditetapkan akan digunakan atau header host dari permintaan masuk jika kebijakan penerusan ke asal (FTO) menyertakan host. SNI tidak dapat menyertakan titik dua (:) dan tidak dapat menjadi alamat IP. SNI dapat mencapai 253 karakter.

diizinkan CertificateNames (opsional, untuk asal kustom non-S3)

Anda dapat menyertakan daftar nama sertifikat yang valid untuk digunakan CloudFront untuk memvalidasi pencocokan domain dari sertifikat TLS server asal Anda selama jabat tangan TLS dengan server asal Anda. Bidang ini mengharapkan array nama domain yang valid dan dapat menyertakan domain wildcard, seperti `*.example.com`

Anda dapat menentukan hingga 20 nama sertifikat yang diizinkan. Setiap nama sertifikat dapat memiliki hingga 64 karakter.

Example- Perbarui ke asal permintaan Amazon S3

Contoh berikut mengubah asal permintaan viewer ke bucket S3, mengaktifkan OAC, dan mengatur ulang header kustom yang dikirim ke asal.

```
cf.updateRequestOrigin({
  "domainName" : "amzn-s3-demo-bucket-in-us-east-1.s3.us-east-1.amazonaws.com",
  "originAccessControlConfig": {
    "enabled": true,
    "signingBehavior": "always",
    "signingProtocol": "sigv4",
    "originType": "s3"
  },
  // Empty object resets any header configured on the assigned origin
  "customHeaders": {}
});
```

Example— Perbarui ke asal permintaan Application Load Balancer

Contoh berikut mengubah asal permintaan viewer ke asal Application Load Balancer dan menetapkan header dan batas waktu kustom.

```
cf.updateRequestOrigin({
  "domainName" : "example-1234567890.us-east-1.elb.amazonaws.com",
  "timeouts": {
    "readTimeout": 30,
    "connectionTimeout": 5
  },
  "customHeaders": {
    "x-stage": "production",
    "x-region": "us-east-1"
  }
});
```

Example— Perbarui ke asal dengan Origin Shield diaktifkan

Dalam contoh berikut, asal dalam distribusi mengaktifkan Origin Shield. Kode fungsi hanya memperbarui nama domain yang digunakan untuk asal dan menghilangkan semua parameter opsional lainnya. Dalam hal ini, Origin Shield akan tetap digunakan dengan nama domain asal yang dimodifikasi karena parameter Origin Shield tidak diperbarui.

```
cf.updateRequestOrigin({
  "domainName" : "www.example.com"
});
```

Example— Perbarui header host, SNI, dan nama sertifikat yang diizinkan

Warning

Untuk sebagian besar kasus penggunaan, Anda tidak perlu menggunakan jenis modifikasi ini untuk permintaan yang menuju ke asal Anda. Parameter ini tidak boleh digunakan kecuali Anda memahami dampak dari mengubah nilai-nilai ini.

Contoh berikut mengubah nama domain, header host, SNI, dan sertifikat yang diizinkan atas permintaan ke asal.

```
cf.updateRequestOrigin({
  "domainName": "www.example.com",
  "hostHeader": "test.example.com",
  "sni": "test.example.net",
  "allowedCertificateNames": ["*.example.com", "*.example.net"],
});
```

pilih `RequestOriginById()` metode

Gunakan `selectRequestOriginById()` untuk memperbarui asal yang ada dengan memilih asal berbeda yang sudah dikonfigurasi dalam distribusi Anda. Metode ini menggunakan semua pengaturan yang sama yang ditentukan oleh asal yang diperbarui.

Metode ini hanya menerima asal yang sudah ditentukan dalam distribusi yang sama yang digunakan saat menjalankan fungsi. Origins direferensikan oleh ID asal, yang merupakan nama asal yang Anda tentukan saat mengatur asal.

Jika Anda memiliki asal VPC yang dikonfigurasi dalam distribusi Anda, Anda dapat menggunakan metode ini untuk memperbarui asal Anda ke asal VPC Anda. Untuk informasi selengkapnya, lihat [Batasi akses dengan asal VPC](#).

Jika Anda memiliki asal dengan TLS timbal balik (mTLS) yang diaktifkan dalam distribusi Anda, Anda dapat menggunakan metode ini untuk memperbarui asal Anda ke asal mTLS-enabled Anda. Untuk informasi selengkapnya, lihat [Menggunakan CloudFront Fungsi dengan TLS timbal balik asal](#).

Permintaan

```
cf.selectRequestOriginById(origin_id, {origin_overrides})
```

Pada contoh sebelumnya, `origin_id` adalah string yang menunjuk ke nama asal asal dalam distribusi yang menjalankan fungsi. `origin_overrides` Parameter dapat berisi yang berikut:

Hostheader (opsional, untuk asal-usul kustom non-S3)

Header host untuk digunakan saat membuat permintaan ke asal. Jika ini tidak disediakan, nilai dari `domainName` parameter digunakan.

Jika header host atau parameter nama domain tidak disediakan, nama domain dari asal yang ditetapkan akan digunakan atau header host dari permintaan masuk jika kebijakan penerusan ke asal (FTO) menyertakan host. Header host tidak dapat menyertakan titik dua (:) dan tidak bisa menjadi alamat IP. Header host dapat mencapai 253 karakter.

sni (opsional, untuk asal kustom non-S3)

Server Name Indication (SNI) adalah ekstensi ke protokol Transport Layer Security (TLS) dimana klien menunjukkan nama host mana yang ingin disambungkan pada awal proses jabat tangan TLS. Nilai ini harus cocok dengan nama umum pada sertifikat TLS di server asal Anda. Jika tidak, server asal Anda mungkin akan menimbulkan kesalahan.

Jika ini tidak disediakan, nilai dari `hostHeader` parameter digunakan. Jika header host tidak disediakan, nilai dari `domainName` parameter digunakan.

Jika header host atau parameter nama domain tidak disediakan, nama domain dari asal yang ditetapkan akan digunakan atau header host dari permintaan masuk jika kebijakan penerusan ke asal (FTO) menyertakan host. SNI tidak dapat menyertakan titik dua (:) dan tidak dapat menjadi alamat IP. SNI dapat mencapai 253 karakter.

diizinkan CertificateNames (opsional, untuk asal kustom non-S3)

Anda dapat menyertakan daftar nama sertifikat yang valid untuk digunakan CloudFront untuk memvalidasi pencocokan domain dari sertifikat TLS server asal Anda selama jabat tangan TLS dengan server asal Anda. Bidang ini mengharapkan array nama domain yang valid dan dapat menyertakan domain wildcard, seperti `*.example.com`

Anda dapat menentukan hingga 20 nama sertifikat yang diizinkan. Setiap nama sertifikat dapat memiliki hingga 64 karakter.

Permintaan

```
selectRequestOriginById(origin_id)
```

Dalam contoh sebelumnya, `origin_id` adalah string yang menunjuk ke nama asal asal dalam distribusi yang menjalankan fungsi.

Example— Pilih asal permintaan Amazon S3

Contoh berikut memilih asal yang dinamai `amzn-s3-demo-bucket-in-us-east-1` dari daftar asal yang terkait dengan distribusi, dan menerapkan pengaturan konfigurasi `amzn-s3-demo-bucket-in-us-east-1` asal ke permintaan.

```
cf.selectRequestOriginById("amzn-s3-demo-bucket-in-us-east-1");
```

Example— Pilih asal permintaan Application Load Balancer

Contoh berikut memilih asal Application Load Balancer `myALB-prod` bernama dari daftar asal yang terkait dengan distribusi, dan menerapkan pengaturan konfigurasi untuk `myALB-prod` permintaan.

```
cf.selectRequestOriginById("myALB-prod");
```

Example— Pilih asal permintaan Application Load Balancer dan ganti header host

Seperti contoh sebelumnya, contoh berikut memilih asal Application Load Balancer yang `myALB-prod` dinamai dari daftar asal yang terkait dengan distribusi, dan menerapkan pengaturan konfigurasi untuk `myALB-prod` permintaan. Namun, contoh ini mengesampingkan nilai header host menggunakan `origin_overrides`

```
cf.overrideRequestOrigin("myALB-prod",{
    "hostHeader" : "test.example.com"
});
```

metode `createRequestOriginGroup()`

Gunakan `createRequestOriginGroup()` untuk mendefinisikan dua asal untuk digunakan sebagai [grup asal](#) untuk failover dalam skenario yang membutuhkan ketersediaan tinggi.

Grup asal mencakup dua asal (primer dan sekunder) dan kriteria failover yang Anda tentukan. Anda membuat grup asal untuk mendukung failover asal. CloudFront Saat membuat atau memperbarui grup asal menggunakan metode ini, Anda dapat menentukan grup asal, bukan satu asal. CloudFront akan failover dari asal primer ke asal sekunder, menggunakan kriteria failover.

Jika Anda memiliki asal VPC yang dikonfigurasi dalam distribusi Anda, Anda dapat menggunakan metode ini untuk membuat grup asal menggunakan asal VPC. Untuk informasi selengkapnya, lihat [Batasi akses dengan asal VPC](#).

Jika Anda memiliki asal dengan TLS timbal balik (mTLS) yang diaktifkan dalam distribusi Anda, Anda dapat menggunakan metode ini untuk membuat grup asal yang menyertakan asal mTLS-enabled Anda. Untuk informasi selengkapnya, lihat [Menggunakan CloudFront Fungsi dengan TLS timbal balik asal](#).

Permintaan

```
createRequestOriginGroup({origin_group_properties})
```

Dalam contoh sebelumnya, `origin_group_properties` dapat berisi yang berikut:

OriginIds (wajib)

Array dari `origin_ids`, di mana `origin_id` adalah string yang menunjuk ke nama asal asal dalam distribusi yang menjalankan fungsi. Anda harus memberikan dua asal sebagai bagian dari array. Asal pertama dalam daftar adalah asal primer dan yang kedua berfungsi sebagai asal kedua untuk tujuan failover.

OriginOverrides (opsional)

Beberapa pengaturan lanjutan diizinkan untuk ditimpa dengan menggunakan `{origin_overrides}` parameter. `origin_overrides` Dapat berisi yang berikut:

Hostheader (opsional, untuk asal-usul kustom non-S3)

Header host untuk digunakan saat membuat permintaan ke asal. Jika ini tidak disediakan, nilai dari `domainName` parameter digunakan.

Jika header host atau parameter nama domain tidak disediakan, nama domain dari asal yang ditetapkan akan digunakan atau header host dari permintaan masuk jika kebijakan penerusan ke asal (FTO) menyertakan host. Header host tidak dapat menyertakan titik dua (:) dan tidak bisa menjadi alamat IP. Header host dapat mencapai 253 karakter.

sni (opsional, untuk asal kustom non-S3)

Server Name Indication (SNI) adalah ekstensi ke protokol Transport Layer Security (TLS) dimana klien menunjukkan nama host mana yang ingin disambungkan pada awal proses

handshaking TLS. Nilai ini harus cocok dengan nama umum pada sertifikat TLS di server asal Anda, jika tidak, server asal Anda dapat menimbulkan kesalahan.

Jika ini tidak disediakan, nilai dari `hostHeader` parameter digunakan. Jika header host tidak disediakan, nilai dari `domainName` parameter digunakan.

Jika header host atau parameter nama domain tidak disediakan, nama domain dari asal yang ditetapkan akan digunakan atau header host dari permintaan masuk jika kebijakan penerusan ke asal (FTO) menyertakan host. SNI tidak dapat menyertakan titik dua (:) dan tidak dapat menjadi alamat IP. SNI dapat mencapai 253 karakter.

diizinkan `CertificateNames` (opsional, untuk asal kustom non-S3)

Anda dapat menyertakan daftar nama sertifikat yang valid untuk digunakan CloudFront untuk memvalidasi pencocokan domain dari sertifikat TLS server asal Anda selama jabat tangan TLS dengan server asal Anda. Bidang ini mengharapkan array nama domain yang valid dan dapat menyertakan domain wildcard, seperti `*.example.com`

Anda dapat menentukan hingga 20 nama sertifikat yang diizinkan. Setiap nama sertifikat dapat memiliki hingga 64 karakter.

`SelectionCriteria` (opsional)

Pilih apakah akan menggunakan kriteria failover default asal atau menggunakan logika failover `media-quality-score` berbasis. Nilai yang valid adalah sebagai berikut:

- `default` menggunakan kriteria failover, berdasarkan kode status yang ditentukan dalam `failoverCriteria`. Jika Anda tidak mengatur `selectionCriteria` fungsi, default akan digunakan.
- `media-quality-score` digunakan ketika kemampuan perutean sadar media sedang digunakan.

`FailOverCriteria` (wajib)

Array kode status yang, ketika dikembalikan dari asal primer, akan memicu failover CloudFront ke asal sekunder. Jika Anda mengisi grup asal yang ada, larik ini akan mengisi semua kode status failover yang ditetapkan dalam konfigurasi asli grup asal.

Saat Anda menggunakan `media-quality-score` `selectionCriteria`, CloudFront akan mencoba untuk merutekan permintaan berdasarkan skor kualitas media. Jika asal yang dipilih mengembalikan kode kesalahan yang diatur dalam array ini, CloudFront akan failover ke asal lainnya.

Example— Buat grup asal permintaan

Contoh berikut membuat grup asal untuk permintaan menggunakan ID asal. ID asal ini berasal dari konfigurasi grup asal untuk distribusi yang digunakan untuk menjalankan fungsi ini.

Secara opsional, Anda dapat menggunakan `originOverrides` untuk mengganti konfigurasi grup asal untuk `sni`, `hostHeader` dan `allowedCertificateNames`

```
import cf from 'cloudfront';

function handler(event) {
  cf.createRequestOriginGroup({
    "originIds": [
      {
        "originId": "origin-1",
        "originOverrides": {
          "hostHeader": "hostHeader.example.com",
          "sni": "sni.example.com",
          "allowedCertificateNames": ["cert1.example.com",
"cert2.example.com", "cert3.example.com"]
        }
      },
      {
        "originId": "origin-2",
        "originOverrides": {
          "hostHeader": "hostHeader2.example.com",
          "sni": "sni2.example.com",
          "allowedCertificateNames": ["cert4.example.com",
"cert5.example.com"]
        }
      }
    ],
    "failoverCriteria": {
      "statusCodes": [500]
    }
  });

  event.request.headers['x-hookx'] = { value: 'origin-overrides' };
  return event.request;
}
```

Metode pembantu untuk properti CloudFront SaaS Manager

Gunakan fungsi pembantu berikut untuk CloudFront SaaS Manager untuk mengambil nilai untuk distribusi multi-penyewa Anda dalam fungsi yang Anda buat. Untuk menggunakan contoh di halaman ini, Anda harus terlebih dahulu membuat CloudFront fungsi dengan menggunakan JavaScript runtime 2.0. Untuk informasi selengkapnya, lihat [JavaScript fitur runtime 2.0 untuk Fungsi CloudFront](#).

Topik

- [Grup koneksi](#)
- [Penyewa distribusi](#)

Grup koneksi

Grup koneksi yang terkait dengan penyewa distribusi Anda memiliki nama domain.

Untuk mendapatkan nilai ini, gunakan endpoint bidang context subobject dari objek acara.

Permintaan

```
const value = event.context.endpoint;
```

Respons

Responsnya adalah string yang berisi nama domain grup koneksi, seperti `d111111abcdef8.cloudfront.net`. `endpointBidang` hanya muncul ketika fungsi Anda dipanggil untuk distribusi multi-penyewa dengan grup koneksi terkait. Untuk informasi selengkapnya, lihat [Objek konteks](#).

Penyewa distribusi

CloudFront Fungsi memiliki modul yang menyediakan akses ke nilai penyewa distribusi tertentu.

Untuk menggunakan modul ini, sertakan pernyataan berikut di baris pertama kode fungsi Anda:

```
import cf from 'cloudfront';
```

Anda dapat menggunakan contoh berikut hanya dalam `handler` fungsi, baik secara langsung atau melalui fungsi panggilan bersarang.

Tenant.idbidang **distribusi**

Gunakan bidang ini untuk mendapatkan nilai ID penyewa distribusi.

Permintaan

```
const value = cf.distributionTenant.id;
```

Respons

Responsnya adalah string yang berisi ID penyewa distribusi, seperti `dt_1a2b3c4d5e6f7`.

Penanganan kesalahan

Jika fungsi Anda dipanggil untuk distribusi standar, menentukan `distributionTenant.id` bidang akan mengembalikan kesalahan `distributionTenant module is not available` tipe. Untuk menangani kasus penggunaan ini, Anda dapat menambahkan try dan catch memblokir kode Anda.

metode **distribusi Tenant.parameters.get ()**

Gunakan metode ini untuk mengembalikan nilai untuk nama parameter penyewa distribusi yang Anda tentukan.

```
distributionTenant.parameters.get("key");
```

key: Nama parameter penyewa distribusi yang ingin Anda ambil nilainya.

Permintaan

```
const value = distributionTenant.parameters.get("key");
```

Respons

Responsnya adalah a string yang berisi nilai untuk parameter penyewa distribusi. Misalnya, jika nama kunci Anda adalah `TenantPath`, maka nilai untuk parameter ini mungkin `tenant1`.

Penanganan kesalahan

Anda mungkin menerima kesalahan berikut:

- Jika fungsi Anda dipanggil untuk distribusi standar, `distributionTenant.parameters.get()` metode akan mengembalikan kesalahan `distributionTenant module is not available` tipe.
- `DistributionTenantParameterKeyNotFound` kesalahan dikembalikan ketika parameter penyewa distribusi yang Anda tentukan tidak ada.

Untuk mengelola kasus penggunaan ini, Anda dapat menambahkan `try` dan `catch` memblokir kode Anda.

Gunakan asinkron and menunggu

CloudFront Fungsi JavaScript runtime function 2.0 menyediakan `async` dan `await` sintaks untuk menangani `Promise` objek. Janji mewakili hasil tertunda yang dapat diakses melalui `await` kata kunci dalam fungsi yang ditandai sebagai `async`. Berbagai WebCrypto fungsi baru menggunakan `Promises`.

Untuk informasi selengkapnya tentang `Promise` objek, lihat [Janji](#).

Note

Anda harus menggunakan JavaScript runtime 2.0 untuk contoh kode berikut. `await` hanya dapat digunakan di dalam `async` fungsi. `async` argumen dan penutupan tidak didukung.

```
async function answer() {
    return 42;
}

// Note: async, await can be used only inside an async function. async arguments and
// closures are not supported.

async function handler(event) {
    // var answer_value = answer(); // returns Promise, not a 42 value
    let answer_value = await answer(); // resolves Promise, 42
    console.log("Answer"+answer_value);
    event.request.headers['answer'] = { value : ""+answer_value };
    return event.request;
}
```

JavaScript Kode contoh berikut menunjukkan cara melihat janji dengan metode `then` rantai. Anda dapat menggunakan `catch` untuk melihat kesalahan.

Warning

Menggunakan kombinator janji (misalnya, `Promise.all`, `Promise.any`, dan metode rantai janji (misalnya, `then` dan `catch`) dapat memerlukan penggunaan memori fungsi tinggi. Jika fungsi Anda melebihi kuota [memori fungsi maksimum](#), itu akan gagal dijalankan. Untuk menghindari kesalahan ini, kami sarankan Anda menggunakan `await` sintaks alih-alih `promise` metode.

```
async function answer() {
    return 42;
}

async function squared_answer() {
    return answer().then(value => value * value)
}

// Note: async, await can be used only inside an async function. async arguments and
// closures are not supported.
async function handler(event) {
    // var answer_value = answer(); // returns Promise, not a 42 value
    let answer_value = await squared_answer(); // resolves Promise, 42
    console.log("Answer"+answer_value);
    event.request.headers['answer'] = { value : ""+answer_value };
    return event.request;
}
```

Dukungan CWT untuk Fungsi CloudFront

Bagian ini memberikan detail tentang dukungan untuk CBOR Web Tokens (CWT) di CloudFront Fungsi Anda, yang memungkinkan otentikasi dan otorisasi berbasis token yang aman di Lokasi Edge. CloudFront Dukungan ini disediakan sebagai modul, dapat diakses di CloudFront Fungsi Anda.

Untuk menggunakan modul ini, buat CloudFront Function menggunakan JavaScript runtime 2.0 dan sertakan pernyataan berikut di baris pertama kode fungsi:

```
import cf from 'cloudfront';
```

Metode yang terkait dengan modul ini dapat diakses melalui (di mana * adalah wildcard yang mewakili berbagai fungsi yang ada dalam modul):

```
cf.cwt.*
```

Untuk informasi selengkapnya, lihat [JavaScript fitur runtime 2.0 untuk Fungsi CloudFront](#).

Saat ini, modul hanya mendukung struktur MAC0 dengan algoritma HS256 (HMAC-SHA256) dengan batas 1KB untuk ukuran token maksimum.

Struktur token

Bagian ini mencakup struktur token yang diharapkan oleh modul CWT. Modul mengharapkan token ditandai dengan benar dan dapat diidentifikasi (misalnya COSE MAC0). Selain itu, untuk struktur token, modul mengikuti standar yang ditetapkan oleh [CBOR Object Signing and Encryption \(COSE\) \[RFC 8152\]](#).

```
( // CWT Tag (Tag value: 61) --- optional
  ( // COSE MAC0 Structure Tag (Tag value: 17) --- required
    [
      protectedHeaders,
      unprotectedHeaders,
      payload,
      tag,
    ]
  )
)
```

Example: CWT menggunakan struktur COSE MAC0

```
61( // CWT tag
  17( // COSE_MAC0 tag
    [
      { // Protected Headers
        1: 4 // algorithm : HMAC-256-64
      },
      { // Unprotected Headers
        4: h'53796d6d6574726963323536' // kid : Symmetric key id
      },
      { // Payload
        1: "https://iss.example.com", // iss
      }
    ]
  )
)
```

```
        2: "exampleUser", // sub
        3: "https://aud.example.com", // aud
        4: 1444064944, // exp
        5: 1443944944, // nbf
        6: 1443944944, // iat
      },
      h'093101ef6d789200' // tag
    ]
  )
)
```

Note

Tag CWT bersifat opsional saat menghasilkan token. Namun, tag struktur COSE diperlukan.

metode `validateToken ()`

Fungsi menerjemahkan dan memvalidasi token CWT menggunakan kunci yang ditentukan. Jika validasi berhasil, ia mengembalikan token CWT yang diterjemahkan. Jika tidak, itu akan menimbulkan kesalahan. Harap dicatat bahwa fungsi ini tidak melakukan validasi pada set klaim.

Permintaan

```
cf.cwt.validateToken(token, handlerContext{key})
```

Parameter

token (wajib)

Token yang dikodekan untuk validasi. Ini pasti JavaScript Buffer.

HandlerContext (Diperlukan)

JavaScript Objek yang menyimpan konteks untuk panggilan `ValidateToken`. Saat ini, hanya properti kunci yang didukung.

kunci (Diperlukan)

Kunci rahasia untuk perhitungan intisari pesan. Dapat disediakan baik sebagai string atau JavaScript Buffer.

Respons

Ketika `validateToken()` metode mengembalikan token berhasil divalidasi, respon dari fungsi adalah `CWTObject` dalam format berikut. Setelah diterjemahkan, semua kunci klaim direpresentasikan sebagai string.

```
CWTObject {  
  protectedHeaders,  
  unprotectedHeaders,  
  payload  
}
```

Contoh - Validasi token dengan anak dikirim sebagai bagian dari token

Contoh ini menunjukkan validasi token CWT, di mana anak diekstraksi dari header. Anak itu kemudian diteruskan ke CloudFront Fungsi KeyValueCollection untuk mengambil kunci rahasia yang digunakan untuk memvalidasi token.

```
import cf from 'cloudfront'  
  
const CwtClaims = {  
  iss: 1,  
  aud: 3,  
  exp: 4  
}  
  
async function handler(event) {  
  try {  
    let request = event.request;  
    let encodedToken = request.headers['x-cwt-token'].value;  
    let kid = request.headers['x-cwt-kid'].value;  
  
    // Retrieve the secret key from the kvs  
    let secretKey = await cf.kvs().get(kid);  
  
    // Now you can use the secretKey to decode & validate the token.  
    let tokenBuffer = Buffer.from(encodedToken, 'base64url');  
  
    let handlerContext = {  
      key: secretKey,  
    }  
  }  
}
```

```
    try {
        let cwtObj = cf.cwt.validateToken(tokenBuffer, handlerContext);

        // Check if token is expired
        const currentTime = Math.floor(Date.now() / 1000); // Current time in
seconds
        if (cwtObj[CwtClaims.exp] && cwtObj[CwtClaims.exp] < currentTime) {
            return {
                statusCode: 401,
                statusDescription: 'Token expired'
            };
        }
    } catch (error) {
        return {
            statusCode: 401,
            statusDescription: 'Invalid token'
        };
    }
} catch (error) {
    return {
        statusCode: 402,
        statusDescription: 'Token processing failed'
    };
}
return request;
}
```

metode generateToken ()

Fungsi ini menghasilkan token CWT baru menggunakan pengaturan payload dan konteks yang disediakan.

Permintaan

```
cf.cwt.generateToken(generatorContext, payload)
```

Parameter

GeneratorContext (Diperlukan)

Ini JavaScript Object yang digunakan sebagai konteks untuk menghasilkan token dan berisi pasangan nilai kunci berikut:

CWTTag (Opsional)

Nilai ini adalah boolean, yang jika `true` menentukan `cwtTag` harus ditambahkan.

CoseTag (Diperlukan)

Menentukan jenis tag COSE. Saat ini hanya mendukung `MAC0`.

kunci (Diperlukan)

Kunci rahasia untuk menghitung intisari pesan. Nilai ini dapat berupa string atau JavaScript `Buffer`.

payload (Wajib)

Payload token untuk pengkodean. Muatan harus dalam `CWTObject` format.

Respons

Mengembalikan JavaScript Buffer yang berisi token dikodekan.

Example: Menghasilkan token CWT

```
import cf from 'cloudfront';

const CwtClaims = {
  iss: 1,
  sub: 2,
  exp: 4
};

const CatClaims = {
  catu: 401,
  catnip: 402,
  catm: 403,
  catr: 404
};

const Catu = {
  host: 1,
  path: 2,
  ext: 3
};

const CatuMatchTypes = {
```

```
    prefix_match: 1,
    suffix_match: 2,
    exact_match: 3
  };

  const Catr = {
    renewal_method: 1,
    next_renewal_time: 2,
    max_uses: 3
  };

  async function handler(event) {
    try {
      const response = {
        statusCode: 200,
        statusDescription: 'OK',
        headers: {}
      };

      const commonAccessToken = {
        protected: {
          1: "5",
        },
        unprotected: {},
        payload: {
          [CwtClaims.iss]: "cloudfront-documentation",
          [CwtClaims.sub]: "cwt-support-on-cloudfront-functions",
          [CwtClaims.exp]: 1740000000,
          [CatClaims.catu]: {
            [Catu.host]: {
              [CatuMatchTypes.suffix_match]: ".cloudfront.net"
            },
            [Catu.path]: {
              [CatuMatchTypes.prefix_match]: "/media/live-stream/cf-4k/"
            },
            [Catu.ext]: {
              [CatuMatchTypes.exact_match]: [
                ".m3u8",
                ".ts",
                ".mpd"
              ]
            }
          },
          [CatClaims.catnip]: [
```

```

        "[IP_ADDRESS]",
        "[IP_ADDRESS]"
    ],
    [CatClaims.catm]: [
        "GET",
        "HEAD"
    ],
    [CatClaims.catr]: {
        [Catr.renewal_method]: "header_renewal",
        [Catr.next_renewal_time]: 1750000000,
        [Catr.max_uses]: 5
    }
}
};

if (!request.headers['x-cwt-kid']) {
    throw new Error('Missing x-cwt-kid header');
}

const kid = request.headers['x-cwt-kid'].value;
const secretKey = await cf.kvs().get(kid);

if (!secretKey) {
    throw new Error('Secret key not found for provided kid');
}

try {
    const genContext = {
        cwtTag: true,
        coseTag: "MAC0",
        key: secretKey
    };

    const tokenBuffer = cf.cwt.generateToken(commonAccessToken, genContext);
    response.headers['x-generated-cwt-token'] = { value:
tokenBuffer.toString('base64url') };

    return response;
} catch (tokenError) {
    return {
        statusCode: 401,
        statusDescription: 'Could not generate the token'
    };
}
}

```

```

    } catch (error) {
      return {
        statusCode: 402,
        statusDescription: 'Token processing failed'
      };
    }
  }
}

```

Example: Segarkan token berdasarkan beberapa logika

```

import cf from 'cloudfront'

const CwtClaims = {
  iss: 1,
  aud: 3,
  exp: 4
}

async function handler(event) {
  try {
    let request = event.request;
    let encodedToken = request.headers['x-cwt-token'].value;
    let kid = request.headers['x-cwt-kid'].value;
    let secretKey = await cf.kvs().get(kid); // Retrieve the secret key from the
    kvs

    // Now you can use the secretKey to decode & validate the token.
    let tokenBuffer = Buffer.from(encodedToken, 'base64url');

    let handlerContext = {
      key: secretKey,
    }

    try {
      let cwtJSON = cf.cwt.validateToken(tokenBuffer, handlerContext);

      // Check if token is expired
      const currentTime = Math.floor(Date.now() / 1000); // Current time in
      seconds

      if (cwtJSON[CwtClaims.exp] && cwtJSON[CwtClaims.exp] < currentTime) {
        // We can regenerate the token and add 8 hours to the expiry time
        cwtJSON[CwtClaims.exp] = Math.floor(Date.now() / 1000) + (8 * 60 * 60);
      }
    }
  }
}

```

```

        let genContext = {
            coseTag: "MAC0",
            key: secretKey
        }

        let newTokenBuffer = cf.cwt.generateToken(cwtJSON, genContext);
        request.headers['x-cwt-regenerated-token'] =
newTokenBuffer.toString('base64url');
    }
    } catch (error) {
        return {
            statusCode: 401,
            statusDescription: 'Invalid token'
        };
    }
}
catch (error) {
    return {
        statusCode: 402,
        statusDescription: 'Token processing failed'
    };
}
return request;
}

```

Metode pembantu umum

Halaman ini menyediakan metode pembantu tambahan di dalam CloudFront Fungsi. Untuk menggunakan metode ini, buat CloudFront fungsi menggunakan JavaScript runtime 2.0.

```
import cf from 'cloudfront';
```

Untuk informasi selengkapnya, lihat [JavaScript fitur runtime 2.0 untuk Fungsi CloudFront](#).

edgeLocationmetadata

Metode ini membutuhkan penggunaan cloudfront modul.

Note

Anda hanya dapat menggunakan metode ini untuk fungsi permintaan pemirsa. Untuk fungsi respons pemirsa, metode ini kosong.

Gunakan JavaScript objek ini untuk mendapatkan kode bandara lokasi tepi, wilayah [Regional Edge Cache](#) yang diharapkan atau alamat IP CloudFront server yang digunakan untuk menangani permintaan. Metadata ini hanya tersedia pemicu peristiwa permintaan penampil.

```
cf.edgeLocation = {  
  name: SEA  
  serverIp: 1.2.3.4  
  region: us-west-2  
}
```

`cf.edgeLocation`Objek dapat berisi yang berikut:

`name`

[Kode IATA](#) tiga huruf dari lokasi tepi yang menangani permintaan.

`ServerIP`

IPv6 Alamat IPv4 atau server yang menangani permintaan.

`region`

CloudFront Regional Edge Cache (REC) yang diharapkan akan digunakan permintaan jika ada cache yang hilang. Nilai ini tidak diperbarui jika REC yang diharapkan tidak tersedia dan REC cadangan digunakan untuk permintaan tersebut. Ini tidak termasuk lokasi Origin Shield yang digunakan, kecuali jika REC utama dan Origin Shield berada di lokasi yang sama.

Note

CloudFront Fungsi tidak dipanggil untuk kedua kalinya ketika CloudFront dikonfigurasi untuk menggunakan failover asal. Untuk informasi selengkapnya, lihat [Optimalkan ketersediaan tinggi dengan failover CloudFront asal](#).

Metode `rawQueryString()`

Metode ini tidak memerlukan `cloudFront` modul.

Gunakan `rawQueryString()` metode ini untuk mengambil string kueri yang tidak diurai dan tidak diubah sebagai string.

Permintaan


```
function handler(event) {  
    var request = event.request;  
    const qs = request.rawQueryString();  
}
```

Respons

Mengembalikan string query lengkap dari permintaan masuk sebagai nilai string tanpa memimpin?.

- Jika tidak ada string query, tetapi ? hadir, fungsi mengembalikan string kosong.
- Jika tidak ada string kueri dan ? tidak hadir, fungsi kembali undefined.

Kasus 1: String kueri lengkap dikembalikan (tanpa memimpin?)

URL permintaan masuk: `https://example.com/page?name=John&age=25&city=Boston`

`rawQueryString()` mengembalikan: `"name=John&age=25&city=Boston"`

Kasus 2: String kosong dikembalikan (kapan ? ada tetapi tanpa parameter)

URL permintaan masuk: `https://example.com/page?`

`rawQueryString()` mengembalikan: `""`

Kasus 3: **undefined** dikembalikan (tidak ada string kueri dan tidak?)

URL permintaan masuk: `https://example.com/page`

`rawQueryString()` mengembalikan: `undefined`

Metode pembantu untuk TLS timbal balik

CloudFront menyediakan fungsi m TLS-specific helper di `cf.mtls` namespace untuk fungsi permintaan penampil. Fungsi-fungsi ini mengganti nama, memformat ulang, atau menggabungkan header sertifikat klien sebelum meneruskan permintaan ke asal Anda.

```
import cf from "cloudfront";
```

cf.mtls.ganti nama ClientCertHeaders

Mengganti nama header metadata sertifikat menjadi nama header khusus.

```
cf.mtls.renameClientCertHeaders({
  "CloudFront-Viewer-Cert-Pem": "X-Client-Cert",
  "CloudFront-Viewer-Cert-Serial-Number": "X-Client-Cert-Serial",
  "CloudFront-Viewer-Cert-Issuer": "X-Client-Cert-Issuer",
});
```

Header sumber yang diizinkan:

- CloudFront-Viewer-Cert-Pem
- CloudFront-Viewer-Cert-Serial-Number
- CloudFront-Viewer-Cert-Issuer
- CloudFront-Viewer-Cert-Subject
- CloudFront-Viewer-Cert-Validity
- CloudFront-Viewer-Cert-Present
- CloudFront-Viewer-Cert-Sha256
- Client-Cert
- Client-Cert-Chain

cf.mtls.ganti nama PemHeaders

Mengganti nama header sertifikat PEM dan secara opsional memformat ulang pengkodean sertifikat.

```
cf.mtls.renamePemHeaders({
  "Client-Cert": {
    "newHeaderName": "X-Leaf-Cert",
    "pemCertFormatInfo": {
      "certHeader": "-----CUSTOM HEADER-----",
      "certFooter": "-----CUSTOM FOOTER-----",
      "certEndMarker": "",
      "keepNewlinesInCertData": true
    }
  },
  "Client-Cert-Chain": {
    "newHeaderName": "X-Intermediate-Certs",
    "pemCertFormatInfo": {
      "certHeader": "-----CUSTOM HEADER-----",
      "certFooter": "-----CUSTOM FOOTER-----",

```

```

        "certEndMarker": "",
        "keepNewlinesInCertData": true
    }
}
});

```

Header sumber yang diizinkan:

- Dalam mode passthrough: Client-Cert, Client-Cert-Chain
- Dalam required/optional mode: Cloudfront-Viewer-Cert-PEM

pemCertFormatInfo bidang:

Untuk Cloudfront-Viewer-Cert-PEM:

- certHeader menggantikan -----BEGIN CERTIFICATE-----.
- certFooter menggantikan -----END CERTIFICATE-----.
- certEndMarker menetapkan string kustom setelah CertFooter.
- keepNewlinesInCertData(default:true) mempertahankan baris baru dalam data base64 saat benar.

Untuk Client-Cert dan Client-Cert-Chain:

- certHeader menggantikan:.
- certFooter menggantikan:.
- certEndMarker menetapkan string kustom setelah CertFooter.
- keepNewlinesInCertData(default:false) mempertahankan baris baru dalam data base64 saat benar.

cf.mtls.combine PemHeaders

Menggabungkan Client-Cert dan Client-Cert-Chain menjadi satu header yang berisi rantai sertifikat lengkap.

```

cf.mtls.combinePemHeaders({
    "newHeaderName": "X-Full-Chain",

```

```
"pemCertFormatInfo": {  
  "certHeader": "-----BEGIN CERTIFICATE-----",  
  "certFooter": "-----END CERTIFICATE-----",  
  "certEndMarker": "\\n",  
  "keepNewlinesInCertData": false  
}  
});
```

`certEndMarker` menetapkan pembatas antara sertifikat.

Note

- Fungsi pembantu ini dapat digunakan di semua mode mTLS (wajib, opsional, dan passthrough). Namun, `cf.mtls.combinePemHeaders` hanya berpengaruh dalam mode passthrough — dalam mode wajib dan opsional, `Client-Cert-Chain` header `Client-Cert` dan tidak ada, jadi fungsinya adalah no-op.
- Jika `cf.mtls.*` metode dan `cf.updateRequestOrigin()` dengan `customHeaders` target nama header yang sama, CloudFront mengembalikan kesalahan 502.
- Dalam mode Passthrough, CloudFront jatuhkan semua yang masuk `Client-Cert` atau `Client-Cert-Chain` header dan tambahkan kembali dari sertifikat klien yang sebenarnya.
- CloudFront tidak menyajikan konten sertifikat PEM mentah lengkap dalam fungsi tepi apa pun.

Buat fungsi

Anda membuat fungsi dalam dua tahap:

1. Buat kode fungsi sebagai JavaScript. Anda dapat menggunakan contoh default dari CloudFront konsol atau menulis sendiri. Untuk informasi selengkapnya, lihat topik berikut:
 - [Tulis kode fungsi](#)
 - [the section called “Struktur peristiwa”](#)
 - [CloudFront Contoh fungsi untuk CloudFront](#)
2. Gunakan CloudFront untuk membuat fungsi dan sertakan kode Anda. Kode ada di dalam fungsi (bukan sebagai referensi).

Console

Untuk membuat fungsi

1. Masuk ke CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home#/functions> dan pilih halaman Fungsi.
2. Pilih Buat fungsi.
3. Masukkan nama fungsi yang unik di dalam Akun AWS, pilih JavaScript versi, lalu pilih Lanjutkan. Halaman detail untuk fungsi baru muncul.

Note

Untuk menggunakan [pasangan kunci-nilai](#) dalam fungsi, Anda harus memilih JavaScript runtime 2.0.

Jika Anda menemukan kesalahan `TaggingConflict` (HTTP 409) saat membuat fungsi dengan tag, tunggu beberapa menit dan coba lagi, atau buat fungsi tanpa tag pada awalnya dan tambahkan sesudahnya.

4. Di bagian Kode fungsi, pilih tab Build dan masukkan kode fungsi Anda. Kode contoh yang disertakan dalam tab Build mengilustrasikan sintaks dasar untuk kode fungsi.
5. Pilih Simpan perubahan.
6. Jika kode fungsi menggunakan pasangan kunci-nilai, Anda harus mengaitkan penyimpanan nilai kunci.

Anda dapat mengaitkan penyimpanan nilai kunci saat pertama kali membuat fungsi. Atau, Anda dapat mengaitkannya nanti, dengan [memperbarui fungsi](#).

Untuk mengaitkan penyimpanan nilai kunci sekarang, ikuti langkah-langkah berikut:

- Buka KeyValueCollection bagian Associate dan pilih Associate existing KeyValueCollection.
- Pilih penyimpanan nilai kunci yang berisi pasangan kunci-nilai dalam fungsi, lalu pilih Associate. KeyValueCollection

CloudFront segera mengaitkan toko dengan fungsinya. Anda tidak perlu menyimpan fungsinya.

CLI

Jika Anda menggunakan CLI, Anda biasanya pertama membuat kode fungsi dalam file, dan kemudian membuat fungsi dengan AWS CLI

Untuk membuat fungsi

1. Buat kode fungsi dalam file, dan simpan di direktori tempat komputer Anda dapat terhubung.
2. Jalankan perintah seperti yang ditunjukkan pada contoh. Contoh ini menggunakan `fileb://` notasi untuk meneruskan file. Ini juga termasuk jeda baris untuk membuat perintah lebih mudah dibaca.

```
aws cloudfront create-function \  
  --name MaxAge \  
  --function-config '{"Comment":"Max Age 2 years","Runtime":"cloudfront-  
js-2.0","KeyValueStoreAssociations":{"Quantity":1,"Items":  
[{"KeyValueStoreARN":"arn:aws:cloudfront::111122223333:key-value-store/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"}]}' \  
  --function-code fileb://function-max-age-v1.js
```

Catatan

- Runtime— Versi JavaScript. Untuk menggunakan [pasangan nilai kunci](#) dalam fungsi, Anda harus menentukan versi 2.0.
- KeyValueStoreAssociations— Jika fungsi Anda menggunakan pasangan kunci-nilai, Anda dapat mengaitkan penyimpanan nilai kunci saat pertama kali membuat fungsi. Atau, Anda dapat mengaitkannya nanti, dengan menggunakan `update-function`. `Quantity` itu selalu 1 karena setiap fungsi hanya dapat memiliki satu penyimpanan nilai kunci yang terkait dengannya.

Ketika perintah berhasil, Anda melihat output seperti berikut ini.

```
ETag: ETVABCEXAMPLE  
FunctionSummary:  
  FunctionConfig:  
    Comment: Max Age 2 years  
    Runtime: cloudfront-js-2.0  
    KeyValueStoreAssociations= \  
    
```

```
{Quantity=1, \
  Items=[{KeyValueStoreARN='arn:aws:cloudfront::111122223333:key-value-
store/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111'}]] \
  FunctionMetadata:
    CreatedTime: '2021-04-18T20:38:56.915000+00:00'
    FunctionARN: arn:aws:cloudfront::111122223333:function/MaxAge
    LastModifiedTime: '2023-11-19T20:38:56.915000+00:00'
    Stage: DEVELOPMENT
  Name: MaxAge
  Status: UNPUBLISHED
Location: https://cloudfront.amazonaws.com/2020-05-31/function/
arn:aws:cloudfront::function/MaxAge
```

Sebagian besar informasi diulang dari permintaan. Informasi lain ditambahkan oleh CloudFront.

Catatan

- ETag— Nilai ini berubah setiap kali Anda memodifikasi penyimpanan nilai kunci. Anda menggunakan nilai ini dan nama fungsi untuk mereferensikan fungsi di masa depan. Pastikan Anda selalu menggunakan arusETag.
- FunctionARN— ARN untuk fungsi Anda CloudFront.
- 111122223333 —. Akun AWS
- Stage— Tahap fungsi (LIVEatauDEVELOPMENT).
- Status— Status fungsi (PUBLISHEDatauUNPUBLISHED).

Setelah Anda membuat fungsi, itu ditambahkan ke DEVELOPMENT panggung. Kami menyarankan Anda [menguji fungsi Anda](#) sebelum [mempublikasikannya](#). Setelah Anda mempublikasikan fungsi Anda, fungsi berubah ke LIVE panggung.

TaggingConflict kesalahan saat membuat fungsi

CloudFront Fungsi digunakan kembali ARNs untuk fungsi dengan nama yang sama. Anda mungkin mengalami kesalahan TaggingConflict (HTTP 409) saat membuat fungsi dengan tag. Ini dapat terjadi dalam skenario berikut:

- Setelah menghapus fungsi — Membuat ulang fungsi dengan nama dan tag yang sama segera setelah penghapusan.
- Setelah upaya pembuatan gagal — Jika permintaan pembuatan sebelumnya gagal (misalnya, karena parameter tidak valid, asosiasi KVS tidak valid, atau pengembalian tumpukan yang gagal). CloudFormation

Jika Anda menerima `TaggingConflict` kesalahan, Anda dapat:

- Tunggu beberapa menit dan coba lagi operasi.
- Gunakan nama fungsi yang berbeda. Ini menghindari konflik penggunaan kembali ARN sepenuhnya.
- Jika template Anda membuat fungsi `KeyValueStore` dan fungsi yang mereferensikannya, tambahkan eksplisit `DependsOn` untuk memastikan KVS sepenuhnya disediakan sebelum fungsi dibuat.

Fungsi uji

Sebelum Anda menerapkan fungsi ke live stage (produksi), Anda dapat menguji fungsi Anda untuk memverifikasi bahwa fungsi berfungsi sebagaimana dimaksud. Untuk menguji fungsi, Anda menentukan objek peristiwa yang mewakili permintaan HTTP atau respons yang dapat diterima CloudFront distribusi Anda dalam produksi.

CloudFront Fungsi melakukan hal berikut:

1. Menjalankan fungsi, menggunakan objek acara yang disediakan sebagai input.
2. Mengembalikan hasil fungsi (objek peristiwa yang dimodifikasi) bersama dengan log fungsi atau pesan kesalahan dan pemanfaatan komputasi fungsi. Untuk informasi lebih lanjut tentang pemanfaatan komputasi, lihat. [the section called “Memahami pemanfaatan komputasi”](#)

Note

Saat Anda menguji suatu fungsi, CloudFront hanya memvalidasi terhadap kesalahan eksekusi fungsi. CloudFront tidak memvalidasi apakah permintaan akan berhasil mengalir setelah diterbitkan. Misalnya, jika fungsi Anda menghapus header yang diperlukan, pengujian akan berhasil karena tidak ada masalah dengan kode. Namun, jika Anda mempublikasikan

fungsi dan mengaitkannya dengan distribusi, fungsi tersebut akan gagal saat permintaan dibuat CloudFront.

Daftar Isi

- [Mengatur objek acara](#)
- [Uji fungsi](#)
- [Memahami pemanfaatan komputasi](#)

Mengatur objek acara

Sebelum Anda menguji suatu fungsi, Anda harus mengatur objek acara untuk mengujinya. Ada beberapa opsi.

Opsi 1: Siapkan objek acara tanpa menyimpannya

Anda dapat mengatur objek acara di editor visual di CloudFront konsol dan tidak menyimpannya.

Anda dapat menggunakan objek acara ini untuk menguji fungsi dari CloudFront konsol, meskipun tidak disimpan.

Opsi 2: Buat objek acara di editor visual

Anda dapat mengatur objek acara di editor visual di CloudFront konsol dan tidak menyimpannya. Anda dapat membuat 10 objek acara untuk setiap fungsi sehingga Anda dapat, misalnya, menguji berbagai kemungkinan input.

Saat Anda membuat objek acara dengan cara ini, Anda dapat menggunakan objek acara untuk menguji fungsi di CloudFront konsol. Anda tidak dapat menggunakannya untuk menguji fungsi menggunakan AWS API atau SDK.

Opsi 3: Buat objek acara menggunakan editor teks

Anda dapat menggunakan editor teks untuk membuat objek acara dalam format JSON. Untuk informasi tentang struktur objek peristiwa, lihat [Struktur peristiwa](#).

Anda dapat menggunakan objek acara ini untuk menguji fungsi menggunakan CLI. Tetapi Anda tidak dapat menggunakannya untuk menguji fungsi di CloudFront konsol.

Untuk membuat objek acara (opsi 1 atau 2)

1. Masuk ke CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home#/functions> dan pilih halaman Fungsi.

Pilih fungsi yang ingin Anda uji.

2. Pada halaman detail fungsi, pilih tab Uji.
3. Untuk jenis Acara, pilih salah satu opsi berikut:
 - Pilih permintaan Viewer jika fungsi memodifikasi permintaan HTTP atau menghasilkan respons berdasarkan permintaan. Bagian Permintaan muncul.
 - Pilih Respons penampil. Bagian Permintaan dan Respons muncul.
4. Lengkapi bidang untuk disertakan dalam acara tersebut. Anda dapat memilih Edit JSON untuk melihat JSON mentah.
5. (Opsional) Untuk menyimpan acara, pilih Simpan dan di acara uji Simpan, masukkan nama lalu pilih Simpan.

Anda juga dapat memilih Edit JSON dan menyalin JSON mentah, dan menyimpannya di file Anda sendiri, di luar. CloudFront

Untuk membuat objek acara (opsi 3)

Buat objek acara menggunakan editor teks. Simpan file di direktori tempat komputer Anda dapat terhubung.

Verifikasi bahwa Anda mengikuti panduan ini:

- Hilang `distributionDomainName`, `distributionId`, dan `requestId` bidang.
- Nama header, cookie, dan string kueri harus huruf kecil.

Salah satu opsi untuk membuat objek acara dengan cara ini adalah membuat sampel menggunakan editor visual. Anda dapat yakin bahwa sampel diformat dengan benar. Anda kemudian dapat menyalin JSON mentah dan menempelkannya ke editor teks dan menyimpan file.

Untuk informasi lebih lanjut tentang struktur suatu peristiwa, lihat [Struktur peristiwa](#).

Uji fungsi

Anda dapat menguji fungsi di CloudFront konsol atau dengan AWS Command Line Interface (AWS CLI).

Console

Untuk menguji fungsi

1. Masuk ke CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home#/functions> dan pilih halaman Fungsi.
2. Pilih fungsi yang ingin Anda uji.
3. Pilih tab Uji.
4. Pastikan acara yang benar ditampilkan. Untuk beralih dari acara yang ditampilkan saat ini, pilih acara lain di bidang Pilih acara uji.
5. Pilih fungsi Uji. Konsol menunjukkan output fungsi, termasuk log fungsi dan pemanfaatan komputasi.

CLI

Anda dapat menguji suatu fungsi dengan menggunakan `aws cloudfront test-function` perintah.

Untuk menguji fungsi

1. Buka jendela baris perintah.
2. Jalankan perintah berikut dari direktori yang sama yang berisi file yang ditentukan.

Contoh ini menggunakan `fileb://` notasi untuk meneruskan dalam file objek acara. Ini juga termasuk jeda baris untuk membuat perintah lebih mudah dibaca.

```
aws cloudfront test-function \  
  --name MaxAge \  
  --if-match ETVABCEXAMPLE \  
  --event-object fileb://event-maxage-test01.json \  
  --stage DEVELOPMENT
```

Catatan

- Anda mereferensikan fungsi dengan namanya dan ETag (dalam `if-match` parameter). Anda mereferensikan objek acara berdasarkan lokasinya di sistem file Anda.
- Panggung bisa `DEVELOPMENT` atau `LIVE`.

Ketika perintah berhasil, Anda melihat output seperti berikut ini.

```
TestResult:
  ComputeUtilization: '21'
  FunctionErrorMessage: ''
  FunctionExecutionLogs: []
  FunctionOutput: '{"response":{"headers":{"cloudfront-functions":
{"value":"generated-by-CloudFront-Functions"},"location":{"value":"https://
aws.amazon.com/cloudfront/"}},"statusDescription":"Found","cookies":
{},"statusCode":302}}'
  FunctionSummary:
    FunctionConfig:
      Comment: MaxAge function
      Runtime: cloudfront-js-2.0
      KeyValueStoreAssociations= \
        {Quantity=1, \
        Items=[{KeyValueStoreARN='arn:aws:cloudfront::111122223333:key-value-
store/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111'}]} \
    FunctionMetadata:
      CreatedTime: '2021-04-18T20:38:56.915000+00:00'
      FunctionARN: arn:aws:cloudfront::111122223333:function/MaxAge
      LastModifiedTime: '2023-17-20T10:38:57.057000+00:00'
      Stage: DEVELOPMENT
      Name: MaxAge
      Status: UNPUBLISHED
```

Catatan

- `FunctionExecutionLogs` berisi daftar baris log yang ditulis fungsi dalam pernyataan `console.log()` (jika ada).

- `ComputeUtilization` berisi informasi tentang menjalankan fungsi Anda. Lihat [the section called “Memahami pemanfaatan komputasi”](#).
- `FunctionOutput` berisi objek peristiwa yang dikembalikan fungsi.

Memahami pemanfaatan komputasi

Penggunaan komputasi adalah jumlah waktu yang dibutuhkan fungsi untuk menjalankan sebagai persentase dari waktu maksimum yang diizinkan. Misalnya, nilai 35 berarti fungsi selesai pada 35% dari waktu maksimum yang diizinkan.

Jika suatu fungsi terus menerus melebihi waktu maksimum yang diizinkan, CloudFront membatasi fungsi tersebut. Daftar berikut menjelaskan kemungkinan fungsi terhambat berdasarkan nilai pemanfaatan komputasi.

Nilai pemanfaatan komputasi:

- 1 — 50 — Fungsinya nyaman di bawah waktu maksimum yang diizinkan dan harus berjalan tanpa pelambatan.
- 51 — 70 — Fungsi mendekati waktu maksimum yang diizinkan. Pertimbangkan untuk mengoptimalkan kode fungsi.
- 71 — 100 — Fungsi ini sangat dekat dengan atau melebihi waktu maksimum yang diizinkan. CloudFront kemungkinan akan membatasi fungsi ini jika Anda mengaitkannya dengan distribusi.

Perbarui fungsi

Anda dapat memperbarui fungsi kapan saja. Perubahan dilakukan hanya pada versi fungsi yang ada di DEVELOPMENT panggung. Untuk menyalin pembaruan dari DEVELOPMENT panggung ke LIVE, Anda harus [mempublikasikan fungsinya](#).

Anda dapat memperbarui kode fungsi di CloudFront konsol atau dengan AWS Command Line Interface (AWS CLI).

Console

Untuk memperbarui kode fungsi

1. Masuk ke CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home#/functions> dan pilih halaman Fungsi.

Pilih fungsi yang akan diperbarui.

2. Pilih Edit dan buat perubahan berikut:

- Perbarui bidang apa pun di bagian Detail.
- Ubah atau hapus penyimpanan nilai kunci terkait. Untuk informasi selengkapnya tentang penyimpanan nilai utama, lihat [the section called “ CloudFront KeyValueStore”](#).
- Ubah kode fungsi. Pilih tab Build, buat perubahan, lalu pilih Simpan perubahan untuk menyimpan perubahan pada kode.

CLI

Untuk memperbarui kode fungsi

1. Buka jendela baris perintah.
2. Jalankan perintah berikut.

Contoh ini menggunakan `fileb://` notasi untuk meneruskan file. Ini juga termasuk jeda baris untuk membuat perintah lebih mudah dibaca.

```
aws cloudfront update-function \  
  --name MaxAge \  
  --function-config '{"Comment":"Max Age 2 years","Runtime":"cloudfront-  
js-2.0","KeyValueStoreAssociations":{"Quantity":1,"Items":  
[{"KeyValueStoreARN":"arn:aws:cloudfront::111122223333:key-value-store/  
a1b2c3d4-5678-90ab-cdef-EXAMPLE1111"}]}}' \  
  --function-code fileb://function-max-age-v1.js \  
  --if-match ETVABCEXAMPLE
```

Catatan

- Anda dapat mengidentifikasi fungsi dengan namanya dan ETag (dalam `if-match` parameter). Pastikan Anda menggunakan arus ETag. Anda bisa mendapatkan nilai ini dari operasi [DescribeFunction](#) API.

- Anda harus menyertakan `function-code`, bahkan jika Anda tidak ingin mengubahnya.
- Hati-hati dengan `function-config`. Anda harus melewati semua yang ingin Anda simpan dalam konfigurasi. Secara khusus, tangani penyimpanan nilai kunci sebagai berikut:
 - Untuk mempertahankan asosiasi penyimpanan nilai kunci yang ada (jika ada), tentukan nama toko yang ada.
 - Untuk mengubah asosiasi, tentukan nama penyimpanan nilai kunci baru.
 - Untuk menghapus asosiasi, hilangkan `KeyValueStoreAssociations` parameter.

Ketika perintah berhasil, Anda melihat output seperti berikut ini.

```
ETag: ETVXYZEXAMPLE
FunctionSummary:
  FunctionConfig:
    Comment: Max Age 2 years \
    Runtime: cloudfront-js-2.0 \
    KeyValueStoreAssociations= \
      {Quantity=1, \
      Items=[{KeyValueStoreARN='arn:aws:cloudfront::111122223333:key-value-
store/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111'}]} \
    FunctionMetadata: \
      CreatedTime: '2021-04-18T20:38:56.915000+00:00' \
      FunctionARN: arn:aws:cloudfront::111122223333:function/MaxAge \
      LastModifiedTime: '2023-12-19T23:41:15.389000+00:00' \
      Stage: DEVELOPMENT \
    Name: MaxAge \
    Status: UNPUBLISHED
```

Sebagian besar informasi diulang dari permintaan. Informasi lain ditambahkan oleh CloudFront.

Catatan

- **ETag**— Nilai ini berubah setiap kali Anda memodifikasi penyimpanan nilai kunci.
- **FunctionARN**— ARN untuk fungsi Anda CloudFront .

- Stage— Tahap untuk fungsi (LIVE atau DEVELOPMENT).
- Status— Status fungsi (PUBLISHED atau UNPUBLISHED).

Publikasikan fungsi

Saat Anda mempublikasikan fungsi Anda, ini menyalin fungsi dari DEVELOPMENT panggung ke LIVE panggung.

Jika perilaku cache tidak terkait dengan fungsi, mempublikasikannya memungkinkan Anda mengaitkannya dengan perilaku cache. Anda hanya dapat mengaitkan perilaku cache dengan fungsi yang ada di tahap LIVE.

Important

- Sebelum Anda mempublikasikan, kami sarankan Anda [menguji fungsinya](#).
- Setelah Anda mempublikasikan fungsi, semua perilaku cache yang terkait dengan fungsi tersebut secara otomatis mulai menggunakan salinan yang baru diterbitkan, segera setelah distribusi selesai digunakan.

Anda dapat mempublikasikan fungsi di CloudFront konsol atau dengan AWS CLI.

Console

Untuk mempublikasikan fungsi

1. Masuk ke CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home#/functions> dan pilih halaman Fungsi.
2. Pilih fungsi yang akan diperbarui.
3. Pilih tab Publish dan kemudian pilih Publish. Jika fungsi Anda sudah dilampirkan ke satu atau beberapa perilaku cache, pilih Publikasikan dan perbarui.
4. (Opsional) Untuk melihat distribusi yang terkait dengan fungsi, pilih CloudFront Distribusi terkait untuk memperluas bagian itu.

Ketika berhasil, spanduk muncul di bagian atas halaman yang mengatakan berhasil **Function name** diterbitkan. Anda juga dapat memilih tab Bangun, lalu pilih Live untuk melihat versi live kode fungsi.

CLI

Untuk mempublikasikan fungsi

1. Buka jendela baris perintah.
2. Jalankan perintah `aws cloudfront publish-function` berikut. Dalam contoh, jeda baris disediakan untuk membuat contoh lebih mudah dibaca.

```
aws cloudfront publish-function \  
  --name MaxAge \  
  --if-match ETVXYZEXAMPLE
```

Ketika perintah berhasil, Anda melihat output seperti berikut ini.

```
FunctionSummary:  
  FunctionConfig:  
    Comment: Max Age 2 years  
    Runtime: cloudfront-js-2.0  
  FunctionMetadata:  
    CreatedTime: '2021-04-18T21:24:21.314000+00:00'  
    FunctionARN: arn:aws:cloudfront::111122223333:function/ExampleFunction  
    LastModifiedTime: '2023-12-19T23:41:15.389000+00:00'  
    Stage: LIVE  
  Name: MaxAge  
  Status: UNASSOCIATED
```

Mengaitkan fungsi dengan distribusi

Untuk menggunakan fungsi dengan distribusi, Anda mengaitkan fungsi dengan satu atau beberapa perilaku cache dalam distribusi. Anda dapat mengaitkan fungsi dengan beberapa perilaku cache di beberapa distribusi.

Anda dapat mengaitkan fungsi dengan salah satu dari berikut ini:

- Perilaku cache yang ada
- Perilaku cache baru dalam distribusi yang ada

- Perilaku cache baru dalam distribusi baru

Ketika mengaitkan fungsi dengan perilaku cache, Anda harus memilih Jenis peristiwa. Jenis acara menentukan kapan CloudFront menjalankan fungsi.

Anda dapat memilih jenis acara berikut:

- Permintaan penampil - Fungsi berjalan saat CloudFront menerima permintaan dari penampil.
- Respons penampil - Fungsi berjalan sebelum CloudFront mengembalikan respons ke penampil.

Anda tidak dapat menggunakan tipe peristiwa yang menghadap asal (permintaan asal dan respons asal) dengan CloudFront Functions. Sebagai gantinya, Anda dapat menggunakan Lambda @Edge. Untuk informasi selengkapnya, lihat [CloudFront peristiwa yang dapat memicu fungsi Lambda @Edge](#).

Note

Sebelum mengaitkan fungsi, [Publikasikan](#) ke tahap LIVE.

Anda dapat mengaitkan fungsi dengan distribusi di CloudFront konsol atau dengan AWS Command Line Interface (AWS CLI). Prosedur berikut menunjukkan cara mengaitkan fungsi dengan perilaku cache yang ada.

Console

Untuk mengaitkan fungsi dengan perilaku cache yang ada

1. Masuk ke CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home#/functions> dan pilih halaman Fungsi.
2. Pilih fungsi yang ingin Anda kaitkan.
3. Pada halaman Fungsi, pilih tab Publikasikan.
4. Pilih fungsi Publikasikan.
5. Pilih Tambah asosiasi. Pada kotak dialog yang muncul, pilih distribusi, jenis peristiwa, dan/atau perilaku cache.

Untuk jenis acara, pilih kapan Anda ingin fungsi ini berjalan:

- Permintaan Penampil - Jalankan fungsi setiap kali CloudFront menerima permintaan.

- Respons Penampil - Jalankan fungsi setiap kali CloudFront mengembalikan respons.
6. Untuk menyimpan konfigurasi, pilih Tambahkan asosiasi.

CloudFront mengaitkan distribusi dengan fungsi. Tunggu beberapa menit sampai distribusi terkait dapat menyelesaikan penyebaran. Anda dapat memilih Lihat distribusi pada halaman detail fungsi untuk memeriksa kemajuan.

CLI

Untuk mengaitkan fungsi dengan perilaku cache yang ada

1. Buka jendela baris perintah.
2. Masukkan perintah berikut untuk menyimpan konfigurasi distribusi untuk distribusi yang perilaku cache-nya ingin Anda kaitkan dengan suatu fungsi. Perintah ini menyimpan konfigurasi distribusi ke file bernama `dist-config.yaml`. Untuk menggunakan perintah ini, lakukan hal berikut:
 - Ganti *DistributionID* dengan ID distribusi.
 - Jalankan perintah pada satu garis. Dalam contoh, jeda baris disediakan untuk membuat contoh lebih mudah dibaca.

```
aws cloudfront get-distribution-config \  
  --id DistributionID \  
  --output yaml > dist-config.yaml
```

Ketika perintah berhasil, AWS CLI mengembalikan tidak ada output.

3. Buka file bernama `dist-config.yaml` yang Anda buat. Edit file untuk membuat perubahan berikut.
 - a. Ubah nama ETag bidang menjadi `IfMatch`, tetapi jangan ubah nilai bidang.
 - b. Dalam perilaku cache, temukan objek bernama `FunctionAssociations`. Memperbarui objek ini untuk menambahkan asosiasi fungsi. Sintaks YAML untuk asosiasi fungsi terlihat seperti contoh berikut.
 - Contoh berikut menunjukkan jenis peristiwa permintaan penampil (pemicu). Untuk menggunakan jenis peristiwa respons penampil, ganti `viewer-request` dengan `viewer-response`.

- Ganti `arn:aws:cloudfront::111122223333:function/ExampleFunction` dengan Amazon Resource Name (ARN) dari fungsi yang Anda kaitkan dengan perilaku cache ini. Untuk mendapatkan fungsi ARN, Anda dapat menggunakan perintah `aws cloudfront list-functions`.

```
FunctionAssociations:
  Items:
    - EventType: viewer-request
      FunctionARN: arn:aws:cloudfront::111122223333:function/ExampleFunction
  Quantity: 1
```

- Setelah melakukan perubahan ini, simpan file.
- Gunakan perintah berikut untuk memperbarui distribusi guna menambahkan asosiasi fungsi. Untuk menggunakan perintah ini, lakukan hal berikut:
 - Ganti `DistributionID` dengan ID distribusi.
 - Jalankan perintah pada satu garis. Dalam contoh, jeda baris disediakan untuk membuat contoh lebih mudah dibaca.

```
aws cloudfront update-distribution \
  --id DistributionID \
  --cli-input-yaml file://dist-config.yaml
```

Setelah perintah berhasil, Anda akan melihat output seperti berikut yang menjelaskan distribusi yang baru saja diperbarui dengan asosiasi fungsi. Contoh output berikut dipotong agar mudah dibaca.

```
Distribution:
  ARN: arn:aws:cloudfront::111122223333:distribution/EBEDLT3BGRBBW
  ... truncated ...
DistributionConfig:
  ... truncated ...
DefaultCacheBehavior:
  ... truncated ...
FunctionAssociations:
  Items:
    - EventType: viewer-request
      FunctionARN: arn:aws:cloudfront::111122223333:function/ExampleFunction
```

```
Quantity: 1
... truncated ...
DomainName: d1111111abcdef8.cloudfront.net
Id: EDFDVBD6EXAMPLE
LastModifiedTime: '2021-04-19T22:39:09.158000+00:00'
Status: InProgress
ETag: E2VJGGQEG1JT8S
```

Distribusi Status berubah menjadi `InProgress` saat distribusi dipindahkan. Ketika konfigurasi distribusi baru mencapai lokasi CloudFront tepi, lokasi tepi itu mulai menggunakan fungsi terkait. Ketika distribusi sepenuhnya digunakan, Status perubahan kembali ke `Deployed`. Ini menunjukkan bahwa CloudFront fungsi terkait hidup di semua lokasi CloudFront tepi di seluruh dunia. Ini biasanya memerlukan waktu beberapa menit.

Amazon CloudFront KeyValueCollection

CloudFront KeyValueCollection adalah datastore nilai kunci latensi rendah yang aman, global, yang memungkinkan akses baca dari dalam [CloudFront Fungsi](#), memungkinkan logika lanjutan yang dapat disesuaikan di lokasi tepi. CloudFront

Dengan CloudFront KeyValueCollection, Anda membuat pembaruan untuk kode fungsi dan pembaruan data yang terkait dengan fungsi secara independen satu sama lain. Pemisahan ini menyederhanakan kode fungsi dan membuatnya mudah untuk memperbarui data tanpa perlu menyebarkan perubahan kode.

Note

Untuk menggunakannya CloudFront KeyValueCollection, CloudFront fungsi Anda harus menggunakan [JavaScript runtime 2.0](#).

Berikut ini adalah prosedur umum untuk menggunakan pasangan kunci-nilai:

- Buat toko nilai kunci, dan isi dengan satu set pasangan kunci-nilai. Anda dapat menambahkan toko nilai kunci Anda ke bucket Amazon S3 atau memasukkannya secara manual.
- Kaitkan penyimpanan nilai kunci dengan CloudFront fungsi Anda.
- Dalam kode fungsi Anda, gunakan nama kunci untuk mengambil nilai yang terkait dengan kunci atau untuk mengevaluasi apakah ada kunci. Untuk informasi selengkapnya tentang penggunaan

pasangan kunci-nilai dalam kode fungsi, dan tentang metode pembantu, lihat. [the section called “Metode pembantu untuk penyimpanan nilai kunci”](#)

Kasus penggunaan

Anda dapat menggunakan pasangan kunci-nilai untuk contoh berikut:

- Penulisan ulang atau pengalihan URL — Pasangan kunci-nilai dapat menahan penulisan ulang atau pengalihan. URLs URLs
- Pengujian A/B dan bendera fitur — Anda dapat membuat fungsi untuk menjalankan eksperimen dengan menetapkan persentase lalu lintas ke versi tertentu dari situs web Anda.
- Otorisasi akses — Anda dapat menerapkan kontrol akses untuk mengizinkan atau menolak permintaan berdasarkan kriteria yang ditentukan oleh Anda dan data yang disimpan di penyimpanan nilai kunci.

Format yang didukung untuk nilai

Anda dapat menyimpan nilai dalam pasangan kunci-nilai dalam salah satu format berikut:

- String
- String yang dikodekan byte
- JSON

Keamanan

CloudFront Fungsi dan semua data penyimpanan nilai utamanya ditangani dengan aman, sebagai berikut:

- CloudFront mengenkripsi setiap penyimpanan nilai kunci saat istirahat dan selama transit (saat membaca atau menulis ke penyimpanan nilai kunci) saat Anda memanggil operasi [CloudFront KeyValueCollectionAPI](#).
- Saat fungsi dijalankan, CloudFront dekripsi setiap pasangan kunci-nilai dalam memori di lokasi tepi. CloudFront

Untuk memulai CloudFront KeyValueCollection, lihat topik berikut.

Topik

- [Bekerja dengan toko nilai kunci](#)
- [Bekerja dengan data nilai kunci](#)
- Untuk informasi lebih lanjut tentang memulai CloudFront KeyValueCollection, lihat posting CloudFront KeyValueCollection AWS blog [Memperkenalkan Amazon](#).

Bekerja dengan toko nilai kunci

Anda harus membuat penyimpanan nilai kunci untuk menahan pasangan kunci-nilai yang ingin Anda gunakan di CloudFront Functions.

Setelah Anda membuat penyimpanan nilai kunci dan menambahkan pasangan kunci-nilai, Anda dapat menggunakan nilai kunci dalam kode CloudFront fungsi Anda.

Untuk memulai, lihat topik berikut:

Topik

- [Buat penyimpanan nilai kunci](#)
- [Kaitkan penyimpanan nilai kunci dengan fungsi](#)
- [Perbarui penyimpanan nilai kunci](#)
- [Dapatkan referensi ke penyimpanan nilai kunci](#)
- [Hapus penyimpanan nilai kunci](#)
- [Format file untuk pasangan nilai kunci](#)

Note

JavaScript Runtime 2.0 menyertakan beberapa metode pembantu untuk bekerja dengan nilai-nilai kunci dalam kode fungsi. Untuk informasi selengkapnya, lihat [the section called “Metode pembantu untuk penyimpanan nilai kunci”](#).

Buat penyimpanan nilai kunci

Anda dapat membuat penyimpanan nilai kunci dan pasangan nilai kunci-nya secara bersamaan. Anda juga dapat membuat penyimpanan nilai kunci kosong sekarang dan kemudian menambahkan pasangan kunci-nilai nanti.

Note

Jika Anda menentukan sumber data dari bucket Amazon S3, Anda harus memiliki izin `s3:GetObject` dan `s3:GetBucketLocation` izin untuk bucket tersebut. Jika Anda tidak memiliki izin ini, tidak CloudFront dapat berhasil membuat penyimpanan nilai kunci Anda.

Putuskan apakah Anda ingin menambahkan pasangan kunci-nilai pada saat yang sama ketika Anda membuat penyimpanan nilai kunci. Anda dapat mengimpor pasangan nilai kunci dengan menggunakan CloudFront konsol, CloudFront API, atau AWS SDKs. Namun, Anda hanya dapat mengimpor file pasangan nilai kunci saat Anda awalnya membuat penyimpanan nilai kunci.

Untuk membuat file pasangan kunci-nilai, lihat [Format file untuk pasangan nilai kunci](#)

Console

Untuk membuat penyimpanan nilai kunci

1. Masuk ke Konsol Manajemen AWS dan buka halaman Fungsi di CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home#/functions>.
2. Pilih KeyValueStoretab, lalu pilih Buat KeyValueStore.
3. Masukkan nama dan deskripsi opsional untuk penyimpanan nilai kunci.
4. URI S3 lengkap:
 - Jika Anda memiliki file pasangan nilai kunci, masukkan path ke bucket Amazon S3 tempat Anda menyimpan file.
 - Biarkan bidang ini kosong jika Anda berencana memasukkan pasangan kunci-nilai secara manual.
5. Pilih Buat. Penyimpanan nilai kunci sekarang ada.

Halaman detail untuk penyimpanan nilai kunci baru muncul. Informasi pada halaman termasuk ID dan ARN dari penyimpanan nilai kunci.

- ID adalah string acak karakter yang unik di Akun AWS.

- ARN memiliki sintaks ini:

Akun AWS:key-value-store/the key value stores ID

6. Lihatlah bagian Pasangan nilai kunci. Jika Anda mengimpor file, bagian ini menunjukkan beberapa pasangan kunci-nilai. Anda dapat melakukan tindakan berikut:
 - Jika Anda mengimpor file, Anda juga dapat menambahkan lebih banyak nilai secara manual.
 - Jika Anda tidak mengimpor file dari bucket Amazon S3, dan jika Anda ingin menambahkan pasangan nilai kunci sekarang, Anda dapat menyelesaikan langkah berikutnya.
 - Anda dapat melewati langkah ini dan menambahkan pasangan kunci-nilai nanti.
7. Untuk menambahkan pasangan sekarang:
 - a. Pilih Tambahkan pasangan kunci-nilai.
 - b. Pilih Tambah pasangan dan masukkan nama dan nilai. Ulangi langkah ini untuk menambahkan lebih banyak pasangan.
 - c. Setelah selesai, pilih Simpan perubahan untuk menyimpan semua pasangan nilai kunci di penyimpanan nilai kunci. Pada kotak dialog yang muncul, pilih Selesai.
8. Untuk mengaitkan penyimpanan nilai kunci dengan fungsi sekarang, selesaikan bagian Fungsi terkait. Untuk informasi selengkapnya, lihat [????](#) atau [????](#).

Anda juga dapat mengaitkan fungsi nanti, baik dari halaman detail penyimpanan nilai kunci ini, atau dari halaman detail fungsi.

AWS CLI

Untuk membuat penyimpanan nilai kunci

- Jalankan perintah berikut untuk membuat penyimpanan nilai kunci dan mengimpor pasangan nilai kunci dari bucket Amazon S3.

```
aws cloudfront create-key-value-store \  
  --name=keyvaluestore1 \  
  --comment="This is my key value store file" \  
  --import-source=SourceType=S3,SourceARN=arn:aws:s3:::amzn-s3-demo-  
  bucket1/kvs-input.json
```

Respons

```
{
  "ETag": "ETVABCEXAMPLE",
  "Location": "https://cloudfront.amazonaws.com/2020-05-31/key-value-store/arn:aws:cloudfront::123456789012:key-value-store/8aa76c93-3198-462c-aaf6-example",
  "KeyValueStore": {
    "Name": "keyvaluestore1",
    "Id": "8aa76c93-3198-462c-aaf6-example",
    "Comment": "This is my key value store file",
    "ARN": "arn:aws:cloudfront::123456789012:key-value-store/8aa76c93-3198-462c-aaf6-example",
    "Status": "PROVISIONING",
    "LastModifiedTime": "2024-08-06T22:19:10.813000+00:00"
  }
}
```

API

Untuk membuat penyimpanan nilai kunci

1. Gunakan [CloudFrontCreateKeyValueStore](#) operasi. Operasi ini membutuhkan beberapa parameter:
 - A name dari penyimpan nilai kunci.
 - `commentParameter` yang menyertakan komentar.
 - `import-sourceParameter` yang memungkinkan Anda mengimpor pasangan nilai kunci dari file yang disimpan di bucket Amazon S3. Anda dapat mengimpor dari file hanya ketika Anda pertama kali membuat penyimpanan nilai kunci. Untuk informasi tentang struktur file, lihat [the section called “Format file untuk pasangan nilai kunci”](#).

Respons operasi mencakup informasi berikut:

- Nilai yang diteruskan dalam permintaan, termasuk nama yang Anda tetapkan.
- Data seperti waktu pembuatan.

- Sebuah ETag (misalnya,ETVABCEXAMPLE), ARN yang menyertakan nama penyimpanan nilai kunci (misalnya,arn:aws:cloudfront::123456789012:key-value-store/keyvaluestore1).

Anda akan menggunakan beberapa kombinasi dariETag, ARN, dan nama untuk bekerja dengan penyimpanan nilai kunci secara terprogram.

Status penyimpanan nilai kunci

Saat Anda membuat penyimpanan nilai kunci, penyimpanan data dapat memiliki nilai status berikut.

Nilai	Deskripsi
Penyediaan	Penyimpanan nilai kunci telah dibuat dan CloudFront sedang memproses sumber data yang Anda tentukan.
Siap	Penyimpanan nilai kunci dibuat dan CloudFront berhasil memproses sumber data yang Anda tentukan.
Impor gagal	CloudFront tidak dapat memproses sumber data yang Anda tentukan. Status ini dapat muncul jika format file Anda tidak valid atau melebihi batas ukuran. Untuk informasi selengkapnya, lihat Format file untuk pasangan nilai kunci .

Kaitkan penyimpanan nilai kunci dengan fungsi

Setelah Anda membuat penyimpanan nilai kunci Anda, Anda dapat memperbarui fungsi Anda untuk mengaitkannya dengan penyimpanan nilai kunci Anda. Anda harus membuat asosiasi ini untuk menggunakan pasangan kunci-nilai dari toko itu dalam fungsi itu. Aturan-aturan berikut berlaku:

- Sebuah fungsi hanya dapat memiliki satu penyimpanan nilai kunci
- Anda dapat mengaitkan penyimpanan nilai kunci yang sama dengan beberapa fungsi

Console

Untuk mengaitkan penyimpanan nilai kunci dengan fungsi

1. Masuk ke CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home#/functions> dan pilih halaman Fungsi.

2. Pilih nama fungsi.
3. Buka KeyValueStore bagian Associate dan pilih Associate existing KeyValueStore.
4. Pilih penyimpanan nilai kunci yang berisi pasangan kunci-nilai dalam fungsi, lalu pilih Associate. KeyValueStore

CloudFront segera mengaitkan toko dengan fungsinya. Anda tidak perlu menyimpan fungsinya.

5. Untuk menentukan penyimpanan nilai kunci yang berbeda, pilih Perbarui terkait KeyValueStore, pilih nama penyimpanan nilai kunci lainnya, lalu pilih Associate KeyValueStore.

Untuk informasi selengkapnya, lihat [the section called “Perbarui fungsi”](#).

AWS CLI

Untuk mengaitkan penyimpanan nilai kunci dengan fungsi

- Jalankan perintah berikut untuk memperbarui *MaxAge* fungsi dan mengaitkan sumber daya penyimpanan nilai kunci.

```
aws cloudfront update-function \  
  --name MaxAge \  
  --function-config '{"Comment":"Max Age 2 years","Runtime":"cloudfront-  
js-2.0","KeyValueStoreAssociations":{"Quantity":1,"Items":  
[{"KeyValueStoreARN":"arn:aws:cloudfront::123456789012:key-value-  
store/8aa76c93-3198-462c-aaf6-example"]}]}\' \  
  --function-code fileb://function-max-age-v1.js \  
  --if-match ETVABCEXAMPLE
```

- Untuk mengaitkan penyimpanan nilai kunci dengan fungsi, tentukan KeyValueStoreAssociations parameter dan nilai kunci menyimpan ARN.
- Untuk mengubah asosiasi, tentukan ARN penyimpanan nilai kunci lainnya.
- Untuk menghapus asosiasi, hapus KeyValueStoreAssociations parameter.

Untuk informasi selengkapnya, lihat [the section called “Perbarui fungsi”](#).

API

Untuk mengaitkan penyimpanan nilai kunci dengan fungsi

- Gunakan Operasi API [UpdateFunction](#). Untuk informasi selengkapnya, lihat [the section called “Perbarui fungsi”](#).

Catatan

- Jika Anda memodifikasi penyimpanan nilai kunci tanpa mengubah pasangan nilai kunci, atau jika Anda hanya memodifikasi pasangan nilai kunci di penyimpanan nilai kunci, Anda tidak perlu mengaitkan penyimpanan nilai kunci lagi. Anda juga tidak perlu mempublikasikan ulang fungsinya.

Namun, kami menyarankan Anda menguji fungsi untuk memverifikasi bahwa itu berfungsi seperti yang diharapkan. Untuk informasi selengkapnya, lihat [Fungsi uji](#).

- Anda dapat melihat semua fungsi yang menggunakan penyimpanan nilai kunci tertentu. Di CloudFront konsol, pilih halaman detail penyimpanan nilai kunci.

Perbarui penyimpanan nilai kunci

Saat memperbarui penyimpanan nilai kunci, Anda dapat mengubah pasangan nilai kunci, atau mengubah asosiasi antara penyimpanan nilai kunci dan fungsi.

Console

Untuk memperbarui penyimpanan nilai kunci

1. Masuk ke Konsol Manajemen AWS dan buka halaman Fungsi di CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home#/functions>.
2. Pilih KeyValueStorestab.
3. Pilih penyimpanan nilai kunci yang ingin Anda perbarui.
 - Untuk memperbarui pasangan kunci-nilai, pilih Edit di bagian Pasangan nilai kunci. Anda dapat menambahkan atau menghapus pasangan kunci-nilai apa pun. Anda juga dapat mengubah nilai untuk pasangan kunci-nilai yang ada. Setelah Anda selesai, pilih Simpan perubahan.

- Untuk memperbarui asosiasi untuk penyimpanan nilai kunci ini, pilih Buka fungsi. Untuk informasi selengkapnya, lihat [the section called “Kaitkan penyimpanan nilai kunci dengan fungsi”](#).

AWS CLI

Untuk memperbarui penyimpanan nilai kunci

1. Mengubah pasangan kunci-nilai — Anda dapat menambahkan lebih banyak pasangan kunci-nilai, menghapus satu atau beberapa pasangan kunci-nilai, dan mengubah nilai pasangan kunci-nilai yang ada. Untuk informasi selengkapnya, lihat [Bekerja dengan data nilai kunci](#).
2. Mengubah asosiasi fungsi untuk penyimpanan nilai kunci — Untuk memperbarui fungsi asosiasi untuk penyimpanan nilai kunci, lihat [Kaitkan penyimpanan nilai kunci dengan fungsi](#).

Tip

Anda akan membutuhkan ARN dari toko nilai kunci. Untuk informasi selengkapnya, lihat [the section called “Dapatkan referensi ke penyimpanan nilai kunci”](#).

API

Untuk memperbarui penyimpanan nilai kunci

1. Mengubah pasangan kunci-nilai — Anda dapat menambahkan lebih banyak pasangan kunci-nilai, menghapus satu atau beberapa pasangan kunci-nilai, dan mengubah nilai pasangan kunci-nilai yang ada. Untuk informasi selengkapnya, lihat [Bekerja dengan data nilai kunci](#).
2. Mengubah asosiasi fungsi untuk penyimpanan nilai kunci — Untuk memperbarui asosiasi fungsi untuk penyimpanan nilai kunci, gunakan operasi [UpdateFunction](#) API. Untuk informasi selengkapnya, lihat [the section called “Perbarui fungsi”](#).

Tip

Anda akan membutuhkan ARN dari toko nilai kunci. Untuk informasi selengkapnya, lihat [the section called “Dapatkan referensi ke penyimpanan nilai kunci”](#).

Dapatkan referensi ke penyimpanan nilai kunci

Untuk bekerja dengan toko nilai kunci secara terprogram, Anda memerlukan ETag dan nama penyimpanan nilai kunci.

Untuk mendapatkan kedua nilai tersebut, Anda dapat menggunakan AWS Command Line Interface (AWS CLI) atau CloudFront API.

AWS CLI

Untuk mendapatkan referensi penyimpanan nilai kunci

1. Untuk mengembalikan daftar penyimpanan nilai kunci, jalankan perintah berikut. Temukan nama penyimpanan nilai kunci yang ingin Anda ubah.

```
aws cloudfront list-key-value-stores
```

2. Dari respons, temukan nama penyimpanan nilai kunci yang Anda inginkan.

Respons

```
{
  "KeyValueStoreList": {
    "Items": [
      {
        "Name": "keyvaluestore3",
        "Id": "37435e19-c205-4271-9e5c-example3",
        "ARN": "arn:aws:cloudfront::123456789012:key-value-store/37435e19-c205-4271-9e5c-example3",
        "Status": "READY",
        "LastModifiedTime": "2024-05-08T14:50:18.876000+00:00"
      },
      {
        "Name": "keyvaluestore2",
        "Id": "47970d59-6408-474d-b850-example2",
        "ARN": "arn:aws:cloudfront::123456789012:key-value-store/47970d59-6408-474d-b850-example2",
        "Status": "READY",
        "LastModifiedTime": "2024-05-30T21:06:22.113000+00:00"
      },
      {
        "Name": "keyvaluestore1",
        "Id": "8aa76c93-3198-462c-aaf6-example",

```

```

        "ARN": "arn:aws:cloudfront::123456789012:key-value-
store/8aa76c93-3198-462c-aaf6-example",
        "Status": "READY",
        "LastModifiedTime": "2024-08-06T22:19:30.510000+00:00"
    }
]
}
}

```

3. Jalankan perintah berikut untuk mengembalikan ETag untuk penyimpanan nilai kunci yang ditentukan.

```

aws cloudfront describe-key-value-store \
  --name=keyvaluestore1

```

Respons

```

{
  "ETag": "E3UN6WX5RR02AG",
  "KeyValueStore": {
    "Name": "keyvaluestore1",
    "Id": "8aa76c93-3198-462c-aaf6-example",
    "Comment": "This is an example KVS",
    "ARN": "arn:aws:cloudfront::123456789012:key-value-
store/8aa76c93-3198-462c-aaf6-example",
    "Status": "READY",
    "LastModifiedTime": "2024-08-06T22:19:30.510000+00:00"
  }
}

```

API

Untuk mendapatkan referensi penyimpanan nilai kunci

1. Gunakan operasi [CloudFront ListKeyValueStores](#) API untuk mengembalikan daftar penyimpanan nilai kunci. Temukan nama penyimpanan nilai kunci yang ingin Anda ubah.
2. Gunakan operasi [CloudFront DescribeKeyValueStore](#) API dan tentukan nama penyimpanan nilai kunci yang Anda kembalikan dari langkah sebelumnya.

Respons termasuk UUID, ARN dari penyimpanan nilai kunci, dan penyimpanan nilai kunci. ETag

- AnETag, seperti E3UN6WX5RR02AG
- UUID adalah 128 bit, seperti 8aa76c93-3198-462c-aaf6-example
- ARN mencakup Akun AWS angka, konstantakey-value-store, dan UUID, seperti contoh berikut:

```
arn:aws:cloudfront::123456789012:key-value-store/8aa76c93-3198-462c-aaf6-example
```

Untuk informasi lebih lanjut tentang DescribeKeyValueStore operasi, lihat [the section called “Tentang CloudFront KeyValueStore”](#).

Hapus penyimpanan nilai kunci

Anda dapat menghapus penyimpanan nilai kunci Anda dengan menggunakan CloudFront konsol Amazon atau API.

Console

Untuk menghapus penyimpanan nilai kunci

1. Masuk ke Konsol Manajemen AWS dan buka halaman Fungsi di CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home#/functions>.
2. Pilih nama fungsi.
3. Di bawah KeyValueStore bagian Terkait, verifikasi apakah penyimpanan nilai kunci dikaitkan dengan fungsi tersebut. Jika ya, hapus asosiasi dengan memilih Disassociate KeyValueStore dan kemudian pilih Hapus asosiasi.
4. Di panel navigasi, pilih halaman Fungsi dan kemudian pilih KeyValueStorestab.
5. Pilih penyimpanan nilai kunci yang ingin Anda hapus dan kemudian pilih Hapus.

AWS CLI

Untuk menghapus penyimpanan nilai kunci

1. Dapatkan ETag dan nama toko nilai kunci. Untuk informasi selengkapnya, lihat [the section called “Dapatkan referensi ke penyimpanan nilai kunci”](#).
2. Verifikasi apakah penyimpanan nilai kunci dikaitkan dengan fungsi. Jika ya, hapus asosiasi. Untuk informasi lebih lanjut tentang kedua langkah ini, lihat [???](#).

3. Setelah Anda memiliki nama dan ETag penyimpanan nilai kunci dan tidak lagi terkait dengan fungsi, Anda dapat menghapusnya.

Jalankan perintah berikut untuk menghapus penyimpanan nilai kunci yang ditentukan.

```
aws cloudfront delete-key-value-store \  
  --name=keyvaluestore1 \  
  --if-match=E3UN6WX5RR02AG
```

API

Untuk menghapus penyimpanan nilai kunci

1. Dapatkan ETag dan nama toko nilai kunci. Untuk informasi selengkapnya, lihat [the section called “Dapatkan referensi ke penyimpanan nilai kunci”](#).
2. Verifikasi apakah penyimpanan nilai kunci dikaitkan dengan fungsi. Jika ya, hapus asosiasi. Untuk informasi lebih lanjut tentang kedua langkah ini, lihat [???](#).
3. Untuk menghapus penyimpanan nilai kunci, gunakan operasi CloudFront [DeleteKeyValueStoreAPI](#).

Format file untuk pasangan nilai kunci

Saat Anda membuat file yang dikodekan UTF-8, gunakan format JSON berikut:

```
{  
  "data": [  
    {  
      "key": "key1",  
      "value": "value"  
    },  
    {  
      "key": "key2",  
      "value": "value"  
    }  
  ]  
}
```

File Anda tidak dapat menyertakan kunci duplikat. Jika Anda menentukan file yang tidak valid di bucket Amazon S3, Anda dapat memperbarui file tersebut untuk menghapus duplikat apa pun dan kemudian mencoba membuat penyimpanan nilai kunci Anda lagi.

Untuk informasi selengkapnya, lihat [Buat penyimpanan nilai kunci](#).

Note

File untuk sumber data Anda dan pasangan nilai kunci-nya memiliki batasan berikut:

- Ukuran file — 5 MB
- Ukuran kunci - 512 karakter
- Ukuran nilai - 1024 karakter

Bekerja dengan data nilai kunci

Topik ini menjelaskan cara menambahkan pasangan kunci-nilai ke penyimpanan nilai kunci yang ada. Untuk menyertakan pasangan kunci-nilai saat Anda awalnya membuat penyimpanan nilai kunci, lihat [the section called “Buat penyimpanan nilai kunci”](#)

Topik

- [Bekerja dengan pasangan nilai kunci \(konsol\)](#)
- [Tentang CloudFront KeyValueCollection](#)
- [Bekerja dengan pasangan nilai kunci \(\)AWS CLI](#)
- [Bekerja dengan pasangan nilai kunci \(API\)](#)

Bekerja dengan pasangan nilai kunci (konsol)

Anda dapat menggunakan CloudFront konsol untuk bekerja dengan pasangan nilai kunci Anda.

Untuk bekerja dengan pasangan kunci-nilai

1. Masuk ke Konsol Manajemen AWS dan buka halaman Fungsi di CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home#/functions>.
2. Pilih KeyValueCollection.
3. Pilih penyimpanan nilai kunci yang ingin Anda ubah.
4. Di bagian Pasangan nilai kunci, pilih Edit.

5. Anda dapat menambahkan pasangan kunci-nilai, menghapus pasangan kunci-nilai, atau mengubah nilai untuk pasangan kunci-nilai yang ada.
6. Setelah Anda selesai, pilih Simpan perubahan.

Tentang CloudFront KeyValueCollection

Tip

CloudFront KeyValueCollection API adalah layanan global yang menggunakan Signature Version 4A (Sigv4a) untuk otentikasi. Menggunakan kredensi sementara dengan Sigv4a memerlukan token sesi versi 2. Untuk informasi selengkapnya, lihat [Menggunakan kredensial sementara dengan API CloudFront KeyValueCollection](#).

Jika Anda menggunakan AWS Command Line Interface (AWS CLI) atau kode Anda sendiri untuk memanggil CloudFront KeyValueCollection API, lihat bagian berikut.

Saat Anda bekerja dengan penyimpanan nilai kunci dan pasangan nilai kunci-nya, layanan yang Anda panggil bergantung pada kasus penggunaan Anda:

- Untuk bekerja dengan pasangan nilai kunci di penyimpanan nilai kunci yang ada, gunakan layanan. CloudFront KeyValueCollection
- Untuk menyertakan pasangan nilai kunci di penyimpanan nilai kunci saat Anda pertama kali membuat penyimpanan nilai kunci, gunakan layanan. CloudFront

Baik CloudFront API dan CloudFront KeyValueCollection API memiliki DescribeKeyValueCollection operasi. Anda memanggil mereka untuk alasan yang berbeda. Untuk memahami perbedaannya, lihat tabel berikut.

	API CloudFront DescribeKeyValueCollection	API CloudFront KeyValueCollection DescribeKeyValueCollection
Data tentang penyimpanan nilai kunci	Mengembalikan data, seperti status dan tanggal penyimpanan nilai kunci itu sendiri terakhir diubah.	Mengembalikan data tentang isi sumber daya penyimpanan —

	API CloudFront DescribeKeyValueStore	API CloudFront KeyValueStore DescribeKey ValueStore
		pasangan kunci-nilai di toko, dan ukuran konten.
Data yang mengidentifikasi penyimpanan an nilai kunci	MengembalikanETag, UUID, dan ARN dari penyimpanan nilai kunci.	Mengembalikan ETag dan ARN dari penyimpan an nilai kunci.

Catatan

- Setiap DescribeKeyValueStore operasi mengembalikan yang berbedaETag. Itu ETags tidak bisa dipertukarkan.
- Saat Anda memanggil operasi API untuk menyelesaikan suatu tindakan, Anda harus menentukan ETag dari API yang sesuai. Misalnya, dalam CloudFront KeyValueStore [DeleteKey](#) operasi, Anda menentukan ETag yang Anda kembalikan dari CloudFront KeyValueStore [DescribeKeyValueStore](#) operasi.
- Saat Anda memanggil CloudFront Fungsi dengan menggunakan CloudFront KeyValueStore, nilai di penyimpanan nilai kunci tidak diperbarui atau diubah selama pemanggilan fungsi. Pembaruan diproses di antara pemanggilan fungsi.

Bekerja dengan pasangan nilai kunci ()AWS CLI

Anda dapat menjalankan AWS Command Line Interface perintah berikut untuk CloudFront KeyValueStore.

Daftar Isi

- [Daftar pasangan kunci-nilai](#)
- [Dapatkan pasangan kunci-nilai](#)
- [Jelaskan penyimpanan nilai kunci](#)
- [Buat pasangan kunci-nilai](#)
- [Hapus pasangan kunci-nilai](#)

- [Perbarui pasangan nilai kunci](#)

Daftar pasangan kunci-nilai

Untuk membuat daftar pasangan kunci-nilai di penyimpanan nilai kunci Anda, jalankan perintah berikut.

```
aws cloudfront-keyvaluestore list-keys \
  --kvs-arn=arn:aws:cloudfront::123456789012:key-value-store/37435e19-c205-4271-9e5c-
  example
```

Respons

```
{
  "Items": [
    {
      "Key": "key1",
      "Value": "value1"
    }
  ]
}
```

Dapatkan pasangan kunci-nilai

Untuk mendapatkan pasangan kunci-nilai di penyimpanan nilai kunci Anda, jalankan perintah berikut.

```
aws cloudfront-keyvaluestore get-key \
  --key=key1 \
  --kvs-arn=arn:aws:cloudfront::123456789012:key-value-store/37435e19-c205-4271-9e5c-
  example
```

Respons

```
{
  "Key": "key1",
  "Value": "value1",
  "ItemCount": 1,
  "TotalSizeInBytes": 11
}
```

Jelaskan penyimpanan nilai kunci

Untuk mendeskripsikan penyimpanan nilai kunci, jalankan perintah berikut.

```
aws cloudfront-keyvaluestore describe-key-value-store \
  --kvs-arn=arn:aws:cloudfront::123456789012:key-value-store/37435e19-c205-4271-9e5c-
  example
```

Respons

```
{
  "ETag": "KV1F83G8C2AR07P",
  "ItemCount": 1,
  "TotalSizeInBytes": 11,
  "KvsARN": "arn:aws:cloudfront::123456789012:key-value-store/37435e19-
  c205-4271-9e5c-example",
  "Created": "2024-05-08T07:48:45.381000-07:00",
  "LastModified": "2024-08-05T13:50:58.843000-07:00",
  "Status": "READY"
}
```

Buat pasangan kunci-nilai

Untuk membuat pasangan kunci-nilai di penyimpanan nilai kunci Anda, jalankan perintah berikut.

```
aws cloudfront-keyvaluestore put-key \
  --if-match=KV1PA6795UKMFR9 \
  --key=key2 \
  --value=value2 \
  --kvs-arn=arn:aws:cloudfront::123456789012:key-value-store/37435e19-c205-4271-9e5c-
  example
```

Respons

```
{
  "ETag": "KV13V1IB3VIYZZH",
  "ItemCount": 3,
  "TotalSizeInBytes": 31
}
```

Hapus pasangan kunci-nilai

Untuk menghapus pasangan kunci-nilai, jalankan perintah berikut.

```
aws cloudfront-keyvaluestore delete-key \  
  --if-match=KV13V1IB3VIYZZH \  
  --key=key1 \  
  --kvs-arn=arn:aws:cloudfront::123456789012:key-value-store/37435e19-c205-4271-9e5c-  
example
```

Keluaran

```
{  
  "ETag": "KV1VC38T7YXB528",  
  "ItemCount": 2,  
  "TotalSizeInBytes": 22  
}
```

Perbarui pasangan nilai kunci

Anda dapat menggunakan `update-keys` perintah untuk memperbarui lebih dari satu pasangan kunci-nilai. Misalnya, untuk menghapus pasangan kunci-nilai yang ada dan membuat yang lain, jalankan perintah berikut.

```
aws cloudfront-keyvaluestore update-keys \  
  --if-match=KV2EUQ1WTGCTBG2 \  
  --kvs-arn=arn:aws:cloudfront::123456789012:key-value-store/37435e19-c205-4271-9e5c-  
example \  
  --deletes '[{"Key":"key2"}]' \  
  --puts '[{"Key":"key3","Value":"value3"}]'
```

Respons

```
{  
  "ETag": "KV3AEGXETSR30VB",  
  "ItemCount": 3,  
  "TotalSizeInBytes": 28  
}
```

Bekerja dengan pasangan nilai kunci (API)

Ikuti bagian ini untuk bekerja dengan pasangan nilai kunci Anda secara terprogram.

Daftar Isi

- [Dapatkan referensi ke penyimpanan nilai kunci](#)
- [Ubah pasangan kunci-nilai di penyimpanan nilai kunci](#)
- [Contoh kode untuk CloudFront KeyValueCollection](#)

Dapatkan referensi ke penyimpanan nilai kunci

Saat Anda menggunakan CloudFront KeyValueCollection API untuk memanggil operasi tulis, Anda perlu menentukan ARN dan penyimpanan ETag nilai kunci. Untuk mendapatkan data ini, lakukan hal berikut:

Untuk mendapatkan referensi ke penyimpanan nilai kunci

1. Gunakan operasi [CloudFront ListKeyValueCollection](#) API untuk mendapatkan daftar penyimpanan nilai kunci. Temukan penyimpanan nilai kunci yang ingin Anda ubah.
2. Gunakan [operasi CloudFrontKeyValueCollection DescribeKeyValueCollection API](#) dan tentukan penyimpanan nilai kunci dari langkah sebelumnya.

Respons termasuk ARN dan penyimpanan ETag nilai kunci.

- ARN mencakup Akun AWS angka, konstantakey-value-store, dan UUID, seperti contoh berikut:

```
arn:aws:cloudfront::123456789012:key-value-store/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
```

- An ETag yang terlihat seperti contoh berikut:

```
ETVABCEXAMPLE2
```

Ubah pasangan kunci-nilai di penyimpanan nilai kunci

Anda dapat menentukan penyimpanan nilai kunci yang berisi pasangan kunci-nilai yang ingin Anda perbarui.

Lihat operasi CloudFront KeyValueCollection API berikut:

- [CloudFrontKeyValueCollection DeleteKey](#)— Menghapus pasangan kunci-nilai
- [CloudFrontKeyValueCollection GetKey](#)— Mengembalikan pasangan kunci-nilai

- [CloudFrontKeyVaultStore ListKeys](#)— Mengembalikan daftar pasangan kunci-nilai
- [CloudFrontKeyVaultStore PutKey](#)— Anda dapat melakukan tugas-tugas berikut:
 - Buat pasangan kunci-nilai dalam satu penyimpanan nilai kunci dengan menentukan nama dan nilai kunci baru.
 - Tetapkan nilai yang berbeda dalam pasangan kunci-nilai yang ada dengan menentukan nama kunci yang ada, dan nilai kunci baru.
- [CloudFrontKeyVaultStore UpdateKeys](#)— Anda dapat melakukan satu atau lebih tindakan berikut dalam satu all-or-nothing operasi:
 - Hapus satu atau lebih pasangan nilai kunci
 - Buat satu atau lebih pasangan kunci-nilai baru
 - Tetapkan nilai yang berbeda dalam satu atau lebih pasangan kunci-nilai yang ada

Contoh kode untuk CloudFront KeyVaultStore

Example

Kode berikut menunjukkan cara memanggil operasi DescribeKeyVaultStore API untuk penyimpanan nilai kunci.

```
const {
  CloudFrontKeyVaultStoreClient,
  DescribeKeyVaultStoreCommand,
} = require("@aws-sdk/client-cloudfront-keyvaultstore");

require("@aws-sdk/signature-v4-crt");

(async () => {
  try {
    const client = new CloudFrontKeyVaultStoreClient({
      region: "us-east-1"
    });
    const input = {
      KvsARN: "arn:aws:cloudfront::123456789012:key-value-store/a1b2c3d4-5678-90ab-
cdef-EXAMPLE11111",
    };
    const command = new DescribeKeyVaultStoreCommand(input);

    const response = await client.send(command);
  } catch (e) {
```

```
console.log(e);  
}  
})();
```

Sesuaikan dengan Fungsi CloudFront Koneksi

CloudFront Fungsi Koneksi memungkinkan Anda menulis JavaScript fungsi ringan untuk validasi sertifikat mTLS dan logika otentikasi khusus. Fungsi Koneksi Anda berjalan selama pembuatan koneksi mTLS untuk memvalidasi sertifikat klien, menerapkan aturan otentikasi khusus perangkat, dan menangani skenario pencabutan sertifikat. Lingkungan runtime Connection Functions menawarkan waktu startup submilidetik, skala segera untuk menangani jutaan koneksi per detik, dan sangat aman. Fungsi Koneksi adalah fitur asli CloudFront, yang berarti Anda dapat membangun, menguji, dan menyebarkan kode Anda sepenuhnya di dalamnya CloudFront.

Saat Anda mengaitkan Fungsi Koneksi dengan CloudFront distribusi berkemampuan MTLS, CloudFront mencegat permintaan koneksi TLS di lokasi CloudFront tepi dan meneruskan informasi sertifikat ke fungsi Anda. Anda dapat memanggil Fungsi Koneksi ketika peristiwa berikut terjadi:

- Selama pembentukan koneksi TLS (permintaan koneksi) - untuk koneksi TLS (mTLS) bersama

Untuk informasi selengkapnya tentang Fungsi Koneksi, lihat topik berikut.

Topik

- [Ikhtisar dan alur kerja](#)
- [Konfigurasi dan batasan](#)
- [Buat Fungsi CloudFront Koneksi untuk validasi TLS \(penampil\) bersama](#)
- [Tulis kode Fungsi CloudFront Koneksi untuk validasi TLS \(penampil\) timbal balik](#)
- [Uji Fungsi CloudFront Koneksi sebelum penerapan](#)
- [Mengaitkan Fungsi Koneksi dengan distribusi](#)
- [Menerapkan pencabutan sertifikat untuk TLS bersama \(penampil\) dengan Fungsi dan CloudFront KeyValueStore](#)

Ikhtisar dan alur kerja

CloudFront Fungsi Koneksi adalah jenis CloudFront Fungsi khusus yang berjalan selama jabat tangan TLS ketika klien mencoba membuat koneksi mTLS. Fungsi Koneksi Anda dapat mengakses

informasi sertifikat klien, parameter konfigurasi mTLS, hasil pemeriksaan pencabutan sertifikat, dan alamat IP klien.

Fungsi koneksi dipanggil setelah CloudFront melakukan validasi sertifikat standar (rantai kepercayaan, kedaluwarsa, verifikasi tanda tangan) tetapi dapat berjalan bahkan jika pemeriksaan pencabutan sertifikat gagal. Ini memungkinkan Anda menerapkan logika khusus untuk menangani sertifikat yang dicabut atau menambahkan kriteria validasi tambahan.

Setelah Anda membuat dan memublikasikan Fungsi Koneksi, pastikan untuk menambahkan asosiasi untuk jenis peristiwa permintaan koneksi dengan distribusi berkemampuan MTLS. Hal ini membuat fungsi berjalan setiap kali klien mencoba untuk membuat koneksi mTLS dengan CloudFront.

CloudFront Fungsi Koneksi mengikuti siklus hidup dua tahap yang memungkinkan Anda mengembangkan dan menguji fungsi sebelum menerapkannya ke produksi. Alur kerja ini memastikan bahwa Fungsi Koneksi Anda berfungsi dengan benar sebelum memengaruhi lalu lintas langsung.

Topik

- [Tahapan fungsi](#)
- [Alur kerja pengembangan](#)
- [Perbedaan dari jenis fungsi lainnya](#)

Tahapan fungsi

Fungsi koneksi ada dalam salah satu dari dua tahap:

- **PENGEMBANGAN** — Fungsi dalam tahap ini dapat dimodifikasi, diuji, dan diperbarui. Gunakan tahap ini untuk menulis dan men-debug kode fungsi Anda.
- **LIVE** — Fungsi pada tahap ini hanya baca dan menangani lalu lintas produksi. Anda tidak dapat memodifikasi fungsi di tahap LIVE secara langsung.

Saat Anda membuat Fungsi Koneksi baru, itu dimulai pada tahap PENGEMBANGAN. Setelah pengujian dan validasi, Anda mempublikasikan fungsi untuk memindahkannya ke tahap LIVE.

Alur kerja pengembangan

Ikuti alur kerja ini untuk mengembangkan dan menerapkan Fungsi Koneksi:

1. **Buat** - Buat Fungsi Koneksi baru di tahap PENGEMBANGAN dengan kode dan konfigurasi awal Anda.
2. **Uji** — Gunakan fungsionalitas pengujian untuk memvalidasi fungsi Anda dengan contoh peristiwa koneksi sebelum penerapan.
3. **Pembaruan** - Ubah kode fungsi dan konfigurasi sesuai kebutuhan berdasarkan hasil pengujian.
4. **Publikasikan** — Ketika siap untuk produksi, publikasikan fungsi untuk memindahkannya dari pengembangan ke tahap LIVE.
5. **Associate** — Kaitkan fungsi yang dipublikasikan dengan distribusi berkemampuan MTLS Anda untuk menangani koneksi langsung.

Untuk membuat perubahan pada fungsi LIVE, Anda harus memperbarui versi PENGEMBANGAN dan mempublikasikannya lagi. Ini menciptakan versi baru di panggung LIVE.

Perbedaan dari jenis fungsi lainnya

Fungsi koneksi berbeda dari permintaan penampil dan fungsi respons penampil dalam beberapa cara penting:

- Fungsi koneksi berjalan setelah jabat tangan mTLS, sebelum pemrosesan HTTP terjadi
- Fungsi koneksi memiliki akses ke informasi sertifikat TLS alih-alih data HTTP request/response
- Fungsi koneksi hanya dapat mengizinkan atau menolak koneksi, bukan memodifikasi data HTTP
- Fungsi koneksi hanya dipanggil untuk koneksi TLS baru, bukan untuk penggunaan kembali koneksi
- Dimulainya kembali sesi TLS tidak didukung dengan mTL untuk memastikan validasi sertifikat terjadi pada setiap koneksi
- Fungsi koneksi berjalan selain permintaan penampil standar dan fungsi respons penampil
- Anda mengaitkan Fungsi Koneksi pada tingkat distribusi, bukan pada tingkat perilaku cache.
- Fungsi koneksi hanya mendukung JavaScript runtime 2.0.

Konfigurasi dan batasan

CloudFront Fungsi Koneksi memiliki persyaratan konfigurasi khusus dan batas layanan karena peran khusus mereka dalam validasi koneksi TLS dan persyaratan kinerja komputasi tepi.

Topik

- [Persyaratan kode fungsi](#)

- [Batas layanan](#)
- [Opsis penyaringan fungsi](#)

Persyaratan kode fungsi

Fungsi koneksi memerlukan JavaScript kode yang memproses peristiwa koneksi TLS. Kode fungsi harus:

- Ditulis di JavaScript
- Memproses peristiwa koneksi dan membuat allow/deny keputusan
- Menyelesaikan eksekusi dalam batas waktu
- Menangani sertifikat dan logika validasi koneksi

Batas layanan

Fungsi koneksi tunduk pada batasan berikut:

- Ukuran fungsi - Kode fungsi dan konfigurasi terbatas ukurannya
- Waktu eksekusi - Fungsi memiliki batas waktu eksekusi yang ketat untuk pemrosesan koneksi TLS
- Batas asosiasi - Setiap distribusi hanya dapat memiliki satu Fungsi Koneksi yang terkait
- Pembatasan tahap - Hanya fungsi panggung LIVE yang dapat dikaitkan dengan distribusi

Opsis penyaringan fungsi

Saat mencantumkan Fungsi Koneksi, Anda dapat menggunakan filter berikut:

- Filter panggung - Filter berdasarkan PENGEMBANGAN atau tahap LANGSUNG
- Filter asosiasi — Filter berdasarkan ID distribusi atau asosiasi ID penyimpanan nilai kunci

Filter ini membantu Anda mengatur dan mengelola Fungsi Koneksi di berbagai lingkungan dan kasus penggunaan.

Buat Fungsi CloudFront Koneksi untuk validasi TLS (penampil) bersama

Anda membuat Fungsi CloudFront Koneksi dalam dua tahap:

1. Buat kode fungsi sebagai JavaScript. Anda dapat menggunakan contoh default dari CloudFront konsol atau menulis sendiri. Untuk informasi selengkapnya, lihat topik berikut:
 - Tulis kode Fungsi CloudFront Koneksi untuk validasi mTLS
 - CloudFront Fungsi Koneksi struktur acara dan format respons
 - Contoh kode fungsi koneksi
2. Gunakan CloudFront untuk membuat Fungsi Koneksi dan sertakan kode Anda. Kode ada di dalam fungsi (bukan sebagai referensi).

Topik

- [CloudFront konsol](#)
- [AWS CLI](#)

CloudFront konsol

Untuk membuat Fungsi Koneksi

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Pilih Buat fungsi.
3. Masukkan nama fungsi yang unik di dalam Akun AWS, pilih fungsi Koneksi sebagai jenis fungsi, lalu pilih Lanjutkan.
4. Halaman detail untuk Fungsi Koneksi baru muncul.

Note

Fungsi koneksi hanya mendukung JavaScript runtime 2.0. Untuk menggunakan KeyValueStore integrasi Fungsi CloudFront Koneksi dalam fungsi, Anda harus menggunakan versi runtime ini.

5. Di bagian Kode fungsi, pilih tab Build dan masukkan kode fungsi koneksi Anda. Kode contoh yang disertakan dalam tab Build menggambarkan sintaks dasar untuk kode Fungsi Koneksi.
6. Pilih Simpan perubahan.
7. Jika kode Fungsi Sambungan digunakan KeyValueStore untuk pemeriksaan pencabutan sertifikat atau validasi perangkat, Anda harus mengaitkan a. KeyValueStore

Anda dapat mengaitkan KeyValueTypeStore ketika Anda pertama kali membuat fungsi. Atau, Anda dapat mengaitkannya nanti, dengan mengaitkan Fungsi Koneksi.

Untuk mengaitkan KeyValueTypeStore sekarang, ikuti langkah-langkah ini:

- Buka KeyValueTypeStore bagian Associate dan pilih Associate existing KeyValueTypeStore.
- Pilih KeyValueTypeStore yang berisi data sertifikat untuk Fungsi Koneksi Anda, lalu pilih Associate KeyValueTypeStore.

CloudFront segera mengaitkan toko dengan fungsinya. Anda tidak perlu menyimpan fungsinya.

AWS CLI

Jika Anda menggunakan AWS CLI, Anda biasanya pertama membuat kode Fungsi Koneksi dalam sebuah file, dan kemudian membuat fungsi dengan AWS CLI.

Untuk membuat Fungsi Koneksi

1. Buat kode Fungsi Koneksi dalam sebuah file, dan simpan di direktori yang dapat disambungkan oleh komputer Anda.
2. Jalankan perintah seperti yang ditunjukkan pada contoh. Contoh ini menggunakan fileb:// notasi untuk meneruskan file. Ini juga termasuk jeda baris untuk membuat perintah lebih mudah dibaca.

```
aws cloudfront create-connection-function \  
  --name CertificateValidator \  
  --connection-function-config '{  
    "Comment":"Device certificate validation",  
    "Runtime":"cloudfront-js-2.0",  
    "KeyValueTypeStoreAssociations":{  
      "Quantity":1,  
      "Items":[  
        "KeyValueTypeStoreARN":"arn:aws:cloudfront::111122223333:key-value-  
store/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"  
      ]  
    }  
  }' \  
  --connection-function-code fileb://certificate-validator.js
```


Note

- Runtime - Fungsi koneksi hanya mendukung JavaScript runtime 2.0 (cloudfront-js-2.0).
- KeyValueStoreAssociations— Jika Fungsi Koneksi Anda digunakan KeyValueStore untuk validasi sertifikat, Anda dapat mengaitkan KeyValueStore saat pertama kali membuat fungsi. Atau, Anda dapat mengaitkannya nanti, dengan menggunakan update-connection-function. Kuantitas selalu 1 karena setiap Fungsi Koneksi hanya dapat memiliki satu KeyValueStore yang terkait dengannya.

3. Ketika perintah berhasil, Anda melihat output seperti berikut ini.

```
ETag: ETVABCEXAMPLE
ConnectionFunctionSummary:
  ConnectionFunctionConfig:
    Comment: Device certificate validation
    Runtime: cloudfront-js-2.0
    KeyValueStoreAssociations:
      Quantity: 1
      Items:
        - KeyValueStoreARN: arn:aws:cloudfront::111122223333:key-value-store/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111
  ConnectionFunctionMetadata:
    CreatedTime: '2024-09-04T16:32:54.292000+00:00'
    ConnectionFunctionARN: arn:aws:cloudfront::111122223333:connection-function/
CertificateValidator
  LastModifiedTime: '2024-09-04T16:32:54.292000+00:00'
  Stage: DEVELOPMENT
  Name: CertificateValidator
  Status: UNPUBLISHED
Location: https://cloudfront.amazonaws.com/2020-05-31/connection-function/arn:aws:cloudfront::connection-function/CertificateValidator
```

Sebagian besar informasi diulang dari permintaan. Informasi lain ditambahkan oleh CloudFront.

Note

- ETag— Nilai ini berubah setiap kali Anda memodifikasi Fungsi Koneksi. Anda memerlukan nilai ini untuk memperbarui atau mempublikasikan fungsi.

- Tahap - Fungsi koneksi baru dimulai pada tahap PENGEMBANGAN. Anda harus mempublikasikan fungsi untuk memindahkannya ke tahap LIVE sebelum mengaitkannya dengan distribusi.
- Status - Status fungsi TIDAK DIPUBLIKASIKAN sampai Anda mempublikasikannya ke panggung LIVE.

Tulis kode Fungsi CloudFront Koneksi untuk validasi TLS (penampil) timbal balik

CloudFront Fungsi Koneksi memungkinkan Anda untuk menulis JavaScript fungsi ringan untuk validasi sertifikat mTLS dan logika otentikasi kustom. Kode Fungsi Koneksi Anda dapat memvalidasi sertifikat klien, menerapkan aturan otentikasi khusus perangkat, menangani skenario pencabutan sertifikat, dan membuat allow/deny keputusan untuk koneksi TLS di lokasi tepi di seluruh dunia. CloudFront

Fungsi koneksi menyediakan cara yang ampuh untuk memperluas CloudFront validasi sertifikat bawaan dengan logika bisnis Anda sendiri. Tidak seperti permintaan penampil dan fungsi respons penampil yang memproses data HTTP, Fungsi Koneksi beroperasi pada lapisan TLS dan memiliki akses ke informasi sertifikat, alamat IP klien, dan detail koneksi TLS. Ini membuatnya ideal untuk menerapkan model keamanan tanpa kepercayaan, sistem otentikasi perangkat, dan kebijakan validasi sertifikat khusus yang melampaui validasi PKI standar.

Kode Fungsi Koneksi Anda berjalan di lingkungan yang aman dan terisolasi dengan waktu startup submilidetik dan dapat menskalakan untuk menangani jutaan koneksi per detik. Runtime dioptimalkan untuk beban kerja validasi sertifikat dan menyediakan integrasi bawaan CloudFront KeyValueStore untuk operasi pencarian data real-time, memungkinkan skenario otentikasi canggih seperti pemeriksaan daftar pencabutan sertifikat dan validasi daftar izin perangkat.

Untuk membantu Anda menulis kode Fungsi Koneksi yang efektif, lihat topik berikut. Untuk contoh kode lengkap dan step-by-step tutorial, lihat bagian tutorial dalam panduan ini dan jelajahi contoh Fungsi Koneksi yang tersedia di CloudFront konsol.

Topik

- [CloudFront Fungsi Koneksi menggunakan kasus dan tujuan](#)
- [CloudFront Fungsi Koneksi struktur acara dan format respons](#)
- [CloudFront Fitur JavaScript runtime Fungsi Koneksi](#)

- [CloudFront Metode pembantu Fungsi Koneksi dan APIs](#)
- [CloudFront KeyValueStore Integrasi Fungsi Koneksi](#)
- [Gunakan async dan await](#)
- [Contoh kode fungsi koneksi](#)

CloudFront Fungsi Koneksi menggunakan kasus dan tujuan

Sebelum menulis Fungsi CloudFront Koneksi Anda, tentukan dengan cermat jenis validasi sertifikat atau logika otentikasi yang perlu Anda terapkan. Fungsi koneksi dirancang untuk kasus penggunaan tertentu yang memerlukan validasi khusus di luar pemeriksaan sertifikat PKI standar. Memahami kasus penggunaan membantu Anda merancang kode efisien yang memenuhi persyaratan keamanan Anda sambil mempertahankan kinerja optimal.

Kasus penggunaan Fungsi Koneksi Umum meliputi:

- Penanganan pencabutan sertifikat — Menerapkan kebijakan khusus untuk menangani sertifikat yang dicabut, termasuk masa tenggang untuk rotasi sertifikat, pengecualian jaringan tepercaya untuk perangkat internal, atau skenario akses darurat di mana sertifikat yang dicabut mungkin memerlukan akses sementara.
- Dukungan mTLS opsional — Menangani koneksi mTL dan non-MTL dengan kebijakan otentikasi yang berbeda, memungkinkan Anda memberikan keamanan yang ditingkatkan untuk klien yang mendukung sertifikat sambil mempertahankan kompatibilitas dengan klien lama.
- Otentikasi berbasis IP — Gabungkan validasi sertifikat dengan pemeriksaan alamat IP klien untuk keamanan yang ditingkatkan, seperti membatasi akses dari wilayah geografis tertentu, jaringan perusahaan, atau rentang IP berbahaya yang diketahui.
- Validasi sertifikat multi-penyewa - Menerapkan aturan validasi khusus penyewa di mana otoritas sertifikat atau kriteria validasi yang berbeda berlaku berdasarkan penerbit sertifikat klien atau atribut subjek.
- Kontrol akses berbasis waktu - Menerapkan pembatasan berbasis waktu di mana sertifikat hanya berlaku selama jam tertentu, jendela pemeliharaan, atau periode bisnis, bahkan jika sertifikat itu sendiri belum kedaluwarsa.

Fungsi koneksi berjalan setelah CloudFront melakukan validasi sertifikat standar (verifikasi rantai kepercayaan, pemeriksaan kedaluwarsa, dan validasi tanda tangan) tetapi sebelum koneksi TLS dibuat. Waktu ini memberi Anda fleksibilitas untuk menambahkan kriteria validasi kustom sambil

memanfaatkan CloudFront validasi sertifikat bawaan. Fungsi Anda menerima hasil validasi standar dan dapat membuat keputusan berdasarkan informasi tentang apakah akan mengizinkan atau menolak koneksi berdasarkan kriteria standar dan khusus.

Saat merancang Fungsi Koneksi Anda, pertimbangkan implikasi kinerja logika validasi Anda. Fungsi memiliki batas eksekusi 5 milidetik, sehingga operasi yang kompleks harus dioptimalkan untuk kecepatan. Gunakan KeyValueStore untuk pencarian data yang cepat daripada perhitungan yang rumit, dan struktur logika validasi Anda agar cepat gagal untuk sertifikat yang tidak valid.

CloudFront Fungsi Koneksi struktur acara dan format respons

CloudFront Fungsi Koneksi menerima struktur peristiwa yang berbeda dari permintaan penampil dan fungsi respons penampil. Alih-alih request/response data HTTP, fungsi koneksi menerima sertifikat dan informasi koneksi yang dapat Anda gunakan untuk membuat keputusan otentikasi.

Topik

- [Struktur acara untuk Fungsi Koneksi](#)
- [Format respons Fungsi Koneksi](#)

Struktur acara untuk Fungsi Koneksi

Fungsi Koneksi menerima objek acara yang berisi sertifikat dan informasi koneksi. Struktur acara fungsi ditunjukkan di bawah ini:

```
{
  "clientCertificate": {
    "certificates": {
      "leaf": {
        "serialNumber": "string",
        "issuer": "string",
        "subject": "string",
        "validity": {
          "notBefore": "string",
          "notAfter": "string",
        },
        "sha256Fingerprint": "string"
      }
    }
  },
  "clientIp": "string",
```

```

"endpoint": "string",
"distributionId": "string",
"connectionId": "string"
}

```

Di bawah ini adalah contoh struktur objek acara:

```

{
  "clientCertificate": {
    "certificates": {
      "leaf": {
        "serialNumber": "00:9e:2a:af:16:56:e5:47:25:7d:2e:38:c3:f9:9d:57:fa",
        "issuer": "C=US, O=Ram, OU=Edge, ST=WA, CN=mTLS-CA, L=Snoqualmie",
        "subject": "C=US, O=Ram, OU=Edge, ST=WA, CN=mTLS-CA, L=Snoqualmie",
        "validity": {
          "notBefore": "2025-09-10T23:43:10Z",
          "notAfter": "2055-09-11T00:43:02Z"
        },
        "sha256Fingerprint": "_w6bJ7a0AlG0j7NUhJxTfsfee-0Ng_xop3_PTgTJpqs="
      }
    }
  },
  "clientIp": "127.0.0.1",
  "endpoint": "d3lch071jze0cb.cloudfront.net",
  "distributionId": "E1NXS4MQZH501R",
  "connectionId": "NpvTe1925xfj24a67sPQr7ae42BIq03FGhJJKfrQYWZcWZFp96SIIg=="
}

```

Format respons Fungsi Koneksi

Fungsi Koneksi Anda harus mengembalikan objek respons yang menunjukkan apakah akan mengizinkan atau menolak koneksi. Gunakan metode pembantu untuk membuat keputusan koneksi:

```

function connectionHandler(connection) {
  // Helper methods to allow or deny connections
  if (/* some logic to determine if function should allow connection */) {
    connection.allow();
  } else {
    connection.deny();
  }
}

```

Tidak seperti permintaan penampil dan fungsi respons penampil, Fungsi Koneksi tidak dapat mengubah permintaan atau tanggapan HTTP. Mereka hanya dapat mengizinkan atau menolak koneksi TLS.

CloudFront Fitur JavaScript runtime Fungsi Koneksi

CloudFront Fungsi Koneksi menggunakan CloudFront Functions JavaScript runtime 2.0, yang menyediakan lingkungan aman dan berkinerja tinggi yang secara khusus dioptimalkan untuk beban kerja validasi sertifikat. Runtime dirancang untuk memulai dalam sub-milidetik dan menangani jutaan eksekusi bersamaan di seluruh CloudFront jaringan edge global.

Lingkungan runtime mencakup dukungan JavaScript bahasa yang komprehensif:

- ECMAScript Dukungan 2020 (ES11) - JavaScript Fitur modern termasuk rantai opsional (?.) , penggabungan batal (??) , dan BigInt untuk menangani nomor seri sertifikat besar
- Objek bawaan - JavaScript Objek standar seperti Objek, Array, JSON, Matematika, dan Tanggal
- Console logging — Gunakan `console.log ()` untuk men-debug dan memantau keputusan validasi sertifikat. Log tersedia secara real-time selama pengujian dan dapat membantu memecahkan masalah logika validasi dalam pengembangan
- KeyValueStore integrasi — Akses asli CloudFront KeyValueStore untuk operasi pencarian data ultra-cepat, memungkinkan pemeriksaan pencabutan sertifikat waktu nyata, validasi daftar izin perangkat, dan pengambilan konfigurasi khusus penyewa

Fungsi koneksi dioptimalkan untuk kinerja tinggi untuk skenario validasi sertifikat. Runtime secara otomatis menangani manajemen memori, pengumpulan sampah, dan pembersihan sumber daya untuk memastikan kinerja yang konsisten di jutaan koneksi bersamaan. Semua operasi dirancang untuk menjadi deterministik dan cepat, dengan KeyValueStore pencarian biasanya selesai dalam mikrodetik.

Lingkungan runtime benar-benar terisolasi antara eksekusi fungsi, memastikan bahwa tidak ada kebocoran data antara koneksi klien yang berbeda. Setiap eksekusi fungsi dimulai dengan keadaan bersih dan tidak memiliki akses ke hasil eksekusi sebelumnya atau data klien dari koneksi lain.

CloudFront Metode pembantu Fungsi Koneksi dan APIs

CloudFront Fungsi Koneksi menyediakan metode pembantu khusus yang dirancang untuk menyederhanakan keputusan validasi sertifikat dan meningkatkan observabilitas. Metode ini

dioptimalkan untuk alur kerja validasi koneksi dan diintegrasikan secara mulus dengan CloudFront sistem pencatatan dan pemantauan koneksi.

- `connection.allow ()` - Izinkan koneksi TLS untuk melanjutkan. Metode ini memberi sinyal CloudFront untuk menyelesaikan jabat tangan TLS dan memungkinkan klien untuk membuat koneksi. Gunakan ini saat validasi sertifikat berlalu dan logika otentikasi kustom apa pun terpenuhi
- `connection.deny ()` - Tolak koneksi TLS dan hentikan jabat tangan. Metode ini segera menutup koneksi dan mencegah lalu lintas HTTP mengalir. Klien akan menerima kesalahan koneksi TLS. Gunakan ini untuk sertifikat yang tidak valid, autentikasi gagal, atau pelanggaran kebijakan
- `koneksi.logCustomData()` — Tambahkan data khusus ke log koneksi (hingga 800 byte teks UTF-8). Metode ini memungkinkan Anda untuk menyertakan hasil validasi, detail sertifikat, atau alasan keputusan dalam log CloudFront koneksi untuk pemantauan keamanan, audit kepatuhan, dan pemecahan masalah

Metode ini menyediakan antarmuka deklaratif yang bersih untuk membuat keputusan koneksi dan mencatat informasi yang relevan untuk pemantauan dan debugging. `allow/deny` Pola memastikan bahwa maksud fungsi Anda jelas dan CloudFront dapat mengoptimalkan penanganan koneksi berdasarkan keputusan Anda. Data pencatatan khusus segera tersedia di log CloudFront koneksi dan dapat digunakan dengan alat analisis log untuk pemantauan keamanan dan wawasan operasional.

Selalu panggil `connection.allow ()` atau `connection.deny ()` sebelum fungsi Anda selesai. Jika tidak ada metode yang dipanggil, CloudFront akan menolak koneksi secara default sebagai tindakan pencegahan keamanan.

CloudFront KeyValueCollection Integrasi Fungsi Koneksi

CloudFront Fungsi Koneksi dapat digunakan CloudFront KeyValueCollection untuk melakukan pencarian data ultra-cepat untuk skenario validasi sertifikat. KeyValueCollection sangat kuat untuk Fungsi Koneksi karena menyediakan akses data global yang akhirnya konsisten dengan waktu pencarian mikrodetik di semua lokasi tepi. CloudFront Ini membuatnya ideal untuk mempertahankan daftar pencabutan sertifikat, daftar izin perangkat, konfigurasi penyewa, dan data validasi lainnya yang perlu diakses selama jabat tangan TLS.

KeyValueCollection integrasi dirancang khusus untuk alur kerja validasi koneksi berkinerja tinggi:

- `KVShandle.exists (key)` - Periksa apakah ada kunci di tanpa mengambil nilainya. `KeyValueStore` Ini adalah metode yang paling efisien untuk skenario validasi biner seperti pemeriksaan pencabutan sertifikat, di mana Anda hanya perlu tahu apakah nomor seri sertifikat ada dalam daftar pencabutan
- `KVShandle.get (key)` - Ambil nilai dari skenario validasi yang lebih kompleks. `KeyValueStore` Gunakan ini saat Anda perlu mengakses data konfigurasi, aturan validasi, atau metadata yang terkait dengan sertifikat atau pengenalan perangkat

`KeyValueStore` operasi asinkron dan harus digunakan dengan sintaks `async/await`. Ini `KeyValueStore` memiliki batas ukuran total 10MB dan mendukung hingga 10 juta pasangan nilai kunci.

`KeyValueStore` data pada akhirnya konsisten di semua lokasi tepi, dengan pembaruan biasanya menyebar dalam hitungan detik.

Untuk kinerja optimal, susun `KeyValueStore` kunci Anda untuk meminimalkan operasi pencarian. Gunakan nomor seri sertifikat sebagai kunci untuk pemeriksaan pencabutan sederhana, atau buat kunci komposit yang menggabungkan hash penerbit dan nomor seri untuk lingkungan multi-CA. Pertimbangkan trade-off antara kompleksitas dan `KeyValueStore` kapasitas utama saat merancang struktur data Anda.

Gunakan `async` dan `await`

Fungsi koneksi mendukung operasi asinkron menggunakan `async/await` sintaks, yang penting saat bekerja dengan `KeyValueStore` operasi atau tugas asinkron lainnya. `async/await` Pola ini memastikan bahwa fungsi Anda menunggu `KeyValueStore` pencarian selesai sebelum membuat keputusan koneksi, sambil mempertahankan karakteristik kinerja tinggi yang diperlukan untuk pemrosesan jabatan TLS.

`async/await` Penggunaan yang tepat sangat penting untuk Fungsi Koneksi karena `KeyValueStore` operasi, meskipun sangat cepat, masih merupakan operasi jaringan yang memerlukan koordinasi di seluruh CloudFront infrastruktur terdistribusi. Runtime secara otomatis menangani resolusi janji dan memastikan bahwa fungsi Anda selesai dalam batas eksekusi 5 milidetik.

Example: Fungsi Koneksi Async dengan `KeyValueStore`

```
import cf from 'cloudfront';

async function connectionHandler(connection) {
    const kvsHandle = cf.kvs();

    // Async operation to check KeyValueStore for certificate revocation
```



```
const isRevoked = await
kvsHandle.exists(connection.clientCertificate.certificates.leaf.serialNumber);

if (isRevoked) {
    // Log the revocation decision with certificate details
    connection.logCustomData(`REVOKED_CERT:
${connection.clientCertificate.certificates.leaf.serialNumber}:
${connection.clientCertificate.certificates.leaf.issuer}`);
    console.log(`Denying connection for revoked certificate:
${connection.clientCertificate.certificates.leaf.serialNumber}`);
    return connection.deny();
}

// Log successful validation for monitoring
connection.logCustomData(`VALID_CERT:
${connection.clientCertificate.certificates.leaf.serialNumber}`);
console.log(`Allowing connection for valid certificate:
${connection.clientCertificate.certificates.leaf.serialNumber}`);
return connection.allow();
}
```

Selalu gunakan `async/await` saat memanggil `KeyValueStore` metode atau operasi asinkron lainnya. Runtime Fungsi Koneksi menangani resolusi janji secara otomatis dan memastikan alur eksekusi yang tepat dalam batasan waktu yang ketat dari pemrosesan jabat tangan TLS. Hindari menggunakan `.then ()` atau pola panggilan balik, karena `async/await` memberikan penanganan kesalahan yang lebih bersih dan kinerja yang lebih baik di lingkungan Fungsi Koneksi.

Saat merancang Fungsi Koneksi `async`, susun kode Anda untuk meminimalkan jumlah `KeyValueStore` operasi dan lakukan sedini mungkin dalam logika validasi Anda. Ini memastikan kinerja maksimum dan mengurangi risiko masalah batas waktu selama periode lalu lintas tinggi. Pertimbangkan untuk mengelompokkan pemeriksaan validasi terkait dan menggunakan `KeyValueStore` metode yang paling efisien (`exists ()` vs `get ()`) untuk kasus penggunaan Anda.

Contoh kode fungsi koneksi

Contoh berikut menunjukkan pola Fungsi Koneksi umum untuk skenario validasi yang berbeda. Gunakan contoh ini sebagai titik awal untuk implementasi Fungsi Koneksi Anda sendiri.

Example: Validasi sertifikat perangkat

Contoh ini memvalidasi nomor seri perangkat dan bidang subjek sertifikat untuk perangkat IoT, konsol game, dan skenario otentikasi klien lainnya:

```

async function connectionHandler(connection) {
  // Custom validation: check device serial number format
  const serialNumber = connection.clientCertificate.certificates.leaf.serialNumber;
  if (!serialNumber.startsWith("DEV")) {
    connection.logCustomData(`INVALID_SERIAL:${serialNumber}`);
    return connection.deny();
  }

  // Validate certificate subject contains required organizational unit
  const subject = connection.clientCertificate.certificates.leaf.subject;
  if (!subject.includes("OU=AuthorizedDevices")) {
    connection.logCustomData(`INVALID_OU:${subject}`);
    return connection.deny();
  }

  // Allow connection for valid devices
  connection.logCustomData(`VALID_DEVICE:${serialNumber}`);
  return connection.allow();
}

```

Fungsi ini melakukan beberapa pemeriksaan validasi di luar validasi sertifikat standar, termasuk format nomor seri perangkat dan verifikasi unit organisasi.

Example: MTL opsional dengan otentikasi campuran

Contoh ini menangani koneksi mTLS dan non-mTLS dengan kebijakan otentikasi yang berbeda:

```

async function connectionHandler(connection) {
  if (connection.clientCertificate) {
    // mTLS connection - enhanced validation for certificate holders
    const subject = connection.clientCertificate.certificates.leaf.subject;
    connection.logCustomData(`MTLS_SUCCESS:${subject}:${connection.clientIp}`);
    console.log(`mTLS connection from: ${subject}`);
    return connection.allow();
  } else {
    // Non-mTLS connection - apply IP-based restrictions
    const clientIp = connection.clientIp;

    // Only allow non-mTLS from specific IP ranges
    if (clientIp.startsWith("203.0.113.") || clientIp.startsWith("198.51.100.")) {
      connection.logCustomData(`NON_MTLS_ALLOWED:${clientIp}`);
      console.log(`Non-mTLS connection allowed from: ${clientIp}`);
      return connection.allow();
    }
  }
}

```

```
    }  
  
    connection.logCustomData(`NON_MTLS_DENIED:${clientIp}`);  
    return connection.deny();  
  }  
}
```

Fungsi ini memberikan keamanan yang ditingkatkan untuk klien dengan sertifikat sambil mempertahankan kompatibilitas dengan klien lama dari rentang IP terpercaya.

Uji Fungsi CloudFront Koneksi sebelum penerapan

Anda dapat menguji Fungsi CloudFront Koneksi pada tahap PENGEMBANGAN menggunakan operasi TestConnectionFunction API. Pengujian memungkinkan Anda memvalidasi logika fungsi Anda dengan contoh peristiwa koneksi sebelum memublikasikan ke tahap LIVE.

Topik

- [Proses pengujian](#)
- [Hasil tes](#)
- [Objek uji koneksi](#)

Proses pengujian

Untuk menguji Fungsi Koneksi:

1. Buat Fungsi Koneksi pada tahap PENGEMBANGAN
2. Siapkan objek koneksi uji yang mewakili peristiwa koneksi TLS
3. Gunakan operasi TestConnectionFunction API untuk menjalankan fungsi Anda dengan data pengujian
4. Tinjau hasil pengujian, termasuk output fungsi, log eksekusi, dan pesan kesalahan apa pun
5. Perbarui kode fungsi Anda sesuai kebutuhan dan ulangi proses pengujian

Hasil tes

Saat Anda menguji Fungsi Koneksi, hasilnya meliputi:

- Ringkasan fungsi — Metadata tentang fungsi yang diuji

- Pemanfaatan komputasi - Metrik kinerja yang menunjukkan penggunaan sumber daya
- Log eksekusi — Output konsol dari fungsi Anda, termasuk pernyataan logging
- Output fungsi - Hasil yang dikembalikan oleh fungsi Anda
- Pesan galat — Setiap kesalahan runtime atau pengecualian yang terjadi selama eksekusi

Objek uji koneksi

Objek uji koneksi adalah gumpalan biner (hingga 40KB) yang mewakili peristiwa koneksi TLS yang akan diproses fungsi Anda. Objek ini berisi sertifikat dan informasi koneksi yang digunakan fungsi Anda untuk membuat keputusan otentikasi.

Note

Struktur dan format spesifik objek uji koneksi ditentukan oleh runtime Fungsi CloudFront Koneksi. Konsultasikan dokumentasi atau kontak CloudFront Fungsi AWS Dukungan untuk detail tentang pembuatan objek pengujian yang sesuai untuk kasus penggunaan Anda.

Setelah membuat Fungsi Koneksi, Anda dapat:

- Uji fungsi — Gunakan fungsionalitas pengujian di konsol atau CLI untuk memvalidasi fungsi Anda dengan contoh peristiwa koneksi. Untuk informasi selengkapnya, lihat Pengujian Fungsi Koneksi.
- Perbarui fungsi - Ubah kode fungsi dan konfigurasi sesuai kebutuhan. Fungsi koneksi dalam tahap PENGEMBANGAN dapat diperbarui kapan saja.
- Publikasikan fungsi — Ketika siap untuk produksi, publikasikan fungsi untuk memindahkannya dari DEVELOPMENT ke tahap LIVE. Untuk informasi selengkapnya, lihat mengaitkan Fungsi Koneksi.
- Kaitkan dengan distribusi — Kaitkan fungsi yang diterbitkan dengan distribusi berkemampuan MTLS untuk menangani koneksi langsung. Untuk informasi selengkapnya, lihat mengaitkan fungsi koneksi.

Mengaitkan Fungsi Koneksi dengan distribusi

Setelah memublikasikan Fungsi Koneksi ke tahap LIVE, Anda harus mengaitkannya dengan distribusi berkemampuan MTLS untuk menangani koneksi langsung. Fungsi koneksi dikaitkan pada tingkat distribusi, tidak seperti permintaan penampil dan fungsi respons penampil yang terkait dengan perilaku cache.

Topik

- [Persyaratan asosiasi](#)
- [Mengatur fungsi dengan filter](#)
- [Pertimbangan deployment](#)

Persyaratan asosiasi

Untuk mengaitkan Fungsi Koneksi dengan distribusi:

- Fungsi harus dalam tahap LIVE
- Distribusi harus mengaktifkan mTL
- Distribusi harus memiliki penyimpanan kepercayaan yang valid yang dikonfigurasi
- Anda hanya dapat mengaitkan satu Fungsi Koneksi per distribusi

Mengatur fungsi dengan filter

CloudFront menyediakan kemampuan penyaringan untuk membantu Anda mengatur dan mengelola fungsi koneksi:

- Filter ID Distribusi - Temukan fungsi yang terkait dengan distribusi tertentu
- Filter penyimpanan nilai kunci - Temukan fungsi yang menggunakan penyimpanan nilai kunci tertentu untuk pencarian data
- Filter tahap - Daftar fungsi dalam pengembangan atau tahap LIVE

Gunakan filter ini saat mengelola beberapa Fungsi Koneksi di berbagai distribusi atau lingkungan pengembangan.

Pertimbangan deployment

Pertimbangkan faktor-faktor ini saat menerapkan Fungsi Koneksi:

- Penyebaran global - Fungsi koneksi diterapkan ke semua lokasi CloudFront tepi di seluruh dunia, yang mungkin memakan waktu beberapa menit
- Manajemen versi - Setiap versi yang diterbitkan membuat fungsi LIVE baru yang menggantikan versi sebelumnya

- Strategi rollback - Rencanakan rollback dengan menyimpan versi kerja sebelumnya dari kode fungsi Anda
- Pengujian dalam produksi - Pertimbangkan untuk menggunakan distribusi terpisah untuk lingkungan pementasan dan produksi

Menerapkan pencabutan sertifikat untuk TLS bersama (penampil) dengan Fungsi dan CloudFront KeyValueStore

Anda dapat menggunakan Fungsi CloudFront Koneksi dengan KeyValueStore untuk menerapkan pemeriksaan pencabutan sertifikat. Ini memungkinkan Anda mempertahankan daftar nomor seri sertifikat yang dicabut dan memeriksa sertifikat klien terhadap daftar ini selama jabat tangan TLS.

Untuk menerapkan pencabutan sertifikat, Anda memerlukan komponen-komponen ini:

- Distribusi yang dikonfigurasi dengan mTLS penampil
- Sebuah KeyValueStore berisi nomor seri sertifikat yang dicabut
- Fungsi Koneksi yang menanyakan status sertifikat KeyValueStore untuk memeriksa

Saat klien terhubung, CloudFront validasi sertifikat terhadap trust store, lalu jalankan Fungsi Koneksi Anda. Fungsi Anda memeriksa nomor seri sertifikat terhadap KeyValueStore dan mengizinkan atau menolak koneksi.

Topik

- [Langkah 1: Buat sertifikat KeyValueStore yang dicabut](#)
- [Langkah 2: Buat Fungsi Koneksi Pencabutan](#)
- [Langkah 3: Uji fungsi pencabutan Anda](#)
- [Langkah 4: Kaitkan fungsi ke distribusi Anda](#)
- [Skenario pencabutan lanjutan](#)

Langkah 1: Buat sertifikat KeyValueStore yang dicabut

Buat KeyValueStore nomor seri sertifikat yang telah dicabut yang dapat diperiksa oleh Fungsi Koneksi Anda selama koneksi mTLS.

Pertama, siapkan nomor seri sertifikat yang telah dicabut dalam format JSON:

```
{
  "data": [
    {
      "key": "ABC123DEF456",
      "value": ""
    },
    {
      "key": "789XYZ012GHI",
      "value": ""
    }
  ]
}
```

Unggah file JSON ini ke bucket S3, lalu buat: KeyValueCollection

```
aws s3 cp revoked-serials.json s3://your-bucket-name/revoked-serials.json
aws cloudfront create-key-value-store \
  --name revoked-serials-kvs \
  --import-source '{
    "SourceType": "S3",
    "SourceARN": "arn:aws:s3:::your-bucket-name/revoked-serials.json"
  }'
```

Tunggu KeyValueCollection hingga penyediaan selesai. Periksa status dengan:

```
aws cloudfront get-key-value-store --name "revoked-serials-kvs"
```

Langkah 2: Buat Fungsi Koneksi Pencabutan

Buat Fungsi Koneksi yang memeriksa nomor seri sertifikat KeyValueCollection untuk menentukan apakah sertifikat dicabut.

Buat Fungsi Koneksi yang memeriksa nomor seri sertifikat terhadap KeyValueCollection:

```
aws cloudfront create-connection-function \
  --name "revocation-control" \
  --connection-function-config file://connection-function-config.json \
  --connection-function-code file://connection-function-code.txt
```

File konfigurasi menentukan KeyValueCollection asosiasi:

```
{
  "Runtime": "cloudfront-js-2.0",
  "Comment": "A function that implements revocation control via KVS",
  "KeyValueStoreAssociations": {
    "Quantity": 1,
    "Items": [
      {
        "KeyValueStoreArn": "arn:aws:cloudfront::account-id:key-value-store/kvs-id"
      }
    ]
  }
}
```

Kode Fungsi Koneksi memeriksa sertifikat KeyValueStore yang dicabut:

```
import cf from 'cloudfront';

async function connectionHandler(connection) {
  const kvsHandle = cf.kvs();

  // Get parsed client serial number from client certificate
  const clientSerialNumber = connection.clientCertInfo.serialNumber;

  // Check KVS to see if serial number exists as a key
  const serialNumberExistsInKvs = await kvsHandle.exists(clientSerialNumber);

  // Deny connection if serial number exists in KVS
  if (serialNumberExistsInKvs) {
    console.log("Connection denied - certificate revoked");
    return connection.deny();
  }

  // Allow connections that don't exist in kvs
  console.log("Connection allowed");
  return connection.allow();
}
```

Langkah 3: Uji fungsi pencabutan Anda

Gunakan CloudFront konsol untuk menguji Fungsi Koneksi Anda dengan sertifikat sampel. Arahkan ke Fungsi Koneksi di konsol dan gunakan tab Uji.

Uji dengan sertifikat sampel

1. Tempelkan sertifikat sampel dalam format PEM ke antarmuka pengujian
2. Secara opsional menentukan alamat IP klien untuk menguji logika berbasis IP
3. Pilih fungsi Uji untuk melihat hasil eksekusi
4. Tinjau log eksekusi untuk memverifikasi logika fungsi Anda

Uji dengan sertifikat yang valid dan dicabut untuk memastikan fungsi Anda menangani kedua skenario dengan benar. Log eksekusi menunjukkan output console.log dan kesalahan apa pun yang terjadi selama eksekusi fungsi.

Langkah 4: Kaitkan fungsi ke distribusi Anda

Setelah Anda mempublikasikan Fungsi Koneksi, kaitkan dengan distribusi berkemampuan MTLS untuk mengaktifkan pemeriksaan pencabutan sertifikat.

Anda dapat mengaitkan fungsi dari halaman pengaturan distribusi atau dari tabel distribusi terkait Fungsi Koneksi. Arahkan ke pengaturan distribusi Anda, gulir ke bagian Authentication Mutual Authentication (mTLS) Viewer, pilih Fungsi Koneksi Anda, dan simpan perubahannya.

Skenario pencabutan lanjutan

Untuk persyaratan pencabutan sertifikat yang lebih kompleks, pertimbangkan konfigurasi tambahan ini:

Topik

- [Konversi Daftar Pencabutan Sertifikat \(CRL\) ke format KeyValueType](#)
- [Menangani beberapa Otoritas Sertifikat](#)
- [Tambahkan data khusus ke log koneksi](#)
- [Kelola pembaruan CRL](#)
- [KeyValueType Kapasitas rencana](#)

Konversi Daftar Pencabutan Sertifikat (CRL) ke format KeyValueType

Jika Anda memiliki file Certificate Revocation List (CRL), Anda dapat mengonversinya ke KeyValueType format JSON menggunakan OpenSSL dan jq:

Konversi CRL ke format KeyValueType

Ekstrak nomor seri dari file CRL:

```
openssl crl -text -noout -in rfc5280_CRL.crl | \
  awk '/Serial Number:/ {print $3}' | \
  cut -d'=' -f2 | \
  sed 's/./&:/g;s/:$///' >> serialnumbers.txt
```

Konversi nomor seri ke format KeyValueType JSON:

```
jq -R -s 'split("\n") | map(select(length > 0)) | {data: map({"key": ., "value": ""})}' \
  serialnumbers.txt >> serialnumbers_kvs.json
```

Unggah file yang diformat ke S3 dan buat KeyValueType seperti yang dijelaskan pada Langkah 1.

Menangani beberapa Otoritas Sertifikat

Jika Anda TrustStore berisi beberapa Otoritas Sertifikat (CAs), sertakan informasi penerbit dalam KeyValueType kunci Anda untuk menghindari konflik antar sertifikat dari CAs yang berbeda yang mungkin memiliki nomor seri yang sama.

Untuk skenario multi-CA, gunakan kombinasi SHA1 hash penerbit dan nomor seri sebagai kuncinya:

```
import cf from 'cloudfront';

async function connectionHandler(connection) {
  const kvsHandle = cf.kvs();
  const clientCert = connection.clientCertInfo;

  // Create composite key with issuer hash and serial number
  const issuer = clientCert.issuer.replace(/[^a-zA-Z0-9]/g, '').substring(0, 20);
  const serialno = clientCert.serialNumber;
  const compositeKey = `${issuer}_${serialno}`;

  const cert_revoked = await kvsHandle.exists(compositeKey);

  if (cert_revoked) {
    console.log(`Blocking revoked cert: ${serialno} from issuer: ${issuer}`);
    connection.deny();
  } else {
    connection.allow();
  }
}
```

```
}  
}
```

Note

Menggunakan pengenal penerbit+nomor seri membuat kunci yang lebih panjang, yang dapat mengurangi jumlah total entri yang dapat Anda simpan di. KeyValueType

Tambahkan data khusus ke log koneksi

Fungsi koneksi dapat menambahkan data khusus ke log CloudFront koneksi menggunakan logCustomData metode ini. Ini memungkinkan Anda menyertakan hasil pemeriksaan pencabutan, informasi sertifikat, atau data relevan lainnya di log Anda.

```
async function connectionHandler(connection) {  
  const kvsHandle = cf.kvs();  
  const clientSerialNumber = connection.clientCertInfo.serialNumber;  
  const serialNumberExistsInKvs = await kvsHandle.exists(clientSerialNumber);  
  
  if (serialNumberExistsInKvs) {  
    // Log revocation details to connection logs  
    connection.logCustomData(`REVOKED:${clientSerialNumber}:DENIED`);  
    console.log("Connection denied - certificate revoked");  
    return connection.deny();  
  }  
  
  // Log successful validation  
  connection.logCustomData(`VALID:${clientSerialNumber}:ALLOWED`);  
  console.log("Connection allowed");  
  return connection.allow();  
}
```

Data kustom dibatasi hingga 800 byte teks UTF-8 yang valid. Jika Anda melebihi batas ini, CloudFront potong data ke batas UTF-8 terdekat yang valid.

Note

Pencatatan data khusus hanya berfungsi saat log koneksi diaktifkan untuk distribusi Anda. Jika log koneksi tidak dikonfigurasi, logCustomData metode ini adalah no-op.

Kelola pembaruan CRL

Otoritas Sertifikat dapat mengeluarkan dua jenis CRLs:

- Lengkap CRLs: Berisi daftar lengkap semua sertifikat yang dicabut
- Delta CRLs: Hanya sertifikat daftar yang dicabut sejak CRL penuh terakhir

Untuk pembaruan CRL lengkap, buat yang baru KeyValueCollection dengan data yang diperbarui dan alihkan asosiasi Fungsi Koneksi ke yang baru. KeyValueCollection Pendekatan ini lebih sederhana daripada menghitung perbedaan dan melakukan pembaruan tambahan.

Untuk pembaruan delta CRL, gunakan perintah `update-keys` untuk menambahkan sertifikat baru yang telah dicabut ke yang sudah ada: KeyValueCollection

```
aws cloudfront update-key-value-store \  
  --name "revoked-serials-kvs" \  
  --if-match "current-etag" \  
  --put file://delta-revoked-serials.json
```

KeyValueCollection Kapasitas rencana

KeyValueCollection memiliki batas ukuran 5 MB dan mendukung hingga 10 juta pasangan nilai kunci. Rencanakan kapasitas daftar pencabutan Anda berdasarkan format kunci dan ukuran data Anda:

- Nomor seri saja: Penyimpanan yang efisien untuk pemeriksaan pencabutan sederhana
- Pengidentifikasi penerbit+nomor seri: Kunci yang lebih panjang untuk lingkungan multi-CA

Untuk daftar pencabutan besar, pertimbangkan untuk menerapkan pendekatan berjenjang di mana Anda memelihara terpisah KeyValueStores untuk kategori sertifikat atau periode waktu yang berbeda.

Sesuaikan di tepi dengan Lambda @Edge

Lambda @Edge adalah perpanjangan dari AWS Lambda. Lambda @Edge adalah layanan komputasi yang memungkinkan Anda menjalankan fungsi yang menyesuaikan konten yang diberikan Amazon CloudFront. Anda dapat membuat fungsi Node.js atau Python di konsol Lambda dalam satu Wilayah AWS, US East (Virginia N.).

Setelah membuat fungsi, Anda dapat menambahkan pemicu menggunakan konsol CloudFront atau konsol Lambda sehingga fungsi berjalan AWS di lokasi yang lebih dekat dengan penampil, tanpa menyediakan atau mengelola server. Secara opsional, Anda dapat menggunakan operasi Lambda CloudFront dan API untuk mengatur fungsi dan pemicu Anda secara terprogram.

Lambda@Edge menskalakan secara otomatis, dari beberapa permintaan per hari menjadi ribuan per detik. Memproses permintaan di AWS lokasi yang lebih dekat dengan penampil alih-alih di server asal secara signifikan mengurangi latensi dan meningkatkan pengalaman pengguna.

Note

Lambda @Edge tidak didukung dengan permintaan gRPC. Untuk informasi lebih lanjut, lihat. [Menggunakan gRPC dengan distribusi CloudFront](#)

Topik

- [Bagaimana Lambda @Edge bekerja dengan permintaan dan tanggapan](#)
- [Cara menggunakan Lambda @Edge](#)
- [Memulai dengan fungsi Lambda @Edge \(konsol\)](#)
- [Siapkan izin dan peran IAM untuk Lambda @Edge](#)
- [Tulis dan buat fungsi Lambda @Edge](#)
- [Tambahkan pemicu untuk fungsi Lambda @Edge](#)
- [Uji dan debug fungsi Lambda @Edge](#)
- [Hapus fungsi dan replika Lambda @Edge](#)
- [Struktur acara Lambda @Edge](#)
- [Bekerja dengan permintaan dan tanggapan](#)
- [Lambda @Edge contoh fungsi](#)

Bagaimana Lambda @Edge bekerja dengan permintaan dan tanggapan

Saat Anda mengaitkan CloudFront distribusi dengan fungsi Lambda @Edge, CloudFront mencegat permintaan dan respons di lokasi tepi. CloudFront Anda dapat menjalankan fungsi Lambda ketika CloudFront peristiwa berikut terjadi:

- Saat CloudFront menerima permintaan dari penampil (permintaan penampil)

- Sebelum CloudFront meneruskan permintaan ke asal (permintaan asal)
- Ketika CloudFront menerima respons dari asal (respons asal)
- Sebelum CloudFront mengembalikan respons ke penampil (respons penampil)

Jika Anda menggunakan AWS WAF, permintaan penampil Lambda @Edge dijalankan setelah AWS WAF aturan apa pun diterapkan.

Untuk informasi selengkapnya, lihat [Bekerja dengan permintaan dan tanggapan](#) dan [Struktur acara Lambda @Edge](#).

Cara menggunakan Lambda @Edge

Ada banyak kegunaan untuk pemrosesan Lambda @Edge dengan CloudFront distribusi Amazon Anda, seperti contoh berikut:

- Fungsi Lambda dapat memeriksa cookie dan menulis ulang URLs sehingga pengguna melihat versi situs yang berbeda untuk pengujian. A/B
- CloudFront dapat mengembalikan objek yang berbeda ke pemirsa berdasarkan perangkat yang mereka gunakan dengan memeriksa User-Agent header, yang mencakup informasi tentang perangkat. Misalnya, CloudFront dapat mengembalikan gambar yang berbeda berdasarkan ukuran layar perangkat mereka. Demikian pula, fungsi tersebut dapat mempertimbangkan nilai Referer header dan menyebabkan CloudFront mengembalikan gambar ke bot yang memiliki resolusi terendah yang tersedia.
- Atau Anda dapat memeriksa cookie untuk kriteria lainnya. Misalnya, di situs web ritel yang menjual pakaian, jika Anda menggunakan cookie untuk menunjukkan warna mana yang dipilih pengguna untuk jaket, fungsi Lambda dapat mengubah permintaan sehingga CloudFront mengembalikan gambar jaket dalam warna yang dipilih.
- Fungsi Lambda dapat menghasilkan respons HTTP saat permintaan CloudFront penampil atau peristiwa permintaan asal terjadi.
- Fungsi dapat memeriksa header atau token otorisasi, dan menyisipkan header untuk mengontrol akses ke konten Anda sebelum CloudFront meneruskan permintaan ke asal Anda.
- Fungsi Lambda juga dapat melakukan panggilan jaringan ke sumber daya eksternal untuk mengonfirmasi kredensial pengguna, atau mengambil konten tambahan untuk menyesuaikan respons.

Untuk informasi selengkapnya, termasuk kode contoh, lihat [Lambda @Edge contoh fungsi](#).

Untuk informasi selengkapnya tentang pengaturan Lambda @Edge di konsol, lihat [Tutorial: Buat fungsi Lambda @Edge dasar \(konsol\)](#)

Memulai dengan fungsi Lambda @Edge (konsol)

Dengan Lambda @Edge, Anda dapat menggunakan CloudFront pemacu untuk menjalankan fungsi Lambda. Saat Anda mengaitkan CloudFront distribusi dengan fungsi Lambda, CloudFront [mencegat permintaan dan respons di lokasi CloudFront tepi dan](#) menjalankan fungsi tersebut. Fungsi Lambda dapat meningkatkan keamanan atau menyesuaikan informasi yang dekat dengan pemirsa Anda untuk meningkatkan kinerja.

Daftar berikut memberikan ikhtisar dasar tentang cara membuat dan menggunakan fungsi Lambda dengan CloudFront

Ikhtisar: Membuat dan menggunakan fungsi Lambda dengan CloudFront

1. Buat fungsi Lambda di Wilayah AS Timur (Virginia N.).
2. Simpan dan publikasikan versi bernomor dari fungsi tersebut.

Jika ingin mengubah fungsi, Anda harus mengedit versi \$LATEST dari fungsi di Wilayah Timur AS (N. Virginia). Kemudian, sebelum Anda mengaturnya agar berfungsi CloudFront, Anda menerbitkan versi bernomor baru.

3. Kaitkan fungsi dengan perilaku CloudFront distribusi dan cache. Kemudian tentukan satu atau lebih CloudFront peristiwa (pemacu) yang menyebabkan fungsi dijalankan. Misalnya, Anda dapat membuat pemacu untuk fungsi yang akan dijalankan saat CloudFront menerima permintaan dari penampil.
4. Saat Anda membuat pemacu, Lambda membuat replika fungsi di AWS lokasi di seluruh dunia.

Tip

Untuk informasi selengkapnya, lihat [membuat dan memperbarui fungsi](#), [struktur acara](#), dan [menambahkan CloudFront pemacu](#). Anda juga dapat menemukan lebih banyak ide dan mendapatkan sampel kode di [Lambda @Edge contoh fungsi](#).

Untuk step-by-step tutorial, lihat topik berikut:

Topik

- [Tutorial: Buat fungsi Lambda @Edge dasar \(konsol\)](#)

Tutorial: Buat fungsi Lambda @Edge dasar (konsol)

Tutorial ini menunjukkan kepada Anda bagaimana memulai dengan Lambda @Edge dengan membuat dan mengonfigurasi contoh fungsi Node.js yang berjalan di CloudFront. Contoh ini menambahkan header keamanan HTTP ke respons saat CloudFront mengambil file. (Ini dapat meningkatkan keamanan dan privasi untuk situs web.)

Anda tidak memerlukan situs web Anda sendiri untuk tutorial ini. Namun, ketika Anda memilih untuk membuat solusi Lambda @Edge Anda sendiri, Anda mengikuti langkah-langkah serupa dan memilih dari opsi yang sama.

Topik

- [Langkah 1: Mendaftar untuk Akun AWS](#)
- [Langkah 2: Buat CloudFront distribusi](#)
- [Langkah 3: Buat fungsi Anda](#)
- [Langkah 4: Tambahkan CloudFront pemicu untuk menjalankan fungsi](#)
- [Langkah 5: Verifikasi bahwa fungsi berjalan](#)
- [Langkah 6: Memecahkan masalah](#)
- [Langkah 7: Bersihkan sumber daya contoh Anda](#)
- [Informasi Terkait](#)

Langkah 1: Mendaftar untuk Akun AWS

Jika Anda belum melakukannya, daftar untuk Akun AWS. Untuk informasi selengkapnya, lihat [Mendaftar untuk Akun AWS](#).

Langkah 2: Buat CloudFront distribusi

Sebelum membuat contoh fungsi Lambda@Edge, Anda harus memiliki CloudFront yang dapat dikerjakan, yang mencakup asal muasal untuk menyajikan konten.

Untuk contoh ini, Anda membuat CloudFront distribusi yang menggunakan bucket Amazon S3 sebagai asal distribusi. Jika Anda sudah memiliki lingkungan untuk digunakan, Anda dapat melewati langkah ini.

Untuk membuat CloudFront distribusi dengan asal Amazon S3

1. Buat bucket Amazon S3 dengan satu atau dua file, seperti file citra, untuk konten sampel. Untuk bantuan, ikuti langkah dalam [Unggah konten Anda ke Amazon S3](#). Pastikan Anda mengatur izin untuk memberikan akses baca publik ke objek dalam keranjang Anda.
2. Buat CloudFront distribusi dan tambahkan bucket S3 Anda sebagai asal, dengan mengikuti langkah-langkah di [Buat distribusi CloudFront web](#). Jika Anda sudah memiliki distribusi, Anda dapat menambahkan bucket sebagai asal distribusi tersebut.

 Tip

Catat ID distribusi Anda. Kemudian dalam tutorial ini ketika Anda menambahkan CloudFront pemicu untuk fungsi Anda, Anda harus memilih ID untuk distribusi Anda dalam daftar dropdown—misalnya, E653W22221KDDL

Langkah 3: Buat fungsi Anda

Pada langkah ini, Anda membuat fungsi Lambda dari templat cetak biru di konsol Lambda. Fungsi menambahkan kode untuk memperbarui header keamanan di CloudFront distribusi.

Untuk membuat fungsi Lambda

1. Masuk ke Konsol Manajemen AWS dan buka AWS Lambda konsol di <https://console.aws.amazon.com/lambda/>.

 Important

Pastikan Anda berada di AS-Timur-1 (Virginia N.) (Wilayah AWS us-east-1). Anda harus berada di Wilayah ini untuk membuat fungsi Lambda@Edge.

2. Pilih Buat fungsi.
3. Pada halaman fungsi Buat, pilih Gunakan cetak biru, lalu filter untuk CloudFront cetak biru dengan memasukkan di bidang pencarian. **cloudfront**

 Note

CloudFront cetak biru hanya tersedia di Wilayah AS-Timur-1 (Virginia N.) (us-east-1).

4. Pilih cetak biru header Modify HTTP response sebagai template untuk fungsi Anda.
5. Masukkan informasi tentang fungsi Anda berikut ini:
 - Nama fungsi — Masukkan nama untuk fungsi Anda.
 - Peran eksekusi - Pilih cara mengatur izin untuk fungsi Anda. Untuk menggunakan templat kebijakan izin Lambda @Edge dasar yang direkomendasikan, pilih Buat peran baru dari AWS templat kebijakan.
 - Nama peran — Masukkan nama untuk peran yang dibuat oleh templat kebijakan.
 - Templat kebijakan — Lambda secara otomatis menambahkan templat kebijakan Izin Lambda @Edge Dasar karena Anda memilih CloudFront cetak biru sebagai dasar fungsi Anda. Templat kebijakan ini menambahkan izin peran eksekusi yang memungkinkan CloudFront untuk menjalankan fungsi Lambda untuk Anda CloudFront di lokasi di seluruh dunia. Untuk informasi selengkapnya, lihat [Siapkan izin dan peran IAM untuk Lambda @Edge](#).
6. Pilih Buat fungsi di bagian bawah halaman.
7. Di panel Deploy to Lambda @Edge yang muncul, pilih Batal. (Untuk tutorial ini, Anda harus memodifikasi kode fungsi sebelum menerapkan fungsi ke Lambda @Edge.)
8. Gulir ke bawah ke bagian Sumber kode halaman.
9. Ganti kode templat dengan fungsi yang memodifikasi header keamanan yang dikembalikan oleh Anda. Misalnya, Anda dapat menggunakan kode yang serupa dengan yang berikut ini:

```
'use strict';
export const handler = (event, context, callback) => {

    //Get contents of response
    const response = event.Records[0].cf.response;
    const headers = response.headers;

    //Set new headers
    headers['strict-transport-security'] = [{key: 'Strict-Transport-Security',
value: 'max-age= 63072000; includeSubdomains; preload'}];
    headers['content-security-policy'] = [{key: 'Content-Security-Policy', value:
"default-src 'none'; img-src 'self'; script-src 'self'; style-src 'self'; object-
src 'none'"}];
    headers['x-content-type-options'] = [{key: 'X-Content-Type-Options', value:
'nosniff'}];
    headers['x-frame-options'] = [{key: 'X-Frame-Options', value: 'DENY'}];
    headers['x-xss-protection'] = [{key: 'X-XSS-Protection', value: '1;
mode=block'}];
```

```
headers['referrer-policy'] = [{key: 'Referrer-Policy', value: 'same-origin'}];

//Return modified response
callback(null, response);
};
```

10. Pilih File, Simpan untuk menyimpan kode yang diperbarui.

11. Pilih Deploy.

Lanjutkan ke bagian berikutnya untuk menambahkan CloudFront pemicu untuk menjalankan fungsi.

Langkah 4: Tambahkan CloudFront pemicu untuk menjalankan fungsi

Sekarang Anda memiliki fungsi Lambda untuk memperbarui header keamanan, konfigurasi CloudFront memicu untuk menjalankan fungsi Anda untuk menambahkan header dalam respons apapun yang CloudFront dari tempat awal distribusi Anda.

Untuk mengkonfigurasi CloudFront pemicu untuk fungsi Anda

1. Di konsol Lambda, pada halaman Ikhtisar fungsi untuk fungsi Anda, pilih Tambah pemicu.
2. Untuk konfigurasi Trigger, pilih CloudFront.
3. Pilih Terapkan ke Lambda @Edge.
4. Di panel Deploy to Lambda @Edge, di bawah pemicu CloudFront Konfigurasi, masukkan informasi berikut:
 - Distribusi — ID CloudFront distribusi untuk dikaitkan dengan fungsi Anda. Dalam daftar dropdown, pilih ID distribusi.
 - Perilaku cache — Perilaku cache yang digunakan dengan pemicu. Untuk contoh ini, biarkan nilai disetel ke *, yang berarti perilaku cache default distribusi Anda. Untuk informasi selengkapnya, lihat [Pengaturan perilaku cache](#) dalam topik [Semua referensi pengaturan distribusi](#).
 - CloudFront event - Pemicu yang menentukan kapan fungsi Anda berjalan. Kami ingin fungsi header keamanan berjalan setiap kali CloudFront mengembalikan respons dari asal. Dalam daftar dropdown, pilih Respons asal. Untuk informasi selengkapnya, lihat [Tambahkan pemicu untuk fungsi Lambda @Edge](#).
5. Pilih kotak centang Konfirmasi penerapan ke Lambda @Edge.
6. Pilih Deploy untuk menambahkan pemicu dan mereplikasi fungsi ke AWS lokasi di seluruh dunia.

7. Tunggu fungsi mereplikasi. Ini biasanya memerlukan waktu beberapa menit.

Anda dapat memeriksa untuk melihat apakah replikasi selesai oleh [menuju ke CloudFront konsol](#) dan melihat distribusi Anda. Tunggu status distribusi berubah dari Deploying ke tanggal dan waktu, yang berarti bahwa fungsi Anda telah direplikasi. Untuk memverifikasi bahwa fungsi bekerja, ikuti langkah-langkah di bagian berikutnya.

Langkah 5: Verifikasi bahwa fungsi berjalan

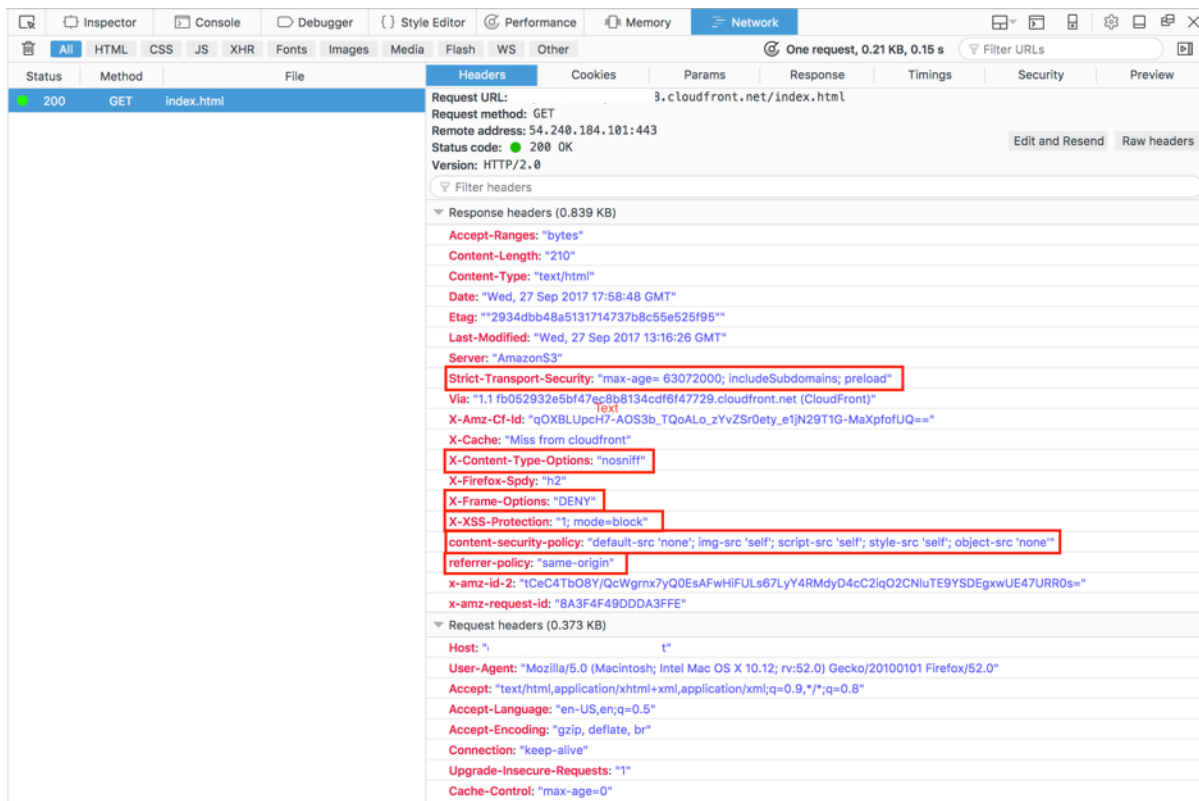
Sekarang setelah Anda membuat fungsi Lambda dan mengonfigurasi pemicu untuk menjalankannya untuk CloudFront distribusi, periksa untuk memastikan bahwa fungsi tersebut memenuhi apa yang Anda harapkan. Dalam contoh ini, kami memeriksa header HTTP yang CloudFront kembali, untuk memastikan bahwa judul keamanan ditambahkan.

Untuk memverifikasi bahwa fungsi Lambda@Edge Anda menambahkan header keamanan

1. Dalam peramban, masukkan URL untuk file dalam buket S3. Misalnya, Anda mungkin menggunakan URL yang serupa dengan `https://d111111abcdef8.cloudfront.net/image.jpg`.

Untuk informasi selengkapnya tentang nama CloudFront domain yang akan digunakan dalam URL file, lihat [Sesuaikan format URL untuk file di CloudFront](#).

2. Buka bilah alat Pengembang Web browser Anda. Misalnya, di jendela browser Anda di Chrome, buka menu konteks (klik kanan), lalu pilih Periksa.
3. Pilih Jaringan tab.
4. Muat ulang halaman untuk menampilkan citra Anda, lalu pilih permintaan HTTP di panel kiri. Anda melihat header HTTP yang ditampilkan dalam panel terpisah.
5. Lihat daftar header HTTP untuk memverifikasi bahwa header keamanan yang diharapkan disertakan dalam daftar. Misalnya, Anda mungkin melihat header yang mirip dengan yang ditampilkan di screenshot berikut.



Jika header keamanan disertakan dalam daftar header Anda, bagus! Anda telah berhasil membuat fungsi Lambda@Edge pertama Anda. Jika CloudFront mengembalikan kesalahan atau ada masalah lain, lanjutkan ke langkah berikutnya untuk memecahkan masalah.

Langkah 6: Memecahkan masalah

Jika CloudFront mengembalikan kesalahan atau tidak menambahkan header keamanan seperti yang diharapkan, Anda dapat menyelidiki eksekusi fungsi Anda dengan melihat CloudWatch Log. Pastikan untuk menggunakan log yang disimpan di AWS lokasi yang paling dekat dengan lokasi di mana fungsi dijalankan.

Misalnya, jika Anda melihat file dari London, coba ubah Wilayah dalam CloudWatch konsol ke Eropa (London).

Untuk memeriksa CloudWatch log untuk fungsi Lambda @Edge Anda

1. Masuk ke Konsol Manajemen AWS dan buka CloudWatch konsol di <https://console.aws.amazon.com/cloudwatch/>.
2. Perubahan Wilayah ke lokasi yang ditampilkan saat Anda melihat file di browser Anda. Di sinilah fungsi beroperasi.

3. Di panel kiri, pilih Log untuk melihat log untuk distribusi Anda.

Untuk informasi selengkapnya, lihat [Pantau CloudFront metrik dengan Amazon CloudWatch](#).

Langkah 7: Bersihkan sumber daya contoh Anda

Jika Anda membuat bucket dan CloudFront distribusi Amazon S3 hanya untuk tutorial ini, hapus AWS sumber daya yang Anda alokasikan sehingga Anda tidak lagi dikenakan biaya. Setelah Anda menghapus AWS sumber daya, konten apa pun yang Anda tambahkan tidak lagi tersedia.

Tugas

- [Hapus ember S3](#)
- [Hapus fungsi Lambda](#)
- [Hapus CloudFront distribusi](#)

Hapus ember S3

Sebelum Anda menghapus bucket Amazon S3, pastikan pembuatan log dinonaktifkan untuk bucket. Jika tidak, AWS terus menulis log ke bucket Anda saat Anda menghapusnya.

Untuk menonaktifkan pembuatan log untuk bucket

1. Buka konsol Amazon S3 di <https://console.aws.amazon.com/s3/>
2. Pilih keranjang Anda, lalu pilih Properti.
3. Dari Properti, pilih Pencatatan.
4. Hapus kotak centang Diaktifkan.
5. Pilih Simpan.

Sekarang, Anda dapat menghapus bucket Anda. Untuk informasi selengkapnya, lihat [Menghapus bucket](#) di Panduan Pengguna Konsol Layanan Penyimpanan Sederhana Amazon.

Hapus fungsi Lambda

Untuk instruksi untuk menghapus asosiasi fungsi Lambda dan opsional fungsi itu sendiri, lihat [Hapus fungsi dan replika Lambda @Edge](#)

Hapus CloudFront distribusi

Sebelum Anda menghapus CloudFront distribusi, Anda harus menonaktifkannya. Distribusi yang dinonaktifkan tidak lagi berfungsi, dan tidak dikenakan biaya. Anda dapat mengaktifkan distribusi yang dinonaktifkan kapan saja. Setelah Anda menghapus distribusi yang dinonaktifkan, maka tidak lagi tersedia.

Untuk menonaktifkan dan menghapus CloudFront distribusi

1. Buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Pilih distribusi yang ingin Anda nonaktifkan, lalu pilih Nonaktifkan.
3. Saat diminta untuk mengonfirmasi, pilih Ya, Nonaktifkan.
4. Pilih distribusi yang dinonaktifkan, lalu pilih Hapus.
5. Saat diminta konfirmasi, pilih Ya, Hapus.

Informasi Terkait

Sekarang, setelah Anda memiliki gagasan dasar tentang cara kerja fungsi Lambda@Edge, pelajari lebih lanjut dengan membaca hal berikut:

- [Lambda @Edge contoh fungsi](#)
- [Praktik Terbaik Desain Lambda @Edge](#)
- [Mengurangi Latensi dan Menggeser Komputasi ke Edge dengan Lambda @Edge](#)

Siapkan izin dan peran IAM untuk Lambda @Edge

Untuk mengonfigurasi Lambda @Edge, Anda harus memiliki izin dan peran IAM berikut untuk: AWS Lambda

- [Izin IAM](#) — Izin ini memungkinkan Anda membuat fungsi Lambda dan mengaitkannya dengan distribusi Anda. CloudFront
- [Peran eksekusi fungsi Lambda \(peran IAM\)](#) — Prinsipal layanan Lambda mengasumsikan peran ini untuk menjalankan fungsi Anda.
- [Peran terkait layanan untuk Lambda @Edge](#) — Peran terkait layanan memungkinkan spesifik untuk Layanan AWS mereplikasi fungsi Lambda ke dan mengaktifkan penggunaan file log. Wilayah AWS CloudWatch CloudFront

Izin IAM diperlukan untuk mengaitkan fungsi Lambda @Edge dengan distribusi CloudFront

Selain izin IAM yang Anda perlukan untuk Lambda, Anda memerlukan izin berikut untuk mengaitkan fungsi Lambda dengan distribusi: CloudFront

- `lambda:GetFunction`— Memberikan izin untuk mendapatkan informasi konfigurasi untuk fungsi Lambda Anda dan URL yang telah ditentukan sebelumnya untuk mengunduh file .zip yang berisi fungsi tersebut.
- `lambda:EnableReplication*`— Memberikan izin ke kebijakan sumber daya sehingga layanan replikasi Lambda bisa mendapatkan kode fungsi dan konfigurasi.
- `lambda:DisableReplication*`— Memberikan izin ke kebijakan sumber daya sehingga layanan replikasi Lambda dapat menghapus fungsi.

Important

Anda harus menambahkan tanda bintang (*) di akhir `lambda:EnableReplication*` dan `lambda:DisableReplication*` tindakan.

- Untuk sumber daya, tentukan ARN dari versi fungsi yang ingin Anda jalankan ketika suatu CloudFront peristiwa terjadi, seperti contoh berikut:

```
arn:aws:lambda:us-east-1:123456789012:function:TestFunction:2
```

- `iam:CreateServiceLinkedRole`— Memberikan izin untuk membuat peran terkait layanan yang digunakan Lambda @Edge untuk mereplikasi fungsi Lambda. CloudFront Setelah Anda mengonfigurasi Lambda @Edge untuk pertama kalinya, peran terkait layanan akan dibuat secara otomatis untuk Anda. Anda tidak perlu menambahkan izin ini ke distribusi lain yang menggunakan Lambda @Edge.
- `cloudfront:UpdateDistribution` atau `cloudfront:CreateDistribution` — Memberikan izin untuk memperbarui atau membuat distribusi.

Untuk informasi selengkapnya, lihat topik berikut:

- [Identity and Access Management untuk Amazon CloudFront](#)
- Izin [akses sumber daya Lambda di Panduan Pengembang](#) AWS Lambda

Peran eksekusi fungsi untuk prinsipal layanan

Anda harus membuat peran IAM yang dapat diasumsikan oleh kepala sekolah `lambda.amazonaws.com` dan `edgelambda.amazonaws.com` layanan ketika mereka menjalankan fungsi Anda.

Tip

Saat membuat fungsi di konsol Lambda, Anda dapat memilih untuk membuat peran eksekusi baru dengan menggunakan templat AWS kebijakan. Langkah ini secara otomatis menambahkan izin Lambda @Edge yang diperlukan untuk menjalankan fungsi Anda. Lihat [Langkah 5 dalam Tutorial: Membuat fungsi Lambda @Edge sederhana](#).

Untuk informasi selengkapnya tentang membuat peran IAM secara manual, lihat [Membuat peran dan melampirkan kebijakan \(konsol\)](#) di Panduan Pengguna IAM.

Example Contoh: Kebijakan kepercayaan peran

Anda dapat menambahkan peran ini di bawah tab Trust Relationship di konsol IAM. Jangan tambahkan kebijakan ini di bawah tab Izin.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": [
          "lambda.amazonaws.com",
          "edgelambda.amazonaws.com"
        ]
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

Untuk informasi selengkapnya tentang izin yang perlu Anda berikan ke peran eksekusi, lihat [Izin akses sumber daya Lambda](#) di AWS Lambda Panduan Pengembang.

Catatan

- Secara default, setiap kali CloudFront peristiwa memicu fungsi Lambda, data ditulis CloudWatch ke Log. Jika Anda ingin menggunakan log ini, peran eksekusi memerlukan izin untuk menulis data ke CloudWatch Log. Anda dapat menggunakan standar AWSLambdaBasicExecutionRole untuk memberikan izin ke peran eksekusi.

Untuk informasi selengkapnya tentang CloudWatch Log, lihat [the section called “Log fungsi tepi”](#).

- Jika kode fungsi Lambda Anda mengakses AWS sumber daya lain, seperti membaca objek dari bucket S3, peran eksekusi memerlukan izin untuk melakukan tindakan tersebut.

Peran terkait layanan untuk Lambda @Edge

[Lambda @Edge menggunakan peran terkait layanan IAM](#). Peran yang terhubung dengan layanan adalah jenis peran IAM unik yang terhubung langsung ke layanan. Peran yang ditautkan dengan layanan ditentukan sebelumnya oleh layanan dan mencakup semua izin yang diperlukan layanan untuk menghubungi layanan AWS lainnya atas nama Anda.

Lambda @Edge menggunakan peran terkait layanan IAM berikut:

- **AWSServiceRoleForLambdaReplicator**— Lambda @Edge menggunakan peran ini untuk memungkinkan Lambda @Edge mereplikasi fungsi ke Wilayah AWS

Saat Anda pertama kali menambahkan pemicu Lambda @Edge CloudFront, peran bernama dibuat **AWSServiceRoleForLambdaReplicator** secara otomatis untuk memungkinkan Lambda @Edge mereplikasi fungsi. Wilayah AWS Peran ini diperlukan untuk menggunakan fungsi Lambda @Edge. ARN untuk **AWSServiceRoleForLambdaReplicator** peran tersebut terlihat seperti contoh berikut:

```
arn:aws:iam::123456789012:role/aws-service-role/replicator.lambda.amazonaws.com/AWSServiceRoleForLambdaReplicator
```

- **AWSServiceRoleForCloudFrontLogger**— CloudFront menggunakan peran ini untuk mendorong file log ke dalam CloudWatch. Anda dapat menggunakan file log untuk men-debug kesalahan validasi Lambda @Edge.

AWSServiceRoleForCloudFrontLoggerPeran dibuat secara otomatis saat Anda menambahkan asosiasi fungsi Lambda @Edge CloudFront untuk memungkinkan mendorong file log kesalahan Lambda @Edge ke CloudWatch ARN untuk AWSServiceRoleForCloudFrontLogger peran yang terlihat seperti ini:

```
arn:aws:iam::account_number:role/aws-service-role/  
logger.cloudfront.amazonaws.com/AWSServiceRoleForCloudFrontLogger
```

Peran yang terhubung dengan layanan memudahkan pengaturan dan penggunaan Lambda@Edge karena Anda tidak perlu menambahkan izin yang diperlukan secara manual. Lambda@Edge mendefinisikan izin peran yang terhubung ke layanan, dan hanya Lambda@Edge yang dapat memegang peran tersebut. Izin yang ditentukan mencakup kebijakan kepercayaan dan kebijakan izin. Kebijakan izin tidak dapat dilampirkan ke entitas IAM lainnya.

Anda harus menghapus sumber daya terkait CloudFront atau Lambda @Edge sebelum dapat menghapus peran terkait layanan. Ini membantu melindungi sumber daya Lambda @Edge Anda sehingga Anda tidak menghapus peran terkait layanan yang masih diperlukan untuk mengakses sumber daya aktif.

Untuk mengetahui informasi selengkapnya tentang peran terkait layanan, lihat [Peran terkait layanan untuk CloudFront](#).

Izin peran terkait layanan untuk Lambda @Edge

Lambda @Edge menggunakan dua peran terkait layanan, bernama dan.

AWSServiceRoleForLambdaReplicatorAWSServiceRoleForCloudFrontLogger Bagian berikut menjelaskan izin untuk masing-masing peran ini.

Daftar Isi

- [Izin peran terkait layanan untuk replikator Lambda](#)
- [Izin peran terkait layanan untuk logger CloudFront](#)

Izin peran terkait layanan untuk replikator Lambda

Peran terkait layanan ini memungkinkan Lambda mereplikasi fungsi Lambda @Edge. Wilayah AWS

Peran terkait layanan AWSServiceRoleForLambdaReplicator memercayai layanan `replicator.lambda.amazonaws.com` untuk menjalankan peran.

Kebijakan izin peran memungkinkan Lambda@Edge menyelesaikan tindakan berikut pada sumber daya yang ditentukan:

- `lambda:CreateFunction` pada `arn:aws:lambda:*:*:function:*`
- `lambda>DeleteFunction` pada `arn:aws:lambda:*:*:function:*`
- `lambda:DisableReplication` pada `arn:aws:lambda:*:*:function:*`
- `iam:PassRole` pada all AWS resources
- `cloudfront:ListDistributionsByLambdaFunction` pada all AWS resources

Izin peran terkait layanan untuk logger CloudFront

Peran terkait layanan ini memungkinkan CloudFront untuk mendorong file log ke dalam CloudWatch sehingga Anda dapat men-debug kesalahan validasi Lambda @Edge.

Peran terkait layanan `AWSServiceRoleForCloudFrontLogger` memercayai layanan `logger.cloudfront.amazonaws.com` untuk menjalankan peran.

Kebijakan izin peran memungkinkan Lambda @Edge menyelesaikan tindakan berikut pada sumber daya yang ditentukan: `arn:aws:logs:*:*:log-group:/aws/cloudfront/*`

- `logs:CreateLogGroup`
- `logs:CreateLogStream`
- `logs:PutLogEvents`

Anda harus mengonfigurasi izin untuk mengizinkan entitas IAM (seperti pengguna, grup, atau peran) untuk menghapus peran yang ditautkan oleh layanan Lambda@Edge. Untuk informasi selengkapnya, lihat [Izin peran terkait layanan](#) dalam Panduan Pengguna IAM.

Membuat peran terkait layanan untuk Lambda @Edge

Anda biasanya tidak membuat peran terkait layanan secara manual untuk Lambda@Edge. Layanan ini membuat peran untuk Anda secara otomatis dalam skenario berikut:

- Saat pertama kali membuat pemicu, layanan akan membuat `AWSServiceRoleForLambdaReplicator` peran (jika belum ada). Peran ini memungkinkan Lambda untuk mereplikasi fungsi Lambda @Edge ke Wilayah AWS

Jika Anda menghapus peran layanan yang ditautkan, peran tersebut akan dibuat lagi saat Anda menambahkan pemicu baru untuk Lambda@Edge dalam distribusi.

- Saat Anda memperbarui atau membuat CloudFront distribusi yang memiliki asosiasi Lambda @Edge, layanan akan membuat AWSServiceRoleForCloudFrontLogger peran (jika peran tersebut belum ada). Peran ini memungkinkan CloudFront untuk mendorong file log Anda ke CloudWatch.

Jika Anda menghapus peran terkait layanan, peran akan dibuat lagi saat Anda memperbarui atau membuat CloudFront distribusi yang memiliki asosiasi Lambda @Edge.

Untuk membuat peran terkait layanan ini secara manual, Anda dapat menjalankan perintah AWS Command Line Interface (AWS CLI) berikut:

Untuk membuat AWSServiceRoleForLambdaReplicator peran

- Jalankan perintah berikut.

```
aws iam create-service-linked-role --aws-service-name
replicator.lambda.amazonaws.com
```

Untuk membuat AWSServiceRoleForCloudFrontLogger peran

- Jalankan perintah berikut.

```
aws iam create-service-linked-role --aws-service-name
logger.cloudfront.amazonaws.com
```

Mengedit peran terkait layanan Lambda @Edge

Lambda @Edge tidak mengizinkan Anda mengedit AWSServiceRoleForLambdaReplicator atau peran yang ditautkan AWSServiceRoleForCloudFrontLogger layanan. Setelah layanan membuat peran terkait layanan, Anda tidak dapat mengubah nama peran karena berbagai entitas mungkin mereferensikan peran tersebut. Namun, Anda dapat menggunakan IAM untuk mengedit deskripsi peran. Untuk informasi selengkapnya, lihat [Mengedit peran terkait layanan](#) dalam Panduan Pengguna IAM.

Didukung Wilayah AWS untuk peran terkait layanan Lambda @Edge

CloudFront mendukung penggunaan peran terkait layanan untuk Lambda @Edge sebagai berikut:

Wilayah AWS

- AS Timur (Virginia Utara)–us-east-1
- AS Timur (Ohio)–us-east-2
- AS Barat (California Utara)–us-west-1
- AS Barat (Oregon)–us-west-2
- Asia Pasifik (Mumbai)–ap-south-1
- Asia Pasifik (Seoul)–ap-northeast-2
- Asia Pasifik (Singapura)–ap-southeast-1
- Asia Pasifik (Sydney)–ap-southeast-2
- Asia Pasifik (Tokyo) – ap-northeast-1
- Eropa (Frankfurt) – eu-central-1
- Eropa (Irlandia)–eu-west-1
- Eropa (London) – eu-west-2
- Amerika Selatan (São Paulo) – sa-east-1

Tulis dan buat fungsi Lambda @Edge

Untuk menggunakan Lambda @Edge, Anda menulis kode untuk fungsi Anda AWS Lambda . Untuk membantu Anda menulis fungsi Lambda @Edge, lihat sumber daya berikut:

- [Struktur acara Lambda @Edge](#)— Memahami struktur acara yang akan digunakan dengan Lambda @Edge.
- [Lambda @Edge contoh fungsi](#)— Contoh fungsi, seperti A/B pengujian dan menghasilkan pengalihan HTTP.

Model pemrograman untuk menggunakan Node.js atau Python dengan Lambda @Edge sama dengan menggunakan Lambda dalam file. Wilayah AWS Untuk informasi selengkapnya, lihat [Membangun fungsi Lambda dengan Node.js](#) atau [Membangun fungsi Lambda dengan Python](#) di Panduan Pengembang AWS Lambda

Dalam fungsi Lambda @Edge Anda, sertakan `callback` parameter dan kembalikan objek yang berlaku untuk peristiwa permintaan atau respons:

- Minta acara – Sertakan `cf.request` keberatan dalam respons.

Jika Anda menghasilkan respons, sertakan `cf.response` keberatan dalam respons. Untuk informasi selengkapnya, lihat [Hasilkan respons HTTP dalam pemicu permintaan](#).

- Peristiwa tanggapan – Sertakan `cf.response` keberatan dalam respons.

Setelah Anda menulis kode Anda sendiri atau menggunakan salah satu contoh, Anda kemudian membuat fungsi di Lambda. Untuk membuat fungsi atau mengedit yang sudah ada, lihat topik berikut:

Topik

- [Buat fungsi Lambda @Edge](#)
- [Edit fungsi Lambda](#)

Setelah Anda membuat fungsi di Lambda, Anda mengatur Lambda untuk menjalankan fungsi berdasarkan CloudFront peristiwa tertentu, yang disebut pemicu. Untuk informasi selengkapnya, lihat [Tambahkan pemicu untuk fungsi Lambda @Edge](#).

Buat fungsi Lambda @Edge

AWS Lambda Untuk mengatur menjalankan fungsi Lambda yang didasarkan pada CloudFront peristiwa, ikuti prosedur ini.

Untuk membuat fungsi Lambda@Edge

1. Masuk ke Konsol Manajemen AWS dan buka AWS Lambda konsol di <https://console.aws.amazon.com/lambda/>.
2. Jika Anda sudah memiliki satu atau beberapa fungsi Lambda, pilih Buat fungsi.

Jika Anda tidak memiliki fungsi apa pun, pilih Mulai Sekarang.

3. Pada daftar Wilayah di bagian atas halaman, pilih AS Timur (N. Virginia).
4. Buat fungsi menggunakan kode Anda sendiri atau buat fungsi dimulai dengan CloudFront cetak biru.
 - Untuk membuat fungsi menggunakan kode Anda sendiri, pilih Penulis dari awal.

- Untuk menampilkan daftar cetak biru CloudFront, masukkan cloudfront di bidang filter, lalu pilih Enter.

Jika Anda menemukan cetak biru yang ingin digunakan, pilih nama cetak biru tersebut.

5. Di Informasi dasar , tentukan nilai-nilai berikut:

- a. Nama — Masukkan nama untuk fungsi Anda.
- b. Peran — Untuk memulai dengan cepat, pilih Buat peran baru dari templat. Anda juga dapat memilih Pilih peran yang ada atau Buat peran khusus, lalu ikuti petunjuk untuk melengkapi informasi untuk bagian ini.
- c. Nama peran — Masukkan nama untuk peran tersebut.
- d. Templat kebijakan — Pilih izin Lambda Edge Dasar.

6. Jika Anda memilih Penulis dari awal dalam langkah 4, lompat ke langkah 7.

Jika Anda memilih cetak biru di langkah 4, bagian cloudfront memungkinkan Anda membuat satu pemicu, yang mengaitkan fungsi ini dengan cache dalam distribusi dan peristiwa. CloudFront Kami sarankan Anda memilih Hapus pada titik ini, jadi tidak ada pemicu untuk fungsi saat dibuat. Kemudian Anda dapat menambahkan pemicu nanti.

 Tip

Kami menyarankan Anda menguji dan men-debug fungsi sebelum menambahkan pemicu. Jika Anda menambahkan pemicu sekarang, fungsi akan berjalan segera setelah Anda membuat fungsi dan selesai mereplikasi ke AWS lokasi di seluruh dunia, dan distribusi yang sesuai diterapkan.

7. Pilih Buat fungsi.

Lambda membuat dua versi fungsi Anda: \$LATEST dan Versi 1. Anda dapat mengedit versi \$LATEST saja, tetapi konsol awalnya menampilkan Versi 1.

8. Untuk mengedit fungsi, pilih Versi 1 di dekat bagian atas halaman, di bawah ARN untuk fungsi . Lalu, pada Versi pilih, pilih \$LATEST. (Jika Anda meninggalkan fungsi dan kemudian kembali, label tombol adalah Pengukur.)
9. Di Konfigurasi pilih tab yang sesuai Jenis entri kode. Kemudian ikuti perintah untuk mengedit atau mengunggah kode Anda.
10. Untuk Waktu pengoperasian, pilih nilai berdasarkan kode fungsi Anda.

11. Di Tanda , tambahkan tag yang berlaku.
12. Pilih Tindakan, lalu pilih Terbitkan versi baru.
13. Masukkan deskripsi untuk versi baru fungsi.
14. Pilih Terbitkan.
15. Uji dan jalankan debug fungsinya. Untuk informasi selengkapnya tentang pengujian di konsol Lambda, lihat [Memanggil fungsi Lambda menggunakan konsol di Panduan Pengembang.AWS Lambda](#)
16. Saat Anda siap menjalankan fungsi untuk CloudFront acara, publikasikan versi lain dan edit fungsi untuk menambahkan pemicu. Untuk informasi selengkapnya, lihat [Tambahkan pemicu untuk fungsi Lambda @Edge](#).

Edit fungsi Lambda

Setelah Anda membuat fungsi Lambda @Edge, Anda dapat menggunakan konsol Lambda untuk mengeditnya.

Catatan

- Versi asli diberi label \$LATEST.
- Anda dapat mengedit versi \$LATEST saja.
- Setiap kali Anda mengedit versi \$LATEST, Anda harus menerbitkan versi bernomor baru.
- Anda tidak dapat membuat pemicu untuk \$LATEST.
- Saat Anda menerbitkan versi baru suatu fungsi, Lambda tidak otomatis menyalin pemicu dari versi sebelumnya ke versi baru. Anda harus mereproduksi pemicu untuk versi baru.
- Saat Anda menambahkan pemicu untuk suatu CloudFront peristiwa ke suatu fungsi, jika sudah ada pemicu untuk distribusi, perilaku cache, dan peristiwa yang sama untuk versi sebelumnya dari fungsi yang sama, Lambda menghapus pemicu dari versi sebelumnya.
- Setelah Anda membuat pembaruan pada CloudFront distribusi, seperti menambahkan pemicu, Anda harus menunggu perubahan menyebar ke lokasi tepi sebelum fungsi yang Anda tentukan dalam pemicu akan berfungsi.

Untuk mengedit fungsi Lambda

1. Masuk ke Konsol Manajemen AWS dan buka AWS Lambda konsol di <https://console.aws.amazon.com/lambda/>.
2. Pada daftar Wilayah di bagian atas halaman, pilih AS Timur (N. Virginia).
3. Dalam daftar fungsi, pilih nama fungsi.

Secara default, konsol menampilkan versi \$LATEST. Anda dapat melihat versi sebelumnya (pilih Pengukur), tetapi Anda hanya dapat mengedit \$LATEST.

4. Di Kode , untuk Jenis entri kode, pilih untuk mengedit kode di browser, mengunggah file .zip, atau mengunggah file dari Amazon S3.
5. Pilih salah satu Simpan atau Simpan dan uji.
6. Pilih Tindakan, dan pilih Terbitkan versi baru.
7. Di Terbitkan versi baru dari \$LATEST kotak dialog, masukkan deskripsi versi baru. Uraian ini muncul dalam daftar versi, bersama dengan nomor versi yang dibuat secara otomatis.
8. Pilih Terbitkan.

Versi baru secara otomatis menjadi versi terbaru. Nomor versi muncul pada Versi di sudut kiri atas halaman.

Note

Jika Anda belum menambahkan pemicu untuk fungsi Anda, lihat [Tambahkan pemicu untuk fungsi Lambda @Edge](#).

9. Pilih Pemicu tab.
10. Pilih Tambahkan pemicu.
11. Di Tambahkan pemicu kotak dialog, pilih kotak titik-titik, lalu pilih CloudFront.

Note

Jika Anda telah membuat satu atau beberapa pemicu untuk suatu fungsi, CloudFront adalah layanan default.

12. Tentukan nilai berikut untuk menunjukkan kapan Anda ingin fungsi Lambda menjalankan.
 - a. ID Distribusi — Pilih ID distribusi yang ingin Anda tambahkan pemicu.

- b. Perilaku cache - Pilih perilaku cache yang menentukan objek yang ingin Anda jalankan fungsinya.
- c. CloudFront event — Pilih CloudFront acara yang menyebabkan fungsi dijalankan.
- d. Aktifkan pemicu dan replikasi - Pilih kotak centang ini sehingga Lambda mereplikasi fungsi secara global. Wilayah AWS

13. Pilih Kirim.

14. Untuk menambahkan lebih banyak pemicu untuk fungsi ini, ulangi langkah 10 hingga 13.

Untuk informasi selengkapnya tentang menguji dan men-debug fungsi di konsol Lambda, [lihat Memanggil fungsi Lambda menggunakan](#) konsol di Panduan Pengembang.AWS Lambda

Saat Anda siap menjalankan fungsi untuk CloudFront acara, publikasikan versi lain dan edit fungsi untuk menambahkan pemicu. Untuk informasi selengkapnya, lihat [Tambahkan pemicu untuk fungsi Lambda @Edge](#).

Tambahkan pemicu untuk fungsi Lambda @Edge

Pemicu Lambda @Edge adalah salah satu kombinasi dari CloudFront distribusi, perilaku cache, dan peristiwa yang menyebabkan fungsi dijalankan. Misalnya, Anda dapat membuat pemicu yang menyebabkan fungsi dijalankan saat CloudFront menerima permintaan dari penampil untuk perilaku cache tertentu yang Anda siapkan untuk distribusi Anda. Anda dapat menentukan satu atau lebih CloudFront pemicu.

Tip

Saat membuat CloudFront distribusi, Anda menentukan setelan yang memberi tahu CloudFront cara merespons saat menerima permintaan yang berbeda. Pengaturan default disebut perilaku cache default untuk distribusi. Anda dapat mengatur perilaku cache tambahan yang menentukan cara CloudFront merespons dalam keadaan tertentu, misalnya, saat menerima permintaan untuk jenis file tertentu. Untuk informasi selengkapnya, lihat [Pengaturan perilaku cache](#).

Saat pertama kali membuat fungsi Lambda, Anda hanya dapat menentukan satu pemicu. Anda dapat menambahkan lebih banyak pemicu ke fungsi yang sama nanti dengan menggunakan konsol Lambda atau dengan mengedit distribusi di CloudFront konsol.

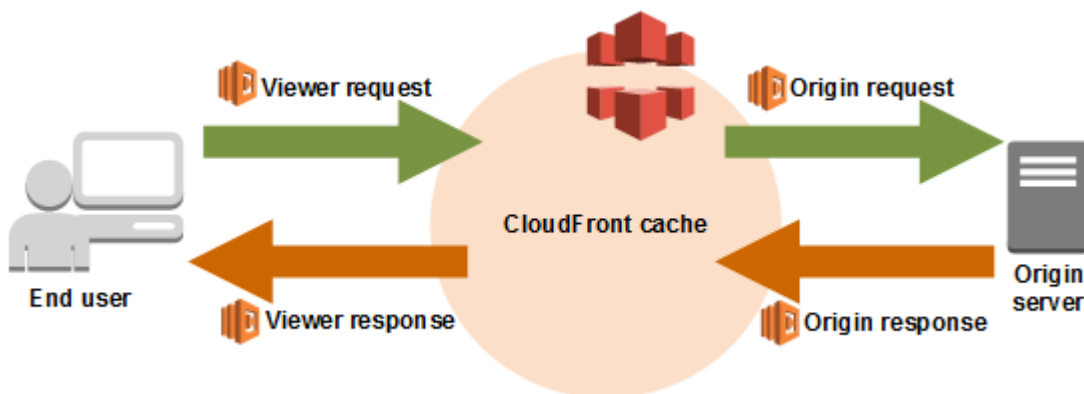
- Konsol Lambda berfungsi dengan baik jika Anda ingin menambahkan lebih banyak pemicu ke fungsi untuk distribusi yang sama. CloudFront
- CloudFront Konsol bisa lebih baik jika Anda ingin menambahkan pemicu untuk beberapa distribusi karena lebih mudah untuk menemukan distribusi yang ingin Anda perbarui. Anda juga dapat memperbarui CloudFront pengaturan lain secara bersamaan.

Topik

- [CloudFront peristiwa yang dapat memicu fungsi Lambda @Edge](#)
- [Pilih acara untuk memicu fungsi](#)
- [Tambahkan pemicu ke fungsi Lambda @Edge](#)

CloudFront peristiwa yang dapat memicu fungsi Lambda @Edge

Untuk setiap perilaku cache dalam CloudFront distribusi Amazon, Anda dapat menambahkan hingga empat pemicu (asosiasi) yang menyebabkan fungsi Lambda dijalankan saat peristiwa CloudFront tertentu terjadi. CloudFront pemicu dapat didasarkan pada salah satu dari empat CloudFront peristiwa, seperti yang ditunjukkan pada diagram berikut.



CloudFront Peristiwa yang dapat digunakan untuk memicu fungsi Lambda @Edge adalah sebagai berikut:

Permintaan penampil

Fungsi dijalankan ketika CloudFront menerima permintaan dari penampil, sebelum memeriksa untuk melihat apakah objek yang diminta ada dalam CloudFront cache.

Fungsi tidak dijalankan dalam kasus berikut:

- Saat mengambil halaman kesalahan kustom.

- Ketika CloudFront secara otomatis mengalihkan permintaan HTTP ke HTTPS (ketika nilai [Kebijakan protokol penampil](#) adalah Redirect HTTP ke HTTPS).

Permintaan asal

Fungsi dijalankan hanya ketika CloudFront meneruskan permintaan ke asal Anda. Ketika objek yang diminta ada di CloudFront cache, fungsi tidak dijalankan.

Respons asal

Fungsi mengeksekusi setelah CloudFront menerima respon dari asal dan sebelum cache objek dalam respon. Perhatikan bahwa fungsi menjalankan bahkan jika kesalahan dikembalikan dari asal.

Fungsi tidak dijalankan dalam kasus berikut:

- Ketika file yang diminta dalam CloudFront cache dan tidak kedaluwarsa.
- Saat respons dihasilkan dari fungsi yang dipicu oleh peristiwa permintaan asal.

Respons penampil

Fungsi menjalankan sebelum mengembalikan file yang diminta ke penampil. Perhatikan bahwa fungsi dijalankan terlepas dari apakah file sudah dalam CloudFront cache.

Fungsi tidak dijalankan dalam kasus berikut:

- Saat asal mengembalikan kode status HTTP sebesar 400 atau lebih.
- Saat halaman kesalahan kustom dikembalikan.
- Saat respons dihasilkan dari fungsi yang dipicu oleh peristiwa permintaan penampil.
- Ketika CloudFront secara otomatis mengalihkan permintaan HTTP ke HTTPS (ketika nilai [Kebijakan protokol penampil](#) adalah Redirect HTTP ke HTTPS).

Saat Anda menambahkan beberapa pemicu ke perilaku cache yang sama, Anda dapat menggunakannya untuk menjalankan fungsi yang sama atau menjalankan fungsi yang berbeda untuk setiap pemicu. Anda juga dapat mengaitkan fungsi yang sama dengan lebih dari satu distribusi.

Note

Ketika sebuah CloudFront peristiwa memicu eksekusi fungsi Lambda, fungsi harus selesai CloudFront sebelum dapat melanjutkan.

Misalnya, jika fungsi Lambda dipicu oleh peristiwa permintaan CloudFront penampil, tidak CloudFront akan mengembalikan respons ke penampil atau meneruskan permintaan ke asal hingga fungsi Lambda selesai berjalan.

Ini berarti bahwa setiap permintaan yang memicu fungsi Lambda meningkatkan latensi untuk permintaan, jadi Anda ingin fungsi tersebut dijalankan secepat mungkin.

Pilih acara untuk memicu fungsi

Saat Anda memutuskan CloudFront acara mana yang ingin Anda gunakan untuk memicu fungsi Lambda, pertimbangkan hal berikut:

Saya CloudFront ingin menyimpan objek yang diubah oleh fungsi Lambda

Untuk cache objek yang telah dimodifikasi oleh fungsi Lambda sehingga CloudFront dapat melayani objek dari lokasi tepi saat diminta berikutnya, gunakan permintaan asal atau peristiwa respons asal.

Ini mengurangi beban dari awal, mengurangi latensi untuk permintaan berikutnya, dan mengurangi biaya dari invoking Lambda@Edge pada permintaan berikutnya.

Misalnya, jika Anda ingin menambahkan, menghapus, atau mengubah header untuk objek yang dikembalikan oleh asal dan Anda CloudFront ingin menyimpan hasil cache, gunakan peristiwa respons asal.

Saya ingin fungsi dijalankan untuk setiap permintaan

Untuk menjalankan fungsi untuk setiap permintaan yang CloudFront diterima untuk distribusi, gunakan permintaan penampil atau peristiwa respons penampil.

Permintaan asal dan kejadian respons asal hanya terjadi ketika objek yang diminta tidak di-cache di lokasi tepi dan CloudFront meneruskan permintaan ke asal.

Saya ingin fungsinya mengubah kunci cache

Untuk mengubah nilai yang Anda gunakan sebagai dasar untuk caching, gunakan acara permintaan penampil.

Misalnya, jika fungsi mengubah URL untuk menyertakan singkatan bahasa pada jalur (misalnya, karena pengguna memilih bahasa dari daftar tarik turun), gunakan acara permintaan penampil:

- URL dalam permintaan penampil - `https://example.com/en/ index.html`
- URL ketika permintaan berasal dari alamat IP di Jerman `https://example.com/de/ - index.html`

Anda juga menggunakan acara permintaan pemirsa jika Anda menyimpan cache berdasarkan cookie atau header permintaan.

 Note

Jika fungsi mengubah cookie atau header, konfigurasi CloudFront untuk meneruskan bagian permintaan yang berlaku ke asal. Untuk informasi selengkapnya, lihat topik berikut:

- [Konten cache berdasarkan cookie](#)
- [Konten cache berdasarkan header permintaan](#)

Fungsi mempengaruhi respon dari asal

Untuk mengubah permintaan dengan cara yang memengaruhi respons dari asal, gunakan peristiwa permintaan asal.

Biasanya, sebagian besar peristiwa permintaan pemirsa tidak diteruskan ke asal. CloudFront menanggapi permintaan dengan objek yang sudah ada di cache tepi. Jika fungsi mengubah permintaan berdasarkan peristiwa permintaan asal, CloudFront cache respons ke permintaan asal yang diubah.

Tambahkan pemicu ke fungsi Lambda @Edge

Anda dapat menggunakan AWS Lambda konsol atau CloudFront konsol Amazon untuk menambahkan pemicu ke fungsi Lambda @Edge Anda.

 Important

Anda dapat membuat pemicu hanya untuk versi bernomor dari fungsi Anda (bukan \$LATEST).

Lambda console

Untuk menambahkan pemicu CloudFront acara ke fungsi Lambda @Edge

1. Masuk ke Konsol Manajemen AWS dan buka AWS Lambda konsol di <https://console.aws.amazon.com/lambda/>.
2. Pada daftar Wilayah di bagian atas halaman, pilih AS Timur (N. Virginia).

3. Di Fungsi , pilih nama fungsi yang ingin Anda tambahkan pemicunya.
4. Pada halaman Ikhtisar fungsi, pilih tab Versi.
5. Pilih versi yang ingin Anda tambahkan pemicu.

Setelah memilih versi, nama tombol berubah menjadi Versi: \$LATEST atau Versi: nomor versi.

6. Pilih Pemicu tab.
7. Pilih Tambahkan pemicu.
8. Untuk konfigurasi Trigger, pilih Pilih sumber **cloudfront**, masukkan, lalu pilih CloudFront.

 Note

Jika Anda sudah membuat satu atau lebih pemicu, CloudFront adalah layanan default.

9. Tentukan nilai berikut untuk menunjukkan kapan Anda ingin fungsi Lambda menjalankan.
 - a. Distribusi - Pilih distribusi yang ingin Anda tambahkan pemicu.
 - b. Perilaku cache - Pilih perilaku cache yang menentukan objek yang ingin Anda jalankan fungsinya.

 Note

Jika Anda menentukan * untuk perilaku cache, fungsi Lambda menerapkan perilaku cache default.

- c. CloudFront event — Pilih CloudFront acara yang menyebabkan fungsi dijalankan.
 - d. Sertakan isi - Pilih kotak centang ini jika Anda ingin mengakses badan permintaan dalam fungsi Anda.
 - e. Konfirmasikan penerapan ke Lambda @Edge — Pilih kotak centang ini AWS Lambda sehingga mereplikasi fungsi ke global. Wilayah AWS
10. Pilih Tambahkan.

Fungsi mulai memproses permintaan untuk CloudFront peristiwa yang ditentukan saat CloudFront distribusi yang diperbarui diterapkan. Untuk menentukan apakah distribusi diterapkan, pilih Distribusi dalam panel navigasi. Ketika distribusi diterapkan, nilai kolom Status untuk distribusi berubah dari Deploying ke tanggal dan waktu penerapan.

CloudFront console

Untuk menambahkan pemicu CloudFront acara ke fungsi Lambda @Edge

1. Dapatkan ARN dari fungsi Lambda yang ingin Anda tambahkan pemicu untuk:
 - a. Masuk ke Konsol Manajemen AWS dan buka AWS Lambda konsol di <https://console.aws.amazon.com/lambda/>.
 - b. Pada daftar Wilayah di bagian atas halaman, pilih AS Timur (N. Virginia).
 - c. Dalam daftar fungsi, pilih nama fungsi yang ingin Anda tambahkan pemicunya.
 - d. Pada halaman Ikhtisar fungsi, pilih tab Versi, dan pilih versi bernomor yang ingin Anda tambahkan pemicu.
 - e. Pilih Salin ARN tombol untuk menyalin ARN ke clipboard Anda. ARN untuk fungsi Lambda terlihat seperti ini:

```
arn:aws:lambda:us-east-1:123456789012:function:TestFunction:2
```

Nomor di bagian akhir (2 dalam contoh ini) adalah nomor versi fungsi.

2. Buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
3. Dalam daftar distribusi, pilih ID distribusi yang ingin Anda tambahkan pemicunya.
4. Pilih Perilaku tab.
5. Pilih perilaku cache yang ingin Anda tambahkan pemicu, lalu pilih Edit.
6. Untuk asosiasi Fungsi, dalam daftar Jenis fungsi, pilih Lambda @Edge untuk saat Anda ingin menjalankan fungsi: untuk permintaan penampil, respons penampil, permintaan asal, atau respons asal.

Untuk informasi selengkapnya, lihat [Pilih acara untuk memicu fungsi](#).

7. Di kotak teks Fungsi ARN>Nama, tempel ARN dari fungsi Lambda yang ingin Anda jalankan saat acara yang dipilih terjadi. Ini adalah nilai yang Anda salin dari konsol Lambda.
8. Pilih Sertakan isi jika Anda ingin mengakses badan permintaan dalam fungsi Anda.

Jika Anda hanya ingin mengganti badan permintaan, Anda tidak perlu memilih opsi ini.

9. Untuk menjalankan fungsi yang sama untuk lebih banyak jenis acara, ulangi langkah 6 dan 7.
10. Pilih Simpan perubahan.
11. Untuk menambahkan pemicu ke lebih banyak perilaku cache untuk distribusi ini, ulangi langkah 5 hingga 10.

Fungsi mulai memproses permintaan untuk CloudFront peristiwa yang ditentukan saat CloudFront distribusi yang diperbarui diterapkan. Untuk menentukan apakah distribusi diterapkan, pilih Distribusi dalam panel navigasi. Saat distribusi diterapkan, nilai kolom Status untuk distribusi berubah dari Deploying ke waktu dan tanggal penerapan.

Uji dan debug fungsi Lambda @Edge

Penting untuk menguji kode fungsi Lambda @Edge Anda secara mandiri, untuk memastikan bahwa itu menyelesaikan tugas yang dimaksudkan, dan untuk melakukan pengujian integrasi, untuk memastikan bahwa fungsi berfungsi dengan benar. CloudFront

Selama pengujian integrasi atau setelah fungsi Anda di-deploy, Anda mungkin perlu men-debug CloudFront kesalahan, seperti kesalahan HTTP 5xx. Kesalahan dapat menjadi respons tidak valid yang dikembalikan dari fungsi Lambda, kesalahan eksekusi saat fungsi dipicu, atau kesalahan akibat perotasian eksekusi oleh layanan Lambda. Bagian-bagian dalam topik ini membagikan strategi untuk menentukan jenis kegagalan mana yang menjadi masalahnya, kemudian langkah-langkah yang dapat Anda ambil untuk memperbaiki masalah.

Note

Saat Anda meninjau file CloudWatch log atau metrik saat Anda memecahkan masalah kesalahan, ketahuilah bahwa kesalahan tersebut ditampilkan atau disimpan di lokasi Wilayah AWS terdekat dengan lokasi di mana fungsi dijalankan. Jadi, jika Anda memiliki situs web atau aplikasi web dengan pengguna di Britania Raya, dan Anda memiliki fungsi Lambda yang terkait dengan distribusi Anda, misalnya, Anda harus mengubah Wilayah untuk CloudWatch melihat metrik atau file log untuk London. Wilayah AWS Untuk informasi selengkapnya, lihat [the section called “Tentukan Wilayah Lambda @Edge”](#).

Topik

- [Uji fungsi Lambda @Edge Anda](#)
- [Identifikasi kesalahan fungsi Lambda @Edge di CloudFront](#)
- [Memecahkan masalah respons fungsi Lambda @Edge yang tidak valid \(kesalahan validasi\)](#)
- [Memecahkan masalah kesalahan eksekusi fungsi Lambda @Edge](#)
- [Tentukan Wilayah Lambda @Edge](#)

- [Tentukan apakah akun Anda mendorong log ke CloudWatch](#)

Uji fungsi Lambda @Edge Anda

Terdapat dua langkah untuk menguji fungsi Lambda Anda: pengujian mandiri dan pengujian integrasi.

Uji fungsionalitas mandiri

Sebelum Anda menambahkan fungsi Lambda CloudFront, pastikan untuk menguji fungsionalitas terlebih dahulu dengan menggunakan kemampuan pengujian di konsol Lambda atau dengan menggunakan metode lain. Untuk informasi selengkapnya tentang pengujian di konsol Lambda, lihat [Memanggil fungsi Lambda menggunakan konsol di Panduan Pengembang](#). AWS Lambda

Uji operasi fungsi Anda di CloudFront

Penting untuk menyelesaikan pengujian integrasi, di mana fungsi Anda dikaitkan dengan distribusi dan berjalan berdasarkan CloudFront peristiwa. Pastikan bahwa fungsi dipicu untuk acara yang tepat, dan mengembalikan respons yang valid dan benar untuk CloudFront. Misalnya, pastikan bahwa struktur acara sudah benar, bahwa hanya header yang valid yang disertakan, dan sebagainya.

Saat Anda mengulangi pengujian integrasi dengan fungsi Anda di konsol Lambda, lihat langkah-langkah dalam tutorial Lambda @Edge saat Anda memodifikasi kode atau mengubah CloudFront pemicu yang memanggil fungsi Anda. Misalnya, pastikan bahwa Anda bekerja dalam versi bernomor dari fungsi Anda, seperti yang dijelaskan dalam langkah tutorial ini: [Langkah 4: Tambahkan CloudFront pemicu untuk menjalankan fungsi](#).

Saat Anda membuat perubahan dan menerapkannya, ketahuilah bahwa fungsi dan CloudFront pemicu Anda yang diperbarui akan memakan waktu beberapa menit untuk mereplikasi di semua Wilayah. Ini biasanya memerlukan waktu beberapa menit, tetapi dapat memakan waktu hingga 15 menit.

Anda dapat memeriksa untuk melihat apakah replikasi selesai dengan pergi ke CloudFront konsol dan melihat distribusi Anda.

Untuk memeriksa apakah replikasi Anda telah selesai digunakan

1. Buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Pilih nama distribusi.

3. Periksa status distribusi yang akan diubah dari Sedang Berlangsung kembali ke Diterapkan, yang berarti fungsi Anda telah direplikasi. Kemudian ikuti langkah-langkah di bagian berikutnya untuk memverifikasi bahwa fungsi berfungsi.

Ketahui bahwa pengujian di konsol hanya memvalidasi logika fungsi Anda, dan tidak menerapkan kuota layanan apa pun (sebelumnya dikenal sebagai batas) yang khusus untuk Lambda @Edge.

Identifikasi kesalahan fungsi Lambda @Edge di CloudFront

Setelah Anda memverifikasi bahwa logika fungsi Anda berfungsi dengan benar, Anda mungkin masih melihat kesalahan HTTP 5xx saat fungsi Anda berjalan. CloudFront Kesalahan HTTP 5xx dapat dikembalikan karena berbagai alasan, yang dapat mencakup kesalahan fungsi Lambda atau masalah lain di dalamnya. CloudFront

- Jika Anda menggunakan fungsi Lambda @Edge, Anda dapat menggunakan grafik di CloudFront konsol untuk membantu melacak penyebab kesalahan, lalu bekerja untuk memperbaikinya. Misalnya, Anda dapat melihat apakah kesalahan HTTP 5xx disebabkan oleh CloudFront atau oleh fungsi Lambda, dan kemudian, untuk fungsi tertentu, Anda dapat melihat file log terkait untuk menyelidiki masalah tersebut.
- Untuk memecahkan masalah kesalahan HTTP secara umum di CloudFront, lihat langkah-langkah pemecahan masalah dalam topik berikut: [Memecahkan masalah kode status respons kesalahan di CloudFront](#)

Apa yang menyebabkan kesalahan fungsi Lambda @Edge di CloudFront

Ada beberapa alasan mengapa fungsi Lambda dapat menyebabkan kesalahan HTTP 5xx, dan langkah-langkah pemecahan masalah yang harus Anda ambil bergantung pada jenis kesalahan. Kesalahan dapat dikategorikan sebagai berikut:

Kesalahan eksekusi fungsi Lambda

Kesalahan eksekusi terjadi ketika CloudFront tidak mendapatkan respons dari Lambda karena ada pengecualian yang tidak tertangani dalam fungsi atau ada kesalahan dalam kode. Misalnya, jika kode menyertakan callback(Kesalahan).

Respons fungsi Lambda yang tidak valid dikembalikan ke CloudFront

Setelah fungsi berjalan, CloudFront menerima respons dari Lambda. Kesalahan dikembalikan jika struktur objek tanggapan tidak sesuai dengan [Struktur acara Lambda @Edge](#), atau respons berisi header yang tidak valid atau kolom tidak valid lainnya.

Eksekusi di CloudFront dibatasi karena kuota layanan Lambda (sebelumnya dikenal sebagai batas)

Eksekusi throttle layanan Lambda di setiap Wilayah, dan menghasilkan kesalahan jika Anda melebihi kuota. Untuk informasi selengkapnya, lihat [Kuotas di Lambda@Edge](#).

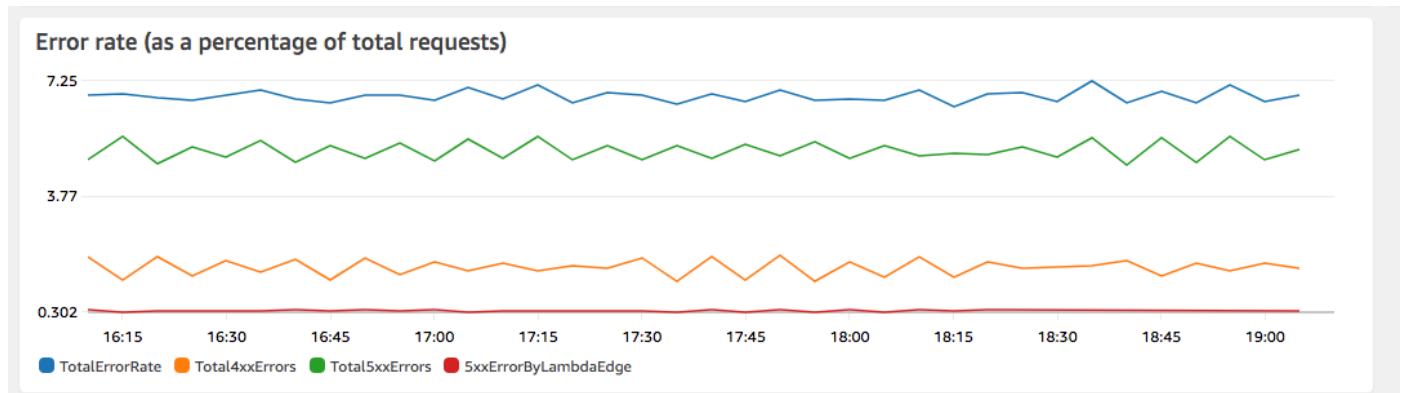
Cara menentukan jenis kegagalan

Untuk membantu Anda memutuskan di mana harus fokus saat Anda men-debug dan bekerja untuk menyelesaikan kesalahan yang dikembalikan oleh CloudFront, akan sangat membantu untuk mengidentifikasi CloudFront mengapa mengembalikan kesalahan HTTP. Untuk memulai, Anda dapat menggunakan grafik yang disediakan di bagian Pemantauan CloudFront konsol di Konsol Manajemen AWS. Untuk informasi selengkapnya tentang melihat grafik di bagian Pemantauan CloudFront konsol, lihat [Pantau CloudFront metrik dengan Amazon CloudWatch](#).

Grafik berikut akan sangat membantu ketika Anda ingin melacak apakah kesalahan dikembalikan oleh asal atau fungsi Lambda, dan untuk mempersempit jenis masalah ketika itu adalah kesalahan dari fungsi Lambda.

Grafik harga kesalahan

Salah satu grafik yang dapat Anda lihat pada Ikhtisar untuk setiap distribusi Anda adalah Tingkat kesalahan grafik. Grafik ini menampilkan tingkat kesalahan sebagai persentase dari total permintaan yang datang ke distribusi Anda. Grafik menunjukkan tingkat kesalahan total, total 4xx kesalahan, total 5xx kesalahan, dan total 5xx kesalahan dari fungsi Lambda. Berdasarkan jenis dan volume kesalahan, Anda dapat mengambil langkah untuk menyelidiki dan memecahkan masalah penyebab.



- Jika Anda melihat kesalahan Lambda, Anda dapat menyelidiki lebih lanjut dengan melihat jenis kesalahan tertentu yang dikembalikan oleh fungsi tersebut. Kesalahan Lambda@Edge tab menyertakan grafik yang mengategorikan kesalahan fungsi berdasarkan jenis untuk membantu Anda menemukan masalah dari fungsi tertentu.
- Jika Anda melihat CloudFront kesalahan, Anda dapat memecahkan masalah dan bekerja untuk memperbaiki kesalahan asal atau mengubah konfigurasi Anda CloudFront . Untuk informasi selengkapnya, lihat [Memecahkan masalah kode status respons kesalahan di CloudFront](#).

Grafik kesalahan pelaksanaan dan respons fungsi tidak valid

Kesalahan Lambda@Edge tab mencakup grafik yang mengkategorikan kesalahan Lambda@Edge untuk distribusi tertentu, berdasarkan jenis. Misalnya, satu grafik menunjukkan semua kesalahan eksekusi oleh Wilayah AWS.

Untuk mempermudah pemecahan masalah, Anda dapat mencari masalah tertentu dengan membuka dan memeriksa file log untuk fungsi tertentu berdasarkan Wilayah.

Untuk melihat file log untuk fungsi tertentu menurut Wilayah

1. Pada tab kesalahan Lambda @Edge, di bawah fungsi Lambda @Edge Terkait, pilih nama fungsi, lalu pilih Lihat metrik.
2. Selanjutnya, pada halaman dengan nama fungsi Anda, di sudut kanan atas, pilih Lihat log fungsi, lalu pilih Wilayah.

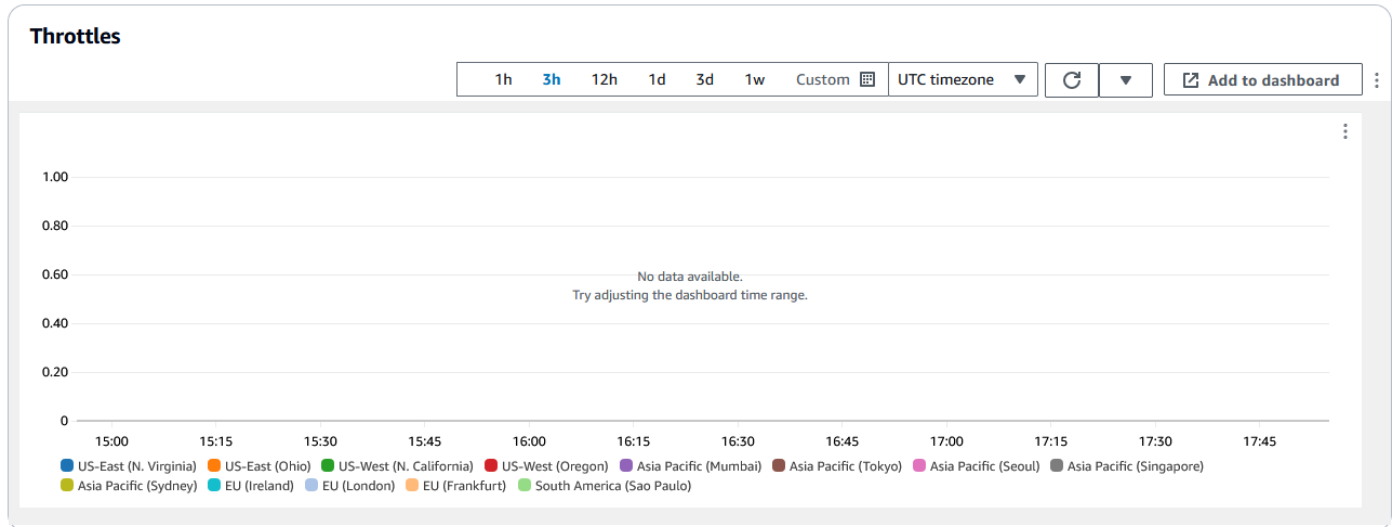
Misalnya, jika Anda melihat masalah dalam grafik Kesalahan untuk Wilayah AS Barat (Oregon), pilih Wilayah itu dari daftar tarik-turun. Ini membuka CloudWatch konsol Amazon.

3. Di CloudWatch konsol untuk Wilayah itu, di bawah Aliran log, pilih aliran log untuk melihat peristiwa fungsi tersebut.

Selain itu, baca bagian berikut dalam bab ini untuk rekomendasi lebih lanjut tentang pemecahan masalah dan memperbaiki kesalahan.

Grafik trotel

Kesalahan Lambda@Edge juga mencakup Trotel grafik. Terkadang, layanan Lambda merombak invokasi fungsi Anda dengan basis per Wilayah, jika Anda mencapai kuota konkurensi regional (sebelumnya disebut batas). Jika Anda melihat kesalahan yang melebihi, fungsi Anda telah mencapai kuota yang dikenakan layanan Lambda pada eksekusi di Wilayah. Untuk informasi lebih lanjut, termasuk cara meminta peningkatan kuota, lihat [Kuotas di Lambda@Edge](#).



Sebagai contoh tentang cara menggunakan informasi ini dalam mengatasi masalah kesalahan HTTP, lihat [Empat langkah untuk melakukan debug pengiriman konten Anda di AWS](#).

Memecahkan masalah respons fungsi Lambda @Edge yang tidak valid (kesalahan validasi)

Jika Anda mengidentifikasi bahwa masalah Anda adalah kesalahan validasi Lambda, itu berarti fungsi Lambda Anda mengembalikan respons yang tidak valid. CloudFront Ikuti panduan di bagian ini untuk mengambil langkah-langkah untuk meninjau fungsi Anda dan memastikan bahwa respons Anda sesuai dengan CloudFront persyaratan.

CloudFront memvalidasi respons dari fungsi Lambda dengan dua cara:

- Respon Lambda harus sesuai dengan struktur objek yang diperlukan. Contoh struktur objek yang buruk mencakup hal berikut: JSON yang tidak dapat dipisahkan, kolom wajib yang hilang, dan objek tidak valid dalam respons. Untuk informasi lebih lanjut, lihat [Struktur acara Lambda @Edge](#).

- Respons harus menyertakan hanya nilai objek yang valid. Kesalahan akan terjadi jika respons mencakup objek valid tetapi memiliki nilai yang tidak didukung. Contohnya meliputi yang berikut ini: menambahkan atau memperbarui header yang masuk daftar tidak diizinkan atau hanya baca (lihat [Pembatasan pada fungsi edge](#)), melebihi ukuran izi maksimum (lihat dalam Pembatasan Ukuran Respons yang Dihasilkan dalam topik [Kesalahan Lambda@Edge](#)) dan karakter atau nilai tidak valid (lihat [Struktur acara Lambda @Edge](#)).

Ketika Lambda mengembalikan respons yang tidak valid CloudFront, pesan kesalahan ditulis ke file log yang CloudFront mendorong ke CloudWatch Wilayah tempat fungsi Lambda dijalankan. Ini adalah perilaku default untuk mengirim file log CloudWatch ketika ada respons yang tidak valid. Namun, jika Anda mengaitkan fungsi Lambda CloudFront sebelum fungsionalitas dirilis, fungsi tersebut mungkin tidak diaktifkan untuk fungsi Anda. Untuk informasi lebih lanjut, lihat Tentukan apakah Akun Anda Mendorong Log ke CloudWatch nanti dalam topik.

CloudFront mendorong file log ke Wilayah yang sesuai dengan tempat fungsi Anda dijalankan, di grup log yang terkait dengan distribusi Anda. Grup log memiliki format berikut: `/aws/cloudfront/LambdaEdge/DistributionId`, di *DistributionId* mana ID distribusi Anda. Untuk menentukan Wilayah tempat Anda dapat menemukan file CloudWatch log, lihat Menentukan Wilayah Lambda @Edge nanti dalam topik ini.

Jika kesalahan dapat direproduksi, Anda dapat membuat permintaan baru yang menghasilkan kesalahan dan kemudian menemukan id permintaan dalam CloudFront respons gagal (`X-Amz-Cf-Idheader`) untuk menemukan satu kegagalan dalam file log. Entri file log mencakup informasi yang dapat membantu Anda mengidentifikasi mengapa kesalahan dikembalikan, dan juga mencantumkan id permintaan Lambda yang sesuai sehingga Anda dapat menganalisis akar masalah dalam konteks permintaan tunggal.

Jika kesalahan terputus-putus, Anda dapat menggunakan log CloudFront akses untuk menemukan id permintaan untuk permintaan yang gagal, dan kemudian mencari CloudWatch log untuk pesan kesalahan yang sesuai. Untuk informasi lebih lanjut, lihat bagian sebelumnya, Menentukan Jenis Kegagalan.

Memecahkan masalah kesalahan eksekusi fungsi Lambda @Edge

Jika masalahnya adalah kesalahan eksekusi Lambda, akan sangat membantu untuk membuat pernyataan logging untuk fungsi Lambda, untuk menulis pesan ke file CloudWatch log yang memantau eksekusi fungsi Anda CloudFront dan menentukan apakah berfungsi seperti yang

diharapkan. Kemudian Anda dapat mencari pernyataan tersebut di file CloudWatch log untuk memverifikasi bahwa fungsi Anda berfungsi.

Note

Bahkan jika Anda belum mengubah fungsi Lambda@Edge Anda, pembaruan pada lingkungan pelaksanaan fungsi Lambda dapat memengaruhinya dan dapat mengembalikan kesalahan pelaksanaan. Untuk informasi tentang pengujian dan migrasi ke versi yang lebih baru, lihat [Pembaruan mendatang untuk lingkungan eksekusi AWS Lambda dan AWS Lambda @Edge](#).

Tentukan Wilayah Lambda @Edge

Untuk melihat Wilayah tempat fungsi Lambda @Edge Anda menerima lalu lintas, lihat metrik untuk fungsi di CloudFront konsol di. Konsol Manajemen AWS Metrik ditampilkan untuk setiap AWS Wilayah. Di halaman yang sama, Anda dapat memilih Wilayah dan melihat file log untuk Wilayah tersebut sehingga Anda dapat menyelidiki masalah. Anda harus meninjau file CloudWatch log di AWS Wilayah yang benar untuk melihat file log yang dibuat saat CloudFront menjalankan fungsi Lambda Anda.

Untuk informasi selengkapnya tentang melihat grafik di bagian Pemantauan CloudFront konsol, lihat [Pantau CloudFront metrik dengan Amazon CloudWatch](#).

Tentukan apakah akun Anda mendorong log ke CloudWatch

Secara default, CloudFront memungkinkan pencatatan respons fungsi Lambda yang tidak valid, dan mendorong file log ke CloudWatch dengan menggunakan salah satu file. [Peran terkait layanan untuk Lambda @Edge](#) Jika Anda memiliki fungsi Lambda @Edge yang Anda tambahkan CloudFront sebelum fitur log respons fungsi Lambda yang tidak valid dirilis, logging diaktifkan saat Anda memperbarui konfigurasi Lambda @Edge Anda, misalnya, dengan menambahkan pemicu. CloudFront

Anda dapat memverifikasi bahwa mendorong file log ke CloudWatch diaktifkan untuk akun Anda dengan melakukan hal berikut:

- Periksa untuk melihat apakah log muncul CloudWatch — Pastikan Anda melihat di Wilayah tempat fungsi Lambda @Edge dijalankan. Untuk informasi selengkapnya, lihat [Tentukan Wilayah Lambda @Edge](#).

- Tentukan apakah peran terkait layanan terkait ada di akun Anda di IAM — Anda harus memiliki peran `AWSServiceRoleForCloudFrontLogger` IAM di akun Anda. Untuk informasi selengkapnya tentang peran ini, silakan lihat [Peran terkait layanan untuk Lambda @Edge](#).

Hapus fungsi dan replika Lambda @Edge

Anda dapat menghapus fungsi Lambda @Edge hanya ketika replika fungsi telah dihapus oleh CloudFront. Replika fungsi Lambda secara otomatis dihapus dalam situasi berikut:

- Setelah Anda menghapus kaitan terakhir untuk fungsi dari semua CloudFront distribusi. Jika lebih dari satu distribusi menggunakan fungsi, replika akan dihapus hanya setelah Anda menghapus asosiasi fungsi dari distribusi terakhir.
- Setelah Anda menghapus distribusi terakhir yang terkait dengan fungsi tersebut.

Replika biasanya dihapus dalam beberapa jam. Anda tidak dapat menghapus replika fungsi Lambda@Edge secara manual. Ini membantu mencegah situasi di mana replika dihapus yang masih digunakan, yang akan mengakibatkan kesalahan.

Warning

Jangan membuat aplikasi yang menggunakan replika fungsi Lambda @Edge di luar. CloudFront Replika ini dihapus saat asosiasi mereka dengan distribusi dihapus, atau saat distribusi itu sendiri dihapus. Replika yang aplikasi luarnya mungkin akan dihapus tanpa peringatan, sehingga menyebabkannya gagal.

Untuk menghapus asosiasi fungsi Lambda @Edge dari distribusi CloudFront

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Pilih ID distribusi dengan asosiasi fungsi Lambda @Edge yang ingin Anda hapus.
3. Pilih Perilaku tab.
4. Pilih perilaku cache yang memiliki asosiasi fungsi Lambda @Edge yang ingin Anda hapus, lalu pilih Edit.
5. Di bawah Asosiasi fungsi, Jenis fungsi, pilih Tidak ada asosiasi untuk menghapus asosiasi fungsi Lambda @Edge.

6. Pilih Simpan perubahan.

Setelah menghapus asosiasi fungsi Lambda @Edge dari CloudFront distribusi, Anda dapat menghapus fungsi Lambda atau versi fungsi dari. AWS Lambda Tunggu beberapa jam setelah menghapus asosiasi fungsi sehingga replika fungsi Lambda @Edge dapat dibersihkan. Setelah itu, Anda dapat menghapus fungsi dengan menggunakan konsol Lambda,, AWS CLI Lambda API, atau SDK. AWS

Anda juga dapat menghapus versi tertentu dari fungsi Lambda jika versi tersebut tidak memiliki CloudFront distribusi yang terkait dengannya. Setelah menghapus semua asosiasi untuk versi fungsi Lambda, tunggu beberapa jam. Kemudian Anda dapat menghapus versi fungsi.

Struktur acara Lambda @Edge

Topik berikut menjelaskan objek peristiwa permintaan dan respons yang CloudFront diteruskan ke fungsi Lambda @Edge saat dipicu.

Topik

- [Pemilihan asal dinamis](#)
- [Permintaan acara](#)
- [Peristiwa respon](#)

Pemilihan asal dinamis

Anda dapat menggunakan [pola jalur dalam perilaku cache](#) untuk merutekan permintaan ke asal berdasarkan jalur dan nama objek yang diminta, seperti `images/*.jpg`. Menggunakan Lambda@Edge, Anda juga dapat merutekan permintaan ke asal berdasarkan karakteristik lain, seperti nilai-nilai dalam header permintaan.

Ada sejumlah cara agar pemilihan asal dinamis ini dapat berguna. Misalnya, Anda dapat mendistribusikan permintaan lintas asal-usul di area geografis yang berbeda untuk membantu menyeimbangkan beban global. Atau Anda dapat secara selektif merutekan permintaan ke asal-usul berbeda yang masing-masing melayani fungsi tertentu: penanganan bot, optimalisasi SEO, autentikasi, dan sebagainya. Untuk contoh kode yang mendemonstrasikan cara menggunakan fitur ini, lihat [Pemilihan asal dinamis berbasis konten - contoh](#).

Dalam peristiwa permintaan CloudFront asal, `origin` objek dalam struktur peristiwa berisi informasi tentang asal yang akan diarahkan ke permintaan, berdasarkan pola jalur. Anda dapat memperbarui

nilai di `origin` mengajukan keberatan untuk mengirimkan permintaan ke negara asal yang berbeda. Saat Anda memperbarui `origin` objek, Anda tidak perlu menentukan asal dalam distribusi. Anda juga dapat mengganti objek asal Amazon S3 dengan objek asal kustom, dan sebaliknya. Namun, Anda hanya dapat menentukan asal tunggal per permintaan; asal kustom atau asal Amazon S3, tetapi tidak keduanya.

Permintaan acara

Topik berikut menunjukkan struktur objek yang CloudFront diteruskan ke fungsi Lambda untuk acara [permintaan penampil dan asal](#). Contoh-contoh ini menunjukkan GET tanpa isi. Berikut ini contoh adalah daftar semua bidang yang mungkin muncul dalam peristiwa permintaan penampil dan asal.

Topik

- [Contoh permintaan penampil](#)
- [Contoh permintaan asal](#)
- [Permintaan bidang acara](#)

Contoh permintaan penampil

Contoh berikut menunjukkan objek acara permintaan penampil.

```
{
  "Records": [
    {
      "cf": {
        "config": {
          "distributionDomainName": "d111111abcdef8.cloudfront.net",
          "distributionId": "EDFDVBD6EXAMPLE",
          "eventType": "viewer-request",
          "requestId": "4TyzHTaYWb1GX1qTfsHhEqV6HUdd_BzoBZnwfnvQc_1oF26C1koUSEQ=="
        },
        "request": {
          "clientIp": "203.0.113.178",
          "headers": {
            "host": [
              {
                "key": "Host",
                "value": "d111111abcdef8.cloudfront.net"
              }
            ],
            "user-agent": [
```

```

        {
            "key": "User-Agent",
            "value": "curl/7.66.0"
        }
    ],
    "accept": [
        {
            "key": "accept",
            "value": "*/*"
        }
    ]
},
"method": "GET",
"querystring": "",
"uri": "/"
}
}
}
]
}

```

Contoh permintaan asal

Contoh berikut menunjukkan objek peristiwa permintaan asal usul.

```

{
  "Records": [
    {
      "cf": {
        "config": {
          "distributionDomainName": "d111111abcdef8.cloudfront.net",
          "distributionId": "EDFDVBD6EXAMPLE",
          "eventType": "origin-request",
          "requestId": "4TyzHTaYWb1GX1qTfsHhEqV6HUDd_BzoBZnwfnvQc_1oF26C1koUSEQ=="
        },
        "request": {
          "clientIp": "203.0.113.178",
          "headers": {
            "x-forwarded-for": [
              {
                "key": "X-Forwarded-For",
                "value": "203.0.113.178"
              }
            ]
          },

```

```
    "user-agent": [
      {
        "key": "User-Agent",
        "value": "Amazon CloudFront"
      }
    ],
    "via": [
      {
        "key": "Via",
        "value": "2.0 2afae0d44e2540f472c0635ab62c232b.cloudfront.net
(CloudFront)"
      }
    ],
    "host": [
      {
        "key": "Host",
        "value": "example.org"
      }
    ],
    "cache-control": [
      {
        "key": "Cache-Control",
        "value": "no-cache"
      }
    ]
  },
  "method": "GET",
  "origin": {
    "custom": {
      "customHeaders": {},
      "domainName": "example.org",
      "keepaliveTimeout": 5,
      "path": "",
      "port": 443,
      "protocol": "https",
      "readTimeout": 30,
      "responseCompletionTimeout": 30,
      "sslProtocols": [
        "TLSv1",
        "TLSv1.1",
        "TLSv1.2"
      ]
    }
  }
},
```

```
        "querystring": "",
        "uri": "/"
    }
}
]
```

Permintaan bidang acara

Meminta data objek peristiwa dimuat dalam dua subobjek: `config` (`Records.cf.config`) and `request` (`Records.cf.request`). Daftar berikut menjelaskan setiap bidang subobject.

Bidang di objek konfigurasi

Daftar berikut menjelaskan bidang dalam `config` objek (`Records.cf.config`).

distributionDomainName (hanya baca)

Nama domain distribusi yang terkait dengan permintaan.

distributionID (hanya baca)

ID distribusi yang terkait dengan permintaan.

eventType (hanya baca)

Jenis pemicu yang terkait dengan permintaan: `viewer-request` atau `origin-request`.

requestId (hanya baca)

String terenkripsi yang secara unik mengidentifikasi permintaan. `viewer-to-CloudFront` `requestId` Nilai juga muncul di log CloudFront akses sebagai `x-edge-request-id`. Untuk informasi selengkapnya, lihat [Akses log \(log standar\)](#) dan [Bidang file log](#).

Bidang di objek permintaan

Daftar berikut menjelaskan bidang dalam `request` objek (`Records.cf.request`).

clientIp (hanya baca)

Alamat IP penampil yang membuat permintaan. Jika penampil menggunakan proksi HTTP atau penyeimbang beban untuk mengirim permintaan, nilainya adalah alamat IP proksi atau penyeimbang beban.

header (baca/tulis)

Header pada permintaan. Perhatikan hal-hal berikut:

- Kunci dalam `headers` objek adalah versi huruf kecil nama header HTTP standar. Menggunakan tombol huruf kecil memberi Anda akses huruf kecil ke nilai header.
- Setiap objek header (misalnya, `headers["accept"]` atau `headers["host"]`) adalah serangkaian pasangan utama-nilai. Untuk header tertentu, larik berisi satu pasangan nilai kunci untuk setiap nilai dalam permintaan.
- `key` berisi nama case-sensitive dari header seperti yang muncul dalam permintaan HTTP; misalnya,, `HostUser-Agent`, `X-Forwarded-ForCookie`, dan sebagainya.
- `value` berisi nilai header sebagaimana muncul dalam permintaan HTTP.
- Ketika fungsi Lambda Anda menambahkan atau memodifikasi header permintaan dan Anda tidak menyertakan bidang `headerkey`, Lambda @Edge secara otomatis menyisipkan header `key` menggunakan nama header yang Anda berikan. Terlepas dari bagaimana Anda telah memformat nama header, kunci header yang dimasukkan secara otomatis diformat dengan kapitalisasi awal untuk setiap bagian, dipisahkan oleh tanda hubung (-).

Misalnya, Anda dapat menambahkan header seperti berikut, tanpa header `key`:

```
"user-agent": [  
  {  
    "value": "ExampleCustomUserAgent/1.X.0"  
  }  
]
```

Dalam contoh ini, Lambda@Edge secara otomatis memasukkan `"key": "User-Agent"`.

Untuk informasi tentang pembatasan penggunaan header, lihat [Pembatasan pada fungsi edge](#).

method (hanya baca)

Metode HTTP permintaan.

querystring (baca/tulis)

String kueri, jika ada, dalam permintaan. Jika permintaan tidak menyertakan string kueri, objek acara masih menyertakan `querystring` dengan nilai kosong. Untuk informasi selengkapnya tentang string kueri, lihat [Konten cache berdasarkan parameter string kueri](#).

uri (baca/tulis)

Jalur relatif objek yang diminta. Jika fungsi Lambda Anda memodifikasi `uri` perhatikan hal-hal berikut:

- `uri` nilai harus dimulai dengan garis miring ke depan (/).
- Saat fungsi mengubah `uri` yang mengubah objek yang diminta oleh penampil.
- Ketika fungsi mengubah `uri` nilai, itu tidak mengubah perilaku cache untuk permintaan atau asal permintaan yang dikirim.

body (baca/tulis)

Isi permintaan HTTP. `body` struktur dapat memuat kolom berikut:

inputTruncated (hanya baca)

Bendera Boolean yang menunjukkan apakah tubuh dijejali Lambda@Edge. Untuk informasi selengkapnya, lihat [Pembatasan pada isi permintaan dengan opsi sertakan isi](#).

action (baca/tulis)

Tindakan yang ingin Anda lakukan dengan tubuh. Opsi untuk `action` adalah sebagai berikut:

- `read-only`: Ini adalah pengaturan default. Saat mengembalikan respons dari fungsi Lambda, jika `action` adalah hanya baca, Lambda@Edge mengabaikan setiap perubahan pada encoding atau data.
- `replace`: Tentukan ini saat Anda ingin mengganti tubuh yang dikirim ke asal.

encoding (baca/tulis)

Pengodean untuk tubuh. Saat Lambda@Edge mengekspos tubuh ke fungsi Lambda, pertama-tama tubuh berubah menjadi base64-encoding. Jika Anda memilih `replace` untuk `action` untuk mengganti tubuh, Anda dapat memilih untuk menggunakan pengodean base64 (default) atau `text`. Jika Anda menentukan encoding sebagai base64 tetapi tubuh tidak validbase64, CloudFront mengembalikan kesalahan.

data (baca/tulis)

Isi konten permintaan.

origin (baca/tulis) (hanya peristiwa awal)

Asal pengiriman permintaan ke. `origin`Struktur harus berisi tepat satu asal, yang dapat berupa asal khusus atau asal Amazon S3.

Bergantung pada jenis asal yang Anda tentukan (asal kustom atau Amazon S3), Anda harus menentukan bidang berikut dalam permintaan Anda:

customHeaders (baca/tulis) (kustom dan berasal dari Amazon S3)

(Opsional) Anda dapat menyertakan header khusus dengan permintaan dengan menentukan nama header dan pasangan nilai untuk setiap header kustom. Anda tidak dapat menambahkan header yang tidak diizinkan, dan header dengan nama yang sama tidak dapat hadir. `Records.cf.request.headers` [Catatan tentang header permintaan](#) juga berlaku untuk header kustom. Untuk informasi lebih lanjut, lihat [Header khusus yang tidak CloudFront dapat ditambahkan ke permintaan asal](#) dan [Pembatasan pada fungsi edge](#).

domainName (baca/tulis) (kustom dan berasal dari Amazon S3)

Nama domain asal. Nama domain tidak bisa kosong.

- Untuk asal kustom – Tentukan nama domain DNS, seperti `www.example.com`. Nama domain tidak dapat menyertakan titik dua (:), dan tidak bisa menjadi alamat IP. Nama domain dapat terdiri dari hingga 253 karakter.
- Untuk asal Amazon S3 – Tentukan nama domain DNS bucket Amazon S3, seperti `amzn-s3-demo-bucket.s3.eu-west-1.amazonaws.com`. Nama bisa sampai 128 karakter dan harus berupa huruf kecil.

path (baca/tulis) (kustom dan berasal dari Amazon S3)

Jalur direktori di tempat asal permintaan harus menemukan konten. Jalur harus dimulai dengan garis miring (/) tetapi tidak boleh diakhiri dengan satu (misalnya, seharusnya tidak diakhiri dengan `example-path/`). Hanya untuk asal kustom, alur harus diencode URL dan memiliki panjang maksimum 255 karakter.

keepaliveTimeout (baca/tulis) (hanya asal sesuai undang-undang)

Berapa lama, dalam hitungan detik, yang CloudFront harus mencoba mempertahankan koneksi ke asal setelah menerima paket terakhir dari respons. Nilainya harus berupa angka dari 1-120, inklusif.

port (baca/tulis) (hanya asal sesuai undang-undang)

Port yang CloudFront harus terhubung ke asal kustom Anda. Port harus 80, 443, atau nomor dalam kisaran 1024–65535, termasuk.

protocol (baca/tulis) (hanya asal sesuai undang-undang)

Protokol koneksi yang CloudFront harus digunakan saat menghubungkan ke asal Anda. Nilai dapat berupa `http` atau `https`.

readTimeout (baca/tulis) (kustom dan berasal dari Amazon S3)

Berapa lama, dalam hitungan detik, CloudFront harus menunggu tanggapan setelah mengirim permintaan ke asal Anda. Ini juga menentukan berapa lama CloudFront harus menunggu setelah menerima paket tanggapan sebelum menerima paket berikutnya. Nilainya harus berupa angka dari 1-120, inklusif.

Jika Anda membutuhkan kuota yang lebih tinggi, lihat [Batas waktu respons per asal](#).

responseCompletionTimeout (baca/tulis) (kustom dan berasal dari Amazon S3)

Waktu (dalam detik) permintaan dari CloudFront ke asal dapat tetap terbuka dan menunggu tanggapan. Jika respons lengkap tidak diterima dari asal saat ini, CloudFront akhiri koneksi.

Nilai untuk `responseCompletionTimeout` harus sama dengan atau lebih besar dari nilai untuk `readTimeout`. Jika Anda menetapkan nilai ini ke 0, itu akan menghapus nilai sebelumnya yang Anda tetapkan dan kembali ke default. Anda juga dapat melakukannya dengan menghapus `responseCompletionTimeout` bidang dari permintaan acara.

sslProtocols (baca/tulis) (hanya asal sesuai undang-undang)

SSL/TLS Protokol minimum yang CloudFront dapat digunakan saat membuat koneksi HTTPS dengan asal Anda. Nilai dapat berupa: TLSv1.2, TLSv1.1, TLSv1, atau SSLv3.

authMethod (baca/tulis) (hanya asal-usul Amazon S3)

Jika Anda menggunakan [identitas akses asal \(OAI\)](#), setel bidang ini ke `origin-access-identity`. Jika Anda tidak menggunakan OAI, atur `none`. Jika Anda mengatur `authMethod` untuk `origin-access-identity`, ada beberapa persyaratan:

- Anda harus menentukan `region` (lihat bidang berikut).
- Anda harus menggunakan OAI yang sama ketika Anda mengubah permintaan dari satu asal Amazon S3 ke yang lain.
- Anda tidak dapat menggunakan OAI saat mengubah permintaan dari asal kustom ke asal Amazon S3.

 Note

Bidang ini tidak mendukung [kontrol akses asal \(OAC\)](#).

region (baca/tulis) (hanya asal-usul Amazon S3)

AWS Wilayah ember Amazon S3 Anda. Ini hanya diperlukan ketika Anda mengatur `authMethod` untuk `origin-access-identity`.

Peristiwa respon

Topik berikut menunjukkan struktur objek yang CloudFront diteruskan ke fungsi Lambda untuk [penampil dan peristiwa respons asal](#). Berikut ini contoh adalah daftar semua bidang yang mungkin muncul di penampil dan kejadian respons asal.

Topik

- [Contoh respon asal](#)
- [Contoh respons penampil](#)
- [Bidang acara respons](#)

Contoh respon asal

Contoh berikut menunjukkan objek peristiwa respons asal usul.

```
{
  "Records": [
    {
      "cf": {
        "config": {
          "distributionDomainName": "d111111abcdef8.cloudfront.net",
          "distributionId": "EDFDVBD6EXAMPLE",
          "eventType": "origin-response",
          "requestId": "4TyzHTaYWb1GX1qTfsHhEqV6HUDd_BzoBZnwfnvQc_1oF26ClkoUSEQ=="
        },
        "request": {
          "clientIp": "203.0.113.178",
          "headers": {
            "x-forwarded-for": [
              {
                "key": "X-Forwarded-For",
                "value": "203.0.113.178"
              }
            ]
          },
          "user-agent": [
```

```

        {
            "key": "User-Agent",
            "value": "Amazon CloudFront"
        }
    ],
    "via": [
        {
            "key": "Via",
            "value": "2.0 8f22423015641505b8c857a37450d6c0.cloudfront.net
(CloudFront)"
        }
    ],
    "host": [
        {
            "key": "Host",
            "value": "example.org"
        }
    ],
    "cache-control": [
        {
            "key": "Cache-Control",
            "value": "no-cache"
        }
    ]
},
"method": "GET",
"origin": {
    "custom": {
        "customHeaders": {},
        "domainName": "example.org",
        "keepaliveTimeout": 5,
        "path": "",
        "port": 443,
        "protocol": "https",
        "readTimeout": 30,
        "responseCompletionTimeout": 30,
        "sslProtocols": [
            "TLSv1",
            "TLSv1.1",
            "TLSv1.2"
        ]
    }
},
"queryString": "",

```

```
    "uri": "/"
  },
  "response": {
    "headers": [
      {
        "key": "Access-Control-Allow-Credentials",
        "value": "true"
      }
    ],
    "access-control-allow-origin": [
      {
        "key": "Access-Control-Allow-Origin",
        "value": "*"
      }
    ],
    "date": [
      {
        "key": "Date",
        "value": "Mon, 13 Jan 2020 20:12:38 GMT"
      }
    ],
    "referrer-policy": [
      {
        "key": "Referrer-Policy",
        "value": "no-referrer-when-downgrade"
      }
    ],
    "server": [
      {
        "key": "Server",
        "value": "ExampleCustomOriginServer"
      }
    ],
    "x-content-type-options": [
      {
        "key": "X-Content-Type-Options",
        "value": "nosniff"
      }
    ],
    "x-frame-options": [
      {
        "key": "X-Frame-Options",
        "value": "DENY"
      }
    ]
  }
}
```

```

    }
  ],
  "x-xss-protection": [
    {
      "key": "X-XSS-Protection",
      "value": "1; mode=block"
    }
  ],
  "content-type": [
    {
      "key": "Content-Type",
      "value": "text/html; charset=utf-8"
    }
  ],
  "content-length": [
    {
      "key": "Content-Length",
      "value": "9593"
    }
  ]
},
"status": "200",
"statusDescription": "OK"
}
}
}
]
}

```

Contoh respons penampil

Contoh berikut menunjukkan objek acara respons penampil.

```

{
  "Records": [
    {
      "cf": {
        "config": {
          "distributionDomainName": "d111111abcdef8.cloudfront.net",
          "distributionId": "EDFDVBD6EXAMPLE",
          "eventType": "viewer-response",
          "requestId": "4TyzHTaYWb1GX1qTfsHhEqV6HUDd_BzoBZnwfnvQc_1oF26C1koUSEQ=="
        },
        "request": {

```

```
"clientIp": "203.0.113.178",
"headers": {
  "host": [
    {
      "key": "Host",
      "value": "d111111abcdef8.cloudfront.net"
    }
  ],
  "user-agent": [
    {
      "key": "User-Agent",
      "value": "curl/7.66.0"
    }
  ],
  "accept": [
    {
      "key": "accept",
      "value": "*/*"
    }
  ]
},
"method": "GET",
"queryString": "",
"uri": "/"
},
"response": {
  "headers": {
    "access-control-allow-credentials": [
      {
        "key": "Access-Control-Allow-Credentials",
        "value": "true"
      }
    ],
    "access-control-allow-origin": [
      {
        "key": "Access-Control-Allow-Origin",
        "value": "*"
      }
    ],
    "date": [
      {
        "key": "Date",
        "value": "Mon, 13 Jan 2020 20:14:56 GMT"
      }
    ]
  }
}
```



```
],
"referrer-policy": [
  {
    "key": "Referrer-Policy",
    "value": "no-referrer-when-downgrade"
  }
],
"server": [
  {
    "key": "Server",
    "value": "ExampleCustomOriginServer"
  }
],
"x-content-type-options": [
  {
    "key": "X-Content-Type-Options",
    "value": "nosniff"
  }
],
"x-frame-options": [
  {
    "key": "X-Frame-Options",
    "value": "DENY"
  }
],
"x-xss-protection": [
  {
    "key": "X-XSS-Protection",
    "value": "1; mode=block"
  }
],
"age": [
  {
    "key": "Age",
    "value": "2402"
  }
],
"content-type": [
  {
    "key": "Content-Type",
    "value": "text/html; charset=utf-8"
  }
],
"content-length": [
```

```
{
  {
    "key": "Content-Length",
    "value": "9593"
  }
],
"status": "200",
"statusDescription": "OK"
}
}
```

Bidang acara respons

Data objek peristiwa respons dimuat dalam tiga subobjek: `config` (`Records.cf.config`), `request` (`Records.cf.request`), dan `response` (`Records.cf.response`). Untuk informasi lebih lanjut tentang bidang di objek permintaan, lihat [Bidang di objek permintaan](#). Daftar berikut menjelaskan bidang dalam `config` dan `response` subobjects.

Bidang di objek konfigurasi

Daftar berikut menjelaskan bidang dalam `config` objek (`Records.cf.config`).

distributionDomainName (hanya baca)

Nama domain distribusi yang terkait dengan respons.

distributionID (hanya baca)

ID distribusi yang terkait dengan respons.

eventType (hanya baca)

Jenis pemicu yang terkait dengan respons: `origin-response` atau `viewer-response`.

requestId (hanya baca)

String terenkripsi yang secara unik mengidentifikasi viewer-to-CloudFront permintaan yang terkait dengan respons ini. `requestIdNilai` juga muncul di log CloudFront akses sebagai `x-edge-request-id`. Untuk informasi selengkapnya, lihat [Akses log \(log standar\)](#) dan [Bidang file log](#).

Bidang di objek respons

Daftar berikut menjelaskan bidang dalam response objek (`Records.cf.response`). Untuk informasi tentang penggunaan fungsi `Lambda@Edge` untuk membuat respons HTTP, lihat [Hasilkan respons HTTP dalam pemicu permintaan](#).

headers (baca/tulis)

Header dalam respons. Perhatikan hal-hal berikut:

- Kunci dalam `headers` objek adalah versi huruf kecil nama header HTTP standar. Menggunakan tombol huruf kecil memberi Anda akses huruf kecil ke nilai header.
- Setiap objek header (misalnya, `headers["content-type"]` atau `headers["content-length"]`) adalah serangkaian pasangan utama-nilai. Untuk header tertentu, larik berisi satu pasangan nilai kunci untuk setiap nilai dalam respons.
- key berisi nama case-sensitive header seperti yang muncul dalam respons HTTP; misalnya,, Content-Type Content-Length Cookie, dan sebagainya.
- `value` berisi nilai header sebagaimana muncul dalam respons HTTP.
- Ketika fungsi `Lambda` Anda menambahkan atau memodifikasi header respons dan Anda tidak menyertakan bidang `headerkey`, `Lambda @Edge` secara otomatis menyisipkan header key menggunakan nama header yang Anda berikan. Terlepas dari bagaimana Anda telah memformat nama header, kunci header yang dimasukkan secara otomatis diformat dengan kapitalisasi awal untuk setiap bagian, dipisahkan oleh tanda hubung (-).

Misalnya, Anda dapat menambahkan header seperti berikut, tanpa header key:

```
"content-type": [  
  {  
    "value": "text/html; charset=UTF-8"  
  }  
]
```

Dalam contoh ini, `Lambda@Edge` secara otomatis memasukkan "key": "Content-Type".

Untuk informasi tentang pembatasan penggunaan header, lihat [Pembatasan pada fungsi edge](#).

status

Kode status HTTP dari respons.

statusDescription

Deskripsi status HTTP untuk respons.

Bekerja dengan permintaan dan tanggapan

Untuk menggunakan permintaan dan tanggapan Lambda @Edge, lihat topik berikut:

Topik

- [Gunakan fungsi Lambda @Edge dengan failover asal](#)
- [Hasilkan respons HTTP dalam pemicu permintaan](#)
- [Perbarui tanggapan HTTP di pemicu respons asal](#)
- [Akses badan permintaan dengan memilih opsi include body](#)

Gunakan fungsi Lambda @Edge dengan failover asal

Anda dapat menggunakan fungsi Lambda @Edge dengan CloudFront distribusi yang telah Anda atur dengan grup asal, misalnya, untuk failover asal yang Anda konfigurasi untuk membantu memastikan ketersediaan tinggi. Untuk menggunakan fungsi Lambda dengan kelompok asal, tentukan fungsi dalam permintaan asal atau pemicu respons asal untuk kelompok asal saat Anda membuat perilaku cache.

Untuk informasi selengkapnya, lihat berikut ini:

- Buat grup asal: [Buat grup asal](#)
- Bagaimana asal failover bekerja dengan Lambda@Edge: [Gunakan failover asal dengan fungsi Lambda@Edge](#)

Hasilkan respons HTTP dalam pemicu permintaan

Saat CloudFront menerima permintaan, Anda dapat menggunakan fungsi Lambda untuk menghasilkan respons HTTP yang CloudFront kembali langsung ke penampil tanpa meneruskan respons ke asal. Membuat respons HTTP mengurangi beban pada asal mula, dan biasanya juga mengurangi latensi bagi penampil.

Beberapa skenario umum untuk menghasilkan tanggapan HTTP mencakup hal berikut:

- Mengembalikan halaman web kecil ke penampil

- Mengembalikan kode status HTTP 301 atau 302 untuk mengarahkan pengguna ke halaman web lain
- Mengembalikan kode status HTTP 401 ke penampil saat pengguna belum mengautentikasi

Fungsi Lambda@Edge dapat menghasilkan respons HTTP saat CloudFront peristiwa yang terjadi:

Peristiwa permintaan penampil

Saat fungsi dipicu oleh peristiwa permintaan penampil, CloudFront mengembalikan respons ke penampil dan tidak menyimpannya di cache.

Acara permintaan asal

Saat fungsi dipicu oleh peristiwa permintaan asal, CloudFront periksa cache tepi untuk respons yang sebelumnya dihasilkan oleh fungsi tersebut.

- Jika respons ada di cache, fungsi tidak dijalankan dan CloudFront mengembalikan respons cache ke penampil.
- Jika respons tidak ada dalam cache, fungsi akan dijalankan, CloudFront mengembalikan respons ke penampil, dan juga menyimpan datanya.

Untuk melihat beberapa kode sampel untuk menghasilkan respons HTTP, lihat [Lambda @Edge contoh fungsi](#). Anda juga dapat mengganti respons HTTP dalam pemicu respons. Untuk informasi selengkapnya, lihat [Perbarui tanggapan HTTP di pemicu respons asal](#).

Model pemrograman

Bagian ini menjelaskan model pemrograman untuk menggunakan Lambda@Edge untuk menghasilkan respons HTTP.

Topik

- [Objek respons](#)
- [Kesalahan](#)
- [Bidang yang diperlukan](#)

Objek respons

Tanggapan yang Anda kembalikan sebagai `result` parameter dari `callback` harus memiliki struktur berikut (perhatikan bahwa hanya status bidang wajib diisi).

```
const response = {
  body: 'content',
  bodyEncoding: 'text' | 'base64',
  headers: {
    'header name in lowercase': [{
      key: 'header name in standard case',
      value: 'header value'
    }],
    ...
  },
  status: 'HTTP status code (string)',
  statusDescription: 'status description'
};
```

Objek respons dapat mencakup nilai-nilai berikut:

body

Tubuh, jika ada, yang CloudFront ingin Anda kembalikan dalam respons yang dihasilkan.

bodyEncoding

Pengodean untuk nilai yang Anda tentukan dalam body. Satu-satunya pengodean yang valid adalah text dan base64. Jika Anda memasukkan body dalam response objek tetapi dihilangkan `bodyEncoding`, CloudFront perlakukan tubuh sebagai teks.

Jika Anda menentukan `bodyEncoding` sebagai base64 tetapi bagian tubuh tidak valid dasar64, CloudFront mengembalikan kesalahan.

headers

Header yang CloudFront ingin Anda kembalikan dalam respons yang dihasilkan. Perhatikan hal-hal berikut:

- Kunci dalam `headers` objek adalah versi huruf kecil nama header HTTP standar. Menggunakan tombol huruf kecil memberi Anda akses huruf kecil ke nilai header.
- Setiap header (misalnya, `headers["accept"]` atau `headers["host"]`) adalah serangkaian pasangan kunci. Untuk header tertentu, larik berisi satu pasangan nilai kunci untuk setiap nilai dalam respons yang dihasilkan.
- `key` (opsional) adalah nama judul yang peka terhadap kasus sebagaimana muncul dalam permintaan HTTP; misalnya, `accept` atau `host`.

- Tentukan `value` sebagai nilai header.
- Jika Anda tidak menyertakan bagian kunci header dari pasangan nilai kunci, Lambda@Edge secara otomatis memasukkan kunci header menggunakan nama header yang Anda berikan. Terlepas dari bagaimana Anda telah memformat nama header, kunci header yang disisipkan secara otomatis diformat dengan kapitalisasi awal untuk setiap bagian, dipisahkan oleh tanda hubung (-).

Misalnya, Anda dapat menambahkan header seperti berikut, tanpa tombol header: `'content-type': [{ value: 'text/html;charset=UTF-8' }]`

Dalam contoh ini, Lambda@Edge membuat tombol header berikut: `Content-Type`.

Untuk informasi tentang pembatasan penggunaan header, lihat [Pembatasan pada fungsi edge](#).

status

Kode status HTTP. Berikan kode status sebagai string. CloudFront menggunakan kode status yang disediakan untuk hal-hal berikut:

- Kembalikan dalam respons
- Cache di cache CloudFront tepi, ketika respons dihasilkan oleh fungsi yang dipicu oleh peristiwa permintaan asal
- Masuk CloudFront [Akses log \(log standar\)](#)

Jika status nilainya tidak antara 200 dan 599, CloudFront mengembalikan kesalahan ke penampil.

statusDescription

Deskripsi yang CloudFront ingin Anda kembalikan dalam respons, untuk menyertai kode status HTTP. Anda tidak perlu menggunakan deskripsi standar, seperti OK untuk kode status HTTP sebesar 200.

Kesalahan

Berikut ini adalah kemungkinan kesalahan untuk respons HTTP yang dihasilkan.

Respon Berisi Tubuh dan Menentukan 204 (Tidak Ada Konten) untuk Status

Ketika fungsi dipicu oleh permintaan penampil, CloudFront mengembalikan kode status HTTP 502 (Bad Gateway) ke penampil ketika kedua hal berikut ini benar:

- Nilai dari status adalah 204 (Tidak Ada Konten)
- Jawaban mencakup nilai untuk body

Ini karena Lambda@Edge mengenakan pembatasan opsional yang ditemukan dalam RFC 2616, yang menyatakan bahwa HTTP 204 respons tidak harus berisi isi pesan.

Pembatasan Ukuran Respons yang Dihasilkan

Ukuran maksimum respons yang dihasilkan oleh fungsi Lambda tergantung pada peristiwa yang memicu fungsi:

- Peristiwa permintaan penampil – 40 KB
- Acara permintaan asal – 1 MB

Jika respons lebih besar dari ukuran yang diizinkan, CloudFront mengembalikan kode status HTTP 502 (Bad Gateway) ke penampil.

Bidang yang diperlukan

status bidang wajib diisi.

Semua kolom lain bersifat opsional.

Perbarui tanggapan HTTP di pemicu respons asal

Saat CloudFront menerima respons HTTP dari server asal, jika ada pemicu respons asal yang terkait dengan perilaku cache, Anda dapat memodifikasi respons HTTP untuk mengganti apa yang dikembalikan dari asal.

Beberapa skenario umum untuk memperbarui respons HTTP mencakup hal berikut:

- Mengubah status untuk mengatur kode status HTTP 200 dan membuat konten tubuh statis untuk kembali ke penampil ketika asal mengembalikan kode status kesalahan (4xx atau 5xx). Untuk kode sampel, lihat [Contoh: Gunakan pemicu respons asal untuk memperbarui kode status kesalahan ke 200](#).
- Mengubah status untuk mengatur kode status HTTP 301 atau HTTP 302, untuk mengarahkan pengguna ke situs web lain saat asal mengembalikan kode status kesalahan (4xx atau 5xx). Untuk kode sampel, lihat [Contoh: Gunakan pemicu respons asal untuk memperbarui kode status kesalahan ke 302](#).

Note

Fungsi harus mengembalikan nilai status antara 200 dan 599 (inklusif), jika tidak CloudFront mengembalikan kesalahan ke penampil.

Anda juga dapat mengganti respons HTTP di penampil dan peristiwa permintaan asal. Untuk informasi selengkapnya, lihat [Hasilkan respons HTTP dalam pemicu permintaan](#).

Saat Anda bekerja dengan respons HTTP, Lambda @Edge tidak mengekspos badan yang dikembalikan oleh server asal ke pemicu respons asal. Anda dapat membuat badan konten statis dengan mengaturnya ke nilai yang diinginkan, atau menghapus tubuh di dalam fungsi dengan mengatur nilai yang akan kosong. Jika Anda tidak memperbarui bidang isi dalam fungsi Anda, badan asli yang dikembalikan oleh server asal dikembalikan ke penampil.

Akses badan permintaan dengan memilih opsi include body

Anda dapat memilih untuk menggunakan Lambda@Edge mengekspos tubuh dalam permintaan metode HTTP yang dapat ditulis (POST, PUT, DELETE, dan seterusnya), sehingga Anda dapat mengaksesnya dalam fungsi Lambda Anda. Anda dapat memilih akses hanya-baca, atau Anda dapat menentukan bahwa Anda akan mengganti isi.

Untuk mengaktifkan opsi ini, pilih Sertakan Tubuh saat membuat CloudFront untuk fungsi Anda yang digunakan untuk permintaan pemirsa atau acara permintaan asal. Untuk informasi lebih lanjut, lihat [Tambahkan pemicu untuk fungsi Lambda @Edge](#), atau untuk mempelajari penggunaan Sertakan Tubuh dengan fungsi Anda, lihat [Struktur acara Lambda @Edge](#).

Skenario saat Anda mungkin ingin menggunakan fitur ini meliputi hal berikut:

- Memproses formulir web, seperti formulir “hubungi kami”, tanpa mengirim data input pelanggan kembali ke server asal.
- Mengumpulkan data beacon web yang dikirim oleh browser penampil dan memprosesnya di tepi.

Untuk kode sampel, lihat [Lambda @Edge contoh fungsi](#).

Note

Jika badan permintaan berukuran besar, Lambda@Edge akan memotongnya. Untuk informasi terperinci tentang ukuran dan pemotongan maksimum, lihat [Pembatasan pada isi permintaan dengan opsi sertakan isi](#).

Lambda @Edge contoh fungsi

Lihat contoh berikut untuk menggunakan fungsi Lambda dengan Amazon. CloudFront

Note

Jika Anda memilih runtime Node.js 18 atau yang lebih baru untuk fungsi Lambda @Edge Anda, `index.mjs` file dibuat untuk Anda secara otomatis. Untuk menggunakan contoh kode berikut, ganti nama `index.mjs` file menjadi `index.js` sebagai gantinya.

Topik

- [Contoh Umum](#)
- [Hasilkan tanggapan - contoh](#)
- [String kueri - contoh](#)
- [Personalisasi konten berdasarkan header negara atau jenis perangkat - contoh](#)
- [Pemilihan asal dinamis berbasis konten - contoh](#)
- [Perbarui status kesalahan - contoh](#)
- [Akses badan permintaan - contoh](#)

Contoh Umum

Contoh berikut menunjukkan cara umum untuk menggunakan Lambda @Edge di. CloudFront

Topik

- [Contoh: A/B pengujian](#)
- [Contoh: Ganti header respons](#)

Contoh: A/B pengujian

Anda dapat menggunakan contoh berikut untuk menguji dua versi citra yang berbeda tanpa membuat pengalihan atau mengubah URL. Contoh ini membaca cookie di permintaan penampil dan memodifikasi URL permintaan dengan sesuai. Jika penampil tidak mengirim cookie dengan salah satu nilai yang diharapkan, contoh secara acak menetapkan penampil ke salah satu URLs

Node.js

```
'use strict';

exports.handler = (event, context, callback) => {
  const request = event.Records[0].cf.request;
  const headers = request.headers;

  if (request.uri !== '/experiment-pixel.jpg') {
    // do not process if this is not an A-B test request
    callback(null, request);
    return;
  }

  const cookieExperimentA = 'X-Experiment-Name=A';
  const cookieExperimentB = 'X-Experiment-Name=B';
  const pathExperimentA = '/experiment-group/control-pixel.jpg';
  const pathExperimentB = '/experiment-group/treatment-pixel.jpg';

  /*
   * Lambda at the Edge headers are array objects.
   *
   * Client may send multiple Cookie headers, i.e.:
   * > GET /viewerRes/test HTTP/1.1
   * > User-Agent: curl/7.18.1 (x86_64-unknown-linux-gnu) libcurl/7.18.1
   * OpenSSL/1.0.1u zlib/1.2.3
   * > Cookie: First=1; Second=2
   * > Cookie: ClientCode=abc
   * > Host: example.com
   *
   * You can access the first Cookie header at headers["cookie"][0].value
   * and the second at headers["cookie"][1].value.
   *
   * Header values are not parsed. In the example above,
   * headers["cookie"][0].value is equal to "First=1; Second=2"
   */
```

```
let experimentUri;
if (headers.cookie) {
  for (let i = 0; i < headers.cookie.length; i++) {
    if (headers.cookie[i].value.indexOf(cookieExperimentA) >= 0) {
      console.log('Experiment A cookie found');
      experimentUri = pathExperimentA;
      break;
    } else if (headers.cookie[i].value.indexOf(cookieExperimentB) >= 0) {
      console.log('Experiment B cookie found');
      experimentUri = pathExperimentB;
      break;
    }
  }
}

if (!experimentUri) {
  console.log('Experiment cookie has not been found. Throwing dice...');
  if (Math.random() < 0.75) {
    experimentUri = pathExperimentA;
  } else {
    experimentUri = pathExperimentB;
  }
}

request.uri = experimentUri;
console.log(`Request uri set to "${request.uri}"`);
callback(null, request);
};
```

Python

```
import json
import random

def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']
    headers = request['headers']

    if request['uri'] != '/experiment-pixel.jpg':
        # Not an A/B Test
        return request
```

```

    cookieExperimentA, cookieExperimentB = 'X-Experiment-Name=A', 'X-Experiment-
Name=B'
    pathExperimentA, pathExperimentB = '/experiment-group/control-pixel.jpg', '/
experiment-group/treatment-pixel.jpg'

    ...

Lambda at the Edge headers are array objects.

Client may send multiple cookie headers. For example:
> GET /viewerRes/test HTTP/1.1
> User-Agent: curl/7.18.1 (x86_64-unknown-linux-gnu) libcurl/7.18.1
OpenSSL/1.0.1u zlib/1.2.3
> Cookie: First=1; Second=2
> Cookie: ClientCode=abc
> Host: example.com

You can access the first Cookie header at headers["cookie"][0].value
and the second at headers["cookie"][1].value.

Header values are not parsed. In the example above,
headers["cookie"][0].value is equal to "First=1; Second=2"
'''

experimentUri = ""

for cookie in headers.get('cookie', []):
    if cookieExperimentA in cookie['value']:
        print("Experiment A cookie found")
        experimentUri = pathExperimentA
        break
    elif cookieExperimentB in cookie['value']:
        print("Experiment B cookie found")
        experimentUri = pathExperimentB
        break

if not experimentUri:
    print("Experiment cookie has not been found. Throwing dice...")
    if random.random() < 0.75:
        experimentUri = pathExperimentA
    else:
        experimentUri = pathExperimentB

request['uri'] = experimentUri
print(f"Request uri set to {experimentUri}")

```

```
return request
```

Contoh: Ganti header respons

Contoh berikut menunjukkan cara mengubah nilai header respons berdasarkan nilai header lain.

Node.js

```
export const handler = async (event) => {
  const response = event.Records[0].cf.response;
  const headers = response.headers;

  const headerNameSrc = 'X-Amz-Meta-Last-Modified';
  const headerNameDst = 'Last-Modified';

  if (headers[headerNameSrc.toLowerCase()]) {
    headers[headerNameDst.toLowerCase()] = [{
      key: headerNameDst,
      value: headers[headerNameSrc.toLowerCase()][0].value,
    }];
    console.log(`Response header "${headerNameDst}" was set to ` +
      `"${headers[headerNameDst.toLowerCase()][0].value}"`);
  }

  return response;
};
```

Python

```
import json

def lambda_handler(event, context):
    response = event['Records'][0]['cf']['response']
    headers = response['headers']

    header_name_src = 'X-Amz-Meta-Last-Modified'
    header_name_dst = 'Last-Modified'

    if headers.get(header_name_src.lower()):
        headers[header_name_dst.lower()] = [{
            'key': header_name_dst,
            'value': headers[header_name_src.lower()][0]['value']
```

```
    ]]  
    print(f'Response header "{header_name_dst}" was set to '  
          f'"{headers[header_name_dst.lower()][0]["value"]}"')  
  
    return response
```

Hasilkan tanggapan - contoh

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan Lambda @Edge untuk menghasilkan tanggapan.

Topik

- [Contoh: Sajikan konten statis \(respons yang dihasilkan\)](#)
- [Contoh: Menghasilkan pengalihan HTTP \(respons yang dihasilkan\)](#)

Contoh: Sajikan konten statis (respons yang dihasilkan)

Contoh berikut menunjukkan cara menggunakan fungsi Lambda untuk melayani konten situs web statis, yang mengurangi beban pada server asal dan mengurangi latensi keseluruhan.

Note

Anda dapat membuat tanggapan HTTP untuk permintaan penampil dan peristiwa permintaan asal. Untuk informasi selengkapnya, lihat [the section called “Hasilkan respons HTTP dalam pemicu permintaan”](#).

Anda juga dapat mengganti atau menghapus isi respons HTTP dalam peristiwa respons asal. Untuk informasi selengkapnya, lihat [the section called “Perbarui tanggapan HTTP di pemicu respons asal”](#).

Node.js

```
'use strict';  
  
const content = `  
<\!DOCTYPE html>  
<html lang="en">  
  <head>
```

```
<meta charset="utf-8">
<title>Simple Lambda@Edge Static Content Response</title>
</head>
<body>
  <p>Hello from Lambda@Edge!</p>
</body>
</html>
`;

exports.handler = (event, context, callback) => {
  /*
   * Generate HTTP OK response using 200 status code with HTML body.
   */
  const response = {
    status: '200',
    statusDescription: 'OK',
    headers: {
      'cache-control': [{
        key: 'Cache-Control',
        value: 'max-age=100'
      }],
      'content-type': [{
        key: 'Content-Type',
        value: 'text/html'
      }]
    },
    body: content,
  };
  callback(null, response);
};
```

Python

```
import json

CONTENT = """
<\!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Simple Lambda@Edge Static Content Response</title>
</head>
<body>
```



```
<p>Hello from Lambda@Edge!</p>
</body>
</html>
"""

def lambda_handler(event, context):
    # Generate HTTP OK response using 200 status code with HTML body.
    response = {
        'status': '200',
        'statusDescription': 'OK',
        'headers': {
            'cache-control': [
                {
                    'key': 'Cache-Control',
                    'value': 'max-age=100'
                }
            ],
            "content-type": [
                {
                    'key': 'Content-Type',
                    'value': 'text/html'
                }
            ]
        },
        'body': CONTENT
    }
    return response
```

Contoh: Menghasilkan pengalihan HTTP (respons yang dihasilkan)

Contoh berikut ini menunjukkan cara membuat pengalihan HTTP.

Note

Anda dapat membuat tanggapan HTTP untuk permintaan penampil dan peristiwa permintaan asal. Untuk informasi selengkapnya, lihat [Hasilkan respons HTTP dalam pemicu permintaan](#).

Node.js

```
'use strict';
```

```
exports.handler = (event, context, callback) => {
  /*
   * Generate HTTP redirect response with 302 status code and Location header.
   */
  const response = {
    status: '302',
    statusDescription: 'Found',
    headers: {
      location: [{
        key: 'Location',
        value: 'https://docs.aws.amazon.com/lambda/latest/dg/lambda-
edge.html',
      }],
    },
  };
  callback(null, response);
};
```

Python

```
def lambda_handler(event, context):

    # Generate HTTP redirect response with 302 status code and Location header.

    response = {
        'status': '302',
        'statusDescription': 'Found',
        'headers': {
            'location': [{
                'key': 'Location',
                'value': 'https://docs.aws.amazon.com/lambda/latest/dg/lambda-
edge.html'
            }]
        }
    }

    return response
```

String kueri - contoh

Contoh berikut menunjukkan cara Anda dapat menggunakan Lambda @Edge dengan string kueri.

Topik

- [Contoh: Tambahkan header berdasarkan parameter string kueri](#)
- [Contoh: Normalisasi parameter string kueri untuk meningkatkan rasio hit cache](#)
- [Contoh: Mengarahkan pengguna yang tidak diautentikasi ke halaman login](#)

Contoh: Tambahkan header berdasarkan parameter string kueri

Contoh berikut menunjukkan cara mendapatkan pasangan nilai kunci dari parameter string pencarian, kemudian menambahkan header berdasarkan nilai tersebut.

Node.js

```
'use strict';

const querystring = require('querystring');
exports.handler = (event, context, callback) => {
  const request = event.Records[0].cf.request;

  /* When a request contains a query string key-value pair but the origin server
   * expects the value in a header, you can use this Lambda function to
   * convert the key-value pair to a header. Here's what the function does:
   * 1. Parses the query string and gets the key-value pair.
   * 2. Adds a header to the request using the key-value pair that the function
   got in step 1.
   */

  /* Parse request querystring to get javascript object */
  const params = querystring.parse(request.querystring);

  /* Move auth param from querystring to headers */
  const headerName = 'Auth-Header';
  request.headers[headerName.toLowerCase()] = [{ key: headerName, value:
params.auth }];
  delete params.auth;

  /* Update request querystring */
  request.querystring = querystring.stringify(params);

  callback(null, request);
};
```

Python

```
from urllib.parse import parse_qs, urlencode

def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']

    '''
    When a request contains a query string key-value pair but the origin server
    expects the value in a header, you can use this Lambda function to
    convert the key-value pair to a header. Here's what the function does:
        1. Parses the query string and gets the key-value pair.
        2. Adds a header to the request using the key-value pair that the function
    got in step 1.
    '''

    # Parse request querystring to get dictionary/json
    params = {k : v[0] for k, v in parse_qs(request['querystring']).items()}

    # Move auth param from querystring to headers
    headerName = 'Auth-Header'
    request['headers'][headerName.lower()] = [{'key': headerName, 'value':
params['auth']}]}
    del params['auth']

    # Update request querystring
    request['querystring'] = urlencode(params)

    return request
```

Contoh: Normalisasi parameter string kueri untuk meningkatkan rasio hit cache

Contoh berikut menunjukkan cara meningkatkan rasio hit cache Anda dengan membuat perubahan berikut pada string kueri sebelum CloudFront meneruskan permintaan ke asal Anda:

- Berikan kombinasi nilai kunci dengan nama parameter.
- Ubah kasus pasangan yang bernilai kunci menjadi huruf kecil.

Untuk informasi selengkapnya, lihat [Konten cache berdasarkan parameter string kueri](#).

Node.js

```
'use strict';

const querystring = require('querystring');

exports.handler = (event, context, callback) => {
  const request = event.Records[0].cf.request;
  /* When you configure a distribution to forward query strings to the origin and
   * to cache based on an allowlist of query string parameters, we recommend
   * the following to improve the cache-hit ratio:
   * - Always list parameters in the same order.
   * - Use the same case for parameter names and values.
   *
   * This function normalizes query strings so that parameter names and values
   * are lowercase and parameter names are in alphabetical order.
   *
   * For more information, see:
   * https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/
QueryStringParameters.html
   */

  console.log('Query String: ', request.querystring);

  /* Parse request query string to get javascript object */
  const params = querystring.parse(request.querystring.toLowerCase());
  const sortedParams = {};

  /* Sort param keys */
  Object.keys(params).sort().forEach(key => {
    sortedParams[key] = params[key];
  });

  /* Update request querystring with normalized */
  request.querystring = querystring.stringify(sortedParams);

  callback(null, request);
};
```

Python

```
from urllib.parse import parse_qs, urlencode
```

```
def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']
    ...

    When you configure a distribution to forward query strings to the origin and
    to cache based on an allowlist of query string parameters, we recommend
    the following to improve the cache-hit ratio:
    Always list parameters in the same order.
    - Use the same case for parameter names and values.

    This function normalizes query strings so that parameter names and values
    are lowercase and parameter names are in alphabetical order.

    For more information, see:
    https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/
    QueryStringParameters.html
    ...
    print("Query string: ", request["querystring"])

    # Parse request query string to get js object
    params = {k : v[0] for k, v in parse_qs(request['querystring'].lower()).items()}

    # Sort param keys
    sortedParams = sorted(params.items(), key=lambda x: x[0])

    # Update request querystring with normalized
    request['querystring'] = urlencode(sortedParams)

    return request
```

Contoh: Mengarahkan pengguna yang tidak diautentikasi ke halaman login

Contoh berikut menunjukkan cara mengalihkan pengguna ke halaman masuk jika mereka belum memasukkan kredensial mereka.

Node.js

```
'use strict';

function parseCookies(headers) {
    const parsedCookie = {};
    if (headers.cookie) {
        headers.cookie[0].value.split(';').forEach((cookie) => {
```

```
        if (cookie) {
            const parts = cookie.split('=');
            parsedCookie[parts[0].trim()] = parts[1].trim();
        }
    });
}
return parsedCookie;
}

exports.handler = (event, context, callback) => {
    const request = event.Records[0].cf.request;
    const headers = request.headers;

    /* Check for session-id in request cookie in viewer-request event,
     * if session-id is absent, redirect the user to sign in page with original
     * request sent as redirect_url in query params.
     */

    /* Check for session-id in cookie, if present then proceed with request */
    const parsedCookies = parseCookies(headers);
    if (parsedCookies && parsedCookies['session-id']) {
        callback(null, request);
        return;
    }

    /* URI encode the original request to be sent as redirect_url in query params */
    const encodedRedirectUrl = encodeURIComponent(`https://${headers.host[0].value}${request.uri}?${request.querystring}`);
    const response = {
        status: '302',
        statusDescription: 'Found',
        headers: {
            location: [{
                key: 'Location',
                value: `https://www.example.com/signin?redirect_url=${encodedRedirectUrl}`,
            }],
        },
    };
    callback(null, response);
};
```

Python

```
import urllib

def parseCookies(headers):
    parsedCookie = {}
    if headers.get('cookie'):
        for cookie in headers['cookie'][0]['value'].split(';'):
            if cookie:
                parts = cookie.split('=')
                parsedCookie[parts[0].strip()] = parts[1].strip()
    return parsedCookie

def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']
    headers = request['headers']

    '''
    Check for session-id in request cookie in viewer-request event,
    if session-id is absent, redirect the user to sign in page with original
    request sent as redirect_url in query params.
    '''

    # Check for session-id in cookie, if present, then proceed with request
    parsedCookies = parseCookies(headers)

    if parsedCookies and parsedCookies['session-id']:
        return request

    # URI encode the original request to be sent as redirect_url in query params
    redirectUrl = "https://%s%s?%s" % (headers['host'][0]['value'], request['uri'],
    request['querystring'])
    encodedRedirectUrl = urllib.parse.quote_plus(redirectUrl.encode('utf-8'))

    response = {
        'status': '302',
        'statusDescription': 'Found',
        'headers': {
            'location': [{
                'key': 'Location',
                'value': 'https://www.example.com/signin?redirect_url=%s' %
encodedRedirectUrl
            }]
        }
    }
```



```
}  
return response
```

Personalisasi konten berdasarkan header negara atau jenis perangkat - contoh

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan Lambda @Edge untuk menyesuaikan perilaku berdasarkan lokasi atau jenis perangkat yang digunakan oleh penampil.

Topik

- [Contoh: Mengarahkan permintaan penampil ke URL khusus negara](#)
- [Contoh: Sajikan berbagai versi objek berdasarkan perangkat](#)

Contoh: Mengarahkan permintaan penampil ke URL khusus negara

Contoh berikut menunjukkan cara membuat respons pengalihan HTTP dengan URL khusus negara dan mengembalikan respons ke penampil. Ini berguna saat Anda ingin memberikan tanggapan khusus negara. Sebagai contoh:

- Jika Anda memiliki subdomain spesifik negara, seperti kami.example.com dan tw.example.com, Anda dapat membuat respons pengalihan saat penampil meminta contoh.com.
- Jika Anda melakukan streaming video tetapi tidak memiliki hak untuk melakukan streaming konten di negara tertentu, Anda dapat mengarahkan pengguna di negara tersebut ke halaman yang menjelaskan mengapa mereka tidak dapat melihat video tersebut.

Perhatikan hal-hal berikut:

- Anda harus mengonfigurasi distribusi Anda ke cache berdasarkan CloudFront-Viewer-Country header. Untuk informasi selengkapnya, lihat [Cache berdasarkan header permintaan yang dipilih](#).
- CloudFront menambahkan CloudFront-Viewer-Country header setelah acara permintaan penampil. Untuk menggunakan contoh ini, Anda harus membuat pemicu untuk kejadian permintaan asal.

Node.js

```
'use strict';
```

```

/* This is an origin request function */
exports.handler = (event, context, callback) => {
  const request = event.Records[0].cf.request;
  const headers = request.headers;

  /*
   * Based on the value of the CloudFront-Viewer-Country header, generate an
   * HTTP status code 302 (Redirect) response, and return a country-specific
   * URL in the Location header.
   * NOTE: 1. You must configure your distribution to cache based on the
   *         CloudFront-Viewer-Country header. For more information, see
   *         https://docs.aws.amazon.com/console/cloudfront/cache-on-selected-
headers
   *         2. CloudFront adds the CloudFront-Viewer-Country header after the
viewer
   *         request event. To use this example, you must create a trigger for
the
   *         origin request event.
   */

  let url = 'https://example.com/';
  if (headers['cloudfront-viewer-country']) {
    const countryCode = headers['cloudfront-viewer-country'][0].value;
    if (countryCode === 'TW') {
      url = 'https://tw.example.com/';
    } else if (countryCode === 'US') {
      url = 'https://us.example.com/';
    }
  }

  const response = {
    status: '302',
    statusDescription: 'Found',
    headers: {
      location: [{
        key: 'Location',
        value: url,
      }],
    },
  };
  callback(null, response);
};

```

Python

```
# This is an origin request function

def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']
    headers = request['headers']

    ...

    Based on the value of the CloudFront-Viewer-Country header, generate an
    HTTP status code 302 (Redirect) response, and return a country-specific
    URL in the Location header.
    NOTE: 1. You must configure your distribution to cache based on the
           CloudFront-Viewer-Country header. For more information, see
           https://docs.aws.amazon.com/console/cloudfront/cache-on-selected-headers
        2. CloudFront adds the CloudFront-Viewer-Country header after the viewer
           request event. To use this example, you must create a trigger for the
           origin request event.
    ...

    url = 'https://example.com/'
    viewerCountry = headers.get('cloudfront-viewer-country')
    if viewerCountry:
        countryCode = viewerCountry[0]['value']
        if countryCode == 'TW':
            url = 'https://tw.example.com/'
        elif countryCode == 'US':
            url = 'https://us.example.com/'

    response = {
        'status': '302',
        'statusDescription': 'Found',
        'headers': {
            'location': [{
                'key': 'Location',
                'value': url
            }]
        }
    }

    return response
```

Contoh: Sajikan berbagai versi objek berdasarkan perangkat

Contoh berikut ini menunjukkan cara menyajikan versi objek berbeda berdasarkan jenis perangkat yang digunakan pengguna, misalnya, perangkat seluler atau tablet. Perhatikan hal-hal berikut:

- Anda harus mengonfigurasi distribusi Anda ke cache berdasarkan CloudFront-Is-*-Viewer yang berbeda. Untuk informasi selengkapnya, lihat [Cache berdasarkan header permintaan yang dipilih](#).
- CloudFront menambahkan CloudFront-Is-*-Viewer header setelah acara permintaan penampil. Untuk menggunakan contoh ini, Anda harus membuat pemicu untuk kejadian permintaan asal.

Node.js

```
'use strict';

/* This is an origin request function */
exports.handler = (event, context, callback) => {
  const request = event.Records[0].cf.request;
  const headers = request.headers;

  /*
   * Serve different versions of an object based on the device type.
   * NOTE: 1. You must configure your distribution to cache based on the
   *         CloudFront-Is-*-Viewer headers. For more information, see
   *         the following documentation:
   *         https://docs.aws.amazon.com/console/cloudfront/cache-on-selected-
headers
   *         https://docs.aws.amazon.com/console/cloudfront/cache-on-device-type
   *         2. CloudFront adds the CloudFront-Is-*-Viewer headers after the viewer
   *         request event. To use this example, you must create a trigger for
the
   *         origin request event.
   */

  const desktopPath = '/desktop';
  const mobilePath = '/mobile';
  const tabletPath = '/tablet';
  const smarttvPath = '/smarttv';

  if (headers['cloudfront-is-desktop-viewer'])
```

```

        && headers['cloudfront-is-desktop-viewer'][0].value === 'true') {
            request.uri = desktopPath + request.uri;
        } else if (headers['cloudfront-is-mobile-viewer']
            && headers['cloudfront-is-mobile-viewer'][0].value === 'true') {
            request.uri = mobilePath + request.uri;
        } else if (headers['cloudfront-is-tablet-viewer']
            && headers['cloudfront-is-tablet-viewer'][0].value === 'true') {
            request.uri = tabletPath + request.uri;
        } else if (headers['cloudfront-is-smarttv-viewer']
            && headers['cloudfront-is-smarttv-viewer'][0].value === 'true') {
            request.uri = smarttvPath + request.uri;
        }
        console.log(`Request uri set to "${request.uri}"`);

        callback(null, request);
    };

```

Python

```

# This is an origin request function
def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']
    headers = request['headers']

    ...

    Serve different versions of an object based on the device type.
    NOTE: 1. You must configure your distribution to cache based on the
        CloudFront-Is-*-Viewer headers. For more information, see
        the following documentation:
        https://docs.aws.amazon.com/console/cloudfront/cache-on-selected-headers
        https://docs.aws.amazon.com/console/cloudfront/cache-on-device-type
        2. CloudFront adds the CloudFront-Is-*-Viewer headers after the viewer
        request event. To use this example, you must create a trigger for the
        origin request event.

    ...

    desktopPath = '/desktop';
    mobilePath = '/mobile';
    tabletPath = '/tablet';
    smarttvPath = '/smarttv';

    if 'cloudfront-is-desktop-viewer' in headers and headers['cloudfront-is-desktop-
viewer'][0]['value'] == 'true':

```

```
request['uri'] = desktopPath + request['uri']
elif 'cloudfront-is-mobile-viewer' in headers and headers['cloudfront-is-mobile-viewer'][0]['value'] == 'true':
    request['uri'] = mobilePath + request['uri']
elif 'cloudfront-is-tablet-viewer' in headers and headers['cloudfront-is-tablet-viewer'][0]['value'] == 'true':
    request['uri'] = tabletPath + request['uri']
elif 'cloudfront-is-smarttv-viewer' in headers and headers['cloudfront-is-smarttv-viewer'][0]['value'] == 'true':
    request['uri'] = smarttvPath + request['uri']

print("Request uri set to %s" % request['uri'])

return request
```

Pemilihan asal dinamis berbasis konten - contoh

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan Lambda @Edge untuk merutekan ke asal yang berbeda berdasarkan informasi dalam permintaan.

Topik

- [Contoh: Menggunakan pemicu permintaan asal untuk mengubah dari asal kustom ke asal Amazon S3](#)
- [Contoh: Gunakan pemicu permintaan asal untuk mengubah Wilayah asal Amazon S3](#)
- [Contoh: Menggunakan pemicu permintaan asal untuk mengubah dari asal Amazon S3 ke asal kustom](#)
- [Contoh: Gunakan pemicu permintaan asal untuk mentransfer lalu lintas secara bertahap dari satu bucket Amazon S3 ke bucket lainnya](#)
- [Contoh: Gunakan pemicu permintaan asal untuk mengubah nama domain asal berdasarkan header negara](#)

Contoh: Menggunakan pemicu permintaan asal untuk mengubah dari asal kustom ke asal Amazon S3

Fungsi ini mendemonstrasikan bagaimana pemicu permintaan asal dapat digunakan untuk mengubah dari asal kustom ke asal Amazon S3 dari mana konten diambil, berdasarkan properti permintaan.

Node.js

```
'use strict';

const querystring = require('querystring');

exports.handler = (event, context, callback) => {
    const request = event.Records[0].cf.request;

    /**
     * Reads query string to check if S3 origin should be used, and
     * if true, sets S3 origin properties.
     */

    const params = querystring.parse(request.querystring);

    if (params['useS3origin']) {
        if (params['useS3origin'] === 'true') {
            const s3DomainName = 'amzn-s3-demo-bucket.s3.amazonaws.com';

            /* Set S3 origin fields */
            request.origin = {
                s3: {
                    domainName: s3DomainName,
                    region: '',
                    authMethod: 'origin-access-identity',
                    path: '',
                    customHeaders: {}
                }
            };
            request.headers['host'] = [{ key: 'host', value: s3DomainName}];
        }
    }

    callback(null, request);
};
```

Python

```
from urllib.parse import parse_qs

def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']
```

```

'''
Reads query string to check if S3 origin should be used, and
if true, sets S3 origin properties
'''
params = {k: v[0] for k, v in parse_qs(request['querystring']).items()}
if params.get('useS3Origin') == 'true':
    s3DomainName = 'amzn-s3-demo-bucket.s3.amazonaws.com'

    # Set S3 origin fields
    request['origin'] = {
        's3': {
            'domainName': s3DomainName,
            'region': '',
            'authMethod': 'origin-access-identity',
            'path': '',
            'customHeaders': {}
        }
    }
    request['headers']['host'] = [{'key': 'host', 'value': s3DomainName}]
return request

```

Contoh: Gunakan pemicu permintaan asal untuk mengubah Wilayah asal Amazon S3

Fungsi ini menunjukkan bagaimana pemicu permintaan asal dapat digunakan untuk mengubah asal Amazon S3 dari mana konten diambil, berdasarkan properti permintaan.

Dalam contoh ini, kami menggunakan nilai `CloudFront-Viewer-Country` header untuk memperbarui nama domain bucket S3 ke bucket di Wilayah yang lebih dekat dengan penampil. Ini dapat berguna dalam beberapa cara:

- Ini mengurangi keterlambatan saat Wilayah yang ditentukan lebih dekat ke negara penampil.
- Ini menyediakan kedaulatan data dengan memastikan bahwa data dilayani dari asal yang berada di negara yang sama dengan asal permintaan tersebut.

Untuk menggunakan contoh ini, Anda harus melakukan hal berikut:

- Konfigurasi distribusi Anda ke cache berdasarkan `CloudFront-Viewer-Country` header. Untuk informasi selengkapnya, lihat [Cache berdasarkan header permintaan yang dipilih](#).

- Buat pemicu untuk fungsi ini di acara permintaan asal. CloudFront menambahkan CloudFront-Viewer-Country header setelah peristiwa permintaan penampil, jadi untuk menggunakan contoh ini, Anda harus memastikan bahwa fungsi tersebut dijalankan untuk permintaan asal.

Note

Kode contoh berikut menggunakan identitas akses asal (OAI) yang sama untuk semua bucket S3 yang Anda gunakan untuk asal Anda. Untuk informasi selengkapnya, lihat [Identitas akses asal](#).

Node.js

```
'use strict';

exports.handler = (event, context, callback) => {
    const request = event.Records[0].cf.request;

    /**
     * This blueprint demonstrates how an origin-request trigger can be used to
     * change the origin from which the content is fetched, based on request
     * properties.
     * In this example, we use the value of the CloudFront-Viewer-Country header
     * to update the S3 bucket domain name to a bucket in a Region that is closer to
     * the viewer.
     *
     * This can be useful in several ways:
     *     1) Reduces latencies when the Region specified is nearer to the viewer's
     *        country.
     *     2) Provides data sovereignty by making sure that data is served from an
     *        origin that's in the same country that the request came from.
     *
     * NOTE: 1. You must configure your distribution to cache based on the
     *        CloudFront-Viewer-Country header. For more information, see
     *        https://docs.aws.amazon.com/console/cloudfront/cache-on-selected-
     * headers
     *        2. CloudFront adds the CloudFront-Viewer-Country header after the
     * viewer
     *        request event. To use this example, you must create a trigger for
     * the
     *        origin request event.
```

```

    */

    const countryToRegion = {
      'DE': 'eu-central-1',
      'IE': 'eu-west-1',
      'GB': 'eu-west-2',
      'FR': 'eu-west-3',
      'JP': 'ap-northeast-1',
      'IN': 'ap-south-1'
    };

    if (request.headers['cloudfront-viewer-country']) {
      const countryCode = request.headers['cloudfront-viewer-country'][0].value;
      const region = countryToRegion[countryCode];

      /**
       * If the viewer's country is not in the list you specify, the request
       * goes to the default S3 bucket you've configured.
       */
      if (region) {
        /**
         * If you've set up OAI, the bucket policy in the destination bucket
         * should allow the OAI GetObject operation, as configured by default
         * for an S3 origin with OAI. Another requirement with OAI is to provide
         * the Region so it can be used for the SIGV4 signature. Otherwise, the
         * Region is not required.
         */
        request.origin.s3.region = region;
        const domainName = `amzn-s3-demo-bucket-in-${region}.s3.
${region}.amazonaws.com`;
        request.origin.s3.domainName = domainName;
        request.headers['host'] = [{ key: 'host', value: domainName }];
      }
    }

    callback(null, request);
  };

```

Python

```

def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']

```

```
...
```

This blueprint demonstrates how an origin-request trigger can be used to change the origin from which the content is fetched, based on request properties.

In this example, we use the value of the CloudFront-Viewer-Country header to update the S3 bucket domain name to a bucket in a Region that is closer to the viewer.

This can be useful in several ways:

- 1) Reduces latencies when the Region specified is nearer to the viewer's country.
- 2) Provides data sovereignty by making sure that data is served from an origin that's in the same country that the request came from.

NOTE: 1. You must configure your distribution to cache based on the CloudFront-Viewer-Country header. For more information, see <https://docs.aws.amazon.com/console/cloudfront/cache-on-selected-headers>

2. CloudFront adds the CloudFront-Viewer-Country header after the viewer request event. To use this example, you must create a trigger for the origin request event.

```
...
```

```
countryToRegion = {
    'DE': 'eu-central-1',
    'IE': 'eu-west-1',
    'GB': 'eu-west-2',
    'FR': 'eu-west-3',
    'JP': 'ap-northeast-1',
    'IN': 'ap-south-1'
}
```

```
viewerCountry = request['headers'].get('cloudfront-viewer-country')
```

```
if viewerCountry:
```

```
    countryCode = viewerCountry[0]['value']
    region = countryToRegion.get(countryCode)
```

```
# If the viewer's country is not in the list you specify, the request
# goes to the default S3 bucket you've configured
```

```
if region:
```

```
    ...
```

If you've set up OAI, the bucket policy in the destination bucket should allow the OAI GetObject operation, as configured by default for an S3 origin with OAI. Another requirement with OAI is to provide the Region so it can be used for the SIGV4 signature. Otherwise, the

```

        Region is not required.
        ...
        request['origin']['s3']['region'] = region
        domainName = 'amzn-s3-demo-bucket-in-{0}.s3.
{0}.amazonaws.com'.format(region)
        request['origin']['s3']['domainName'] = domainName
        request['headers']['host'] = [{'key': 'host', 'value': domainName}]

    return request

```

Contoh: Menggunakan pemicu permintaan asal untuk mengubah dari asal Amazon S3 ke asal kustom

Fungsi ini menunjukkan bagaimana pemicu permintaan asal dapat digunakan untuk mengubah asal kustom dari mana konten diambil, berdasarkan sifat permintaan.

Node.js

```

'use strict';

const querystring = require('querystring');

exports.handler = (event, context, callback) => {
    const request = event.Records[0].cf.request;

    /**
     * Reads query string to check if custom origin should be used, and
     * if true, sets custom origin properties.
     */

    const params = querystring.parse(request.querystring);

    if (params['useCustomOrigin']) {
        if (params['useCustomOrigin'] === 'true') {

            /* Set custom origin fields*/
            request.origin = {
                custom: {
                    domainName: 'www.example.com',
                    port: 443,
                    protocol: 'https',
                    path: '',

```

```

        sslProtocols: ['TLSv1', 'TLSv1.1'],
        readTimeout: 5,
        keepaliveTimeout: 5,
        customHeaders: {}
    }
};
request.headers['host'] = [{ key: 'host', value: 'www.example.com'}];
}
}
callback(null, request);
};

```

Python

```

from urllib.parse import parse_qs

def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']

    # Reads query string to check if custom origin should be used, and
    # if true, sets custom origin properties

    params = {k: v[0] for k, v in parse_qs(request['queryString']).items()}

    if params.get('useCustomOrigin') == 'true':
        # Set custom origin fields
        request['origin'] = {
            'custom': {
                'domainName': 'www.example.com',
                'port': 443,
                'protocol': 'https',
                'path': '',
                'sslProtocols': ['TLSv1', 'TLSv1.1'],
                'readTimeout': 5,
                'keepaliveTimeout': 5,
                'customHeaders': {}
            }
        }
        request['headers']['host'] = [{'key': 'host', 'value':
'www.example.com'}]

    return request

```

Contoh: Gunakan pemicu permintaan asal untuk mentransfer lalu lintas secara bertahap dari satu bucket Amazon S3 ke bucket lainnya

Fungsi ini menunjukkan bagaimana Anda dapat secara bertahap mentransfer lalu lintas dari satu bucket Amazon S3 ke bucket lain dengan cara yang terkontrol.

Node.js

```
'use strict';

function getRandomInt(min, max) {
  /* Random number is inclusive of min and max*/
  return Math.floor(Math.random() * (max - min + 1)) + min;
}

exports.handler = (event, context, callback) => {
  const request = event.Records[0].cf.request;
  const BLUE_TRAFFIC_PERCENTAGE = 80;

  /**
   * This Lambda function demonstrates how to gradually transfer traffic from
   * one S3 bucket to another in a controlled way.
   * We define a variable BLUE_TRAFFIC_PERCENTAGE which can take values from
   * 1 to 100. If the generated randomNumber less than or equal to
   BLUE_TRAFFIC_PERCENTAGE, traffic
   * is re-directed to blue-bucket. If not, the default bucket that we've
   configured
   * is used.
   */

  const randomNumber = getRandomInt(1, 100);

  if (randomNumber <= BLUE_TRAFFIC_PERCENTAGE) {
    const domainName = 'blue-bucket.s3.amazonaws.com';
    request.origin.s3.domainName = domainName;
    request.headers['host'] = [{ key: 'host', value: domainName}];
  }
  callback(null, request);
};
```

Python

```
import math
```

```

import random

def getRandomInt(min, max):
    # Random number is inclusive of min and max
    return math.floor(random.random() * (max - min + 1)) + min

def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']
    BLUE_TRAFFIC_PERCENTAGE = 80

    ...

    This Lambda function demonstrates how to gradually transfer traffic from
    one S3 bucket to another in a controlled way.
    We define a variable BLUE_TRAFFIC_PERCENTAGE which can take values from
    1 to 100. If the generated randomNumber less than or equal to
    BLUE_TRAFFIC_PERCENTAGE, traffic
    is re-directed to blue-bucket. If not, the default bucket that we've configured
    is used.
    ...

    randomNumber = getRandomInt(1, 100)

    if randomNumber <= BLUE_TRAFFIC_PERCENTAGE:
        domainName = 'blue-bucket.s3.amazonaws.com'
        request['origin']['s3']['domainName'] = domainName
        request['headers']['host'] = [{'key': 'host', 'value': domainName}]

    return request

```

Contoh: Gunakan pemicu permintaan asal untuk mengubah nama domain asal berdasarkan header negara

Fungsi ini menunjukkan bagaimana Anda dapat mengubah nama domain asal berdasarkan CloudFront-Viewer-Country header, sehingga konten disajikan dari asal yang lebih dekat ke negara pemirsa.

Mengimplementasikan fungsi ini untuk distribusi Anda dapat memiliki keuntungan seperti berikut ini:

- Mengurangi keterlambatan jika Wilayah yang ditentukan lebih dekat dengan negara penampil
- Memberikan kedaulatan data dengan memastikan bahwa data tersebut berasal dari negara yang sama dengan negara asal permintaan

Perhatikan bahwa untuk mengaktifkan fungsi ini, Anda harus mengonfigurasi distribusi Anda ke cache berdasarkan CloudFront-Viewer-Country header. Untuk informasi selengkapnya, lihat [the section called “Cache berdasarkan header permintaan yang dipilih”](#).

Node.js

```
'use strict';

exports.handler = (event, context, callback) => {
    const request = event.Records[0].cf.request;

    if (request.headers['cloudfront-viewer-country']) {
        const countryCode = request.headers['cloudfront-viewer-country'][0].value;
        if (countryCode === 'GB' || countryCode === 'DE' || countryCode === 'IE' )
        {
            const domainName = 'eu.example.com';
            request.origin.custom.domainName = domainName;
            request.headers['host'] = [{key: 'host', value: domainName}];
        }
    }

    callback(null, request);
};
```

Python

```
def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']

    viewerCountry = request['headers'].get('cloudfront-viewer-country')
    if viewerCountry:
        countryCode = viewerCountry[0]['value']
        if countryCode == 'GB' or countryCode == 'DE' or countryCode == 'IE':
            domainName = 'eu.example.com'
            request['origin']['custom']['domainName'] = domainName
            request['headers']['host'] = [{'key': 'host', 'value': domainName}]
    return request
```


Perbarui status kesalahan - contoh

Contoh berikut memberikan panduan bagaimana Anda dapat menggunakan Lambda @Edge untuk mengubah status kesalahan yang dikembalikan ke pengguna.

Topik

- [Contoh: Gunakan pemicu respons asal untuk memperbarui kode status kesalahan ke 200](#)
- [Contoh: Gunakan pemicu respons asal untuk memperbarui kode status kesalahan ke 302](#)

Contoh: Gunakan pemicu respons asal untuk memperbarui kode status kesalahan ke 200

Fungsi ini menunjukkan bagaimana Anda dapat memperbarui status respons menjadi 200 dan menghasilkan konten tubuh statis untuk kembali ke penampil dalam skenario berikut:

- Fungsi dipicu dalam respons asal.
- Status respons dari server asal adalah kode status kesalahan (4xx atau 5xx).

Node.js

```
'use strict';

exports.handler = (event, context, callback) => {
  const response = event.Records[0].cf.response;

  /**
   * This function updates the response status to 200 and generates static
   * body content to return to the viewer in the following scenario:
   * 1. The function is triggered in an origin response
   * 2. The response status from the origin server is an error status code (4xx or
5xx)
   */

  if (response.status >= 400 && response.status <= 599) {
    response.status = 200;
    response.statusDescription = 'OK';
    response.body = 'Body generation example';
  }

  callback(null, response);
};
```

Python

```
def lambda_handler(event, context):
    response = event['Records'][0]['cf']['response']

    '''
    This function updates the response status to 200 and generates static
    body content to return to the viewer in the following scenario:
    1. The function is triggered in an origin response
    2. The response status from the origin server is an error status code (4xx or
    5xx)
    '''

    if int(response['status']) >= 400 and int(response['status']) <= 599:
        response['status'] = 200
        response['statusDescription'] = 'OK'
        response['body'] = 'Body generation example'
    return response
```

Contoh: Gunakan pemicu respons asal untuk memperbarui kode status kesalahan ke 302

Fungsi ini menunjukkan cara Anda dapat memperbarui kode status HTTP ke 302 untuk mengalihkan ke jalur lain (perilaku kesalahan) yang memiliki asal berbeda yang dikonfigurasi. Perhatikan hal-hal berikut:

- Fungsi dipicu dalam respons asal.
- Status respons dari server asal adalah kode status kesalahan (4xx atau 5xx).

Node.js

```
'use strict';

exports.handler = (event, context, callback) => {
    const response = event.Records[0].cf.response;
    const request = event.Records[0].cf.request;

    /**
     * This function updates the HTTP status code in the response to 302, to
     * redirect to another
     * path (cache behavior) that has a different origin configured. Note the
     * following:
```

```

    * 1. The function is triggered in an origin response
    * 2. The response status from the origin server is an error status code (4xx or
5xx)
    */

    if (response.status >= 400 && response.status <= 599) {
        const redirect_path = `/plan-b/path?${request.querystring}`;

        response.status = 302;
        response.statusDescription = 'Found';

        /* Drop the body, as it is not required for redirects */
        response.body = '';
        response.headers['location'] = [{ key: 'Location', value: redirect_path }];
    }

    callback(null, response);
};

```

Python

```

def lambda_handler(event, context):
    response = event['Records'][0]['cf']['response']
    request = event['Records'][0]['cf']['request']

    ...

    This function updates the HTTP status code in the response to 302, to redirect
to another
    path (cache behavior) that has a different origin configured. Note the
following:
    1. The function is triggered in an origin response
    2. The response status from the origin server is an error status code (4xx or
5xx)
    ...

    if int(response['status']) >= 400 and int(response['status']) <= 599:
        redirect_path = '/plan-b/path?%s' % request['querystring']

        response['status'] = 302
        response['statusDescription'] = 'Found'

        # Drop the body as it is not required for redirects
        response['body'] = ''

```

```
response['headers']['location'] = [{'key': 'Location', 'value':  
redirect_path}]  
  
return response
```

Akses badan permintaan - contoh

Contoh berikut menunjukkan bagaimana Anda dapat menggunakan Lambda @Edge untuk bekerja dengan permintaan POST.

Note

Untuk menggunakan contoh ini, Anda harus mengaktifkan opsi include body dalam asosiasi fungsi Lambda distribusi. Ini tidak diaktifkan secara default.

- Untuk mengaktifkan pengaturan ini di CloudFront konsol, pilih kotak centang untuk Sertakan Tubuh di Asosiasi Fungsi Lambda.
- Untuk mengaktifkan setelan ini di CloudFront API atau dengan CloudFormation, setel IncludeBody bidang ke true in LambdaFunctionAssociation.

Topik

- [Contoh: Gunakan pemicu permintaan untuk membaca formulir HTML](#)
- [Contoh: Gunakan pemicu permintaan untuk memodifikasi formulir HTML](#)

Contoh: Gunakan pemicu permintaan untuk membaca formulir HTML

Fungsi ini menunjukkan bagaimana Anda dapat memproses isi permintaan POST yang dibuat oleh formulir HTML (formulir web), seperti formulir “hubungi kami”. Misalnya, Anda mungkin memiliki formulir HTML seperti berikut ini:

```
<html>  
  <form action="https://example.com" method="post">  
    Param 1: <input type="text" name="name1"><br>  
    Param 2: <input type="text" name="name2"><br>  
    input type="submit" value="Submit">  
  </form>  
</html>
```

Untuk fungsi contoh berikut, fungsi harus dipicu dalam CloudFront permintaan penampil atau permintaan asal.

Node.js

```
'use strict';

const querystring = require('querystring');

/**
 * This function demonstrates how you can read the body of a POST request
 * generated by an HTML form (web form). The function is triggered in a
 * CloudFront viewer request or origin request event type.
 */

exports.handler = (event, context, callback) => {
  const request = event.Records[0].cf.request;

  if (request.method === 'POST') {
    /* HTTP body is always passed as base64-encoded string. Decode it. */
    const body = Buffer.from(request.body.data, 'base64').toString();

    /* HTML forms send the data in query string format. Parse it. */
    const params = querystring.parse(body);

    /* For demonstration purposes, we only log the form fields here.
     * You can put your custom logic here. For example, you can store the
     * fields in a database, such as Amazon DynamoDB, and generate a response
     * right from your Lambda@Edge function.
     */
    for (let param in params) {
      console.log(`For "${param}" user submitted "${params[param]}".\n`);
    }
  }
  return callback(null, request);
};
```

Python

```
import base64
from urllib.parse import parse_qs

...
```

Say there is a POST request body generated by an HTML such as:

```
<html>
<form action="https://example.com" method="post">
  Param 1: <input type="text" name="name1"><br>
  Param 2: <input type="text" name="name2"><br>
  input type="submit" value="Submit">
</form>
</html>
```

```
...
```

```
...
```

This function demonstrates how you can read the body of a POST request generated by an HTML form (web form). The function is triggered in a CloudFront viewer request or origin request event type.

```
...
```

```
def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']

    if request['method'] == 'POST':
        # HTTP body is always passed as base64-encoded string. Decode it
        body = base64.b64decode(request['body']['data'])

        # HTML forms send the data in query string format. Parse it
        params = {k: v[0] for k, v in parse_qs(body).items()}

        ...

        For demonstration purposes, we only log the form fields here.
        You can put your custom logic here. For example, you can store the
        fields in a database, such as Amazon DynamoDB, and generate a response
        right from your Lambda@Edge function.
        ...

        for key, value in params.items():
            print("For %s use submitted %s" % (key, value))

    return request
```

Contoh: Gunakan pemicu permintaan untuk memodifikasi formulir HTML

Fungsi ini menunjukkan bagaimana Anda dapat memodifikasi isi permintaan POST yang dibuat oleh formulir HTML (formulir web). Fungsi ini dipicu dalam permintaan CloudFront penampil atau permintaan asal.

Node.js

```
'use strict';

const querystring = require('querystring');

exports.handler = (event, context, callback) => {
  var request = event.Records[0].cf.request;
  if (request.method === 'POST') {
    /* Request body is being replaced. To do this, update the following
    /* three fields:
    *   1) body.action to 'replace'
    *   2) body.encoding to the encoding of the new data.
    *
    *       Set to one of the following values:
    *
    *       text - denotes that the generated body is in text format.
    *             Lambda@Edge will propagate this as is.
    *       base64 - denotes that the generated body is base64 encoded.
    *             Lambda@Edge will base64 decode the data before sending
    *             it to the origin.
    *   3) body.data to the new body.
    */
    request.body.action = 'replace';
    request.body.encoding = 'text';
    request.body.data = getUpdatedBody(request);
  }
  callback(null, request);
};

function getUpdatedBody(request) {
  /* HTTP body is always passed as base64-encoded string. Decode it. */
  const body = Buffer.from(request.body.data, 'base64').toString();

  /* HTML forms send data in query string format. Parse it. */
  const params = querystring.parse(body);
```

```

/* For demonstration purposes, we're adding one more param.
 *
 * You can put your custom logic here. For example, you can truncate long
 * bodies from malicious requests.
 */
params['new-param-name'] = 'new-param-value';
return querystring.stringify(params);
}

```

Python

```

import base64
from urllib.parse import parse_qs, urlencode

def lambda_handler(event, context):
    request = event['Records'][0]['cf']['request']
    if request['method'] == 'POST':
        ...

        Request body is being replaced. To do this, update the following
        three fields:
            1) body.action to 'replace'
            2) body.encoding to the encoding of the new data.

            Set to one of the following values:

                text - denotes that the generated body is in text format.
                    Lambda@Edge will propagate this as is.
                base64 - denotes that the generated body is base64 encoded.
                    Lambda@Edge will base64 decode the data before sending
                    it to the origin.
            3) body.data to the new body.
        ...

        request['body']['action'] = 'replace'
        request['body']['encoding'] = 'text'
        request['body']['data'] = getUpdatedBody(request)
    return request

def getUpdatedBody(request):
    # HTTP body is always passed as base64-encoded string. Decode it
    body = base64.b64decode(request['body']['data'])

    # HTML forms send data in query string format. Parse it
    params = {k: v[0] for k, v in parse_qs(body).items()}

```



```
# For demonstration purposes, we're adding one more param

# You can put your custom logic here. For example, you can truncate long
# bodies from malicious requests
params['new-param-name'] = 'new-param-value'
return urlencode(params)
```

Pembatasan pada fungsi edge

Topik berikut menjelaskan batasan yang berlaku untuk CloudFront Fungsi dan Lambda @Edge. Beberapa batasan berlaku untuk semua fungsi tepi, sementara yang lain hanya berlaku untuk CloudFront Fungsi atau Lambda @Edge.

Setiap topik memberikan informasi terperinci tentang batasan dan kendala yang harus Anda pertimbangkan saat mengembangkan dan menerapkan fungsi edge. CloudFront

Memahami batasan ini membantu Anda memastikan bahwa fungsi edge Anda beroperasi seperti yang diharapkan dan mematuhi fitur yang didukung.

Topik

- [Pembatasan pada semua fungsi edge](#)
- [Pembatasan CloudFront Fungsi](#)
- [Pembatasan Lambda@Edge](#)

Untuk informasi tentang kuota (sebelumnya disebut sebagai batas), lihat [Kuota pada Fungsi CloudFront](#) dan [Kuotas di Lambda@Edge](#).

Pembatasan pada semua fungsi edge

Pembatasan berikut berlaku untuk semua fungsi edge, baik CloudFront Fungsi maupun Lambda @Edge.

Topik

- [Akun AWS kepemilikan](#)
- [Menggabungkan CloudFront Fungsi dengan Lambda @Edge](#)
- [Kode status HTTP](#)

- [Header HTTP](#)
- [String pertanyaan](#)
- [URI](#)
- [URI, string kueri, dan pengkodean header](#)
- [Streaming Microsoft yang Lancar](#)
- [Penandaan](#)

Akun AWS kepemilikan

Untuk mengaitkan fungsi tepi dengan CloudFront distribusi, fungsi dan distribusi harus dimiliki oleh yang sama Akun AWS.

Menggabungkan CloudFront Fungsi dengan Lambda @Edge

Untuk perilaku cache tertentu, pembatasan berikut berlaku:

- Setiap jenis kejadian (permintaan penampil, permintaan asal, respons asal, dan respons penampil) hanya dapat memiliki satu asosiasi fungsi edge.
- Anda tidak dapat menggabungkan CloudFront Fungsi dan Lambda @Edge dalam acara penampil (permintaan penampil dan respons penampil).

Semua kombinasi fungsi edge lainnya diperbolehkan. Tabel berikut menjelaskan kombinasi yang diizinkan.

		CloudFront Fungsi	
		Permintaan penampil	Respons pemirsa
Lambda@Edge	Permintaan penampil	Tidak diizinkan	Tidak diizinkan
	Permintaan asal	Diizinkan	Diizinkan
	Respon asal	Diizinkan	Diizinkan
	Respons pemirsa	Tidak diizinkan	Tidak diizinkan

Kode status HTTP

CloudFront tidak memanggil fungsi tepi untuk peristiwa respons penampil saat asal mengembalikan kode status HTTP 400 atau lebih tinggi.

Fungsi Lambda@Edge untuk peristiwa respons asal dipanggil untuk semua respons asal, termasuk saat asal mengembalikan kode status HTTP 400 atau lebih tinggi. Untuk informasi selengkapnya, lihat [Perbarui tanggapan HTTP di pemicu respons asal](#).

Header HTTP

Header HTTP tertentu tidak diizinkan, yang berarti mereka tidak terkena fungsi tepi dan fungsi tidak dapat menambahkannya. Header lainnya hanya baca, yang berarti fungsi dapat membacanya tetapi tidak dapat menambahkan, memodifikasi, atau menghapusnya.

Topik

- [Header yang diizinkan](#)
- [Header hanya-baca](#)

Header yang diizinkan

Header HTTP berikut tidak terkena fungsi edge, dan fungsi tidak dapat menambahkannya. Jika fungsi Anda menambahkan salah satu header ini, itu gagal CloudFront validasi dan CloudFront mengembalikan kode status HTTP 502 (Bad Gateway) ke penampil.

- Connection
- Expect
- Keep-Alive
- Proxy-Authenticate
- Proxy-Authorization
- Proxy-Connection
- Trailer
- Upgrade
- X-Accel-Buffering
- X-Accel-Charset

- X-Accel-Limit-Rate
- X-Accel-Redirect
- X-Amz-Cf-*
- X-Amzn-Auth
- X-Amzn-Cf-Billing
- X-Amzn-Cf-Id
- X-Amzn-Cf-Xff
- X-Amzn-Errortype
- X-Amzn-Fle-Profile
- X-Amzn-Header-Count
- X-Amzn-Header-Order
- X-Amzn-Lambda-Integration-Tag
- X-Amzn-RequestId
- X-Cache
- X-Edge-*
- X-Forwarded-Proto
- X-Real-IP

Header hanya-baca

Header berikut ini hanya untuk dibaca. Fungsi Anda dapat membacanya dan menggunakannya sebagai input ke logika fungsi, tetapi tidak dapat mengubah nilainya. Jika fungsi Anda menambahkan atau mengedit header hanya-baca, permintaan gagal CloudFront validasi dan CloudFront mengembalikan kode status HTTP 502 (Bad Gateway) ke penampil.

Header hanya-baca dalam peristiwa permintaan penampil

Header berikut ini hanya-baca di peristiwa permintaan penampil.

- CDN-Loop
- Content-Length
- Host

- Transfer-Encoding
- Via

Header hanya-baca dalam peristiwa permintaan asal (hanya Lambda@Edge)

Header berikut hanya-baca dalam peristiwa permintaan asal, yang hanya ada di Lambda@Edge.

- Accept-Encoding
- CDN-Loop
- Content-Length
- If-Modified-Since
- If-None-Match
- If-Range
- If-Unmodified-Since
- Transfer-Encoding
- Via

Header hanya-baca dalam peristiwa respons asal (hanya Lambda@Edge)

Header berikut hanya-baca dalam peristiwa respon asal, yang hanya ada di Lambda@Edge.

- Transfer-Encoding
- Via

Header hanya-baca dalam peristiwa respons penampil

Header berikut hanya-baca dalam peristiwa respons penampil untuk Fungsi CloudFront dan Lambda @Edge.

- Warning
- Via

Header berikut hanya-baca dalam peristiwa respons penampil untuk Lambda @Edge.

- Content-Length

- Content-Encoding
- Transfer-Encoding

String pertanyaan

Pembatasan berikut berlaku untuk fungsi yang membaca, memperbarui, atau membuat string kueri dalam permintaan URI.

- (Lambda@Edge saja) Untuk mengakses string kueri dalam permintaan asal atau fungsi respons asal, kebijakan cache atau kebijakan permintaan asal harus disetel ke semua untuk string kueri.
- Sebuah fungsi dapat membuat atau memperbarui string kueri untuk permintaan penampil dan peristiwa permintaan asal (peristiwa permintaan asal hanya ada di Lambda@Edge).
- Sebuah fungsi dapat membaca string kueri, tetapi tidak dapat membuat atau memperbarui satu, untuk respons asal dan peristiwa respons penampil (peristiwa respons asal hanya ada di Lambda@Edge).
- Jika fungsi membuat atau memperbarui string kueri, pembatasan berikut berlaku:
 - String kueri tidak dapat menyertakan spasi, karakter kontrol, atau pengenalan fragmen (#).
 - Ukuran total URI, termasuk string kueri, harus kurang dari 8.192 karakter.
 - Kami sarankan Anda menggunakan persen pengkodean untuk URI dan string kueri. Untuk informasi selengkapnya, lihat [URI, string kueri, dan pengkodean header](#).

URI

Jika sebuah fungsi mengubah URI untuk permintaan, hal itu tidak mengubah perilaku cache untuk permintaan tersebut atau asal permintaan tersebut yang diteruskan.

Ukuran total URI, termasuk string kueri, harus kurang dari 8.192 karakter.

URI, string kueri, dan pengkodean header

Nilai untuk URI, string kueri, dan header yang diteruskan ke fungsi edge dikodekan UTF-8. Fungsi Anda harus menggunakan pengkodean UTF-8 untuk URI, string kueri, dan nilai header yang dikembalikan. Pengkodean persen kompatibel dengan pengkodean UTF-8.

Daftar berikut menjelaskan cara CloudFront menangani pengkodean untuk URI, string kueri, dan header:

- Ketika nilai dalam permintaan dikodekan UTF-8, teruskan nilai CloudFront ke fungsi Anda tanpa mengubahnya.
- Ketika nilai dalam permintaan dikodekan [ISO-8859-1, CloudFront konversi nilai ke UTF-8 encoding](#) sebelum meneruskannya ke fungsi Anda.
- Ketika nilai dalam permintaan dikodekan menggunakan beberapa pengkodean karakter lain, CloudFront asumsikan bahwa mereka dikodekan ISO-8859-1 dan mencoba mengonversi dari ISO-8859-1 ke UTF-8.

Important

Karakter yang dikonversi mungkin merupakan interpretasi yang tidak akurat dari nilai-nilai dalam permintaan asli. Hal ini dapat menyebabkan fungsi atau asal Anda memberi hasil yang tidak diinginkan.

Nilai untuk URI, string kueri, dan header yang CloudFront diteruskan ke asal Anda bergantung pada apakah suatu fungsi mengubah nilai:

- Jika fungsi tidak mengubah URI, string kueri, atau header, CloudFront teruskan nilai yang diterimanya dalam permintaan ke asal Anda.
- Jika fungsi mengubah URI, string kueri, atau header, CloudFront teruskan nilai yang dikodekan UTF-8.

Streaming Microsoft yang Lancar

Anda tidak dapat menggunakan fungsi edge dengan CloudFront distribusi yang Anda gunakan untuk streaming file media yang telah Anda transkode ke dalam format Microsoft Smooth Streaming.

Penandaan

Anda tidak dapat menambahkan tag ke fungsi tepi. Untuk informasi selengkapnya tentang penandaan CloudFront, lihat [Menandai CloudFront sumber daya](#).

Pembatasan CloudFront Fungsi

Pembatasan berikut hanya berlaku untuk CloudFront Fungsi.

Daftar Isi

- [Beberapa catatan](#)
- [Isi permintaan](#)
- [Menggunakan kredensial sementara dengan API CloudFront KeyValueStore](#)
- [Waktu Aktif](#)
- [Hitung pemanfaatan](#)

Untuk informasi tentang kuota (sebelumnya disebut sebagai batas), lihat. [Kuota pada Fungsi CloudFront](#)

Beberapa catatan

Log fungsi di CloudFront Fungsi terpotong pada 10 KB.

Isi permintaan

CloudFront Fungsi tidak dapat mengakses isi permintaan HTTP.

Menggunakan kredensial sementara dengan API CloudFront KeyValueStore

Anda dapat menggunakan AWS Security Token Service (AWS STS) untuk menghasilkan kredensial keamanan sementara (juga dikenal sebagai token sesi). Token sesi memungkinkan Anda untuk sementara mengambil peran AWS Identity and Access Management (IAM) sehingga Anda dapat mengakses Layanan AWS.

Untuk memanggil [CloudFront KeyValueStore API](#), gunakan titik akhir Regional AWS STS untuk mengembalikan token sesi versi 2. Jika Anda menggunakan endpoint global for AWS STS (`sts.amazonaws.com`), AWS STS akan menghasilkan token sesi versi 1, yang tidak didukung oleh Signature Version 4A (Sigv4a). Akibatnya, Anda akan menerima kesalahan otentikasi.

Untuk memanggil CloudFront KeyValueStore API, Anda dapat menggunakan opsi berikut:

AWS CLI dan AWS SDKs

Anda dapat mengonfigurasi AWS CLI atau AWS SDK untuk menggunakan titik AWS STS akhir Regional. Untuk informasi selengkapnya, lihat [Titik akhir AWS STS regional](#) di AWS SDK dan Panduan Referensi Alat.

Untuk informasi selengkapnya tentang AWS STS titik akhir yang tersedia, lihat [Wilayah dan titik akhir](#) di Panduan Pengguna IAM.

SAML

Anda dapat mengkonfigurasi SAFL untuk menggunakan AWS STS endpoint Regional. Untuk informasi selengkapnya, lihat [Cara menggunakan titik akhir SAM regional untuk posting blog failover](#).

API `SetSecurityTokenServicePreferences`

Alih-alih menggunakan AWS STS titik akhir Regional, Anda dapat mengonfigurasi titik akhir global AWS STS untuk mengembalikan token sesi versi 2. Untuk melakukannya, gunakan operasi [SetSecurityTokenServicePreferences](#) API untuk mengonfigurasi Akun AWS.

Example Contoh: perintah IAM CLI

```
aws iam set-security-token-service-preferences --global-endpoint-token-version v2Token
```

Tip

Kami menyarankan Anda menggunakan titik akhir AWS STS Regional alih-alih opsi ini. Titik akhir regional memberikan ketersediaan dan skenario failover yang lebih tinggi.

Penyedia identitas khusus

Jika Anda menggunakan penyedia identitas khusus yang melakukan federasi dan mengambil peran, gunakan salah satu opsi sebelumnya untuk sistem penyedia identitas induk yang bertanggung jawab untuk membuat token sesi.

Waktu Aktif

Lingkungan runtime CloudFront Functions tidak mendukung evaluasi kode dinamis, dan membatasi akses ke jaringan, sistem file, variabel lingkungan, dan timer. Untuk informasi selengkapnya, lihat [Fitur yang dibatasi](#).

Note

Untuk menggunakannya CloudFront KeyValueCollection, CloudFront fungsi Anda harus menggunakan [JavaScript runtime 2.0](#).

Hitung pemanfaatan

CloudFront Fungsi memiliki batas waktu yang dibutuhkan untuk menjalankan, diukur sebagai pemanfaatan komputasi. Pemanfaatan komputasi adalah angka antara 0 dan 100 yang menunjukkan jumlah waktu yang fungsi butuhkan untuk berjalan sebagai persentase dari waktu maksimum yang diizinkan. Misalnya, pemanfaatan komputasi 35 berarti bahwa fungsi selesai pada 35% dari waktu maksimum yang diizinkan.

Saat Anda [menguji fungsi](#), Anda dapat melihat nilai pemanfaatan komputasi dalam output dari peristiwa uji. Untuk fungsi produksi, Anda dapat melihat [metrik pemanfaatan komputasi](#) pada [halaman Pemantauan di CloudFront konsol](#), atau di CloudWatch

Pembatasan Lambda@Edge

Pembatasan berikut hanya berlaku untuk Lambda@Edge.

Daftar Isi

- [Resolusi DNS](#)
- [Kode status HTTP](#)
- [Versi fungsi Lambda](#)
- [Wilayah Lambda](#)
- [Izin peran Lambda](#)
- [Fitur Lambda](#)
- [Waktu aktif yang didukung](#)
- [CloudFront header](#)
- [Pembatasan pada isi permintaan dengan opsi sertakan isi](#)
- [Batas waktu respons dan batas waktu tetap hidup \(hanya asal khusus\)](#)

Untuk informasi tentang kuota, lihat [Kuotas di Lambda@Edge](#).

Resolusi DNS

CloudFront melakukan resolusi DNS pada nama domain asal sebelum menjalankan fungsi permintaan asal Anda Lambda @Edge. Jika layanan DNS untuk domain Anda mengalami masalah dan tidak CloudFront dapat menyelesaikan nama domain untuk mendapatkan alamat

IP, fungsi Lambda @Edge Anda tidak akan dipanggil. CloudFront akan mengembalikan [kode status HTTP 502 \(Bad Gateway\)](#) ke klien. Untuk informasi selengkapnya, lihat [Kesalahan DNS \(\) NonS3OriginDnsError](#).

Jika logika fungsi Anda memodifikasi nama domain asal, CloudFront akan melakukan resolusi DNS lain pada nama domain yang diperbarui setelah fungsi selesai dijalankan.

Untuk informasi selengkapnya tentang mengelola failover DNS, lihat [Mengonfigurasi failover DNS](#) di Panduan Pengembang Amazon Route 53.

Kode status HTTP

Fungsi Lambda @Edge untuk peristiwa respons penampil tidak dapat mengubah kode status HTTP respons, terlepas dari apakah respons berasal dari asal atau cache. CloudFront

Versi fungsi Lambda

Anda harus menggunakan versi bernomor dari fungsi Lambda, bukan \$LATEST atau alias.

Wilayah Lambda

Fungsi Lambda harus di Wilayah US East (N. Virginia).

Izin peran Lambda

Peran eksekusi IAM yang terkait dengan fungsi Lambda harus mengizinkan layanan utama `lambda.amazonaws.com` dan `edgelambda.amazonaws.com` untuk menjalankan peran tersebut. Untuk informasi selengkapnya, lihat [Siapkan izin dan peran IAM untuk Lambda @Edge](#).

Fitur Lambda

Fitur Lambda berikut tidak didukung oleh Lambda@Edge:

- [Konfigurasi manajemen runtime Lambda selain Auto](#) (default)
- Konfigurasi fungsi Lambda Anda untuk mengakses sumber daya di dalam VPC Anda
- [Lambda berfungsi antrian surat mati](#)
- [Variabel lingkungan Lambda](#) (kecuali untuk variabel lingkungan cadangan, yang didukung secara otomatis)

- Lambda berfungsi dengan [Mengelola AWS Lambda dependensi](#) dengan lapisan
- [Menggunakan AWS X-Ray](#)
- Konkurensi terprovisi Lambda

Note

Fungsi Lambda @Edge memiliki kemampuan [konkurensi Regional](#) yang sama dengan semua fungsi Lambda. Untuk informasi selengkapnya, lihat [Kuotas di Lambda@Edge](#).

- [Buat fungsi Lambda menggunakan gambar kontainer](#)
- [Fungsi Lambda yang menggunakan arsitektur arm64](#)
- Lambda berfungsi dengan penyimpanan fana lebih dari 512 MB
- Menggunakan [kunci terkelola pelanggan untuk mengenkripsi paket penerapan.zip Anda](#)

Waktu aktif yang didukung

Lambda @Edge mendukung runtime Node.js dan Python versi terbaru. Untuk daftar versi yang didukung dan tanggal penghentian masa depannya, lihat [Runtime yang didukung](#) di Panduan Pengembang AWS Lambda

Tip

- Sebagai praktik terbaik, gunakan versi terbaru dari runtime yang disediakan untuk peningkatan kinerja dan fitur baru.
- Anda tidak dapat membuat atau memperbarui fungsi dengan versi Node.js yang tidak digunakan lagi. Anda hanya dapat mengaitkan fungsi yang ada dengan versi ini dengan CloudFront distribusi. Fungsi dengan versi ini yang terkait dengan distribusi akan terus berjalan. Namun, kami menyarankan Anda memindahkan fungsi Anda ke versi Node.js yang lebih baru. Untuk informasi selengkapnya, [lihat kebijakan penghentian waktu](#) proses di Panduan AWS Lambda Pengembang dan jadwal rilis [Node.js](#) pada. GitHub

CloudFront header

Fungsi Lambda @Edge dapat membaca, mengedit, menghapus, atau menambahkan CloudFront header apa pun yang tercantum di. [Tambahkan header CloudFront permintaan](#)

Catatan

- Jika Anda CloudFront ingin menambahkan header ini, Anda harus mengonfigurasi CloudFront untuk menambahkannya dengan menggunakan [kebijakan cache atau kebijakan permintaan asal](#).
- CloudFront menambahkan header setelah peristiwa permintaan penampil, yang berarti header tidak tersedia untuk fungsi Lambda @Edge dalam permintaan penampil. Header hanya tersedia untuk fungsi Lambda @Edge dalam permintaan asal dan respons asal.
- Jika permintaan penampil menyertakan header yang memiliki nama ini, dan Anda mengonfigurasi CloudFront untuk menambahkan header ini menggunakan [kebijakan cache atau kebijakan permintaan asal](#), maka CloudFront timpa nilai header yang ada dalam permintaan penampil. Fungsi yang menghadap pemirsa melihat nilai header dari permintaan penampil, sementara fungsi yang menghadap ke asal melihat nilai header yang ditambahkan. CloudFront
- Jika fungsi permintaan penampil menambahkan CloudFront-Viewer-Country header, itu gagal validasi dan CloudFront mengembalikan kode status HTTP 502 (Bad Gateway) ke penampil.

Pembatasan pada isi permintaan dengan opsi sertakan isi

Saat Anda memilih opsi Sertakan Tubuh untuk mengekspos badan permintaan ke fungsi Lambda @Edge Anda, informasi dan batas ukuran berikut berlaku untuk bagian tubuh yang diekspose atau diganti.

- CloudFront selalu base64 mengkodekan badan permintaan sebelum memaparkannya ke Lambda @Edge.
- Jika badan permintaan besar, CloudFront potong sebelum memaparkannya ke Lambda @Edge, sebagai berikut:
 - Untuk peristiwa permintaan penampil, isi dipotong pada 40 KB.
 - Untuk peristiwa permintaan asal, isi dipotong pada 1 MB.
- Jika Anda mengakses isi permintaan sebagai hanya-baca, CloudFront kirimkan badan permintaan asli lengkap ke asal.
- Jika fungsi Lambda @Edge Anda menggantikan badan permintaan, batas ukuran berikut berlaku untuk badan yang dikembalikan fungsi:

- Jika fungsi Lambda@Edge mengembalikan isi sebagai teks biasa:
 - Untuk acara permintaan pemirsa, batas isi adalah 40 KB.
 - Untuk acara permintaan asal, batas isi adalah 1 MB.
- Jika fungsi Lambda@Edge mengembalikan isi sebagai teks berenkode base64:
 - Untuk acara permintaan pemirsa, batas isi adalah 53,2 KB.
 - Untuk acara permintaan asal, batas isi adalah 1,33 MB.

Note

Jika fungsi Lambda @Edge Anda mengembalikan badan yang melebihi batas ini, permintaan Anda akan gagal dengan kode status HTTP 502 (). [Kesalahan validasi Lambda](#) Kami menyarankan Anda memperbarui fungsi Lambda @Edge Anda sehingga tubuh tidak melebihi batas ini.

Batas waktu respons dan batas waktu tetap hidup (hanya asal khusus)

Jika Anda menggunakan fungsi Lambda @Edge untuk mengatur batas waktu respons atau batas waktu keep-alive untuk asal distribusi, verifikasi bahwa Anda menentukan nilai yang dapat didukung oleh asal Anda. Lihat informasi yang lebih lengkap di [Tanggapan dan kuota batas waktu tetap hidup](#).

Laporan, metrik, dan log

CloudFront menyediakan beberapa opsi untuk melaporkan, memantau, dan mencatat CloudFront sumber daya Anda. Anda dapat menyelesaikan tugas-tugas berikut:

- Lihat dan unduh laporan untuk melihat penggunaan dan aktivitas CloudFront distribusi Anda, termasuk laporan penagihan, statistik cache, konten populer, dan perujuk teratas.
- Pantau dan lacak CloudFront, termasuk [fungsi komputasi tepi](#) Anda, langsung di CloudFront konsol atau dengan menggunakan Amazon CloudWatch. CloudFront mengirimkan metrik ke CloudWatch untuk distribusi dan fungsi edge, baik Lambda @Edge dan Functions. CloudFront
- Lihat log untuk permintaan penampil yang diterima CloudFront distribusi Anda dengan log standar atau log akses real-time. Selain log permintaan penampil, Anda dapat menggunakan CloudWatch Log untuk mendapatkan log untuk fungsi edge Anda, baik Lambda @Edge maupun CloudFront Functions. Anda juga dapat menggunakan AWS CloudTrail untuk mendapatkan log aktivitas CloudFront API di Akun AWS.
- Lacak perubahan konfigurasi ke CloudFront sumber daya Anda menggunakan AWS Config.

Untuk informasi selengkapnya tentang opsi ini, lihat topik berikut.

Topik

- [AWS laporan penagihan dan penggunaan untuk CloudFront](#)
- [Lihat CloudFront laporan di konsol](#)
- [Pantau CloudFront metrik dengan Amazon CloudWatch](#)
- [CloudFront dan logging fungsi tepi](#)
- [Lacak perubahan konfigurasi dengan AWS Config](#)

AWS laporan penagihan dan penggunaan untuk CloudFront

AWS menyediakan dua laporan penggunaan untuk CloudFront:

- Laporan AWS penagihan adalah tampilan tingkat tinggi dari semua aktivitas Layanan AWS yang Anda gunakan, termasuk. CloudFront

- Laporan AWS penggunaan adalah ringkasan aktivitas untuk layanan tertentu, digabungkan berdasarkan jam, hari, atau bulan. Ini juga mencakup grafik penggunaan yang memberikan representasi grafis dari CloudFront penggunaan Anda.

Note

Seperti yang lain Layanan AWS, CloudFront menagih Anda hanya untuk apa yang Anda gunakan. Untuk informasi selengkapnya, lihat [harga CloudFront](#).

Topik

- [Lihat laporan AWS penagihan untuk CloudFront](#)
- [Lihat laporan AWS penggunaan untuk CloudFront](#)
- [Menafsirkan laporan AWS tagihan dan penggunaan Anda untuk CloudFront](#)

Lihat laporan AWS penagihan untuk CloudFront

Anda dapat melihat ringkasan AWS penggunaan dan biaya Anda, yang tercantum berdasarkan layanan, di halaman Tagihan di AWS Manajemen Penagihan dan Biaya konsol.

Untuk melihat laporan AWS penagihan

1. Masuk ke Konsol Manajemen AWS dan buka AWS Manajemen Penagihan dan Biaya konsol di <https://console.aws.amazon.com/costmanagement/>.
2. Di panel navigasi, pilih Tagihan.
3. Pilih periode Penagihan (misalnya, Agustus 2023).
4. Pada tab Biaya berdasarkan layanan, pilih CloudFront, lalu perluas Global atau Wilayah AWS namanya.
5. Untuk mengunduh laporan penagihan terperinci dalam format CSV, pilih Unduh semua ke CSV.

Untuk informasi selengkapnya tentang AWS tagihan [Anda, lihat Melihat tagihan Anda](#) di Panduan AWS Billing Pengguna.

Laporan penagihan mencakup nilai-nilai berikut yang berlaku untuk CloudFront:

- ProductCode – AmazonCloudFront

- **UsageType** – Salah satu nilai berikut:
 - Kode yang mengidentifikasi jenis transfer data
 - Invalidations
 - Executions-CloudFrontFunctions
 - KeyValueStore-APIOperations
 - KeyValueStore-EdgeReads
 - RealTimeLog-KinesisDataStream
 - SSL-Cert-Custom
- **ItemDescription**— Deskripsi tarif penagihan untuk **UsageType**
- **UsageStart Tanggal dan UsageEndDate**— Hari dimana penggunaan berlaku, dalam Coordinated Universal Time (UTC).
- **UsageQuantity** – Salah satu nilai berikut:
 - Jumlah permintaan selama periode waktu yang ditentukan
 - Jumlah data yang ditransfer dalam gigabyte
 - Jumlah objek yang tidak valid
 - Jumlah bulan prorata bahwa Anda memiliki sertifikat SSL yang terkait dengan distribusi yang diaktifkan. CloudFront Misalnya, jika Anda memiliki satu sertifikat yang terkait dengan distribusi yang diaktifkan selama satu bulan dan sertifikat lain yang terkait dengan distribusi yang diaktifkan selama setengah bulan, nilai ini akan menjadi 1,5.

Lihat laporan AWS penggunaan untuk CloudFront

AWS menyediakan laporan CloudFront penggunaan yang lebih rinci daripada laporan penagihan tetapi kurang rinci daripada log CloudFront akses. Laporan penggunaan menyediakan data penggunaan agregat berdasarkan jam, hari, atau bulan, dan mencantumkan operasi berdasarkan wilayah dan jenis penggunaan, seperti data yang ditransfer keluar dari wilayah Australia.

Untuk melihat laporan AWS penggunaan

1. Masuk ke Konsol Manajemen AWS dan buka AWS Manajemen Penagihan dan Biaya konsol di <https://console.aws.amazon.com/costmanagement/>.
2. Di panel navigasi, pilih Cost Explorer.
3. Pada halaman Laporan biaya dan penggunaan baru, di panel parameter Laporan, pilih rentang tanggal dan perincian laporan.

4. Di bawah Filter, Layanan, pilih CloudFront.
5. Pilih jenis Penggunaan.
6. Di bawah Rincian biaya dan penggunaan, pilih Unduh sebagai CSV.

Untuk informasi selengkapnya tentang laporan AWS penggunaan, lihat [Laporan AWS Penggunaan](#) di Panduan Ekspor Data AWS Pengguna.

Laporan CloudFront penggunaan mencakup nilai-nilai berikut:

- Layanan – AmazonCloudFront
- Operasi - Metode HTTP. Nilai mencakup DELETE, GET, HEAD, OPTIONS, PATCH, POST, dan PUT.
- UsageType – Salah satu nilai berikut:
 - Kode yang mengidentifikasi jenis transfer data
 - Invalidations
 - Executions-CloudFrontFunctions
 - KeyValueStore-APIOperations
 - KeyValueStore-EdgeReads
 - RealTimeLog-KinesisDataStream
 - SSL-Cert-Custom
- Sumber Daya — Baik ID CloudFront distribusi yang terkait dengan penggunaan atau ID sertifikat sertifikat SSL yang telah Anda kaitkan dengan CloudFront distribusi.
- StartTime/EndTime— Hari dimana penggunaan berlaku untuk, dalam Coordinated Universal Time (UTC).
- UsageValue— 1) Jumlah permintaan selama periode waktu yang ditentukan atau 2) jumlah data yang ditransfer dalam byte.

Jika Anda menggunakan Amazon S3 sebagai asal CloudFront, pertimbangkan untuk menjalankan laporan penggunaan untuk Amazon S3 juga. Namun, jika Anda menggunakan Amazon S3 untuk tujuan selain sebagai asal CloudFront distribusi Anda, mungkin tidak jelas bagian mana yang berlaku untuk penggunaan Anda. CloudFront

 Tip

Untuk informasi terperinci tentang setiap permintaan yang CloudFront diterima untuk objek Anda, aktifkan log CloudFront akses untuk distribusi Anda. Untuk informasi selengkapnya, lihat [the section called “Akses log \(log standar\)”](#).

Untuk informasi selengkapnya tentang memahami jenis CloudFront biaya dan penggunaan pada laporan Anda, lihat [the section called “Menafsirkan laporan AWS tagihan dan penggunaan Anda untuk CloudFront”](#).

Menafsirkan laporan AWS tagihan dan penggunaan Anda untuk CloudFront

Setelah Anda memiliki laporan [penagihan dan laporan penggunaan](#), Anda dapat menggunakan topik ini untuk memahami cara menafsirkan setiap CloudFront tagihan yang muncul pada tagihan Anda dan jenis penggunaan yang sesuai untuk setiap tagihan. Topik ini mencakup kode dan Wilayah AWS singkatan yang dapat muncul di kedua laporan.

Sebagian besar kode di kedua kolom mencakup singkatan dua huruf yang menunjukkan lokasi kegiatan. Dalam tabel berikut, *region* dalam kode diganti dalam AWS tagihan Anda dan dalam laporan penggunaan dengan salah satu singkatan dua huruf berikut:

- AP: Hong Kong, Filipina, Korea Selatan, Taiwan, dan Singapura (Asia Pasifik)
- AU: Australia
- CA: Kanada
- EU: Eropa dan Israel
- IN: India
- JP: Jepang
- ME: Timur Tengah
- SA: Amerika Selatan
- US: Amerika Serikat
- ZA: Afrika Selatan

Untuk informasi selengkapnya tentang penetapan harga menurut Wilayah AWS, lihat [CloudFront harga Amazon](#).

Catatan

- Tabel ini tidak termasuk biaya untuk mentransfer objek Anda dari bucket Amazon S3 CloudFront ke lokasi tepi. Biaya-biaya ini, jika ada, muncul di bagian Data Transfer AWS dari tagihan AWS Anda.
- Kolom pertama mencantumkan biaya yang muncul dalam laporan AWS tagihan Anda dan menjelaskan apa artinya masing-masing.
- Kolom kedua mencantumkan item yang muncul dalam laporan AWS penggunaan dan menunjukkan korelasi antara biaya tagihan dan item laporan penggunaan.

CloudFront biaya dalam AWS tagihan Anda	Nilai di UsageType kolom dalam laporan AWS penggunaan
<i>region</i>- DataTransfer -Keluar-Byte Total byte yang disajikan dari lokasi CloudFront tepi sebagai <i>region</i> tanggapan terhadap pengguna GET dan HEAD permintaan.	<i>region</i>-Out-bytes-http-static: Byte dilayani melalui HTTP untuk objek dengan TTL $\geq$ 3.600 detik. <i>region</i>-Out-bytes-https-static: Byte dilayani melalui HTTPS untuk objek dengan TTL $\geq$ 3.600 detik. <i>region</i>-Out-bytes-HTTP-Dinamis: Byte dilayani melalui HTTP untuk objek dengan TTL $<$ 3.600 detik. <i>region</i>-Out-bytes-https-dinamis: Byte dilayani melalui HTTPS untuk objek dengan TTL $<$ 3.600 detik. <i>region</i>-Out-bytes-http-proxy: Byte dikembalikan dari CloudFront ke pemirsa melalui HTTP sebagai tanggapan

CloudFront biaya dalam AWS tagihan Anda	Nilai di UsageType kolom dalam laporan AWS penggunaan
	terhadapDELETE,OPTIONS,PATCH,POST, dan PUT permintaan. <i>region</i>-Out-bytes-https-proxy: Byte dikembalikan dari CloudFront ke pemirsa melalui HTTPS sebagai tanggapan terhadapDELETE,OPTIONS,PATCH,POST, dan PUT permintaan. Ini termasuk byte yang dikembalikan dari ke CloudFront pemirsa melalui gRPC.
<i>region</i>- DataTransfer -Keluar- OBytes Total byte yang ditransfer dari lokasi CloudFront tepi ke fungsi asal atau tepi Anda sebagai respons terhadapDELETE,,OPTIONS, PATCHPOST, dan PUT permintaan. Biaya termasuk transfer data untuk WebSocket data dari klien ke server.	<i>region</i>-Keluar- OBytes -HTTP-Proksi Total byte yang ditransfer melalui HTTP dari lokasi CloudFront tepi ke fungsi asal atau tepi Anda sebagai respons terhadapDELETE,,OPTIONS,, PATCHPOST, dan PUT permintaan. <i>region</i>-Keluar- OBytes -HTTPS-Proxy Total byte yang ditransfer melalui HTTPS dari lokasi CloudFront tepi ke fungsi asal atau tepi Anda sebagai respons terhadapDELETE,,OPTIONS,, PATCHPOST, dan PUT permintaan. Ini termasuk byte yang ditransfer melalui gRPC CloudFront dari lokasi tepi ke asal CloudFront atau Fungsi Anda.

CloudFront biaya dalam AWS tagihan Anda	Nilai di UsageType kolom dalam laporan AWS penggunaan
<i>region</i>-Permintaan-Tier1 Jumlah HTTP GET dan HEAD permintaan.	<i>region</i>-Permintaan-HTTP-Statik Jumlah HTTP GET dan HEAD permintaan yang disajikan untuk objek dengan TTL ≥ 3.600 detik. <i>region</i>-Permintaan-HTTP-Dinamis Jumlah HTTP GET dan HEAD permintaan yang disajikan untuk objek dengan TTL < 3.600 detik.
<i>region</i>-Permintaan-Tier2-HTTPS Jumlah HTTPS GET dan HEAD permintaan.	<i>region</i>-Permintaan-HTTPS-Statik Jumlah HTTPS GET dan HEAD permintaan yang disajikan untuk objek dengan TTL ≥ 3.600 detik. <i>region</i>-Permintaan-HTTPS-Dinamis Jumlah HTTPS GET dan HEAD permintaan yang disajikan untuk objek dengan TTL < 3.600 detik.
<i>region</i>-Permintaan-HTTP-Proxy Jumlah HTTPDELETE,,OPTIONS, PATCHPOST, dan PUT permintaan yang CloudFront diteruskan ke fungsi asal atau tepi Anda. Juga termasuk jumlah WebSocket permintaan HTTP (GET permintaan dengan Upgrade : websocket header) yang CloudFront diteruskan ke fungsi asal atau tepi Anda.	<i>region</i>-Permintaan-HTTP-Proxy Sama seperti item yang sesuai di CloudFront tagihan Anda.

CloudFront biaya dalam AWS tagihan Anda	Nilai di UsageType kolom dalam laporan AWS penggunaan
<i>region</i>-Permintaan-HTTPS-Proxy Jumlah HTTPSDELETE,,OPTIONS, PATCHPOST, dan PUT permintaan yang CloudFront diteruskan ke fungsi asal atau tepi Anda. Juga termasuk jenis permintaan berikut: - Jumlah WebSocket permintaan HTTPS (GET permintaan dengan Upgrade : websocket header) yang CloudFront diteruskan ke fungsi asal atau tepi Anda. - Jumlah permintaan gRPC HTTPS.	<i>region</i>-Permintaan-HTTPS-Proxy Sama seperti item yang sesuai di CloudFront tagihan Anda.
<i>region</i>-Permintaan-HTTPS-Proxy-FLE Jumlah HTTPSDELETE,, OPTIONSPATCH, dan POST permintaan yang diproses dengan enkripsi tingkat lapangan yang CloudFront diteruskan ke fungsi asal atau tepi Anda.	<i>region</i>-Permintaan-HTTPS-Proxy-FLE Sama seperti item yang sesuai di CloudFront tagihan Anda.
<i>region</i>-Byte- OriginShield Total byte yang ditransfer dari asal ke cache tepi regional mana pun, termasuk cache tepi regional yang diaktifkan sebagai Origin Shield.	<i>region</i>-Byte- OriginShield Sama seperti item yang sesuai di CloudFront tagihan Anda.
<i>region</i>-OBytes-OriginShield Total byte yang ditransfer ke asal dari cache tepi regional mana pun, termasuk cache tepi regional yang diaktifkan sebagai Origin Shield.	<i>region</i>-OBytes-OriginShield Sama seperti item yang sesuai di CloudFront tagihan Anda.

CloudFront biaya dalam AWS tagihan Anda	Nilai di UsageType kolom dalam laporan AWS penggunaan
<i>region</i>-Permintaan- OriginShield Jumlah permintaan yang masuk ke Origin Shield sebagai lapisan tambahan. Untuk permintaan dinamis (non-cacheable) yang berhubungan dengan asal usul, Shield Asal selalu merupakan lapisan tambahan. Untuk permintaan yang dapat di-cache, Origin Shield terkadang merupakan lapisan tambahan. Untuk informasi selengkapnya, lihat the section called “Perkiraan biaya Origin Shield”.	<i>region</i>-Permintaan- OriginShield Sama seperti item yang sesuai di CloudFront tagihan Anda.
Pembatalan Biaya untuk membatalkan objek (menghapus objek dari lokasi CloudFront tepi). Untuk informasi selengkapnya, lihat Membayar untuk penghapusan validasi file.	Pembatalan Sama seperti item yang sesuai di CloudFront tagihan Anda.
SSL-Cert-Kustom Biaya untuk menggunakan sertifikat SSL dengan nama domain CloudFront alternatif seperti example.com alih-alih menggunakan sertifikat CloudFront SSL default dan nama domain yang CloudFront ditetapkan untuk distribusi Anda.	SSL-Cert-Kustom Sama seperti item yang sesuai di CloudFront tagihan Anda.
RealTimeLog-KinesisDataStream Biaya untuk jumlah baris yang dihasilkan untuk log akses real-time.	RealTimeLog-KinesisDataStream Sama seperti item yang sesuai di CloudFront tagihan Anda.

CloudFront biaya dalam AWS tagihan Anda	Nilai di UsageType kolom dalam laporan AWS penggunaan
Eksekusi- CloudFrontFunctions Biaya untuk jumlah pemanggilan CloudFront Fungsi. region-Lambda-Edge-Permintaan Biaya untuk jumlah pemanggilan fungsi Lambda @Edge.	Eksekusi- CloudFrontFunctions Sama seperti item yang sesuai di CloudFront tagihan Anda. region-Lambda-Edge-Permintaan Sama seperti item yang sesuai di CloudFront tagihan Anda.
region-Lambda-Edge-GB-Detik Biaya untuk durasi dari saat fungsi Lambda @Edge Anda dipanggil saat kembali atau dihentikan.	region-Lambda-Edge-GB-Detik Sama seperti item yang sesuai di CloudFront tagihan Anda.
KeyValueStore-EdgeReads Biaya untuk jumlah panggilan baca ke CloudFront KeyValueStore metode,, <code>get()</code>, <code>exists()</code>, dan <code>meta()</code>. Untuk informasi selengkapnya, lihat Metode pembantu untuk penyimpanan nilai kunci.	KeyValueStore-EdgeReads Sama seperti item yang sesuai di CloudFront tagihan Anda.
KeyValueStore-APIOperations Biaya untuk jumlah panggilan ke CloudFront KeyValueStoreAPI.	KeyValueStore-APIOperations Sama seperti item yang sesuai di CloudFront tagihan Anda.

Lihat CloudFront laporan di konsol

Setiap laporan memberikan informasi dan visualisasi terperinci, sehingga Anda dapat mengoptimalkan pengiriman konten, mengidentifikasi kemacetan kinerja, dan membuat keputusan berdasarkan data. Baik Anda perlu memantau efisiensi cache, menganalisis pola lalu lintas, atau memahami pemirsa dengan lebih baik, Anda dapat menggunakan laporan ini untuk memantau dan menganalisis CloudFront distribusi Anda secara efektif.

Anda dapat melihat laporan CloudFront aktivitas berikut di konsol:

Topik

- [Lihat laporan statistik CloudFront cache](#)
- [Lihat laporan objek CloudFront populer](#)
- [Lihat laporan perujuk CloudFront teratas](#)
- [Lihat laporan CloudFront penggunaan](#)
- [Lihat laporan CloudFront pemirsa](#)

Sebagian besar laporan ini didasarkan pada data dalam log CloudFront akses, yang berisi informasi rinci tentang setiap permintaan pengguna yang CloudFront diterima. Anda tidak perlu mengaktifkan log akses untuk melihat laporan. Untuk informasi selengkapnya, lihat [Akses log \(log standar\)](#).

Lihat laporan statistik CloudFront cache

Laporan statistik CloudFront cache Amazon menunjukkan informasi berikut:

- Total permintaan — Jumlah total permintaan untuk semua kode status HTTP (misalnya, 200 atau 404) dan semua metode (misalnya, GET, HEAD, atau POST)
- Persentase permintaan penampil menurut jenis hasil — Hits, misses, dan error sebagai persentase dari total permintaan penampil untuk distribusi yang dipilih CloudFront
- Byte ditransfer ke pemirsa — Total byte dan byte dari kesalahan
- Kode status HTTP - Permintaan penampil dengan kode status HTTP
- Persentase permintaan GET yang tidak selesai diunduh - Permintaan GET Viewer yang tidak selesai mengunduh objek yang diminta sebagai persentase dari total permintaan

Data untuk statistik ini diambil dari sumber yang sama dengan log CloudFront akses. Namun, Anda tidak perlu mengaktifkan [log akses](#) untuk melihat statistik cache.

Anda dapat menampilkan bagan untuk rentang tanggal tertentu dalam 60 hari terakhir, dengan poin data setiap jam atau setiap hari. Anda biasanya dapat melihat data tentang permintaan yang CloudFront diterima beberapa jam yang lalu, tetapi data terkadang dapat tertunda sebanyak 24 jam.

Topik

- [Lihat laporan statistik CloudFront cache di konsol](#)

- [Unduh data dalam format CSV](#)
- [Bagaimana grafik statistik cache terkait dengan data dalam log CloudFront standar \(log akses\)](#)

Lihat laporan statistik CloudFront cache di konsol

Anda dapat melihat laporan statistik CloudFront cache di konsol.

Untuk melihat laporan statistik CloudFront cache

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Statistik Cache.
3. Di CloudFront Laporan Statistik Cache panel, untuk Tanggal Mulai dan Tanggal Akhir, pilih rentang tanggal yang ingin Anda tampilkan bagan statistik cache. Rentang yang tersedia tergantung pada nilai yang Anda pilih Keceragaman:
 - Setiap Hari – Untuk menampilkan grafik dengan satu titik data per hari, pilih rentang tanggal mana pun dalam 60 hari sebelumnya.
 - Per Jam – Untuk menampilkan bagan dengan satu titik data setiap jam, pilih rentang tanggal hingga 14 hari dalam 60 hari sebelumnya.

Tanggal dan waktu ada dalam Waktu Universal Terkoordinasi (UTC).

4. Untuk Keceragaman, menentukan apakah akan menampilkan satu titik data per hari atau satu titik data per jam dalam bagan. Jika Anda menetapkan rentang tanggal lebih dari 14 hari, opsi untuk menentukan satu titik data per jam tidak tersedia.
5. Untuk Lokasi Penampil, pilih benua yang membuat permintaan penampil berasal, atau pilih Semua Lokasi. Grafik statistik cache mencakup data untuk permintaan yang CloudFront diterima dari lokasi yang ditentukan.
6. Di Distribusi pilih distribusi yang ingin Anda tampilkan datanya dalam bagan penggunaan:
 - Distribusi individual — Bagan menampilkan data untuk CloudFront distribusi yang dipilih. Daftar Distribusi menampilkan ID distribusi dan nama domain alternatif (CNAMEs) untuk distribusi, jika ada. Jika distribusi tidak memiliki nama domain alternatif, daftar tersebut menyertakan nama domain asal untuk distribusi.
 - Semua distribusi — Bagan menampilkan data yang dijumlahkan untuk semua distribusi yang terkait dengan distribusi saat ini Akun AWS, tidak termasuk distribusi yang telah Anda hapus.

7. Pilih Perbarui.

Tip

- Untuk melihat data untuk titik data harian atau per jam dalam bagan, arahkan kursor ke titik data.
- Untuk grafik yang menunjukkan data yang ditransfer, Anda dapat mengubah skala vertikal menjadi gigabyte, megabyte, atau kilobyte.

Unduh data dalam format CSV

Anda dapat mengunduh laporan statistik cache dalam format CSV. Bagian ini menjelaskan cara mengunduh laporan dan menjelaskan nilai-nilai dalam laporan.

Untuk mengunduh laporan statistik cache dalam format CSV

1. Saat melihat laporan statistik cache, pilih CSV.
2. Di Pembukaan nama file kotak dialog, pilih apakah ingin membuka atau menyimpan file.

Informasi tentang laporan

Beberapa baris pertama pada laporan mencakup informasi berikut:

Versi

Versi format untuk file CSV ini.

Laporan

Nama laporan.

DistributionID

ID distribusi yang Anda jalankan laporannya, atau ALL jika Anda menjalankan laporan untuk semua distribusi.

StartDateUTC

Permulaan rentang tanggal saat Anda menjalankan laporan dalam Waktu Universal Terkoordinasi (UTC).

EndDateUTC

Akhir rentang tanggal saat Anda menjalankan laporan dalam Waktu Universal Terkoordinasi (UTC).

GeneratedTimeUTC

Tanggal dan waktu Anda menjalankan laporan, dalam Waktu Universal Terkoordinasi (UTC).

Keserbagunaan

Apakah setiap baris dalam laporan mewakili satu jam atau satu hari.

ViewerLocation

Kontainer yang diminta oleh pemirsa berasal dari, atau ALL, jika Anda memilih untuk mengunduh laporan untuk semua lokasi.

Data dalam laporan statistik cache

Laporan mencakup nilai-nilai berikut:

DistributionID

ID distribusi yang Anda jalankan laporannya, atau ALL jika Anda menjalankan laporan untuk semua distribusi.

FriendlyName

Nama domain alternatif (CNAME) untuk distribusi, jika ada. Jika distribusi tidak memiliki nama domain alternatif, daftar tersebut mencakup nama domain asal untuk distribusi tersebut.

ViewerLocation

Kontainer yang diminta oleh pemirsa berasal dari, atau ALL, jika Anda memilih untuk mengunduh laporan untuk semua lokasi.

TimeBucket

Jam atau hari saat data berlaku, dalam Waktu Universal Terkoordinasi (UTC).

RequestCount

Total jumlah permintaan untuk semua kode status HTTP (misalnya, 200 atau 404) dan semua metode (misalnya, GET, HEAD, atau POST).

HitCount

Jumlah permintaan penampil yang objeknya disajikan dari cache CloudFront tepi.

MissCount

Jumlah permintaan penampil yang objeknya saat ini tidak berada dalam cache tepi, jadi CloudFront harus mendapatkan objek dari asal Anda.

ErrorCount

Jumlah permintaan pemirsa yang mengakibatkan kesalahan, CloudFront tidak memiliki objek.

IncompleteDownloadCount

Jumlah permintaan penampil yang menjadi titik awal pengunduhan penampil, tetapi tidak selesai.

HTTP2xx

Jumlah permintaan penampil di mana kode status HTTP adalah nilai 2xx (dihitung).

HTTP3xx

Jumlah permintaan penampil di mana kode status HTTP adalah nilai 3xx (dissional tindakan diperlukan).

HTTP4xx

Jumlah permintaan penampil di mana kode status HTTP adalah nilai 4xx (kesalahan klien).

HTTP5xx

Jumlah permintaan penampil di mana kode status HTTP adalah nilai 5xx (kesalahan server).

TotalBytes

Jumlah total byte yang disajikan kepada CloudFront pemirsa oleh sebagai tanggapan atas semua permintaan untuk semua metode HTTP.

BytesFromMisses

Jumlah byte yang disajikan ke penampil untuk objek yang tidak berada di cache tepi pada saat permintaan. Nilai ini adalah perkiraan byte yang baik yang ditransfer dari cache asal Anda ke CloudFront edge. Namun, ini mengecualikan permintaan objek yang sudah ada di cache tepi, tetapi sudah kedaluwarsa.

Bagaimana grafik statistik cache terkait dengan data dalam log CloudFront standar (log akses)

Tabel berikut menunjukkan bagaimana grafik statistik cache di CloudFront konsol sesuai dengan nilai dalam log CloudFront akses. Untuk informasi selengkapnya tentang log CloudFront akses, lihat [Akses log \(log standar\)](#).

Total permintaan

Bagan ini menunjukkan jumlah total permintaan untuk semua kode status HTTP (misalnya, 200 atau 404) dan semua metode (misalnya, GET, HEAD, atau POST). Total permintaan yang ditunjukkan dalam bagan ini sama dengan jumlah total permintaan dalam file log akses untuk periode waktu yang sama.

Persentase permintaan pemirsa menurut jenis hasil

Bagan ini menunjukkan klik, kesalahan, dan kesalahan sebagai persentase dari total permintaan penampil untuk CloudFront distribusi yang dipilih:

- Hit - Permintaan penampil yang objeknya disajikan dari cache CloudFront tepi. Dalam log akses, ini adalah permintaan yang `x-edge-response-result-type` adalah Hit.
- Miss — Permintaan penampil yang objeknya saat ini tidak berada dalam cache tepi, jadi CloudFront harus mendapatkan objek dari asal Anda. Dalam log akses, ini adalah permintaan yang `x-edge-response-result-type` adalah Miss.
- Kesalahan - Permintaan penampil yang mengakibatkan kesalahan, jadi CloudFront tidak melayani objek. Dalam log akses, ini adalah permintaan yang `x-edge-response-result-type` adalah Error, LimitExceeded, atau CapacityExceeded.

Bagan tidak menyertakan temuan pembaruan—permintaan objek yang ada di cache tepi tetapi telah kedaluwarsa. Dalam log akses, refresh hits adalah permintaan dengan nilai `x-edge-response-result-type` adalah RefreshHit.

Byte ditransfer ke pemirsa

Bagan ini menunjukkan dua nilai:

- Total byte — Jumlah total byte yang disajikan kepada CloudFront pemirsa oleh sebagai tanggapan atas semua permintaan untuk semua metode HTTP. Dalam log CloudFront akses, Total Bytes adalah jumlah nilai dalam `sc-bytes` kolom untuk semua permintaan selama periode waktu yang sama.

- Byte dari kesalahan — Jumlah byte yang disajikan kepada pemirsa untuk objek yang tidak ada di cache tepi pada saat permintaan. Dalam log CloudFront akses, byte dari kesalahan adalah jumlah nilai dalam `sc-bytes` kolom untuk permintaan yang nilainya `x-edge-result-type Miss`. Nilai ini adalah perkiraan byte yang baik yang ditransfer dari cache asal Anda ke CloudFront edge. Namun, ini mengecualikan permintaan objek yang sudah ada di cache tepi, tetapi sudah kedaluwarsa.

Kode status HTTP

Bagan ini menampilkan permintaan penampil dengan kode status HTTP. Dalam log CloudFront akses, kode status muncul di `sc-status` kolom:

- 2xx – Permintaan berhasil.
- 3xx – Diperlukan tindakan tambahan. Misalnya, 301 (Moved Permanently) berarti bahwa objek yang diminta telah berpindah ke lokasi yang berbeda.
- 4xx – Klien tersebut tampaknya melakukan kesalahan. Misalnya, 404 (Tidak Ditemukan) berarti klien meminta objek yang tidak dapat ditemukan.
- 5xx – Server asal tidak memenuhi permintaan. Misalnya, 503 (Layanan Tidak Tersedia) berarti bahwa server asal saat ini tidak tersedia.

Persentase permintaan GET yang belum selesai diunduh

Bagan ini menampilkan penampil GET permintaan yang belum selesai mengunduh objek yang diminta sebagai persentase dari total permintaan. Biasanya, mengunduh sebuah objek tidak selesai karena penampil membatalkan unduhan, misalnya, dengan mengeklik tautan lain atau dengan menutup peramban. Dalam log CloudFront akses, permintaan ini memiliki nilai `200` di `sc-status` kolom dan nilai `Error` di `x-edge-result-type` kolom.

Lihat laporan objek CloudFront populer

Lihat laporan objek CloudFront populer Amazon untuk melihat 50 objek paling populer untuk distribusi selama rentang tanggal tertentu dalam 60 hari sebelumnya. Anda juga dapat melihat statistik tentang objek tersebut, termasuk yang berikut ini:

- Jumlah permintaan untuk objek
- Jumlah hit dan miss
- Rasio hit
- Jumlah byte yang dilayani untuk kesalahan

- Total byte yang dilayani
- Jumlah unduhan yang tidak lengkap
- Jumlah permintaan berdasarkan kode status HTTP (2xx, 3xx, 4xx, dan 5xx)

Data untuk statistik ini diambil dari sumber yang sama dengan log CloudFront akses. Namun, Anda tidak perlu mengaktifkan [akses logging](#) untuk melihat objek populer.

Topik

- [Lihat laporan objek CloudFront populer di konsol](#)
- [Bagaimana CloudFront menghitung statistik objek populer](#)
- [Unduh data dalam format CSV](#)
- [Bagaimana data dalam laporan objek populer terkait dengan data dalam log CloudFront standar \(log akses\)](#)

Lihat laporan objek CloudFront populer di konsol

Anda dapat melihat laporan objek CloudFront populer di konsol.

Untuk melihat objek populer untuk CloudFront distribusi

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Objek Populer.
3. Di CloudFront Laporan Objek Populer panel, untuk Tanggal Mulai dan Tanggal Akhir, pilih rentang tanggal yang ingin Anda tampilkan daftar objek populer. Anda dapat memilih rentang tanggal apa pun dalam 60 hari sebelumnya.

Tanggal dan waktu ada dalam Waktu Universal Terkoordinasi (UTC).

4. Di Distribusi pilih distribusi yang ingin Anda tampilkan untuk daftar objek populer.
5. Pilih Perbarui.

Bagaimana CloudFront menghitung statistik objek populer

Untuk mendapatkan hitungan akurat dari 50 objek teratas dalam distribusi Anda, CloudFront hitung permintaan untuk semua objek Anda dalam interval 10 menit dimulai pada tengah malam dan

pertahankan total 150 objek teratas selama 24 jam ke depan. (CloudFront juga mempertahankan total harian untuk 150 objek teratas selama 60 hari.)

Di dekat bagian bawah daftar, objek terus-menerus naik ke atau turun dari daftar, sehingga total untuk objek tersebut adalah perkiraan. 50 objek di bagian atas daftar 150 objek mungkin naik dan jatuh dalam daftar, tetapi mereka jarang turun dari daftar sama sekali, sehingga total untuk objek tersebut lebih dapat diandalkan.

Ketika sebuah objek turun dari daftar 150 objek teratas dan kemudian naik ke daftar lagi selama satu hari, CloudFront menambahkan perkiraan jumlah permintaan untuk periode bahwa objek itu hilang dari daftar. Perkiraan didasarkan pada jumlah permintaan yang diterima oleh objek mana pun yang berada di bagian bawah daftar selama periode waktu tersebut.

Jika objek naik ke 50 objek teratas di kemudian hari, perkiraan jumlah permintaan yang CloudFront diterima saat objek berada di luar 150 objek teratas biasanya menyebabkan jumlah permintaan dalam laporan objek populer melebihi jumlah permintaan yang muncul di log akses untuk objek itu.

Unduh data dalam format CSV

Anda dapat mengunduh laporan objek populer dalam format CSV. Bagian ini menjelaskan cara mengunduh laporan dan menjelaskan nilai-nilai dalam laporan.

Untuk mengunduh laporan objek populer dalam format CSV

1. Saat melihat laporan objek populer, pilih CSV.
2. Di Pembukaan nama file kotak dialog, pilih apakah ingin membuka atau menyimpan file.

Informasi tentang laporan

Beberapa baris pertama pada laporan mencakup informasi berikut:

Versi

Versi format untuk file CSV ini.

Laporan

Nama laporan.

DistributionID

ID distribusi yang Anda jalankan laporannya.

StartDateUTC

Permulaan rentang tanggal saat Anda menjalankan laporan dalam Waktu Universal Terkoordinasi (UTC).

EndDateUTC

Akhir rentang tanggal saat Anda menjalankan laporan dalam Waktu Universal Terkoordinasi (UTC).

GeneratedTimeUTC

Tanggal dan waktu Anda menjalankan laporan, dalam Waktu Universal Terkoordinasi (UTC).

Data dalam laporan objek populer

Laporan mencakup nilai-nilai berikut:

DistributionID

ID distribusi yang Anda jalankan laporannya.

FriendlyName

Nama domain alternatif (CNAME) untuk distribusi, jika ada. Jika distribusi tidak memiliki nama domain alternatif, daftar tersebut mencakup nama domain asal untuk distribusi tersebut.

Objek

500 karakter terakhir URL untuk objek.

RequestCount

Total jumlah permintaan untuk objek ini.

HitCount

Jumlah permintaan penampil yang objeknya disajikan dari cache CloudFront tepi.

MissCount

Jumlah permintaan penampil yang objeknya saat ini tidak berada dalam cache tepi, jadi CloudFront harus mendapatkan objek dari asal Anda.

HitCountPct

Nilai dari `HitCount` sebagai persentase nilai `RequestCount`.

BytesFromMisses

Jumlah byte yang disajikan ke penampil untuk objek ini saat objek tidak berada di cache tepi pada saat permintaan.

TotalBytes

Jumlah total byte yang disajikan kepada pemirsa oleh CloudFront untuk objek ini sebagai tanggapan atas semua permintaan untuk semua metode HTTP.

IncompleteDownloadCount

Jumlah permintaan penampil untuk objek ini saat penampil mulai tetapi tidak selesai mengunduh objek.

HTTP2xx

Jumlah permintaan penampil di mana kode status HTTP adalah nilai 2xx (dihitung).

HTTP3xx

Jumlah permintaan penampil di mana kode status HTTP adalah nilai 3xx (dissional tindakan diperlukan).

HTTP4xx

Jumlah permintaan penampil di mana kode status HTTP adalah nilai 4xx (kesalahan klien).

HTTP5xx

Jumlah permintaan penampil di mana kode status HTTP adalah nilai 5xx (kesalahan server).

Bagaimana data dalam laporan objek populer terkait dengan data dalam log CloudFront standar (log akses)

Daftar berikut menunjukkan bagaimana nilai dalam laporan objek populer di CloudFront konsol sesuai dengan nilai dalam log CloudFront akses. Untuk informasi lebih lanjut tentang CloudFront log akses, lihat [Akses log \(log standar\)](#).

URL

500 karakter terakhir URL yang digunakan penampil untuk mengakses objek.

Permintaan

Total jumlah permintaan objek. Nilai ini umumnya sesuai dengan jumlah GET permintaan untuk objek dalam log CloudFront akses.

Menabrak

Jumlah permintaan pemirsa untuk objek yang dilayani dari CloudFront cache tepi. Dalam log akses, ini adalah permintaan yang `x-edge-response-result-type` adalah `Hit`.

Kegagalan

Jumlah permintaan penampil yang objeknya tidak berada dalam cache tepi, jadi CloudFront diambil objek dari asal Anda. Dalam log akses, ini adalah permintaan yang `x-edge-response-result-type` adalah `Miss`.

Rasio hit

Nilai dari Menabrak sebagai persentase nilai Permintaan kolom.

Byte dari kesalahan

Jumlah byte yang disajikan ke penampil untuk objek yang tidak berada di cache tepi pada saat permintaan. Dalam log CloudFront akses, byte dari kesalahan adalah jumlah nilai dalam `sc-bytes` kolom untuk permintaan yang nilainya `x-edge-result-type Miss`.

Total byte

Jumlah total byte yang CloudFront disajikan kepada pemirsa dalam menanggapi semua permintaan objek untuk semua metode HTTP. Dalam log CloudFront akses, total byte adalah jumlah nilai dalam `sc-bytes` kolom untuk semua permintaan selama periode waktu yang sama.

Unduhan tidak lengkap

Jumlah permintaan penampil yang tidak selesai mengunduh objek yang diminta. Biasanya, alasan pengunduhan tidak selesai adalah penampil membatalkannya, misalnya, dengan mengeklik tautan lain atau dengan menutup peramban. Dalam log CloudFront akses, permintaan ini memiliki nilai `200` di `sc-status` kolom dan nilai `Error` di `x-edge-result-type` kolom.

2xx

Jumlah permintaan yang kode status HTTP adalah `2xx`, `Successful`. Dalam log CloudFront akses, kode status muncul di `sc-status` kolom.

3xx

Jumlah permintaan kode status HTTP 3xx, `Redirection`. 3xx kode status menunjukkan bahwa tindakan tambahan diperlukan. Misalnya, 301 (`Moved Permanently`) berarti bahwa objek yang diminta telah berpindah ke lokasi yang berbeda.

4xx

Jumlah permintaan kode status HTTP 4xx, `Client Error`. 4xx kode status menunjukkan bahwa klien tersebut tampaknya melakukan kesalahan. Misalnya, 404 (`Tidak Ditemukan`) berarti klien meminta objek yang tidak dapat ditemukan.

5xx

Jumlah permintaan kode status HTTP 5xx, `Server Error`. 5xx kode status menunjukkan bahwa server asal tidak memenuhi permintaan. Misalnya, 503 (`Layanan Tidak Tersedia`) berarti bahwa server asal saat ini tidak tersedia.

Lihat laporan perujuk CloudFront teratas

Laporan perujuk CloudFront teratas mencakup yang berikut untuk rentang tanggal apa pun dalam 60 hari sebelumnya:

- 25 perujuk teratas (domain situs web yang menghasilkan permintaan HTTP dan HTTPS terbanyak untuk objek yang CloudFront mendistribusikan untuk distribusi Anda)
- Jumlah permintaan dari perujuk
- Jumlah permintaan dari perujuk sebagai persentase dari jumlah total permintaan selama periode yang ditentukan

Data untuk laporan perujuk teratas diambil dari sumber yang sama dengan log CloudFront akses. Namun, Anda tidak perlu mengaktifkan [pencatatan akses](#) untuk melihat perujuk teratas.

Perujuk teratas dapat berupa mesin pencari, situs web lain yang menautkan langsung ke objek Anda, atau situs web Anda sendiri. Misalnya, jika `https://example.com/index.html` link ke 10 grafis, `example.com` adalah perujuk untuk semua 10 grafis.

Note

Jika pengguna memasukkan URL langsung ke baris alamat peramban, tidak ada perujuk untuk objek yang diminta.

Topik

- [Lihat laporan perujuk CloudFront teratas di konsol](#)
- [Bagaimana CloudFront menghitung statistik perujuk teratas](#)
- [Unduh data dalam format CSV](#)
- [Bagaimana data dalam laporan perujuk teratas terkait dengan data dalam log CloudFront standar \(log akses\)](#)

Lihat laporan perujuk CloudFront teratas di konsol

Anda dapat melihat laporan perujuk CloudFront teratas di konsol.

Untuk melihat perujuk teratas untuk distribusi CloudFront

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Top Referrers.
3. Di CloudFront Laporan Pemberi Referensi Teratas panel, untuk Tanggal Mulai dan Tanggal Akhir, pilih rentang tanggal yang ingin Anda tampilkan daftar perujuk teratas.

Tanggal dan waktu ada dalam Waktu Universal Terkoordinasi (UTC).

4. Di Distribusi pilih distribusi yang ingin Anda tampilkan daftar perujuk teratas.
5. Pilih Perbarui.

Bagaimana CloudFront menghitung statistik perujuk teratas

Untuk mendapatkan hitungan akurat dari 25 perujuk teratas, CloudFront hitung permintaan untuk semua objek Anda dalam interval 10 menit dan pertahankan total 75 perujuk teratas. Di dekat bagian bawah daftar, perujuk terus-menerus naik atau turun dari daftar, sehingga total untuk perujuk tersebut adalah perkiraan.

25 perujuk di bagian atas daftar 75 perujuk mungkin naik dan masuk dalam daftar, tetapi mereka jarang turun dari daftar sama sekali, sehingga total untuk perujuk tersebut biasanya lebih dapat diandalkan.

Unduh data dalam format CSV

Anda dapat mengunduh laporan perujuk teratas dalam format CSV. Bagian ini menjelaskan cara mengunduh laporan dan menjelaskan nilai-nilai dalam laporan.

Untuk mengunduh laporan perujuk teratas dalam format CSV

1. Saat melihat laporan Perujuk Teratas, pilih CSV.
2. Di Pembukaan nama file kotak dialog, pilih apakah ingin membuka atau menyimpan file.

Informasi tentang laporan

Beberapa baris pertama pada laporan mencakup informasi berikut:

Versi

Versi format untuk file CSV ini.

Laporan

Nama laporan.

DistributionID

ID distribusi yang Anda jalankan laporannya, atau ALL jika Anda menjalankan laporan untuk semua distribusi.

StartDateUTC

Permulaan rentang tanggal saat Anda menjalankan laporan dalam Waktu Universal Terkoordinasi (UTC).

EndDateUTC

Akhir rentang tanggal saat Anda menjalankan laporan dalam Waktu Universal Terkoordinasi (UTC).

GeneratedTimeUTC

Tanggal dan waktu Anda menjalankan laporan, dalam Waktu Universal Terkoordinasi (UTC).

Data dalam laporan perujuk teratas

Laporan mencakup nilai-nilai berikut:

DistributionID

ID distribusi yang Anda jalankan laporannya, atau ALL jika Anda menjalankan laporan untuk semua distribusi.

FriendlyName

Nama domain alternatif (CNAME) untuk distribusi, jika ada. Jika distribusi tidak memiliki nama domain alternatif, daftar tersebut mencakup nama domain asal untuk distribusi tersebut.

Pemberi referensi

Nama domain perujuk.

RequestCount

Total jumlah permintaan dari nama domain di `Referer` kolom.

RequestsPct

Jumlah permintaan yang diajukan oleh perujuk sebagai persentase dari jumlah total permintaan selama periode tertentu.

Bagaimana data dalam laporan perujuk teratas terkait dengan data dalam log CloudFront standar (log akses)

Daftar berikut menunjukkan bagaimana nilai dalam laporan Perujuk Teratas di CloudFront konsol sesuai dengan nilai dalam log CloudFront akses. Untuk informasi lebih lanjut tentang CloudFront log akses, lihat [Akses log \(log standar\)](#).

Pemberi referensi

Nama domain perujuk. Dalam log akses, pemberi referensi tercantum di `cs(Referer)` kolom.

Jumlah permintaan

Total jumlah permintaan dari nama domain di Pemberi referensi kolom. Nilai ini umumnya sesuai dengan jumlah GET permintaan dari perujuk di log CloudFront akses.

% Permintaan

Jumlah permintaan yang diajukan oleh perujuk sebagai persentase dari jumlah total permintaan selama periode tertentu. Jika Anda memiliki lebih dari 25 perujuk, maka Anda tidak dapat menghitung Permintaan% berdasarkan data dalam tabel ini karena kolom jumlah permintaan tidak menyertakan semua permintaan selama periode yang ditentukan.

Lihat laporan CloudFront penggunaan

Laporan CloudFront penggunaan mencakup informasi berikut:

- Jumlah permintaan - Menampilkan jumlah total permintaan yang CloudFront merespons dari lokasi tepi di wilayah yang dipilih selama setiap interval waktu untuk CloudFront distribusi yang ditentukan.
- Data yang ditransfer oleh protokol dan data yang ditransfer oleh tujuan - Keduanya menunjukkan jumlah total data yang ditransfer dari lokasi CloudFront tepi di wilayah yang dipilih selama setiap interval waktu untuk CloudFront distribusi yang ditentukan. Mereka memisahkan data secara berbeda, sebagai berikut:
 - Dengan protokol — Memisahkan data dengan protokol: HTTP atau HTTPS.
 - Berdasarkan tujuan — Memisahkan data berdasarkan tujuan: ke pemirsa Anda atau ke asal Anda.

Laporan CloudFront penggunaan didasarkan pada laporan AWS penggunaan untuk CloudFront. Laporan ini tidak memerlukan konfigurasi tambahan apa pun. Untuk informasi selengkapnya, lihat [Lihat laporan AWS penggunaan untuk CloudFront](#).

Anda dapat melihat laporan untuk rentang tanggal tertentu dalam 60 hari terakhir, dengan titik data setiap jam atau setiap hari. Anda biasanya dapat melihat data tentang permintaan yang CloudFront diterima baru-baru ini empat jam yang lalu, tetapi data kadang-kadang dapat ditunda sebanyak 24 jam.

Untuk informasi selengkapnya, lihat [Bagaimana grafik penggunaan terkait dengan data dalam laporan CloudFront penggunaan](#).

Topik

- [Melihat laporan CloudFront penggunaan di konsol](#)
- [Unduh data dalam format CSV](#)

- [Bagaimana grafik penggunaan terkait dengan data dalam laporan CloudFront penggunaan](#)

Melihat laporan CloudFront penggunaan di konsol

Anda dapat melihat laporan CloudFront penggunaan di konsol.

Untuk melihat laporan CloudFront penggunaan

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Laporan Penggunaan.
3. Di CloudFront Laporan Penggunaan panel, untuk Tanggal Mulai dan Tanggal Akhir, pilih rentang tanggal yang ingin Anda tampilkan bagan penggunaan. Rentang yang tersedia tergantung pada nilai yang Anda pilih Keserbagunaan:
 - Setiap Hari — Untuk menampilkan grafik dengan satu titik data per hari, pilih rentang tanggal mana pun dalam 60 hari sebelumnya.
 - Per Jam — Untuk menampilkan grafik dengan satu titik data setiap jam, pilih rentang tanggal hingga 14 hari dalam 60 hari sebelumnya.

Tanggal dan waktu ada dalam Waktu Universal Terkoordinasi (UTC).

4. Untuk Keserbagunaan, menentukan apakah akan menampilkan satu titik data per hari atau satu titik data per jam dalam bagan. Jika Anda menetapkan rentang tanggal lebih dari 14 hari, opsi untuk menentukan satu titik data per jam tidak tersedia.
5. Untuk Wilayah Penagihan, pilih wilayah CloudFront penagihan yang memiliki data yang ingin Anda lihat, atau pilih Semua Wilayah. Bagan penggunaan mencakup data untuk permintaan yang CloudFront proses di lokasi tepi di wilayah tertentu. Wilayah tempat permintaan CloudFront proses mungkin atau mungkin tidak sesuai dengan lokasi pemirsa Anda.

Pilih hanya Wilayah yang termasuk dalam kelas harga untuk distribusi Anda. Jika tidak, grafik penggunaan mungkin tidak akan berisi data apa pun. Misalnya, jika Anda memilih Kelas Harga 200 untuk distribusi Anda, wilayah penagihan Amerika Selatan dan Australia tidak termasuk, sehingga CloudFront umumnya tidak akan memproses permintaan Anda dari wilayah tersebut. Untuk informasi selengkapnya tentang kelas harga, lihat [CloudFront harga](#).

6. Di Distribusi pilih distribusi yang ingin Anda tampilkan datanya dalam bagan penggunaan:

- Distribusi individual — Bagan menampilkan data untuk CloudFront distribusi yang dipilih. Daftar Distribusi menampilkan ID distribusi dan nama domain alternatif (CNAMEs) untuk distribusi, jika ada. Jika distribusi tidak memiliki nama domain alternatif, daftar tersebut mencakup nama domain asal untuk distribusi tersebut.
- Semua distribusi (tidak termasuk dihapus) - Bagan menampilkan data yang dijumlahkan untuk semua distribusi yang terkait dengan AWS akun saat ini, tidak termasuk distribusi yang telah Anda hapus.
- Semua Distribusi yang Dihapus — Bagan menampilkan data yang dijumlahkan untuk semua distribusi yang terkait dengan AWS akun saat ini dan yang dihapus dalam 60 hari terakhir.

7. Pilih Perbarui Grafik.

Tip

- Untuk melihat data untuk titik data harian atau per jam dalam bagan, arahkan kursor ke titik data.
- Untuk grafik yang menunjukkan data yang ditransfer, perhatikan bahwa Anda dapat mengubah skala vertikal menjadi gigabyte, megabyte, atau kilobyte.

Unduh data dalam format CSV

Anda dapat mengunduh laporan penggunaan dalam format CSV. Bagian ini menjelaskan cara mengunduh laporan dan menjelaskan nilai-nilai dalam laporan.

Untuk mengunduh laporan penggunaan dalam format CSV

1. Saat melihat laporan Penggunaan, pilih CSV.
2. Di Pembukaan nama file kotak dialog, pilih apakah ingin membuka atau menyimpan file.

Informasi tentang laporan

Beberapa baris pertama pada laporan mencakup informasi berikut:

Versi

Versi format untuk file CSV ini.

Laporan

Nama laporan.

DistributionID

ID distribusi yang Anda jalankan laporannya, ALL jika Anda menjalankan laporan untuk semua distribusi, atau ALL_DELETED jika Anda menjalankan laporan untuk semua distribusi yang dihapus.

StartDateUTC

Permulaan rentang tanggal saat Anda menjalankan laporan dalam Waktu Universal Terkoordinasi (UTC).

EndDateUTC

Akhir rentang tanggal saat Anda menjalankan laporan dalam Waktu Universal Terkoordinasi (UTC).

GeneratedTimeUTC

Tanggal dan waktu Anda menjalankan laporan, dalam Waktu Universal Terkoordinasi (UTC).

Keserbagunaan

Apakah setiap baris dalam laporan mewakili satu jam atau satu hari.

BillingRegion

Kontiner yang diminta oleh pemirsa berasal dari, atau ALL, jika Anda memilih untuk mengunduh laporan untuk semua wilayah penagihan.

Data dalam laporan penggunaan

Laporan mencakup nilai-nilai berikut:

DistributionID

ID distribusi yang Anda jalankan laporannya, ALL jika Anda menjalankan laporan untuk semua distribusi, atau ALL_DELETED jika Anda menjalankan laporan untuk semua distribusi yang dihapus.

FriendlyName

Nama domain alternatif (CNAME) untuk distribusi, jika ada. Jika distribusi tidak memiliki nama domain alternatif, daftar tersebut mencakup nama domain asal untuk distribusi tersebut.

BillingRegion

Wilayah CloudFront penagihan tempat Anda menjalankan laporan, atau ALL.

TimeBucket

Jam atau hari saat data berlaku, dalam Waktu Universal Terkoordinasi (UTC).

HTTP

Jumlah permintaan HTTP yang CloudFront merespons dari lokasi tepi di wilayah yang dipilih selama setiap interval waktu untuk CloudFront distribusi yang ditentukan. Nilai-nilai meliputi:

- Jumlah GET dan HEAD permintaan, yang menyebabkan transfer data CloudFront ke pemirsa Anda
- Jumlah DELETE, OPTIONS, PATCH, POST, dan PUT permintaan, yang menyebabkan CloudFront transfer data ke asal Anda

HTTPS

Jumlah permintaan HTTPS yang CloudFront merespons dari lokasi tepi di wilayah yang dipilih selama setiap interval waktu untuk CloudFront distribusi yang ditentukan. Nilai-nilai meliputi:

- Jumlah GET dan HEAD permintaan, yang menyebabkan transfer data CloudFront ke pemirsa Anda
- Jumlah DELETE, OPTIONS, PATCH, POST, dan PUT permintaan, yang menyebabkan CloudFront transfer data ke asal Anda

HTTPBytes

Jumlah total data yang ditransfer melalui HTTP dari lokasi CloudFront tepi di wilayah penagihan yang dipilih selama periode waktu untuk CloudFront distribusi yang ditentukan. Nilai-nilai meliputi:

- Data yang ditransfer dari CloudFront pemirsa Anda sebagai tanggapan GET dan HEAD permintaan
- Data yang ditransfer dari pemirsa Anda ke CloudFront untuk DELETE, OPTIONS, PATCH, POST, dan PUT permintaan
- Data yang ditransfer dari CloudFront pemirsa Anda sebagai tanggapan terhadap DELETE, OPTIONS, PATCH, POST, dan PUT permintaan

HTTPSBytes

Jumlah total data yang ditransfer melalui HTTPS dari lokasi CloudFront tepi di wilayah penagihan yang dipilih selama periode waktu untuk CloudFront distribusi yang ditentukan. Nilai-nilai meliputi:

- Data yang ditransfer dari CloudFront pemirsa Anda sebagai tanggapan GET dan HEAD permintaan
- Data yang ditransfer dari pemirsa Anda ke CloudFront untuk DELETEOPTIONS,PATCH,POST,, dan PUT permintaan
- Data yang ditransfer dari CloudFront pemirsa Anda sebagai tanggapan terhadap DELETEOPTIONS,PATCH,,POST, dan PUT permintaan

BytesIn

Jumlah total data yang ditransfer dari asal Anda CloudFront untukDELETE,,OPTIONS,PATCH,POST, dan PUT permintaan di wilayah yang dipilih selama setiap interval waktu untuk CloudFront distribusi yang ditentukan.

BytesOut

Jumlah total data yang ditransfer melalui HTTP dan HTTPS dari CloudFront pemirsa Anda di wilayah yang dipilih selama setiap interval waktu untuk CloudFront distribusi yang ditentukan. Nilai-nilai meliputi:

- Data yang ditransfer dari CloudFront pemirsa Anda sebagai tanggapan GET dan HEAD permintaan
- Data yang ditransfer dari CloudFront pemirsa Anda sebagai tanggapan terhadap DELETEOPTIONS,PATCH,,POST, dan PUT permintaan

Bagaimana grafik penggunaan terkait dengan data dalam laporan CloudFront penggunaan

Daftar berikut menunjukkan bagaimana bagan penggunaan di CloudFront konsol sesuai dengan nilai di kolom Jenis Penggunaan dalam laporan CloudFront penggunaan.

Topik

- [Jumlah permintaan](#)
- [Data yang ditransfer oleh protokol](#)
- [Data yang ditransfer oleh tujuan](#)

Jumlah permintaan

Bagan ini menunjukkan jumlah total permintaan yang CloudFront merespons dari lokasi tepi di wilayah yang dipilih selama setiap interval waktu untuk CloudFront distribusi yang ditentukan, dipisahkan oleh protokol (HTTP atau HTTPS) dan tipe (statis, dinamis, atau proxy).

Jumlah permintaan HTTP

- **region**-Requests-HTTP-Static: Jumlah HTTP GET dan HEAD permintaan yang disajikan untuk objek dengan TTL ≥ 3600 detik
- **region**-Requests-HTTP-Dynamic: Jumlah HTTP GET dan HEAD permintaan yang disajikan untuk objek dengan TTL < 3600 detik
- **region**-Requests-HTTP-Proxy: Jumlah HTTPDELETE,,, OPTIONS PATCHPOST, dan PUT permintaan yang diteruskan ke asal Anda CloudFront

Jumlah permintaan HTTPS

- **region**-Requests-HTTPS-Static: Jumlah HTTPS GET dan HEAD permintaan yang disajikan untuk objek dengan TTL ≥ 3600 detik
- **region**-Requests-HTTPS-Dynamic: Jumlah HTTPS GET dan HEAD permintaan yang disajikan untuk objek dengan TTL < 3600 detik
- **region**-Requests-https-proxy: Jumlah HTTPSDELETE,,, OPTIONS PATCHPOST, dan permintaan yang diteruskan ke asal Anda PUT CloudFront

Data yang ditransfer oleh protokol

Bagan ini menunjukkan jumlah total data yang ditransfer dari lokasi CloudFront tepi di wilayah yang dipilih selama setiap interval waktu untuk CloudFront distribusi yang ditentukan, dipisahkan oleh protokol (HTTP atau HTTPS), jenis (statis, dinamis, atau proxy), dan tujuan (pemirsa atau asal).

Data ditransfer melalui HTTP

- **region**-Out-bytes-http-static: Byte dilayani melalui HTTP untuk objek dengan TTL ≥ 3600 detik
- **region**-Out-bytes-HTTP-Dynamic: Bytes dilayani melalui HTTP untuk objek dengan TTL < 3600 detik
- **region**-Out-bytes-http-proxy: Byte dikembalikan dari CloudFront ke pemirsa melalui HTTP sebagai tanggapan atas,,, dan permintaan DELETE OPTIONS PATCH POST PUT

- **region**-Out- OBytes -HTTP-proxy: Total byte yang ditransfer melalui HTTP dari lokasi CloudFront tepi ke asal Anda sebagai tanggapan terhadap DELETE,,,,, OPTIONS dan permintaan PATCH POST PUT

Data ditransfer melalui HTTPS

- **region**-Out-bytes-https-static: Byte disajikan melalui HTTPS untuk objek dengan TTL ≥ 3600 detik
- **region**-Out-bytes-https-Dynamic: Byte disajikan melalui HTTPS untuk objek dengan TTL < 3600 detik
- **region**-Out-bytes-https-proxy: Byte dikembalikan dari CloudFront ke pemirsa melalui HTTPS sebagai tanggapan atas,,,,, dan permintaan DELETE OPTIONS PATCH POST PUT
- **region**-Out- OBytes -HTTPS-proxy: Total byte yang ditransfer melalui HTTPS dari lokasi CloudFront tepi ke asal Anda sebagai tanggapan terhadap DELETE,,,,, dan permintaan OPTIONS PATCH POST PUT

Data yang ditransfer oleh tujuan

Bagan ini menunjukkan jumlah total data yang ditransfer dari lokasi CloudFront tepi di wilayah yang dipilih selama setiap interval waktu untuk CloudFront distribusi yang ditentukan, dipisahkan oleh tujuan (pemirsa atau asal), protokol (HTTP atau HTTPS), dan tipe (statis, dinamis, atau proxy).

Data yang ditransfer dari CloudFront pemirsa Anda

- **region**-Out-bytes-http-static: Byte dilayani melalui HTTP untuk objek dengan TTL ≥ 3600 detik
- **region**-Out-bytes-https-static: Byte disajikan melalui HTTPS untuk objek dengan TTL ≥ 3600 detik
- **region**-Out-bytes-HTTP-Dynamic: Bytes dilayani melalui HTTP untuk objek dengan TTL < 3600 detik
- **region**-Out-bytes-https-Dynamic: Byte disajikan melalui HTTPS untuk objek dengan TTL < 3600 detik
- **region**-Out-bytes-http-proxy: Byte dikembalikan dari CloudFront ke pemirsa melalui HTTP sebagai tanggapan atas,,,,, dan permintaan DELETE OPTIONS PATCH POST PUT
- **region**-Out-bytes-https-proxy: Byte dikembalikan dari CloudFront ke pemirsa melalui HTTPS sebagai tanggapan atas,,,,, dan permintaan DELETE OPTIONS PATCH POST PUT

Data yang ditransfer dari CloudFront asal Anda

- **region**-Out- OBytes -HTTP-proxy: Total byte yang ditransfer melalui HTTP dari lokasi CloudFront tepi ke asal Anda sebagai tanggapan terhadap DELETE,,,,, OPTIONS dan permintaan PATCH POST PUT
- **region**-Out- OBytes -HTTPS-proxy: Total byte yang ditransfer melalui HTTPS dari lokasi CloudFront tepi ke asal Anda sebagai tanggapan terhadap DELETE,,,,, dan permintaan OPTIONS PATCH POST PUT

Lihat laporan CloudFront pemirsa

Laporan CloudFront pemirsa mencakup informasi berikut untuk rentang tanggal apa pun dalam 60 hari sebelumnya:

- Perangkat — Jenis perangkat yang paling sering digunakan untuk mengakses konten Anda (seperti Desktop atau Seluler)
- Browser — 10 browser teratas yang paling sering digunakan untuk mengakses konten Anda (seperti Chrome atau Firefox)
- Sistem operasi — 10 sistem operasi teratas yang paling sering digunakan saat mengakses konten Anda (seperti Linux, macOS, atau Windows)
- Lokasi — 50 lokasi teratas (negara atau negara bagian/teritori AS) dari pemirsa yang paling sering mengakses konten Anda
- Anda juga dapat melihat lokasi dengan titik data per jam untuk rentang tanggal hingga 14 hari dalam 60 hari sebelumnya

Note

Anda tidak perlu mengaktifkan [pencatatan akses](#) untuk melihat bagan dan laporan pemirsa.

Topik

- [Melihat bagan dan laporan pemirsa di konsol](#)
- [Unduh data dalam format CSV](#)
- [Data yang disertakan dalam laporan pemirsa](#)

- [Bagaimana data dalam laporan lokasi terkait dengan data dalam log CloudFront standar \(log akses\)](#)

Melihat bagan dan laporan pemirsa di konsol

Anda dapat melihat bagan dan laporan CloudFront pemirsa di konsol.

Untuk melihat bagan dan laporan CloudFront pemirsa

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Viewers.
3. Di CloudFront Penampil panel, untuk Tanggal Mulai dan Tanggal Akhir, pilih rentang tanggal yang ingin Anda tampilkan grafik penampil dan laporannya.

Untuk bagan Lokasi, rentang yang tersedia bergantung pada nilai yang Anda pilih Keserbagunaan:

- Setiap Hari – Untuk menampilkan grafik dengan satu titik data per hari, pilih rentang tanggal mana pun dalam 60 hari sebelumnya.
- Per Jam – Untuk menampilkan bagan dengan satu titik data setiap jam, pilih rentang tanggal hingga 14 hari dalam 60 hari sebelumnya.

Tanggal dan waktu ada dalam Waktu Universal Terkoordinasi (UTC).

4. (Khusus Petak dan Sistem Operasi saja) Untuk Pengelompokan, tentukan apakah Anda ingin mengelompokkan browser dan sistem operasi dengan nama (Chrome, Firefox) atau dengan nama dan versi (Chrome 40.0, Firefox 35.0).
5. (Grafik lokasi saja) Untuk Keserbagunaan, menentukan apakah akan menampilkan satu titik data per hari atau satu titik data per jam dalam bagan. Jika Anda menetapkan rentang tanggal lebih dari 14 hari, opsi untuk menentukan satu titik data per jam tidak tersedia.
6. (Grafik lokasi saja) Untuk Rincian, menentukan apakah akan menampilkan lokasi teratas berdasarkan negara atau menurut negara bagian A.S.
7. Di daftar Distribusi, pilih distribusi yang ingin Anda tampilkan datanya dalam diagram penggunaan:

- Distribusi individual — Bagan menampilkan data untuk CloudFront distribusi yang dipilih. Distribusi menampilkan ID distribusi dan nama domain alternatif (CNAME) untuk distribusi, jika ada. Jika distribusi tidak memiliki nama domain alternatif, daftar tersebut mencakup nama domain asal untuk distribusi tersebut.
- Semua distribusi (terkecuali yang dihapus) – Diagram menampilkan data terjumlah untuk semua distribusi yang terkait dengan akun AWS saat ini, tidak termasuk distribusi yang telah Anda hapus.

8. Pilih Perbarui.

Untuk melihat data untuk titik data harian atau per jam dalam bagan, arahkan kursor ke titik data.

Unduh data dalam format CSV

Anda dapat mengunduh setiap laporan penampil dalam format CSV. Bagian ini menjelaskan cara mengunduh laporan dan menjelaskan nilai dalam laporan.

Untuk mengunduh laporan penampil dalam format CSV

1. Saat melihat laporan Viewer, pilih CSV.
2. Pilih data yang ingin Anda unduh, misalnya, Perangkat atau Tren Perangkat.
3. Di Pembukaan nama file kotak dialog, pilih apakah ingin membuka atau menyimpan file.

Data yang disertakan dalam laporan pemirsa

Beberapa baris pertama dari setiap laporan mencakup informasi berikut:

Versi

Versi format untuk file CSV ini.

Laporan

Nama laporan.

DistributionID

ID distribusi yang Anda jalankan laporannya, atau ALL jika Anda menjalankan laporan untuk semua distribusi.

StartDateUTC

Permulaan rentang tanggal saat Anda menjalankan laporan dalam Waktu Universal Terkoordinasi (UTC).

EndDateUTC

Akhir rentang tanggal saat Anda menjalankan laporan dalam Waktu Universal Terkoordinasi (UTC).

GeneratedTimeUTC

Tanggal dan waktu Anda menjalankan laporan, dalam Waktu Universal Terkoordinasi (UTC).

Pengelompokan (hanya laporan browser dan sistem operasi)

Apakah data dikelompokkan menurut nama atau nama dan versi peramban atau sistem operasi.

Keserbagunaan

Apakah setiap baris dalam laporan mewakili satu jam atau satu hari.

Detail (hanya laporan lokasi)

Apakah permintaan dicantumkan berdasarkan negara atau oleh negara bagian A.S.

Topik berikut menjelaskan informasi dalam laporan pemirsa yang berbeda.

Topik

- [Laporan perangkat](#)
- [Laporan tren perangkat](#)
- [Laporan browser](#)
- [Laporan tren browser](#)
- [Laporan sistem operasi](#)
- [Laporan tren sistem operasi](#)
- [Laporan lokasi](#)
- [Laporan tren lokasi](#)

Laporan perangkat

Laporan mencakup nilai-nilai berikut:

DistributionID

ID distribusi yang Anda jalankan laporannya, atau ALL jika Anda menjalankan laporan untuk semua distribusi.

FriendlyName

Nama domain alternatif (CNAME) untuk distribusi, jika ada. Jika distribusi tidak memiliki nama domain alternatif, daftar tersebut mencakup nama domain asal untuk distribusi tersebut.

Permintaan

Jumlah permintaan yang CloudFront diterima dari setiap jenis perangkat.

RequestsPct

Jumlah permintaan yang CloudFront diterima dari setiap jenis perangkat sebagai persentase dari jumlah total permintaan yang CloudFront diterima dari semua perangkat.

Khusus

Permintaan yang User-Agent Header HTTP tidak terkait dengan salah satu jenis perangkat standar, misalnya, Desktop atau Mobile.

Laporan tren perangkat

Laporan mencakup nilai-nilai berikut:

DistributionID

ID distribusi yang Anda jalankan laporannya, atau ALL jika Anda menjalankan laporan untuk semua distribusi.

FriendlyName

Nama domain alternatif (CNAME) untuk distribusi, jika ada. Jika distribusi tidak memiliki nama domain alternatif, daftar tersebut mencakup nama domain asal untuk distribusi tersebut.

TimeBucket

Jam atau hari saat data berlaku, dalam Waktu Universal Terkoordinasi (UTC).

Desktop

Jumlah permintaan yang CloudFront diterima dari komputer desktop selama periode tersebut.

Seluler

Jumlah permintaan yang CloudFront diterima dari perangkat seluler selama periode tersebut. Perangkat seluler dapat mencakup tablet dan ponsel. Jika tidak CloudFront dapat menentukan apakah permintaan berasal dari perangkat seluler atau tablet, permintaan akan dihitung di Mobile kolom.

TV Cerdas

Jumlah permintaan yang CloudFront diterima dari smart TVs selama periode tersebut.

Tablet

Jumlah permintaan yang CloudFront diterima dari tablet selama periode tersebut. Jika tidak CloudFront dapat menentukan apakah permintaan berasal dari perangkat seluler atau tablet, permintaan akan dihitung di Mobile kolom.

Tidak Diketahui

Permintaan yang User-Agent Header HTTP tidak terkait dengan salah satu jenis perangkat standar, misalnya, Desktop atau Mobile.

Kosong

Jumlah permintaan yang CloudFront diterima yang tidak menyertakan nilai di User-Agent header HTTP selama periode tersebut.

Laporan browser

Laporan mencakup nilai-nilai berikut:

DistributionID

ID distribusi yang Anda jalankan laporannya, atau ALL jika Anda menjalankan laporan untuk semua distribusi.

FriendlyName

Nama domain alternatif (CNAME) untuk distribusi, jika ada. Jika distribusi tidak memiliki nama domain alternatif, daftar tersebut mencakup nama domain asal untuk distribusi tersebut.

Kelompok

Browser atau browser dan versi yang CloudFront menerima permintaan dari, tergantung pada nilaiGrouping. Selain nama peramban, nilai yang mungkin termasuk yang berikut ini:

- Bot/Crawler – terutama permintaan dari mesin pencari yang mengindeks konten Anda.
- Kosong – permintaan yang User-Agent Header HTTP kosong.
- Lainnya — browser yang CloudFront diidentifikasi tetapi itu bukan yang paling populer. Jika Bot/Crawler, Empty, dan/atau Unknown tidak muncul di antara sembilan nilai pertama, kemudian nilai-nilai tersebut juga disertakan dalam Other.
- Tidak Diketahui – permintaan yang User-Agent Header HTTP tidak terkait dengan peramban standar. Sebagian besar permintaan dalam kategori ini berasal dari aplikasi atau skrip khusus.

Permintaan

Jumlah permintaan yang CloudFront diterima dari setiap jenis browser.

RequestsPct

Jumlah permintaan yang CloudFront diterima dari setiap jenis browser sebagai persentase dari jumlah total permintaan yang CloudFront diterima selama periode waktu tersebut.

Laporan tren browser

Laporan mencakup nilai-nilai berikut:

DistributionID

ID distribusi yang Anda jalankan laporannya, atau ALL jika Anda menjalankan laporan untuk semua distribusi.

FriendlyName

Nama domain alternatif (CNAME) untuk distribusi, jika ada. Jika distribusi tidak memiliki nama domain alternatif, daftar tersebut mencakup nama domain asal untuk distribusi tersebut.

TimeBucket

Jam atau hari saat data berlaku, dalam Waktu Universal Terkoordinasi (UTC).

(Browser)

Kolom yang tersisa dalam laporan mencantumkan peramban atau peramban dan versinya, bergantung pada nilai Grouping. Selain nama peramban, nilai yang mungkin termasuk yang berikut ini:

- Bot/Crawler – terutama permintaan dari mesin pencari yang mengindeks konten Anda.
- Kosong – permintaan yang User-Agent Header HTTP kosong.

- Lainnya — browser yang CloudFront diidentifikasi tetapi itu bukan yang paling populer. Jika Bot/Crawler, Empty, dan/atau Unknown tidak muncul di antara sembilan nilai pertama, kemudian nilai-nilai tersebut juga disertakan dalam Other.
- Tidak Diketahui – permintaan yang User-Agent Header HTTP tidak terkait dengan peramban standar. Sebagian besar permintaan dalam kategori ini berasal dari aplikasi atau skrip khusus.

Laporan sistem operasi

Laporan mencakup nilai-nilai berikut:

DistributionID

ID distribusi yang Anda jalankan laporannya, atau ALL jika Anda menjalankan laporan untuk semua distribusi.

FriendlyName

Nama domain alternatif (CNAME) untuk distribusi, jika ada. Jika distribusi tidak memiliki nama domain alternatif, daftar tersebut mencakup nama domain asal untuk distribusi tersebut.

Kelompok

Sistem operasi atau sistem operasi dan versi yang CloudFront menerima permintaan dari, tergantung pada nilaiGrouping. Selain nama sistem operasi, nilai yang mungkin termasuk yang berikut ini:

- Bot/Crawler – terutama permintaan dari mesin pencari yang mengindeks konten Anda.
- Kosong – permintaan yang User-Agent Header HTTP kosong.
- Lainnya — sistem operasi yang CloudFront diidentifikasi tetapi itu bukan yang paling populer. Jika Bot/Crawler, Empty, dan/atau Unknown tidak muncul di antara sembilan nilai pertama, kemudian nilai-nilai tersebut juga disertakan dalam Other.
- Tidak Diketahui – permintaan yang User-Agent Header HTTP tidak terkait dengan peramban standar. Sebagian besar permintaan dalam kategori ini berasal dari aplikasi atau skrip khusus.

Permintaan

Jumlah permintaan yang CloudFront diterima dari setiap jenis sistem operasi.

RequestsPct

Jumlah permintaan yang CloudFront diterima dari setiap jenis sistem operasi sebagai persentase dari jumlah total permintaan yang CloudFront diterima selama periode waktu tersebut.

Laporan tren sistem operasi

Laporan mencakup nilai-nilai berikut:

DistributionID

ID distribusi yang Anda jalankan laporannya, atau ALL jika Anda menjalankan laporan untuk semua distribusi.

FriendlyName

Nama domain alternatif (CNAME) untuk distribusi, jika ada. Jika distribusi tidak memiliki nama domain alternatif, daftar tersebut mencakup nama domain asal untuk distribusi tersebut.

TimeBucket

Jam atau hari saat data berlaku, dalam Waktu Universal Terkoordinasi (UTC).

(Sistem operasi)

Kolom yang tersisa di daftar laporan sistem operasi atau sistem operasi dan versinya, bergantung pada nilai Grouping. Selain nama sistem operasi, nilai yang mungkin termasuk yang berikut ini:

- Bot/Crawler – terutama permintaan dari mesin pencari yang mengindeks konten Anda.
- Kosong – permintaan yang User-Agent Header HTTP kosong.
- Lainnya — sistem operasi yang CloudFront diidentifikasi tetapi itu bukan yang paling populer. Jika Bot/Crawler, Empty, dan/atau Unknown tidak muncul di antara sembilan nilai pertama, kemudian nilai-nilai tersebut juga disertakan dalam Other.
- Tidak Diketahui – permintaan di mana sistem operasi tidak ditentukan dalam User-Agent Header HTTP.

Laporan lokasi

Laporan mencakup nilai-nilai berikut:

DistributionID

ID distribusi yang Anda jalankan laporannya, atau ALL jika Anda menjalankan laporan untuk semua distribusi.

FriendlyName

Nama domain alternatif (CNAME) untuk distribusi, jika ada. Jika distribusi tidak memiliki nama domain alternatif, daftar tersebut mencakup nama domain asal untuk distribusi tersebut.

LocationCode

Singkatan untuk lokasi yang CloudFront menerima permintaan dari. Untuk informasi selengkapnya tentang nilai yang memungkinkan, lihat deskripsi Lokasi di [Bagaimana data dalam laporan lokasi terkait dengan data dalam log CloudFront standar \(log akses\)](#).

LocationName

Nama lokasi yang CloudFront menerima permintaan dari.

Permintaan

Jumlah permintaan yang CloudFront diterima dari setiap lokasi.

RequestsPct

Jumlah permintaan yang CloudFront diterima dari setiap lokasi sebagai persentase dari jumlah total permintaan yang CloudFront diterima dari semua lokasi selama periode waktu tersebut.

TotalBytes

Jumlah byte yang CloudFront disajikan kepada pemirsa di negara atau negara bagian ini, untuk distribusi dan periode yang ditentukan.

Laporan tren lokasi

Laporan mencakup nilai-nilai berikut:

DistributionID

ID distribusi yang Anda jalankan laporannya, atau ALL jika Anda menjalankan laporan untuk semua distribusi.

FriendlyName

Nama domain alternatif (CNAME) untuk distribusi, jika ada. Jika distribusi tidak memiliki nama domain alternatif, daftar tersebut mencakup nama domain asal untuk distribusi tersebut.

TimeBucket

Jam atau hari saat data berlaku, dalam Waktu Universal Terkoordinasi (UTC).

(Lokasi)

Kolom yang tersisa dalam laporan mencantumkan lokasi yang CloudFront menerima permintaan dari. Untuk informasi selengkapnya tentang nilai yang memungkinkan, lihat deskripsi Lokasi di

[Bagaimana data dalam laporan lokasi terkait dengan data dalam log CloudFront standar \(log akses\).](#)

Bagaimana data dalam laporan lokasi terkait dengan data dalam log CloudFront standar (log akses)

Daftar berikut menunjukkan bagaimana data dalam laporan Lokasi di CloudFront konsol sesuai dengan nilai dalam log CloudFront akses. Untuk informasi selengkapnya tentang log CloudFront akses, lihat [Akses log \(log standar\)](#).

Lokasi

Negara atau negara bagian A.S. menyatakan bahwa penampil berada di dalamnya. Dalam log akses, `c-ip` kolom berisi alamat IP perangkat yang sedang berjalan oleh penampil. Kami menggunakan data geolokasi untuk mengidentifikasi lokasi geografis perangkat berdasarkan alamat IP.

Jika Anda menampilkan laporan Lokasi menurut negara, daftar negara didasarkan pada [ISO 3166-2, Kode untuk representasi nama negara dan subdivisi mereka — Bagian 2: Kode subdivisi negara](#). Daftar negara mencakup nilai-nilai tambahan berikut:

- Proksi Anonim – Permintaan yang berasal dari proksi anonim.
- Penyedia Satelit – Permintaan yang berasal dari penyedia satelit yang menyediakan layanan internet ke beberapa negara. Pemirsa mungkin berada di negara-negara dengan risiko penipuan yang tinggi.
- Eropa (Tidak Diketahui) – Permintaan yang berasal dari IP di blok yang digunakan oleh beberapa negara Eropa. Negara tempat permintaan berasal tidak dapat ditentukan. CloudFront menggunakan Eropa (Tidak Diketahui) sebagai default.
- Asia/Pasifik (Tidak Diketahui) — Permintaan berasal dari IP di blok yang digunakan oleh beberapa negara di kawasan tersebut. Asia/Pacific Negara tempat permintaan berasal tidak dapat ditentukan. CloudFront menggunakan Asia/Pasifik (Tidak Diketahui) sebagai default.

Jika Anda menampilkan Lokasi laporan berdasarkan negara bagian A.S., perhatikan bahwa laporan dapat mencakup wilayah A.S. dan wilayah Angkatan Bersenjata A.S.

Note

Jika tidak CloudFront dapat menentukan lokasi pengguna, lokasi akan muncul sebagai Tidak Dikenal dalam laporan penampil.

Jumlah Permintaan

Total jumlah permintaan dari negara atau A.S. menyatakan bahwa penampil berada di, untuk distribusi dan periode tertentu. Nilai ini umumnya sesuai dengan jumlah GET permintaan dari alamat IP di negara atau negara bagian dalam log CloudFront akses.

% Permintaan

Salah satu dari yang berikut, bergantung pada nilai yang Anda pilih untuk Rincian:

- Negara – Permintaan dari negara ini sebagai persentase dari jumlah total permintaan.
- Negara Bagian A.S. – Permintaan dari negara bagian ini sebagai persentase dari jumlah total permintaan dari Amerika Serikat.

Jika permintaan berasal dari lebih dari 50 negara, maka Anda tidak dapat menghitung Minta % berdasarkan data di tabel ini karena Jumlah Permintaan kolom tidak menyertakan semua permintaan selama periode tertentu.

Byte

Jumlah byte yang CloudFront disajikan kepada pemirsa di negara atau negara bagian ini, untuk distribusi dan periode yang ditentukan. Untuk mengubah tampilan data di kolom ini menjadi KB, MB, atau GB, pilih tautan di judul kolom.

Pantau CloudFront metrik dengan Amazon CloudWatch

Amazon CloudFront terintegrasi dengan Amazon CloudWatch dan secara otomatis menerbitkan metrik operasional untuk distribusi dan fungsi edge (baik [Lambda @Edge](#) dan Functions). CloudFront Anda dapat menggunakan metrik ini untuk memecahkan masalah, melacak, dan men-debug masalah. Banyak dari metrik ini ditampilkan dalam satu set grafik di CloudFront konsol, dan juga dapat diakses dengan menggunakan CloudFront API atau CLI. Semua metrik ini tersedia di [CloudWatch konsol](#) atau melalui CloudWatch API atau CLI. CloudFront metrik tidak dihitung terhadap [CloudWatch kuota \(sebelumnya dikenal sebagai batas\)](#) dan tidak dikenakan biaya tambahan.

Selain metrik default untuk CloudFront distribusi, Anda dapat mengaktifkan metrik tambahan dengan biaya tambahan. Metrik tambahan berlaku untuk CloudFront distribusi, dan harus dihidupkan untuk setiap distribusi secara terpisah. Untuk informasi lebih lanjut tentang biaya, lihat [the section called “Perkirakan biaya untuk CloudFront metrik tambahan”](#).

Anda juga dapat mengatur alarm berdasarkan metrik ini di CloudFront konsol, atau di konsol, API, atau CLI. CloudWatch Misalnya, Anda dapat menyetel alarm berdasarkan `5xxErrorRate` metrik, yang mewakili persentase semua permintaan penampil yang kode status HTTP responsnya berada dalam kisaran 500 hingga 599, inklusif. Ketika tingkat kesalahan mencapai nilai tertentu untuk jangka waktu tertentu, misalnya, 5% dari permintaan selama 5 menit terus menerus, alarm dipicu. Anda menentukan nilai alarm dan unit waktunya saat Anda membuat alarm.

Catatan

- Saat Anda membuat CloudWatch alarm di CloudFront konsol, alarm akan membuatnya untuk Anda di Wilayah AS Timur (Virginia Utara) (`us-east-1`). Jika Anda membuat alarm dari CloudWatch konsol, Anda harus menggunakan Wilayah yang sama. Karena CloudFront merupakan layanan global, metrik untuk layanan dikirim ke US East (Virginia N.).
- Saat membuat alarm, [CloudWatch harga standar](#) berlaku.

Topik

- [Tampilan CloudFront dan metrik fungsi tepi](#)
- [Buat alarm untuk metrik](#)
- [Unduh data metrik dalam format CSV](#)
- [Jenis metrik untuk CloudFront](#)

Tampilan CloudFront dan metrik fungsi tepi

Anda dapat melihat metrik operasional tentang CloudFront distribusi dan [fungsi edge](#) di konsol. CloudFront

Untuk melihat CloudFront dan metrik fungsi tepi di CloudFront

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Pemantauan.
3. Untuk melihat grafik tentang aktivitas untuk CloudFront distribusi atau fungsi tepi tertentu, pilih salah satunya, lalu pilih Lihat metrik distribusi atau Lihat metrik.
4. Anda dapat menyesuaikan grafik dengan melakukan hal berikut:
 - a. Untuk mengubah rentang waktu untuk informasi yang ditampilkan di grafik, pilih 1 jam (1 jam), 3 jam (3 jam), atau rentang lainnya, atau tentukan rentang kustom.
 - b. Untuk mengubah seberapa sering CloudFront memperbarui informasi dalam grafik, pilih panah bawah di sebelah ikon penyegaran, lalu pilih kecepatan refresh. Kecepatan refresh default adalah 1 menit, tetapi Anda dapat memilih opsi lain.
5. Untuk melihat CloudFront grafik di CloudWatch konsol, pilih Tambahkan ke dasbor. Anda harus menggunakan Wilayah AS Timur (Virginia N.) untuk melihat grafik di CloudWatch konsol.

Topik

- [Metrik CloudFront distribusi default](#)
- [Aktifkan metrik CloudFront distribusi tambahan](#)
- [Metrik fungsi Lambda @Edge default](#)
- [Metrik CloudFront Fungsi Default](#)

Metrik CloudFront distribusi default

Metrik default berikut disertakan untuk semua CloudFront distribusi, tanpa biaya tambahan:

Permintaan

Jumlah total permintaan penampil yang diterima oleh CloudFront, untuk semua metode HTTP dan untuk permintaan HTTP dan HTTPS.

Byte yang diunduh

Jumlah total byte yang diunduh oleh pemirsa untuk GET dan HEAD permintaan.

Byte yang diunggah

Jumlah total byte yang diunggah, digunakan CloudFront, POST dan PUT permintaan OPTIONS pemirsa.

4xx tingkat kesalahan

Persentase dari semua permintaan penampil yang kode status HTTP responsnya 4xx.

5xx tingkat kesalahan

Persentase dari semua permintaan penampil yang kode status HTTP responsnya 5xx.

Total tingkat kesalahan

Persentase dari semua permintaan penampil yang kode status HTTP responsnya 4xx atau 5xx.

Metrik ini ditampilkan dalam grafik untuk setiap CloudFront distribusi di halaman Pemantauan konsol.. CloudFront Pada setiap grafik, total ditampilkan pada granularitas 1 menit. Selain melihat grafik, Anda juga dapat [unduh laporan metrik sebagai file CSV](#).

Aktifkan metrik CloudFront distribusi tambahan

Selain metrik default, Anda dapat mengaktifkan metrik tambahan dengan biaya tambahan. Untuk informasi lebih lanjut tentang biaya, lihat [the section called “Perkiraan biaya untuk CloudFront metrik tambahan”](#).

Metrik tambahan ini harus dihidupkan untuk setiap distribusi secara terpisah:

Laju hit cache

Persentase semua permintaan yang dapat di-cache yang CloudFront menyajikan konten dari cacheable. HTTP POST dan PUT permintaan, dan kesalahan, tidak dianggap sebagai permintaan yang dapat disimpan.

Latensi asal

Total waktu yang dihabiskan dari saat CloudFront menerima permintaan hingga saat mulai memberikan respons ke jaringan (bukan penampil), untuk permintaan yang disajikan dari asal, bukan CloudFront cache. Ini juga dikenal sebagai latensi byte pertama, atau time-to-first-byte.

Tingkat kesalahan menurut kode status

Persentase semua permintaan penampil yang kode status HTTP responsnya adalah kode tertentu dalam 5xx rentang 4xx atau. Metrik ini tersedia untuk semua kode kesalahan berikut: 401, 403, 404, 502, 503, dan 504.

Anda dapat mengaktifkan metrik tambahan di CloudFront konsol, dengan CloudFormation, dengan AWS Command Line Interface (AWS CLI), atau dengan CloudFront API.

Console

Untuk mengaktifkan metrik tambahan

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Pemantauan.
3. Pilih distribusi untuk mengaktifkan metrik tambahan, lalu pilih Lihat metrik distribusi.
4. Pilih Kelola metrik tambahan.
5. Di jendela Kelola metrik tambahan, aktifkan Diaktifkan. Setelah mengaktifkan metrik tambahan, Anda dapat menutup jendela Kelola metrik tambahan.

Setelah Anda mengaktifkan metrik tambahan, metrik tersebut ditampilkan dalam grafik. Pada setiap grafik, total ditampilkan pada granularitas 1 menit. Selain melihat grafik, Anda juga dapat [unduh laporan metrik sebagai file CSV](#).

CloudFormation

Untuk mengaktifkan metrik tambahan CloudFormation, gunakan jenis `AWS::CloudFront::MonitoringSubscription` sumber daya. Contoh berikut menunjukkan sintaks CloudFormation template, dalam format YAMAL, untuk mengaktifkan metrik tambahan.

```
Type: AWS::CloudFront::MonitoringSubscription
Properties:
  DistributionId: EDFDVBD6EXAMPLE
  MonitoringSubscription:
    RealtimeMetricsSubscriptionConfig:
      RealtimeMetricsSubscriptionStatus: Enabled
```

CLI

Untuk mengelola metrik tambahan dengan AWS Command Line Interface (AWS CLI), gunakan salah satu perintah berikut:

Untuk mengaktifkan metrik tambahan untuk distribusi

- Gunakan `create-monitoring-subscription` seperti pada contoh berikut. Ganti **EDFDVBD6EXAMPLE** dengan ID distribusi tempat Anda mengaktifkan metrik tambahan.

```
aws cloudfront create-monitoring-subscription --  
distribution-id EDFDVBD6EXAMPLE --monitoring-subscription  
RealtimeMetricsSubscriptionConfig={RealtimeMetricsSubscriptionStatus=Enabled}
```

Untuk melihat apakah metrik tambahan diaktifkan untuk distribusi

- Gunakan `get-monitoring-subscription` seperti pada contoh berikut. Ganti **EDFDVBD6EXAMPLE** dengan ID distribusi yang Anda periksa.

```
aws cloudfront get-monitoring-subscription --distribution-id EDFDVBD6EXAMPLE
```

Untuk menonaktifkan metrik tambahan untuk distribusi

- Gunakan `delete-monitoring-subscription` seperti pada contoh berikut. Ganti **EDFDVBD6EXAMPLE** dengan ID distribusi tempat Anda menonaktifkan metrik tambahan.

```
aws cloudfront delete-monitoring-subscription --distribution-id EDFDVBD6EXAMPLE
```

API

Untuk mengelola metrik tambahan dengan CloudFront API, gunakan salah satu operasi API berikut.

- Untuk mengaktifkan metrik tambahan untuk distribusi, gunakan [CreateMonitoringSubscription](#).

- Untuk melihat apakah metrik tambahan diaktifkan untuk distribusi, gunakan [GetMonitoringSubscription](#).
- Untuk mematikan metrik tambahan untuk distribusi, gunakan [DeleteMonitoringSubscription](#).

Untuk informasi selengkapnya tentang operasi API ini, lihat dokumentasi referensi API untuk AWS SDK atau klien API lainnya.

Perkirakan biaya untuk CloudFront metrik tambahan

Saat Anda mengaktifkan metrik tambahan untuk distribusi, CloudFront kirimkan hingga 8 metrik ke CloudWatch Wilayah AS Timur (Virginia Utara). CloudWatch membebankan tarif tetap rendah untuk setiap metrik. Nilai ini hanya dibebankan sekali per bulan, per metrik (hingga 8 metrik per distribusi). Ini adalah tarif tetap, jadi biaya Anda tetap sama terlepas dari jumlah permintaan atau tanggapan yang diterima atau dikirim CloudFront distribusi. Untuk tarif per metrik, lihat [halaman CloudWatch harga Amazon dan kalkulator CloudWatch harga](#). Biaya API tambahan berlaku saat Anda mengambil metrik dengan API. CloudWatch

Metrik fungsi Lambda @Edge default

Anda dapat menggunakan CloudWatch metrik untuk memantau, secara real time, masalah dengan fungsi Lambda @Edge Anda. Tidak ada biaya tambahan untuk metrik ini.

Saat Anda melampirkan fungsi Lambda @Edge ke perilaku cache dalam CloudFront distribusi, Lambda mulai mengirim metrik secara otomatis. CloudWatch Metrik tersedia untuk semua Wilayah Lambda, tetapi untuk melihat metrik di CloudWatch konsol atau mendapatkan data metrik dari CloudWatch API, Anda harus menggunakan Wilayah AS Timur (Virginia Utara) (). us-east-1 Nama grup metrik diformat sebagai:AWS/CloudFront/*distribution-ID*, di *distribution-ID* mana ID CloudFront distribusi yang dikaitkan dengan fungsi Lambda @Edge. Untuk informasi selengkapnya tentang CloudWatch metrik, lihat [Panduan CloudWatch Pengguna Amazon](#).

Metrik default berikut ditampilkan dalam grafik untuk setiap fungsi Lambda @Edge di halaman Pemantauan konsol: CloudFront

- 5xx tingkat kesalahan untuk Lambda@Edge
- Kesalahan eksekusi lambda
- Respons tidak valid Lambda
- Trotel Lambda

Grafik mencakup jumlah invokasi, kesalahan, trotel, dan sebagainya. Pada setiap grafik, total ditampilkan pada granularitas 1 menit, dikelompokkan berdasarkan Wilayah AWS .

Jika Anda melihat lonjakan kesalahan yang ingin Anda selidiki, Anda dapat memilih fungsi dan kemudian melihat file log berdasarkan AWS Wilayah, hingga Anda menentukan fungsi mana yang menyebabkan masalah dan di AWS Wilayah mana. Untuk informasi lebih lanjut tentang pemecahan masalah kesalahan Lambda@Edge, lihat:

- [the section called “Cara menentukan jenis kegagalan”](#)
- [Empat Langkah untuk Debugging Pengiriman Konten Anda AWS](#)

Metrik CloudFront Fungsi Default

CloudFront Fungsi mengirimkan metrik operasional ke Amazon CloudWatch sehingga Anda dapat memantau fungsi Anda. Melihat metrik ini dapat membantu Anda memecahkan masalah, melacak, dan memecahkan masalah. CloudFront Fungsi menerbitkan metrik berikut ke: CloudWatch

- Permintaan(FunctionInvocations) – Frekuensi fungsi dimulai (dipanggil) dalam jangka waktu tertentu.
- Kesalahan validasi(FunctionValidationErrors) – Jumlah kesalahan validasi yang dihasilkan oleh fungsi dalam jangka waktu tertentu. Kesalahan validasi terjadi ketika fungsi berjalan berhasil tetapi mengembalikan data yang tidak valid ([objek peristiwa](#) yang tidak valid).
- Kesalahan eksekusi(FunctionExecutionErrors) – Jumlah kesalahan eksekusi yang terjadi dalam jangka waktu tertentu. Eksekusi kesalahan terjadi ketika fungsi gagal untuk menyelesaikan secara tuntas.
- Pemanfaatan komputasi(FunctionComputeUtilization) – Jumlah waktu yang digunakan fungsi untuk berjalan sebagai persentase dari waktu maksimum yang diizinkan. Misalnya, pemanfaatan komputasi 35 berarti bahwa fungsi selesai pada 35% dari waktu maksimum yang diizinkan. Metrik ini adalah angka antara 0 dan 100.

Jika nilai ini mencapai atau mendekati 100, fungsi telah digunakan atau hampir menggunakan waktu eksekusi yang diizinkan dan permintaan berikutnya mungkin dibatasi. Jika fungsi Anda berjalan pada pemanfaatan 80% atau lebih, kami sarankan Anda meninjau fungsi Anda untuk mengurangi waktu eksekusi dan meningkatkan pemanfaatan. Misalnya, Anda mungkin ingin hanya mencatat kesalahan, menyederhanakan ekspresi regex kompleks, atau menghapus penguraian objek JSON kompleks yang tidak perlu.

- **Throttles (FunctionThrottles)** — Berapa kali fungsi itu dibatasi dalam periode waktu tertentu. Fungsi dapat dibatasi karena alasan berikut:
 - Fungsi terus menerus melebihi waktu maksimum yang diizinkan untuk eksekusi
 - Fungsi ini menghasilkan kesalahan kompilasi
 - Ada jumlah permintaan per detik yang luar biasa tinggi

CloudFront KeyValueCollection juga mengirimkan metrik operasional berikut ke Amazon CloudWatch:

- **Read requests (KvsReadRequests)** — Berapa kali fungsi berhasil membaca dari penyimpanan nilai kunci dalam periode waktu tertentu.
- **Kesalahan baca (KvsReadErrors)** - Berapa kali fungsi gagal membaca dari penyimpanan nilai kunci dalam periode waktu tertentu.

Semua metrik ini dipublikasikan CloudWatch di Wilayah AS Timur (Virginia Utara) (`us-east-1`), di namespace. CloudFront Anda juga dapat melihat metrik ini di CloudWatch konsol. Di CloudWatch konsol, Anda dapat melihat metrik per fungsi atau per fungsi per distribusi.

Anda juga dapat menggunakan CloudWatch untuk mengatur alarm berdasarkan metrik ini. Misalnya, Anda dapat menyetel alarm berdasarkan metrik waktu eksekusi (`FunctionComputeUtilization`), yang mewakili persentase waktu yang tersedia yang diperlukan fungsi untuk dijalankan. Ketika waktu eksekusi mencapai nilai tertentu untuk jangka waktu tertentu. Misalnya, jika Anda memilih lebih dari 70% waktu yang tersedia selama 15 menit terus menerus, alarm dipicu. Anda menentukan nilai alarm dan unit waktunya saat Anda membuat alarm.

Note

CloudFront Fungsi mengirimkan metrik CloudWatch hanya untuk fungsi di LIVE tahap yang berjalan sebagai respons terhadap permintaan dan tanggapan produksi. Saat Anda [menguji suatu fungsi](#), CloudFront tidak mengirim metrik apa pun ke CloudWatch. Output pengujian berisi informasi tentang kesalahan, pemanfaatan komputasi, dan log fungsi (`console.log()` pernyataan), tetapi informasi ini tidak dikirim ke CloudWatch.

Untuk informasi tentang cara mendapatkan metrik ini dengan CloudWatch API, lihat [the section called “CloudFront metrik”](#).

Buat alarm untuk metrik

Di CloudFront konsol, Anda dapat mengatur alarm untuk memberi tahu Anda melalui Amazon Simple Notification Service (Amazon SNS) berdasarkan metrik tertentu. CloudFront

Untuk membuat alarm untuk metrik

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Alarm.
3. Pilih Buat alarm.
4. Untuk Detail, tentukan yang berikut ini:
 - a. Nama alarm — Nama untuk alarm.
 - b. CloudFront Distribusi — Distribusi yang Anda buat alarm untuk.
5. Untuk Kondisi, tentukan yang berikut ini:
 - a. Metrik — Metrik yang Anda buat alarm untuk.
 - b. “JIKA” <condition>- Ambang batas kapan CloudWatch harus memicu alarm dan mengirim pemberitahuan ke topik Amazon SNS. Misalnya, untuk menerima pemberitahuan saat 5xx tingkat kesalahan melebihi 1%, sebutkan hal berikut:

Tingkat kesalahan 5xx > 1
 - c. “UNTUK” periode berturut-turut — Periode waktu bahwa kondisi harus dipenuhi sebelum memicu alarm. Saat Anda memilih nilai, arahkan keseimbangan yang tepat antara nilai yang tidak mengkhawatirkan masalah sementara, tetapi akan mengkhawatirkan masalah berkelanjutan atau nyata.
 - d. (Opsional) Beri tahu - Topik Amazon SNS untuk mengirim pemberitahuan jika metrik ini memicu alarm.
6. Pilih Buat alarm.

Catatan

- Saat Anda memasukkan nilai untuk kondisi tersebut, gunakan bilangan bulat tanpa tanda baca. Misalnya, untuk menentukan seribu, masukkan **1000**.
- Untuk 4xx, 5xx, dan total tingkat kesalahan, nilai yang Anda tentukan adalah persentase.

- Untuk permintaan, byte yang diunduh, dan byte diunggah, nilai yang Anda tentukan adalah unit. Misalnya, 1073742000 byte.

Untuk informasi selengkapnya tentang membuat topik Amazon SNS, lihat [Membuat topik Amazon SNS](#) di Panduan Pengembang Layanan Pemberitahuan Sederhana Amazon.

Unduh data metrik dalam format CSV

Anda dapat mengunduh data CloudWatch metrik untuk CloudFront distribusi dalam format CSV.

Untuk mengunduh data metrik dalam format CSV

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Di panel navigasi, pilih Pemantauan.
3. Pilih distribusi dan kemudian pilih Lihat metrik distribusi.
4. Pilih Unduh CSV dan kemudian pilih periode waktu (misalnya, Untuk 1 hari terakhir (periode 1 jam)).
5. Setelah file Anda diunduh, buka untuk melihat informasi berikut.

Topik

- [Informasi tentang laporan](#)
- [Data dalam laporan metrik](#)

Informasi tentang laporan

Beberapa baris pertama pada laporan mencakup informasi berikut:

Versi

Versi CloudFront pelaporan.

Laporan

Nama laporan.

DistributionID

ID distribusi yang Anda jalankan.

StartDateUTC

Permulaan rentang tanggal saat Anda menjalankan laporan dalam Waktu Universal Terkoordinasi (UTC).

EndDateUTC

Akhir rentang tanggal saat Anda menjalankan laporan dalam Waktu Universal Terkoordinasi (UTC).

GeneratedTimeUTC

Tanggal dan waktu Anda menjalankan laporan, dalam Waktu Universal Terkoordinasi (UTC).

Keserbagunaan

Periode waktu untuk setiap baris dalam laporan, misalnya, ONE_MINUTE.

Data dalam laporan metrik

Laporan mencakup nilai-nilai berikut:

DistributionID

ID distribusi yang Anda jalankan.

FriendlyName

Nama domain alternatif (CNAME) untuk distribusi, jika ada. Jika distribusi tidak memiliki nama domain alternatif, daftar tersebut mencakup nama domain asal untuk distribusi tersebut.

TimeBucket

Jam atau hari saat data berlaku, dalam Waktu Universal Terkoordinasi (UTC).

Permintaan

Total jumlah permintaan untuk semua kode status HTTP (misalnya, 200, 404, dan sebagainya) serta semua metode (misalnya, GET, HEAD, POST, dan sebagainya) selama periode waktu tersebut.

BytesDownloaded

Jumlah byte yang mengunduh penampil untuk distribusi yang ditentukan selama periode waktu.

BytesUploaded

Jumlah byte yang di-upload untuk distribusi tertentu selama periode waktu.

TotalErrorRatePct

Persentase permintaan di mana kode status HTTP adalah 4xx atau 5xx kesalahan untuk distribusi tertentu selama periode waktu.

4 xxErrorRate persen

Persentase permintaan di mana kode status HTTP adalah 4xx kesalahan untuk distribusi tertentu selama periode waktu.

5 xxErrorRate persen

Persentase permintaan di mana kode status HTTP adalah 5xx kesalahan untuk distribusi tertentu selama periode waktu.

Jika Anda telah [mengaktifkan metrik tambahan](#) untuk distribusi Anda, maka laporan tersebut juga menyertakan nilai tambahan berikut:

401 ErrorRatePct

Persentase permintaan di mana kode status HTTP adalah 401 kesalahan untuk distribusi tertentu selama periode waktu.

403 ErrorRatePct

Persentase permintaan di mana kode status HTTP adalah 403 kesalahan untuk distribusi tertentu selama periode waktu.

404 ErrorRatePct

Persentase permintaan di mana kode status HTTP adalah 404 kesalahan untuk distribusi tertentu selama periode waktu.

502 ErrorRatePct

Persentase permintaan di mana kode status HTTP adalah 502 kesalahan untuk distribusi tertentu selama periode waktu.

503 ErrorRatePct

Persentase permintaan di mana kode status HTTP adalah 503 kesalahan untuk distribusi tertentu selama periode waktu.

504 ErrorRatePct

Persentase permintaan di mana kode status HTTP adalah 504 kesalahan untuk distribusi tertentu selama periode waktu.

OriginLatency

Total waktu yang dihabiskan, dalam milidetik, dari saat CloudFront menerima permintaan hingga saat mulai memberikan respons ke jaringan (bukan penampil), untuk permintaan yang disajikan dari asal, bukan CloudFront cache. Ini juga dikenal sebagai latensi byte pertama, atau time-to-first-byte.

CacheHitRate

Persentase semua permintaan yang dapat di-cache yang CloudFront menyajikan konten dari cacheable. HTTP POST dan PUT permintaan, dan kesalahan, tidak dianggap sebagai permintaan yang dapat disimpan.

Jenis metrik untuk CloudFront

Anda dapat menggunakan CloudWatch API atau AWS Command Line Interface (AWS CLI) untuk mendapatkan CloudFront metrik dalam program atau aplikasi yang Anda buat. Anda dapat menggunakan data mentah untuk membuat dasbor khusus, alat alarm Anda sendiri, dan sebagainya.

Untuk informasi selengkapnya, lihat [get-metric-data](#) di Referensi AWS CLI Perintah atau operasi [GetMetricData](#) API di Referensi Amazon CloudWatch API.

Topik

- [Nilai untuk semua CloudFront metrik](#)
- [Nilai untuk metrik CloudFront distribusi](#)
- [Nilai untuk metrik CloudFront fungsi](#)

Note

Untuk mendapatkan CloudFront metrik dari CloudWatch API, Anda harus menggunakan Wilayah AS Timur (Virginia N.) (us-east-1). Anda juga perlu mengetahui nilai dan jenis tertentu untuk setiap metrik.

Nilai untuk semua CloudFront metrik

Nilai berikut berlaku untuk semua CloudFront metrik:

Namespace

Nilai untuk Namespace selalu `AWS/CloudFront`.

Dimensi

Setiap CloudFront metrik memiliki dimensi sebagai berikut:

DistributionId

ID CloudFront distribusi yang ingin Anda dapatkan metriknya.

FunctionName

Nama fungsi (dalam CloudFront Fungsi) yang ingin Anda dapatkan metriknya.

Dimensi ini hanya berlaku untuk fungsi.

Region

Nilai untuk Region selalu `Global`, karena CloudFront adalah layanan global.

Nilai untuk metrik CloudFront distribusi

Gunakan informasi dari daftar berikut untuk mendapatkan detail tentang metrik CloudFront distribusi tertentu dari CloudWatch API. Beberapa metrik ini hanya tersedia jika Anda mengaktifkan metrik tambahan untuk distribusi.

Note

Hanya satu statistik, `Average` atau `Sum`, berlaku untuk setiap metrik. Daftar berikut menentukan statistik yang berlaku untuk metrik tersebut.

4xx tingkat kesalahan

Persentase dari semua permintaan penampil yang kode status HTTP responsnya 4xx.

- Nama metrik: `4xxErrorRate`

- Statistik yang valid: Average
- Unit: Percent

401 tingkat kesalahan

Persentase dari semua permintaan penampil yang kode status HTTP responsnya 401. Untuk mendapatkan metrik ini, Anda harus [mengaktifkan metrik tambahan](#) terlebih dahulu.

- Nama metrik: 401ErrorRate
- Statistik yang valid: Average
- Unit: Percent

403 tingkat kesalahan

Persentase dari semua permintaan penampil yang kode status HTTP responsnya 403. Untuk mendapatkan metrik ini, Anda harus [mengaktifkan metrik tambahan](#) terlebih dahulu.

- Nama metrik: 403ErrorRate
- Statistik yang valid: Average
- Unit: Percent

404 tingkat kesalahan

Persentase dari semua permintaan penampil yang kode status HTTP responsnya 404. Untuk mendapatkan metrik ini, Anda harus [mengaktifkan metrik tambahan](#) terlebih dahulu.

- Nama metrik: 404ErrorRate
- Statistik yang valid: Average
- Unit: Percent

5xx tingkat kesalahan

Persentase dari semua permintaan penampil yang kode status HTTP responsnya 5xx.

- Nama metrik: 5xxErrorRate
- Statistik yang valid: Average
- Unit: Percent

502 tingkat kesalahan

Persentase dari semua permintaan penampil yang kode status HTTP responsnya 502. Untuk mendapatkan metrik ini, Anda harus [mengaktifkan metrik tambahan](#) terlebih dahulu.

- Nama metrik: 502ErrorRate
- Statistik yang valid: Average
- Unit: Percent

503 tingkat kesalahan

Persentase dari semua permintaan penampil yang kode status HTTP responsnya 503. Untuk mendapatkan metrik ini, Anda harus [mengaktifkan metrik tambahan](#) terlebih dahulu.

- Nama metrik: 503ErrorRate
- Statistik yang valid: Average
- Unit: Percent

504 tingkat kesalahan

Persentase dari semua permintaan penampil yang kode status HTTP responsnya 504. Untuk mendapatkan metrik ini, Anda harus [mengaktifkan metrik tambahan](#) terlebih dahulu.

- Nama metrik: 504ErrorRate
- Statistik yang valid: Average
- Unit: Percent

Byte yang diunduh

Jumlah total byte yang diunduh oleh pemirsa untuk GET dan HEAD permintaan.

- Nama metrik: BytesDownloaded
- Statistik yang valid: Sum
- Unit: None

Byte yang diunggah

Jumlah total byte yang diunggah, digunakan CloudFront, POST dan PUT permintaan OPTIONS pemirsa.

- Nama metrik: BytesUploaded
- Statistik yang valid: Sum
- Unit: None

Laju hit cache

Persentase semua permintaan cache yang CloudFront menyajikan konten dari cacheable. HTTP POST dan PUT permintaan, dan kesalahan, tidak dianggap sebagai permintaan yang dapat

disimpan. Untuk mendapatkan metrik ini, Anda harus [mengaktifkan metrik tambahan](#) terlebih dahulu.

- Nama metrik: `CacheHitRate`
- Statistik yang valid: `Average`
- Unit: `Percent`

Latensi asal

Total waktu yang dihabiskan, dalam milidetik, dari saat CloudFront menerima permintaan hingga saat mulai memberikan respons ke jaringan (bukan penampil), untuk permintaan yang disajikan dari asal, bukan CloudFront cache. Ini juga dikenal sebagai latensi byte pertama, atau time-to-first-byte. Untuk mendapatkan metrik ini, Anda harus [mengaktifkan metrik tambahan](#) terlebih dahulu.

- Nama metrik: `OriginLatency`
- Statistik yang valid: `Percentile`
- Unit: `Milliseconds`

Note

Untuk mendapatkan `Percentile` statistik dari CloudWatch API, gunakan `ExtendedStatistics` parameter, bukan `Statistics`. Untuk informasi selengkapnya, lihat [GetMetricStatistics](#) di Referensi Amazon CloudWatch API, atau dokumentasi referensi untuk [AWS SDKs](#).

Permintaan

Jumlah total permintaan penampil yang diterima oleh CloudFront, untuk semua metode HTTP dan untuk permintaan HTTP dan HTTPS.

- Nama metrik: `Requests`
- Statistik yang valid: `Sum`
- Unit: `None`

Total tingkat kesalahan

Persentase dari semua permintaan penampil yang kode status HTTP responsnya 4xx atau 5xx.

- Nama metrik: `TotalErrorRate`

- Statistik yang valid: Average
- Unit: Percent

Nilai untuk metrik CloudFront fungsi

Gunakan informasi dari daftar berikut untuk mendapatkan detail tentang metrik CloudFront fungsi tertentu dari CloudWatch API.

Note

Hanya satu statistik, Average atau Sum, berlaku untuk setiap metrik. Daftar berikut menentukan statistik yang berlaku untuk metrik tersebut.

Invokasi

Frekuensi fungsi dimulai (dipanggil) dalam jangka waktu tertentu.

- Nama metrik: `FunctionInvocations`
- Statistik yang valid: Sum
- Unit: None

Kesalahan validasi

Jumlah kesalahan validasi yang dihasilkan oleh fungsi dalam jangka waktu tertentu. kesalahan validasi terjadi ketika fungsi berjalan berhasil tetapi mengembalikan data yang tidak valid (objek peristiwa tidak valid).

- Nama metrik: `FunctionValidationErrors`
- Statistik yang valid: Sum
- Unit: None

Kesalahan eksekusi

Jumlah kesalahan eksekusi yang terjadi dalam jangka waktu tertentu. Kesalahan eksekusi terjadi ketika fungsi gagal untuk menyelesaikan berhasil.

- Nama metrik: `FunctionExecutionErrors`
- Statistik yang valid: Sum
- Unit: None

Hitung pemanfaatan

Jumlah waktu (0-100) yang dibutuhkan fungsi untuk dijalankan sebagai persentase dari waktu maksimum yang diizinkan. Misalnya, nilai 35 berarti fungsi selesai pada 35% dari waktu maksimum yang diizinkan.

- Nama metrik: `FunctionComputeUtilization`
- Statistik yang valid: `Average`
- Unit: `Percent`

Pembatasan

Berapa kali fungsi itu dibatasi dalam periode waktu tertentu.

- Nama metrik: `FunctionThrottles`
- Statistik yang valid: `Sum`
- Unit: `None`

CloudFront dan logging fungsi tepi

Amazon CloudFront menyediakan berbagai jenis logging. Anda dapat mencatat permintaan penampil yang datang ke CloudFront distribusi Anda, atau Anda dapat mencatat aktivitas CloudFront layanan (aktivitas API) di AWS akun Anda. Anda juga bisa mendapatkan log dari CloudFront Fungsi dan fungsi Lambda @Edge Anda.

Mencatat permintaan

CloudFront menyediakan cara-cara berikut untuk mencatat permintaan yang datang ke distribusi Anda.

Akses log (log standar)

CloudFrontlog akses menyediakan catatan rinci tentang setiap permintaan yang dibuat untuk distribusi. Anda dapat menggunakan log untuk skenario, seperti audit keamanan dan akses.

CloudFrontlog akses dikirimkan ke tujuan pengiriman yang Anda tentukan.

Gunakan log akses saat Anda membutuhkan:

- Analisis dan pelaporan historis
- Audit keamanan dan persyaratan kepatuhan

- Retensi log jangka panjang yang hemat biaya

Untuk informasi selengkapnya, lihat [Akses log \(log standar\)](#).

Log akses waktu nyata

CloudFrontLog akses real-time dikirimkan dalam hitungan detik setelah menerima permintaan dan memberikan informasi tentang permintaan yang dibuat ke distribusi secara real time. Anda dapat memilih laju pengambilan sampel untuk log akses real-time Anda—yaitu, persentase permintaan yang ingin Anda terima catatan log akses real-time. Anda juga dapat memilih kolom khusus yang ingin Anda terima dalam catatan log. Log akses real-time ideal untuk pemantauan langsung untuk kinerja pengiriman konten.

CloudFront log akses real-time dikirimkan ke aliran data pilihan Anda di Amazon Kinesis Data Streams. CloudFront biaya untuk log akses real-time, selain biaya yang Anda keluarkan untuk menggunakan Kinesis Data Streams.

Gunakan log akses real-time saat Anda membutuhkan:

- Pemantauan dan peringatan waktu nyata
- Dasbor langsung dan wawasan operasional

Untuk informasi selengkapnya, lihat [Gunakan log akses waktu nyata](#).

Log koneksi

Log koneksi memberikan informasi rinci tentang koneksi antara server dan klien untuk distribusi mTLS diaktifkan. Log koneksi memberikan visibilitas ke informasi sertifikat klien, alasan kegagalan otentikasi mTLS dan apakah koneksi diizinkan atau ditolak.

Seperti log akses (log standar), log koneksi dikirimkan ke tujuan pengiriman yang Anda tentukan.

Note

Untuk mengaktifkan log koneksi, Anda harus terlebih dahulu [mengaktifkan mTL](#) untuk distribusi Anda.

Gunakan log koneksi saat Anda membutuhkan:

- Alasan koneksi yang berhasil atau tidak berhasil selama jabat tangan TLS
- Visibilitas ke informasi sertifikat klien

Untuk informasi selengkapnya, lihat [Observabilitas menggunakan log koneksi](#).

Fungsi tepi logging

Anda dapat menggunakan Amazon CloudWatch Logs untuk mendapatkan log untuk fungsi edge Anda, baik Lambda @Edge maupun CloudFront Functions. Anda dapat mengakses log menggunakan CloudWatch konsol atau CloudWatch Logs API. Untuk informasi selengkapnya, lihat [the section called “Log fungsi tepi”](#).

Aktivitas mencatat layanan

Anda dapat menggunakan AWS CloudTrail untuk mencatat aktivitas CloudFront layanan (aktivitas API) di AWS akun Anda. CloudTrail menyediakan catatan tindakan API yang diambil oleh pengguna, peran, atau AWS layanan di CloudFront. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan API yang dibuat CloudFront, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk informasi selengkapnya, lihat [Pencatatan panggilan CloudFront API Amazon menggunakan AWS CloudTrail](#).

Untuk informasi selengkapnya tentang pencatatan, lihat topik berikut:

Topik

- [Akses log \(log standar\)](#)
- [Gunakan log akses waktu nyata](#)
- [Log fungsi tepi](#)
- [Pencatatan panggilan CloudFront API Amazon menggunakan AWS CloudTrail](#)

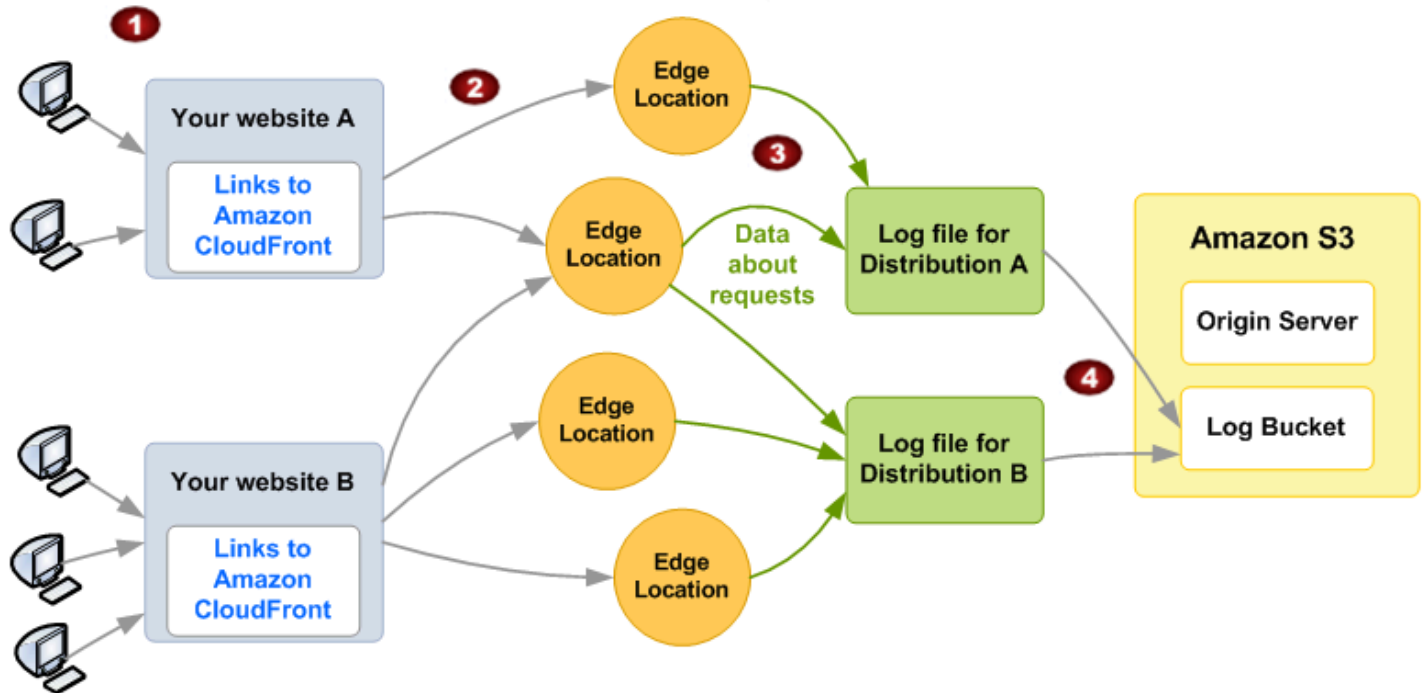
Akses log (log standar)

Anda dapat mengonfigurasi CloudFront untuk membuat file log yang berisi informasi terperinci tentang setiap permintaan pengguna (penampil) yang CloudFront menerima. Ini disebut log akses, juga dikenal sebagai log standar.

Setiap log berisi informasi seperti waktu permintaan diterima, waktu pemrosesan, jalur permintaan, dan respons server. Anda dapat menggunakan log akses ini untuk menganalisis waktu respons dan memecahkan masalah.

Diagram berikut menunjukkan bagaimana CloudFront log informasi tentang permintaan untuk objek Anda. Dalam contoh ini, distribusi dikonfigurasi untuk mengirim log akses ke bucket Amazon S3.

Users in different locations



1. Dalam contoh ini, Anda memiliki dua situs web, A dan B, dan dua CloudFront distribusi yang sesuai. Pengguna meminta objek Anda menggunakan URLs yang terkait dengan distribusi Anda.
2. CloudFront merutekan setiap permintaan ke lokasi tepi yang sesuai.
3. CloudFront menulis data tentang setiap permintaan ke file log khusus untuk distribusi itu. Dalam contoh ini, informasi tentang permintaan yang terkait dengan Distribusi A masuk ke file log untuk Distribusi A. Informasi tentang permintaan yang terkait dengan Distribusi B masuk ke file log untuk Distribusi B.
4. CloudFront menyimpan file log secara berkala untuk distribusi di bucket Amazon S3 yang Anda tentukan saat mengaktifkan logging. CloudFront kemudian mulai menyimpan informasi tentang permintaan berikutnya dalam file log baru untuk distribusi.

Jika pemirsa tidak mengakses konten Anda selama jam tertentu, Anda tidak akan menerima file log apa pun selama satu jam tersebut.

Note

Kami menyarankan Anda menggunakan log untuk memahami sifat permintaan untuk konten Anda, bukan sebagai akuntansi lengkap dari semua permintaan. CloudFront memberikan log akses dengan upaya terbaik. Entri log untuk permintaan tertentu mungkin dikirim dalam

waktu lama setelah permintaan diproses secara aktual dan, dalam kasus yang jarang, entri log mungkin tidak dikirimkan sama sekali. Ketika entri log dihilangkan dari log akses, jumlah entri dalam log akses tidak akan cocok dengan penggunaan yang muncul dalam laporan AWS penagihan dan penggunaan.

CloudFront mendukung dua versi logging standar. Pencatatan standar (lama) mendukung pengiriman log akses Anda ke Amazon S3 saja. Pencatatan standar (v2) mendukung tujuan pengiriman tambahan. Anda dapat mengonfigurasi keduanya atau salah satu opsi logging untuk distribusi Anda. Untuk informasi selengkapnya, lihat topik berikut:

Topik

- [Mengonfigurasi pencatatan standar \(v2\)](#)
- [Mengonfigurasi pencatatan standar \(lama\)](#)
- [Referensi logging standar](#)

Tip

CloudFront juga menawarkan log akses real-time, yang memberi Anda informasi tentang permintaan yang dibuat ke distribusi secara real time (log dikirimkan dalam hitungan detik setelah menerima permintaan). Anda dapat menggunakan log akses real-time untuk memantau, menganalisis, dan mengambil tindakan berdasarkan kinerja pengiriman konten. Lihat informasi yang lebih lengkap di [Gunakan log akses waktu nyata](#).

Mengonfigurasi pencatatan standar (v2)

Anda dapat mengaktifkan log akses (log standar) saat Anda membuat atau memperbarui distribusi. Pencatatan standar (v2) mencakup fitur-fitur berikut:

- Kirim log akses ke Amazon CloudWatch Logs, Amazon Data Firehose, dan Amazon Simple Storage Service (Amazon S3).
- Pilih bidang log yang Anda inginkan. Anda juga dapat memilih [subset bidang log akses real-time](#).
- Pilih format [file log keluaran](#) tambahan.

Jika Anda menggunakan Amazon S3, Anda memiliki fitur opsional berikut:

- Kirim log ke opt-in Wilayah AWS.
- Atur log Anda dengan partisi.
- Aktifkan nama file yang kompatibel dengan HIVE.

Untuk informasi selengkapnya, lihat [Mengirim log ke Amazon S3](#).

Untuk memulai dengan pencatatan standar, selesaikan langkah-langkah berikut:

1. Siapkan izin yang diperlukan untuk yang ditentukan Layanan AWS yang akan menerima log Anda.
2. Konfigurasi logging standar dari CloudFront konsol atau CloudWatch API.
3. Lihat log akses Anda.

Note

- Jika Anda mengaktifkan pencatatan standar (v2), ini tidak memengaruhi atau mengubah logging standar (lama). Anda dapat terus menggunakan logging standar (legacy) untuk distribusi Anda, selain menggunakan logging standar (v2). Untuk informasi selengkapnya, lihat [Mengonfigurasi pencatatan standar \(lama\)](#).
- Jika Anda sudah mengaktifkan pencatatan standar (lama) dan ingin mengaktifkan pencatatan standar (v2) ke Amazon S3, sebaiknya tentukan bucket Amazon S3 yang berbeda atau gunakan jalur terpisah di bucket yang sama (misalnya, gunakan awalan log atau partisi). Ini membantu Anda melacak file log mana yang terkait dengan distribusi mana dan mencegah file log saling menimpa.

Izin

CloudFront menggunakan log CloudWatch vendored untuk mengirimkan log akses. Untuk melakukannya, Anda memerlukan izin untuk yang ditentukan Layanan AWS sehingga Anda dapat mengaktifkan pengiriman logging.

Untuk melihat izin yang diperlukan untuk setiap tujuan pencatatan, pilih dari topik berikut di Panduan Pengguna Amazon CloudWatch Logs.

- [CloudWatch Log](#)
- [Firehose](#)

- [Amazon S3](#)

Setelah menyiapkan izin ke tujuan pencatatan, Anda dapat mengaktifkan pencatatan standar untuk distribusi Anda.

 Note

CloudFront mendukung pengiriman log akses ke yang berbeda Akun AWS (lintas akun). Untuk mengaktifkan pengiriman lintas akun, kedua akun (akun Anda dan akun penerima) harus memiliki izin yang diperlukan. Untuk informasi selengkapnya, lihat [Aktifkan pencatatan standar untuk pengiriman lintas akun](#) bagian atau [contoh pengiriman lintas akun](#) di Panduan Pengguna CloudWatch Log Amazon.

Aktifkan logging standar

Untuk mengaktifkan logging standar, Anda dapat menggunakan CloudFront konsol atau CloudWatch API.

Daftar Isi

- [Aktifkan logging standar \(CloudFront konsol\)](#)
- [Aktifkan logging standar \(CloudWatch API\)](#)

Aktifkan logging standar (CloudFront konsol)

Untuk mengaktifkan pencatatan standar untuk CloudFront distribusi (konsol)

1. Gunakan CloudFront konsol untuk [memperbarui distribusi yang ada](#).
2. Pilih tab Logging.
3. Pilih Tambah, lalu pilih layanan untuk menerima log Anda:
 - CloudWatch Log
 - Firehouse
 - Amazon S3
4. Untuk Tujuan, pilih sumber daya untuk layanan Anda. Jika Anda belum membuat sumber daya, Anda dapat memilih Buat atau melihat dokumentasi berikut.

- Untuk CloudWatch Log, masukkan [nama grup Log](#).
- Untuk Firehose, masuk ke aliran pengiriman [Firehose](#).
- Untuk Amazon S3, masukkan nama [Bucket](#).

 Tip

Untuk menentukan awalan, masukkan awalan setelah nama bucket, seperti. `amzn-s3-demo-bucket.s3.amazonaws.com/MyLogPrefix` Jika Anda tidak menentukan awalan, secara otomatis CloudFront akan menambahkan satu untuk Anda. Untuk informasi selengkapnya, lihat [Mengirim log ke Amazon S3](#).

5. Untuk Pengaturan tambahan - opsional, Anda dapat menentukan opsi berikut:
- a. Untuk pemilihan Bidang, pilih nama bidang log yang ingin Anda kirimkan ke tujuan Anda. Anda dapat memilih [bidang log akses](#) dan subset [bidang log akses real-time](#).
 - b. (Hanya Amazon S3) Untuk Partisi, tentukan jalur untuk mempartisi data file log Anda.
 - c. (Hanya Amazon S3) Untuk format file yang kompatibel dengan HIVE, Anda dapat memilih kotak centang untuk menggunakan jalur S3 yang kompatibel dengan HIVE. Ini membantu menyederhanakan pemuatan data baru ke alat yang kompatibel dengan HIVE Anda.
 - d. Untuk format Output, tentukan format pilihan Anda.

 Note

Jika Anda memilih Parquet, opsi ini CloudWatch dikenakan biaya untuk mengonversi log akses Anda ke Apache Parquet. Untuk informasi selengkapnya, lihat [bagian Vended Logs untuk CloudWatch harga](#).

- e. Untuk pembatas bidang, tentukan cara memisahkan bidang log.
6. Selesaikan langkah-langkah untuk memperbarui atau membuat distribusi Anda.
 7. Untuk menambahkan tujuan lain, ulangi langkah 3—6.
 8. Dari halaman Log, verifikasi bahwa status log standar Diaktifkan di sebelah distribusi.
 9. (Opsional) Untuk mengaktifkan pencatatan cookie, pilih Kelola, Pengaturan, dan aktifkan Pencatatan cookie, lalu pilih Simpan perubahan.

i Tip

Cookie logging adalah pengaturan global yang berlaku untuk semua logging standar untuk distribusi Anda. Anda tidak dapat mengganti pengaturan ini untuk tujuan pengiriman terpisah.

Untuk informasi selengkapnya tentang pengiriman logging standar dan bidang log, lihat [Referensi logging standar](#).

Aktifkan logging standar (CloudWatch API)

Anda juga dapat menggunakan CloudWatch API untuk mengaktifkan logging standar untuk distribusi Anda.

i Catatan

- Saat memanggil CloudWatch API untuk mengaktifkan pencatatan standar, Anda harus menentukan Wilayah AS Timur (Virginia Utara) (us-east-1), meskipun Anda ingin mengaktifkan pengiriman lintas Wilayah ke tujuan lain. Misalnya, jika Anda ingin mengirim log akses ke bucket S3 di Wilayah Eropa (Irlandia) (eu-west-1), gunakan CloudWatch API di us-east-1 Wilayah.
- Ada opsi tambahan untuk memasukkan cookie dalam pencatatan standar. Di CloudFront API, ini adalah IncludeCookies parameternya. Jika Anda mengonfigurasi pencatatan akses dengan menggunakan CloudWatch API dan Anda menentukan bahwa Anda ingin menyertakan cookie, Anda harus menggunakan CloudFront konsol atau CloudFront API untuk memperbarui distribusi Anda untuk menyertakan cookie. Jika tidak, tidak CloudFront dapat mengirim cookie ke tujuan log Anda. Untuk informasi selengkapnya, lihat [Pencatatan cookie](#).

Untuk mengaktifkan pencatatan standar untuk distribusi (CloudWatch API)

1. Setelah Anda membuat distribusi, dapatkan Amazon Resource Name (ARN).

Anda dapat menemukan ARN dari halaman Distribusi di CloudFront konsol atau Anda dapat menggunakan operasi [GetDistribution](#) API. Sebuah distribusi ARN mengikuti format: `arn:aws:cloudfront::123456789012:distribution/d111111abcdef8`

2. Selanjutnya, gunakan operasi CloudWatch [PutDeliverySource](#) API untuk membuat sumber pengiriman untuk distribusi.
 - a. Masukkan nama untuk sumber pengiriman.
 - b. `resourceArn` Lewati distribusi.
 - c. Untuk `logType`, tentukan `ACCESS_LOGS` sebagai jenis log yang dikumpulkan.
 - d. Example Contoh AWS CLI `put-delivery-source` perintah

Berikut ini adalah contoh konfigurasi sumber pengiriman untuk distribusi.

```
aws logs put-delivery-source --name S3-delivery --resource-arn
arn:aws:cloudfront::123456789012:distribution/d111111abcdef8 --log-type
ACCESS_LOGS
```

Keluaran

```
{
  "deliverySource": {
    "name": "S3-delivery",
    "arn": "arn:aws:logs:us-east-1:123456789012:delivery-source:S3-delivery",
    "resourceArns": [
      "arn:aws:cloudfront::123456789012:distribution/d111111abcdef8"
    ],
    "service": "cloudfront",
    "logType": "ACCESS_LOGS"
  }
}
```

3. Gunakan operasi [PutDeliveryDestination](#) API untuk mengonfigurasi tempat menyimpan log Anda.
 - a. Untuk `destinationResourceArn`, tentukan ARN tujuan. Ini bisa berupa grup CloudWatch log Log, aliran pengiriman Firehose, atau bucket Amazon S3.
 - b. Untuk `outputFormat`, tentukan format output untuk log Anda.

c. Example Contoh AWS CLI put-delivery-destination perintah

Berikut ini adalah contoh mengonfigurasi tujuan pengiriman ke bucket Amazon S3.

```
aws logs put-delivery-destination --name S3-destination --delivery-destination-configuration destinationResourceArn=arn:aws:s3:::amzn-s3-demo-bucket
```

Keluaran

```
{
  "name": "S3-destination",
  "arn": "arn:aws:logs:us-east-1:123456789012:delivery-destination:S3-destination",
  "deliveryDestinationType": "S3",
  "deliveryDestinationConfiguration": {
    "destinationResourceArn": "arn:aws:s3:::amzn-s3-demo-bucket"
  }
}
```

 Note

Jika mengirimkan log lintas akun, Anda harus menggunakan operasi [PutDeliveryDestinationPolicy](#) API untuk menetapkan kebijakan AWS Identity and Access Management (IAM) ke akun tujuan. Kebijakan IAM memungkinkan pengiriman dari satu akun ke akun lain.

4. Gunakan operasi [CreateDelivery](#) API untuk menautkan sumber pengiriman ke tujuan yang Anda buat di langkah sebelumnya. Operasi API ini mengaitkan sumber pengiriman dengan tujuan akhir.
 - a. Untuk `deliverySourceName`, tentukan nama sumbernya.
 - b. Untuk `deliveryDestinationArn`, tentukan ARN untuk tujuan pengiriman.
 - c. Untuk `fieldDelimiter`, tentukan string untuk memisahkan setiap bidang log.
 - d. Untuk `recordFields`, tentukan bidang log yang Anda inginkan.
 - e. Jika Anda menggunakan S3, tentukan apakah akan menggunakan `enableHiveCompatiblePath` dan `suffixPath`.

Example Contoh perintah AWS CLI create-delivery

Berikut ini adalah contoh pembuatan pengiriman.

```
aws logs create-delivery --delivery-source-name cf-delivery --delivery-destination-arn arn:aws:logs:us-east-1:123456789012:delivery-destination:S3-destination
```

Keluaran

```
{
  "id": "abcNegnBoTR123",
  "arn": "arn:aws:logs:us-east-1:123456789012:delivery:abcNegnBoTR123",
  "deliverySourceName": "cf-delivery",
  "deliveryDestinationArn": "arn:aws:logs:us-east-1:123456789012:delivery-destination:S3-destination",
  "deliveryDestinationType": "S3",
  "recordFields": [
    "date",
    "time",
    "x-edge-location",
    "sc-bytes",
    "c-ip",
    "cs-method",
    "cs(Host)",
    "cs-uri-stem",
    "sc-status",
    "cs(Referer)",
    "cs(User-Agent)",
    "cs-uri-query",
    "cs(Cookie)",
    "x-edge-result-type",
    "x-edge-request-id",
    "x-host-header",
    "cs-protocol",
    "cs-bytes",
    "time-taken",
    "x-forwarded-for",
    "ssl-protocol",
    "ssl-cipher",
    "x-edge-response-result-type",
    "cs-protocol-version",
```

```
    "fle-status",
    "fle-encrypted-fields",
    "c-port",
    "time-to-first-byte",
    "x-edge-detailed-result-type",
    "sc-content-type",
    "sc-content-len",
    "sc-range-start",
    "sc-range-end",
    "c-country",
    "cache-behavior-path-pattern"
  ],
  "fieldDelimiter": ""
}
```

5. Dari CloudFront konsol, pada halaman Log, verifikasi bahwa status log standar Diaktifkan di sebelah distribusi.

Untuk informasi selengkapnya tentang pengiriman logging standar dan bidang log, lihat [Referensi logging standar](#).

Note

Untuk mengaktifkan logging standar (v2) CloudFront dengan menggunakan AWS CloudFormation, Anda dapat menggunakan properti CloudWatch Log berikut:

- [Pengiriman](#)
- [DeliveryDestination](#)
- [DeliverySource](#)

ResourceArnIni adalah CloudFront distribusi dan LogType harus ACCESS_LOGS sebagai jenis log yang didukung.

Aktifkan pencatatan standar untuk pengiriman lintas akun

Jika Anda mengaktifkan pencatatan standar untuk Anda Akun AWS dan Anda ingin mengirimkan log akses Anda ke akun lain, pastikan Anda mengonfigurasi akun sumber dan akun tujuan dengan benar. Akun sumber dengan CloudFront distribusi mengirimkan log aksesnya ke akun tujuan.

Dalam contoh prosedur ini, akun sumber(111111111111) mengirimkan log aksesnya ke bucket Amazon S3 di akun tujuan (222222222222). Untuk mengirim log akses ke bucket Amazon S3 di akun tujuan, gunakan file. AWS CLI

Konfigurasi akun tujuan

Untuk akun tujuan, selesaikan prosedur berikut.

Untuk mengkonfigurasi akun tujuan

1. Untuk membuat tujuan pengiriman log, Anda dapat memasukkan AWS CLI perintah berikut. Contoh ini menggunakan *MyLogPrefix* string untuk membuat awalan untuk log akses Anda.

```
aws logs put-delivery-destination --name cloudfront-delivery-destination --  
delivery-destination-configuration "destinationResourceArn=arn:aws:s3:::amzn-s3-  
demo-bucket-cloudfront-logs/MyLogPrefix"
```

Keluaran

```
{  
  "deliveryDestination": {  
    "name": "cloudfront-delivery-destination",  
    "arn": "arn:aws:logs:us-east-1:222222222222:delivery-  
destination:cloudfront-delivery-destination",  
    "deliveryDestinationType": "S3",  
    "deliveryDestinationConfiguration": {"destinationResourceArn":  
"arn:aws:s3:::amzn-s3-demo-bucket-cloudfront-logs/MyLogPrefix"  
  }  
}
```

Note

Jika Anda menentukan bucket S3 tanpa awalan, secara otomatis CloudFront akan menambahkan AWSLogs/<*account-ID*>/CloudFront sebagai awalan yang muncul di tujuan pengiriman suffixPath S3. Untuk informasi lebih lanjut, lihat [S3 DeliveryConfiguration](#).

2. Tambahkan kebijakan sumber daya untuk tujuan pengiriman log agar akun sumber dapat membuat pengiriman log.

Dalam kebijakan berikut, ganti **111111111111** dengan ID akun sumber dan tentukan ARN tujuan pengiriman dari output pada langkah 1.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCreateDelivery",
      "Effect": "Allow",
      "Principal": {"AWS": "111111111111"},
      "Action": ["logs:CreateDelivery"],
      "Resource": "arn:aws:logs:us-east-1:222222222222:delivery-destination:cloudfront-delivery-destination"
    }
  ]
}
```

3. Simpan file, seperti `deliverypolicy.json`.
4. Untuk melampirkan kebijakan sebelumnya ke tujuan pengiriman, masukkan AWS CLI perintah berikut.

```
aws logs put-delivery-destination-policy --delivery-destination-name cloudfront-delivery-destination --delivery-destination-policy file://deliverypolicy.json
```

5. Tambahkan pernyataan di bawah ini ke kebijakan bucket Amazon S3 tujuan, menggantikan ARN sumber daya dan ID akun sumber. Kebijakan ini memungkinkan kepala `delivery.logs.amazonaws.com` layanan untuk melakukan `s3:PutObject` tindakan.

```
{
  "Sid": "AWSLogsDeliveryWrite",
  "Effect": "Allow",
  "Principal": {"Service": "delivery.logs.amazonaws.com"},
  "Action": "s3:PutObject",
  "Resource": "arn:aws:s3:::amzn-s3-demo-bucket-cloudfront-logs/*",
  "Condition": {
    "StringEquals": {
      "s3:x-amz-acl": "bucket-owner-full-control",
      "aws:SourceAccount": "111111111111"
    }
  }
}
```

```

    },
    "ArnLike": {"aws:SourceArn": "arn:aws:logs:us-east-1:111111111111:delivery-
source:*"}
  }
}

```

6. Jika Anda menggunakan AWS KMS bucket, tambahkan pernyataan berikut ke kebijakan kunci KMS untuk memberikan izin kepada prinsipal `delivery.logs.amazonaws.com` layanan.

```

{
  "Sid": "Allow Logs Delivery to use the key",
  "Effect": "Allow",
  "Principal": {"Service": "delivery.logs.amazonaws.com"},
  "Action": [
    "kms:Encrypt",
    "kms:Decrypt",
    "kms:ReEncrypt*",
    "kms:GenerateDataKey*",
    "kms:DescribeKey"
  ],
  "Resource": "*",
  "Condition": {
    "StringEquals": {"aws:SourceAccount": "111111111111"},
    "ArnLike": {"aws:SourceArn": "arn:aws:logs:us-east-1:111111111111:delivery-
source:*"}
  }
}

```

Konfigurasi akun sumber

Setelah Anda mengonfigurasi akun tujuan, ikuti prosedur ini untuk membuat sumber pengiriman dan mengaktifkan pencatatan untuk distribusi di akun sumber.

Untuk mengkonfigurasi akun sumber

1. Buat sumber pengiriman untuk pencatatan CloudFront standar sehingga Anda dapat mengirim file log ke CloudWatch Log.

Anda dapat memasukkan AWS CLI perintah berikut, mengganti nama dan ARN distribusi Anda.

```
aws logs put-delivery-source --name s3-cf-delivery --resource-arn
arn:aws:cloudfront::111111111111:distribution/E1TR1RHHV123ABC --log-type
ACCESS_LOGS
```

Keluaran

```
{
  "deliverySource": {
    "name": "s3-cf-delivery",
    "arn": "arn:aws:logs:us-east-1:111111111111:delivery-source:s3-cf-delivery",
    "resourceArns":
    ["arn:aws:cloudfront::111111111111:distribution/E1TR1RHHV123ABC"],
    "service": "cloudfront",
    "logType": "ACCESS_LOGS"
  }
}
```

2. Buat pengiriman untuk memetakan sumber pengiriman log akun sumber dan tujuan pengiriman log akun tujuan.

Dalam AWS CLI perintah berikut, tentukan ARN tujuan pengiriman dari output di [Langkah 1: Konfigurasi akun tujuan](#).

```
aws logs create-delivery --delivery-source-name s3-cf-delivery --
delivery-destination-arn arn:aws:logs:us-east-1:222222222222:delivery-
destination:cloudfront-delivery-destination
```

Keluaran

```
{
  "delivery": {
    "id": "OPmOpLahVzhx1234",
    "arn": "arn:aws:logs:us-east-1:111111111111:delivery:OPmOpLahVzhx1234",
    "deliverySourceName": "s3-cf-delivery",
    "deliveryDestinationArn": "arn:aws:logs:us-east-1:222222222222:delivery-
destination:cloudfront-delivery-destination",
    "deliveryDestinationType": "S3",
    "recordFields": [
      "date",
      "time",

```



```
        "x-edge-location",
        "sc-bytes",
        "c-ip",
        "cs-method",
        "cs(Host)",
        "cs-uri-stem",
        "sc-status",
        "cs(Referer)",
        "cs(User-Agent)",
        "cs-uri-query",
        "cs(Cookie)",
        "x-edge-result-type",
        "x-edge-request-id",
        "x-host-header",
        "cs-protocol",
        "cs-bytes",
        "time-taken",
        "x-forwarded-for",
        "ssl-protocol",
        "ssl-cipher",
        "x-edge-response-result-type",
        "cs-protocol-version",
        "fle-status",
        "fle-encrypted-fields",
        "c-port",
        "time-to-first-byte",
        "x-edge-detailed-result-type",
        "sc-content-type",
        "sc-content-len",
        "sc-range-start",
        "sc-range-end",
        "c-country",
        "cache-behavior-path-pattern"
    ],
    "fieldDelimiter": "\t"
}
```

3. Verifikasi pengiriman lintas akun Anda berhasil.

- a. Dari **source** akun, masuk ke CloudFront konsol dan pilih distribusi Anda. Pada tab Logging, di bawah Type, Anda akan melihat entri yang dibuat untuk pengiriman log lintas akun S3.

- b. Dari *destination* akun, masuk ke konsol Amazon S3 dan pilih bucket Amazon S3 Anda. Anda akan melihat awalan *MyLogPrefix* di nama bucket dan log akses apa pun yang dikirimkan ke folder itu.

Format file keluaran

Bergantung pada tujuan pengiriman yang Anda pilih, Anda dapat menentukan salah satu format berikut untuk file log:

- JSON
- Biasa
- w3c
- Mentah
- Parquet (hanya Amazon S3)

Note

Anda hanya dapat mengatur format output saat pertama kali membuat tujuan pengiriman. Ini tidak dapat diperbarui nanti. Untuk mengubah format output, hapus pengiriman dan buat yang baru.

Untuk informasi selengkapnya, lihat [PutDeliveryDestination](#) di Referensi API Amazon CloudWatch Logs.

Edit pengaturan logging standar

Anda dapat mengaktifkan atau menonaktifkan logging dan memperbarui pengaturan log lainnya dengan menggunakan [CloudFront konsol](#) atau CloudWatch API. Perubahan Anda pada pengaturan log mulai berlaku dalam 12 jam.

Untuk informasi selengkapnya, lihat topik berikut:

- Untuk memperbarui distribusi menggunakan CloudFront konsol, lihat [Perbarui distribusi](#).
- Untuk memperbarui distribusi menggunakan CloudFront API, lihat [UpdateDistribution](#) di Referensi Amazon CloudFront API.

- Untuk informasi selengkapnya tentang operasi API CloudWatch Log, lihat [Referensi API Amazon CloudWatch Logs](#).

Akses bidang log

Anda dapat memilih bidang log yang sama yang didukung oleh pencatatan standar (lama). Untuk informasi selengkapnya, lihat [bidang file log](#).

Selain itu, Anda dapat memilih [bidang log akses real-time](#) berikut.

1. **timestamp(ms)**— Timestamp dalam milidetik.
2. **origin-fbl**— Jumlah detik latensi byte pertama antara CloudFront dan asal Anda.
3. **origin-lbl**— Jumlah detik latensi byte terakhir antara CloudFront dan asal Anda.
4. **asn**— Nomor sistem otonom (ASN) dari pemirsa.
5. **c-country**— Kode negara yang mewakili lokasi geografis pemirsa, sebagaimana ditentukan oleh alamat IP pemirsa. Untuk daftar kode negara, lihat [ISO 3166-1 alpha-2](#).
6. **cache-behavior-path-pattern**— Pola jalur yang mengidentifikasi perilaku cache yang cocok dengan permintaan penampil.

Kirim log ke CloudWatch Log

Untuk mengirim CloudWatch log ke Log, buat atau gunakan grup CloudWatch log Log yang ada. Untuk informasi selengkapnya tentang mengonfigurasi grup CloudWatch log Log, lihat [Bekerja dengan Grup Log dan Aliran Log](#).

Setelah membuat grup log, Anda harus memiliki izin yang diperlukan untuk mengizinkan pencatatan standar. Untuk informasi selengkapnya tentang izin yang diperlukan, lihat [Log yang dikirim ke CloudWatch Log](#) di Panduan Pengguna CloudWatch Log Amazon.

Catatan

- Saat Anda menentukan nama grup CloudWatch log Log, hanya gunakan pola regex. `[\w-]` Untuk informasi selengkapnya, lihat operasi [PutDeliveryDestination](#) API di Referensi API Amazon CloudWatch Logs.
- Pastikan kebijakan sumber daya grup log Anda tidak melebihi batas ukuran. Lihat bagian [Pertimbangan batas ukuran kebijakan sumber daya grup log](#) di topik CloudWatch Log.

Contoh log akses dikirim ke CloudWatch Log

```
{
  "date": "2024-11-14",
  "time": "21:34:06",
  "x-edge-location": "S0F50-P2",
  "asn": "16509",
  "timestamp(ms)": "1731620046814",
  "origin-fbl": "0.251",
  "origin-lbl": "0.251",
  "x-host-header": "d111111abcdef8.cloudfront.net",
  "cs(Cookie)": "examplecookie=value"
}
```

Kirim log ke Firehose

Untuk mengirim log ke Firehose, buat atau gunakan aliran pengiriman Firehose yang ada. Kemudian, tentukan aliran pengiriman Firehose sebagai tujuan pengiriman log. Anda harus menentukan aliran pengiriman Firehose di Wilayah us-east-1 AS Timur (Virginia Utara).

Untuk informasi tentang membuat aliran pengiriman, lihat [Membuat aliran pengiriman Amazon Data Firehose](#).

Setelah membuat aliran pengiriman, Anda harus memiliki izin yang diperlukan untuk mengizinkan pencatatan standar. Untuk informasi selengkapnya, lihat [Log yang dikirim ke Firehose](#) di Panduan Pengguna Amazon CloudWatch Logs.

Note

Saat Anda menentukan nama aliran Firehose, hanya gunakan pola regex. `[\w-]` Untuk informasi selengkapnya, lihat operasi [PutDeliveryDestination](#) API di Referensi API Amazon CloudWatch Logs.

Contoh log akses dikirim ke Firehose

```
{"date":"2024-11-15","time":"19:45:51","x-edge-location":"S0F50-P2","asn":"16509","timestamp(ms)": "1731699951183","origin-fbl":"0.254","origin-lbl":"0.254","x-host-header":"d111111abcdef8.cloudfront.net","cs(Cookie)": "examplecookie=value"}
```

```
{"date":"2024-11-15","time":"19:45:52","x-edge-location":"S0F50-P2","asn":"16509","timestamp(ms)":"1731699952950","origin-fbl":"0.125","origin-lbl":"0.125","x-host-header":"d111111abcdef8.cloudfront.net","cs(Cookie)":"examplecookie=value"}
```

Mengirim log ke Amazon S3

Untuk mengirim log akses ke Amazon S3, buat atau gunakan bucket S3 yang sudah ada. Saat Anda mengaktifkan login CloudFront, tentukan nama bucket. Untuk informasi tentang membuat bucket, lihat [Membuat bucket](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Setelah membuat bucket, Anda harus memiliki izin yang diperlukan untuk mengizinkan pencatatan standar. Untuk informasi selengkapnya, lihat [Log yang dikirim ke Amazon S3](#) di Panduan Pengguna Amazon CloudWatch Logs.

- Setelah mengaktifkan pencatatan, AWS secara otomatis menambahkan kebijakan bucket yang diperlukan untuk Anda.
- [Anda juga dapat menggunakan bucket S3 dalam opt-in. Wilayah AWS](#)

Note

Jika Anda sudah mengaktifkan pencatatan standar (lama) dan ingin mengaktifkan pencatatan standar (v2) ke Amazon S3, sebaiknya tentukan bucket Amazon S3 yang berbeda atau gunakan jalur terpisah di bucket yang sama (misalnya, gunakan awalan log atau partisi). Ini membantu Anda melacak file log mana yang terkait dengan distribusi mana dan mencegah file log saling menimpa.

Topik

- [Tentukan ember S3](#)
- [Partitioning](#)
- [Format nama file yang kompatibel dengan HIVE](#)
- [Contoh jalur untuk mengakses log](#)
- [Contoh log akses dikirim ke Amazon S3](#)

Tentukan ember S3

Saat Anda menentukan bucket S3 sebagai tujuan pengiriman, perhatikan hal berikut.

Nama bucket S3 hanya dapat menggunakan pola regex. `[\w-]` Untuk informasi selengkapnya, lihat operasi [PutDeliveryDestination](#) API di Referensi API Amazon CloudWatch Logs.

Jika Anda menetapkan awalan untuk bucket S3 Anda, log Anda akan muncul di bawah jalur itu. Jika Anda tidak menentukan awalan, secara otomatis CloudFront akan menambahkan `AWSLogs/{account-id}/CloudFront` awalan untuk Anda.

Untuk informasi selengkapnya, lihat [Contoh jalur untuk mengakses log](#).

Partitioning

Anda dapat menggunakan partisi untuk mengatur log akses Anda saat CloudFront mengirimkannya ke bucket S3 Anda. Ini membantu Anda mengatur dan menemukan log akses Anda berdasarkan jalur yang Anda inginkan.

Anda dapat menggunakan variabel berikut untuk membuat jalur folder.

- `{DistributionId}` atau `{distributionid}`
- `{yyyy}`
- `{MM}`
- `{dd}`
- `{HH}`
- `{accountid}`

Anda dapat menggunakan sejumlah variabel dan menentukan nama folder di jalur Anda. CloudFront kemudian gunakan jalur ini untuk membuat struktur folder untuk Anda di ember S3.

Contoh

- `my_distribution_log_data/{DistributionId}/logs`
- `/cloudfront/{DistributionId}/my_distribution_log_data/{yyyy}/{MM}/{dd}/{HH}/logs`

 Note

Anda dapat menggunakan salah satu variabel untuk ID distribusi di jalur sufiks. Namun, jika Anda mengirim log akses ke AWS Glue, Anda harus menggunakan `{distributionid}` variabel karena AWS Glue mengharapkan nama partisi berada dalam huruf kecil. Perbarui konfigurasi log Anda yang ada CloudFront untuk menggantikan `{DistributionId}` dengan `{distributionid}`.

Format nama file yang kompatibel dengan HIVE

Anda dapat menggunakan opsi ini sehingga objek S3 yang berisi log akses terkirim menggunakan struktur awalan yang memungkinkan integrasi dengan Apache Hive. Untuk informasi selengkapnya, lihat operasi API [CreateDelivery](#).

Example Contoh

```
/cloudfront/DistributionId={DistributionId}/my_distribution_log_data/year={yyyy}/  
month={MM}/day={dd}/hour={HH}/logs
```

Untuk informasi selengkapnya tentang partisi dan opsi yang kompatibel dengan HIVE, lihat [elemen S3 DeliveryConfiguration](#) di Referensi API Amazon Logs. CloudWatch

Contoh jalur untuk mengakses log

Saat menentukan bucket S3 sebagai tujuan, Anda dapat menggunakan opsi berikut untuk membuat jalur ke log akses:

- Bucket Amazon S3, dengan atau tanpa awalan
- Partisi, dengan menggunakan variabel yang CloudFront disediakan atau memasukkan variabel Anda sendiri
- Mengaktifkan opsi yang kompatibel dengan HIVE

Tabel berikut menunjukkan bagaimana log akses Anda muncul di bucket, tergantung pada opsi yang Anda pilih.

Bucket Amazon S3 dengan awalan

Nama ember Amazon S3	Partisi yang Anda tentukan di jalur sufiks	Jalur sufiks yang diperbarui	Hive-kompatibel diaktifkan?	Log akses dikirim ke
amzn-s3-demo-bucket/MyLogPrefix	Tidak ada	Tidak ada	Tidak	amzn-s3-demo-bucket/MyLogPrefix/
amzn-s3-demo-bucket/MyLogPrefix	myFolderA/	myFolderA/	Tidak	amzn-s3-demo-bucket/MyLogPrefix/myFolderA/
amzn-s3-demo-bucket/MyLogPrefix	myFolderA/{yyyy}	myFolderA/{yyyy}	Ya	amzn-s3-demo-bucket/MyLogPrefix/myFolderA/year=2025

Bucket Amazon S3 tanpa awalan

Nama ember Amazon S3	Partisi yang Anda tentukan di jalur sufiks	Jalur sufiks yang diperbarui	Hive-kompatibel diaktifkan?	Log akses dikirim ke
amzn-s3-demo-bucket	Tidak ada	AWSLogs/{account-id}/CloudFront/	Tidak	amzn-s3-demo-bucket/AWSLogs / <i><your-account-ID></i> / CloudFront/

Nama ember Amazon S3	Partisi yang Anda tentukan di jalur sufiks	Jalur sufiks yang diperbarui	Hive-kompatibel diaktifkan?	Log akses dikirim ke
amzn-s3-d emo-bucket	myFolderA/	AWSLogs/{ account-id}/CloudFront/myFolderA/	Tidak	amzn-s3-d emo-bucket/AWSLogs / <i><your-account-ID></i> / CloudFront/ myFolderA/
amzn-s3-d emo-bucket	myFolderA/	AWSLogs/{ account-id}/CloudFront/myFolderA/	Ya	amzn-s3-d emo-bucket/AWSLogs /aws-account-id= <i><your-account-ID></i> / CloudFront/ myFolderA/
amzn-s3-d emo-bucket	myFolderA/{yyyy}	AWSLogs/{ account-id}/CloudFront/myFolderA/{yy yy}	Ya	amzn-s3-d emo-bucket/AWSLogs /aws-account-id= <i><your-account-ID></i> / CloudFront/ myFolderA/ year=2025

Akun AWS ID sebagai partisi

Nama ember Amazon S3	Partisi yang Anda tentukan di jalur sufiks	Jalur sufiks yang diperbarui	Hive-kompatibel diaktifkan?	Log akses dikirim ke
amzn-s3-demo-bucket	Tidak ada	AWSLogs/{account-id}/CloudFront/	Ya	amzn-s3-demo-bucket/AWSLogs/aws-account-id=<your-account-ID> /CloudFront/
amzn-s3-demo-bucket	myFolderA/{accountid}	AWSLogs/{account-id}/CloudFront/myFolderA/{accountid}	Ya	amzn-s3-demo-bucket/AWSLogs/aws-account-id=<your-account-ID>/CloudFront/myFolderA/accountid= <your-account-ID>

 Catatan

- {account-id} Variabel dicadangkan untuk CloudFront. CloudFront secara otomatis menambahkan variabel ini ke jalur sufiks jika Anda menentukan bucket Amazon S3 tanpa awalan. Jika log Anda kompatibel dengan HIVE, variabel ini muncul sebagai. aws-account-id

- Anda dapat menggunakan {accountid} variabel sehingga CloudFront menambahkan ID akun Anda ke jalur akhiran. Jika log Anda kompatibel dengan HIVE, variabel ini muncul sebagai. accountid
- Untuk informasi selengkapnya tentang jalur sufiks, lihat [DeliveryConfigurationS3](#).

Contoh log akses dikirim ke Amazon S3

```
#Fields: date time x-edge-location asn timestamp(ms) x-host-header cs(Cookie)
2024-11-14    22:30:25    S0F50-P2    16509    1731623425421
d1111111abcdef8.cloudfront.net    examplecookie=value2
```

Nonaktifkan logging standar

Anda dapat menonaktifkan pencatatan standar untuk distribusi Anda jika Anda tidak lagi membutuhkannya.

Untuk menonaktifkan logging standar

1. Masuk ke CloudFront konsol.
2. Pilih Distribusi dan kemudian pilih ID distribusi Anda.
3. Pilih Logging dan kemudian di bawah Akses log tujuan, pilih tujuan.
4. Pilih Kelola dan kemudian pilih Hapus.
5. Ulangi langkah sebelumnya jika Anda memiliki lebih dari satu logging standar.

Note

Saat Anda menghapus logging standar dari CloudFront konsol, tindakan ini hanya menghapus pengiriman dan tujuan pengiriman. Itu tidak menghapus sumber pengiriman dari Anda Akun AWS. Untuk menghapus sumber pengiriman, tentukan nama sumber pengiriman dalam `aws logs delete-delivery-source --name DeliverySourceName` perintah. Untuk informasi selengkapnya, lihat [DeleteDeliverySource](#) di Referensi API Amazon CloudWatch Logs.

Pemecahan Masalah

Gunakan informasi berikut untuk memperbaiki masalah umum saat Anda bekerja dengan logging CloudFront standar (v2).

Sumber pengiriman sudah ada

Saat mengaktifkan pencatatan standar untuk distribusi, Anda membuat sumber pengiriman. Anda kemudian menggunakan sumber pengiriman tersebut untuk membuat pengiriman ke jenis tujuan yang Anda inginkan: CloudWatch Log, Firehose, Amazon S3. Saat ini, Anda hanya dapat memiliki satu sumber pengiriman per distribusi. Jika Anda mencoba membuat sumber pengiriman lain untuk distribusi yang sama, pesan galat berikut akan muncul.

This ResourceId has already been used in another Delivery Source in this account

Untuk membuat sumber pengiriman lain, hapus yang sudah ada terlebih dahulu. Untuk informasi selengkapnya, lihat [DeleteDeliverySource](#) di Referensi API Amazon CloudWatch Logs.

Saya mengubah jalur sufiks dan ember Amazon S3 tidak dapat menerima log saya

Jika Anda mengaktifkan pencatatan standar (v2) dan menentukan ARN bucket tanpa awalan, CloudFront akan menambahkan default berikut ke jalur sufiks: `AWSLogs/{account-id}/CloudFront` Jika Anda menggunakan CloudFront konsol atau operasi [UpdateDeliveryConfiguration](#) API untuk menentukan jalur sufiks yang berbeda, Anda harus memperbarui kebijakan bucket Amazon S3 untuk menggunakan jalur yang sama.

Example Contoh: Memperbarui jalur sufiks

1. Jalur sufiks default Anda adalah `AWSLogs/{account-id}/CloudFront` dan Anda menggantinya dengan `myFolderA`.
2. Karena jalur akhiran baru Anda berbeda dari jalur yang ditentukan dalam kebijakan bucket Amazon S3, log akses Anda tidak akan dikirimkan.
3. Anda dapat melakukan salah satu langkah berikut:
 - Perbarui izin bucket Amazon S3 dari `keamzn-s3-demo-bucket/AWSLogs/<your-account-ID>/CloudFront/*` menjadi `amzn-s3-demo-bucket/myFolderA/*`
 - Perbarui konfigurasi logging Anda untuk menggunakan akhiran default lagi: `AWSLogs/{account-id}/CloudFront`

Untuk informasi selengkapnya, lihat [izin](#).

Hapus file log

CloudFront tidak secara otomatis menghapus file log dari tujuan Anda. Untuk informasi tentang menghapus file log, lihat topik berikut:

Amazon S3

- [Menghapus objek](#) di Panduan Pengguna Konsol Layanan Penyimpanan Sederhana Amazon

CloudWatch Log

- [Bekerja dengan grup log dan aliran log](#) di Panduan Pengguna Amazon CloudWatch Logs
- [DeleteLogGroup](#) di Referensi API Amazon CloudWatch Logs

Firehose

- [DeleteDeliveryStream](#) di Referensi API Firehose Data Amazon

Harga

CloudFront tidak mengenakan biaya untuk mengaktifkan log standar. Namun, Anda dapat dikenakan biaya untuk pengiriman, konsumsi, penyimpanan, atau akses, tergantung pada tujuan pengiriman log yang Anda pilih. Untuk informasi selengkapnya, lihat [Harga CloudWatch Log Amazon](#). Di bawah Tingkat Berbayar, pilih tab Log, lalu di bawah Log Terjual, lihat informasi untuk setiap tujuan pengiriman.

Untuk informasi selengkapnya tentang harga untuk masing-masing Layanan AWS, lihat topik berikut:

- [Harga Amazon CloudWatch Log](#)
- [Harga Amazon Data Firehose](#)
- [Harga Amazon S3](#)

Note

Tidak ada biaya tambahan untuk pengiriman log ke Amazon S3, meskipun Anda dikenakan biaya Amazon S3 untuk menyimpan dan mengakses file log. Jika Anda mengaktifkan opsi Parquet untuk mengonversi log akses Anda ke Apache Parquet, opsi ini dikenakan biaya.

CloudWatch Untuk informasi selengkapnya, lihat [bagian Vended Logs untuk CloudWatch harga](#).

Mengonfigurasi pencatatan standar (lama)

Catatan

- Topik ini untuk versi logging standar sebelumnya. Untuk versi terbaru, lihat [Mengonfigurasi pencatatan standar \(v2\)](#).
- Jika Anda sudah mengaktifkan pencatatan standar (lama) dan ingin mengaktifkan pencatatan standar (v2) ke Amazon S3, sebaiknya tentukan bucket Amazon S3 yang berbeda atau gunakan jalur terpisah di bucket yang sama (misalnya, gunakan awalan log atau partisi). Ini membantu Anda melacak file log mana yang terkait dengan distribusi mana dan mencegah file log saling menimpa.

Untuk memulai dengan logging standar (legacy), selesaikan langkah-langkah berikut:

1. Pilih bucket Amazon S3 yang akan menerima log Anda dan tambahkan izin yang diperlukan.
2. Konfigurasi logging standar (legacy) dari CloudFront konsol atau CloudFront API. Anda hanya dapat memilih bucket Amazon S3 untuk menerima log Anda.
3. Lihat log akses Anda.

Pilih bucket Amazon S3 untuk log standar

Saat Anda mengaktifkan pencatatan untuk distribusi, Anda menentukan bucket Amazon S3 yang Anda inginkan CloudFront untuk menyimpan file log. Jika Anda menggunakan Amazon S3 sebagai asal Anda, sebaiknya gunakan bucket terpisah untuk file log Anda.

Tentukan bucket Amazon S3 tempat Anda CloudFront ingin menyimpan log akses, misalnya, `amzn-s3-demo-bucket.s3.amazonaws.com`

Anda dapat menyimpan file log untuk beberapa distribusi dalam bucket yang sama. Saat Anda mengaktifkan log, Anda dapat menentukan prefiks opsional untuk nama file, sehingga Anda dapat terus melacak file log yang terkait dengan distribusi yang mana.

Tentang memilih ember S3

- Bucket Anda harus mengaktifkan daftar kontrol akses (ACL). Jika Anda memilih bucket tanpa ACL diaktifkan dari CloudFront konsol, pesan kesalahan akan muncul. Lihat [Izin](#).
- Jangan memilih bucket Amazon S3 dengan [Kepemilikan Objek S3](#) yang disetel ke pemilik bucket yang diberlakukan. Pengaturan itu dinonaktifkan ACLs untuk ember dan objek di dalamnya, yang CloudFront mencegah pengiriman file log ke ember.
- Pencatatan lama tidak mendukung bucket Amazon S3 di wilayah keikutsertaan. Silakan pilih wilayah yang diaktifkan secara default atau gunakan [Standard Logging V2](#) yang mendukung wilayah keikutsertaan dan fitur tambahan. Untuk daftar wilayah default dan opt-in, lihat [Wilayah AWS](#).

Izin

Important

Mulai April 2023, Anda harus mengaktifkan S3 ACLs untuk bucket S3 baru yang digunakan untuk log standar. CloudFront Anda dapat mengaktifkan ACLs saat [membuat bucket](#), atau mengaktifkan [bucket ACLs yang sudah ada](#).

Untuk informasi selengkapnya tentang perubahan, lihat [Pengaturan default untuk bucket S3 baru FAQ](#) di Panduan Pengguna Amazon Simple Storage Service dan [Heads-Up: Perubahan Keamanan Amazon S3 Akan Datang pada bulan April 2023 di Blog](#) Berita.AWS

Anda Akun AWS harus memiliki izin berikut untuk bucket yang Anda tentukan untuk file log:

- ACL untuk ember harus memberi Anda FULL_CONTROL. Jika Anda adalah pemilik bucket, akun Anda memiliki izin ini secara default. Jika tidak, pemilik keranjang harus memperbarui ACL untuk bucket.
- s3:GetBucketAcl
- s3:PutBucketAcl

ACL untuk bucket

Saat Anda membuat atau memperbarui distribusi dan mengaktifkan pencatatan, CloudFront gunakan izin ini untuk memperbarui ACL untuk bucket guna memberikan izin `awslogsdelivery` akun `FULL_CONTROL`. `awslogsdelivery` akun menulis file log ke dalam bucket. Jika akun Anda tidak memiliki izin yang diperlukan untuk memperbarui ACL, membuat atau memperbarui distribusi akan gagal.

Dalam beberapa keadaan, jika Anda secara program mengirimkan permintaan untuk membuat bucket tetapi bucket dengan nama yang ditentukan sudah ada, S3 mengatur ulang izin pada bucket ke nilai default. Jika Anda mengonfigurasi CloudFront untuk menyimpan log akses di bucket S3 dan Anda berhenti mendapatkan log di bucket itu, periksa izin di bucket untuk memastikan bahwa CloudFront memiliki izin yang diperlukan.

Memulihkan ACL untuk ember

Jika Anda menghapus izin untuk `awslogsdelivery` pelanggan, CloudFront tidak akan dapat menyimpan log ke bucket S3. Untuk mengaktifkan CloudFront untuk mulai menyimpan log untuk distribusi Anda lagi, pulihkan izin ACL dengan melakukan salah satu hal berikut:

- Nonaktifkan pencatatan untuk distribusi Anda CloudFront, lalu aktifkan lagi. Untuk informasi selengkapnya, lihat [Pencatatan standar](#).
- Tambahkan izin ACL untuk `awslogsdelivery` secara manual dengan menavigasi ke bucket S3 di konsol Amazon S3 dan menambahkan izin. Untuk menambahkan ACL untuk `awslogsdelivery`, Anda harus memberikan ID kanonik untuk akun tersebut, yang merupakan di Wilayah Tiongkok:

```
c4c1ede66af53448b93c283ce9448c4ba468c9432aa01d700d3878632f77d2d0
```

Untuk informasi selengkapnya tentang menambahkan ACLs ke bucket S3, lihat [Mengonfigurasi ACLs di Panduan](#) Pengguna Layanan Penyimpanan Sederhana Amazon.

ACL untuk setiap file log

Selain ACL pada bucket, ada ACL pada setiap file log. Pemilik keranjang memiliki `FULL_CONTROL` izin di setiap file log, pemilik distribusi (jika berbeda dari pemilik bucket) tidak memiliki izin, dan `awslogsdelivery` akun memiliki izin baca dan tulis.

Menonaktifkan log

Jika Anda menonaktifkan logging, CloudFront tidak menghapus baik ACLs untuk bucket atau file log. Anda dapat menghapus ACLs jika diperlukan.

Kebijakan kunci yang diperlukan untuk bucket SSE-KMS

Jika bucket S3 untuk log standar Anda menggunakan enkripsi sisi server dengan AWS KMS keys (SSE-KMS) dengan menggunakan kunci terkelola pelanggan, Anda harus menambahkan pernyataan berikut ke kebijakan kunci untuk kunci terkelola pelanggan Anda. Hal ini memungkinkan CloudFront untuk menulis file log ke bucket. Anda tidak dapat menggunakan SSE-KMS dengan Kunci yang dikelola AWS karena CloudFront tidak akan dapat menulis file log ke ember.

```
{
  "Sid": "Allow CloudFront to use the key to deliver logs",
  "Effect": "Allow",
  "Principal": {
    "Service": "delivery.logs.amazonaws.com"
  },
  "Action": "kms:GenerateDataKey*",
  "Resource": "*"
}
```

Jika bucket S3 untuk log standar Anda menggunakan SSE-KMS dengan [Kunci Bucket S3](#), Anda juga perlu menambahkan `kms:Decrypt` izin ke pernyataan kebijakan. Dalam hal ini, pernyataan kebijakan penuh terlihat seperti berikut ini.

```
{
  "Sid": "Allow CloudFront to use the key to deliver logs",
  "Effect": "Allow",
  "Principal": {
    "Service": "delivery.logs.amazonaws.com"
  },
  "Action": [
    "kms:GenerateDataKey*",
    "kms:Decrypt"
  ],
  "Resource": "*"
}
```

Note

Saat Anda mengaktifkan SSE-KMS untuk bucket S3 Anda, tentukan ARN lengkap untuk kunci yang dikelola pelanggan. Untuk informasi selengkapnya, lihat [Menentukan enkripsi sisi](#)

[server dengan AWS KMS keys \(SSE-KMS\)](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

Aktifkan pencatatan standar (lama)

Untuk mengaktifkan log standar, gunakan CloudFront konsol atau CloudFront API.

Daftar Isi

- [Aktifkan pencatatan standar \(lama\) \(CloudFront konsol\)](#)
- [Aktifkan pencatatan standar \(lama\) \(CloudFront API\)](#)

Aktifkan pencatatan standar (lama) (CloudFront konsol)

Untuk mengaktifkan log standar untuk CloudFront distribusi (konsol)

1. Gunakan CloudFront konsol untuk membuat [distribusi baru](#) atau [memperbarui yang sudah ada](#).
2. Untuk bagian Pencatatan standar, untuk pengiriman Log, pilih Aktif.
3. (Opsional) Untuk pencatatan Cookie, pilih Aktif jika Anda ingin menyertakan cookie di log Anda. Untuk informasi selengkapnya, lihat [Pencatatan cookie](#).

 Tip

Cookie logging adalah pengaturan global yang berlaku untuk semua log standar untuk distribusi Anda. Anda tidak dapat mengganti pengaturan ini untuk tujuan pengiriman terpisah.

4. Untuk bagian Kirim ke, tentukan Amazon S3 (Legacy).
5. Tentukan bucket Amazon S3 Anda. Jika Anda belum memilikinya, Anda dapat memilih Buat atau melihat dokumentasi untuk [membuat ember](#).
6. (Opsional) Untuk awalan Log, tentukan string, jika ada, yang ingin Anda awalan CloudFront ke nama file log akses untuk distribusi ini, misalnya, `exampleprefix/`. The trailing slash (/) adalah opsional tetapi dianjurkan untuk menyederhanakan browsing file log Anda. Untuk informasi selengkapnya, lihat [Awalan log](#).
7. Selesaikan langkah-langkah untuk memperbarui atau membuat distribusi Anda.
8. Dari halaman Log, verifikasi bahwa status log standar Diaktifkan di sebelah distribusi.

Untuk informasi selengkapnya tentang pengiriman logging standar dan bidang log, lihat [Referensi logging standar](#).

Aktifkan pencatatan standar (lama) (CloudFront API)

Anda juga dapat menggunakan CloudFront API untuk mengaktifkan log standar untuk distribusi Anda.

Untuk mengaktifkan log standar untuk distribusi (CloudFront API)

- Gunakan operasi [CreateDistribution](#) atau [UpdateDistribution](#) API dan konfigurasi [LoggingConfig](#) objek.

Edit pengaturan logging standar

Anda dapat mengaktifkan atau menonaktifkan logging, mengubah bucket Amazon S3 tempat log Anda disimpan, dan mengubah awalan untuk file log menggunakan [CloudFront konsol atau API](#). CloudFront Perubahan Anda pada pengaturan log mulai berlaku dalam 12 jam.

Untuk informasi selengkapnya, lihat topik berikut:

- Untuk memperbarui distribusi menggunakan CloudFront konsol, lihat [Perbarui distribusi](#).
- Untuk memperbarui distribusi menggunakan CloudFront API, lihat [UpdateDistribution](#) di Referensi Amazon CloudFront API.

Mengirim log ke Amazon S3

Saat Anda mengirim log ke Amazon S3, log Anda muncul dalam format berikut.

Format nama file

Nama setiap file log yang CloudFront disimpan di bucket Amazon S3 Anda menggunakan format nama file berikut:

<optional prefix>/<distribution ID>.YYYY-MM-DD-HH.unique-ID.gz

Tanggal dan waktu berada dalam Waktu Universal Terkoordinasi (UTC).

Misalnya, jika Anda menggunakan `example-prefix` karena prefiks, dan ID distribusi Anda adalah `EMLARXS9EXAMPLE`, nama file Anda terlihat mirip dengan ini:

example-prefix/EMLARXS9EXAMPLE.2019-11-14-20.RT4KCN4SGK9.gz

Saat Anda mengaktifkan pencatatan untuk distribusi, Anda dapat menentukan prefiks opsional untuk nama file, sehingga Anda dapat melacak file log yang terkait dengan distribusi mana. Jika Anda menyertakan nilai untuk awalan file log dan awalan Anda tidak diakhiri dengan garis miring (/), CloudFront tambahkan satu secara otomatis. Jika awalan Anda diakhiri dengan garis miring ke depan, jangan CloudFront tambahkan yang lain.

.gzDi akhir nama file menunjukkan bahwa CloudFront telah dikompresi file log menggunakan gzip.

Format file log standar

Setiap entri dalam file log memberikan detail tentang permintaan penampil tunggal. File log memiliki karakteristik sebagai berikut:

- Gunakan [Format file log yang diperluas W3C](#).
- Berisi nilai yang dipisahkan dengan tab.
- Berisi catatan yang tidak selalu kronologis.
- Berisi dua baris header: satu dengan versi format file, dan satu lagi yang mencantumkan kolom W3C yang disertakan dalam setiap catatan.
- Berisi URL berkode setara untuk spasi dan karakter tertentu lainnya dalam nilai kolom.

Ekuivalen berkode URL digunakan untuk karakter berikut ini:

- Kode karakter ASCII 0 melalui 32, inklusif
- Kode karakter ASCII 127 dan lebih tinggi
- Semua karakter dalam tabel berikut

Standar pengkodean URL ditetapkan dalam [RFC 1738](#).

Nilai yang dikodekan URL	Karakter
%3C	<
%3E	>
%22	"
%23	#

Nilai yang dikodekan URL	Karakter
%25	%
%7B	{
%7D	}
%7C	
%5C	\
%5E	^
%7E	~
%5B	[
%5D	]
%60	`
%27	'
%20	yang lebih besar

Hapus file log

CloudFront tidak secara otomatis menghapus file log dari bucket Amazon S3 Anda. Untuk informasi tentang menghapus file log dari bucket Amazon S3, [lihat Menghapus](#) objek di Panduan Pengguna Konsol Layanan Penyimpanan Sederhana Amazon.

Harga

Pencatatan standar adalah fitur opsional dari CloudFront. CloudFront tidak mengenakan biaya untuk mengaktifkan log standar. Namun, Anda memperoleh biaya Amazon S3 biasa untuk menyimpan dan mengakses file di Amazon S3. Anda dapat menghapusnya kapan saja.

Untuk informasi selengkapnya tentang harga Amazon S3, lihat [Harga Amazon S3](#).

Untuk informasi selengkapnya tentang CloudFront harga, lihat [CloudFront Harga](#).

Referensi logging standar

Bagian berikut berlaku untuk logging standar (v2) dan logging standar (legacy).

Topik

- [Waktu pengiriman file log](#)
- [Bagaimana permintaan dicatat saat URL atau header permintaan melebihi ukuran maksimum](#)
- [Bidang file log](#)
- [Menganalisis log](#)

Waktu pengiriman file log

CloudFront mengirimkan log untuk distribusi hingga beberapa kali dalam satu jam. Secara umum, file log berisi informasi tentang permintaan yang CloudFront diterima selama periode waktu tertentu. CloudFront biasanya mengirimkan file log untuk periode waktu tersebut ke tujuan Anda dalam waktu satu jam dari peristiwa yang muncul di log. Namun, perhatikan bahwa beberapa atau semua entri file log untuk periode waktu terkadang dapat tertunda hingga 24 jam. Ketika entri log tertunda, CloudFront simpan dalam file log yang nama file termasuk tanggal dan waktu periode di mana permintaan terjadi, bukan tanggal dan waktu ketika file dikirim.

Saat membuat file log, CloudFront konsolidasikan informasi untuk distribusi Anda dari semua lokasi tepi yang menerima permintaan untuk objek Anda selama periode waktu yang dicakup oleh file log.

CloudFront dapat menyimpan lebih dari satu file untuk jangka waktu tergantung pada berapa banyak permintaan yang CloudFront diterima untuk objek yang terkait dengan distribusi.

CloudFront mulai mengirimkan log akses dengan andal sekitar empat jam setelah Anda mengaktifkan pencatatan. Anda mungkin mendapatkan beberapa log akses sebelum waktu tersebut.

Note

Jika tidak ada pengguna yang meminta objek Anda selama periode waktu tersebut, Anda tidak akan menerima file log untuk periode tersebut.

Bagaimana permintaan dicatat saat URL atau header permintaan melebihi ukuran maksimum

Jika ukuran total semua header permintaan, termasuk cookie, melebihi 20 KB, atau jika URL melampaui 8192 byte, CloudFront tidak dapat menguraikan permintaan sepenuhnya dan tidak

dapat mencatatnya. Karena permintaan tersebut tidak dicatat, Anda tidak akan melihat kode status kesalahan HTTP kembali.

Jika badan permintaan melebihi ukuran maksimum, permintaan akan dicatat, termasuk kode status kesalahan HTTP.

Bidang file log

File log untuk distribusi berisi 33 bidang. Daftar berikut berisi setiap nama kolom, secara berurutan, bersama dengan deskripsi informasi di bidang tersebut.

1. **date**

Tanggal di mana peristiwa terjadi dalam format YYYY-MM-DD. Misalnya, 2019-06-30. Tanggal dan waktu berada dalam Waktu Universal Terkoordinasi (UTC). Untuk WebSocket koneksi, ini adalah tanggal ketika koneksi ditutup.

2. **time**

Waktu ketika CloudFront server selesai menanggapi permintaan (dalam UTC), misalnya, 01:42:39 Untuk WebSocket koneksi, ini adalah waktu ketika koneksi ditutup.

3. **x-edge-location**

Lokasi tepi yang melayani permintaan. Setiap lokasi tepi diidentifikasi oleh kode tiga huruf dan nomor yang ditetapkan secara sewenang-wenang (misalnya, DFW3). Kode tiga huruf biasanya sesuai dengan kode bandara International Air Transport Association (IATA) untuk bandara di dekat lokasi geografis lokasi tepi. (Ringkasan ini mungkin berubah di masa mendatang.)

4. **sc-bytes**

Jumlah total byte yang dikirim server ke penampil sebagai respons terhadap permintaan, termasuk header. Untuk WebSocket dan koneksi gRPC, ini adalah jumlah total byte yang dikirim dari server ke klien melalui koneksi.

5. **c-ip**

Alamat IP penampil yang membuat permintaan, misalnya, 192.0.2.183 atau 2001:0db8:85a3::8a2e:0370:7334. Jika penampil menggunakan proksi HTTP atau penyeimbang beban untuk mengirim permintaan, nilai bidang ini adalah alamat IP dari perantara atau penyeimbang beban. Lihat juga x-forwarded-for bidang.

6. **cs-method**

Metode permintaan HTTP yang diterima dari penampil.

7. **cs(Host)**

Nama domain CloudFront distribusi (misalnya, d1111111abcdef8.cloudfront.net).

8. **cs-uri-stem**

Bagian URL permintaan yang mengidentifikasi jalur dan objek (misalnya, /images/cat.jpg). Tanda tanya (?) dalam URLs dan string kueri tidak termasuk dalam log.

9. **sc-status**

Berisi salah satu nilai berikut:

- Kode status HTTP dari respon server (misalnya, 200).
- 000, yang menunjukkan bahwa penampil menutup koneksi sebelum server dapat merespons permintaan. Jika penampil menutup koneksi setelah server mulai mengirim respons, bidang ini berisi kode status HTTP dari respons yang mulai dikirim server.

10. **cs(Referer)**

Nilai dari `Referer` header dalam permintaan. Ini adalah nama domain yang membuat permintaan. Perujuk umum termasuk mesin pencari, situs web lain yang terhubung langsung ke objek Anda, dan situs web Anda sendiri.

11. **cs(User-Agent)**

Nilai dari `User-Agent` header dalam permintaan. `User-Agent` header mengidentifikasi sumber permintaan, seperti jenis perangkat dan peramban yang mengirimkan permintaan atau, jika permintaan berasal dari mesin pencari, mesin pencari mana.

12. **cs-uri-query**

Bagian atas kueri URL permintaan, jika ada.

Ketika URL tidak berisi string kueri, nilai bidang ini adalah tanda hubung (-). Untuk informasi selengkapnya, lihat [Konten cache berdasarkan parameter string kueri](#).

13. **cs(Cookie)**

`Cookie` header dalam permintaan, termasuk nama—pasangan nilai dan atribut terkait.

Jika Anda mengaktifkan pencatatan cookie, CloudFront catat cookie di semua permintaan terlepas dari cookie mana yang Anda pilih untuk diteruskan ke asal. Ketika permintaan tidak menyertakan

header cookie, nilai bidang ini adalah tanda hubung (-). Untuk informasi selengkapnya tentang cookie, lihat [Konten cache berdasarkan cookie](#).

14x-edge-result-type

Bagaimana server menggolongkan respons setelah byte terakhir meninggalkan server. Dalam beberapa kasus, jenis hasil dapat berubah antara waktu saat server siap mengirimkan respons dan waktu saat server selesai mengirimkan respons. Lihat juga `x-edge-response-result-type` bidang.

Misalnya, dalam streaming HTTP, seandainya server menemukan segmen aliran di cache. Dalam skenario itu, nilai kolom ini biasanya adalah `Hit`. Namun, jika penampil menutup koneksi sebelum server mengirimkan seluruh segmen, jenis hasil akhir (dan nilai kolom ini) adalah `Error`.

WebSocket dan koneksi gRPC akan memiliki nilai `Miss` untuk bidang ini karena kontennya tidak dapat di-cache dan diproksi langsung ke asal.

Nilai yang mungkin termasuk:

- `Hit` – Server melayani objek ke penampil dari cache.
- `RefreshHit` – Server menemukan objek dalam cache tetapi objek telah kedaluwarsa, sehingga server menghubungi asal untuk memverifikasi bahwa cache memiliki versi terbaru dari objek tersebut.
- `Miss` – Permintaan tidak dapat dipenuhi oleh objek di dalam cache, sehingga server meneruskan permintaan ke asal dan mengembalikan hasil ke penampil.
- `LimitExceeded`— Permintaan ditolak karena CloudFront kuota (sebelumnya disebut sebagai batas) terlampaui.
- `CapacityExceeded` Server mengembalikan kode status HTTP 503 karena tidak memiliki kapasitas yang cukup pada saat permintaan untuk melayani objek.
- `Error` – Biasanya, ini berarti permintaan tersebut mengakibatkan kesalahan klien (nilai `sc-status` bidang ada di 4xx atau kesalahan server (nilai `sc-status` bidang ada di 5xx beragam). Jika nilai `sc-status` adalah 200, atau jika nilai bidang ini adalah `Error` dan nilai dari `x-edge-response-result-type` bidang tidak `Error`, artinya permintaan HTTP berhasil tetapi klien terputus sebelum menerima semua byte.
- `Redirect` – Server mengarahkan penampil dari HTTP ke HTTPS sesuai dengan pengaturan distribusi.

- **LambdaExecutionError**— Fungsi Lambda @Edge yang terkait dengan distribusi tidak selesai karena asosiasi yang salah bentuk, batas waktu fungsi, masalah AWS ketergantungan, atau masalah ketersediaan umum lainnya.

15x-edge-request-id

String buram yang secara unik mengidentifikasi permintaan. CloudFront juga mengirimkan string ini di header `x-amz-cf-id` respons.

16x-host-header

Nilai yang disertakan oleh penampil dalam Host header permintaan. Jika Anda menggunakan nama CloudFront domain di objek Anda URLs (seperti `d111111abcdef8.cloudfront.net`), bidang ini berisi nama domain tersebut. Jika Anda menggunakan nama domain alternatif (CNAMEs) di objek Anda URLs (seperti `www.example.com`), bidang ini berisi nama domain alternatif.

Jika Anda menggunakan nama domain alternatif, lihat `cs(Host)` di bidang 7 untuk nama domain yang terkait dengan distribusi Anda.

17cs-protocol

Protokol permintaan pemirsa (`http`, `https`, `grpc`, `ws`, atau `wss`).

18cs-bytes

Jumlah total byte data yang disertakan oleh penampil, termasuk header. Untuk WebSocket dan koneksi gRPC, ini adalah jumlah total byte yang dikirim dari klien ke server pada koneksi.

19time-taken

Jumlah detik (hingga seperseribu detik, misalnya, `0,082`) dari saat server menerima permintaan penampil hingga saat server menulis byte terakhir dari respons ke antrian output, yang diukur pada server. Dari perspektif penampil, total waktu untuk mendapatkan respons penuh akan lebih lama dari nilai ini karena latensi jaringan dan buffering TCP.

20x-forwarded-for

Jika penampil menggunakan proksi HTTP atau timbangantor beban untuk mengirim permintaan, nilai `c-ip` adalah alamat IP dari perantara atau pemukul beban. Dalam hal ini, bidang ini adalah alamat IP penampil yang memulai permintaan. Bidang ini dapat berisi beberapa alamat IP yang dipisahkan koma. Setiap alamat IP dapat berupa IPv4 alamat (misalnya, `192.0.2.183`) atau IPv6 alamat (misalnya, `2001:0db8:85a3::8a2e:0370:7334`).

Jika penampil tidak menggunakan proksi HTTP atau penyeimbang beban, nilai bidang ini adalah tanda hubung (-).

21`ssl-protocol`

Ketika permintaan menggunakan HTTPS, bidang ini berisi SSL/TLS protokol yang dinegosiasikan oleh penampil dan server untuk mentransmisikan permintaan dan respons. Untuk daftar nilai yang mungkin, lihat SSL/TLS protokol yang didukung di [Protokol dan sandi yang didukung antara pemirsa dan CloudFront](#)

Saat `cs-protocol` di kolom 17 adalah `http`, nilai untuk kolom ini adalah tanda hubung (-).

22`ssl-cipher`

Saat permintaan menggunakan HTTPS, bidang ini berisi SSL/TLS sandi yang dinegosiasikan oleh penampil dan server untuk mengenkripsi permintaan dan respons. Untuk daftar nilai yang mungkin, lihat SSL/TLS cipher yang didukung di [Protokol dan sandi yang didukung antara pemirsa dan CloudFront](#)

Saat `cs-protocol` di kolom 17 adalah `http`, nilai untuk kolom ini adalah tanda hubung (-).

23`x-edge-response-result-type`

Bagaimana server mengklasifikasikan respons tepat sebelum mengembalikan respons ke penampil. Lihat juga `x-edge-result-type` bidang. Nilai yang mungkin termasuk:

- `Hit` – Server melayani objek ke penampil dari cache.
- `RefreshHit` – Server menemukan objek dalam cache tetapi objek telah kedaluwarsa, sehingga server menghubungi asal untuk memverifikasi bahwa cache memiliki versi terbaru dari objek tersebut.
- `Miss` – Permintaan tidak dapat dipenuhi oleh objek dalam cache, sehingga server meneruskan permintaan ke server asal dan mengembalikan hasil ke penampil.
- `LimitExceeded`— Permintaan ditolak karena CloudFront kuota (sebelumnya disebut sebagai batas) terlampaui.
- `CapacityExceeded`— Server mengembalikan kesalahan 503 karena tidak memiliki kapasitas yang cukup pada saat permintaan untuk melayani objek.
- `Error` – Biasanya, ini berarti permintaan tersebut mengakibatkan kesalahan klien (nilai `sc-status` bidang ada di 4xx atau kesalahan server (nilai `sc-status` bidang ada di 5xx beragam).

Jika nilai `x-edge-result-type` adalah `Error` dan nilai bidang ini tidak `Error`, klien terputus sebelum menyelesaikan unduhan.

- `Redirect` – Server mengarahkan penampil dari HTTP ke HTTPS sesuai dengan pengaturan distribusi.
- `LambdaExecutionError`— Fungsi Lambda `@Edge` yang terkait dengan distribusi tidak selesai karena asosiasi yang salah bentuk, batas waktu fungsi, masalah AWS ketergantungan, atau masalah ketersediaan umum lainnya.

24.`cs-protocol-version`

Versi HTTP yang ditentukan penampil dalam permintaan. Nilai yang mungkin termasuk adalah `HTTP/0.9`, `HTTP/1.0`, `HTTP/1.1`, `HTTP/2.0`, dan `HTTP/3.0`.

25.`file-status`

Saat [enkripsi tingkat lapangan](#) dikonfigurasi untuk distribusi, bidang ini berisi kode yang menunjukkan apakah badan permintaan berhasil diproses. Ketika server berhasil memproses isi permintaan, mengenkripsi nilai dalam bidang yang ditentukan, dan meneruskan permintaan ke asal, nilai bidang ini adalah `Processed`. Nilai dari `x-edge-result-type` masih dapat menunjukkan kesalahan sisi klien atau sisi server dalam kasus ini.

Nilai yang mungkin untuk kolom ini meliputi:

- `ForwardedByContentType` – Server meneruskan permintaan ke tempat asal tanpa mengurai atau enkripsi karena tidak ada jenis konten yang dikonfigurasi.
- `ForwardedByQueryArgs`— Server meneruskan permintaan ke asal tanpa parsing atau enkripsi karena permintaan berisi argumen kueri yang tidak ada dalam konfigurasi untuk enkripsi tingkat lapangan.
- `ForwardedDueToNoProfile` – Server meneruskan permintaan ke tempat asal tanpa mengurai atau enkripsi karena tidak ada profil yang ditentukan dalam konfigurasi untuk enkripsi tingkat lapangan.
- `MalformedContentTypeClientError` – Server menolak permintaan dan mengembalikan kode status HTTP 400 ke penampil karena nilai `Content-Type` header dalam format yang tidak valid.
- `MalformedInputClientError` – Server menolak permintaan dan mengembalikan kode status HTTP 400 ke penampil karena bodi permintaan dalam format yang tidak valid.
- `MalformedQueryArgsClientError` – Server menolak permintaan dan mengembalikan kode status HTTP 400 ke penampil karena argumen kueri kosong atau dalam format yang tidak valid.

- `RejectedByContentType` – Server menolak permintaan dan mengembalikan kode status HTTP 400 ke penampil karena tidak ada jenis konten yang ditentukan dalam konfigurasi untuk enkripsi tingkat lapangan.
- `RejectedByQueryArgs` – Server menolak permintaan dan mengembalikan kode status HTTP 400 ke penampil karena tidak ada alasan kueri yang ditentukan dalam konfigurasi untuk enkripsi tingkat lapangan.
- `ServerError` – Server asal mengembalikan kesalahan.

Jika permintaan melebihi kuota enkripsi tingkat lapangan (sebelumnya disebut sebagai batas), bidang ini berisi salah satu kode kesalahan berikut, dan server mengembalikan kode status HTTP 400 ke penampil. Untuk daftar kuota saat ini pada enkripsi tingkat lapangan, lihat [Kuotas pada enkripsi tingkat lapangan](#).

- `FieldLengthLimitClientError` – Kolom yang dikonfigurasi untuk dienkripsi melebihi panjang maksimum yang diizinkan.
- `FieldNumberLimitClientError` – Permintaan agar distribusi dikonfigurasi untuk mengenkripsi berisi lebih dari jumlah kolom yang diperbolehkan.
- `RequestLengthLimitClientError` – Panjang badan permintaan melebihi panjang maksimum yang diperbolehkan ketika enkripsi tingkat lapangan dikonfigurasi.

Jika enkripsi tingkat bidang tidak dikonfigurasi untuk distribusi, nilai bidang ini adalah tanda hubung (-).

26.fle-encrypted-fields

Jumlah bidang [enkripsi tingkat lapangan yang dienkripsi](#) dan diteruskan server ke asal. CloudFront server mengalirkan permintaan yang diproses ke asal saat mereka mengenkripsi data, sehingga bidang ini dapat memiliki nilai meskipun nilainya fle-status adalah kesalahan.

Jika enkripsi tingkat bidang tidak dikonfigurasi untuk distribusi, nilai bidang ini adalah tanda hubung (-).

27.c-port

Nomor port permintaan dari penampil.

28.time-to-first-byte

Jumlah detik antara menerima permintaan dan menulis byte pertama respons, sebagaimana diukur pada server.

29x-edge-detailed-result-type

Bidang ini berisi nilai yang sama dengan x-edge-result-type bidang, kecuali dalam kasus berikut:

- Ketika objek disajikan ke penampil dari lapisan [Origin Shield](#), bidang ini berisi `OriginShieldHit`.
- Ketika objek tidak dalam CloudFront cache dan respons dihasilkan oleh [permintaan asal fungsi Lambda @Edge](#), bidang ini berisi `MissGeneratedResponse`.
- Ketika nilai bidang adalah `Error`, x-edge-result-type bidang ini berisi salah satu nilai berikut dengan informasi lebih lanjut tentang kesalahan:
 - `AbortedOrigin` – Server mengalami masalah dengan asal usul.
 - `ClientCommError` – Respons ke penampil terganggu karena masalah komunikasi antara server dan penampil.
 - `ClientGeoBlocked`— Distribusi dikonfigurasi untuk menolak permintaan dari lokasi geografis pemirsa.
 - `ClientHungUpRequest` – Penampil berhenti sebelum waktunya saat mengirim permintaan.
 - `Error`— Terjadi kesalahan yang jenis kesalahannya tidak sesuai dengan kategori lainnya. Jenis kesalahan ini dapat terjadi saat server menjalankan respons kesalahan dari cache.
 - `InternalError`— Terjadi CloudFront kesalahan internal. Ini biasanya menunjukkan bahwa kesalahan berasal dari CloudFront infrastruktur, bukan dari asal atau penampil.
 - `InvalidRequest` – Server menerima permintaan yang tidak valid dari penampil.
 - `InvalidRequestBlocked` – Akses ke sumber daya yang diminta diblokir.
 - `InvalidRequestCertificate`— Distribusi tidak cocok dengan SSL/TLS sertifikat tempat koneksi HTTPS dibuat.
 - `InvalidRequestHeader`— Permintaan mengandung header yang tidak valid.
 - `InvalidRequestMethod` – Distribusi tidak dikonfigurasi untuk menangani metode permintaan HTTP yang digunakan. Ini dapat terjadi ketika distribusi hanya mendukung permintaan yang dapat disimpan.
 - `LambdaLimitExceeded`— Permintaan melebihi batas fungsi Lambda @Edge.
 - `OriginCommError`— Permintaan habis waktu saat menghubungkan ke asal, atau membaca data dari asal.
 - `OriginConnectError`— Server tidak dapat terhubung ke asal.

- `OriginContentRangeLengthError`— Content-Length Header dalam respons asal tidak cocok dengan panjang di Content-Range header.
- `OriginDnsError`— Server tidak dapat menyelesaikan nama domain asal.
- `OriginError` - Asal memberikan jawaban yang salah.
- `OriginHeaderTooBigError` – Header yang dikembalikan oleh asalnya terlalu besar untuk diproses oleh server edge.
- `OriginInvalidResponseError` – Asal memberikan respons tidak valid.
- `OriginReadError`— Server tidak bisa membaca dari asalnya.
- `OriginWriteError`— Server tidak bisa menulis ke asal.
- `OriginZeroSizeObjectError` – Objek berukuran nol yang dikirim dari sumber mengakibatkan kesalahan.
- `SlowReaderOriginError` – Penampil lambat untuk membaca pesan yang menyebabkan kesalahan asal.

30 **sc-content-type**

Nilai HTTP Content-Type header respons.

31 **sc-content-len**

Nilai HTTP Content-Length header respons.

32 **sc-range-start**

Saat tanggapan berisi HTTP Content-Range header, kolom ini berisi nilai mulai rentang.

33 **sc-range-end**

Saat tanggapan berisi HTTP Content-Range header, kolom ini berisi nilai akhir rentang.

34 **distribution-tenant-id**

ID penyewa distribusi.

35 **connection-id**

Pengidentifikasi unik untuk koneksi TLS.

Anda harus mengaktifkan MTL untuk distribusi Anda sebelum Anda bisa mendapatkan informasi untuk bidang ini. Untuk informasi selengkapnya, lihat [Otentikasi TLS timbal balik dengan CloudFront \(Viewer mTLS\)](#).

Berikut ini adalah contoh file log untuk distribusi.

```
#Version: 1.0
#Fields: date time x-edge-location sc-bytes c-ip cs-method cs(Host) cs-uri-stem sc-
status cs(Referer) cs(User-Agent) cs-uri-query cs(Cookie) x-edge-result-type x-edge-
request-id x-host-header cs-protocol cs-bytes time-taken x-forwarded-for ssl-protocol
ssl-cipher x-edge-response-result-type cs-protocol-version fle-status fle-encrypted-
fields c-port time-to-first-byte x-edge-detailed-result-type sc-content-type sc-
content-len sc-range-start sc-range-end
2019-12-04 21:02:31 LAX1 392 192.0.2.100 GET d111111abcdef8.cloudfront.net /
index.html 200 - Mozilla/5.0%20(Windows%20NT%2010.0;%20Win64;
%20x64)%20AppleWebKit/537.36%20(KHTML,%20like
%20Gecko)%20Chrome/78.0.3904.108%20Safari/537.36 - - Hit
S0X4xwn4XV6Q4rgb7XiVG0Hms_BG1TAC4KyHmureZmBNrjGdRLiNIQ== d111111abcdef8.cloudfront.net
https 23 0.001 - TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256 Hit HTTP/2.0 - - 11040 0.001 Hit
text/html 78 - -
2019-12-04 21:02:31 LAX1 392 192.0.2.100 GET d111111abcdef8.cloudfront.net /
index.html 200 - Mozilla/5.0%20(Windows%20NT%2010.0;%20Win64;
%20x64)%20AppleWebKit/537.36%20(KHTML,%20like
%20Gecko)%20Chrome/78.0.3904.108%20Safari/537.36 - - Hit
k6WGMNkEzR5BEM_SaF47gjtX9zBD02m3490Y2an0QPEaUum1Z0Lrow== d111111abcdef8.cloudfront.net
https 23 0.000 - TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256 Hit HTTP/2.0 - - 11040 0.000 Hit
text/html 78 - -
2019-12-04 21:02:31 LAX1 392 192.0.2.100 GET d111111abcdef8.cloudfront.net /
index.html 200 - Mozilla/5.0%20(Windows%20NT%2010.0;%20Win64;
%20x64)%20AppleWebKit/537.36%20(KHTML,%20like
%20Gecko)%20Chrome/78.0.3904.108%20Safari/537.36 - - Hit
f37nTMVvnKvV2ZSvEsivup_c2kZ7VXzYdjC-GUQZ5qNs-89BlWazbw== d111111abcdef8.cloudfront.net
https 23 0.001 - TLSv1.2 ECDHE-RSA-AES128-GCM-SHA256 Hit HTTP/2.0 - - 11040 0.001 Hit
text/html 78 - -
2019-12-13 22:36:27 SEA19-C1 900 192.0.2.200 GET d111111abcdef8.cloudfront.net /
favicon.ico 502 http://www.example.com/ Mozilla/5.0%20(Windows
%20NT%2010.0;%20Win64;%20x64)%20AppleWebKit/537.36%20(KHTML,
%20like%20Gecko)%20Chrome/78.0.3904.108%20Safari/537.36 - - Error
1pkpNfBQ39sYMNjjUQjmH2w1wdJnbHYTbag21o_30fcQgPzdL2RSSQ== www.example.com http 675
0.102 - - - Error HTTP/1.1 - - 25260 0.102 OriginDnsError text/html 507 - -
2019-12-13 22:36:26 SEA19-C1 900 192.0.2.200 GET d111111abcdef8.cloudfront.net / 502
- Mozilla/5.0%20(Windows%20NT%2010.0;%20Win64;%20x64)%20AppleWebKit/537.36%20(KHTML,
%20like%20Gecko)%20Chrome/78.0.3904.108%20Safari/537.36 - - Error
3AqrZGCnF_g0-5K0vfA7c9XLcf4YGvMFSeFdIetR1N_2y8jSis8Zxg== www.example.com http 735
0.107 - - - Error HTTP/1.1 - - 3802 0.107 OriginDnsError text/html 507 - -
2019-12-13 22:37:02 SEA19-C2 900 192.0.2.200 GET d111111abcdef8.cloudfront.net / 502
- curl/7.55.1 - - Error kBkDzGnceVtWHqSCqBUqtA_cEs2T3tFUBbnBNkB9El_uVRhHgcZfcw==
```



```
www.example.com http 387 0.103 - - - Error HTTP/1.1 - - 12644 0.103 OriginDnsError
text/html 507 - -
```

Menganalisis log

Karena Anda dapat menerima beberapa log akses per jam, kami sarankan Anda menggabungkan semua file log yang Anda terima untuk periode waktu tertentu ke dalam satu file. Anda kemudian dapat menganalisis data untuk periode tersebut dengan lebih akurat dan lengkap.

Salah satu cara untuk menganalisis log akses Anda adalah dengan menggunakan [Amazon Athena](#). Athena adalah layanan kueri interaktif yang dapat membantu Anda menganalisis data untuk AWS layanan, termasuk CloudFront. Untuk mempelajari selengkapnya, lihat [Menanyakan CloudFront Log Amazon](#) di Panduan Pengguna Amazon Athena.

Selain itu, posting AWS blog berikut membahas beberapa cara untuk menganalisis log akses.

- [Amazon CloudFront Request Logging](#) (untuk konten yang dikirimkan melalui HTTP)
- [CloudFront Log yang Ditingkatkan, Sekarang Dengan String Kueri](#)

Gunakan log akses waktu nyata

Dengan log akses CloudFront real-time, Anda bisa mendapatkan informasi tentang permintaan yang dibuat ke distribusi secara real time (log dikirimkan dalam hitungan detik setelah menerima permintaan). Anda dapat menggunakan log akses real-time untuk memantau, menganalisis, dan mengambil tindakan berdasarkan kinerja pengiriman konten.

CloudFront log akses real-time dapat dikonfigurasi. Anda dapat memilih:

- Rasio pengambilan sampel untuk log waktu nyata Anda—yaitu, persentase permintaan yang ingin Anda terima catatan log akses real-time.
- Kolom khusus yang ingin Anda terima di catatan log.
- Perilaku cache tertentu (pola jalur) yang ingin Anda terima log waktu nyata.

CloudFront log akses real-time dikirimkan ke aliran data pilihan Anda di Amazon Kinesis Data Streams. Anda dapat membangun [konsumen aliran data Kinesis](#) Anda sendiri, atau menggunakan Amazon Data Firehose untuk mengirim data log ke Amazon Simple Storage Service (Amazon S3), Amazon Redshift, Amazon Service (Service), atau OpenSearch layanan pemrosesan OpenSearch log pihak ketiga.

CloudFront biaya untuk log akses real-time, selain biaya yang Anda keluarkan untuk menggunakan Kinesis Data Streams. Untuk informasi selengkapnya tentang harga, lihat [Amazon CloudFront Harga](#) dan [Harga Amazon Kinesis Data Streams](#).

Important

Kami menyarankan Anda menggunakan log untuk memahami sifat permintaan untuk konten Anda, bukan sebagai akuntansi lengkap dari semua permintaan. CloudFront memberikan log akses real-time dengan upaya terbaik. Entri log untuk permintaan tertentu mungkin dikirim dalam waktu lama setelah permintaan diproses secara aktual dan, dalam kasus yang jarang, entri log mungkin tidak dikirimkan sama sekali. Ketika entri log dihilangkan dari log akses real-time, jumlah entri dalam log akses real-time tidak akan cocok dengan penggunaan yang muncul dalam laporan AWS penagihan dan penggunaan.

Topik

- [Membuat dan menggunakan konfigurasi log akses real-time](#)
- [Memahami konfigurasi log akses real-time](#)
- [Membuat konsumen Kinesis Data Streams](#)
- [Memecahkan masalah log akses waktu nyata](#)

Membuat dan menggunakan konfigurasi log akses real-time

Untuk mendapatkan informasi tentang permintaan yang dibuat untuk distribusi secara real time. Anda dapat menggunakan konfigurasi log akses real-time. Log dikirimkan dalam hitungan detik setelah menerima permintaan. Anda dapat membuat konfigurasi log akses real-time di CloudFront konsol, dengan AWS Command Line Interface (AWS CLI), atau dengan CloudFront API.

Untuk menggunakan konfigurasi log akses real-time, Anda melampirkannya ke satu atau beberapa perilaku cache dalam CloudFront distribusi.

Console

Untuk membuat konfigurasi log akses real-time

1. Masuk ke Konsol Manajemen AWS dan buka halaman Log di CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home?#/logs>.
2. Pilih tab Konfigurasi waktu nyata.

3. Pilih Buat konfigurasi.
4. Untuk Nama, masukkan nama untuk konfigurasi.
5. Untuk Sampling rate, masukkan persentase permintaan yang ingin Anda terima catatan log.
6. Untuk Bidang, pilih bidang yang akan diterima di log akses real-time.
 - Untuk menyertakan semua [bidang CMCD](#) untuk log Anda, pilih CMCD semua kunci.
7. Untuk Endpoint, pilih satu atau beberapa aliran data Kinesis untuk menerima log akses real-time.

 Note

CloudFront log akses real-time dikirimkan ke aliran data yang Anda tentukan di Kinesis Data Streams. Untuk membaca dan menganalisis log akses real-time Anda, Anda dapat membangun konsumen aliran data Kinesis Anda sendiri. Anda juga dapat menggunakan Firehose untuk mengirim data log ke Amazon S3, Amazon Redshift, OpenSearch Amazon Service, atau layanan pemrosesan log pihak ketiga.

8. Untuk peran IAM, pilih Buat peran layanan baru atau pilih peran yang ada. Anda harus memiliki izin untuk membuat peran IAM.
9. (Opsional) Untuk Distribusi, pilih perilaku CloudFront distribusi dan cache untuk dilampirkan ke konfigurasi log akses real-time.
10. Pilih Buat konfigurasi.

Jika berhasil, konsol menampilkan detail konfigurasi log akses real-time yang baru saja Anda buat.

Untuk informasi selengkapnya, lihat [Memahami konfigurasi log akses real-time](#).

AWS CLI

Untuk membuat konfigurasi log akses real-time dengan AWS CLI, gunakan `aws cloudfront create-realtime-log-config` perintah. Anda dapat menggunakan file input untuk memberikan parameter input perintah, daripada menentukan setiap parameter individu sebagai input baris perintah.

Untuk membuat konfigurasi log akses real-time (CLI dengan file input)

1. Gunakan perintah berikut untuk membuat file dengan nama `rtl-config.yaml` yang berisi semua parameter input untuk `create-realtime-log-config` perintah.

```
aws cloudfront create-realtime-log-config --generate-cli-skeleton yml-input >
rtl-config.yml
```

2. Buka file dengan nama `rtl-config.yml` yang baru Anda buat. Edit file untuk menentukan pengaturan konfigurasi log akses real-time yang Anda inginkan, lalu simpan file. Perhatikan hal-hal berikut:

- Untuk `StreamType`, satu-satunya nilai valid adalah `Kinesis`.

Untuk informasi lebih lanjut tentang pengaturan konfigurasi panjang waktu nyata, lihat [Memahami konfigurasi log akses real-time](#).

3. Gunakan perintah berikut untuk membuat konfigurasi log akses real-time menggunakan parameter input dari `rtl-config.yml` file.

```
aws cloudfront create-realtime-log-config --cli-input-yml file://rtl-
config.yml
```

Jika berhasil, output perintah menunjukkan rincian konfigurasi log akses real-time yang baru saja Anda buat.

Untuk melampirkan konfigurasi log akses real-time ke distribusi yang ada (CLI dengan file input)

1. Gunakan perintah berikut untuk menyimpan konfigurasi distribusi untuk CloudFront distribusi yang ingin Anda perbarui. Ganti *distribution_ID* dengan ID distribusi.

```
aws cloudfront get-distribution-config --id distribution_ID --output yml >
dist-config.yml
```

2. Buka file dengan nama `dist-config.yml` yang baru Anda buat. Edit file, buat perubahan berikut pada setiap perilaku cache yang Anda perbarui untuk menggunakan konfigurasi log akses real-time.
 - Dalam perilaku cache, tambahkan kolom bernama `RealtimeLogConfigArn`. Untuk nilai bidang, gunakan ARN konfigurasi log akses real-time yang ingin Anda lampirkan ke perilaku cache ini.

- Ubah nama ETag bidang menjadi `IfMatch`, tetapi jangan ubah nilai bidang.

Simpan file setelah selesai.

3. Gunakan perintah berikut untuk memperbarui distribusi untuk menggunakan konfigurasi log akses real-time. Ganti `distribution_ID` dengan ID distribusi.

```
aws cloudfront update-distribution --id distribution_ID --cli-input-yaml file://  
dist-config.yaml
```

Jika berhasil, output perintah menunjukkan rincian distribusi yang baru saja Anda perbarui.

API

Untuk membuat konfigurasi log akses real-time dengan CloudFront API, gunakan operasi [CreateRealtimeLogConfig](#) API. Untuk informasi selengkapnya tentang parameter yang Anda tentukan dalam panggilan API ini, lihat [Memahami konfigurasi log akses real-time](#) dan dokumentasi referensi API untuk AWS SDK atau klien API lainnya.

Setelah membuat konfigurasi log akses real-time, Anda dapat melampirkannya ke perilaku cache, dengan menggunakan salah satu operasi API berikut:

- Untuk memasangnya ke perilaku cache dalam distribusi yang ada, gunakan [UpdateDistribution](#).
- Untuk memasangnya ke perilaku cache dalam distribusi baru, gunakan [CreateDistribution](#).

Untuk kedua operasi API ini, berikan ARN konfigurasi log akses real-time di `RealtimeLogConfigArn` lapangan, di dalam perilaku cache. Untuk informasi selengkapnya tentang bidang lain yang Anda tentukan dalam panggilan API ini, lihat [Semua referensi pengaturan distribusi](#) dan dokumentasi referensi API untuk AWS SDK atau klien API lainnya.

Memahami konfigurasi log akses real-time

Untuk menggunakan log akses CloudFront real-time, Anda mulai dengan membuat konfigurasi log akses real-time. Konfigurasi log akses real-time berisi informasi tentang bidang log mana yang ingin Anda terima, laju pengambilan sampel untuk catatan log, dan aliran data Kinesis tempat Anda ingin mengirimkan log.

Secara khusus, konfigurasi log akses real-time berisi pengaturan berikut:

Daftar Isi

- [Nama](#)
- [Tingkat pengambilan sampel](#)
- [Bidang](#)
- [Titik Akhir \(Kinesis Data Streams\)](#)
- [IAM Role](#)

Nama

Nama untuk mengidentifikasi konfigurasi log akses real-time.

Tingkat pengambilan sampel

Laju pengambilan sampel adalah jumlah keseluruhan antara 1 dan 100 (inklusif) yang menentukan persentase permintaan penampil yang dikirim ke Kinesis Data Streams sebagai catatan log akses waktu nyata. Untuk menyertakan setiap permintaan penampil dalam log akses waktu nyata Anda, tentukan 100 untuk laju pengambilan sampel. Anda dapat memilih tingkat pengambilan sampel yang lebih rendah untuk mengurangi biaya saat masih menerima sampel representatif data permintaan dalam log akses waktu nyata Anda.

Bidang

Daftar bidang yang disertakan dalam setiap catatan log akses real-time. Setiap catatan log dapat berisi hingga 40 kolom, dan Anda dapat memilih untuk menerima semua bidang yang tersedia, atau hanya kolom yang Anda butuhkan untuk memantau dan menganalisis kinerja.

Daftar berikut berisi setiap nama bidang dan deskripsi informasi dalam bidang tersebut. Kolom tercantum dalam urutan tampilannya dalam catatan log yang dikirim ke Stream Data Kinesis.

Fields 46-63 adalah [data klien media umum \(CMCD\)](#) yang dapat dikirim oleh klien media player dengan setiap permintaan. CDNs Anda dapat menggunakan data ini untuk memahami setiap permintaan, seperti jenis media (audio, video), kecepatan pemutaran, dan panjang streaming. Bidang ini hanya akan muncul di log akses real-time Anda jika dikirimkan ke CloudFront.

1. **timestamp**

Tanggal dan waktu server edge selesai menanggapi permintaan.

2. **c-ip**

Alamat IP penampil yang membuat permintaan, misalnya, 192.0.2.183 atau 2001:0db8:85a3::8a2e:0370:7334. Jika penampil menggunakan proksi HTTP atau penyeimbang beban untuk mengirim permintaan, nilai bidang ini adalah alamat IP dari perantara atau penyeimbang beban. Lihat juga `x-forwarded-for` bidang.

3. **s-ip**

Alamat IP CloudFront server yang melayani permintaan, misalnya, 192.0.2.183 atau 2001:0db8:85a3::8a2e:0370:7334.

4. **time-to-first-byte**

Jumlah detik antara menerima permintaan dan menulis byte pertama respons, sebagaimana diukur pada server.

5. **sc-status**

Kode status HTTP dari respon server (misalnya, 200).

6. **sc-bytes**

Jumlah total byte yang dikirim server ke penampil sebagai respons terhadap permintaan, termasuk header. Untuk WebSocket dan koneksi gRPC, ini adalah jumlah total byte yang dikirim dari server ke klien melalui koneksi.

7. **cs-method**

Metode permintaan HTTP yang diterima dari penampil.

8. **cs-protocol**

Protokol permintaan pemirsa (`http`, `https`, `grpc`, `ws`, atau `wss`).

9. **cs-host**

Nilai yang disertakan oleh penampil dalam Host header permintaan. Jika Anda menggunakan nama CloudFront domain di objek Anda URLs (seperti `d111111abcdef8.cloudfront.net`), bidang ini berisi nama domain tersebut. Jika Anda menggunakan nama domain alternatif (CNAMEs) di objek Anda URLs (seperti `www.example.com`), bidang ini berisi nama domain alternatif.

10. **cs-uri-stem**

Seluruh URL permintaan, termasuk string kueri (jika ada), tetapi tanpa nama domain. Misalnya, `/images/cat.jpg?mobile=true`.

 Note

Dalam [log standar](#), `cs-uri-stem` nilainya tidak menyertakan string kueri.

11.cs-bytes

Jumlah total byte data yang disertakan oleh penampil, termasuk header. Untuk WebSocket dan koneksi gRPC, ini adalah jumlah total byte yang dikirim dari klien ke server pada koneksi.

12.x-edge-location

Lokasi tepi yang melayani permintaan. Setiap lokasi tepi diidentifikasi oleh kode tiga huruf dan nomor yang ditetapkan secara sewenang-wenang (misalnya, DFW3). Kode tiga huruf biasanya sesuai dengan kode bandara International Air Transport Association (IATA) untuk bandara di dekat lokasi geografis lokasi tepi. (Ringkasan ini mungkin berubah di masa mendatang.)

13.x-edge-request-id

String buram yang secara unik mengidentifikasi permintaan. CloudFront juga mengirimkan string ini di header `x-amz-cf-id` respons.

14.x-host-header

Nama domain CloudFront distribusi (misalnya, `d111111abcdef8.cloudfront.net`).

15.time-taken

Jumlah detik (hingga seperseribu detik, misalnya, 0,082) dari saat server menerima permintaan penampil hingga saat server menulis byte terakhir dari respons ke antrian output, yang diukur pada server. Dari perspektif penampil, total waktu untuk mendapatkan respons penuh akan lebih lama dari nilai ini karena latensi jaringan dan buffering TCP.

16.cs-protocol-version

Versi HTTP yang ditentukan penampil dalam permintaan. Nilai yang mungkin termasuk adalah `HTTP/0.9`, `HTTP/1.0`, `HTTP/1.1`, `HTTP/2.0`, dan `HTTP/3.0`.

17.c-ip-version

Versi IP dari permintaan (IPv4 atau IPv6).

18.cs-user-agent

Nilai dari `User-Agent` header dalam permintaan. `User-Agent` header mengidentifikasi sumber permintaan, seperti jenis perangkat dan peramban yang mengirimkan permintaan atau, jika permintaan berasal dari mesin pencari, mesin pencari mana.

19.`cs-referer`

Nilai dari `Referer` header dalam permintaan. Ini adalah nama domain yang membuat permintaan. Perujuk umum termasuk mesin pencari, situs web lain yang terhubung langsung ke objek Anda, dan situs web Anda sendiri.

20.`cs-cookie`

Cookie header dalam permintaan, termasuk nama—pasangan nilai dan atribut terkait.

Note

Field ini dipotong menjadi 800 byte.

21.`cs-uri-query`

Bagian atas kueri URL permintaan, jika ada.

22.`x-edge-response-result-type`

Bagaimana server mengklasifikasikan respons tepat sebelum mengembalikan respons ke penampil. Lihat juga `x-edge-result-type` bidang. Nilai yang mungkin termasuk:

- **Hit** – Server melayani objek ke penampil dari cache.
- **RefreshHit** – Server menemukan objek dalam cache tetapi objek telah kedaluwarsa, sehingga server menghubungi asal untuk memverifikasi bahwa cache memiliki versi terbaru dari objek tersebut.
- **Miss** – Permintaan tidak dapat dipenuhi oleh objek dalam cache, sehingga server meneruskan permintaan ke server asal dan mengembalikan hasil ke penampil.
- **LimitExceeded**— Permintaan ditolak karena CloudFront kuota (sebelumnya disebut sebagai batas) terlampaui.
- **CapacityExceeded**— Server mengembalikan kesalahan 503 karena tidak memiliki kapasitas yang cukup pada saat permintaan untuk melayani objek.
- **Error** – Biasanya, ini berarti permintaan tersebut mengakibatkan kesalahan klien (nilai `sc-status` bidang ada di 4xx atau kesalahan server (nilai `sc-status` bidang ada di 5xx beragam)).

Jika nilai `x-edge-result-type` adalah `Error` dan nilai bidang ini tidak `Error`, klien terputus sebelum menyelesaikan unduhan.

- `Redirect` – Server mengarahkan penampil dari HTTP ke HTTPS sesuai dengan pengaturan distribusi.
- `LambdaExecutionError`— Fungsi Lambda `@Edge` yang terkait dengan distribusi tidak selesai karena asosiasi yang salah bentuk, batas waktu fungsi, masalah AWS ketergantungan, atau masalah ketersediaan umum lainnya.

23 `x-forwarded-for`

Jika penampil menggunakan proksi HTTP atau timbangantor beban untuk mengirim permintaan, nilai `c-ip` adalah alamat IP dari perantara atau pemukul beban. Dalam hal ini, bidang ini adalah alamat IP penampil yang memulai permintaan. Bidang ini dapat berisi beberapa alamat IP yang dipisahkan koma. Setiap alamat IP dapat berupa IPv4 alamat (misalnya, `192.0.2.183`) atau IPv6 alamat (misalnya, `2001:0db8:85a3::8a2e:0370:7334`).

24 `ssl-protocol`

Ketika permintaan menggunakan HTTPS, bidang ini berisi SSL/TLS protokol yang dinegosiasikan oleh penampil dan server untuk mentransmisikan permintaan dan respons. Untuk daftar nilai yang mungkin, lihat SSL/TLS protokol yang didukung di [Protokol dan sandi yang didukung antara pemirsa dan CloudFront](#)

25 `ssl-cipher`

Ketika permintaan menggunakan HTTPS, bidang ini berisi SSL/TLS cipher yang dinegosiasikan oleh penampil dan server untuk mengenkripsi permintaan dan respons. Untuk daftar nilai yang mungkin, lihat SSL/TLS cipher yang didukung di [Protokol dan sandi yang didukung antara pemirsa dan CloudFront](#)

26 `x-edge-result-type`

Bagaimana server menggolongkan respons setelah byte terakhir meninggalkan server. Dalam beberapa kasus, jenis hasil dapat berubah antara waktu saat server siap mengirimkan respons dan waktu saat server selesai mengirimkan respons. Lihat juga `x-edge-response-result-type` bidang.

Misalnya, dalam streaming HTTP, seandainya server menemukan segmen aliran di cache. Dalam skenario itu, nilai kolom ini biasanya adalah `Hit`. Namun, jika penampil menutup koneksi sebelum server mengirimkan seluruh segmen, jenis hasil akhir (dan nilai kolom ini) adalah `Error`.

WebSocket dan koneksi gRPC akan memiliki nilai Miss untuk bidang ini karena kontennya tidak dapat di-cache dan diproksi langsung ke asal.

Nilai yang mungkin termasuk:

- **Hit** – Server melayani objek ke penampil dari cache.
- **RefreshHit** – Server menemukan objek dalam cache tetapi objek telah kedaluwarsa, sehingga server menghubungi asal untuk memverifikasi bahwa cache memiliki versi terbaru dari objek tersebut.
- **Miss** – Permintaan tidak dapat dipenuhi oleh objek di dalam cache, sehingga server meneruskan permintaan ke asal dan mengembalikan hasil ke penampil.
- **LimitExceeded**— Permintaan ditolak karena CloudFront kuota (sebelumnya disebut sebagai batas) terlampaui.
- **CapacityExceeded**Server mengembalikan kode status HTTP 503 karena tidak memiliki kapasitas yang cukup pada saat permintaan untuk melayani objek.
- **Error** – Biasanya, ini berarti permintaan tersebut mengakibatkan kesalahan klien (nilai `sc-status` bidang ada di 4xx atau kesalahan server (nilai `sc-status` bidang ada di 5xx beragam). Jika nilai `sc-status` adalah 200, atau jika nilai bidang ini adalah **Error** dan nilai dari `x-edge-response-result-type` bidang tidak **Error**, artinya permintaan HTTP berhasil tetapi klien terputus sebelum menerima semua byte.
- **Redirect** – Server mengarahkan penampil dari HTTP ke HTTPS sesuai dengan pengaturan distribusi.
- **LambdaExecutionError**— Fungsi Lambda `@Edge` yang terkait dengan distribusi tidak selesai karena asosiasi yang salah bentuk, batas waktu fungsi, masalah AWS ketergantungan, atau masalah ketersediaan umum lainnya.

27.fle-encrypted-fields

Jumlah bidang [enkripsi tingkat lapangan yang dienkrpsi](#) dan diteruskan server ke asal. CloudFront server mengalirkan permintaan yang diproses ke asal saat mereka mengenkripsi data, sehingga bidang ini dapat memiliki nilai meskipun nilainya `fle-status` adalah kesalahan.

28.fle-status

Saat [enkripsi tingkat lapangan](#) dikonfigurasi untuk distribusi, bidang ini berisi kode yang menunjukkan apakah badan permintaan berhasil diproses. Ketika server berhasil memproses isi permintaan, mengenkripsi nilai dalam bidang yang ditentukan, dan meneruskan permintaan

ke asal, nilai bidang ini adalah `Processed`. Nilai dari `x-edge-result-type` masih dapat menunjukkan kesalahan sisi klien atau sisi server dalam kasus ini.

Nilai yang mungkin untuk kolom ini meliputi:

- `ForwardedByContentType` – Server meneruskan permintaan ke tempat asal tanpa mengurai atau enkripsi karena tidak ada jenis konten yang dikonfigurasi.
- `ForwardedByQueryArgs`— Server meneruskan permintaan ke asal tanpa parsing atau enkripsi karena permintaan berisi argumen kueri yang tidak ada dalam konfigurasi untuk enkripsi tingkat lapangan.
- `ForwardedDueToNoProfile` – Server meneruskan permintaan ke tempat asal tanpa mengurai atau enkripsi karena tidak ada profil yang ditentukan dalam konfigurasi untuk enkripsi tingkat lapangan.
- `MalformedContentTypeClientError` – Server menolak permintaan dan mengembalikan kode status HTTP 400 ke penampil karena nilai `Content-Type` header dalam format yang tidak valid.
- `MalformedInputClientError` – Server menolak permintaan dan mengembalikan kode status HTTP 400 ke penampil karena bodi permintaan dalam format yang tidak valid.
- `MalformedQueryArgsClientError` – Server menolak permintaan dan mengembalikan kode status HTTP 400 ke penampil karena argumen kueri kosong atau dalam format yang tidak valid.
- `RejectedByContentType` – Server menolak permintaan dan mengembalikan kode status HTTP 400 ke penampil karena tidak ada jenis konten yang ditentukan dalam konfigurasi untuk enkripsi tingkat lapangan.
- `RejectedByQueryArgs` – Server menolak permintaan dan mengembalikan kode status HTTP 400 ke penampil karena tidak ada alasan kueri yang ditentukan dalam konfigurasi untuk enkripsi tingkat lapangan.
- `ServerError` – Server asal mengembalikan kesalahan.

Jika permintaan melebihi kuota enkripsi tingkat lapangan (sebelumnya disebut sebagai batas), bidang ini berisi salah satu kode kesalahan berikut, dan server mengembalikan kode status HTTP 400 ke penampil. Untuk daftar kuota saat ini pada enkripsi tingkat lapangan, lihat [Kuotas pada enkripsi tingkat lapangan](#).

- `FieldLengthLimitClientError` – Kolom yang dikonfigurasi untuk dienkripsi melebihi panjang maksimum yang diizinkan.
- `FieldNumberLimitClientError` – Permintaan agar distribusi dikonfigurasi untuk mengenkripsi berisi lebih dari jumlah kolom yang diperbolehkan.

- `RequestLengthLimitClientError` – Panjang badan permintaan melebihi panjang maksimum yang diperbolehkan ketika enkripsi tingkat lapangan dikonfigurasi.

29 `sc-content-type`

Nilai HTTP Content-Type header respons.

30 `sc-content-len`

Nilai HTTP Content-Length header respons.

31 `sc-range-start`

Saat tanggapan berisi HTTP Content-Range header, kolom ini berisi nilai mulai rentang.

32 `sc-range-end`

Saat tanggapan berisi HTTP Content-Range header, kolom ini berisi nilai akhir rentang.

33 `c-port`

Nomor port permintaan dari penampil.

34 `x-edge-detailed-result-type`

Bidang ini berisi nilai yang sama dengan `x-edge-result-type` bidang, kecuali dalam kasus berikut:

- Ketika objek disajikan ke penampil dari lapisan [Origin Shield](#), bidang ini berisi `OriginShieldHit`.
- Ketika objek tidak dalam CloudFront cache dan respons dihasilkan oleh [permintaan asal fungsi Lambda @Edge](#), bidang ini berisi `MissGeneratedResponse`.
- Ketika nilai bidang adalah `Error`, `x-edge-result-type` bidang ini berisi salah satu nilai berikut dengan informasi lebih lanjut tentang kesalahan:
 - `AbortedOrigin` – Server mengalami masalah dengan asal usul.
 - `ClientCommError` – Respons ke penampil terganggu karena masalah komunikasi antara server dan penampil.
 - `ClientGeoBlocked`— Distribusi dikonfigurasi untuk menolak permintaan dari lokasi geografis pemirsa.
 - `ClientHungUpRequest` – Penampil berhenti sebelum waktunya saat mengirim permintaan.
 - `Error`— Terjadi kesalahan yang jenis kesalahannya tidak sesuai dengan kategori lainnya.

Jenis kesalahan ini dapat terjadi saat server menjalankan respons kesalahan dari cache.

- `InternalErrorTerjadi` CloudFront kesalahan internal. Ini biasanya menunjukkan bahwa kesalahan berasal dari CloudFront infrastruktur, bukan dari asal atau penampil.
- `InvalidRequest` – Server menerima permintaan yang tidak valid dari penampil.
- `InvalidRequestBlocked` – Akses ke sumber daya yang diminta diblokir.
- `InvalidRequestCertificate`— Distribusi tidak cocok dengan SSL/TLS sertifikat tempat koneksi HTTPS dibuat.
- `InvalidRequestHeader` Permintaan mengandung header yang tidak valid.
- `InvalidRequestMethod` – Distribusi tidak dikonfigurasi untuk menangani metode permintaan HTTP yang digunakan. Ini dapat terjadi ketika distribusi hanya mendukung permintaan yang dapat disimpan.
- `LambdaLimitExceeded`— Permintaan melebihi batas fungsi Lambda @Edge.
- `OriginCommError`— Permintaan habis waktu saat menghubungkan ke asal, atau membaca data dari asal.
- `OriginConnectError`— Server tidak dapat terhubung ke asal.
- `OriginContentRangeLengthError`— Content-Length Header dalam respons asal tidak cocok dengan panjang di Content-Range header.
- `OriginDnsError`— Server tidak dapat menyelesaikan nama domain asal.
- `OriginError` - Asal memberikan jawaban yang salah.
- `OriginHeaderTooBigError` – Header yang dikembalikan oleh asalnya terlalu besar untuk diproses oleh server edge.
- `OriginInvalidResponseError` – Asal memberikan respons tidak valid.
- `OriginReadError`— Server tidak bisa membaca dari asalnya.
- `OriginWriteError`— Server tidak bisa menulis ke asal.
- `OriginZeroSizeObjectError` – Objek seukuran nol yang dikirim dari sumber mengakibatkan kesalahan.
- `SlowReaderOriginError` – Penampil lambat untuk membaca pesan yang menyebabkan kesalahan asal.

35c-country

Kode negara yang mewakili lokasi geografis pemirsa, sebagaimana ditentukan oleh alamat IP pemirsa. Untuk daftar kode negara, lihat [ISO 3166-1 alpha-2](#).

36cs-accept-encoding

Gunakan log akses waktu nyata

Nilai dari Accept-Encoding header di permintaan penampil.

37.**cs-accept**

Nilai dari Accept header di permintaan penampil.

38.**cache-behavior-path-pattern**

Pola jalur yang mengidentifikasi perilaku cache yang sesuai dengan permintaan penampil.

39.**cs-headers**

Header HTTP (nama dan nilai) dalam permintaan penampil.

Note

Field ini dipotong menjadi 800 byte.

40.**cs-header-names**

Nama header HTTP (bukan nilai) pada permintaan penampil.

Note

Field ini dipotong menjadi 800 byte.

41.**cs-headers-count**

Jumlah header HTTP di permintaan penampil.

42.**primary-distribution-id**

Ketika penerapan berkelanjutan diaktifkan, ID ini mengidentifikasi distribusi mana yang utama dalam distribusi saat ini.

43.**primary-distribution-dns-name**

Ketika penerapan berkelanjutan diaktifkan, nilai ini menunjukkan nama domain utama yang terkait dengan CloudFront distribusi saat ini (misalnya, d111111abcdef8.cloudfront.net).

44.**origin-fbl**

Jumlah detik latensi byte pertama antara CloudFront dan asal Anda.

45.**origin-lbl**

Jumlah detik latensi byte terakhir antara CloudFront dan asal Anda.

46.asn

Nomor sistem otonom (ASN) dari pemirsa.

47.



Bidang CMCD dalam log akses real-time

Untuk informasi selengkapnya tentang bidang ini, lihat dokumen [CTA Specification Web Application Video Ecosystem - Common Media Client Data CTA-5004](#).

48.cmcd-encoded-bitrate

Bitrate yang dikodekan dari objek audio atau video yang diminta.

49.cmcd-buffer-length

Panjang buffer dari objek media yang diminta.

50.cmcd-buffer-starvation

Apakah buffer kelaparan di beberapa titik antara permintaan sebelumnya dan permintaan objek. Ini dapat menyebabkan pemain berada dalam stat rebuffering, yang dapat menghentikan pemutaran video atau audio.

51.cmcd-content-id

String unik yang mengidentifikasi konten saat ini.

52.cmcd-object-duration

Durasi pemutaran objek yang diminta (dalam milidetik).

53.cmcd-deadline

Batas waktu dari waktu permintaan bahwa sampel pertama objek ini harus tersedia, sehingga status buffer underrun atau masalah pemutaran lainnya dihindari.

54.cmcd-measured-throughput

Throughput antara klien dan server, yang diukur oleh klien.

55.cmcd-next-object-request

Jalur relatif dari objek yang diminta berikutnya.

56.**cmcd-next-range-request**

Jika permintaan berikutnya adalah permintaan objek sebagian, string ini menunjukkan rentang byte yang akan diminta.

57.**cmcd-object-type**

Jenis media dari objek saat ini yang diminta.

58.**cmcd-playback-rate**

1 jika real-time, 2 jika kecepatan ganda, 0 jika tidak bermain.

59.**cmcd-requested-maximum-throughput**

Throughput maksimum yang diminta yang dianggap klien cukup untuk pengiriman aset.

60.**cmcd-streaming-format**

Format streaming yang menentukan permintaan saat ini.

61.**cmcd-session-id**

GUID yang mengidentifikasi sesi pemutaran saat ini.

62.**cmcd-stream-type**

Token mengidentifikasi ketersediaan segmen. v= semua segmen tersedia. l= segmen menjadi tersedia dari waktu ke waktu.

63.**cmcd-startup**

Kunci disertakan tanpa nilai jika objek dibutuhkan segera selama startup, pencarian, atau pemulihan setelah peristiwa buffer-kosong.

64.**cmcd-top-bitrate**

Rendition bitrate tertinggi yang dapat dimainkan klien.

65.**cmcd-version**

Versi spesifikasi ini digunakan untuk menafsirkan nama kunci dan nilai yang ditentukan. Jika kunci ini dihilangkan, klien dan server harus menafsirkan nilai-nilai yang didefinisikan oleh versi 1.

66.**x-host**

Bidang ini dikirim untuk permintaan asal dan ini menunjukkan domain server asal yang digunakan untuk melayani objek. Jika terjadi kesalahan, Anda dapat menggunakan bidang ini untuk

menemukan asal terakhir yang dicoba, misalnya: `cd8jhdejh6a.mediapackagev2.us-east-1.amazonaws.com`.

67 **sr-reason**

Bidang ini memberikan alasan mengapa asal dipilih. Ini kosong ketika permintaan ke asal primer berhasil.

Jika failover asal terjadi, bidang akan berisi kode kesalahan HTTP yang mengarah ke failover, seperti `Failover:403` atau `Failover:502`. Dalam kasus failover asal, jika permintaan yang dicoba lagi juga gagal dan Anda belum mengonfigurasi halaman kesalahan khusus, maka `x-status` tunjukkan respons asal kedua. Namun, jika Anda telah mengonfigurasi halaman kesalahan khusus bersama dengan failover asal, maka ini akan berisi respons asal kedua jika permintaan gagal dan halaman kesalahan khusus dikembalikan sebagai gantinya.

Jika tidak ada failover asal yang terjadi tetapi pemilihan asal ketahanan sadar kualitas media (MQAR) terjadi, maka ini akan dicatat sebagai `MediaQuality`. Untuk informasi selengkapnya, lihat [Ketahanan sadar kualitas media](#).

68 **x-edge-mqcs**

Bidang ini menunjukkan Skor Keyakinan Kualitas Media (MQCS) (rentang: 0 — 100) untuk segmen media yang CloudFront diambil di header respons CMSD dari v2. `MediaPackage`. Bidang ini tersedia untuk permintaan yang cocok dengan perilaku cache yang memiliki grup asal berkemampuan MQAR. CloudFront mencatat bidang ini untuk segmen media yang juga disajikan dari cache-nya selain permintaan asal. Untuk informasi selengkapnya, lihat [Ketahanan sadar kualitas media](#).

69 **distribution-tenant-id**

ID penyewa distribusi.

70 **connection-id**

Pengenal unik untuk koneksi TLS.

Anda harus mengaktifkan MTL untuk distribusi Anda sebelum Anda bisa mendapatkan informasi untuk bidang ini. Untuk informasi selengkapnya, lihat [Otentikasi TLS timbal balik dengan CloudFront \(Viewer mTLS\)](#).

Titik Akhir (Kinesis Data Streams)

Titik akhir berisi informasi tentang Kinesis Data Streams tempat Anda ingin mengirim log waktu nyata. Anda menyediakan Amazon Resource Name (ARN) dari aliran data.

Untuk informasi selengkapnya tentang membuat Kinesis Data Streams, lihat topik berikut di Panduan Pengembang Amazon Kinesis Data Streams.

- [Membuat dan mengelola aliran](#)
- [Lakukan operasi Kinesis Data Streams dasar menggunakan AWS CLI](#)
- [Membuat aliran](#) (menggunakan AWS SDK untuk Java)

Saat Anda membuat aliran data, Anda perlu menentukan jumlah shard. Gunakan informasi berikut untuk membantu Anda memperkirakan jumlah shard yang Anda butuhkan.

Untuk memperkirakan jumlah shard untuk aliran data Kinesis Anda

1. Hitung (atau perkirakan) jumlah permintaan per detik saat CloudFront yang diterima.

Anda dapat menggunakan [laporan CloudFront penggunaan](#) (di CloudFront konsol) dan [CloudFront metrik](#) (di CloudWatch konsol CloudFront dan Amazon) untuk membantu menghitung permintaan per detik.

2. Tentukan ukuran khas dari catatan log akses real-time tunggal.

Secara umum, satu catatan log adalah sekitar 500 byte. Catatan besar yang mencakup semua kolom yang tersedia biasanya sekitar 1 KB.

Jika Anda tidak yakin ukuran catatan log Anda, Anda dapat mengaktifkan log waktu nyata dengan laju pengambilan sampel rendah (misalnya, 1%), dan kemudian menghitung ukuran rekaman rata-rata menggunakan data pemantauan di Kinesis Data Streams (total byte masuk dibagi dengan jumlah total catatan).

3. Di halaman [harga Amazon Kinesis Data Streams](#) AWS Kalkulator Harga, di bawah, pilih Buat perkiraan kustom Anda sekarang.

- Dalam kalkulator, masukkan jumlah permintaan (catatan) per detik.
- Masukkan ukuran catatan rata-rata dari satu catatan log.
- Pilih Tampilkan perhitungan.

Kalkulator harga menunjukkan jumlah pecahan yang Anda butuhkan dan perkiraan biaya.

IAM Role

Peran AWS Identity and Access Management (IAM) yang memberikan CloudFront izin untuk mengirimkan log akses real-time ke aliran data Kinesis Anda.

Saat membuat konfigurasi log akses real-time dengan CloudFront konsol, Anda dapat memilih Buat peran layanan baru agar konsol membuat peran IAM untuk Anda.

Saat Anda membuat konfigurasi log akses real-time dengan AWS CloudFormation atau CloudFront API (AWS CLI atau SDK), Anda harus membuat peran IAM sendiri dan menyediakan peran ARN. Untuk membuat peran IAM sendiri, gunakan kebijakan berikut.

Kebijakan kepercayaan peran IAM

Untuk menggunakan kebijakan kepercayaan peran IAM berikut, ganti **111122223333** dengan Akun AWS nomor Anda. ConditionElemen dalam kebijakan ini membantu mencegah [masalah wakil yang membingungkan](#) karena hanya CloudFront dapat mengambil peran ini atas nama distribusi di Anda Akun AWS.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudfront.amazonaws.com"
      },
      "Action": "sts:AssumeRole",
      "Condition": {
        "StringEquals": {
          "aws:SourceAccount": "111122223333"
        }
      }
    }
  ]
}
```

```
}
```

Kebijakan izin peran IAM untuk aliran data yang tidak terenkripsi

Untuk menggunakan kebijakan berikut, ganti *arn:aws:kinesis:us-east-2:123456789012:stream/StreamName* dengan ARN aliran data Kinesis Anda.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "kinesis:DescribeStreamSummary",
        "kinesis:DescribeStream",
        "kinesis:PutRecord",
        "kinesis:PutRecords"
      ],
      "Resource": [
        "arn:aws:kinesis:us-east-2:123456789012:stream/StreamName"
      ]
    }
  ]
}
```

Kebijakan izin peran IAM untuk aliran data terenkripsi

Untuk menggunakan kebijakan berikut, ganti *arn:aws:kinesis:us-east-2:123456789012:stream/StreamName* dengan ARN aliran data Kinesis Anda dan dengan *arn:aws:kms:us-east-2:123456789012:key/e58a3d0b-fe4f-4047-a495-ae03cc73d486* ARN Anda. AWS KMS key

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```

    {
      "Effect": "Allow",
      "Action": [
        "kinesis:DescribeStreamSummary",
        "kinesis:DescribeStream",
        "kinesis:PutRecord",
        "kinesis:PutRecords"
      ],
      "Resource": [
        "arn:aws:kinesis:us-east-2:123456789012:stream/StreamName"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "kms:GenerateDataKey"
      ],
      "Resource": [
        "arn:aws:kms:us-east-2:123456789012:key/e58a3d0b-fe4f-4047-a495-ae03cc73d486"
      ]
    }
  ]
}

```

Membuat konsumen Kinesis Data Streams

Untuk membaca dan menganalisis log akses real-time, Anda membuat atau menggunakan konsumen Kinesis Data Streams. Saat Anda membangun konsumen untuk log CloudFront waktu nyata, penting untuk mengetahui bahwa bidang di setiap catatan log akses real-time selalu dikirimkan dalam urutan yang sama, seperti yang tercantum di [Bidang](#) bagian. Pastikan Anda membangun konsumen untuk mengakomodasi pesanan tetap ini.

Misalnya, pertimbangkan konfigurasi log akses real-time yang hanya mencakup tiga bidang ini: `time-to-first-byte`, `sc-status`, dan `c-country`. Dalam skenario ini, kolom terakhir, `c-country`, adalah selalu nomor kolom 3 dalam setiap catatan log. Namun, jika nanti Anda menambahkan bidang ke konfigurasi log akses real-time, penempatan setiap bidang dalam catatan dapat berubah.

Misalnya, jika Anda menambahkan bidang `sc-bytes` dan `time-taken` konfigurasi log akses real-time, bidang ini dimasukkan ke dalam setiap catatan log sesuai dengan urutan yang ditunjukkan di [Bidang](#) bagian. Urutan yang dihasilkan dari semua lima bidang adalah `time-to-first-byte`, `sc-status`, `sc-bytes`, `time-taken`, dan `c-country`. Bidang `c-country` awalnya merupakan bidang nomor 3, tapi sekarang menjadi bidang nomor 5. Pastikan aplikasi konsumen Anda dapat menangani bidang yang mengubah posisi dalam catatan log, jika Anda menambahkan bidang ke konfigurasi log akses real-time Anda.

Memecahkan masalah log akses waktu nyata

Setelah membuat konfigurasi log akses real-time, Anda mungkin menemukan bahwa tidak ada catatan (atau tidak semua rekaman) yang dikirim ke Kinesis Data Streams. Dalam kasus ini, Anda harus memverifikasi terlebih dahulu bahwa CloudFront distribusi menerima permintaan penampil. Jika ya, Anda dapat memeriksa pengaturan berikut untuk melanjutkan pemecahan masalah.

Izin peran IAM

Untuk mengirimkan catatan log akses real-time ke aliran data Kinesis Anda, CloudFront gunakan peran IAM dalam konfigurasi log akses real-time. Pastikan bahwa kebijakan kepercayaan peran dan kebijakan izin peran sesuai dengan kebijakan yang ditunjukkan dalam [IAM Role](#).

Perutean Data Kinesis

Jika CloudFront menulis catatan log akses real-time ke aliran data Kinesis Anda lebih cepat daripada yang dapat ditangani oleh aliran, Kinesis Data Streams mungkin membatasi permintaan dari CloudFront. Dalam hal ini, Anda dapat meningkatkan jumlah pecahan dalam aliran data Kinesis. Setiap shard dapat mendukung penulisan hingga 1.000 catatan per detik, hingga maksimum penulisan data 1 MB per detik.

Log fungsi tepi

[Anda dapat menggunakan Amazon CloudWatch Logs untuk mendapatkan log untuk fungsi edge Anda, baik Lambda @Edge maupun CloudFront Functions.](#) Anda dapat mengakses log dengan menggunakan CloudWatch konsol atau CloudWatch Logs API.

Important

Kami menyarankan Anda menggunakan log untuk memahami sifat permintaan untuk konten Anda, bukan sebagai akuntansi lengkap dari semua permintaan. CloudFront memberikan log fungsi tepi dengan upaya terbaik. Entri log untuk permintaan tertentu mungkin dikirim dalam

waktu lama setelah permintaan diproses secara aktual dan, dalam kasus yang jarang, entri log mungkin tidak dikirimkan sama sekali. Ketika entri log dihilangkan dari log fungsi edge, jumlah entri dalam log fungsi edge tidak akan cocok dengan penggunaan yang muncul dalam laporan AWS penagihan dan penggunaan.

Topik

- [Log Lambda @Edge](#)
- [CloudFront Fungsi log](#)

Log Lambda @Edge

Lambda @Edge secara otomatis mengirim log fungsi ke CloudWatch Log, membuat aliran log di Wilayah AWS tempat fungsi dipanggil. Saat Anda membuat atau memodifikasi fungsi AWS Lambda, Anda dapat menggunakan nama grup CloudWatch log default atau menyesuaikannya.

- Nama grup log default adalah `/aws/lambda/<FunctionName>` di mana `<FunctionName>` adalah nama yang Anda tentukan saat Anda membuat fungsi. Saat mengirim log ke CloudWatch, Lambda @Edge akan secara otomatis menambahkan `us-east-1` awalan ke nama fungsi, sehingga nama grup log adalah `/aws/lambda/us-east-1.<FunctionName>`. Awalan ini sesuai dengan Wilayah AWS tempat fungsi itu dibuat. Awalan ini tetap menjadi bagian dari nama grup log, bahkan di Wilayah lain tempat fungsi dipanggil.
- Jika Anda menentukan nama grup log kustom, seperti `/MyLogGroup`, Lambda @Edge tidak akan menambahkan awalan Region. Nama grup log tetap sama di semua Wilayah lain tempat fungsi dipanggil.

Note

Jika Anda membuat grup log kustom dan menentukan nama yang sama dengan default `/aws/lambda/<FunctionName>`, Lambda @Edge menambahkan `us-east-1` awalan ke nama fungsi.

Selain menyesuaikan nama grup log, fungsi Lambda @Edge mendukung format JSON dan log teks biasa, dan pemfilteran tingkat log. Untuk informasi selengkapnya, lihat [Mengonfigurasi kontrol logging lanjutan untuk fungsi Lambda](#) di Panduan AWS Lambda Pengembang.

 Note

Log throttle Lambda@Edge berdasarkan volume permintaan dan ukuran log.

Anda harus meninjau file CloudWatch log di Wilayah yang benar untuk melihat file log fungsi Lambda @Edge Anda. Untuk melihat Wilayah tempat fungsi Lambda @Edge Anda berjalan, lihat grafik metrik untuk fungsi di konsol. CloudFront Metrik ditampilkan untuk setiap Wilayah . Pada halaman yang sama, Anda dapat memilih Wilayah dan kemudian melihat file log untuk Wilayah tersebut untuk menyelidiki masalah.

Untuk mempelajari lebih lanjut tentang cara menggunakan CloudWatch Log dengan fungsi Lambda @Edge, lihat topik berikut:

- Untuk informasi selengkapnya tentang melihat grafik di bagian Pemantauan CloudFront konsol, lihat [the section called “Pantau CloudFront metrik dengan Amazon CloudWatch”](#).
- Untuk informasi tentang izin yang diperlukan untuk mengirim data ke CloudWatch Log, lihat [the section called “Mengatur izin dan peran IAM”](#).
- Untuk informasi tentang menambahkan logging ke fungsi Lambda @Edge, lihat [AWS Lambda fungsi logging di Node.js](#) atau [AWS Lambda fungsi logging di Python](#) di Panduan Pengembang.AWS Lambda
- Untuk informasi tentang kuota CloudWatch Log (sebelumnya dikenal sebagai batas), lihat [Kuota CloudWatch log di Panduan Pengguna](#) Amazon CloudWatch Logs.

CloudFront Fungsi log

Jika kode CloudFront fungsi berisi `console.log()` pernyataan, CloudFront Fungsi secara otomatis mengirimkan baris log ini ke CloudWatch Log. Jika tidak ada `console.log()` pernyataan, tidak ada yang dikirim ke CloudWatch Log.

CloudFront Fungsi selalu membuat aliran log di Wilayah AS Timur (Virginia N.us-east-1), tidak peduli lokasi tepi mana yang menjalankan fungsi tersebut. Nama aliran log dalam format YYYY/M/D/*UUID*.

Nama grup log menggunakan format berikut:

- Untuk CloudFront Fungsi pada tingkat perilaku cache, formatnya adalah `/aws/cloudfront/function/<FunctionName>`

- Untuk CloudFront Fungsi di tingkat distribusi (Fungsi Koneksi), formatnya adalah `/aws/cloudfront/connection-function/<FunctionName>`

Itu `<FunctionName>` adalah nama yang Anda berikan ke fungsi saat Anda membuatnya.

Example Permintaan penampil

Berikut ini menunjukkan contoh pesan log yang dikirim ke CloudWatch Log. Setiap baris dimulai dengan ID yang secara unik mengidentifikasi permintaan. CloudFront Pesan dimulai dengan START baris yang menyertakan ID CloudFront distribusi, dan diakhiri dengan END garis. Baris log yang dihasilkan pernyataan `console.log()` dalam fungsi berada di antara baris START dan END.

```
U7b4hR_RaxMADupvKAvr8_m9gsGXvioUggLV50yq-vmAtH8HADpjhw== START DistributionID:
E3E5D42GADAXZZ
U7b4hR_RaxMADupvKAvr8_m9gsGXvioUggLV50yq-vmAtH8HADpjhw== Example function log output
U7b4hR_RaxMADupvKAvr8_m9gsGXvioUggLV50yq-vmAtH8HADpjhw== END
```

Example Permintaan koneksi

Berikut ini menunjukkan contoh pesan log yang dikirim ke CloudWatch Log. Setiap baris dimulai dengan ID yang secara unik mengidentifikasi permintaan. CloudFront Pesan dimulai dengan START baris yang menyertakan ID CloudFront distribusi, dan diakhiri dengan END garis. Baris log yang dihasilkan pernyataan `console.log()` dalam fungsi berada di antara baris START dan END.

```
U7b4hR_RaxMADupvKAvr8_m9gsGXvioUggLV50yq-vmAtH8HADpjhw== START DistributionID:
E3E5D42GADA123
U7b4hR_RaxMADupvKAvr8_m9gsGXvioUggLV50yq-vmAtH8HADpjhw== 1.2.3.4
U7b4hR_RaxMADupvKAvr8_m9gsGXvioUggLV50yq-vmAtH8HADpjhw== END
```

Note

CloudFront Fungsi mengirimkan log ke CloudWatch hanya untuk fungsi di LIVE tahap yang berjalan sebagai respons terhadap permintaan dan tanggapan produksi. Saat Anda [menguji suatu fungsi](#), CloudFront tidak mengirim log apa pun ke CloudWatch. Output pengujian berisi informasi tentang kesalahan, pemanfaatan komputasi, dan log fungsi (`console.log()` pernyataan), tetapi informasi ini tidak dikirim ke CloudWatch

CloudFront Fungsi menggunakan [peran terkait layanan AWS Identity and Access Management](#) (IAM) untuk mengirim log ke CloudWatch Log di akun Anda. Peran terkait layanan adalah peran IAM yang ditautkan langsung ke file. Layanan AWS Peran terkait layanan telah ditentukan sebelumnya oleh layanan dan mencakup semua izin yang diperlukan layanan untuk memanggil orang lain untuk Anda. Layanan AWS CloudFront Fungsi menggunakan peran `AWSServiceRoleForCloudFrontLogger` terkait layanan. Untuk informasi selengkapnya tentang peran ini, lihat [the section called “Peran terkait layanan untuk Lambda @Edge”](#) (Lambda@Edge menggunakan peran terkait layanan yang sama).

Ketika fungsi gagal dengan kesalahan validasi atau kesalahan eksekusi, informasi tersebut dicatat dalam log [standar dan log akses real-time](#). Untuk informasi spesifik tentang kesalahan, lihat `x-edge-detailed-result-type` bidang `x-edge-result-type` `x-edge-response-result-type`, dan.

Pencatatan panggilan CloudFront API Amazon menggunakan AWS CloudTrail

CloudFront terintegrasi dengan [AWS CloudTrail](#), layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau Layanan AWS. CloudTrail menangkap semua panggilan API untuk CloudFront sebagai peristiwa. Panggilan yang diambil termasuk panggilan dari CloudFront konsol dan panggilan kode ke operasi CloudFront API. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan yang dibuat CloudFront, alamat IP dari mana permintaan dibuat, kapan dibuat, dan detail tambahan.

Setiap entri peristiwa atau log berisi informasi tentang entitas yang membuat permintaan tersebut. Informasi identitas membantu Anda menentukan berikut hal ini:

- Baik permintaan tersebut dibuat dengan kredensial pengguna root atau pengguna.
- Apakah permintaan dibuat atas nama pengguna IAM Identity Center.
- Apakah permintaan tersebut dibuat dengan kredensial keamanan sementara untuk satu peran atau pengguna gabungan.
- Apakah permintaan tersebut dibuat oleh Layanan AWS lain.

CloudTrail aktif di Akun AWS ketika Anda membuat akun dan Anda secara otomatis memiliki akses ke riwayat CloudTrail Acara. Riwayat CloudTrail Acara menyediakan catatan yang dapat dilihat, dapat dicari, dapat diunduh, dan tidak dapat diubah dari 90 hari terakhir dari peristiwa manajemen yang direkam dalam file. Wilayah AWS Untuk informasi selengkapnya, lihat [Bekerja](#)

[dengan riwayat CloudTrail Acara](#) di Panduan AWS CloudTrail Pengguna. Tidak ada CloudTrail biaya untuk melihat riwayat Acara.

Untuk catatan acara yang sedang berlangsung dalam 90 hari Akun AWS terakhir Anda, buat jejak atau penyimpanan data acara [CloudTrailDanau](#).

CloudTrail jalan setapak

Jejak memungkinkan CloudTrail untuk mengirimkan file log ke bucket Amazon S3. Semua jalur yang dibuat menggunakan Konsol Manajemen AWS Multi-region. Anda dapat membuat jalur Single-region atau Multi-region dengan menggunakan. AWS CLI Membuat jejak Multi-wilayah disarankan karena Anda menangkap aktivitas Wilayah AWS di semua akun Anda. Jika Anda membuat jejak wilayah Tunggal, Anda hanya dapat melihat peristiwa yang dicatat di jejak. Wilayah AWS Untuk informasi selengkapnya tentang jejak, lihat [Membuat jejak untuk Anda Akun AWS](#) dan [Membuat jejak untuk organisasi](#) di Panduan AWS CloudTrail Pengguna.

Anda dapat mengirimkan satu salinan acara manajemen yang sedang berlangsung ke bucket Amazon S3 Anda tanpa biaya CloudTrail dengan membuat jejak, namun, ada biaya penyimpanan Amazon S3. Untuk informasi selengkapnya tentang CloudTrail harga, lihat [AWS CloudTrail Harga](#). Untuk informasi tentang harga Amazon S3, lihat [Harga Amazon S3](#).

CloudTrail Menyimpan data acara danau

CloudTrail Lake memungkinkan Anda menjalankan kueri berbasis SQL pada acara Anda. CloudTrail [Lake mengonversi peristiwa yang ada dalam format JSON berbasis baris ke format Apache ORC](#). ORC adalah format penyimpanan kolumnar yang dioptimalkan untuk pengambilan data dengan cepat. Peristiwa digabungkan ke dalam penyimpanan data peristiwa, yang merupakan kumpulan peristiwa yang tidak dapat diubah berdasarkan kriteria yang Anda pilih dengan menerapkan pemilih acara [tingkat lanjut](#). Penyeleksi yang Anda terapkan ke penyimpanan data acara mengontrol peristiwa mana yang bertahan dan tersedia untuk Anda kueri. Untuk informasi lebih lanjut tentang CloudTrail Danau, lihat [Bekerja dengan AWS CloudTrail Danau](#) di Panduan AWS CloudTrail Pengguna.

CloudTrail Penyimpanan data acara danau dan kueri menimbulkan biaya. Saat Anda membuat penyimpanan data acara, Anda memilih [opsi harga](#) yang ingin Anda gunakan untuk penyimpanan data acara. Opsi penetapan harga menentukan biaya untuk menelan dan menyimpan peristiwa, dan periode retensi default dan maksimum untuk penyimpanan data acara. Untuk informasi selengkapnya tentang CloudTrail harga, lihat [AWS CloudTrail Harga](#).

Note

CloudFront adalah layanan global. CloudTrail merekam peristiwa untuk CloudFront di Wilayah AS Timur (Virginia N.). Untuk informasi selengkapnya, lihat [Acara layanan global](#) di Panduan AWS CloudTrail Pengguna.

Jika Anda menggunakan kredensial keamanan sementara dengan menggunakan AWS Security Token Service, panggilan ke titik akhir regional, seperti `us-west-2`, masuk CloudTrail ke Wilayah yang sesuai.

Untuk informasi selengkapnya tentang CloudFront titik akhir, lihat [CloudFront titik akhir dan kuota](#) di. Referensi Umum AWS

CloudFront peristiwa data di CloudTrail

[Peristiwa data](#) memberikan informasi tentang operasi sumber daya yang dilakukan pada atau di sumber daya (misalnya, membaca atau menulis ke CloudFront distribusi). Ini juga dikenal sebagai operasi bidang data. Peristiwa data sering kali merupakan aktivitas bervolume tinggi. Secara default, CloudTrail tidak mencatat peristiwa data. Riwayat CloudTrail peristiwa tidak merekam peristiwa data.

Biaya tambahan berlaku untuk peristiwa data. Untuk informasi selengkapnya tentang CloudTrail harga, lihat [AWS CloudTrail Harga](#).

Anda dapat mencatat peristiwa data untuk jenis CloudFront sumber daya menggunakan CloudTrail konsol AWS CLI, atau operasi CloudTrail API. Untuk informasi selengkapnya tentang cara mencatat peristiwa data, lihat [Mencatat peristiwa data dengan Konsol Manajemen AWS](#) dan [Mencatat peristiwa data dengan AWS Command Line Interface](#) di Panduan AWS CloudTrail Pengguna.

Tabel berikut mencantumkan jenis CloudFront sumber daya yang dapat Anda log peristiwa data. Kolom tipe peristiwa data (konsol) menunjukkan nilai yang akan dipilih dari daftar tipe peristiwa Data di CloudTrail konsol. Kolom nilai `resources.type` menunjukkan **resources.type** nilai, yang akan Anda tentukan saat mengonfigurasi penyeleksi acara lanjutan menggunakan `or`. AWS CLI CloudTrail APIs CloudTrailKolom Data yang APIs dicatat ke menampilkan panggilan API yang dicatat CloudTrail untuk jenis sumber daya.

Jenis peristiwa data (konsol)	nilai <code>resources.type</code>	Data APIs masuk CloudTrail
CloudFront KeyValueCollection	<code>AWS::CloudFront::KeyValueCollection</code>	DeleteKeys

Jenis peristiwa data (konsol)	nilai resources.type	Data APIs masuk CloudTrail
		• DescribeKeyValueStore • GetKey • ListKeys • PutKeys • UpdateKeys

Anda dapat mengonfigurasi pemilih acara lanjutan untuk memfilter pada `eventNameReadOnly`, dan `resources.ARN` bidang untuk mencatat hanya peristiwa yang penting bagi Anda. Untuk informasi selengkapnya tentang bidang ini, lihat [AdvancedFieldSelector](#) di Referensi AWS CloudTrail API.

CloudFront acara manajemen di CloudTrail

[Acara manajemen](#) memberikan informasi tentang operasi manajemen yang dilakukan pada sumber daya di Akun AWS. Ini juga dikenal sebagai operasi bidang kontrol. Secara default, CloudTrail mencatat peristiwa manajemen.

Amazon CloudFront mencatat semua operasi pesawat CloudFront kontrol sebagai peristiwa manajemen. Untuk daftar operasi bidang CloudFront kontrol Amazon yang CloudFront masuk ke log CloudTrail, lihat [Referensi Amazon CloudFront API](#).

CloudFront contoh acara

Peristiwa mewakili permintaan tunggal dari sumber manapun dan mencakup informasi tentang operasi API yang diminta, tanggal dan waktu operasi, parameter permintaan, dan sebagainya. CloudTrail file log bukanlah jejak tumpukan yang diurutkan dari panggilan API publik, sehingga peristiwa tidak muncul dalam urutan tertentu.

Daftar Isi

- [Contoh: UpdateDistribution](#)
- [Contoh: UpdateKeys](#)

Contoh: UpdateDistribution

Contoh berikut menunjukkan CloudTrail peristiwa yang menunjukkan [UpdateDistribution](#) operasi.

Untuk panggilan ke CloudFront API, eventSource adalah `cloudfront.amazonaws.com`.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE:role-session-name",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/role-session-name",
    "accountId": "111122223333",
    "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/Admin",
        "accountId": "111122223333",
        "userName": "Admin"
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2024-02-02T19:23:50Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2024-02-02T19:26:01Z",
  "eventSource": "cloudfront.amazonaws.com",
  "eventName": "UpdateDistribution",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "52.94.133.137",
  "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/121.0.0.0 Safari/537.36",
  "requestParameters": {
    "distributionConfig": {
      "defaultRootObject": "",
      "aliases": {
        "quantity": 3,
        "items": [
          "alejandro_rosalez.awsps.myinstance.com",
          "cross-testing.alejandro_rosalez.awsps.myinstance.com",
          "*.alejandro_rosalez.awsps.myinstance.com"
        ]
      }
    },
    "cacheBehaviors": {
```

```
        "quantity": 0,
        "items": []
    },
    "httpVersion": "http2and3",
    "originGroups": {
        "quantity": 0,
        "items": []
    },
    "viewerCertificate": {
        "minimumProtocolVersion": "TLSv1.2_2021",
        "cloudFrontDefaultCertificate": false,
        "aCMCertificateArn": "arn:aws:acm:us-east-1:111122223333:certificate/
a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "sLSupportMethod": "sni-only"
    },
    "webACLId": "arn:aws:wafv2:us-east-1:111122223333:global/webacl/testing-
acl/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
    "customErrorResponses": {
        "quantity": 0,
        "items": []
    },
    "logging": {
        "includeCookies": false,
        "prefix": "",
        "enabled": false,
        "bucket": ""
    },
    "priceClass": "PriceClass_All",
    "restrictions": {
        "geoRestriction": {
            "restrictionType": "none",
            "quantity": 0,
            "items": []
        }
    },
    "isIPv6Enabled": true,
    "callerReference": "1578329170895",
    "continuousDeploymentPolicyId": "",
    "enabled": true,
    "defaultCacheBehavior": {
        "targetOriginId": "d1111111abcdef8",
        "minTTL": 0,
        "compress": false,
        "maxTTL": 31536000,
```



```
"functionAssociations": {
  "quantity": 0,
  "items": []
},
"trustedKeyGroups": {
  "quantity": 0,
  "items": [],
  "enabled": false
},
"smoothStreaming": false,
"fieldLevelEncryptionId": "",
"defaultTTL": 86400,
"lambdaFunctionAssociations": {
  "quantity": 0,
  "items": []
},
"viewerProtocolPolicy": "redirect-to-https",
"forwardedValues": {
  "cookies": {"forward": "none"},
  "queryStringCacheKeys": {
    "quantity": 0,
    "items": []
  },
  "queryString": false,
  "headers": {
    "quantity": 1,
    "items": ["*"]
  }
},
"trustedSigners": {
  "items": [],
  "enabled": false,
  "quantity": 0
},
"allowedMethods": {
  "quantity": 2,
  "items": [
    "HEAD",
    "GET"
  ],
  "cachedMethods": {
    "quantity": 2,
    "items": [
      "HEAD",
```

```

        "GET"
      ]
    }
  },
  "staging": false,
  "origins": {
    "quantity": 1,
    "items": [
      {
        "originPath": "",
        "connectionTimeout": 10,
        "customOriginConfig": {
          "originReadTimeout": 30,
          "httpSPort": 443,
          "originProtocolPolicy": "https-only",
          "originKeepaliveTimeout": 5,
          "httpSPort": 80,
          "originSslProtocols": {
            "quantity": 3,
            "items": [
              "TLSv1",
              "TLSv1.1",
              "TLSv1.2"
            ]
          }
        },
        "id": "d111111abcdef8",
        "domainName": "d111111abcdef8.cloudfront.net",
        "connectionAttempts": 3,
        "customHeaders": {
          "quantity": 0,
          "items": []
        },
        "originShield": {"enabled": false},
        "originAccessControlId": ""
      }
    ]
  },
  "comment": "HIDDEN_DUE_TO_SECURITY_REASONS"
},
"id": "EDFDVBD6EXAMPLE",
"ifMatch": "E1RTLUR9YES760"
},

```

```
"responseElements": {
  "distribution": {
    "activeTrustedSigners": {
      "quantity": 0,
      "enabled": false
    },
    "id": "EDFDVBD6EXAMPLE",
    "domainName": "d111111abcdef8.cloudfront.net",
    "distributionConfig": {
      "defaultRootObject": "",
      "aliases": {
        "quantity": 3,
        "items": [
          "alejandro_rosalez.awsps.myinstance.com",
          "cross-testing.alejandro_rosalez.awsps.myinstance.com",
          "*.alejandro_rosalez.awsps.myinstance.com"
        ]
      },
      "cacheBehaviors": {"quantity": 0},
      "httpVersion": "http2and3",
      "originGroups": {"quantity": 0},
      "viewerCertificate": {
        "minimumProtocolVersion": "TLSv1.2_2021",
        "cloudFrontDefaultCertificate": false,
        "aCMCertificateArn": "arn:aws:acm:us-east-1:111122223333:certificate/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
        "sLSupportMethod": "sni-only",
        "certificateSource": "acm",
        "certificate": "arn:aws:acm:us-east-1:111122223333:certificate/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
      },
      "webACLId": "arn:aws:wafv2:us-east-1:111122223333:global/webacl/testing-acl/a1b2c3d4-5678-90ab-cdef-EXAMPLE22222",
      "customErrorResponses": {"quantity": 0},
      "logging": {
        "includeCookies": false,
        "prefix": "",
        "enabled": false,
        "bucket": ""
      },
      "priceClass": "PriceClass_All",
      "restrictions": {
        "geoRestriction": {
          "restrictionType": "none",
```

```
        "quantity": 0
      }
    },
    "isIPv6Enabled": true,
    "callerReference": "1578329170895",
    "continuousDeploymentPolicyId": "",
    "enabled": true,
    "defaultCacheBehavior": {
      "targetOriginId": "d1111111abcdef8",
      "minTTL": 0,
      "compress": false,
      "maxTTL": 31536000,
      "functionAssociations": {"quantity": 0},
      "trustedKeyGroups": {
        "quantity": 0,
        "enabled": false
      },
    },
    "smoothStreaming": false,
    "fieldLevelEncryptionId": "",
    "defaultTTL": 86400,
    "lambdaFunctionAssociations": {"quantity": 0},
    "viewerProtocolPolicy": "redirect-to-https",
    "forwardedValues": {
      "cookies": {"forward": "none"},
      "queryStringCacheKeys": {"quantity": 0},
      "queryString": false,
      "headers": {
        "quantity": 1,
        "items": ["*"]
      }
    },
    "trustedSigners": {
      "enabled": false,
      "quantity": 0
    },
    "allowedMethods": {
      "quantity": 2,
      "items": [
        "HEAD",
        "GET"
      ],
      "cachedMethods": {
        "quantity": 2,
        "items": [
```

```

        "HEAD",
        "GET"
    ]
}
},
"staging": false,
"origins": {
    "quantity": 1,
    "items": [
        {
            "originPath": "",
            "connectionTimeout": 10,
            "customOriginConfig": {
                "originReadTimeout": 30,
                "hTTPSPort": 443,
                "originProtocolPolicy": "https-only",
                "originKeepaliveTimeout": 5,
                "hTTPPort": 80,
                "originSslProtocols": {
                    "quantity": 3,
                    "items": [
                        "TLSv1",
                        "TLSv1.1",
                        "TLSv1.2"
                    ]
                }
            },
            "id": "d111111abcdef8",
            "domainName": "d111111abcdef8.cloudfront.net",
            "connectionAttempts": 3,
            "customHeaders": {"quantity": 0},
            "originShield": {"enabled": false},
            "originAccessControlId": ""
        }
    ]
},
"comment": "HIDDEN_DUE_TO_SECURITY_REASONS"
},
"aliasICPRecordals": [
    {
        "cNAME": "alejandro_rosalez.awsps.myinstance.com",
        "iCPRecordalStatus": "APPROVED"
    },

```

```

        {
            "cNAME": "cross-testing.alejandro_rosalez.awsps.myinstance.com",
            "iCPRecordalStatus": "APPROVED"
        },
        {
            "cNAME": "/*.alejandro_rosalez.awsps.myinstance.com",
            "iCPRecordalStatus": "APPROVED"
        }
    ],
    "aRN": "arn:aws:cloudfront::111122223333:distribution/EDFDVBD6EXAMPLE",
    "status": "InProgress",
    "lastModifiedTime": "Feb 2, 2024 7:26:01 PM",
    "activeTrustedKeyGroups": {
        "enabled": false,
        "quantity": 0
    },
    "InProgressInvalidationBatches": 0
},
"eTag": "E1YHBLAB2BJY1G"
},
"requestID": "4e6b66f9-d548-11e3-a8a9-73e33example",
"eventID": "5ab02562-0fc5-43d0-b7b6-90293example",
"readOnly": false,
"eventType": "AwsApiCall",
"apiVersion": "2020_05_31",
"managementEvent": true,
"recipientAccountId": "111122223333",
"eventCategory": "Management",
"tlsDetails": {
    "tlsVersion": "TLSv1.3",
    "cipherSuite": "TLS_AES_128_GCM_SHA256",
    "clientProvidedHostHeader": "cloudfront.amazonaws.com"
},
"sessionCredentialFromConsole": "true"
}

```

Contoh: UpdateKeys

Contoh berikut menunjukkan CloudTrail peristiwa yang menunjukkan [UpdateKeys](#) operasi.

Untuk panggilan ke CloudFront KeyValueCollection API, eventSource itu `edgekeyvaluestore.amazonaws.com` bukan `cloudfront.amazonaws.com`.

```
{
  "eventVersion": "1.09",
  "userIdentity": {
    "type": "AssumedRole",
    "principalId": "AIDACKCEVSQ6C2EXAMPLE:role-session-name",
    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/role-session-name",
    "accountId": "111122223333",
    "accessKeyId": "ASIAIOSFODNN7EXAMPLE",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "AIDACKCEVSQ6C2EXAMPLE",
        "arn": "arn:aws:iam::111122223333:role/Admin",
        "accountId": "111122223333",
        "userName": "Admin"
      },
      "attributes": {
        "creationDate": "2023-11-01T23:41:14Z",
        "mfaAuthenticated": "false"
      }
    }
  },
  "eventTime": "2023-11-01T23:41:28Z",
  "eventSource": "edgekeyvaluestore.amazonaws.com",
  "eventName": "UpdateKeys",
  "awsRegion": "us-east-1",
  "sourceIPAddress": "3.235.183.252",
  "userAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.0.0 Safari/537.36",
  "requestParameters": {
    "kvsARN": "arn:aws:cloudfront::111122223333:key-value-store/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111",
    "ifMatch": "KV306B1CX531EBP",
    "deletes": [
      {"key": "key1"}
    ]
  },
  "responseElements": {
    "itemCount": 0,
    "totalSizeInBytes": 0,
    "eTag": "KVDC9VEVZ71ZG0"
  },
  "requestID": "5ccf104c-acce-4ea1-b7fc-73e33example",
}
```

```
{
  "eventID": "a0b1b5c7-906c-439d-9925-90293example",
  "readOnly": false,
  "resources": [
    {
      "accountId": "111122223333",
      "type": "AWS::CloudFront::KeyValueStore",
      "ARN": "arn:aws:cloudfront::111122223333:key-value-store/a1b2c3d4-5678-90ab-cdef-EXAMPLE11111"
    }
  ],
  "eventType": "AwsApiCall",
  "managementEvent": false,
  "recipientAccountId": "111122223333",
  "eventCategory": "Data",
  "tlsDetails": {
    "tlsVersion": "TLSv1.3",
    "cipherSuite": "TLS_AES_128_GCM_SHA256",
    "clientProvidedHostHeader": "111122223333.cloudfront-kvs.global.api.aws"
  }
}
```

Untuk informasi tentang konten CloudTrail rekaman, lihat [konten CloudTrail rekaman](#) di Panduan AWS CloudTrail Pengguna.

Lacak perubahan konfigurasi dengan AWS Config

Untuk merekam dan mengevaluasi konfigurasi sumber AWS daya Anda, Anda dapat menggunakan AWS Config, yang memberi Anda tampilan terperinci tentang konfigurasi distribusi Anda. Ini termasuk bagaimana sumber daya terkait satu sama lain dan bagaimana mereka dikonfigurasi di masa lalu, sehingga Anda dapat meninjau perubahan dari waktu ke waktu.

Anda juga dapat menggunakan AWS Config untuk merekam perubahan konfigurasi pada pengaturan CloudFront distribusi Anda. Anda dapat menangkap perubahan pada status distribusi, kelas harga, asal, pengaturan pembatasan geografis, dan konfigurasi Lambda @Edge.

Note

AWS Config tidak merekam tag nilai kunci untuk distribusi CloudFront streaming.

Daftar Isi

- [Mengatur AWS Config dengan CloudFront](#)
- [Lihat riwayat CloudFront konfigurasi](#)
- [CloudFront Evaluasi konfigurasi dengan Aturan AWS Config](#)

Mengatur AWS Config dengan CloudFront

Saat menyiapkan AWS Config, Anda dapat memilih untuk merekam semua AWS sumber daya yang didukung atau hanya merekam beberapa sumber daya tertentu, seperti merekam perubahan CloudFront hanya untuk. Untuk daftar CloudFront sumber daya yang didukung, lihat CloudFront bagian [Amazon](#) dari topik Jenis Sumber Daya yang Didukung di Panduan AWS Config Pengembang.

Catatan

- Untuk melacak perubahan konfigurasi pada CloudFront distribusi Anda, Anda harus masuk ke CloudFront konsol di AS Timur (Virginia Utara) Wilayah AWS.
- Mungkin ada keterlambatan dalam merekam sumber daya dengan AWS Config. AWS Config merekam sumber daya hanya setelah menemukan sumber daya.

Console

Untuk mengatur AWS Config dengan CloudFront

1. Masuk ke Konsol Manajemen AWS dan buka [AWS Config konsol](#).
2. Pilih Mulai Sekarang.
3. Pada halaman Pengaturan, untuk jenis sumber daya yang akan direkam, tentukan jenis AWS sumber daya yang AWS Config ingin Anda rekam. Jika Anda hanya ingin merekam CloudFront perubahan, pilih Jenis tertentu, lalu, di bawah CloudFront, pilih distribusi atau distribusi streaming yang ingin Anda lacak perubahannya.

Untuk menambahkan atau mengubah distribusi yang ingin dilacak, pilih Pengaturan di sebelah kiri, setelah menyelesaikan pengaturan awal Anda.

4. Tentukan opsi tambahan yang diperlukan untuk AWS Config: mengatur pemberitahuan, menentukan lokasi untuk informasi konfigurasi, dan menambahkan aturan untuk mengevaluasi jenis sumber daya.

Untuk informasi selengkapnya, lihat [Menyiapkan AWS Config dengan Konsol](#) di Panduan AWS Config Pengembang.

AWS CLI

Untuk mengatur AWS Config dengan CloudFront menggunakan AWS CLI, lihat [Menyiapkan AWS Config dengan AWS CLI di Panduan AWS Config](#) Pengembang.

AWS Config API

Untuk mengatur AWS Config CloudFront penggunaan AWS Config API, lihat operasi [StartConfigurationRecorder](#) API di Referensi AWS Config API.

Lihat riwayat CloudFront konfigurasi

Setelah AWS Config mulai merekam perubahan konfigurasi pada distribusi Anda, Anda bisa mendapatkan riwayat konfigurasi distribusi apa pun yang telah Anda konfigurasi CloudFront.

Anda dapat melihat riwayat konfigurasi dengan cara berikut.

Console

Untuk setiap sumber daya yang direkam, Anda dapat melihat halaman timeline yang menyediakan riwayat detail konfigurasi. Untuk melihat halaman ini, pilih ikon abu-abu di kolom Konfigurasi Kronologi pada halaman Host Khusus.

Untuk informasi selengkapnya, lihat [Melihat Detail Konfigurasi di AWS Config Konsol](#) di Panduan AWS Config Pengembang.

AWS CLI

Untuk mendapatkan daftar semua distribusi Anda, jalankan [list-discovered-resources](#) perintah, seperti yang ditunjukkan pada contoh berikut.

```
aws configservice list-discovered-resources --resource-type
AWS::CloudFront::Distribution
```

Untuk mendapatkan detail konfigurasi distribusi untuk interval waktu tertentu, jalankan [get-resource-config-history](#) perintah.

Untuk informasi selengkapnya, lihat [Melihat Detail Konfigurasi Menggunakan CLI](#) di Panduan Developer AWS Config .

AWS Config API

Untuk mendapatkan daftar semua distribusi Anda, gunakan operasi [ListDiscoveredResources](#) API.

Untuk mendapatkan detail konfigurasi distribusi untuk interval waktu tertentu, gunakan operasi [GetResourceConfigHistory](#) API. Untuk informasi lebih lanjut, lihat [Referensi API AWS Config](#).

CloudFront Evaluasi konfigurasi dengan Aturan AWS Config

Anda dapat mengevaluasi konfigurasi terhadap konfigurasi yang diinginkan dengan AWS Config Aturan. Misalnya, AWS Config Aturan membantu Anda mengevaluasi apakah CloudFront sumber daya Anda mematuhi praktik terbaik keamanan umum. Anda dapat memilih aturan terkelola seperti kebijakan penampil HTTPS, SNI enabled, OAC enabled, origin failover enabled, AWS WAF WebACL, atau kebijakan AWS Shield Advanced resource yang akan dipicu saat konfigurasi berubah.

Aturan terkelola dapat menjalankan evaluasi secara berkala, pada frekuensi yang Anda pilih. AWS Firewall Manager bergantung pada AWS Config peringatan dan perbaikan otomatis. Untuk informasi selengkapnya, lihat [Mengevaluasi Sumber Daya dengan AWS Config Aturan](#) dan [Daftar Aturan AWS Config Terkelola](#) di Panduan AWS Config Pengembang.

Keamanan di Amazon CloudFront

Keamanan cloud di AWS adalah prioritas tertinggi. Sebagai AWS pelanggan, Anda mendapat manfaat dari pusat data dan arsitektur jaringan yang dibangun untuk memenuhi persyaratan organisasi yang paling sensitif terhadap keamanan.

Keamanan adalah tanggung jawab bersama antara Anda AWS dan Anda. [Model tanggung jawab bersama](#) menggambarkan hal ini sebagai keamanan dari cloud dan keamanan di cloud:

- Keamanan cloud — AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan AWS layanan di AWS Cloud. AWS juga memberi Anda layanan yang dapat Anda gunakan dengan aman. Auditor pihak ketiga secara berkala menguji dan memverifikasi efektivitas keamanan kami sebagai bagian dari [Program kepatuhan AWS](#). Untuk mempelajari tentang program kepatuhan yang berlaku untuk Amazon CloudFront, lihat [AWS Layanan dalam Lingkup berdasarkan Program Kepatuhan](#).
- Keamanan di cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan. Anda juga bertanggung jawab atas faktor-faktor lain termasuk sensitivitas data, kebutuhan organisasi, serta undang-undang dan peraturan yang berlaku.

Dokumentasi ini membantu Anda memahami cara menerapkan model tanggung jawab bersama saat menggunakan CloudFront. Topik berikut menunjukkan cara mengonfigurasi CloudFront untuk memenuhi tujuan keamanan dan kepatuhan Anda. Anda juga belajar cara menggunakan AWS layanan lain yang membantu Anda memantau dan mengamankan CloudFront sumber daya Anda.

Topik

- [Perlindungan data di Amazon CloudFront](#)
- [Identity and Access Management untuk Amazon CloudFront](#)
- [Pencatatan dan pemantauan di Amazon CloudFront](#)
- [Validasi kepatuhan untuk Amazon CloudFront](#)
- [Ketahanan di Amazon CloudFront](#)
- [Keamanan infrastruktur di Amazon CloudFront](#)

Perlindungan data di Amazon CloudFront

[Model tanggung jawab AWS bersama model](#) berlaku untuk perlindungan data di Amazon CloudFront. Seperti yang dijelaskan dalam model AWS ini, bertanggung jawab untuk melindungi infrastruktur global yang menjalankan semua AWS Cloud. Anda bertanggung jawab untuk mempertahankan kendali atas konten yang di-host pada infrastruktur ini. Anda juga bertanggung jawab atas tugas-tugas konfigurasi dan manajemen keamanan untuk Layanan AWS yang Anda gunakan. Untuk informasi selengkapnya tentang privasi data, lihat [FAQ Privasi Data AWS](#). Untuk informasi tentang perlindungan data di Eropa, lihat [Pusat Peraturan Umum Perlindungan Data \(GDPR\)](#).

Untuk tujuan perlindungan data, kami menyarankan Anda melindungi Akun AWS kredensial dan mengatur pengguna individu dengan AWS IAM Identity Center atau AWS Identity and Access Management (IAM). Dengan cara itu, setiap pengguna hanya diberi izin yang diperlukan untuk memenuhi tanggung jawab tugasnya. Kami juga menyarankan supaya Anda mengamankan data dengan cara-cara berikut:

- Gunakan autentikasi multi-faktor (MFA) pada setiap akun.
- Gunakan SSL/TLS untuk berkomunikasi dengan AWS sumber daya. Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Siapkan API dan pencatatan aktivitas pengguna dengan AWS CloudTrail. Untuk informasi tentang penggunaan CloudTrail jejak untuk menangkap AWS aktivitas, lihat [Bekerja dengan CloudTrail jejak](#) di AWS CloudTrail Panduan Pengguna.
- Gunakan solusi AWS enkripsi, bersama dengan semua kontrol keamanan default di dalamnya Layanan AWS.
- Gunakan layanan keamanan terkelola tingkat lanjut seperti Amazon Macie, yang membantu menemukan dan mengamankan data sensitif yang disimpan di Amazon S3.
- Jika Anda memerlukan modul kriptografi tervalidasi FIPS 140-3 saat mengakses AWS melalui antarmuka baris perintah atau API, gunakan titik akhir FIPS. Lihat informasi selengkapnya tentang titik akhir FIPS yang tersedia di [Standar Pemrosesan Informasi Federal \(FIPS\) 140-3](#).

Kami sangat merekomendasikan agar Anda tidak pernah memasukkan informasi identifikasi yang sensitif, seperti nomor rekening pelanggan Anda, ke dalam tanda atau bidang isian bebas seperti bidang Nama. Ini termasuk saat Anda bekerja dengan CloudFront atau lainnya Layanan AWS menggunakan konsol, API AWS CLI, atau AWS SDK. Data apa pun yang Anda masukkan ke dalam tanda atau bidang isian bebas yang digunakan untuk nama dapat digunakan untuk log penagihan atau log diagnostik. Saat Anda memberikan URL ke server eksternal, kami sangat menganjurkan

supaya Anda tidak menyertakan informasi kredensial di dalam URL untuk memvalidasi permintaan Anda ke server itu.

Amazon CloudFront menyediakan beberapa opsi yang dapat Anda gunakan untuk membantu mengamankan konten yang dikirimkannya:

- Konfigurasi koneksi HTTPS.
- Konfigurasi enkripsi tingkat bidang untuk menyediakan keamanan tambahan untuk data tertentu selama transit.
- Batasi akses ke konten agar hanya orang tertentu, atau orang di area tertentu, yang bisa melihatnya.

Topik berikut menjelaskan opsi secara lebih terperinci.

Topik

- [Enkripsi saat bergerak](#)
- [Enkripsi saat diam](#)
- [Batasi Akses ke Konten](#)

Enkripsi saat bergerak

Untuk mengenkripsi data Anda selama transit, Anda mengonfigurasi Amazon CloudFront agar pemirsa menggunakan HTTPS untuk meminta file Anda, sehingga koneksi dienkripsi saat CloudFront berkomunikasi dengan pemirsa. Anda juga dapat mengonfigurasi CloudFront untuk menggunakan HTTPS untuk mendapatkan file dari asal Anda, sehingga koneksi dienkripsi saat CloudFront berkomunikasi dengan asal Anda.

Untuk informasi selengkapnya, lihat [Gunakan HTTPS dengan CloudFront](#).

Field-level enkripsi menambahkan lapisan keamanan tambahan bersama dengan HTTPS yang memungkinkan Anda melindungi data tertentu selama pemrosesan sistem sehingga hanya aplikasi tertentu yang dapat melihatnya. Dengan mengonfigurasi enkripsi tingkat lapangan CloudFront, Anda dapat mengunggah informasi sensitif yang dikirimkan pengguna dengan aman ke server web Anda. Informasi sensitif yang diberikan oleh klien Anda dienkripsi pada tepi yang lebih dekat dengan pengguna. Data tersebut tetap terenkripsi di seluruh aplikasi Anda, memastikan bahwa hanya aplikasi yang memerlukan data—dan memiliki kredensial untuk mendekripsinya—bisa melakukannya.

Untuk informasi selengkapnya, lihat [Gunakan enkripsi tingkat lapangan untuk membantu melindungi data sensitif](#).

Titik akhir CloudFront API, `cloudfront.amazonaws.com` dan `cloudfront-fips.amazonaws.com`, hanya menerima lalu lintas HTTPS. Ini berarti bahwa ketika Anda mengirim dan menerima informasi menggunakan CloudFront API, data Anda—termasuk konfigurasi distribusi, kebijakan cache dan kebijakan permintaan asal, grup kunci dan kunci publik, dan kode fungsi dalam CloudFront fungsi—selalu dienkripsi saat transit. Selain itu, semua permintaan yang dikirim ke titik akhir CloudFront API ditandatangani dengan AWS kredensial dan masuk. AWS CloudTrail

Kode fungsi dan konfigurasi dalam CloudFront Fungsi selalu dienkripsi saat transit saat disalin ke titik keberadaan lokasi tepi (POP), dan di antara lokasi penyimpanan lain yang digunakan oleh CloudFront

Enkripsi saat diam

Kode fungsi dan konfigurasi dalam CloudFront Fungsi selalu disimpan dalam format terenkripsi pada POP lokasi tepi, dan di lokasi penyimpanan lain yang digunakan oleh CloudFront

Batasi Akses ke Konten

Banyak perusahaan yang mendistribusikan konten melalui internet ingin membatasi akses ke dokumen, data bisnis, aliran media, atau konten yang dimaksudkan untuk subset pengguna. Untuk menyajikan konten ini dengan aman menggunakan Amazon CloudFront, Anda dapat melakukan satu atau beberapa hal berikut:

Menggunakan URL atau cookie bertanda tangan

Anda dapat membatasi akses ke konten yang ditujukan untuk pengguna terpilih—misalnya, pengguna yang telah membayar biaya—dengan menyajikan konten pribadi ini melalui CloudFront penggunaan URL yang ditandatangani atau cookie yang ditandatangani. Untuk informasi selengkapnya, lihat [Sajikan konten pribadi dengan URL yang ditandatangani dan cookie yang ditandatangani](#).

Batasi akses ke konten dalam bucket Amazon S3

Jika Anda membatasi akses ke konten Anda dengan menggunakan, misalnya, URL yang CloudFront ditandatangani atau cookie yang ditandatangani, Anda juga tidak ingin orang melihat file dengan menggunakan URL langsung untuk file tersebut. Sebagai gantinya, Anda ingin mereka mengakses file hanya dengan menggunakan CloudFront URL, sehingga perlindungan Anda berfungsi.

Jika Anda menggunakan bucket Amazon S3 sebagai asal untuk CloudFront distribusi, Anda dapat mengatur kontrol akses asal (OAC) yang memungkinkan untuk membatasi akses ke bucket S3.

Untuk informasi selengkapnya, lihat [the section called “Batasi akses ke asal Amazon S3”](#).

Membatasi akses ke konten yang disajikan oleh Penyeimbang Beban Aplikasi (Application Load Balancer)

Bila Anda menggunakan CloudFront Application Load Balancer di Elastic Load Balancing sebagai asal, Anda dapat CloudFront mengonfigurasi untuk mencegah pengguna mengakses Application Load Balancer secara langsung. Hal ini memungkinkan pengguna untuk mengakses Application Load Balancer hanya melalui CloudFront, memastikan bahwa Anda mendapatkan manfaat menggunakan CloudFront. Untuk informasi selengkapnya, lihat [Batasi akses ke Application Load Balancers](#).

Gunakan ACL AWS WAF web

Anda dapat menggunakan AWS WAF, layanan firewall aplikasi web, untuk membuat daftar kontrol akses web (web ACL) untuk membatasi akses ke konten Anda. Berdasarkan kondisi yang Anda tentukan, seperti alamat IP tempat permintaan berasal atau nilai string kueri, CloudFront merespons permintaan baik dengan konten yang diminta atau dengan kode status HTTP 403 (Terlarang). Untuk informasi selengkapnya, lihat [Gunakan AWS WAF perlindungan](#).

Gunakan pembatasan geo

Anda bisa menggunakan pembatasan geo, juga dikenal sebagai pemblokiran geo, untuk mencegah pengguna di lokasi geografis tertentu agar tidak mengakses konten yang Anda layani melalui distribusi CloudFront. Ada beberapa opsi yang bisa dipilih saat Anda mengonfigurasi pembatasan geografis. Untuk informasi selengkapnya, lihat [Batasi distribusi geografis konten Anda](#).

Identity and Access Management untuk Amazon CloudFront

AWS Identity and Access Management (IAM) adalah Layanan AWS yang membantu administrator mengontrol akses ke AWS sumber daya dengan aman. Administrator IAM mengontrol siapa yang dapat diautentikasi (masuk) dan diberi wewenang (memiliki izin) untuk menggunakan sumber daya. CloudFront IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

Topik

- [Audiens](#)
- [Mengautentikasi dengan identitas](#)
- [Mengelola akses menggunakan kebijakan](#)
- [Bagaimana Amazon CloudFront bekerja dengan IAM](#)
- [Contoh kebijakan berbasis identitas untuk Amazon CloudFront](#)
- [AWS kebijakan terkelola untuk Amazon CloudFront](#)
- [Gunakan peran terkait layanan untuk CloudFront](#)
- [Memecahkan masalah CloudFront identitas dan akses Amazon](#)

Audiens

Cara Anda menggunakan AWS Identity and Access Management (IAM) berbeda berdasarkan peran Anda:

- Pengguna layanan - minta izin dari administrator Anda jika Anda tidak dapat mengakses fitur (lihat [Memecahkan masalah CloudFront identitas dan akses Amazon](#))
- Administrator layanan - tentukan akses pengguna dan mengirimkan permintaan izin (lihat [Bagaimana Amazon CloudFront bekerja dengan IAM](#))
- Administrator IAM - tulis kebijakan untuk mengelola akses (lihat [Contoh kebijakan berbasis identitas untuk Amazon CloudFront](#))

Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensi identitas Anda. Anda harus diautentikasi sebagai Pengguna root akun AWS, pengguna IAM, atau dengan mengasumsikan peran IAM.

Anda dapat masuk sebagai identitas federasi menggunakan kredensial dari sumber identitas seperti AWS IAM Identity Center (Pusat Identitas IAM), autentikasi masuk tunggal, atau kredensial. Google/Facebook Untuk informasi selengkapnya tentang cara masuk, lihat [Cara masuk ke Akun AWS Anda](#) dalam Panduan Pengguna AWS Sign-In .

Untuk akses terprogram, AWS sediakan SDK dan CLI untuk menandatangani permintaan secara kriptografis. Untuk informasi selengkapnya, lihat [AWS Signature Version 4 untuk permintaan API](#) dalam Panduan Pengguna IAM.

Akun AWS pengguna root

Saat Anda membuat Akun AWS, Anda mulai dengan satu identitas masuk yang disebut pengguna Akun AWS root yang memiliki akses lengkap ke semua Layanan AWS dan sumber daya. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari. Untuk tugas yang memerlukan kredensial pengguna root, lihat [Tugas yang memerlukan kredensial pengguna root](#) dalam Panduan Pengguna IAM.

Identitas terfederasi

Sebagai praktik terbaik, mewajibkan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses Layanan AWS menggunakan kredensi sementara.

Identitas federasi adalah pengguna dari direktori perusahaan Anda, penyedia identitas web, atau Directory Service yang mengakses Layanan AWS menggunakan kredensi dari sumber identitas. Identitas terfederasi mengambil peran yang memberikan kredensial sementara.

Untuk manajemen akses terpusat, kami menyarankan AWS IAM Identity Center. Untuk informasi selengkapnya, lihat [Apa itu Pusat Identitas IAM?](#) dalam Panduan Pengguna AWS IAM Identity Center

Pengguna dan grup IAM

[Pengguna IAM](#) adalah identitas dengan izin khusus untuk satu orang atau aplikasi. Sebaiknya gunakan kredensial sementara alih-alih pengguna IAM dengan kredensial jangka panjang. Untuk informasi selengkapnya, lihat [Mewajibkan pengguna manusia untuk menggunakan federasi dengan penyedia identitas untuk mengakses AWS menggunakan kredensi sementara](#) di Panduan Pengguna IAM.

[Grup IAM](#) menentukan kumpulan pengguna IAM dan mempermudah pengelolaan izin untuk pengguna dalam jumlah besar. Untuk mempelajari selengkapnya, lihat [Kasus penggunaan untuk pengguna IAM](#) dalam Panduan Pengguna IAM.

Peran IAM

[Peran IAM](#) adalah identitas dengan izin khusus yang menyediakan kredensial sementara. Anda dapat mengambil peran dengan [beralih dari pengguna ke peran IAM \(konsol\)](#) atau dengan memanggil operasi AWS CLI atau AWS API. Untuk informasi selengkapnya, lihat [Metode untuk mengambil peran](#) dalam Panduan Pengguna IAM.

Peran IAM berguna untuk akses pengguna terfederasi, izin pengguna IAM sementara, akses lintas akun, akses lintas layanan, dan aplikasi yang berjalan di Amazon EC2. Untuk informasi selengkapnya, lihat [Akses sumber daya lintas akun di IAM](#) dalam Panduan Pengguna IAM.

Mengelola akses menggunakan kebijakan

Anda mengontrol akses AWS dengan membuat kebijakan dan melampirkannya ke AWS identitas atau sumber daya. Kebijakan menentukan izin saat dikaitkan dengan identitas atau sumber daya. AWS mengevaluasi kebijakan ini ketika kepala sekolah membuat permintaan. Sebagian besar kebijakan disimpan AWS sebagai dokumen JSON. Untuk informasi selengkapnya tentang dokumen kebijakan JSON, lihat [Gambaran umum kebijakan JSON](#) dalam Panduan Pengguna IAM.

Menggunakan kebijakan, administrator menentukan siapa yang memiliki akses ke apa dengan mendefinisikan principal mana yang dapat melakukan tindakan pada sumber daya apa, dan dalam kondisi apa.

Secara default, pengguna dan peran tidak memiliki izin. Administrator IAM membuat kebijakan IAM dan menambahkannya ke peran, yang kemudian dapat diambil oleh pengguna. Kebijakan IAM mendefinisikan izin terlepas dari metode yang Anda gunakan untuk melakukan operasinya.

Kebijakan berbasis identitas

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang Anda lampirkan ke identitas (pengguna, grup, atau peran). Kebijakan ini mengontrol tindakan apa yang bisa dilakukan oleh identitas tersebut, terhadap sumber daya yang mana, dan dalam kondisi apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat [Tentukan izin IAM kustom dengan kebijakan yang dikelola pelanggan](#) dalam Panduan Pengguna IAM.

Kebijakan berbasis identitas dapat berupa kebijakan inline (disematkan langsung ke dalam satu identitas) atau kebijakan terkelola (kebijakan mandiri yang dilampirkan pada banyak identitas). Untuk mempelajari cara memilih antara kebijakan terkelola dan kebijakan inline, lihat [Pilih antara kebijakan terkelola dan kebijakan inline](#) dalam Panduan Pengguna IAM.

Kebijakan berbasis sumber daya

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contohnya termasuk kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Anda harus [menentukan principal](#) dalam kebijakan berbasis sumber daya.

Kebijakan berbasis sumber daya merupakan kebijakan inline yang terletak di layanan tersebut. Anda tidak dapat menggunakan kebijakan AWS terkelola dari IAM dalam kebijakan berbasis sumber daya.

Jenis-jenis kebijakan lain

AWS mendukung jenis kebijakan tambahan yang dapat menetapkan izin maksimum yang diberikan oleh jenis kebijakan yang lebih umum:

- Batasan izin – Menetapkan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM. Untuk informasi selengkapnya, lihat [Batasan izin untuk entitas IAM](#) dalam Panduan Pengguna IAM.
- Kebijakan kontrol layanan (SCPs) — Tentukan izin maksimum untuk organisasi atau unit organisasi di AWS Organizations. Untuk informasi selengkapnya, lihat [Kebijakan kontrol layanan](#) dalam Panduan Pengguna AWS Organizations .
- Kebijakan kontrol sumber daya (RCPs) — Tetapkan izin maksimum yang tersedia untuk sumber daya di akun Anda. Untuk informasi selengkapnya, lihat [Kebijakan kontrol sumber daya \(RCPs\)](#) di Panduan AWS Organizations Pengguna.
- Kebijakan sesi – Kebijakan lanjutan yang diteruskan sebagai parameter saat membuat sesi sementara untuk peran atau pengguna terfederasi. Untuk informasi selengkapnya, lihat [Kebijakan sesi](#) dalam Panduan Pengguna IAM.

Berbagai jenis kebijakan

Ketika beberapa jenis kebijakan berlaku pada suatu permintaan, izin yang dihasilkan lebih rumit untuk dipahami. Untuk mempelajari cara AWS menentukan apakah akan mengizinkan permintaan saat beberapa jenis kebijakan terlibat, lihat [Logika evaluasi kebijakan](#) di Panduan Pengguna IAM.

Bagaimana Amazon CloudFront bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses CloudFront, pelajari fitur IAM yang tersedia untuk digunakan. CloudFront

Fitur IAM yang dapat Anda gunakan dengan Amazon CloudFront

Fitur IAM	CloudFront dukungan
Kebijakan berbasis identitas	Ya

Fitur IAM	CloudFront dukungan
Kebijakan berbasis sumber daya	Tidak
Tindakan kebijakan	Ya
Sumber daya kebijakan	Ya
kunci-kunci persyaratan kebijakan (spesifik layanan)	Ya
ACLs	Tidak
ABAC (tanda dalam kebijakan)	Parsial
Kredensial sementara	Ya
Sesi akses teruskan (FAS)	Tidak
Peran layanan	Tidak
Peran terkait layanan	Ya

Untuk mendapatkan tampilan tingkat tinggi tentang cara CloudFront dan AWS layanan lain bekerja dengan sebagian besar fitur IAM, lihat [AWS layanan yang bekerja dengan IAM di Panduan Pengguna IAM](#).

Kebijakan berbasis identitas untuk CloudFront

Mendukung kebijakan berbasis identitas: Ya

Kebijakan berbasis identitas adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke sebuah identitas, seperti pengguna IAM, grup pengguna IAM, atau peran IAM. Kebijakan ini mengontrol jenis tindakan yang dapat dilakukan oleh pengguna dan peran, di sumber daya mana, dan berdasarkan kondisi seperti apa. Untuk mempelajari cara membuat kebijakan berbasis identitas, lihat [Tentukan izin IAM kustom dengan kebijakan terkelola pelanggan](#) dalam Panduan Pengguna IAM.

Dengan kebijakan berbasis identitas IAM, Anda dapat menentukan secara spesifik apakah tindakan dan sumber daya diizinkan atau ditolak, serta kondisi yang menjadi dasar dikabulkan atau ditolaknya

tindakan tersebut. Untuk mempelajari semua elemen yang dapat Anda gunakan dalam kebijakan JSON, lihat [Referensi elemen kebijakan JSON IAM](#) dalam Panduan Pengguna IAM.

Contoh kebijakan berbasis identitas untuk CloudFront

Untuk melihat contoh kebijakan CloudFront berbasis identitas, lihat. [Contoh kebijakan berbasis identitas untuk Amazon CloudFront](#)

Kebijakan berbasis sumber daya dalam CloudFront

Mendukung kebijakan berbasis sumber daya: Tidak

Kebijakan berbasis sumber daya adalah dokumen kebijakan JSON yang Anda lampirkan ke sumber daya. Contoh kebijakan berbasis sumber daya adalah kebijakan kepercayaan peran IAM dan kebijakan bucket Amazon S3. Dalam layanan yang mendukung kebijakan berbasis sumber daya, administrator layanan dapat menggunakannya untuk mengontrol akses ke sumber daya tertentu. Untuk sumber daya tempat kebijakan dilampirkan, kebijakan menentukan tindakan apa yang dapat dilakukan oleh principal tertentu pada sumber daya tersebut dan dalam kondisi apa. Anda harus [menentukan principal](#) dalam kebijakan berbasis sumber daya. Prinsipal dapat mencakup akun, pengguna, peran, pengguna federasi, atau. Layanan AWS

Untuk mengaktifkan akses lintas akun, Anda dapat menentukan secara spesifik seluruh akun atau entitas IAM di akun lain sebagai principal dalam kebijakan berbasis sumber daya. Untuk informasi selengkapnya, lihat [Akses sumber daya lintas akun di IAM](#) dalam Panduan Pengguna IAM.

Tindakan kebijakan untuk CloudFront

Mendukung tindakan kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen `Action` dari kebijakan JSON menjelaskan tindakan yang dapat Anda gunakan untuk mengizinkan atau menolak akses dalam sebuah kebijakan. Sertakan tindakan dalam kebijakan untuk memberikan izin untuk melakukan operasi terkait.

Untuk melihat daftar CloudFront tindakan, lihat [Tindakan yang ditentukan oleh Amazon CloudFront](#) di Referensi Otorisasi Layanan.

Tindakan kebijakan CloudFront menggunakan awalan berikut sebelum tindakan:

```
cloudfront
```

Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan tersebut dengan koma.

```
"Action": [  
    "cloudfront:action1",  
    "cloudfront:action2"  
]
```

Untuk melihat contoh kebijakan CloudFront berbasis identitas, lihat. [Contoh kebijakan berbasis identitas untuk Amazon CloudFront](#)

Sumber daya kebijakan untuk CloudFront

Mendukung sumber daya kebijakan: Ya

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, di mana utama dapat melakukan tindakan pada sumber daya, dan dalam kondisi apa.

Elemen kebijakan JSON `Resource` menentukan objek yang menjadi target penerapan tindakan. Praktik terbaiknya, tentukan sumber daya menggunakan [Amazon Resource Name \(ARN\)](#). Untuk tindakan yang tidak mendukung izin di tingkat sumber daya, gunakan wildcard (*) untuk menunjukkan bahwa pernyataan tersebut berlaku untuk semua sumber daya.

```
"Resource": "*"
```

Untuk melihat daftar jenis sumber daya dan jenis CloudFront sumber daya ARNs, lihat [Sumber daya yang ditentukan oleh Amazon CloudFront](#) di Referensi Otorisasi Layanan. Untuk mempelajari tindakan mana yang dapat Anda tentukan ARN dari setiap sumber daya, lihat [Tindakan yang ditentukan oleh Amazon](#). CloudFront

Untuk melihat contoh kebijakan CloudFront berbasis identitas, lihat. [Contoh kebijakan berbasis identitas untuk Amazon CloudFront](#)

Kunci kondisi kebijakan untuk CloudFront

Mendukung kunci kondisi kebijakan khusus layanan: Yes

Administrator dapat menggunakan kebijakan AWS JSON untuk menentukan siapa yang memiliki akses ke apa. Yaitu, principal dapat melakukan tindakan pada suatu sumber daya, dan dalam suatu syarat.

Elemen `Condition` menentukan ketika pernyataan dieksekusi berdasarkan kriteria yang ditetapkan. Anda dapat membuat ekspresi bersyarat yang menggunakan [operator kondisi](#), misalnya sama dengan atau kurang dari, untuk mencocokkan kondisi dalam kebijakan dengan nilai-nilai yang diminta. Untuk melihat semua kunci kondisi AWS global, lihat [kunci konteks kondisi AWS global](#) di Panduan Pengguna IAM.

Untuk melihat daftar kunci CloudFront kondisi, lihat [Kunci kondisi untuk Amazon CloudFront](#) di Referensi Otorisasi Layanan. Untuk mempelajari tindakan dan sumber daya yang dapat Anda gunakan kunci kondisi, lihat [Tindakan yang ditentukan oleh Amazon CloudFront](#).

Untuk melihat contoh kebijakan CloudFront berbasis identitas, lihat. [Contoh kebijakan berbasis identitas untuk Amazon CloudFront](#)

ACLs di CloudFront

Mendukung ACLs: Tidak

Access control lists (ACLs) mengontrol prinsipal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACLs mirip dengan kebijakan berbasis sumber daya, meskipun mereka tidak menggunakan format dokumen kebijakan JSON.

ABAC dengan CloudFront

Mendukung ABAC (tag dalam kebijakan): Sebagian

Kontrol akses berbasis atribut (ABAC) adalah strategi otorisasi yang menentukan izin berdasarkan atribut tanda. Anda dapat melampirkan tag ke entitas dan AWS sumber daya IAM, lalu merancang kebijakan ABAC untuk mengizinkan operasi saat tag prinsipal cocok dengan tag pada sumber daya.

Untuk mengendalikan akses berdasarkan tanda, berikan informasi tentang tanda di [elemen kondisi](#) dari kebijakan menggunakan kunci kondisi `aws:ResourceTag/key-name`, `aws:RequestTag/key-name`, atau `aws:TagKeys`.

Jika sebuah layanan mendukung ketiga kunci kondisi untuk setiap jenis sumber daya, nilainya adalah Ya untuk layanan tersebut. Jika suatu layanan mendukung ketiga kunci kondisi untuk hanya beberapa jenis sumber daya, nilainya adalah Parsial.

Untuk informasi selengkapnya tentang ABAC, lihat [Tentukan izin dengan otorisasi ABAC](#) dalam Panduan Pengguna IAM. Untuk melihat tutorial yang menguraikan langkah-langkah pengaturan ABAC, lihat [Menggunakan kontrol akses berbasis atribut \(ABAC\)](#) dalam Panduan Pengguna IAM.

CloudFront mendukung ABAC untuk CloudFront sumber daya yang mendukung tag, seperti distribusi, penyewa, atau toko kepercayaan.

Menggunakan kredensi sementara dengan CloudFront

Mendukung kredensial sementara: Ya

Kredensi sementara menyediakan akses jangka pendek ke AWS sumber daya dan secara otomatis dibuat saat Anda menggunakan federasi atau beralih peran. AWS merekomendasikan agar Anda secara dinamis menghasilkan kredensi sementara alih-alih menggunakan kunci akses jangka panjang. Untuk informasi selengkapnya, lihat [Kredensial keamanan sementara di IAM](#) dan [Layanan AWS yang berfungsi dengan IAM](#) dalam Panduan Pengguna IAM.

Teruskan sesi akses untuk CloudFront

Mendukung sesi akses maju (FAS): Tidak

Sesi akses teruskan (FAS) menggunakan izin dari pemanggilan utama Layanan AWS, dikombinasikan dengan permintaan Layanan AWS untuk membuat permintaan ke layanan hilir. Untuk detail kebijakan ketika mengajukan permintaan FAS, lihat [Sesi akses terusan](#).

Peran layanan untuk CloudFront

Mendukung peran layanan: Tidak

Peran layanan adalah [peran IAM](#) yang diambil oleh sebuah layanan untuk melakukan tindakan atas nama Anda. Administrator IAM dapat membuat, mengubah, dan menghapus peran layanan dari dalam IAM. Untuk informasi selengkapnya, lihat [Buat sebuah peran untuk mendelegasikan izin ke Layanan AWS](#) dalam Panduan pengguna IAM.

 Warning

Mengubah izin untuk peran layanan dapat merusak CloudFront fungsionalitas. Edit peran layanan hanya jika CloudFront memberikan panduan untuk melakukannya.

Peran terkait layanan untuk CloudFront

Mendukung peran terkait layanan: Ya

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS Layanan tersebut dapat menjalankan peran untuk melakukan tindakan atas nama Anda. Peran terkait layanan muncul di Anda Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

CloudFront menggunakan peran terkait layanan untuk melakukan tindakan untuk Anda. Untuk informasi selengkapnya tentang membuat atau mengelola peran CloudFront terkait layanan, lihat. [Gunakan peran terkait layanan untuk CloudFront](#) Untuk informasi selengkapnya tentang membuat atau mengelola peran terkait layanan Lambda @Edge, lihat. [Peran terkait layanan untuk Lambda @Edge](#)

Untuk detail tentang pembuatan atau manajemen peran terkait layanan, lihat [Layanan AWS yang berfungsi dengan IAM](#). Cari layanan dalam tabel yang memiliki Yes di kolom Peran terkait layanan. Pilih tautan Ya untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Contoh kebijakan berbasis identitas untuk Amazon CloudFront

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau mengubah sumber daya CloudFront. Untuk memberikan izin kepada pengguna untuk melakukan tindakan di sumber daya yang mereka perlukan, administrator IAM dapat membuat kebijakan IAM.

Untuk mempelajari cara membuat kebijakan berbasis identitas IAM dengan menggunakan contoh dokumen kebijakan JSON ini, lihat [Membuat kebijakan IAM \(konsol\) di Panduan Pengguna IAM](#).

Untuk detail tentang tindakan dan jenis sumber daya yang ditentukan oleh CloudFront, termasuk format ARNs untuk setiap jenis sumber daya, lihat [Kunci tindakan, sumber daya, dan kondisi untuk Amazon CloudFront](#) di Referensi Otorisasi Layanan.

Topik

- [Praktik terbaik kebijakan](#)

- [Mengizinkan pengguna melihat izin mereka sendiri](#)
- [Izin untuk mengakses CloudFront secara terprogram](#)
- [Izin diperlukan untuk menggunakan konsol CloudFront](#)
- [Contoh kebijakan yang dikelola pelanggan](#)

Praktik terbaik kebijakan

Kebijakan berbasis identitas menentukan apakah seseorang dapat membuat, mengakses, atau menghapus CloudFront sumber daya di akun Anda. Tindakan ini membuat Akun AWS Anda dikenai biaya. Ketika Anda membuat atau mengedit kebijakan berbasis identitas, ikuti panduan dan rekomendasi ini:

- Mulailah dengan kebijakan AWS terkelola dan beralih ke izin hak istimewa paling sedikit — Untuk mulai memberikan izin kepada pengguna dan beban kerja Anda, gunakan kebijakan AWS terkelola yang memberikan izin untuk banyak kasus penggunaan umum. Mereka tersedia di Anda Akun AWS. Kami menyarankan Anda mengurangi izin lebih lanjut dengan menentukan kebijakan yang dikelola AWS pelanggan yang khusus untuk kasus penggunaan Anda. Untuk informasi selengkapnya, lihat [Kebijakan yang dikelola AWS](#) atau [Kebijakan yang dikelola AWS untuk fungsi tugas](#) dalam Panduan Pengguna IAM.
- Menerapkan izin dengan hak akses paling rendah – Ketika Anda menetapkan izin dengan kebijakan IAM, hanya berikan izin yang diperlukan untuk melakukan tugas. Anda melakukannya dengan mendefinisikan tindakan yang dapat diambil pada sumber daya tertentu dalam kondisi tertentu, yang juga dikenal sebagai izin dengan hak akses paling rendah. Untuk informasi selengkapnya tentang cara menggunakan IAM untuk mengajukan izin, lihat [Kebijakan dan izin dalam IAM](#) dalam Panduan Pengguna IAM.
- Gunakan kondisi dalam kebijakan IAM untuk membatasi akses lebih lanjut – Anda dapat menambahkan suatu kondisi ke kebijakan Anda untuk membatasi akses ke tindakan dan sumber daya. Sebagai contoh, Anda dapat menulis kondisi kebijakan untuk menentukan bahwa semua permintaan harus dikirim menggunakan SSL. Anda juga dapat menggunakan ketentuan untuk memberikan akses ke tindakan layanan jika digunakan melalui yang spesifik Layanan AWS, seperti CloudFormation. Untuk informasi selengkapnya, lihat [Elemen kebijakan JSON IAM: Kondisi](#) dalam Panduan Pengguna IAM.
- Gunakan IAM Access Analyzer untuk memvalidasi kebijakan IAM Anda untuk memastikan izin yang aman dan fungsional – IAM Access Analyzer memvalidasi kebijakan baru dan yang sudah ada sehingga kebijakan tersebut mematuhi bahasa kebijakan IAM (JSON) dan praktik terbaik IAM. IAM Access Analyzer menyediakan lebih dari 100 pemeriksaan kebijakan dan rekomendasi yang

dapat ditindaklanjuti untuk membantu Anda membuat kebijakan yang aman dan fungsional. Untuk informasi selengkapnya, lihat [Validasi kebijakan dengan IAM Access Analyzer](#) dalam Panduan Pengguna IAM.

- Memerlukan otentikasi multi-faktor (MFA) - Jika Anda memiliki skenario yang mengharuskan pengguna IAM atau pengguna root di Anda, Akun AWS aktifkan MFA untuk keamanan tambahan. Untuk meminta MFA ketika operasi API dipanggil, tambahkan kondisi MFA pada kebijakan Anda. Untuk informasi selengkapnya, lihat [Amankan akses API dengan MFA](#) dalam Panduan Pengguna IAM.

Untuk informasi selengkapnya tentang praktik terbaik dalam IAM, lihat [Praktik terbaik keamanan di IAM](#) dalam Panduan Pengguna IAM.

Mengizinkan pengguna melihat izin mereka sendiri

Contoh ini menunjukkan cara membuat kebijakan yang mengizinkan pengguna IAM melihat kebijakan inline dan terkelola yang dilampirkan ke identitas pengguna mereka. Kebijakan ini mencakup izin untuk menyelesaikan tindakan ini di konsol atau menggunakan API atau secara terprogram. AWS CLI AWS

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupsWithUser",
        "iam:ListAttachedUserPolicies",
        "iam:ListUserPolicies",
        "iam:GetUser"
      ],
      "Resource": ["arn:aws:iam::*:user/${aws:username}"]
    },
    {
      "Sid": "NavigateInConsole",
      "Effect": "Allow",
      "Action": [
        "iam:GetGroupPolicy",
        "iam:GetPolicyVersion",
```

```
        "iam:GetPolicy",
        "iam:ListAttachedGroupPolicies",
        "iam:ListGroupPolicies",
        "iam:ListPolicyVersions",
        "iam:ListPolicies",
        "iam:ListUsers"
    ],
    "Resource": "*"
}
]
```

Izin untuk mengakses CloudFront secara terprogram

Berikut ini menunjukkan kebijakan izin. Sid, atau ID pernyataan, bersifat opsional.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowAllCloudFrontPermissions",
      "Effect": "Allow",
      "Action": ["cloudfront:*"],
      "Resource": "*"
    }
  ]
}
```

Kebijakan ini memberikan izin untuk melakukan semua CloudFront operasi, yang cukup untuk mengakses CloudFront secara terprogram. Jika Anda menggunakan konsol untuk mengakses CloudFront, lihat [izin diperlukan untuk menggunakan konsol CloudFront](#).

Untuk daftar tindakan dan ARN yang Anda tentukan untuk memberikan atau menolak izin untuk menggunakan setiap tindakan, lihat [Tindakan, sumber daya, dan kunci kondisi untuk Amazon CloudFront](#) di Referensi Otorisasi Layanan.

Izin diperlukan untuk menggunakan konsol CloudFront

Untuk memberikan akses penuh ke CloudFront konsol, Anda memberikan izin dalam kebijakan izin berikut:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "acm:ListCertificates",
        "cloudfront:*",
        "cloudwatch:DescribeAlarms",
        "cloudwatch:PutMetricAlarm",
        "cloudwatch:GetMetricStatistics",
        "elasticloadbalancing:DescribeLoadBalancers",
        "iam:ListServerCertificates",
        "sns:ListSubscriptionsByTopic",
        "sns:ListTopics",
        "waf:GetWebACL",
        "waf:ListWebACLs"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListAllMyBuckets",
        "s3:PutBucketPolicy"
      ],
      "Resource": "arn:aws:s3:::*"
    }
  ]
}
```

Inilah mengapa izin diperlukan:

acm:ListCertificates

Saat Anda membuat dan memperbarui distribusi menggunakan CloudFront konsol dan Anda ingin mengonfigurasi CloudFront agar memerlukan HTTPS antara penampil dan CloudFront atau antara CloudFront dan asal, memungkinkan Anda melihat daftar sertifikat ACM.

Izin ini tidak diperlukan jika Anda tidak menggunakan CloudFront konsol.

cloudfront:*

Memungkinkan Anda melakukan semua CloudFront tindakan.

cloudwatch:DescribeAlarms dan **cloudwatch:PutMetricAlarm**

Memungkinkan Anda membuat dan melihat CloudWatch alarm di CloudFront konsol. Lihat juga `sns:ListSubscriptionsByTopic` dan `sns:ListTopics`.

Izin ini tidak diperlukan jika Anda tidak menggunakan CloudFront konsol.

cloudwatch:GetMetricStatistics

Mari kita CloudFront membuat CloudWatch metrik di CloudFront konsol.

Izin ini tidak diperlukan jika Anda tidak menggunakan CloudFront konsol.

elasticloadbalancing:DescribeLoadBalancers

Saat membuat dan memperbarui distribusi, Anda bisa melihat daftar load balancer Elastic Load Balancing dalam daftar asal yang tersedia.

Izin ini tidak diperlukan jika Anda tidak menggunakan CloudFront konsol.

iam:ListServerCertificates

Saat Anda membuat dan memperbarui distribusi menggunakan CloudFront konsol dan Anda ingin mengonfigurasi CloudFront agar memerlukan HTTPS antara penampil dan CloudFront atau antara CloudFront dan asal, memungkinkan Anda melihat daftar sertifikat di penyimpanan sertifikat IAM.

Izin ini tidak diperlukan jika Anda tidak menggunakan CloudFront konsol.

s3:ListAllMyBuckets

Saat Anda membuat dan memperbarui distribusi, memungkinkan Anda melakukan operasi berikut:

- Lihat daftar bucket S3 dalam daftar asal yang tersedia
- Lihat daftar bucket S3 yang dapat Anda gunakan untuk menyimpan log akses

Izin ini tidak diperlukan jika Anda tidak menggunakan CloudFront konsol.

S3:PutBucketPolicy

Saat Anda membuat atau memperbarui distribusi yang membatasi akses ke bucket S3, izinkan pengguna memperbarui kebijakan bucket untuk memberikan akses ke CloudFront identitas akses asal. Untuk informasi selengkapnya, lihat [the section called “Gunakan identitas akses asal \(warisan, tidak disarankan\)”](#).

Izin ini tidak diperlukan jika Anda tidak menggunakan CloudFront konsol.

sns:ListSubscriptionsByTopic dan **sns:ListTopics**

Saat Anda membuat CloudWatch alarm di CloudFront konsol, Anda dapat memilih topik SNS untuk notifikasi.

Izin ini tidak diperlukan jika Anda tidak menggunakan CloudFront konsol.

waf:GetWebACL dan **waf:ListWebACLs**

Memungkinkan Anda melihat daftar AWS WAF web ACLs di CloudFront konsol.

Izin ini tidak diperlukan jika Anda tidak menggunakan CloudFront konsol.

Tindakan hanya izin untuk konsol CloudFront

Anda dapat melakukan CloudFront tindakan berikut di halaman [CloudFront Security Savings Bundle](#). Tindakan API berikut tidak dimaksudkan untuk dipanggil oleh kode Anda, dan tidak termasuk dalam AWS CLI dan AWS SDKs.

Tindakan	Deskripsi
CreateSavingsPlan	Memberikan izin untuk membuat rencana tabungan baru.
GetSavingsPlan	Memberikan izin untuk mendapatkan rencana tabungan.

Tindakan	Deskripsi
ListRateCards	Memberikan izin untuk membuat daftar kartu CloudFront tarif untuk akun tersebut.
ListSavingsPlans	Memberikan izin untuk membuat daftar rencana tabungan di akun.
ListUsages	Memberikan izin untuk mendaftar CloudFront penggunaan.
UpdateSavingsPlan	Memberikan izin untuk memperbarui rencana tabungan.

Catatan

- Untuk informasi selengkapnya tentang paket CloudFront tabungan, lihat bagian CloudFront Security Savings Bundle di [Amazon CloudFront FAQs](#).
- Jika Anda membuat paket tabungan untuk CloudFront dan kemudian ingin menghapusnya nanti, hubungi [AWS Dukungan](#).

Contoh kebijakan yang dikelola pelanggan

Anda dapat membuat kebijakan IAM kustom Anda sendiri untuk mengizinkan izin tindakan CloudFront API. Anda dapat melampirkan kebijakan kustom ini ke pengguna IAM atau grup yang memerlukan izin yang ditentukan. Kebijakan ini berfungsi saat Anda menggunakan CloudFront API, file AWS SDKs, atau AWS CLI. Contoh-contoh berikut menunjukkan izin untuk beberapa kasus penggunaan umum. Untuk kebijakan yang memberikan akses penuh kepada pengguna CloudFront, lihat [izin diperlukan untuk menggunakan konsol CloudFront](#).

Contoh

- [Contoh 1: Mengizinkan akses baca ke semua distribusi](#)
- [Contoh 2: Mengizinkan pembuatan, pembaruan, dan penghapusan distribusi](#)
- [Contoh 3: Mengizinkan pembuatan dan pencatatan invalidasi](#)
- [Contoh 4: Izinkan membuat distribusi](#)

Contoh 1: Mengizinkan akses baca ke semua distribusi

Kebijakan izin berikut memberikan izin pengguna untuk melihat semua distribusi di konsol: CloudFront

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "acm:ListCertificates",
        "cloudfront:GetDistribution",
        "cloudfront:GetDistributionConfig",
        "cloudfront:ListDistributions",
        "cloudfront:ListCloudFrontOriginAccessIdentities",
        "elasticloadbalancing:DescribeLoadBalancers",
        "iam:ListServerCertificates",
        "sns:ListSubscriptionsByTopic",
        "sns:ListTopics",
        "waf:GetWebACL",
        "waf:ListWebACLs"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListAllMyBuckets"
      ],
      "Resource": "arn:aws:s3:::*"
    }
  ]
}
```

Contoh 2: Mengizinkan pembuatan, pembaruan, dan penghapusan distribusi

Kebijakan izin berikut memungkinkan pengguna untuk membuat, memperbarui, dan menghapus distribusi menggunakan konsol: CloudFront

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "acm:ListCertificates",
        "cloudfront:CreateDistribution",
        "cloudfront:DeleteDistribution",
        "cloudfront:GetDistribution",
        "cloudfront:GetDistributionConfig",
        "cloudfront:ListDistributions",
        "cloudfront:UpdateDistribution",
        "cloudfront:ListCloudFrontOriginAccessIdentities",
        "elasticloadbalancing:DescribeLoadBalancers",
        "iam:ListServerCertificates",
        "sns:ListSubscriptionsByTopic",
        "sns:ListTopics",
        "waf:GetWebACL",
        "waf:ListWebACLs"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListAllMyBuckets",
        "s3:PutBucketPolicy"
      ],
      "Resource": "arn:aws:s3:::*"
    }
  ]
}
```

`cloudfront:ListCloudFrontOriginAccessIdentities` memungkinkan pengguna untuk secara otomatis memberikan ke identitas akses asal yang sudah ada izin untuk mengakses objek di keranjang Amazon S3. Jika Anda juga ingin pengguna dapat membuat identitas akses asal, Anda juga perlu mengizinkan `cloudfront:CreateCloudFrontOriginAccessIdentity` izin.

Contoh 3: Mengizinkan pembuatan dan pencatatan invalidasi

Kebijakan izin berikut memungkinkan pengguna untuk membuat dan membuat daftar ketidakabsahan. Ini termasuk akses baca ke CloudFront distribusi karena Anda membuat dan melihat pembatalan dengan terlebih dahulu menampilkan pengaturan untuk distribusi:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "acm:ListCertificates",
        "cloudfront:GetDistribution",
        "cloudfront:GetStreamingDistribution",
        "cloudfront:GetDistributionConfig",
        "cloudfront:ListDistributions",
        "cloudfront:ListCloudFrontOriginAccessIdentities",
        "cloudfront:CreateInvalidation",
        "cloudfront:GetInvalidation",
        "cloudfront:ListInvalidations",
        "elasticloadbalancing:DescribeLoadBalancers",
        "iam:ListServerCertificates",
        "sns:ListSubscriptionsByTopic",
        "sns:ListTopics",
        "waf:GetWebACL",
        "waf:ListWebACLs"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "s3:ListAllMyBuckets"
      ],
      "Resource": "arn:aws:s3:::*"
    }
  ]
}
```

Contoh 4: Izinkan membuat distribusi

Kebijakan izin berikut memberikan izin kepada pengguna untuk membuat dan mencantumkan distribusi di konsol. CloudFront Untuk `CreateDistribution` tindakan, tentukan karakter wildcard (*) untuk Resource alih-alih wildcard untuk distribusi ARN (). `arn:aws:cloudfront::123456789012:distribution/*` Untuk informasi selengkapnya tentang Resource elemen, lihat [elemen kebijakan IAM JSON: Sumber daya](#) dalam Panduan Pengguna IAM.

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": "cloudfront:CreateDistribution",
      "Resource": "*"
    },
    {
      "Sid": "VisualEditor1",
      "Effect": "Allow",
      "Action": "cloudfront:ListDistributions",
      "Resource": "*"
    }
  ]
}
```

AWS kebijakan terkelola untuk Amazon CloudFront

Untuk menambahkan izin ke pengguna, grup, dan peran, lebih mudah menggunakan kebijakan AWS terkelola daripada menulis kebijakan sendiri. Dibutuhkan waktu dan keahlian untuk [membuat kebijakan terkelola pelanggan IAM](#) yang hanya memberi pengguna Anda izin yang mereka butuhkan. Untuk memulai dengan cepat, Anda dapat menggunakan kebijakan AWS terkelola kami. Kebijakan ini mencakup kasus penggunaan umum dan tersedia di Akun AWS Anda. Untuk informasi selengkapnya tentang kebijakan AWS [AWS terkelola](#), lihat [kebijakan terkelola](#) di Panduan Pengguna IAM.

AWS layanan memelihara dan memperbarui kebijakan AWS terkelola. Anda tidak dapat mengubah izin dalam kebijakan AWS terkelola. Layanan terkadang menambahkan izin tambahan ke kebijakan yang dikelola AWS untuk mendukung fitur-fitur baru. Jenis pembaruan ini akan memengaruhi semua identitas (pengguna, grup, dan peran) di mana kebijakan tersebut dilampirkan. Layanan kemungkinan besar akan memperbarui kebijakan AWS terkelola saat fitur baru diluncurkan atau ketika izin baru tersedia. Layanan tidak menghapus izin dari kebijakan AWS terkelola, sehingga pembaruan kebijakan tidak akan merusak izin yang ada.

Selain itu, AWS mendukung kebijakan terkelola untuk fungsi pekerjaan yang mencakup beberapa layanan. Misalnya, kebijakan `ReadOnlyAccess` AWS terkelola menyediakan akses hanya-baca ke semua AWS layanan dan sumber daya. Saat layanan meluncurkan fitur baru, AWS tambahkan izin hanya-baca untuk operasi dan sumber daya baru. Untuk melihat daftar dan deskripsi dari kebijakan fungsi tugas, lihat [kebijakan yang dikelola AWS untuk fungsi tugas](#) di Panduan Pengguna IAM.

Topik

- [AWS kebijakan terkelola: `CloudFrontReadOnlyAccess`](#)
- [AWS kebijakan terkelola: `CloudFrontFullAccess`](#)
- [AWS kebijakan terkelola: `AWSCloudFrontLogger`](#)
- [AWS kebijakan terkelola: `AWSLambdaReplikator`](#)
- [AWS kebijakan terkelola: `AWSCloudFrontVPCOriginServiceRolePolicy`](#)
- [CloudFront pembaruan kebijakan AWS terkelola](#)

AWS kebijakan terkelola: `CloudFrontReadOnlyAccess`

Anda dapat melampirkan kebijakan `CloudFrontReadOnlyAccess` ke identitas IAM Anda. Kebijakan ini mengizinkan izin hanya-baca untuk sumber daya. CloudFront ini juga memungkinkan izin hanya-baca ke sumber daya AWS layanan lain yang terkait dengan CloudFront dan yang terlihat di konsol. CloudFront

Detail izin

Kebijakan ini mencakup izin berikut.

- `cloudfront:Describe*`— Memungkinkan kepala sekolah untuk mendapatkan informasi tentang metadata tentang sumber daya. CloudFront

- `cloudfront:Get*`— Memungkinkan kepala sekolah untuk mendapatkan informasi rinci dan konfigurasi untuk sumber daya. CloudFront
- `cloudfront:List*`— Memungkinkan kepala sekolah untuk mendapatkan daftar sumber daya. CloudFront
- `cloudfront-keyvaluestore:Describe*`- Memungkinkan kepala sekolah untuk mendapatkan informasi tentang penyimpanan nilai kunci.
- `cloudfront-keyvaluestore:Get*`- Memungkinkan prinsipal untuk mendapatkan informasi rinci dan konfigurasi untuk penyimpanan nilai kunci.
- `cloudfront-keyvaluestore:List*`- Memungkinkan kepala sekolah untuk mendapatkan daftar toko nilai utama.
- `acm:DescribeCertificate`— Memungkinkan kepala sekolah untuk mendapatkan rincian tentang sertifikat ACM.
- `acm:ListCertificates`— Memungkinkan kepala sekolah untuk mendapatkan daftar sertifikat ACM.
- `iam:ListServerCertificates`— Memungkinkan kepala sekolah untuk mendapatkan daftar sertifikat server yang disimpan di IAM.
- `route53:List*`— Memungkinkan kepala sekolah untuk mendapatkan daftar sumber daya Route 53.
- `waf:ListWebACLs`— Memungkinkan kepala sekolah untuk mendapatkan daftar web di. ACLs AWS WAF
- `waf:GetWebACL`— Memungkinkan kepala sekolah untuk mendapatkan informasi rinci tentang web di. ACLs AWS WAF
- `wafv2:ListWebACLs`— Memungkinkan kepala sekolah untuk mendapatkan daftar web di. ACLs AWS WAF
- `wafv2:GetWebACL`— Memungkinkan kepala sekolah untuk mendapatkan informasi rinci tentang web di. ACLs AWS WAF
- `pricingplanmanager:GetSubscription`— Memungkinkan akses hanya-baca kepala sekolah untuk mendapatkan detail tentang langganan paket harga.
- `pricingplanmanager:ListSubscriptions`— Memungkinkan akses hanya-baca kepala sekolah ke daftar langganan paket harga.
- `ec2:DescribeIpamPools`— Memungkinkan kepala sekolah untuk mendapatkan informasi rinci tentang kolam IPAM Anda.

- `ec2:GetIpamPoolCidrs`— Memungkinkan kepala sekolah untuk mendapatkan yang CIDRs disediakan ke kolam IPAM.

Untuk melihat izin kebijakan ini, lihat [CloudFrontReadOnlyAccess](#) di Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: CloudFrontFullAccess

Anda dapat melampirkan kebijakan CloudFrontFullAccess ke identitas IAM Anda. Kebijakan ini memungkinkan izin administratif untuk CloudFront sumber daya. Ini juga memungkinkan izin hanya-baca ke sumber daya AWS layanan lain yang terkait dengan CloudFront dan yang terlihat di konsol. CloudFront

Detail izin

Kebijakan ini mencakup izin berikut.

- `s3:ListAllMyBuckets`— Memungkinkan kepala sekolah untuk mendapatkan daftar semua ember Amazon S3.
- `acm:DescribeCertificate`— Memungkinkan kepala sekolah untuk mendapatkan rincian tentang sertifikat ACM.
- `acm:ListCertificates`— Memungkinkan kepala sekolah untuk mendapatkan daftar sertifikat ACM.
- `acm:RequestCertificate`— Memungkinkan kepala sekolah untuk meminta sertifikat terkelola dari ACM.
- `cloudfront:*`— Memungkinkan kepala sekolah untuk melakukan semua tindakan pada semua sumber daya. CloudFront
- `cloudfront-keyvaluestore:*`— Memungkinkan prinsipal untuk melakukan semua tindakan pada penyimpanan nilai kunci.
- `iam:ListServerCertificates`— Memungkinkan kepala sekolah untuk mendapatkan daftar sertifikat server yang disimpan di IAM.
- `waf:ListWebACLs`— Memungkinkan kepala sekolah untuk mendapatkan daftar web di. ACLs AWS WAF
- `waf:GetWebACL`— Memungkinkan kepala sekolah untuk mendapatkan informasi rinci tentang web di. ACLs AWS WAF
- `waf:CreateWebACLs`— Memungkinkan prinsipal untuk membuat ACL web di. AWS WAF

- `wafv2:ListWebACLs`— Memungkinkan kepala sekolah untuk mendapatkan daftar web di. ACLs AWS WAF
- `wafv2:GetWebACL`— Memungkinkan kepala sekolah untuk mendapatkan informasi rinci tentang web di. ACLs AWS WAF
- `kinesis:ListStreams`— Memungkinkan kepala sekolah untuk mendapatkan daftar aliran Amazon Kinesis.
- `elasticloadbalancing:DescribeLoadBalancers`- Memungkinkan prinsipal untuk mendapatkan informasi rinci tentang penyeimbang beban di Elastic Load Balancing.
- `kinesis:DescribeStream`— Memungkinkan kepala sekolah untuk mendapatkan informasi rinci tentang aliran Kinesis.
- `iam:ListRoles`— Memungkinkan kepala sekolah untuk mendapatkan daftar peran di IAM.
- `pricingplanmanager:AssociateResourcesToSubscription`- Memungkinkan kepala sekolah untuk mengaitkan sumber daya ke langganan. Ini memungkinkan sumber daya dicakup oleh paket harga langganan.
- `pricingplanmanager:CancelSubscription`- Memungkinkan kepala sekolah untuk membatalkan langganan yang ada.
- `pricingplanmanager:CancelSubscriptionChange`- Memungkinkan prinsipal untuk membatalkan perubahan yang tertunda ke langganan yang ada, seperti peningkatan paket, sebelum perubahan diterapkan.
- `pricingplanmanager>CreateSubscription`- Memungkinkan kepala sekolah untuk membuat langganan ke paket harga.
- `pricingplanmanager:DisassociateResourcesFromSubscription`- Memungkinkan prinsipal untuk menghapus hubungan antara sumber daya dan langganan yang ada.
- `pricingplanmanager:UpdateSubscription`- Memungkinkan kepala sekolah untuk memodifikasi langganan yang ada, seperti mengubah paket harga.
- `pricingplanmanager:GetSubscription`— Memungkinkan akses hanya-baca kepala sekolah untuk mendapatkan detail tentang langganan paket harga.
- `pricingplanmanager:ListSubscriptions`— Memungkinkan akses hanya-baca kepala sekolah ke daftar langganan paket harga.
- `ec2:DescribeInstances`- Memungkinkan prinsipal untuk mendapatkan informasi rinci tentang instans di Amazon EC2.
- `ec2:DescribeInternetGateways`- Memungkinkan kepala sekolah untuk mendapatkan informasi rinci tentang gateway internet di Amazon EC2.

- `ec2:DescribeIpamPools`— Memungkinkan kepala sekolah untuk mendapatkan informasi rinci tentang kolam IPAM Anda.
- `ec2:GetIpamPoolCidrs`— Memungkinkan kepala sekolah untuk mendapatkan yang CIDRs disediakan ke kolam IPAM.

Untuk melihat izin kebijakan ini, lihat [CloudFrontFullAccess](#) di Referensi Kebijakan AWS Terkelola.

Important

Jika Anda CloudFront ingin membuat dan menyimpan log akses, Anda perlu memberikan izin tambahan. Untuk informasi selengkapnya, lihat [Izin](#).

AWS kebijakan terkelola: AWS CloudFront Logger

Anda tidak dapat melampirkan `AWS CloudFront Logger` kebijakan ke identitas IAM Anda. Kebijakan ini dilampirkan pada peran terkait layanan yang memungkinkan CloudFront untuk melakukan tindakan atas nama Anda. Untuk informasi selengkapnya, lihat [the section called “Peran terkait layanan untuk Lambda @Edge”](#).

Kebijakan ini memungkinkan CloudFront untuk mendorong file log ke Amazon CloudWatch. Untuk detail tentang izin yang disertakan dalam kebijakan ini, lihat [the section called “Izin peran terkait layanan untuk logger CloudFront”](#).

Untuk melihat izin kebijakan ini, lihat [AWS CloudFront Logger](#) di Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AWS Lambda Replikator

Anda tidak dapat melampirkan kebijakan `AWS Lambda Replikator` ke identitas IAM Anda. Kebijakan ini dilampirkan pada peran terkait layanan yang memungkinkan CloudFront untuk melakukan tindakan atas nama Anda. Untuk informasi selengkapnya, lihat [the section called “Peran terkait layanan untuk Lambda @Edge”](#).

Kebijakan ini memungkinkan CloudFront untuk membuat, menghapus, dan menonaktifkan fungsi AWS Lambda untuk mereplikasi fungsi Lambda @Edge. Wilayah AWS Untuk detail tentang izin yang disertakan dalam kebijakan ini, lihat [the section called “Izin peran terkait layanan untuk replikator Lambda”](#).

Untuk melihat izin kebijakan ini, lihat [AWS Lambda Replikator](#) di Referensi Kebijakan AWS Terkelola.

AWS kebijakan terkelola: AWSCloud Depan VPCOrigin ServiceRolePolicy

Anda tidak dapat melampirkan VPCOrigin ServiceRolePolicy kebijakan AWSCloudFront ke entitas IAM Anda. Kebijakan ini dilampirkan pada peran terkait layanan yang memungkinkan CloudFront untuk melakukan tindakan atas nama Anda. Untuk informasi selengkapnya, lihat [Gunakan peran terkait layanan untuk CloudFront](#).

Kebijakan ini memungkinkan CloudFront untuk mengelola antarmuka jaringan elastis EC2 dan grup keamanan atas nama Anda. Untuk detail tentang izin yang disertakan dalam kebijakan ini, lihat [the section called “Izin peran terkait layanan untuk VPC Origins CloudFront”](#).

Untuk melihat izin kebijakan ini, lihat [AWSCloudBagian Depan VPCOrigin ServiceRolePolicy](#) di Referensi Kebijakan AWS Terkelola.

CloudFront pembaruan kebijakan AWS terkelola

Lihat detail tentang pembaruan kebijakan AWS terkelola CloudFront sejak layanan ini mulai melacak perubahan ini. Untuk peringatan otomatis tentang perubahan pada halaman ini, berlangganan umpan RSS di halaman [Riwayat CloudFront dokumen](#).

Ubah	Deskripsi	Date
CloudFrontReadOnlyAccess – Pembaruan ke kebijakan yang sudah ada	CloudFront menambahkan izin baru untuk Amazon EC2. Izin baru memungkinkan prinsipal untuk menggunakan dan tindakan. <code>ec2:DescribeIpamPools</code> <code>ec2:GetIpamPoolCidrs</code>	November 24, 2025
CloudFrontFullAccess – Pembaruan ke kebijakan yang sudah ada	CloudFront menambahkan izin baru untuk Amazon EC2. Izin baru memungkinkan prinsipal untuk menggunakan dan tindakan. <code>ec2:DescribeIpamPools</code> <code>ec2:GetIpamPoolCidrs</code>	November 24, 2025

Ubah	Deskripsi	Date
CloudFrontFullAccess – Pembaruan ke kebijakan yang sudah ada	CloudFront menambahkan izin baru untuk membuat sumber daya AWS WAF ACL, dan menambahkan izin buat, perbarui, hapus, dan baca ke AWS Pricing Plan Manager.	November 18, 2025
CloudFrontFullAccess – Pembaruan ke kebijakan yang sudah ada	CloudFront menambahkan izin baru untuk membuat sumber daya AWS WAF ACL, dan menambahkan izin buat, perbarui, hapus, dan baca ke AWS Pricing Plan Manager.	November 18, 2025
CloudFrontReadOnlyAccess – Pembaruan ke kebijakan yang sudah ada	CloudFront menambahkan izin baru untuk akses hanya-baca ke AWS Pricing Plan Manager.	November 18, 2025
CloudFrontReadOnlyAccess – Pembaruan ke kebijakan yang sudah ada	CloudFront menambahkan izin baru untuk akses hanya-baca ke AWS Pricing Plan Manager.	November 18, 2025
CloudFrontReadOnlyAccess – Pembaruan ke kebijakan yang sudah ada	CloudFront menambahkan izin baru untuk ACM. Izin baru memungkinkan kepala sekolah untuk mendapatkan rincian tentang sertifikat ACM.	April 28, 2025

Ubah	Deskripsi	Date
CloudFrontFullAccess – Pembaruan ke kebijakan yang sudah ada	CloudFront menambahkan izin baru untuk ACM. Izin baru memungkinkan kepala sekolah untuk mendapatkan rincian tentang sertifikat ACM dan meminta sertifikat terkelola dari ACM.	April 28, 2025
CloudFrontFullAccess – Pembaruan ke kebijakan yang sudah ada	CloudFront menambahkan izin baru untuk Amazon EC2 dan Elastic Load Balancing. Izin baru memungkinkan CloudFront untuk mendapatkan informasi terperinci tentang penyeimbang beban di Elastic Load Balancing dan instans serta gateway internet di Amazon EC2.	November 20, 2024
AWSCloudDepan VPCOrigin ServiceRolePolicy - Kebijakan baru	CloudFront menambahkan kebijakan baru. Kebijakan ini memungkinkan CloudFront untuk mengelola antarmuka jaringan elastis EC2 dan grup keamanan atas nama Anda.	November 20, 2024

Ubah	Deskripsi	Date
CloudFrontReadOnlyAccess dan CloudFrontFullAccess - Perbarui ke dua kebijakan yang ada.	CloudFront menambahkan izin baru untuk penyimpanan nilai utama. Izin baru memungkinkan pengguna untuk mendapatkan informasi tentang, dan mengambil tindakan pada, penyimpanan nilai utama.	Desember 19, 2023
CloudFrontReadOnlyAccess – Pembaruan ke kebijakan yang ada	CloudFront menambahkan izin baru untuk menggambarkan CloudFront Fungsi. Izin ini memungkinkan pengguna, grup, atau peran untuk membaca informasi dan metadata tentang suatu fungsi, tetapi bukan kode fungsi.	8 September 2021
CloudFront mulai melacak perubahan	CloudFront mulai melacak perubahan untuk kebijakan yang AWS dikelola.	8 September 2021

Gunakan peran terkait layanan untuk CloudFront

Amazon CloudFront menggunakan peran AWS Identity and Access Management [terkait layanan](#) (IAM). Peran terkait layanan adalah jenis unik peran IAM yang ditautkan langsung ke CloudFront. Peran terkait layanan telah ditentukan sebelumnya oleh CloudFront dan mencakup semua izin yang diperlukan layanan untuk memanggil AWS layanan lain atas nama Anda.

Peran terkait layanan membuat pengaturan CloudFront lebih mudah karena Anda tidak perlu menambahkan izin yang diperlukan secara manual. CloudFront mendefinisikan izin peran terkait layanan, dan kecuali ditentukan lain, hanya CloudFront dapat mengambil perannya. Izin yang

ditentukan mencakup kebijakan kepercayaan dan kebijakan izin, dan kebijakan izin tersebut tidak dapat dilampirkan ke entitas IAM lainnya.

Anda dapat menghapus peran tertaut layanan hanya setelah menghapus sumber daya terkait terlebih dahulu. Ini melindungi CloudFront sumber daya Anda karena Anda tidak dapat secara tidak sengaja menghapus izin untuk mengakses sumber daya.

Untuk informasi tentang layanan lain yang mendukung peran terkait layanan, silakan lihat [layanan AWS yang bisa digunakan dengan IAM](#) dan carilah layanan yang memiliki opsi Ya di kolom Peran terkait layanan. Pilih Ya dengan sebuah tautan untuk melihat dokumentasi peran terkait layanan untuk layanan tersebut.

Izin peran terkait layanan untuk VPC Origins CloudFront

CloudFront VPC Origins menggunakan peran terkait layanan bernama `AWSServiceRoleForCloudFrontVPCOrigin`— Memungkinkan CloudFront untuk mengelola antarmuka jaringan elastis EC2 dan grup keamanan atas nama Anda.

Peran tertaut layanan `AWSServiceRoleForCloudFrontVPCOrigin` memercayai layanan berikut untuk mengambil peran tersebut:

- `vpcorigin.cloudfront.amazonaws.com`

Kebijakan izin peran bernama `AWSCloudFrontVPCOriginServiceRolePolicy` memungkinkan CloudFront VPC Origins untuk menyelesaikan tindakan berikut pada sumber daya yang ditentukan:

- Tindakan: `ec2:CreateNetworkInterface` pada `arn:aws:ec2:*:*:network-interface/*`
- Aksi: `ec2:CreateNetworkInterface` pada `arn:aws:ec2:*:*:subnet/*` dan `arn:aws:ec2:*:*:security-group/*`
- Tindakan: `ec2:CreateSecurityGroup` pada `arn:aws:ec2:*:*:security-group/*`
- Tindakan: `ec2:CreateSecurityGroup` pada `arn:aws:ec2:*:*:vpc/*`
- Tindakan: `ec2:ModifyNetworkInterfaceAttribute`, `ec2>DeleteNetworkInterface`, `ec2:DeleteNetworkInterface` dan `ec2:UnassignIpv6Addresses` seterusnya supported AWS resources that have the `aws:ResourceTag/aws.cloudfront.vpcorigin` tag enabled
- Tindakan: `ec2:DescribeNetworkInterfaces`, `ec2:DescribeSecurityGroups`, `ec2:DescribeInstances` dan `ec2:DescribeAddresses` seterusnya all AWS resources that the actions support

- Aksi: `ec2:CreateTags` pada `arn:aws:ec2:*:*:security-group/*` dan `arn:aws:ec2:*:*:network-interface/*`
- Tindakan: `elasticloadbalancing:DescribeLoadBalancers`, `elasticloadbalancing:Describe` dan `elasticloadbalancing:DescribeTargetGroups` seterusnya all AWS resources that the actions support

Anda harus mengonfigurasi izin agar pengguna, grup, atau peran Anda membuat, mengedit, atau menghapus peran terkait layanan. Untuk informasi selengkapnya, lihat [Izin peran terkait layanan](#) dalam Panduan Pengguna IAM.

Membuat peran terkait layanan untuk CloudFront VPC Origins

Anda tidak perlu membuat peran terkait layanan secara manual. Saat Anda membuat asal VPC di API Konsol Manajemen AWS, atau AWS API AWS CLI, CloudFront VPC Origins membuat peran terkait layanan untuk Anda.

Jika Anda menghapus peran terkait layanan ini, dan ingin membuatnya lagi, Anda dapat mengulangi proses yang sama untuk membuat kembali peran tersebut di akun Anda. Saat Anda membuat asal VPC, VPC Origins membuat CloudFront peran terkait layanan untuk Anda lagi.

Mengedit peran terkait layanan untuk VPC Origins CloudFront

CloudFront VPC Origins tidak memungkinkan Anda mengedit peran terkait `AWSServiceRoleForCloudFrontVPCOrigin` layanan. Setelah membuat peran terkait layanan, Anda tidak dapat mengubah nama peran karena berbagai entitas mungkin merujuk peran tersebut. Namun, Anda dapat mengedit penjelasan peran menggunakan IAM. Untuk informasi selengkapnya, lihat [Mengedit peran terkait layanan](#) dalam Panduan Pengguna IAM.

Menghapus peran terkait layanan untuk VPC Origins CloudFront

Jika Anda tidak perlu lagi menggunakan fitur atau layanan yang memerlukan peran terkait layanan, sebaiknya hapus peran tersebut. Dengan begitu, Anda tidak memiliki entitas yang tidak digunakan yang tidak dipantau atau dipelihara secara aktif. Tetapi, Anda harus membersihkan sumber daya peran yang terhubung dengan layanan sebelum menghapusnya secara manual.

 Note

Jika CloudFront layanan menggunakan peran saat Anda mencoba menghapus sumber daya, maka penghapusan mungkin gagal. Jika hal itu terjadi, tunggu beberapa menit dan coba mengoperasikannya lagi.

Untuk menghapus sumber daya CloudFront VPC Origins yang digunakan oleh `AWSServiceRoleForCloudFrontVPCOrigin`

- Hapus sumber daya asal VPC di akun Anda.
- Mungkin perlu beberapa waktu CloudFront untuk menyelesaikan penghapusan sumber daya dari akun Anda. Jika Anda tidak dapat langsung menghapus peran terkait layanan, tunggu dan coba lagi.

Untuk menghapus peran tertaut layanan secara manual menggunakan IAM

Gunakan konsol IAM, the AWS CLI, atau AWS API untuk menghapus peran `AWSServiceRoleForCloudFrontVPCOrigin` terkait layanan. Untuk informasi selengkapnya, lihat [Menghapus peran terkait layanan](#) dalam Panduan Pengguna IAM.

Wilayah yang Didukung untuk CloudFront peran terkait layanan VPC Origins

CloudFront VPC Origins tidak mendukung penggunaan peran terkait layanan di setiap Wilayah tempat layanan tersedia. Anda dapat menggunakan `AWSServiceRoleForCloudFrontVPCOrigin` peran di Wilayah berikut.

Nama wilayah	Identitas wilayah	Support di CloudFront
US East (Northern Virginia)	us-east-1	Ya
US East (Ohio)	us-east-2	Ya
AS Barat (California Utara)	us-west-1 (kecuali AZ usw1-az2)	Ya
US West (Oregon)	us-west-2	Ya

Nama wilayah	Identitas wilayah	Support di CloudFront
Africa (Cape Town)	af-south-1	Ya
Asia Pasifik (Hong Kong)	ap-east-1	Ya
Asia Pasifik (Jakarta)	ap-southeast-3	Ya
Asia Pacific (Melbourne)	ap-southeast-4	Ya
Asia Pacific (Mumbai)	ap-south-1	Ya
Asia Pasifik (Hyderabad)	ap-south-2	Ya
Asia Pacific (Osaka)	ap-northeast-3	Ya
Asia Pacific (Seoul)	ap-northeast-2	Ya
Asia Pacific (Singapore)	ap-southeast-1	Ya
Asia Pacific (Sydney)	ap-southeast-2	Ya
Asia Pasifik (Tokyo)	ap-northeast-1 (kecuali AZ apne1-az3)	Ya
Kanada (Pusat)	ca-central-1 (kecuali AZ cac1-az3)	Ya
Kanada Barat (Calgary)	ca-west-1	Ya
Eropa (Frankfurt)	eu-central-1	Ya
Eropa (Irlandia)	eu-west-1	Ya
Eropa (London)	eu-west-2	Ya
Europe (Milan)	eu-south-1	Ya
Europe (Paris)	eu-west-3	Ya
Eropa (Spanyol)	eu-south-2	Ya

Nama wilayah	Identitas wilayah	Support di CloudFront
Eropa (Stockholm)	eu-north-1	Ya
Europe (Zurich)	eu-central-2	Ya
Israel (Tel Aviv)	il-central-1	Ya
Timur Tengah (Bahrain)	me-south-1	Ya
Timur Tengah (UAE)	me-central-1	Ya
South America (São Paulo)	sa-east-1	Ya

Memecahkan masalah CloudFront identitas dan akses Amazon

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan CloudFront dan IAM.

Topik

- [Saya tidak berwenang untuk melakukan tindakan di CloudFront](#)
- [Saya tidak berwenang untuk melakukan iam: PassRole](#)
- [Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses CloudFront sumber daya saya](#)

Saya tidak berwenang untuk melakukan tindakan di CloudFront

Jika Anda menerima pesan kesalahan bahwa Anda tidak memiliki otorisasi untuk melakukan tindakan, kebijakan Anda harus diperbarui agar Anda dapat melakukan tindakan tersebut.

Contoh kesalahan berikut terjadi ketika pengguna IAM mateojackson mencoba menggunakan konsol untuk melihat detail tentang suatu sumber daya *my-example-widget* rekaan, tetapi tidak memiliki izin `cloudfront:GetWidget` rekaan.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
cloudfront:GetWidget on resource: my-example-widget
```

Dalam hal ini, kebijakan untuk pengguna `mateo.jackson` harus diperbarui untuk mengizinkan akses ke sumber daya `my-example-widget` dengan menggunakan tindakan `cloudfront:GetWidget`.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya tidak berwenang untuk melakukan `iam:PassRole`

Jika Anda menerima kesalahan yang tidak diizinkan untuk melakukan `iam:PassRole` tindakan, kebijakan Anda harus diperbarui agar Anda dapat meneruskan peran CloudFront.

Beberapa Layanan AWS memungkinkan Anda untuk meneruskan peran yang ada ke layanan tersebut alih-alih membuat peran layanan baru atau peran terkait layanan. Untuk melakukannya, Anda harus memiliki izin untuk meneruskan peran ke layanan.

Contoh kesalahan berikut terjadi ketika pengguna IAM bernama `marymajor` mencoba menggunakan konsol tersebut untuk melakukan tindakan di CloudFront. Namun, tindakan tersebut memerlukan layanan untuk mendapatkan izin yang diberikan oleh peran layanan. Mary tidak memiliki izin untuk meneruskan peran tersebut pada layanan.

```
User: arn:aws:iam::123456789012:user/marymajor is not authorized to perform:
iam:PassRole
```

Dalam kasus ini, kebijakan Mary harus diperbarui agar dia mendapatkan izin untuk melakukan tindakan `iam:PassRole` tersebut.

Jika Anda memerlukan bantuan, hubungi AWS administrator Anda. Administrator Anda adalah orang yang memberi Anda kredensial masuk.

Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses CloudFront sumber daya saya

Anda dapat membuat peran yang dapat digunakan pengguna di akun lain atau orang-orang di luar organisasi Anda untuk mengakses sumber daya Anda. Anda dapat menentukan siapa saja yang dipercaya untuk mengambil peran tersebut. Untuk layanan yang mendukung kebijakan berbasis sumber daya atau daftar kontrol akses (ACLs), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi berikut:

- Untuk mempelajari apakah CloudFront mendukung fitur ini, lihat [Bagaimana Amazon CloudFront bekerja dengan IAM](#).
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda di seluruh sumber daya Akun AWS yang Anda miliki, lihat [Menyediakan akses ke pengguna IAM di pengguna lain Akun AWS yang Anda miliki](#) di Panduan Pengguna IAM.
- Untuk mempelajari cara menyediakan akses ke sumber daya Anda kepada pihak ketiga Akun AWS, lihat [Menyediakan akses yang Akun AWS dimiliki oleh pihak ketiga](#) dalam Panduan Pengguna IAM.
- Untuk mempelajari cara memberikan akses melalui federasi identitas, lihat [Menyediakan akses ke pengguna terautentikasi eksternal \(federasi identitas\)](#) dalam Panduan Pengguna IAM.
- Untuk mempelajari perbedaan antara menggunakan peran dan kebijakan berbasis sumber daya untuk akses lintas akun, lihat [Akses sumber daya lintas akun di IAM di Panduan Pengguna IAM](#).

Pencatatan dan pemantauan di Amazon CloudFront

Pemantauan adalah bagian penting dari menjaga ketersediaan dan kinerja CloudFront dan AWS solusi Anda. Anda harus mengumpulkan data pemantauan dari semua bagian AWS solusi Anda sehingga Anda dapat lebih mudah men-debug kegagalan multi-titik jika terjadi. AWS menyediakan beberapa alat untuk memantau CloudFront sumber daya dan aktivitas Anda, dan menanggapi potensi insiden:

CloudWatch Alarm Amazon

Menggunakan CloudWatch alarm, Anda menonton satu metrik selama periode waktu yang Anda tentukan. Jika metrik melebihi ambang batas tertentu, pemberitahuan akan dikirim ke topik atau AWS Auto Scaling kebijakan Amazon SNS. CloudWatch alarm tidak memanggil tindakan ketika metrik berada dalam keadaan tertentu. Sebaliknya, kondisi tersebut harus diubah dan dipertahankan selama periode tertentu.

Untuk informasi selengkapnya, lihat [Pantau CloudFront metrik dengan Amazon CloudWatch](#).

AWS CloudTrail log

CloudTrail menyediakan catatan tindakan API yang diambil oleh pengguna, peran, atau AWS layanan di CloudFront. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menentukan permintaan API yang dibuat CloudFront, alamat IP dari mana permintaan dibuat, siapa yang membuat permintaan, kapan dibuat, dan detail tambahan.

Untuk informasi selengkapnya, lihat [Pencatatan panggilan CloudFront API Amazon menggunakan AWS CloudTrail](#).

CloudFront log standar dan log akses real-time

CloudFront log memberikan catatan rinci tentang permintaan yang dibuat untuk distribusi. Log ini bermanfaat untuk banyak aplikasi. Misalnya, informasi log bisa bermanfaat untuk audit keamanan dan akses.

Untuk informasi selengkapnya, lihat [Akses log \(log standar\)](#) dan [Membuat dan menggunakan konfigurasi log akses real-time](#).

Log fungsi tepi

Log yang dihasilkan oleh fungsi edge, baik CloudFront Fungsi maupun Lambda @Edge, dikirim langsung ke Amazon CloudWatch Logs dan tidak disimpan di mana pun oleh CloudFront. CloudFront Fungsi menggunakan [peran terkait layanan AWS Identity and Access Management \(IAM\)](#) untuk mengirim log buatan pelanggan langsung ke CloudWatch Log di akun Anda.

Untuk informasi selengkapnya, lihat [Log fungsi tepi](#).

CloudFront laporan konsol

CloudFront Konsol mencakup berbagai laporan, termasuk laporan statistik cache, laporan objek populer, dan laporan perujuk teratas. Sebagian besar laporan CloudFront konsol didasarkan pada data dalam log CloudFront akses, yang berisi informasi terperinci tentang setiap permintaan pengguna yang CloudFront diterima. Namun, Anda tidak perlu mengaktifkan log akses untuk melihat laporan.

Untuk informasi selengkapnya, lihat [Lihat CloudFront laporan di konsol](#).

Validasi kepatuhan untuk Amazon CloudFront

Auditor pihak ketiga menilai keamanan dan kepatuhan Amazon CloudFront sebagai bagian dari beberapa program AWS kepatuhan. Hal ini mencakup SOC, PCI, HIPAA, dan lainnya.

Untuk daftar AWS layanan dalam lingkup program kepatuhan tertentu, lihat [AWS Layanan dalam Lingkup berdasarkan Program Kepatuhan](#). Untuk informasi umum, lihat [Program Kepatuhan AWS](#).

Anda dapat mengunduh laporan audit pihak ketiga menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat [Mengunduh Laporan di AWS Artifact](#).

Tanggung jawab kepatuhan Anda saat menggunakan CloudFront ditentukan oleh sensitivitas data Anda, tujuan kepatuhan perusahaan Anda, dan hukum dan peraturan yang berlaku. AWS menyediakan sumber daya berikut untuk membantu kepatuhan:

- [Panduan Memulai Cepat Keamanan dan Kepatuhan — Panduan](#) penerapan ini membahas pertimbangan arsitektur dan memberikan langkah-langkah untuk menerapkan lingkungan dasar yang berfokus pada keamanan dan kepatuhan. AWS
- [Arsitektur untuk Keamanan dan Kepatuhan HIPAA AWS](#) — Whitepaper ini menjelaskan bagaimana perusahaan dapat menggunakan AWS untuk membuat aplikasi yang sesuai dengan HIPAA.

Program kepatuhan AWS HIPAA mencakup CloudFront (tidak termasuk pengiriman konten melalui CloudFront Embedded POPs) sebagai layanan yang memenuhi syarat HIPAA. Jika Anda memiliki Adendum Rekanan Bisnis (BAA) yang dieksekusi AWS, Anda dapat menggunakan CloudFront (tidak termasuk pengiriman konten melalui CloudFront Embedded POPs) untuk mengirimkan konten yang berisi informasi kesehatan yang dilindungi (PHI). Untuk informasi lebih lanjut, lihat [Kepatuhan HIPAA](#).

- [AWS Sumber Daya Kepatuhan](#) — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- [AWS Config](#) AWS Layanan ini menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- [AWS Security Hub CSPM](#) AWS Layanan ini menggunakan kontrol keamanan untuk mengevaluasi konfigurasi sumber daya dan standar keamanan untuk membantu Anda mematuhi berbagai kerangka kerja kepatuhan. Untuk informasi selengkapnya tentang menggunakan CSPM Security Hub untuk mengevaluasi CloudFront sumber daya, lihat [CloudFront kontrol Amazon](#) di AWS Security Hub CSPM Panduan Pengguna.

CloudFront praktik terbaik kepatuhan

Bagian ini memberikan praktik dan rekomendasi terbaik untuk kepatuhan saat Anda menggunakan Amazon CloudFront untuk menayangkan konten Anda.

Jika Anda menjalankan beban kerja yang sesuai dengan PCI atau sesuai dengan HIPAA yang didasarkan pada [model tanggung jawab AWS bersama](#), sebaiknya Anda mencatat CloudFront data penggunaan Anda selama 365 hari terakhir untuk tujuan audit di masa mendatang. Untuk mencatat data penggunaan, Anda bisa melakukan hal berikut ini:

- Aktifkan log CloudFront akses. Untuk informasi selengkapnya, lihat [Akses log \(log standar\)](#).

- Tangkap permintaan yang dikirim ke CloudFront API. Untuk informasi selengkapnya, lihat [Pencatatan panggilan CloudFront API Amazon menggunakan AWS CloudTrail](#).

Selain itu, lihat berikut ini untuk detail tentang bagaimana CloudFront sesuai dengan standar PCI DSS dan SOC.

Standar Keamanan Data Industri Kartu Pembayaran (PCI DSS)

CloudFront (tidak termasuk pengiriman konten melalui CloudFront Embedded POPs) mendukung pemrosesan, penyimpanan, dan transmisi data kartu kredit oleh pedagang atau penyedia layanan, dan telah divalidasi sebagai sesuai dengan Standar Keamanan Data Industri Kartu Pembayaran (PCI) Data Security Standard (DSS). Untuk informasi selengkapnya tentang PCI DSS, termasuk cara meminta salinan PCI AWS Compliance Package, lihat [PCI DSS Level 1](#).

Sebagai praktik terbaik keamanan, kami menyarankan Anda untuk tidak menyimpan informasi kartu kredit di cache CloudFront tepi. Misalnya, Anda dapat mengonfigurasi asal Anda untuk menyertakan `Cache-Control:no-cache=" field-name "` header dalam tanggapan yang berisi informasi kartu kredit, seperti empat digit terakhir dari nomor kartu kredit dan informasi kontak pemilik kartu.

Kontrol Sistem dan Organisasi (SOC)

CloudFront (tidak termasuk pengiriman konten melalui CloudFront Embedded POPs) sesuai dengan tindakan Sistem dan Kontrol Organisasi (SOC), termasuk SOC 1, SOC 2, dan SOC 3. Laporan SOC adalah laporan pemeriksaan pihak ketiga independen yang menunjukkan bagaimana AWS mencapai kontrol dan tujuan kepatuhan utama. Audit ini memastikan adanya perlindungan dan prosedur yang sesuai untuk melindungi dari risiko yang dapat memengaruhi keamanan, kerahasiaan, dan ketersediaan data pelanggan dan perusahaan. Hasil audit pihak ketiga ini tersedia di [situs web Kepatuhan AWS SOC](#), di mana Anda dapat melihat laporan yang dipublikasikan untuk mendapatkan informasi lebih lanjut tentang kontrol yang mendukung AWS operasi dan kepatuhan.

Ketahanan di Amazon CloudFront

Infrastruktur AWS global dibangun di sekitar AWS Wilayah dan Zona Ketersediaan. AWS Wilayah menyediakan beberapa Zona Ketersediaan yang terpisah dan terisolasi secara fisik, yang terhubung dengan jaringan latensi rendah, throughput tinggi, dan sangat redundan. Dengan Zona Ketersediaan, Anda dapat merancang serta mengoperasikan aplikasi dan basis data yang secara otomatis melakukan failover di antara Zona Ketersediaan tanpa gangguan. Zona Ketersediaan

memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur biasa yang terdiri dari satu atau beberapa pusat data.

Untuk informasi selengkapnya tentang AWS Wilayah dan Availability Zone, lihat [Infrastruktur AWS Global](#).

CloudFront failover asal

Selain dukungan infrastruktur AWS global, Amazon CloudFront menawarkan fitur failover asal untuk membantu mendukung kebutuhan ketahanan data Anda. CloudFront adalah layanan global yang mengirimkan konten Anda melalui jaringan pusat data di seluruh dunia yang disebut lokasi tepi atau titik kehadiran (POPs). Jika konten Anda belum di-cache di lokasi tepi, CloudFront ambil dari asal yang telah Anda identifikasi sebagai sumber untuk versi definitif konten.

Anda dapat meningkatkan ketahanan dan meningkatkan ketersediaan untuk skenario tertentu dengan mengatur CloudFront dengan failover asal. Untuk memulai, Anda membuat grup asal tempat Anda menetapkan asal utama untuk CloudFront ditambah asal kedua. CloudFront secara otomatis beralih ke asal kedua ketika asal utama mengembalikan respons kegagalan kode status HTTP tertentu. Untuk informasi selengkapnya, lihat [Optimalkan ketersediaan tinggi dengan failover CloudFront asal](#).

Keamanan infrastruktur di Amazon CloudFront

Sebagai layanan terkelola, Amazon CloudFront dilindungi oleh keamanan jaringan AWS global. Untuk informasi tentang layanan AWS keamanan dan cara AWS melindungi infrastruktur, lihat [Keamanan AWS Cloud](#). Untuk mendesain AWS lingkungan Anda menggunakan praktik terbaik untuk keamanan infrastruktur, lihat [Perlindungan Infrastruktur dalam Kerangka Kerja](#) yang AWS Diarsiteksikan dengan Baik Pilar Keamanan.

Anda menggunakan panggilan API yang AWS dipublikasikan untuk mengakses CloudFront melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Pengangkutan (TLS). Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Sandi cocok dengan sistem kerahasiaan maju sempurna (perfect forward secrecy, PFS) seperti DHE (Ephemeral Diffie-Hellman) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

CloudFront Fungsi menggunakan penghalang isolasi yang sangat aman antar AWS akun, memastikan bahwa lingkungan pelanggan aman terhadap serangan saluran samping seperti Spectre dan Meltdown. Fungsi tidak dapat mengakses atau memodifikasi data milik pelanggan lain. Fungsi berjalan dalam proses single-threaded khusus pada CPU khusus tanpa hyperthreading. Dalam setiap titik keberadaan lokasi CloudFront tepi (POP) tertentu, CloudFront Fungsi hanya melayani satu pelanggan pada satu waktu, dan semua data khusus pelanggan dihapus di antara eksekusi fungsi.

Pemecahan Masalah

Gunakan bagian ini untuk memecahkan masalah umum yang mungkin Anda temui saat menyiapkan Amazon CloudFront untuk mendistribusikan konten Anda.

Setiap topik memberikan panduan terperinci tentang mengidentifikasi akar penyebab masalah umum dan step-by-step instruksi untuk menyelesaikannya.

Topik

- [Memecahkan masalah distribusi](#)
- [Memecahkan masalah kode status respons kesalahan di CloudFront](#)
- [Pengujian beban CloudFront](#)

Memecahkan masalah distribusi

Gunakan informasi di sini untuk membantu Anda mendiagnosis dan memperbaiki kesalahan sertifikat, masalah yang ditolak akses, atau masalah umum lainnya yang mungkin Anda temui saat menyiapkan situs web atau aplikasi Anda dengan distribusi Amazon CloudFront .

Topik

- [CloudFront mengembalikan Access Denied kesalahan](#)
- [CloudFront mengembalikan InvalidViewerCertificate kesalahan ketika saya mencoba menambahkan nama domain alternatif](#)
- [CloudFront mengembalikan kesalahan catatan DNS yang tidak dikonfigurasi dengan benar ketika saya mencoba menambahkan CNAME baru](#)
- [Saya tidak dapat melihat file dalam distribusi saya](#)
- [Pesan galat: Sertifikat: <certificate-id>sedang digunakan oleh CloudFront](#)

CloudFront mengembalikan Access Denied kesalahan

Jika Anda menggunakan bucket Amazon S3 sebagai asal CloudFront distribusi, Anda mungkin melihat pesan galat Access Denied (403) dalam contoh berikut.

Daftar Isi

- [Anda menentukan objek yang hilang dari asal Amazon S3](#)

- [Asal Amazon S3 Anda tidak memiliki izin IAM](#)
- [Anda menggunakan kredensial yang tidak valid atau tidak memiliki izin yang memadai](#)

Anda menentukan objek yang hilang dari asal Amazon S3

Verifikasi bahwa objek yang diminta di bucket Anda ada. Nama objek peka huruf besar/kecil. Memasukkan nama objek yang tidak valid dapat mengembalikan kode kesalahan akses ditolak.

Misalnya, jika Anda mengikuti [CloudFront tutorial](#) untuk membuat distribusi dasar, Anda membuat bucket Amazon S3 sebagai asal dan mengunggah file `contohindex.html`.

Di browser web Anda, jika Anda memasukkan, `https://d111111abcdef8.cloudfront.net/INDEX.HTML` bukan `https://d111111abcdef8.cloudfront.net/index.html`, Anda mungkin melihat pesan serupa karena `index.html` file di jalur URL peka huruf besar/kecil.

```
<Error>
<Code>AccessDenied</Code>
<Message>Access Denied</Message>
<RequestId>22Q367AHT7Y1ABCD</RequestId>
<HostId>
ABCDE/Vg+7PSNa/d/IffQ8Fb92TGQ0KH0ZwG5iEKbc6+e06DdMS1ZW+ryB9GFRIVtS66rSSy6So=
</HostId>
</Error>
```

Asal Amazon S3 Anda tidak memiliki izin IAM

Pastikan Anda telah memilih bucket Amazon S3 yang benar sebagai domain dan nama asal. Asal (Amazon S3) harus memiliki izin yang benar.

Jika Anda tidak menentukan izin yang benar, `AccessDenied` pesan dapat muncul untuk pemirsa Anda.

Saat Anda mendistribusikan konten dari Amazon S3 dan Anda juga menggunakan enkripsi sisi layanan AWS Key Management Service (AWS KMS) (SSE-KMS), ada izin IAM tambahan yang perlu Anda tentukan untuk kunci KMS dan bucket Amazon S3. CloudFront Distribusi Anda memerlukan izin ini untuk menggunakan kunci KMS, yang digunakan untuk enkripsi bucket Amazon S3 asal..

Konfigurasi kebijakan bucket Amazon S3 memungkinkan distribusi mengambil objek CloudFront terenkripsi untuk pengiriman konten.

Untuk memverifikasi izin bucket Amazon S3 dan kunci KMS

1. Verifikasi bahwa kunci KMS yang Anda gunakan adalah kunci yang sama dengan yang digunakan bucket Amazon S3 Anda untuk enkripsi default. Untuk informasi selengkapnya, lihat [Menentukan enkripsi sisi server dengan AWS KMS \(SSE-KMS\)](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.
2. Verifikasi bahwa objek dalam bucket dienkripsi dengan kunci KMS yang sama. Anda dapat memilih objek apa pun dari bucket Amazon S3 dan memeriksa pengaturan enkripsi sisi server untuk memverifikasi ARN kunci KMS.
3. Edit kebijakan bucket Amazon S3 untuk memberikan CloudFront izin memanggil operasi `GetObject` API dari bucket Amazon S3. Untuk contoh kebijakan bucket Amazon S3 yang menggunakan kontrol akses asal, lihat. [Berikan CloudFront izin untuk mengakses bucket S3](#)
4. Edit kebijakan kunci KMS untuk memberikan CloudFront izin untuk melakukan tindakan `keEncrypt`, `Decrypt`, dan `GenerateDataKey*`. Untuk menyelaraskan dengan izin hak istimewa terkecil, tentukan `Condition` elemen sehingga hanya CloudFront distribusi yang ditentukan yang dapat melakukan tindakan yang tercantum. Anda dapat menyesuaikan kebijakan untuk AWS KMS kebijakan yang ada. Untuk contoh kebijakan kunci KMS, lihat. [SSE-KMS](#)

Jika Anda menggunakan identitas akses asal (OAI) alih-alih OAC, izin ke bucket Amazon S3 sedikit berbeda karena Anda memberikan izin ke identitas, bukan. Layanan AWS Untuk informasi selengkapnya, lihat [Berikan izin identitas akses asal untuk membaca file di bucket Amazon S3](#).

Jika Anda masih tidak dapat melihat file dalam distribusi, lihat [Saya tidak dapat melihat file dalam distribusi saya](#).

Anda menggunakan kredensial yang tidak valid atau tidak memiliki izin yang memadai

Pesan galat Access Denied dapat muncul jika Anda menggunakan AWS SCT kredensial yang salah atau kedaluwarsa (kunci akses dan kunci rahasia) atau peran IAM atau pengguna Anda tidak memiliki izin yang diperlukan untuk melakukan tindakan pada sumber daya. CloudFront Untuk informasi selengkapnya tentang pesan kesalahan akses ditolak, lihat [Memecahkan masalah akses ditolak pesan kesalahan](#) di Panduan Pengguna IAM.

Untuk informasi tentang cara kerja IAM CloudFront, lihat [Identity and Access Management untuk Amazon CloudFront](#).

CloudFront mengembalikan InvalidViewerCertificate kesalahan ketika saya mencoba menambahkan nama domain alternatif

Jika CloudFront menampilkan InvalidViewerCertificate kesalahan saat Anda mencoba menambahkan nama domain alternatif (CNAME) ke distribusi Anda, tinjau informasi berikut untuk membantu memecahkan masalah. Kesalahan ini dapat menunjukkan bahwa salah satu masalah berikut ini harus diselesaikan sebelum Anda dapat menambahkan nama domain alternatif dengan sukses.

Kesalahan berikut tercantum dalam urutan CloudFront pemeriksaan otorisasi untuk menambahkan nama domain alternatif. Ini dapat membantu Anda memecahkan masalah karena berdasarkan kesalahan yang CloudFront muncul, Anda dapat mengetahui pemeriksaan verifikasi mana yang berhasil diselesaikan.

Tidak ada sertifikat terlampir pada distribusi Anda.

Untuk menambahkan nama domain alternatif (CNAME), Anda harus melampirkan sertifikat yang valid dan tepercaya ke distribusi Anda. Silakan tinjau persyaratan, dapatkan sertifikat yang valid yang memenuhinya, lampirkan ke distribusi Anda, kemudian coba lagi. Untuk informasi selengkapnya, lihat [Persyaratan untuk menggunakan nama domain alternatif](#).

Ada terlalu banyak sertifikat dalam rantai sertifikat untuk sertifikat yang Anda lampirkan.

Anda hanya dapat memiliki hingga lima sertifikat dalam rantai sertifikat. Kurangi jumlah sertifikat dalam rantai, kemudian coba lagi.

Rantai sertifikat mencakup satu atau beberapa sertifikat yang tidak berlaku untuk tanggal saat ini.

Rantai sertifikat untuk sertifikat yang telah Anda tambahkan memiliki satu atau beberapa sertifikat yang tidak valid, baik karena sertifikat belum valid atau sertifikat telah kedaluwarsa. Periksa kolom Tidak Valid Sebelum dan Tidak Valid Setelah di sertifikat dalam rantai sertifikat Anda untuk memastikan bahwa semua sertifikat valid berdasarkan tanggal yang Anda cantumkan.

Sertifikat yang Anda lampirkan tidak ditandatangani oleh Otoritas Sertifikat (CA) tepercaya.

Sertifikat yang Anda lampirkan CloudFront untuk memverifikasi nama domain alternatif tidak dapat berupa sertifikat yang ditandatangani sendiri. Informasi tersebut harus ditandatangani oleh CA tepercaya. Untuk informasi selengkapnya, lihat [Persyaratan untuk menggunakan nama domain alternatif](#).

Sertifikat yang Anda lampirkan tidak diformat dengan benar

Nama domain dan format alamat IP yang disertakan dalam sertifikat, dan format sertifikat itu sendiri, harus mengikuti standar sertifikat.

Ada kesalahan CloudFront internal.

CloudFront diblokir oleh masalah internal dan tidak dapat melakukan pemeriksaan validasi untuk sertifikat. Dalam skenario ini, CloudFront mengembalikan kode status HTTP 500 dan menunjukkan bahwa ada CloudFront masalah internal dengan melampirkan sertifikat. Tunggu beberapa menit, lalu coba lagi untuk menambahkan nama domain alternatif dengan sertifikat.

Sertifikat yang Anda lampirkan tidak mencakup nama domain alternatif yang Anda coba tambahkan.

Untuk setiap nama domain alternatif yang Anda tambahkan, CloudFront Anda harus melampirkan SSL/TLS sertifikat yang valid dari Otoritas Sertifikat (CA) tepercaya yang mencakup nama domain, untuk memvalidasi otorisasi Anda untuk menggunakannya. Harap perbarui sertifikat Anda untuk menyertakan nama domain yang mencakup CNAME yang Anda coba tambahkan. Untuk informasi lebih lanjut dan contoh penggunaan nama domain dengan wildcard, lihat [Persyaratan untuk menggunakan nama domain alternatif](#).

CloudFront mengembalikan kesalahan catatan DNS yang tidak dikonfigurasi dengan benar ketika saya mencoba menambahkan CNAME baru

Bila Anda memiliki entri DNS wildcard yang menunjuk ke CloudFront distribusi, jika Anda mencoba menambahkan CNAME baru dengan nama yang lebih spesifik, Anda mungkin mengalami kesalahan berikut:

```
One or more aliases specified for the distribution includes an incorrectly configured DNS record that points to another CloudFront distribution. You must update the DNS record to correct the problem.
```

Kesalahan ini terjadi karena CloudFront kueri DNS terhadap CNAME dan entri DNS wildcard menyelesaikan distribusi lain.

Untuk mengatasi ini, pertama buat distribusi lain, lalu buat entri DNS yang menunjuk ke distribusi baru. Terakhir, tambahkan CNAME yang lebih spesifik. Untuk informasi selengkapnya tentang cara menambahkan CNAMEs, lihat [Tambahkan nama domain alternatif](#).

Saya tidak dapat melihat file dalam distribusi saya

Jika Anda tidak dapat melihat file dalam CloudFront distribusi Anda, lihat topik berikut untuk beberapa solusi umum.

Apakah Anda mendaftar untuk keduanya CloudFront dan Amazon S3?

Untuk menggunakan Amazon CloudFront dengan asal Amazon S3, Anda harus mendaftar untuk keduanya CloudFront dan Amazon S3, secara terpisah. Untuk informasi selengkapnya tentang mendaftar CloudFront dan Amazon S3, lihat [Siapkan Akun AWS](#)

Apakah izin bucket dan objek Amazon S3 Anda disetel dengan benar?

Jika Anda menggunakan CloudFront dengan asal Amazon S3, versi asli konten Anda disimpan dalam bucket S3. Untuk menayangkan konten kepada pemirsa, sebaiknya gunakan CloudFront Origin Access Control (OAC) untuk mengamankan akses bucket Amazon S3. Ini berarti bucket S3 Anda hanya dapat dijangkau melalui CloudFront OAC mengontrol akses penampil dan pengiriman aman melalui CloudFront. Untuk informasi lebih lanjut tentang OAC, lihat [the section called “Batasi akses ke asal Amazon S3”](#).

Untuk informasi selengkapnya tentang mengelola akses bucket, lihat [Memblokir akses publik ke penyimpanan Amazon S3 di Panduan Pengguna Amazon S3](#).

Properti objek dan properti bucket bersifat independen. Anda harus secara eksplisit memberikan hak istimewa ke setiap objek di Amazon S3. Objek tidak mewarisi properti dari bucket, dan properti objek harus ditetapkan secara terpisah dari bucket.

Apakah nama domain alternatif Anda (CNAME) dikonfigurasi dengan benar?

Jika Anda sudah memiliki catatan CNAME untuk nama domain Anda, perbarui catatan tersebut atau ganti dengan yang baru yang menunjuk ke nama domain distribusi Anda.

Selain itu, pastikan CNAME Anda menyimpan poin ke nama domain distribusi Anda, bukan bucket Amazon S3. Anda dapat mengonfirmasi bahwa catatan CNAME di sistem DNS Anda menunjukkan nama domain distribusi Anda. Untuk melakukannya, gunakan alat DNS seperti dig.

Contoh berikut menunjukkan permintaan penggalian untuk nama domain yang disebut `images.example.com` dan bagian terkait dari tanggapan. Di bawah ANSWER SECTION, lihat baris yang berisi CNAME. Catatan CNAME untuk nama domain Anda diatur dengan benar jika nilai di sisi kanan CNAME adalah nama domain CloudFront distribusi Anda. Jika itu adalah bucket server asal Amazon S3 atau beberapa nama domain lainnya, maka catatan CNAME diatur dengan tidak benar.


```
[prompt]> dig images.example.com

; <<> DiG 9.3.3rc2 <<> images.example.com
;; global options: printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15917
;; flags: qr rd ra; QUERY: 1, ANSWER: 9, AUTHORITY: 2, ADDITIONAL: 0
;; QUESTION SECTION:
;images.example.com.      IN      A
;; ANSWER SECTION:
images.example.com. 10800 IN CNAME d111111abcdef8.cloudfront.net.
...
...
```

Untuk informasi lebih lanjut tentang CNAMEs, lihat [Gunakan URL kustom dengan menambahkan nama domain alternatif \(CNames\)](#).

Apakah Anda mereferensikan URL yang benar untuk CloudFront distribusi Anda?

Pastikan URL yang Anda referensikan menggunakan nama domain (atau CNAME) CloudFront distribusi Anda, bukan bucket Amazon S3 atau asal kustom Anda.

Apakah Anda memerlukan bantuan untuk memecahkan masalah asal kustom?

Jika Anda AWS perlu membantu Anda memecahkan masalah asal kustom, kami mungkin perlu memeriksa entri X-Amz-Cf-Id header dari permintaan Anda. Jika Anda belum mencatat entri ini, Anda mungkin ingin mempertimbangkannya untuk masa depan. Untuk informasi selengkapnya, lihat [the section called “Gunakan Amazon EC2 \(atau asal kustom lainnya\)”](#). Untuk bantuan lebih lanjut, lihat [Pusat Dukungan AWS](#).

Pesan galat: Sertifikat: <certificate-id>sedang digunakan oleh CloudFront

Masalah: Anda mencoba menghapus sertifikat SSL/TLS dari toko sertifikat IAM, dan Anda mendapatkan pesan “Sertifikat: <certificate-id>sedang digunakan oleh.” CloudFront

Solusi: Setiap CloudFront distribusi harus dikaitkan dengan CloudFront sertifikat default atau dengan SSL/TLS certificate. Before you can delete an SSL/TLS certificate, you must either rotate the certificate (replace the current custom SSL/TLS certificate with another custom SSL/TLS certificate) or revert from using a custom SSL/TLS sertifikat khusus untuk menggunakan CloudFront sertifikat default. Untuk memperbaikinya, selesaikan langkah-langkah di salah satu prosedur berikut:

- [Putar SSL/TLS sertifikat](#)
- [Kembalikan dari SSL/TLS sertifikat kustom ke sertifikat default CloudFront](#)

Memecahkan masalah kode status respons kesalahan di CloudFront

Jika CloudFront meminta objek dari asal Anda, dan asal mengembalikan kode status HTTP 4xx atau 5xx, ada masalah dengan komunikasi antara CloudFront dan asal Anda.

Topik ini juga mencakup langkah-langkah pemecahan masalah untuk kode status ini saat menggunakan Lambda @Edge atau Functions. CloudFront

Topik berikut memberikan penjelasan rinci tentang penyebab potensial di balik respons kesalahan ini dan menawarkan step-by-step panduan tentang cara mendiagnosis dan menyelesaikan masalah yang mendasarinya.

Topik

- [Kode status HTTP 400 \(Permintaan Buruk\)](#)
- [Kode status HTTP 401 \(Tidak Sah\)](#)
- [Kode status HTTP 403 \(Metode tidak valid\)](#)
- [Kode status HTTP 403 \(Izin Ditolak\)](#)
- [Kode status HTTP 404 \(Tidak Ditemukan\)](#)
- [Kode status HTTP 412 \(Prasyarat Gagal\)](#)
- [Kode status HTTP 500 \(Kesalahan Server Internal\)](#)
- [Kode status HTTP 502 \(Gerbang Buruk\)](#)
- [Kode status HTTP 503 \(Layanan Tidak Tersedia\)](#)
- [Kode status HTTP 504 \(Gateway Timeout\)](#)

Kode status HTTP 400 (Permintaan Buruk)

CloudFront mengembalikan 400 permintaan buruk ketika klien mengirim beberapa data yang tidak valid dalam permintaan seperti konten yang hilang atau salah dalam payload atau parameter. Ini juga bisa mewakili kesalahan klien generik.

Asal Amazon S3 mengembalikan kesalahan 400

Jika Anda menggunakan asal Amazon S3 dengan distribusi Anda, CloudFront distribusi Anda mungkin mengirim respons kesalahan dengan kode status HTTP 400 Permintaan Buruk, dan pesan yang mirip dengan berikut ini:

Header otorisasi salah bentuk; wilayah '<AWS Region>' salah; mengharapkan " <AWS Region>

Contoh:

Header otorisasi salah bentuk; wilayah 'us-east-1' salah; mengharapkan 'us-west-2'

Masalah ini dapat terjadi pada skenario berikut:

1. Asal CloudFront distribusi Anda adalah ember Amazon S3.
2. Anda memindahkan ember S3 dari satu AWS Wilayah ke Wilayah lainnya. Artinya, Anda menghapus bucket S3, lalu Anda membuat bucket baru dengan nama bucket yang sama, tetapi di AWS Wilayah yang berbeda dari tempat bucket S3 asli berada.

Untuk memperbaiki kesalahan ini, perbarui CloudFront distribusi Anda sehingga menemukan bucket S3 di AWS Wilayah bucket saat ini.

Untuk memperbarui CloudFront distribusi Anda

1. Masuk ke Konsol Manajemen AWS dan buka CloudFront konsol di <https://console.aws.amazon.com/cloudfront/v4/home>.
2. Pilih distribusi yang menghasilkan kesalahan ini.
3. Pilih Grup Asal dan Asal.
4. Temukan asal bucket S3 yang Anda pindahkan. Pilih kotak centang di samping asal ini, lalu pilih Edit.
5. Pilih Ya, Edit. Anda tidak perlu mengubah pengaturan apa pun sebelum memilih Ya, Edit.

Saat Anda menyelesaikan langkah-langkah ini, CloudFront pindahkan distribusi Anda. Saat distribusi diterapkan, Anda melihat status Deploying di bawah kolom Terakhir dimodifikasi. Beberapa saat setelah penerapan selesai, Anda harus berhenti menerima respons `AuthorizationHeaderMalformed` kesalahan.

Asal Application Load Balancer mengembalikan kesalahan 400

Jika Anda menggunakan asal Application Load Balancer dengan CloudFront distribusi Anda, kemungkinan penyebab kesalahan 400 termasuk yang berikut:

- Klien mengirim permintaan cacat yang tidak memenuhi spesifikasi HTTP.
- Header permintaan melebihi 16 KB per baris permintaan, 16 KB per header tunggal, atau 64 KB untuk seluruh header permintaan.
- Klien menutup koneksi sebelum mengirim badan permintaan lengkap.

Kode status HTTP 401 (Tidak Sah)

Kode status respons 401 Tidak Sah menunjukkan bahwa permintaan klien belum selesai karena tidak memiliki kredensial otentikasi yang valid untuk sumber daya yang diminta. Kode status ini dikirim dengan header WWW-Authenticate respons HTTP yang berisi informasi tentang bagaimana klien dapat meminta sumber daya lagi setelah meminta pengguna untuk kredensial otentikasi. Untuk informasi lebih lanjut, lihat [401 Tidak Sah](#).

Dalam CloudFront, jika asal Anda mengharapkan Authorization header untuk mengautentikasi permintaan, CloudFront perlu meneruskan Authorization header ke asal untuk menghindari kesalahan 401 Tidak Sah. Saat CloudFront meneruskan permintaan penampil ke asal Anda, CloudFront hapus beberapa header penampil secara default, termasuk header. Authorization Untuk memastikan bahwa asal Anda selalu menerima header Authorization di permintaan asal, Anda memiliki opsi berikut:

- Tambahkan Authorization header ke kunci cache dengan menggunakan kebijakan cache. Semua header di kunci cache secara otomatis disertakan dalam permintaan asal. Untuk informasi selengkapnya, lihat [Mengontrol kunci cache dengan kebijakan](#).
- Gunakan kebijakan permintaan asal yang meneruskan semua header penampil ke asal. Anda tidak dapat meneruskan Authorization header satu per satu dalam kebijakan permintaan asal, tetapi saat Anda meneruskan semua header penampil, CloudFront sertakan Authorization header dalam permintaan penampil. CloudFront menyediakan kebijakan permintaan AllViewer asal terkelola untuk kasus penggunaan ini. Untuk informasi selengkapnya, lihat [Gunakan kebijakan permintaan asal terkelola](#).

Untuk informasi selengkapnya, [lihat Bagaimana cara mengonfigurasi CloudFront untuk meneruskan header Otorisasi ke asal?](#)

Kode status HTTP 403 (Metode tidak valid)

CloudFront mengembalikan kesalahan 403 (Metode tidak valid) jika Anda mencoba menggunakan metode HTTP yang belum Anda tentukan dalam distribusi. CloudFront Anda dapat menentukan salah satu opsi berikut untuk distribusi Anda:

- CloudFront hanya ke depan GET dan HEAD permintaan.
- CloudFront hanya ke depan GET, HEAD, dan OPTIONS permintaan.
- CloudFront ke depan GET, HEAD, OPTIONS, PUT, PATCH, POST, dan DELETE permintaan. (Jika memilih opsi ini, Anda mungkin perlu membatasi akses ke bucket Amazon S3 atau custom origin agar pengguna tidak dapat melakukan operasi yang tidak Anda inginkan. Misalnya, Anda mungkin tidak ingin pengguna memiliki izin untuk menghapus objek dari asal Anda.

Kode status HTTP 403 (Izin Ditolak)

Kesalahan HTTP 403 berarti klien tidak berwenang untuk mengakses sumber daya yang diminta. Klien memahami permintaan tersebut, tetapi tidak dapat mengotorisasi akses penampil. Berikut ini adalah penyebab umum ketika CloudFront mengembalikan kode status ini:

Topik

- [CNAME alternatif tidak dikonfigurasi dengan benar](#)
- [AWS WAF dikonfigurasi pada CloudFront distribusi atau di asal](#)
- [Asal kustom mengembalikan kesalahan 403](#)
- [Asal Amazon S3 mengembalikan kesalahan 403](#)
- [Pembatasan geografis mengembalikan kesalahan 403](#)
- [URL yang ditandatangani atau konfigurasi cookie yang ditandatangani mengembalikan kesalahan 403](#)
- [Distribusi bertumpuk menyebabkan kesalahan 403](#)

CNAME alternatif tidak dikonfigurasi dengan benar

Verifikasi bahwa Anda menentukan CNAME yang benar untuk distribusi kami. Untuk menggunakan CNAME alternatif, bukan CloudFront URL default:

1. Buat catatan CNAME di DNS Anda untuk mengarahkan CNAME ke CloudFront URL distribusi.

2. Tambahkan CNAME dalam konfigurasi CloudFront distribusi Anda.

Jika Anda membuat catatan DNS tetapi tidak menambahkan CNAME dalam konfigurasi CloudFront distribusi Anda, maka permintaan akan mengembalikan kesalahan 403. Untuk informasi selengkapnya tentang mengonfigurasi CNAME kustom, lihat. [Gunakan URL kustom dengan menambahkan nama domain alternatif \(CNames\)](#)

AWS WAF dikonfigurasi pada CloudFront distribusi atau di asal

Ketika AWS WAF berada di antara klien dan CloudFront, tidak CloudFront dapat membedakan antara kode kesalahan 403 yang dikembalikan oleh asal Anda dan kode kesalahan 403 yang dikembalikan AWS WAF ketika permintaan diblokir.

Untuk menemukan sumber kode status 403, periksa aturan daftar kontrol akses AWS WAF web (ACL) Anda untuk permintaan yang diblokir. Untuk informasi selengkapnya, lihat topik berikut:

- [AWS WAF daftar kontrol akses web \(webACLs\)](#)
- [Menguji dan menyetel perlindungan Anda AWS WAF](#)

Asal kustom mengembalikan kesalahan 403

Jika Anda menggunakan custom origin, Anda mungkin melihat error 403 jika Anda memiliki konfigurasi firewall kustom di asal. Untuk memecahkan masalah, buat permintaan langsung ke asal. Jika Anda dapat mereplikasi kesalahan tanpa CloudFront, maka asal menyebabkan kesalahan 403.

Jika asal kustom menyebabkan kesalahan, periksa log asal untuk mengidentifikasi apa yang mungkin menyebabkan kesalahan. Untuk informasi selengkapnya, lihat topik pemecahan masalah berikut:

- [Bagaimana cara memecahkan masalah kesalahan HTTP 403 dari API Gateway?](#)
- [Bagaimana cara mengatasi kesalahan terlarang Application Load Balancer HTTP 403?](#)

Asal Amazon S3 mengembalikan kesalahan 403

Anda mungkin melihat kesalahan 403 karena alasan berikut:

- CloudFront tidak memiliki akses ke bucket Amazon S3. Ini dapat terjadi jika identitas akses asal (OAI) atau kontrol akses asal (OAC) tidak diaktifkan untuk distribusi Anda dan bucket bersifat pribadi.

- Jalur yang ditentukan dalam URL yang diminta tidak benar.
- Objek yang diminta tidak ada.
- Header host diteruskan dengan titik akhir REST API. Untuk informasi selengkapnya, lihat [spesifikasi bucket header HTTP Host](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.
- Anda mengkonfigurasi halaman kesalahan kustom. Untuk informasi selengkapnya, lihat [Bagaimana CloudFront proses kesalahan ketika Anda telah mengkonfigurasi halaman kesalahan kustom](#).

Pembatasan geografis mengembalikan kesalahan 403

Jika Anda mengaktifkan pembatasan geografis (juga dikenal sebagai geoblocking) untuk mencegah pengguna di lokasi geografis tertentu mengakses konten yang Anda distribusikan melalui CloudFront distribusi, pengguna yang diblokir akan menerima kesalahan 403.

Untuk informasi selengkapnya, lihat [Batasi distribusi geografis konten Anda](#).

URL yang ditandatangani atau konfigurasi cookie yang ditandatangani mengembalikan kesalahan 403

Jika Anda mengaktifkan Batasi akses penampil untuk konfigurasi perilaku distribusi Anda, permintaan yang tidak menggunakan cookie yang ditandatangani atau ditandatangani URLs menghasilkan kesalahan 403. Untuk informasi selengkapnya, lihat topik berikut:

- [Sajikan konten pribadi dengan URL yang ditandatangani dan cookie yang ditandatangani](#)
- [Bagaimana cara memecahkan masalah yang terkait dengan URL yang ditandatangani atau cookie yang ditandatangani? CloudFront](#)

Distribusi bertumpuk menyebabkan kesalahan 403

Jika Anda memiliki dua atau lebih distribusi dalam rantai permintaan ke titik akhir asal, CloudFront mengembalikan kesalahan 403. Kami tidak menyarankan menempatkan satu distribusi di depan yang lain.

Kode status HTTP 404 (Tidak Ditemukan)

CloudFront mengembalikan kesalahan 404 (Tidak Ditemukan) saat klien mencoba mengakses sumber daya yang tidak ada. Jika Anda menerima kesalahan ini dengan CloudFront distribusi Anda, penyebab umum termasuk yang berikut:

- Sumber daya tidak ada.
- URL tidak benar.
- Asal kustom mengembalikan 404.
- Halaman kesalahan kustom mengembalikan 404. (Kode kesalahan apa pun dapat diterjemahkan ke 404.) Untuk informasi selengkapnya, lihat [Bagaimana CloudFront proses kesalahan ketika Anda telah mengkonfigurasi halaman kesalahan kustom](#).
- Halaman kesalahan kustom secara tidak sengaja dihapus, menghasilkan 404 karena permintaan mencari halaman kesalahan kustom yang dihapus. Untuk informasi selengkapnya, lihat [Cara CloudFront memproses kesalahan jika Anda belum mengonfigurasi halaman kesalahan kustom](#).
- Jalur asal yang salah. Jika jalur asal diisi, nilainya ditambahkan ke jalur setiap permintaan dari browser sebelum permintaan diteruskan ke asal. Untuk informasi selengkapnya, lihat [Jalur asal](#).

Kode status HTTP 412 (Prasyarat Gagal)

CloudFront mengembalikan kode kesalahan 412 (Prasyarat Gagal) ketika akses ke sumber daya target telah ditolak. Dalam beberapa kasus, server dikonfigurasi untuk menerima permintaan hanya setelah kondisi tertentu terpenuhi. Jika salah satu kondisi yang ditentukan tidak terpenuhi, maka server tidak mengizinkan klien untuk mengakses sumber daya yang diberikan. Sebagai gantinya, server merespons dengan kode kesalahan 412.

Penyebab umum kesalahan 412 di CloudFront antaranya:

- Permintaan bersyarat pada metode selain GET atau HEAD ketika kondisi yang ditentukan oleh If-None-Match header If-Unmodified-Since atau tidak terpenuhi. Dalam hal ini, permintaan, biasanya unggahan atau modifikasi sumber daya, tidak dapat dibuat.
- Kondisi di satu atau beberapa bidang permintaan dalam operasi CloudFront [UpdateDistribution](#) API dievaluasi sebagai false.

Kode status HTTP 500 (Kesalahan Server Internal)

Kode status HTTP 500 (Internal Server Error) menunjukkan bahwa server mengalami kondisi tak terduga yang mencegahnya memenuhi permintaan. Berikut ini adalah beberapa penyebab umum dari 500 kesalahan di Amazon CloudFront.

Topik

- [Server asal mengembalikan kesalahan 500 ke CloudFront](#)

Server asal mengembalikan kesalahan 500 ke CloudFront

Server asal Anda mungkin mengembalikan kesalahan 500 ke CloudFront. Lihat topik pemecahan masalah berikut untuk informasi lebih lanjut:

- Jika Amazon S3 mengembalikan kesalahan 500, lihat [Bagaimana cara memecahkan masalah kesalahan HTTP 500 atau 503 dari Amazon S3?](#)
- Jika API Gateway menampilkan kesalahan 500, lihat [Bagaimana cara mengatasi kesalahan 5xx untuk API Gateway REST API?](#)
- Jika Elastic Load Balancing mengembalikan kesalahan 500, lihat [HTTP 500: Kesalahan server internal](#) dalam Panduan Pengguna untuk Penyeimbang Beban Aplikasi.

Jika daftar sebelumnya tidak menyelesaikan kesalahan 500, masalahnya mungkin dengan CloudFront Point of Presence yang mengembalikan kesalahan server internal. Anda dapat menghubungi [Dukungan](#) untuk bantuan.

Kode status HTTP 502 (Gerbang Buruk)

CloudFront mengembalikan kode status HTTP 502 (Bad Gateway) ketika CloudFront tidak dapat melayani objek yang diminta karena tidak dapat terhubung ke server asal.

Jika Anda menggunakan Lambda @Edge, masalahnya mungkin kesalahan validasi Lambda. Jika Anda menerima kesalahan HTTP 502 dengan kode NonS3OriginDnsError kesalahan, kemungkinan ada masalah konfigurasi DNS yang CloudFront mencegah tersambung ke asal.

Topik

- [Kegagalan negosiasi SSL/TLS antara CloudFront dan server asal kustom](#)
- [Origin tidak merespons dengan cipher/protokol yang didukung](#)

- [Sertifikat SSL/TLS pada asal kedaluwarsa, tidak valid, ditandatangani sendiri, atau rantai sertifikat dalam urutan yang salah](#)
- [Origin tidak merespons pada port tertentu dalam pengaturan asal](#)
- [Kesalahan validasi Lambda](#)
- [CloudFront kesalahan validasi fungsi](#)
- [Kesalahan DNS \(\) NonS3OriginDnsError](#)
- [Kesalahan asal Application Load Balancer 502](#)
- [Kesalahan 502 asal API Gateway](#)

Kegagalan negosiasi SSL/TLS antara CloudFront dan server asal kustom

Jika Anda menggunakan custom origin yang memerlukan HTTPS antara CloudFront dan asal Anda, nama domain yang tidak cocok dapat menyebabkan kesalahan. SSL/TLS Sertifikat asal Anda harus menyertakan nama domain yang cocok dengan Domain Asal yang Anda tentukan untuk CloudFront distribusi atau Host header permintaan asal.

Jika nama domain tidak cocok, SSL/TLS jabat tangan gagal, dan CloudFront mengembalikan kode status HTTP 502 (Bad Gateway) dan menetapkan X-Cache header ke `Error from cloudfront`

Untuk menentukan apakah nama domain dalam sertifikat cocok dengan Domain Asal dalam distribusi atau Host header, Anda dapat menggunakan pemeriksa SSL online atau OpenSSL. Jika nama domain tidak cocok, Anda memiliki dua opsi:

- Dapatkan SSL/TLS sertifikat baru yang menyertakan nama domain yang berlaku.

Jika Anda menggunakan AWS Certificate Manager (ACM), lihat [Meminta sertifikat publik](#) di Panduan AWS Certificate Manager Pengguna untuk meminta sertifikat baru.

- Ubah konfigurasi distribusi sehingga CloudFront tidak lagi mencoba menggunakan SSL untuk terhubung dengan asal Anda.

Pemeriksa SSL online

Untuk menemukan alat uji SSL, cari “pemeriksaan ssl online” di internet. Biasanya, Anda menentukan nama domain Anda, dan alat mengembalikan berbagai informasi tentang SSL/TLS sertifikat Anda. Konfirmasikan bahwa sertifikat berisi nama domain Anda di Nama Umum atau Nama Alternatif Subjek bidang.

OpenSSL

Untuk membantu memecahkan masalah kesalahan HTTP 502 CloudFront, Anda dapat menggunakan OpenSSL untuk mencoba membuat koneksi ke server asal Anda. SSL/TLS Jika OpenSSL tidak dapat membuat koneksi, itu dapat menunjukkan masalah dengan konfigurasi SSL/TLS server asal Anda. Jika OpenSSL dapat membuat koneksi, ia mengembalikan informasi tentang sertifikat server asal, termasuk nama umum sertifikat Subject CN (bidang) dan nama Subject Alternative Name alternatif subjek (bidang).

Gunakan perintah OpenSSL berikut untuk menguji koneksi ke server asal Anda (*origin domain* ganti dengan nama domain server asal Anda, seperti example.com):

```
openssl s_client -connect origin domain name:443
```

Jika yang berikut ini benar:

- Server asal Anda mendukung beberapa nama domain dengan beberapa sertifikat SSL/TLS
- Distribusi Anda dikonfigurasi untuk meneruskan Host header ke asal

Kemudian tambahkan `-servername` opsi ke perintah OpenSSL, seperti pada contoh berikut (*CNAME* ganti dengan CNAME yang dikonfigurasi dalam distribusi Anda):

```
openssl s_client -connect origin domain name:443 -servername CNAME
```

Origin tidak merespons dengan cipher/protokol yang didukung

CloudFront terhubung ke server asal menggunakan cipher dan protokol. Untuk daftar cipher dan protokol yang CloudFront mendukung, lihat [the section called “Protokol dan cipher yang didukung antara dan asal CloudFront”](#). Jika asal Anda tidak merespons dengan salah satu cipher atau protokol ini di bursa SSL/TLS, gagal terhubung. CloudFront [Anda dapat memvalidasi bahwa asal Anda mendukung cipher dan protokol dengan menggunakan alat online seperti SSL Labs](#). Ketikkan nama domain asal Anda di Nama host, lalu pilih Kirim. Meninjau Nama umum dan Nama alternatif bidang dari pengujian untuk melihat apakah sesuai dengan nama domain asal Anda. Setelah uji selesai, temukan Protokol dan Cipher Suites bagian dalam hasil uji untuk melihat cipher atau protokol mana yang didukung oleh asal Anda. Bandingkan dengan daftar [the section called “Protokol dan cipher yang didukung antara dan asal CloudFront”](#).

Sertifikat SSL/TLS pada asal kedaluwarsa, tidak valid, ditandatangani sendiri, atau rantai sertifikat dalam urutan yang salah

Jika server asal mengembalikan berikut ini, CloudFront menjatuhkan koneksi TCP, mengembalikan kode status HTTP 502 (Bad Gateway), dan menetapkan X-Cache header ke: `Error from cloudfront`

- Sertifikat yang kedaluwarsa
- Sertifikat tidak valid
- Sertifikat yang ditandatangani sendiri
- Rantai sertifikat dalam urutan yang salah

 Note

Jika rantai lengkap sertifikat, termasuk sertifikat perantara, tidak ada, CloudFront jatuhkan koneksi TCP.

Untuk informasi tentang menginstal SSL/TLS sertifikat di server asal kustom Anda, lihat [the section called “Memerlukan HTTPS ke custom origin”](#).

Origin tidak merespons pada port tertentu dalam pengaturan asal

Ketika Anda membuat asal pada CloudFront distribusi Anda, Anda dapat mengatur port yang CloudFront terhubung ke asal dengan untuk lalu lintas HTTP dan HTTPS. Secara default, ini adalah TCP 80/443. Anda memiliki opsi untuk mengubah port ini. Jika asal Anda menolak lalu lintas pada port ini karena alasan apa pun, atau jika server backend Anda tidak merespons pada port, CloudFront akan gagal terhubung.

Untuk memecahkan masalah ini, periksa firewall apa pun yang berjalan di infrastruktur Anda dan pastikan firewall tersebut tidak menghalangi rentang IP yang didukung. Untuk informasi selengkapnya, lihat [rentang alamat AWS IP](#) di Panduan Pengguna Amazon VPC. Selain itu, verifikasi apakah server web Anda berjalan di tempat asalnya.

Kesalahan validasi Lambda

Jika Anda menggunakan Lambda @Edge, kode status HTTP 502 dapat menunjukkan bahwa respons fungsi Lambda Anda salah dibentuk atau menyertakan konten yang tidak valid. Untuk

informasi selengkapnya tentang pemecahan masalah kesalahan Lambda@Edge, lihat [Uji dan debug fungsi Lambda @Edge](#).

CloudFront kesalahan validasi fungsi

Jika Anda menggunakan CloudFront fungsi, kode status HTTP 502 dapat menunjukkan bahwa CloudFront fungsi tersebut mencoba menambahkan, menghapus, atau mengubah header hanya-baca. Kesalahan ini tidak muncul selama pengujian, tetapi akan muncul setelah Anda menerapkan fungsi dan menjalankan permintaan. Untuk mengatasi kesalahan ini, periksa dan perbarui CloudFront fungsi Anda. Untuk informasi selengkapnya, lihat [Perbarui fungsi](#).

Kesalahan DNS () NonS3originDnsError

Kesalahan HTTP 502 dengan kode NonS3originDnsError kesalahan menunjukkan bahwa ada masalah konfigurasi DNS yang CloudFront mencegah koneksi ke asal. Jika Anda mendapatkan kesalahan ini CloudFront, pastikan konfigurasi DNS asal sudah benar dan berfungsi.

Ketika CloudFront menerima permintaan untuk objek yang kedaluwarsa atau tidak dalam cache, itu membuat permintaan ke asal untuk mendapatkan objek. Untuk membuat permintaan yang berhasil ke asal, CloudFront lakukan resolusi DNS pada domain asal. Jika layanan DNS untuk domain Anda mengalami masalah, tidak CloudFront dapat menyelesaikan nama domain untuk mendapatkan alamat IP, yang menghasilkan kesalahan HTTP 502 ()NonS3originDnsError. Untuk memperbaiki masalah ini, hubungi penyedia DNS Anda, atau, jika Anda menggunakan Amazon Route 53, lihat [Mengapa saya tidak dapat mengakses situs web saya yang menggunakan layanan DNS Route 53?](#)

Untuk mengatasi masalah ini lebih lanjut, pastikan bahwa [server nama otoritatif](#) domain akar atau puncak zona asal Anda (seperti `example.com`) berfungsi dengan benar. Anda dapat menggunakan perintah berikut untuk menemukan server nama untuk asal puncak Anda, dengan alat seperti [dig](#) atau [nslookup](#):

```
dig OriginAPEXDomainName NS +short
```

```
nslookup -query=NS OriginAPEXDomainName
```

Saat Anda memiliki nama server nama Anda, gunakan perintah berikut untuk menanyakan nama domain asal Anda terhadap server tersebut guna memastikan bahwa setiap server menjawabnya dengan jawaban:

```
dig OriginDomainName @NameServer
```

```
nslookup OriginDomainName NameServer
```

Important

Pastikan Anda melakukan pemecahan masalah DNS ini menggunakan komputer yang terhubung ke internet publik. CloudFront menyelesaikan domain asal menggunakan DNS publik di internet, jadi penting untuk memecahkan masalah dalam konteks yang sama.

Jika asal Anda adalah subdomain yang otoritas DNSNYA didelegasikan ke server nama yang berbeda dari domain root, pastikan bahwa catatan name server (NS) dan start of authority (SOA) dikonfigurasi dengan benar untuk subdomain. Anda dapat memeriksa catatan ini menggunakan perintah yang mirip dengan contoh sebelumnya.

Untuk informasi selengkapnya tentang DNS, lihat [konsep Sistem Nama Domain \(DNS\) dalam dokumentasi](#) Amazon Route 53.

Kesalahan asal Application Load Balancer 502

Jika Anda menggunakan Application Load Balancer sebagai asal Anda dan menerima kesalahan 502, lihat [Bagaimana cara mengatasi kesalahan Application Load Balancer HTTP 502?](#) .

Kesalahan 502 asal API Gateway

Jika Anda menggunakan API Gateway dan menerima kesalahan 502, lihat [Bagaimana cara mengatasi kesalahan HTTP 502 dari API Gateway REST dengan integrasi proxy APIs Lambda?](#) .

Kode status HTTP 503 (Layanan Tidak Tersedia)

Kode status HTTP 503 (Layanan Tidak Tersedia) biasanya menunjukkan masalah kinerja pada server asal. Dalam kasus yang jarang terjadi, ini menunjukkan bahwa CloudFront sementara tidak dapat memenuhi permintaan karena kendala sumber daya di lokasi tepi.

Jika Anda menggunakan Lambda @Edge atau CloudFront Functions, masalahnya mungkin kesalahan eksekusi atau kesalahan Lambda @Edge limit exceeded.

Jika Anda menggunakan Origin Mutual TLS (origin mTLS), masalah ini mungkin disebabkan oleh upaya gagal berulang untuk membuat koneksi dengan sertifikat tertentu.

Topik

- [Server asal tidak memiliki kapasitas yang cukup untuk mendukung tingkat permintaan](#)
- [CloudFront menyebabkan kesalahan karena kendala sumber daya di lokasi tepi](#)
- [CloudFront menyebabkan kesalahan untuk asal yang dikonfigurasi dengan TLS timbal balik](#)
- [Lambda @Edge atau Kesalahan eksekusi CloudFront Fungsi](#)
- [Batas Lambda @Edge terlampaui](#)

Server asal tidak memiliki kapasitas yang cukup untuk mendukung tingkat permintaan

Ketika server asal tidak tersedia atau tidak dapat melayani permintaan masuk, ia mengembalikan kode status HTTP 503 (Layanan Tidak Tersedia). CloudFront kemudian menyampaikan kesalahan kembali ke pengguna. Untuk mengatasi masalah ini, coba solusi berikut:

- Jika Anda menggunakan Amazon S3 sebagai server asal Anda:
 - Anda dapat mengirim 3.500 PUT/COPY/POST/DELETE or 5,500 GET/HEAD permintaan per detik per awalan Amazon S3 yang dipartisi. Saat Amazon S3 mengembalikan respons Perlahan 503, ini biasanya menunjukkan tingkat permintaan yang berlebihan terhadap awalan Amazon S3 tertentu.

Karena tingkat permintaan berlaku per awalan dalam bucket S3, objek harus didistribusikan di beberapa awalan. Saat tingkat permintaan pada awalan meningkat secara bertahap, Amazon S3 meningkatkan skala untuk menangani permintaan untuk setiap awalan secara terpisah. Akibatnya, tingkat permintaan keseluruhan yang ditangani bucket adalah kelipatan dari jumlah awalan.

- Untuk informasi selengkapnya, lihat [Mengoptimalkan performa Amazon S3](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.
- Jika Anda menggunakan Elastic Load Balancing sebagai server asal Anda:
 - Pastikan bahwa instance backend Anda dapat merespons pemeriksaan kesehatan.
 - Pastikan bahwa load balancer dan instance backend Anda dapat menangani beban.

Untuk informasi lebih lanjut, lihat:

- [Bagaimana cara memecahkan masalah 503 kesalahan yang dikembalikan saat menggunakan Classic Load Balancer?](#)
- [Bagaimana cara mengatasi kesalahan 503 \(layanan tidak tersedia\) dari Application Load Balancer saya?](#)
- Jika Anda menggunakan custom origin:

- Periksa log aplikasi untuk memastikan bahwa asal Anda memiliki sumber daya yang cukup, seperti memori, CPU, dan ukuran disk.
- Jika Anda menggunakan Amazon EC2 sebagai backend, pastikan bahwa jenis instans memiliki sumber daya yang sesuai untuk memenuhi permintaan masuk. Untuk informasi lebih lanjut, lihat [Tipe instans](#) di Panduan Pengguna Amazon EC2.
- Jika Anda menggunakan API Gateway:
 - Kesalahan ini terkait dengan integrasi backend saat API Gateway API tidak dapat menerima respons. Server backend mungkin:
 - Kelebihan beban melebihi kapasitas dan tidak dapat memproses permintaan klien baru.
 - Di bawah pemeliharaan sementara.
 - Untuk mengatasi kesalahan ini, lihat log aplikasi API Gateway Anda untuk menentukan apakah ada masalah dengan kapasitas backend, integrasi, atau yang lainnya.

CloudFront menyebabkan kesalahan karena kendala sumber daya di lokasi tepi

Anda akan menerima kesalahan ini dalam situasi langka yang tidak CloudFront dapat merutekan permintaan ke lokasi tepi terbaik berikutnya yang tersedia, sehingga tidak dapat memenuhi permintaan. Kesalahan ini umum terjadi saat Anda melakukan pengujian pemuatan pada CloudFront distribusi. Untuk membantu mencegah hal ini, ikuti [the section called “Pengujian beban CloudFront”](#) panduan untuk menghindari kesalahan 503 (kapasitas terlampaui).

Jika ini terjadi di lingkungan produksi Anda, hubungi [Dukungan](#).

CloudFront menyebabkan kesalahan untuk asal yang dikonfigurasi dengan TLS timbal balik

Jika distribusi Anda menggunakan TLS bersama (mTLS) untuk terhubung ke asal kustom, CloudFront dapat mengembalikan kode status HTTP 503 ketika tidak dapat membuat koneksi ke asal. Ini dapat terjadi ketika asal berulang kali gagal menyelesaikan jabat tangan TLS atau tidak dapat dijangkau karena masalah DNS atau konektivitas. Kondisi ini bersifat sementara — CloudFront secara otomatis melanjutkan lalu lintas normal setelah asal dapat dijangkau lagi.

Untuk menyelesaikan masalah ini:

- Konfirmasikan bahwa sertifikat klien yang benar dikaitkan dengan distribusi Anda dan belum kedaluwarsa. Jika Anda baru saja memperbarui sertifikat, berikan waktu untuk perubahan menyebar sebelum berlaku.

- Konfirmasikan bahwa asal dapat dijangkau dan bahwa catatan DNS diselesaikan dengan benar.

Jika masalah tetap ada meskipun telah mengonfigurasi sertifikat yang tepat di lingkungan produksi Anda, hubungi [Dukungan](#)

Lambda @Edge atau Kesalahan eksekusi CloudFront Fungsi

Jika Anda menggunakan Lambda @Edge atau CloudFront Functions, kode status HTTP 503 dapat menunjukkan bahwa fungsi Anda mengembalikan kesalahan eksekusi.

Untuk detail selengkapnya tentang cara mengidentifikasi dan mengatasi kesalahan Lambda @Edge, lihat [Uji dan debug fungsi Lambda @Edge](#)

Untuk informasi selengkapnya tentang CloudFront fungsi pengujian, lihat [Fungsi uji](#).

Batas Lambda @Edge terlampaui

Jika Anda menggunakan Lambda @Edge, kode status HTTP 503 dapat menunjukkan bahwa Lambda mengembalikan kesalahan. Kesalahan tersebut dapat disebabkan oleh salah satu hal berikut:

- Jumlah eksekusi fungsi melebihi salah satu kuota yang ditetapkan Lambda untuk membatasi eksekusi dalam (eksekusi bersamaan atau frekuensi Wilayah AWS pemanggilan).
- Fungsi melampaui kuota waktu habis fungsi Lambda.

Untuk informasi selengkapnya tentang kuota Lambda @Edge, lihat [Kuotas di Lambda@Edge](#) Untuk detail selengkapnya tentang cara mengidentifikasi dan mengatasi kesalahan Lambda @Edge, lihat [the section called “Uji dan debug”](#) Anda juga dapat melihat [kuota layanan Lambda di Panduan Pengembang](#).AWS Lambda

Kode status HTTP 504 (Gateway Timeout)

Kode status HTTP 504 (batas waktu gateway) menunjukkan bahwa ketika CloudFront meneruskan permintaan ke asal (karena objek yang diminta tidak berada di cache tepi), salah satu hal berikut terjadi:

- Asal mengembalikan kode status HTTP 504 ke CloudFront.
- Asal tidak merespons sebelum permintaan kedaluwarsa.

CloudFront akan mengembalikan kode status HTTP 504 jika lalu lintas diblokir ke asal oleh firewall atau grup keamanan, atau jika asal tidak dapat diakses di internet. Periksa masalah tersebut terlebih dahulu. Kemudian, jika akses bukan masalahnya, jelajahi penundaan aplikasi dan batas waktu server untuk membantu Anda mengidentifikasi dan memperbaiki masalah.

Topik

- [Konfigurasi firewall di server asal Anda untuk memungkinkan CloudFront lalu lintas](#)
- [Konfigurasi grup keamanan di server asal Anda untuk mengizinkan CloudFront lalu lintas](#)
- [Jadikan server asal kustom Anda dapat diakses di internet](#)
- [Temukan dan perbaiki tanggapan yang tertunda dari aplikasi di server asal Anda](#)

Konfigurasi firewall di server asal Anda untuk memungkinkan CloudFront lalu lintas

Jika firewall di server asal Anda memblokir CloudFront lalu lintas, CloudFront mengembalikan kode status HTTP 504, jadi sebaiknya pastikan itu bukan masalahnya sebelum memeriksa masalah lain.

Metode yang Anda gunakan untuk menentukan apakah masalah dengan firewall Anda bergantung pada sistem yang digunakan server asal Anda:

- Jika Anda menggunakan IPTable firewall di server Linux, Anda dapat mencari alat dan informasi untuk membantu Anda bekerja dengannya IPTables.
- Jika Anda menggunakan Windows Firewall di server Windows, lihat [Menambahkan atau Mengedit Aturan Firewall](#) di dokumentasi Microsoft.

Saat Anda mengevaluasi konfigurasi firewall di server asal Anda, cari firewall atau aturan keamanan apa pun yang memblokir lalu lintas dari lokasi CloudFront tepi, berdasarkan rentang alamat IP yang dipublikasikan. Untuk informasi selengkapnya, lihat [Lokasi dan rentang alamat IP server CloudFront edge](#).

Jika rentang alamat CloudFront IP diizinkan untuk terhubung ke server asal Anda, pastikan untuk memperbarui aturan keamanan server Anda untuk memasukkan perubahan. Anda dapat berlangganan topik Amazon SNS dan menerima pemberitahuan saat file rentang alamat IP diperbarui. Setelah menerima notifikasi, Anda dapat menggunakan kode untuk mengambil file, menguraikannya, dan melakukan penyesuaian terhadap lingkungan lokal Anda. Untuk informasi selengkapnya, lihat [Berlangganan Perubahan Alamat IP AWS Publik melalui Amazon SNS](#) di Blog AWS Berita.

Konfigurasi grup keamanan di server asal Anda untuk mengizinkan CloudFront lalu lintas

Jika asal Anda menggunakan Elastic Load Balancing, tinjau [grup keamanan ELB](#) dan pastikan grup keamanan mengizinkan lalu lintas masuk. CloudFront

Anda juga dapat menggunakan AWS Lambda untuk secara otomatis memperbarui grup keamanan Anda untuk memungkinkan lalu lintas masuk dari CloudFront.

Jadikan server asal kustom Anda dapat diakses di internet

Jika tidak CloudFront dapat mengakses server asal kustom Anda karena tidak tersedia untuk umum di internet, CloudFront mengembalikan kesalahan HTTP 504.

CloudFront lokasi tepi terhubung ke server asal melalui internet. Jika asal kustom Anda ada di jaringan pribadi, tidak CloudFront dapat mencapainya. Karena itu, Anda tidak dapat menggunakan server pribadi, termasuk [Classic Load Balancer internal](#), sebagai server asal. CloudFront

Untuk memeriksa bahwa lalu lintas internet dapat terhubung ke server asal Anda, jalankan perintah berikut (di mana *OriginDomainName* adalah nama domain untuk server Anda):

Untuk lalu lintas HTTPS:

- `nc -zv OriginDomainName 443`
- `telnet OriginDomainName 443`

Untuk lalu lintas HTTP:

- `nc -zv OriginDomainName 80`
- `telnet OriginDomainName 80`

Temukan dan perbaiki tanggapan yang tertunda dari aplikasi di server asal Anda

Waktu habis server sering kali merupakan hasil dari aplikasi yang memerlukan waktu sangat lama untuk merespons, atau nilai habis waktu yang diatur terlalu rendah.

Perbaiki cepat untuk membantu menghindari kesalahan HTTP 504 adalah dengan hanya mengatur CloudFront nilai waktu habis untuk distribusi Anda. Namun, kami menyarankan Anda untuk terlebih dahulu memastikan bahwa Anda mengatasi masalah performa dan latensi dengan aplikasi dan

server asal. Kemudian Anda dapat menetapkan nilai batas waktu yang wajar yang membantu mencegah kesalahan HTTP 504 dan memberikan respons yang baik kepada pengguna.

Berikut ikhtisar langkah-langkah yang dapat Anda ambil untuk menemukan masalah kinerja dan memperbaikinya:

1. Ukur latensi (responsifitas) beban tinggi dan tipikal dari aplikasi web Anda.
2. Tambahkan sumber daya tambahan, seperti CPU atau memori, jika diperlukan. Mengambil langkah lain untuk mengatasi masalah, seperti mengatur kueri basis data untuk mengakomodasi skenario muatan tinggi.
3. Jika perlu, sesuaikan nilai batas waktu untuk CloudFront distribusi Anda.

Berikut ini adalah rincian tentang setiap langkah.

Ukur latensi biasa dan beban tinggi

Untuk menentukan apakah satu atau lebih server aplikasi web backend mengalami latensi tinggi, jalankan perintah curl Linux berikut di setiap server:

```
curl -w "DNS Lookup Time: %{time_namelookup} \nConnect time: %{time_connect} \nTLS Setup: %{time_appconnect} \nRedirect Time: %{time_redirect} \nTime to first byte: %{time_starttransfer} \nTotal time: %{time_total} \n" -o /dev/null https://www.example.com/yourobject
```

Note

Jika Anda menjalankan Windows di server, Anda dapat mencari dan mengunduh curl bagi Windows untuk menjalankan perintah yang serupa.

Saat Anda mengukur dan mengevaluasi latensi aplikasi yang berjalan di server Anda, ingatlah hal berikut:

- Nilai latensi relatif terhadap setiap aplikasi. Namun, waktu untuk byte pertama dalam milidetik daripada detik atau lebih, masuk akal.
- Jika Anda mengukur latensi aplikasi di bawah beban normal dan tidak masalah, ketahuilah bahwa pemirsa mungkin masih mengalami batas waktu di bawah beban tinggi. Jika permintaannya tinggi, server dapat memiliki respons tertunda atau tidak merespons sama sekali. Untuk membantu mencegah masalah latensi beban tinggi, periksa sumber daya server Anda seperti CPU, memori,

dan pembacaan dan penulisan disk untuk memastikan bahwa server Anda memiliki kapasitas untuk menskalakan beban tinggi.

Anda dapat menjalankan perintah Linux berikut untuk memeriksa memori yang digunakan oleh proses Apache:

```
watch -n 1 "echo -n 'Apache Processes: ' && ps -C apache2 --no-headers | wc -l && free -m"
```

- Pemanfaatan CPU yang tinggi di server dapat secara signifikan mengurangi kinerja aplikasi. Jika Anda menggunakan instans Amazon EC2 untuk server backend Anda, tinjau CloudWatch metrik server untuk memeriksa penggunaan CPU. Untuk informasi selengkapnya, lihat [Panduan CloudWatch Pengguna Amazon](#). Atau jika Anda menggunakan server Anda sendiri, lihat dokumentasi bantuan server untuk petunjuk tentang cara memeriksa pemanfaatan CPU.
- Periksa masalah potensial lainnya di bawah beban tinggi, seperti kueri basis data yang berjalan lambat ketika ada volume permintaan yang tinggi.

Tambahkan sumber daya, dan atur server dan basis data

Setelah Anda mengevaluasi responsivitas aplikasi dan server Anda, pastikan Anda memiliki sumber daya yang memadai untuk situasi lalu lintas dan beban tinggi yang biasa terjadi:

- Jika Anda memiliki server Anda sendiri, pastikan server memiliki CPU, memori, dan ruang disk yang cukup untuk menangani permintaan penampil, berdasarkan evaluasi Anda.
- Jika Anda menggunakan instans Amazon EC2 sebagai server backend Anda, pastikan bahwa jenis instance memiliki sumber daya yang sesuai untuk memenuhi permintaan masuk. Untuk informasi lebih lanjut, lihat [Tipe instans](#) di Panduan Pengguna Amazon EC2.

Selain itu, pertimbangkan langkah-langkah penyetelan berikut untuk membantu menghindari timeout:

- Jika nilai Time to First Byte yang dikembalikan oleh perintah curl tampaknya tinggi, ambil langkah untuk meningkatkan kinerja aplikasi Anda. Meningkatkan responsivitas aplikasi secara bergantian akan membantu mengurangi kesalahan waktu habis.
- Tune kueri basis data untuk memastikan bahwa mereka dapat menangani volume permintaan yang tinggi tanpa kinerja yang lambat.
- Persiapkan [tetap-sif \(tetap\)](#) di server backend Anda. Opsi ini membantu menghindari keterlambatan yang terjadi ketika koneksi harus dibangun kembali untuk permintaan berikutnya atau pengguna.

- Jika Anda menggunakan Elastic Load Balancing sebagai asal Anda, berikut ini adalah kemungkinan penyebab kesalahan 504:
 - Penyeimbang beban gagal membuat koneksi ke target sebelum batas waktu koneksi berakhir (10 detik).
 - Penyeimbang beban membuat koneksi ke target tetapi target tidak merespons sebelum periode batas waktu idle berlalu.
 - Daftar kontrol akses jaringan (ACL) untuk subnet tidak mengizinkan lalu lintas dari target ke node penyeimbang beban pada port sementara (1024-65535).
 - Target mengembalikan header konten-panjang yang lebih besar dari isi entitas. Load balancer menghabiskan waktu menunggu byte yang hilang.
 - Targetnya adalah fungsi Lambda dan Lambda tidak merespons sebelum batas waktu koneksi berakhir.

Untuk informasi selengkapnya tentang mengurangi latensi, lihat [Bagaimana cara mengatasi masalah latensi tinggi pada ELB Classic Load Balancer saya?](#)

- Jika Anda menggunakan MediaTailor sebagai asal Anda, berikut ini adalah kemungkinan penyebab kesalahan 504:
 - Jika kerabat URLs salah penanganan, MediaTailor dapat menerima cacat URLs dari para pemain.
 - Jika MediaPackage adalah asal manifes untuk MediaTailor, MediaPackage 404 kesalahan manifes dapat MediaTailor menyebabkan mengembalikan kesalahan 504.
 - Permintaan ke server MediaTailor asal membutuhkan waktu lebih dari 2 detik untuk diselesaikan.
- Jika Anda menggunakan Amazon API Gateway sebagai asal Anda, berikut ini adalah kemungkinan penyebab kesalahan 504:
 - Permintaan integrasi membutuhkan waktu lebih lama daripada parameter batas waktu integrasi maksimum API Gateway REST API Anda. Untuk informasi selengkapnya, [lihat Bagaimana cara memecahkan masalah error batas waktu API HTTP 504 dengan API Gateway?](#)

Jika diperlukan, sesuaikan nilai CloudFront batas waktu

Jika Anda telah mengevaluasi dan menangani performa aplikasi lambat, kapasitas server asal, dan masalah lain, tetapi penampil masih mengalami kesalahan HTTP 504, maka Anda harus mempertimbangkan mengubah waktu yang ditentukan dalam distribusi Anda untuk timeout respons asal. Untuk informasi selengkapnya, lihat [the section called “Batas waktu respons”](#).

Pengujian beban CloudFront

Metode pengujian beban tradisional tidak berfungsi dengan baik CloudFront karena CloudFront menggunakan DNS untuk menyeimbangkan beban di seluruh lokasi tepi yang tersebar secara geografis dan di dalam setiap lokasi tepi. Ketika klien meminta konten dari CloudFront, klien menerima respons DNS yang mencakup satu set alamat IP. Jika Anda menguji dengan mengirimkan permintaan ke salah satu alamat IP yang dikembalikan DNS, Anda hanya menguji sebagian kecil sumber daya di satu lokasi CloudFront tepi, yang tidak secara akurat mewakili pola lalu lintas yang sebenarnya. Bergantung pada volume data yang diminta, pengujian dengan cara ini dapat membebani dan menurunkan kinerja subset kecil server itu. CloudFront

CloudFront dirancang untuk skala bagi pemirsa yang memiliki alamat IP klien yang berbeda dan resolver DNS yang berbeda di beberapa wilayah geografis. Untuk melakukan pengujian beban yang menilai CloudFront kinerja secara akurat, kami sarankan Anda melakukan semua hal berikut:

- Kirimkan permintaan klien dari beberapa wilayah geografis.
- Konfigurasi pengujian Anda sehingga setiap klien membuat permintaan DNS independen. Setiap klien kemudian akan menerima satu set alamat IP yang berbeda dari DNS.
- Untuk setiap klien yang membuat permintaan, sebarkan permintaan klien Anda di seluruh kumpulan alamat IP yang dikembalikan oleh DNS. Ini memastikan bahwa beban didistribusikan di beberapa server di lokasi CloudFront tepi.

Catatan

- Pengujian beban tidak diizinkan pada perilaku cache yang memiliki [permintaan penampil Lambda @Edge](#) atau [pemicu respons penampil](#).
- Pengujian beban tidak diizinkan pada asal yang mengaktifkan [Origin Shield](#).

Kuota

Anda dapat meminta kenaikan CloudFront kuota dengan menggunakan opsi berikut:

- Anda dapat menggunakan konsol Service Quotas atau AWS Command Line Interface Untuk informasi selengkapnya, lihat topik berikut:
 - [Meminta peningkatan kuota pada Panduan Pengguna Service Quotas](#)
 - [permintaan-layanan-kuota-peningkatan dalam Referensi Perintah AWS CLI](#)
- Jika CloudFront kuota tidak tersedia di Service Quotas, gunakan untuk membuat AWS Support Center Console kasus peningkatan [kuota layanan](#).

CloudFront tunduk pada kuota berikut.

Topik

- [Kuota umum](#)
- [Kuota umum di distribusi](#)
- [Kuotas Umum tentang Kebijakan](#)
- [Kuota pada mTL dan toko kepercayaan](#)
- [Kuota pada Fungsi CloudFront](#)
- [Kuota pada Fungsi Koneksi](#)
- [Kuota pada toko nilai utama](#)
- [Kuotas di Lambda@Edge](#)
- [Kuota pada sertifikat SSL](#)
- [Kuotas pada invalidasi](#)
- [Kuotas pada kelompok utama](#)
- [Kuota pada koneksi WebSocket](#)
- [Kuotas pada enkripsi tingkat lapangan](#)
- [Kuota pada cookie \(pengaturan cache warisan\)](#)
- [Kuota pada string kueri \(pengaturan cache warisan\)](#)
- [Kuota pada header](#)
- [Kuota pada distribusi multi-penyewa](#)
- [Informasi Terkait](#)

Kuota umum

Entitas	Kuota standar
Laju transfer data per distribusi (Kuota ini tidak berlaku untuk distribusi yang berlangganan paket harga CloudFront flat-rate. Untuk informasi lebih lanjut, lihat ???.)	150 Gbps Minta kuota yang lebih tinggi
Permintaan per detik per distribusi (Kuota ini tidak berlaku untuk distribusi yang berlangganan paket harga CloudFront flat-rate. Untuk informasi lebih lanjut, lihat ???.)	250.000 Minta kuota yang lebih tinggi
Tanda yang dapat ditambahkan ke distribusi	50 Minta kuota yang lebih tinggi
File yang dapat Anda sajikan per distribusi	Tidak ada kuota
Panjang maksimum permintaan atau respons asal, termasuk header dan string kueri, tetapi tidak termasuk isi isi	32.768 byte
Panjang maksimum URL	8.192 byte
Jumlah maksimum konfigurasi pengiriman log akses real-time per Akun AWS	150
Jumlah maksimum asosiasi per web ACL	100 Minta kuota yang lebih tinggi

Kuota umum di distribusi

Entitas	Kuota default
Nama domain alternatif (CNAME) per distribusi	100

Entitas	Kuota default
Untuk informasi selengkapnya, lihat Gunakan URL kustom dengan menambahkan nama domain alternatif (CNames) .	Minta kuota yang lebih tinggi
Perilaku Cache per distribusi	75 Minta kuota yang lebih tinggi
Upaya koneksi per asal Untuk informasi selengkapnya, lihat Upaya koneksi .	1-3
Waktu habis penerbangan lanjutan per asal Untuk informasi selengkapnya, lihat Batas waktu koneksi .	1-10 detik
Batas waktu respons per asal Ini juga dikenal sebagai batas waktu baca asal atau batas waktu permintaan asal. Untuk informasi selengkapnya, lihat Batas waktu respons .	1-120 detik Minta kuota yang lebih tinggi
Keep-alive batas waktu per asal Untuk informasi selengkapnya, lihat Keep-alive batas waktu (khusus dan hanya asal VPC) .	1-300 detik Minta kuota yang lebih tinggi
Distribusi per Akun AWS Untuk informasi selengkapnya, lihat Buat distribusi .	500 Minta kuota yang lebih tinggi
Distribusi per kontrol akses asal	100 Minta kuota yang lebih tinggi

Entitas	Kuota default
Distribusi dalam rantai permintaan ke titik akhir asal Kami tidak menyarankan menempatkan satu distribusi di depan yang lain. Melebihi kuota ini menghasilkan kesalahan 403.	2
Kompresi file: berbagai ukuran file yang CloudFront dikompres Untuk informasi selengkapnya, lihat Sajikan file terkompresi.	1.000 hingga 10.000.000 byte
Ukuran file cache maksimum per respons HTTP GET. Hanya tanggapan untuk HTTP GET yang di-cache. Tanggapan untuk POST atau PUT tidak di-cache.	50 GB
Ukuran maksimum dari badan permintaan HTTP.	64 GB
Kontrol akses asal per Akun AWS	100 Minta kuota yang lebih tinggi
Identitas akses asal per Akun AWS	100 Minta kuota yang lebih tinggi
Kota Asal per distribusi	100 Minta kuota yang lebih tinggi
Grup asal per distribusi	10 Minta kuota yang lebih tinggi

Entitas	Kuota default
Distribusi pementasan per Akun AWS	20
Untuk informasi selengkapnya, lihat the section called “Gunakan penerapan berkelanjutan untuk menguji perubahan dengan aman” .	Minta kuota yang lebih tinggi
Distribusi yang terkait dengan asal VPC yang sama	50
Asal VPC per Akun AWS	25
	Minta kuota yang lebih tinggi
Jumlah maksimum distribusi yang dapat dikaitkan dengan satu daftar IP statis Anycast.	100
	Minta kuota yang lebih tinggi

Kuotas Umum tentang Kebijakan

Entitas	Kuota default
Kebijakan cache khusus per Akun AWS	20
(Tidak berlaku untuk kebijakan cache CloudFront terkelola)	Minta kuota yang lebih tinggi
Distribusi yang terkait dengan kebijakan cache yang sama	100
Kueri string per kebijakan cache	10
	Minta kuota yang lebih tinggi
Header sesuai kebijakan cache	10
	Minta kuota yang lebih tinggi

Entitas	Kuota default
Kebijakan cookie per cache	10 Minta kuota yang lebih tinggi
Total panjang gabungan dari semua string kueri, header, dan nama cookie dalam kebijakan cache	1024
Kebijakan permintaan asal kustom per Akun AWS (Tidak berlaku untuk kebijakan permintaan asal CloudFront terkelola)	20 Minta kuota yang lebih tinggi
Distribusi yang terkait dengan kebijakan permintaan asal yang sama	100
Kueri string per kebijakan permintaan asal	10 Minta kuota yang lebih tinggi
Kebijakan permintaan header per asal	10 Minta kuota yang lebih tinggi
Kebijakan permintaan cookie per asal	10 Minta kuota yang lebih tinggi
Total panjang gabungan dari semua string kueri, header, dan nama cookie dalam kebijakan permintaan asal	1024
Kebijakan header respon kustom per Akun AWS (Tidak berlaku untuk kebijakan header respons CloudFront terkelola)	20 Minta kuota yang lebih tinggi

Entitas	Kuota default
Distribusi yang terkait dengan kebijakan header respons yang sama	100 Minta kuota yang lebih tinggi
Header kustom per kebijakan header respon	10 Minta kuota yang lebih tinggi
Kebijakan penerapan berkelanjutan per Akun AWS	20 Minta kuota yang lebih tinggi

Kuota pada mTL dan toko kepercayaan

Entitas	Kuota default
Toko kepercayaan per Akun AWS	20 Minta kuota yang lebih tinggi
Distribusi per toko kepercayaan	25
Ukuran bundel CA	64 KB Minta kuota yang lebih tinggi
Ukuran sertifikat dalam bundel CA	16384 Minta kuota yang lebih tinggi
Jumlah sertifikat dalam bundel CA	25

Entitas	Kuota default
Kedalaman rantai sertifikat	4

Kuota pada Fungsi CloudFront

Entitas	Kuota default
Fungsi per Akun AWS	100
Ukuran fungsi maksimum Kuota ini tidak dapat disesuaikan. Untuk menyimpan data tambahan untuk CloudFront Fungsi Anda, buat penyimpanan nilai kunci dan tambahkan pasangan nilai kunci Anda. Untuk informasi selengkapnya, lihat Amazon CloudFront KeyValueCollection .	10 KB
Memori fungsi maksimum	2 MB
Distribusi yang terkait dengan fungsi yang sama	100

Selain kuota ini, ada beberapa batasan lain saat menggunakan CloudFront Fungsi. Untuk informasi selengkapnya, lihat [Pembatasan CloudFront Fungsi](#).

Kuota pada Fungsi Koneksi

Entitas	Kuota default
Fungsi Koneksi per Akun AWS	100
Ukuran Fungsi Koneksi Maksimum Kuota ini tidak dapat disesuaikan. Untuk menyimpan data tambahan untuk Fungsi Koneksi Anda, buat penyimpanan nilai kunci dan tambahkan pasangan nilai kunci Anda. Untuk informasi selengkapnya, lihat Amazon CloudFront KeyValueCollection .	10 KB

Entitas	Kuota default
Memori Fungsi Koneksi Maksimum	2 MB
Distribusi yang terkait dengan Fungsi Koneksi yang sama	100

Selain kuota ini, ada beberapa batasan lain saat menggunakan Fungsi Koneksi. Untuk informasi selengkapnya, lihat [Kaitkan Fungsi CloudFront Koneksi](#).

Kuota pada toko nilai utama

Entitas	Kuota default
Ukuran maksimum kunci dalam pasangan kunci-nilai	512 Byte
Ukuran maksimum nilai dalam pasangan kunci-nilai	1 KB
Pasangan nilai kunci maksimum yang dapat Anda perbarui dalam satu permintaan API	50 kunci atau muatan 3 MB, mana yang tercapai terlebih dahulu
Ukuran maksimum penyimpanan nilai kunci individu	5 MB
Jumlah maksimum fungsi yang dapat dikaitkan dengan satu penyimpanan nilai kunci	10
Jumlah maksimum penyimpanan nilai kunci per fungsi	1
Jumlah maksimum penyimpanan nilai kunci per akun	50
	Minta kuota yang lebih tinggi

Kuotas di Lambda@Edge

Kuota umum

Entitas	Kuota default
Distribusi per Akun AWS yang dapat memiliki fungsi Lambda @Edge	500 Minta kuota yang lebih tinggi
Fungsi Lambda@Edge per distribusi	100 Minta kuota yang lebih tinggi
Eksekusi yang bersamaan	1.000 (di masing-masing Wilayah AWS) Minta kuota yang lebih tinggi

 Catatan

- AWS Lambda mengelola kuota konkurensi untuk Lambda @Edge. Semua fungsi Lambda dalam Wilayah AWS berbagi kuota ini.
- Kami menyarankan Anda meninjau kuota eksekusi bersamaan di semua Wilayah AWS tempat yang Anda harapkan berasal dari permintaan pemirsa. Selain itu, setiap instance fungsi Lambda @Edge Anda dapat melayani hingga 10 permintaan per detik. Batas pemanggilan total adalah 10 kali batas konkurensi Anda.

Untuk informasi selengkapnya, lihat topik berikut di Panduan AWS Lambda Pengembang:

- [Memahami penskalaan fungsi Lambda](#)
-

Entitas	Kuota default
Permintaan API Lambda	
Distribusi yang terkait dengan fungsi yang sama	500
Ukuran maksimal terkompresi fungsi Lambda dan pustaka yang disertakan	50 MB
Permintaan Lambda @Edge per detik (masing-masing didukung Wilayah AWS). Untuk informasi selengkapnya, lihat Kuota konkurensi di Panduan AWS Lambda Pengembang.	10.000
Ukuran memori fungsi	Sama seperti Kuota Lambda
Waktu fungsi habis. Fungsi ini dapat membuat panggilan jaringan ke sumber daya seperti instans bucket Amazon S3, tabel DynamoDB, atau Amazon EC2 di Wilayah AWS.	30 detik

Kuota yang berbeda berdasarkan jenis peristiwa

Entitas	Peristiwa permintaan penampil dan respons penampil	Peristiwa permintaan asal dan respons asal
Ukuran respons yang dihasilkan oleh fungsi Lambda, termasuk header dan tubuh	40 KB	1 MB

Catatan

- Untuk daftar kuota Lambda @Edge tambahan yang dapat ditingkatkan dari Service Quotas, lihat titik [akhir CloudFront Amazon](#) dan kuota di. Referensi Umum AWS

- Selain kuota ini, ada beberapa pembatasan lain saat menggunakan fungsi Lambda@Edge. Untuk informasi selengkapnya, lihat [Pembatasan Lambda@Edge](#).

Kuota pada sertifikat SSL

Entitas	Kuota default
Sertifikat SSL per Akun AWS saat melayani permintaan HTTPS menggunakan alamat IP khusus (tidak ada kuota saat melayani permintaan HTTPS menggunakan SNI)	2
Untuk informasi selengkapnya, lihat Gunakan HTTPS dengan CloudFront .	Minta kuota yang lebih tinggi
Sertifikat SSL yang dapat dikaitkan dengan distribusi CloudFront	1

Jika sertifikat SSL Anda khusus untuk komunikasi HTTPS antara pemirsa dan CloudFront, dan jika Anda menggunakan AWS Certificate Manager (ACM) atau penyimpanan sertifikat IAM untuk menyediakan atau mengimpor sertifikat Anda, kuota tambahan berlaku. Untuk informasi selengkapnya, lihat [Kuota tentang penggunaan SSL/TLS sertifikat dengan CloudFront \(HTTPS antara pemirsa dan CloudFront hanya\)](#).

Ada juga kuota pada jumlah sertifikat SSL yang dapat Anda impor ke AWS Certificate Manager (ACM) atau upload ke AWS Identity and Access Management (IAM). Untuk informasi selengkapnya, lihat [Tingkatkan kuota untuk sertifikat SSL/TLS](#).

Kuotas pada invalidasi

Entitas	Kuota default
Jumlah jalur atau tag	150 jalur atau tag per detik
Untuk informasi selengkapnya, lihat Membatalkan file untuk menghapus konten dan Membatalkan konten dengan tag cache .	

Entitas	Kuota default
Jumlah pembatalan wildcard	1 pembatalan wildcard per detik
Pembatalan tag: jumlah maksimum file yang dapat diproses oleh satu pembatalan berbasis tag	Tidak ada kuota
Pembatalan wildcard: jumlah maksimum file yang dapat diproses oleh satu pembatalan wildcard	Tidak ada kuota

Kuotas pada kelompok utama

Entitas	Kuota standar
Kunci publik dalam satu kelompok kunci	5 Minta kuota yang lebih tinggi
Grup kunci yang terkait dengan perilaku cache tunggal	4 Minta kuota yang lebih tinggi
Grup kunci per Akun AWS	10 Minta kuota yang lebih tinggi
Distribusi yang terkait dengan satu grup kunci	100 Minta kuota yang lebih tinggi

Kuota pada koneksi WebSocket

Entitas	Kuota standar
Waktu habis respons asal (waktu habis tunggal)	10 menit
	Jika CloudFront belum mendeteksi byte yang dikirim dari asal ke klien dalam 10 menit terakhir, koneksi dianggap idle dan ditutup.

Kuotas pada enkripsi tingkat lapangan

Entitas	Kuota standar
Panjang maksimum kolom untuk mengenkripsi	16 KB
Untuk informasi selengkapnya, lihat Gunakan enkripsi tingkat lapangan untuk membantu melindungi data sensitif .	
Jumlah maksimum bidang dalam badan permintaan saat enkripsi tingkat bidang dikonfigurasi	10
Panjang maksimum badan permintaan saat enkripsi tingkat lapangan dikonfigurasi	1 MB
Jumlah maksimum konfigurasi enkripsi tingkat lapangan yang dapat dikaitkan dengan satu Akun AWS	10
Jumlah maksimum profil enkripsi tingkat lapangan yang dapat dikaitkan dengan satu Akun AWS	10
Jumlah maksimum kunci publik yang dapat ditambahkan ke satu Akun AWS	10

Entitas	Kuota standar
Jumlah maksimum kolom untuk mengenkripsi yang dapat ditentukan dalam satu profil	10
Jumlah maksimum CloudFront distribusi yang dapat dikaitkan dengan konfigurasi enkripsi tingkat lapangan	20
Jumlah maksimum pemetaan profil argumen kueri yang dapat dimasukkan dalam konfigurasi enkripsi tingkat lapangan	5

Kuota pada cookie (pengaturan cache warisan)

Kuota ini berlaku untuk CloudFront pengaturan cache lama. Sebaiknya gunakan [kebijakan cache](#) atau [kebijakan permintaan asal](#) alih-alih pengaturan lama.

Entitas	Kuota standar
Cookie per perilaku cache	10
Untuk informasi selengkapnya, lihat Konten cache berdasarkan cookie .	Minta kuota yang lebih tinggi
Jumlah total byte dalam nama cookie (tidak berlaku jika Anda mengonfigurasi CloudFront untuk meneruskan semua cookie ke asal)	512 dikurangi jumlah cookie

Kuota pada string kueri (pengaturan cache warisan)

Kuota ini berlaku untuk CloudFront pengaturan cache lama. Sebaiknya gunakan [kebijakan cache](#) atau [kebijakan permintaan asal](#) alih-alih pengaturan lama.

Entitas	Kuota standar
Jumlah karakter maksimum dalam string kueri	128 karakter
Total jumlah maksimal karakter untuk semua string kueri dalam parameter yang sama	512 karakter

Entitas	Kuota standar
String kueri per perilaku cache	10
Untuk informasi selengkapnya, lihat Konten cache berdasarkan parameter string kueri .	Minta kuota yang lebih tinggi

Kuota pada header

Entitas	Kuota standar
Header per perilaku cache (pengaturan cache warisan)	10
Untuk informasi selengkapnya, lihat the section called “Konten cache berdasarkan header permintaan” .	Minta kuota yang lebih tinggi
Teruskan header per perilaku cache	25
	Minta kuota yang lebih tinggi
Header khusus: jumlah maksimum header khusus yang dapat Anda konfigurasi CloudFront untuk ditambahkan ke permintaan asal	30
Untuk informasi selengkapnya, lihat the section called “Tambahkan header khusus ke permintaan asal” .	Minta kuota yang lebih tinggi
Header khusus: jumlah maksimum header kustom yang dapat Anda tambahkan ke kebijakan header respons	10
	Minta kuota yang lebih tinggi
Header kustom: panjang maksimum nama header	256 karakter
Header kustom: panjang maksimum nilai header	1.783 karakter
Header kustom: panjang maksimum untuk semua nilai header dan nama digabungkan	10.240 karakter

Entitas	Kuota standar
Panjang maksimum nilai Content-Security-Policy header	1.783 karakter Minta kuota yang lebih tinggi
Panjang maksimum nilai CORS (Access-Control-Allow-Origin) header	1.783 karakter

Kuota pada distribusi multi-penyewa

Entitas	Kuota default
Jumlah maksimum penyewa distribusi per Akun AWS	10.000 Minta kuota yang lebih tinggi
Jumlah maksimum distribusi multi-tenant per Akun AWS	20 Minta kuota yang lebih tinggi
Jumlah maksimum grup koneksi per Akun AWS	100 Minta kuota yang lebih tinggi
Jumlah maksimum alias per penyewa distribusi	100 Minta kuota yang lebih tinggi
Jumlah maksimum parameter per penyewa distribusi	5 Minta kuota yang lebih tinggi

Entitas	Kuota default
Jumlah maksimum parameter per distribusi multi-tenant	5 Minta kuota yang lebih tinggi
Jumlah maksimum parameter dalam suatu bidang dalam distribusi multi-tenant	2 Minta kuota yang lebih tinggi
Jumlah maksimum grup koneksi per daftar IP statis Anycast	5 Minta kuota yang lebih tinggi

Untuk informasi selengkapnya tentang distribusi multi-penyewa, lihat. [Memahami cara kerja distribusi multi-tenant](#)

Informasi Terkait

Untuk informasi selengkapnya, lihat [CloudFront titik akhir dan kuota Amazon](#) di. Referensi Umum AWS

Contoh kode untuk CloudFront menggunakan AWS SDKs

Contoh kode berikut menunjukkan cara menggunakan CloudFront kit pengembangan AWS perangkat lunak (SDK).

Tindakan merupakan kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Sementara tindakan menunjukkan cara memanggil fungsi layanan individual, Anda dapat melihat tindakan dalam konteks dalam skenario terkait.

Skenario adalah contoh kode yang menunjukkan kepada Anda bagaimana menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan atau dikombinasikan dengan yang lain Layanan AWS.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Contoh kode

- [Contoh dasar untuk CloudFront menggunakan AWS SDKs](#)
 - [Tindakan untuk CloudFront menggunakan AWS SDKs](#)
 - [Gunakan CreateDistribution dengan AWS SDK atau CLI](#)
 - [Gunakan CreateFunction dengan AWS SDK](#)
 - [Gunakan CreateInvalidation dengan CLI](#)
 - [Gunakan CreateKeyGroup dengan AWS SDK](#)
 - [Gunakan CreatePublicKey dengan AWS SDK atau CLI](#)
 - [Gunakan DeleteDistribution dengan AWS SDK atau CLI](#)
 - [Gunakan GetCloudFrontOriginAccessIdentity dengan CLI](#)
 - [Gunakan GetCloudFrontOriginAccessIdentityConfig dengan CLI](#)
 - [Gunakan GetDistribution dengan CLI](#)
 - [Gunakan GetDistributionConfig dengan AWS SDK atau CLI](#)
 - [Gunakan ListCloudFrontOriginAccessIdentities dengan CLI](#)
 - [Gunakan ListDistributions dengan AWS SDK atau CLI](#)
 - [Gunakan UpdateDistribution dengan AWS SDK atau CLI](#)
 - [Skenario untuk CloudFront menggunakan AWS SDKs](#)

- [Buat SDK sumber daya manajer SaaS AWS](#)
- [Hapus sumber CloudFront penandatanganan menggunakan AWS SDK](#)
- [Memulai dengan CloudFront distribusi dasar](#)
- [Memulai dengan WAF](#)
- [Buat cookie yang ditandatangani URLs dan menggunakan AWS SDK](#)
- [CloudFront Contoh fungsi untuk CloudFront](#)
 - [Tambahkan header keamanan HTTP ke acara respons penampil CloudFront Fungsi](#)
 - [Tambahkan header CORS ke acara respons penampil CloudFront Fungsi](#)
 - [Tambahkan header kontrol cache ke acara respons penampil CloudFront Fungsi](#)
 - [Tambahkan header IP klien sejati ke acara permintaan penampil CloudFront Fungsi](#)
 - [Tambahkan header asal ke acara permintaan penampil CloudFront Fungsi](#)
 - [Tambahkan index.html untuk meminta URLs tanpa nama file dalam acara permintaan penampil CloudFront Fungsi](#)
 - [Menormalkan parameter string kueri dalam permintaan penampil CloudFront Fungsi](#)
 - [Mengalihkan ke URL baru dalam acara permintaan penampil CloudFront Fungsi](#)
 - [Menulis ulang URI permintaan berdasarkan KeyValueCollection konfigurasi untuk acara permintaan penampil CloudFront Fungsi](#)
 - [Rutekan permintaan ke asal yang lebih dekat dengan penampil dalam acara permintaan penampil CloudFront Fungsi](#)
 - [Gunakan pasangan nilai kunci dalam permintaan penampil CloudFront Fungsi](#)
 - [Validasi token sederhana dalam permintaan penampil CloudFront Fungsi](#)

Contoh dasar untuk CloudFront menggunakan AWS SDKs

Contoh kode berikut menunjukkan cara menggunakan dasar-dasar Amazon CloudFront dengan AWS SDKs.

Contoh

- [Tindakan untuk CloudFront menggunakan AWS SDKs](#)
 - [Gunakan CreateDistribution dengan AWS SDK atau CLI](#)
 - [Gunakan CreateFunction dengan AWS SDK](#)
 - [Gunakan CreateInvalidation dengan CLI](#)

- [Gunakan CreateKeyGroup dengan AWS SDK](#)
- [Gunakan CreatePublicKey dengan AWS SDK atau CLI](#)
- [Gunakan DeleteDistribution dengan AWS SDK atau CLI](#)
- [Gunakan GetCloudFrontOriginAccessIdentity dengan CLI](#)
- [Gunakan GetCloudFrontOriginAccessIdentityConfig dengan CLI](#)
- [Gunakan GetDistribution dengan CLI](#)
- [Gunakan GetDistributionConfig dengan AWS SDK atau CLI](#)
- [Gunakan ListCloudFrontOriginAccessIdentities dengan CLI](#)
- [Gunakan ListDistributions dengan AWS SDK atau CLI](#)
- [Gunakan UpdateDistribution dengan AWS SDK atau CLI](#)

Tindakan untuk CloudFront menggunakan AWS SDKs

Contoh kode berikut menunjukkan bagaimana melakukan CloudFront tindakan individu dengan AWS SDKs. Setiap contoh menyertakan tautan ke GitHub, di mana Anda dapat menemukan instruksi untuk mengatur dan menjalankan kode.

Kutipan ini memanggil CloudFront API dan merupakan kutipan kode dari program yang lebih besar yang harus dijalankan dalam konteks. Anda dapat melihat tindakan dalam konteks di [Skenario untuk CloudFront menggunakan AWS SDKs](#).

Contoh berikut hanya mencakup tindakan yang paling umum digunakan. Untuk daftar lengkapnya, lihat [Referensi Amazon CloudFront API](#).

Contoh

- [Gunakan CreateDistribution dengan AWS SDK atau CLI](#)
- [Gunakan CreateFunction dengan AWS SDK](#)
- [Gunakan CreateInvalidation dengan CLI](#)
- [Gunakan CreateKeyGroup dengan AWS SDK](#)
- [Gunakan CreatePublicKey dengan AWS SDK atau CLI](#)
- [Gunakan DeleteDistribution dengan AWS SDK atau CLI](#)
- [Gunakan GetCloudFrontOriginAccessIdentity dengan CLI](#)
- [Gunakan GetCloudFrontOriginAccessIdentityConfig dengan CLI](#)
- [Gunakan GetDistribution dengan CLI](#)

- [Gunakan GetDistributionConfig dengan AWS SDK atau CLI](#)
- [Gunakan ListCloudFrontOriginAccessIdentities dengan CLI](#)
- [Gunakan ListDistributions dengan AWS SDK atau CLI](#)
- [Gunakan UpdateDistribution dengan AWS SDK atau CLI](#)

Gunakan **CreateDistribution** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateDistribution`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Buat penyewa distribusi dan distribusi multi-tenant](#)
- [Memulai dengan CloudFront distribusi dasar](#)

CLI

AWS CLI

Contoh 1: Untuk membuat CloudFront distribusi

`create-distribution` Contoh berikut membuat distribusi untuk bucket S3 bernama `amzn-s3-demo-bucket`, dan juga menentukan `index.html` sebagai objek root default, menggunakan argumen baris perintah.

```
aws cloudfront create-distribution \  
  --origin-domain-name amzn-s3-demo-bucket.s3.amazonaws.com \  
  --default-root-object index.html
```

Output:

```
{  
  "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/  
EMLARXS9EXAMPLE",  
  "ETag": "E9LHASXEXAMPLE",  
  "Distribution": {  
    "Id": "EMLARXS9EXAMPLE",  
    "ARN": "arn:aws:cloudfront::123456789012:distribution/EMLARXS9EXAMPLE",  
    "Status": "InProgress",  
    "LastModifiedTime": "2019-11-22T00:55:15.705Z",
```

```

    "InProgressInvalidationBatches": 0,
    "DomainName": "d1111111abcdef8.cloudfront.net",
    "ActiveTrustedSigners": {
        "Enabled": false,
        "Quantity": 0
    },
    "DistributionConfig": {
        "CallerReference": "cli-example",
        "Aliases": {
            "Quantity": 0
        },
        "DefaultRootObject": "index.html",
        "Origins": {
            "Quantity": 1,
            "Items": [
                {
                    "Id": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
                    "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                    "OriginPath": "",
                    "CustomHeaders": {
                        "Quantity": 0
                    },
                    "S3OriginConfig": {
                        "OriginAccessIdentity": ""
                    }
                }
            ]
        },
        "OriginGroups": {
            "Quantity": 0
        },
        "DefaultCacheBehavior": {
            "TargetOriginId": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-
example",
            "ForwardedValues": {
                "QueryString": false,
                "Cookies": {
                    "Forward": "none"
                },
                "Headers": {
                    "Quantity": 0
                },
                "QueryStringCacheKeys": {
                    "Quantity": 0
                }
            }
        }
    }
}

```

```
    }
  },
  "TrustedSigners": {
    "Enabled": false,
    "Quantity": 0
  },
  "ViewerProtocolPolicy": "allow-all",
  "MinTTL": 0,
  "AllowedMethods": {
    "Quantity": 2,
    "Items": [
      "HEAD",
      "GET"
    ],
    "CachedMethods": {
      "Quantity": 2,
      "Items": [
        "HEAD",
        "GET"
      ]
    }
  },
  "SmoothStreaming": false,
  "DefaultTTL": 86400,
  "MaxTTL": 31536000,
  "Compress": false,
  "LambdaFunctionAssociations": {
    "Quantity": 0
  },
  "FieldLevelEncryptionId": ""
},
"CacheBehaviors": {
  "Quantity": 0
},
"CustomErrorResponses": {
  "Quantity": 0
},
"Comment": "",
"Logging": {
  "Enabled": false,
  "IncludeCookies": false,
  "Bucket": "",
  "Prefix": ""
},
}
```

```

        "PriceClass": "PriceClass_All",
        "Enabled": true,
        "ViewerCertificate": {
            "CloudFrontDefaultCertificate": true,
            "MinimumProtocolVersion": "TLSv1",
            "CertificateSource": "cloudfront"
        },
        "Restrictions": {
            "GeoRestriction": {
                "RestrictionType": "none",
                "Quantity": 0
            }
        },
        "WebACLId": "",
        "HttpVersion": "http2",
        "IsIPV6Enabled": true
    }
}

```

Contoh 2: Untuk membuat CloudFront distribusi menggunakan file JSON

`create-distribution` Contoh berikut membuat distribusi untuk bucket S3 bernama `amzn-s3-demo-bucket`, dan juga menentukan `index.html` sebagai objek root default, menggunakan file JSON.

```

aws cloudfront create-distribution \
  --distribution-config file://dist-config.json

```

Isi dari `dist-config.json`:

```

{
    "CallerReference": "cli-example",
    "Aliases": {
        "Quantity": 0
    },
    "DefaultRootObject": "index.html",
    "Origins": {
        "Quantity": 1,
        "Items": [
            {
                "Id": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
                "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",

```



```
        "OriginPath": "",
        "CustomHeaders": {
            "Quantity": 0
        },
        "S3OriginConfig": {
            "OriginAccessIdentity": ""
        }
    }
]
},
"OriginGroups": {
    "Quantity": 0
},
"DefaultCacheBehavior": {
    "TargetOriginId": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
    "ForwardedValues": {
        "QueryString": false,
        "Cookies": {
            "Forward": "none"
        },
        "Headers": {
            "Quantity": 0
        },
        "QueryStringCacheKeys": {
            "Quantity": 0
        }
    },
    "TrustedSigners": {
        "Enabled": false,
        "Quantity": 0
    },
    "ViewerProtocolPolicy": "allow-all",
    "MinTTL": 0,
    "AllowedMethods": {
        "Quantity": 2,
        "Items": [
            "HEAD",
            "GET"
        ],
        "CachedMethods": {
            "Quantity": 2,
            "Items": [
                "HEAD",
                "GET"
            ]
        }
    }
}
```

```

        ]
    },
    "SmoothStreaming": false,
    "DefaultTTL": 86400,
    "MaxTTL": 31536000,
    "Compress": false,
    "LambdaFunctionAssociations": {
        "Quantity": 0
    },
    "FieldLevelEncryptionId": ""
},
"CacheBehaviors": {
    "Quantity": 0
},
"CustomErrorResponses": {
    "Quantity": 0
},
"Comment": "",
"Logging": {
    "Enabled": false,
    "IncludeCookies": false,
    "Bucket": "",
    "Prefix": ""
},
"PriceClass": "PriceClass_All",
"Enabled": true,
"ViewerCertificate": {
    "CloudFrontDefaultCertificate": true,
    "MinimumProtocolVersion": "TLSv1",
    "CertificateSource": "cloudfront"
},
"Restrictions": {
    "GeoRestriction": {
        "RestrictionType": "none",
        "Quantity": 0
    }
},
"WebACLId": "",
"HttpVersion": "http2",
"IsIPV6Enabled": true
}

```

Lihat Contoh 1 untuk keluaran sampel.

Contoh 3: Untuk membuat distribusi CloudFront multi-tenant dengan sertifikat

`create-distribution` Contoh berikut membuat CloudFront distribusi dengan dukungan multi-tenant dan menentukan sertifikat TLS.

```
aws cloudfront create-distribution \  
  --distribution-config file://dist-config.json
```

Isi dari `dist-config.json`:

```
{  
  "CallerReference": "cli-example-with-cert",  
  "Comment": "CLI example distribution",  
  "DefaultRootObject": "index.html",  
  "Origins": {  
    "Quantity": 1,  
    "Items": [  
      {  
        "Id": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",  
        "DomainName": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",  
        "OriginPath": "/{{tenantName}}",  
        "CustomHeaders": {  
          "Quantity": 0  
        },  
        "S3OriginConfig": {  
          "OriginAccessIdentity": ""  
        }  
      }  
    ]  
  },  
  "DefaultCacheBehavior": {  
    "TargetOriginId": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",  
    "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e5ABC",  
    "ViewerProtocolPolicy": "allow-all",  
    "AllowedMethods": {  
      "Quantity": 2,  
      "Items": ["HEAD", "GET"],  
      "CachedMethods": {  
        "Quantity": 2,  
        "Items": ["HEAD", "GET"]  
      }  
    }  
  }  
}
```

```

    }
  },
  "Enabled": true,
  "ViewerCertificate": {
    "ACMCertificateArn": "arn:aws:acm:us-
east-1:123456789012:certificate/191306a1-db01-49ca-90ef-fc414ee5dabc",
    "SSLSupportMethod": "sni-only"
  },
  "HttpVersion": "http2",
  "ConnectionMode": "tenant-only",
  "TenantConfig": {
    "ParameterDefinitions": [
      {
        "Name": "tenantName",
        "Definition": {
          "StringSchema": {
            "Comment": "tenantName parameter",
            "DefaultValue": "root",
            "Required": false
          }
        }
      }
    ]
  }
}

```

Output:

```

{
  "Location": "https://cloudfront.amazonaws.com/2020-05-31/distribution/
E1HVIAU7UABC",
  "ETag": "E20LT7R1BABC",
  "Distribution": {
    "Id": "E1HVIAU7U12ABC",
    "ARN": "arn:aws:cloudfront::123456789012:distribution/E1HVIAU7U12ABC",
    "Status": "InProgress",
    "LastModifiedTime": "2025-07-10T20:33:31.117000+00:00",
    "InProgressInvalidationBatches": 0,
    "DomainName": "example.com",
    "ActiveTrustedSigners": {
      "Enabled": false,
      "Quantity": 0
    }
  },
}

```

```

    "ActiveTrustedKeyGroups": {
      "Enabled": false,
      "Quantity": 0
    },
    "DistributionConfig": {
      "CallerReference": "cli-example-with-cert",
      "DefaultRootObject": "index.html",
      "Origins": {
        "Quantity": 1,
        "Items": [
          {
            "Id": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
            "DomainName": "amzn-s3-demo-bucket.s3.us-
east-1.amazonaws.com",
            "OriginPath": "/{{tenantName}}",
            "CustomHeaders": {
              "Quantity": 0
            },
            "S3OriginConfig": {
              "OriginAccessIdentity": ""
            },
            "ConnectionAttempts": 3,
            "ConnectionTimeout": 10,
            "OriginShield": {
              "Enabled": false
            },
            "OriginAccessControlId": ""
          }
        ]
      },
      "OriginGroups": {
        "Quantity": 0
      },
      "DefaultCacheBehavior": {
        "TargetOriginId": "amzn-s3-demo-bucket.s3.us-
east-1.amazonaws.com",
        "TrustedKeyGroups": {
          "Enabled": false,
          "Quantity": 0
        },
        "ViewerProtocolPolicy": "allow-all",
        "AllowedMethods": {
          "Quantity": 2,
          "Items": ["HEAD", "GET"],

```

```

        "CachedMethods": {
            "Quantity": 2,
            "Items": ["HEAD", "GET"]
        },
        "Compress": false,
        "LambdaFunctionAssociations": {
            "Quantity": 0
        },
        "FunctionAssociations": {
            "Quantity": 0
        },
        "FieldLevelEncryptionId": "",
        "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e5ABC",
        "GrpcConfig": {
            "Enabled": false
        }
    },
    "CacheBehaviors": {
        "Quantity": 0
    },
    "CustomErrorResponses": {
        "Quantity": 0
    },
    "Comment": "CLI example distribution",
    "Logging": {
        "Enabled": false,
        "IncludeCookies": false,
        "Bucket": "",
        "Prefix": ""
    },
    "Enabled": true,
    "ViewerCertificate": {
        "CloudFrontDefaultCertificate": false,
        "ACMCertificateArn": "arn:aws:acm:us-east-1:123456789012:certificate/1954f095-11b6-4daf-9952-0c308a00abc",
        "SSLSupportMethod": "sni-only",
        "MinimumProtocolVersion": "TLSv1.2_2021",
        "Certificate": "arn:aws:acm:us-east-1:123456789012:certificate/1954f095-11b6-4daf-9952-0c308a00abc",
        "CertificateSource": "acm"
    },
    "Restrictions": {
        "GeoRestriction": {

```

```

        "RestrictionType": "none",
        "Quantity": 0
      },
      "WebACLId": "",
      "HttpVersion": "http2",
      "TenantConfig": {
        "ParameterDefinitions": [
          {
            "Name": "tenantName",
            "Definition": {
              "StringSchema": {
                "Comment": "tenantName parameter",
                "DefaultValue": "root",
                "Required": false
              }
            }
          }
        ]
      },
      "ConnectionMode": "tenant-only"
    }
  }
}

```

Untuk informasi selengkapnya, lihat [Bekerja dengan distribusi](#) di Panduan CloudFront Pengembang Amazon.

Contoh 4: Untuk membuat distribusi CloudFront multi-tenant tanpa sertifikat

create-distributionContoh berikut membuat CloudFront distribusi dengan dukungan multi-tenant tetapi tanpa sertifikat TLS.

```

aws cloudfront create-distribution \
  --distribution-config file://dist-config.json

```

Isi dari dist-config.json:

```

{
  "CallerReference": "cli-example",
  "Comment": "CLI example distribution",
  "DefaultRootObject": "index.html",
  "Origins": {

```

```
"Quantity": 1,
"Items": [
  {
    "Id": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
    "DomainName": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
    "OriginPath": "/{{tenantName}}",
    "CustomHeaders": {
      "Quantity": 0
    },
    "S3OriginConfig": {
      "OriginAccessIdentity": ""
    }
  }
],
"DefaultCacheBehavior": {
  "TargetOriginId": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
  "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e5ABC",
  "ViewerProtocolPolicy": "allow-all",
  "AllowedMethods": {
    "Quantity": 2,
    "Items": [
      "HEAD",
      "GET"
    ],
    "CachedMethods": {
      "Quantity": 2,
      "Items": [
        "HEAD",
        "GET"
      ]
    }
  }
},
"Enabled": true,
"HttpVersion": "http2",
"ConnectionMode": "tenant-only",
"TenantConfig": {
  "ParameterDefinitions": [
    {
      "Name": "tenantName",
      "Definition": {
        "StringSchema": {
          "Comment": "tenantName parameter",
```



```

        "DefaultValue": "root",
        "Required": false
      }
    }
  ]
}

```

Output:

```

{
  "Location": "https://cloudfront.amazonaws.com/2020-05-31/distribution/
E2GJ5J9QN12ABC",
  "ETag": "E37YLVVQIABC",
  "Distribution": {
    "Id": "E2GJ5J9QNABC",
    "ARN": "arn:aws:cloudfront::123456789012:distribution/E2GJ5J9QN12ABC",
    "Status": "InProgress",
    "LastModifiedTime": "2025-07-10T20:35:20.565000+00:00",
    "InProgressInvalidationBatches": 0,
    "DomainName": "example.com",
    "ActiveTrustedSigners": {
      "Enabled": false,
      "Quantity": 0
    },
    "ActiveTrustedKeyGroups": {
      "Enabled": false,
      "Quantity": 0
    },
    "DistributionConfig": {
      "CallerReference": "cli-example-no-cert",
      "DefaultRootObject": "index.html",
      "Origins": {
        "Quantity": 1,
        "Items": [
          {
            "Id": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
            "DomainName": "amzn-s3-demo-bucket.s3.us-
east-1.amazonaws.com",
            "OriginPath": "{{tenantName}}",
            "CustomHeaders": {
              "Quantity": 0
            }
          }
        ]
      }
    }
  }
}

```

```

        },
        "S3OriginConfig": {
            "OriginAccessIdentity": ""
        },
        "ConnectionAttempts": 3,
        "ConnectionTimeout": 10,
        "OriginShield": {
            "Enabled": false
        },
        "OriginAccessControlId": ""
    }
]
},
"OriginGroups": {
    "Quantity": 0
},
"DefaultCacheBehavior": {
    "TargetOriginId": "amzn-s3-demo-bucket.s3.us-east-1.amazonaws.com",
    "TrustedKeyGroups": {
        "Enabled": false,
        "Quantity": 0
    },
    "ViewerProtocolPolicy": "allow-all",
    "AllowedMethods": {
        "Quantity": 2,
        "Items": [
            "HEAD",
            "GET"
        ],
        "CachedMethods": {
            "Quantity": 2,
            "Items": [
                "HEAD",
                "GET"
            ]
        }
    },
    "Compress": false,
    "LambdaFunctionAssociations": {
        "Quantity": 0
    },
    "FunctionAssociations": {
        "Quantity": 0
    }
}

```

```
    },
    "FieldLevelEncryptionId": "",
    "CachePolicyId": "658327ea-f89d-4fab-a63d-7e88639e5ABC",
    "GrpcConfig": {
      "Enabled": false
    }
  },
  "CacheBehaviors": {
    "Quantity": 0
  },
  "CustomErrorResponses": {
    "Quantity": 0
  },
  "Comment": "CLI example distribution",
  "Logging": {
    "Enabled": false,
    "IncludeCookies": false,
    "Bucket": "",
    "Prefix": ""
  },
  "Enabled": true,
  "ViewerCertificate": {
    "CloudFrontDefaultCertificate": true,
    "SSLSupportMethod": "sni-only",
    "MinimumProtocolVersion": "TLSv1",
    "CertificateSource": "cloudfront"
  },
  "Restrictions": {
    "GeoRestriction": {
      "RestrictionType": "none",
      "Quantity": 0
    }
  },
  "WebACLId": "",
  "HttpVersion": "http2",
  "TenantConfig": {
    "ParameterDefinitions": [
      {
        "Name": "tenantName",
        "Definition": {
          "StringSchema": {
            "Comment": "tenantName parameter",
            "DefaultValue": "root",
            "Required": false
          }
        }
      }
    ]
  }
}
```

```

    }
  }
},
"ConnectionMode": "tenant-only"
}
}
}

```

Untuk informasi selengkapnya, lihat [Mengonfigurasi distribusi](#) di Panduan CloudFront Pengembang Amazon.

- Untuk detail API, lihat [CreateDistribution](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Contoh berikut menggunakan bucket Amazon Simple Storage Service (Amazon S3) sebagai sumber konten.

Setelah membuat distribusi, kode membuat [CloudFrontWaiter](#) untuk menunggu sampai distribusi diterapkan sebelum mengembalikan distribusi.

```

import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.internal.waiters.ResponseOrException;
import software.amazon.awssdk.services.cloudfront.CloudFrontClient;
import
    software.amazon.awssdk.services.cloudfront.model.CreateDistributionResponse;
import software.amazon.awssdk.services.cloudfront.model.Distribution;
import software.amazon.awssdk.services.cloudfront.model.GetDistributionResponse;
import software.amazon.awssdk.services.cloudfront.model.ItemSelection;
import software.amazon.awssdk.services.cloudfront.model.Method;

```

```

import software.amazon.awssdk.services.cloudfront.model.ViewerProtocolPolicy;
import software.amazon.awssdk.services.cloudfront.waiters.CloudFrontWaiter;
import software.amazon.awssdk.services.s3.S3Client;

import java.time.Instant;

public class CreateDistribution {

    private static final Logger logger =
        LoggerFactory.getLogger(CreateDistribution.class);

    public static Distribution createDistribution(CloudFrontClient
        cloudFrontClient, S3Client s3Client,
            final String bucketName, final String keyGroupId, final
            String originAccessControlId) {

        final String region = s3Client.headBucket(b ->
            b.bucket(bucketName)).sdkHttpResponse().headers()
            .get("x-amz-bucket-region").get(0);
        final String originDomain = bucketName + ".s3." + region +
            ".amazonaws.com";
        String originId = originDomain; // Use the originDomain value for
        the originId.

        // The service API requires some deprecated methods, such as
        // DefaultCacheBehavior.Builder#minTTL and #forwardedValue.
        CreateDistributionResponse createDistResponse =
            cloudFrontClient.createDistribution(builder -> builder
                .distributionConfig(b1 -> b1
                    .origins(b2 -> b2
                        .quantity(1)
                        .items(b3 -> b3

                            .domainName(originDomain)

                            .id(originId)

                            .s3OriginConfig(builder4 -> builder4
                                .originAccessIdentity(
                                    ""))

                            .originAccessControlId(

```

```

        originAccessControlId)))
        .defaultCacheBehavior(b2 -> b2

    .viewerProtocolPolicy(ViewerProtocolPolicy.ALLOW_ALL)

    .targetOriginId(originId)

        .minTTL(200L)

    .forwardedValues(b5 -> b5

    .cookies(cp -> cp

        .forward(ItemSelection.NONE))

    .queryString(true))

    .trustedKeyGroups(b3 -> b3

    .quantity(1)

    .items(keyGroupId)

    .enabled(true))

    .allowedMethods(b4 -> b4

    .quantity(2)

    .items(Method.HEAD, Method.GET)

    .cachedMethods(b5 -> b5

        .quantity(2)

        .items(Method.HEAD,

            Method.GET)))

        .cacheBehaviors(b -> b
            .quantity(1)
            .items(b2 -> b2

    .pathPattern("/index.html")

```

```
.viewerProtocolPolicy(
    ViewerProtocolPolicy.ALLOW_ALL)

.targetOriginId(originId)

.trustedKeyGroups(b3 -> b3
    .quantity(1)
    .items(keyGroupId)
    .enabled(true))

.minTTL(200L)

.forwardedValues(b4 -> b4
    .cookies(cp -> cp
        .forward(ItemSelection.NONE))
    .queryString(true))

.allowedMethods(b5 -> b5.quantity(2)
    .items(Method.HEAD,
        Method.GET)
    .cachedMethods(b6 -> b6
        .quantity(2)
        .items(Method.HEAD,
            Method.GET))))
    .enabled(true)
    .comment("Distribution built with
java")

.callerReference(Instant.now().toString()));
```

```

        final Distribution distribution =
createDistResponse.distribution();
        logger.info("Distribution created. DomainName: [{}] Id: [{}]",
distribution.domainName(),
                        distribution.id());
        logger.info("Waiting for distribution to be deployed ...");
        try (CloudFrontWaiter cfWaiter =
CloudFrontWaiter.builder().client(cloudFrontClient).build()) {
            ResponseOrException<GetDistributionResponse>
responseOrException = cfWaiter
                                .waitUntilDistributionDeployed(builder ->
builder.id(distribution.id()))
                                .matched();
            responseOrException.response()
                                .orElseThrow(() -> new
RuntimeException("Distribution not created"));
            logger.info("Distribution deployed. DomainName: [{}] Id:
[{}]", distribution.domainName(),
                                distribution.id());
        }
        return distribution;
    }
}

```

- Untuk detail API, lihat [CreateDistribution](#) di Referensi AWS SDK for Java 2.x API.

PowerShell

Alat untuk PowerShell V4

Contoh 1: Membuat CloudFront distribusi dasar, dikonfigurasi dengan logging dan caching.

```

$origin = New-Object Amazon.CloudFront.Model.Origin
$origin.DomainName = "amzn-s3-demo-bucket.s3.amazonaws.com"
$origin.Id = "UniqueOrigin1"
$origin.S3OriginConfig = New-Object Amazon.CloudFront.Model.S3OriginConfig
$origin.S3OriginConfig.OriginAccessIdentity = ""
New-CFDistribution `
    -DistributionConfig_Enabled $true `
    -DistributionConfig_Comment "Test distribution" `
    -Origins_Item $origin `
    -Origins_Quantity 1 `

```



```

-Logging_Enabled $true `
-Logging_IncludeCookie $true `
-Logging_Bucket amzn-s3-demo-logging-bucket.s3.amazonaws.com `
-Logging_Prefix "help/" `
-DistributionConfig_CallerReference Client1 `
-DistributionConfig_DefaultRootObject index.html `
-DefaultCacheBehavior_TargetOriginId $origin.Id `
-ForwardedValues_QueryString $true `
-Cookies_Forward all `
-WhitelistedNames_Quantity 0 `
-TrustedSigners_Enabled $false `
-TrustedSigners_Quantity 0 `
-DefaultCacheBehavior_ViewerProtocolPolicy allow-all `
-DefaultCacheBehavior_MinTTL 1000 `
-DistributionConfig_PriceClass "PriceClass_All" `
-CacheBehaviors_Quantity 0 `
-Aliases_Quantity 0

```

- Untuk detail API, lihat [CreateDistribution](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V4).

Alat untuk PowerShell V5

Contoh 1: Membuat CloudFront distribusi dasar, dikonfigurasi dengan logging dan caching.

```

$origin = New-Object Amazon.CloudFront.Model.Origin
$origin.DomainName = "amzn-s3-demo-bucket.s3.amazonaws.com"
$origin.Id = "UniqueOrigin1"
$origin.S3OriginConfig = New-Object Amazon.CloudFront.Model.S3OriginConfig
$origin.S3OriginConfig.OriginAccessIdentity = ""
New-CFDistribution `
    -DistributionConfig_Enabled $true `
    -DistributionConfig_Comment "Test distribution" `
    -Origins_Item $origin `
    -Origins_Quantity 1 `
    -Logging_Enabled $true `
    -Logging_IncludeCookie $true `
    -Logging_Bucket amzn-s3-demo-logging-bucket.s3.amazonaws.com `
    -Logging_Prefix "help/" `
    -DistributionConfig_CallerReference Client1 `
    -DistributionConfig_DefaultRootObject index.html `
    -DefaultCacheBehavior_TargetOriginId $origin.Id `
    -ForwardedValues_QueryString $true `
    -Cookies_Forward all `

```

```
-WhitelistedNames_Quantity 0 `
-TrustedSigners_Enabled $false `
-TrustedSigners_Quantity 0 `
-DefaultCacheBehavior_ViewerProtocolPolicy allow-all `
-DefaultCacheBehavior_MinTTL 1000 `
-DistributionConfig_PriceClass "PriceClass_All" `
-CacheBehaviors_Quantity 0 `
-Aliases_Quantity 0
```

- Untuk detail API, lihat [CreateDistribution](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V5).

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CreateFunction** dengan AWS SDK

Contoh kode berikut menunjukkan cara menggunakan `CreateFunction`.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
import software.amazon.awssdk.core.SdkBytes;
import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudfront.CloudFrontClient;
import software.amazon.awssdk.services.cloudfront.model.CloudFrontException;
import software.amazon.awssdk.services.cloudfront.model.CreateFunctionRequest;
import software.amazon.awssdk.services.cloudfront.model.CreateFunctionResponse;
import software.amazon.awssdk.services.cloudfront.model.FunctionConfig;
import software.amazon.awssdk.services.cloudfront.model.FunctionRuntime;
import java.io.InputStream;

/**
```

```
* Before running this Java V2 code example, set up your development
* environment, including your credentials.
*
* For more information, see the following documentation topic:
*
* https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-started.html
*/
public class CreateFunction {

    public static void main(String[] args) {
        final String usage = ""

            Usage:
                <functionName> <filePath>

            Where:
                functionName - The name of the function to create.\s
                filePath - The path to a file that contains the application
logic for the function.\s
            """;

        if (args.length != 2) {
            System.out.println(usage);
            System.exit(1);
        }

        String functionName = args[0];
        String filePath = args[1];
        CloudFrontClient cloudFrontClient = CloudFrontClient.builder()
            .region(Region.AWS_GLOBAL)
            .build();

        String funArn = createNewFunction(cloudFrontClient, functionName,
filePath);
        System.out.println("The function ARN is " + funArn);
        cloudFrontClient.close();
    }

    public static String createNewFunction(CloudFrontClient cloudFrontClient,
String functionName, String filePath) {
        try {
            InputStream fileIs =
CreateFunction.class.getClassLoader().getResourceAsStream(filePath);
```

```

        SdkBytes functionCode = SdkBytes.fromInputStream(fileIs);

        FunctionConfig config = FunctionConfig.builder()
            .comment("Created by using the CloudFront Java API")
            .runtime(FunctionRuntime.CLOUDFRONT_JS_1_0)
            .build();

        CreateFunctionRequest functionRequest =
        CreateFunctionRequest.builder()
            .name(functionName)
            .functionCode(functionCode)
            .functionConfig(config)
            .build();

        CreateFunctionResponse response =
        cloudFrontClient.createFunction(functionRequest);
        return response.functionSummary().functionMetadata().functionARN();

    } catch (CloudFrontException e) {
        System.err.println(e.getMessage());
        System.exit(1);
    }
    return "";
}
}

```

- Untuk detail API, lihat [CreateFunction](#) di Referensi AWS SDK for Java 2.x API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CreateInvalidation** dengan CLI

Contoh kode berikut menunjukkan cara menggunakan `CreateInvalidation`.

CLI

AWS CLI

Untuk membuat pembatalan untuk distribusi CloudFront

`create-invalidation` Contoh berikut membuat pembatalan untuk file tertentu dalam distribusi yang ditentukan CloudFront :

```
aws cloudfront create-invalidation \  
  --distribution-id EDFDVBD6EXAMPLE \  
  --paths "/example-path/example-file.jpg" "/example-path/example-file2.png"
```

Output:

```
{  
  "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/  
EDFDVBD6EXAMPLE/invalidation/I1JLWSDAP8FU89",  
  "Invalidation": {  
    "Id": "I1JLWSDAP8FU89",  
    "Status": "InProgress",  
    "CreateTime": "2019-12-05T18:24:51.407Z",  
    "InvalidationBatch": {  
      "Paths": {  
        "Quantity": 2,  
        "Items": [  
          "/example-path/example-file2.png",  
          "/example-path/example-file.jpg"  
        ]  
      },  
      "CallerReference": "cli-1575570291-670203"  
    }  
  }  
}
```

Pada contoh sebelumnya, AWS CLI secara otomatis menghasilkan acak. `CallerReference` Untuk menentukan sendiri `CallerReference`, atau untuk menghindari meneruskan parameter pembatalan sebagai argumen baris perintah, Anda dapat menggunakan file JSON. Contoh berikut membuat pembatalan untuk dua file, dengan menyediakan parameter pembatalan dalam file JSON bernama: `inv-batch.json`

```
aws cloudfront create-invalidation \  
  --distribution-id EDFDVBD6EXAMPLE \  
  --invalidation-batch file://inv-batch.json
```

Isi dari `inv-batch.json`:

```
{
  "Paths": {
    "Quantity": 2,
    "Items": [
      "/example-path/example-file.jpg",
      "/example-path/example-file2.png"
    ]
  },
  "CallerReference": "cli-example"
}
```

Output:

```
{
  "Location": "https://cloudfront.amazonaws.com/2019-03-26/distribution/EDFDVBD6EXAMPLE/invalidation/I2J0I21PCUY0IK",
  "Invalidation": {
    "Id": "I2J0I21PCUY0IK",
    "Status": "InProgress",
    "CreateTime": "2019-12-05T18:40:49.413Z",
    "InvalidationBatch": {
      "Paths": {
        "Quantity": 2,
        "Items": [
          "/example-path/example-file.jpg",
          "/example-path/example-file2.png"
        ]
      },
      "CallerReference": "cli-example"
    }
  }
}
```

- Untuk detail API, lihat [CreateInvalidation](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell V4

Contoh 1: Contoh ini membuat pembatalan baru pada distribusi dengan ID EXAMPLENSTXAXE. CallerReference Ini adalah ID unik yang dipilih oleh pengguna; dalam hal ini, cap waktu yang mewakili 15 Mei 2019 pukul 9:00 pagi digunakan. Variabel \$Paths

menyimpan tiga jalur ke file gambar dan media yang tidak diinginkan pengguna sebagai bagian dari cache distribusi. Nilai parameter `-Paths_Quantity` adalah jumlah total jalur yang ditentukan dalam parameter `-Paths_Item`.

```
$Paths = "/images/*.gif", "/images/image1.jpg", "/videos/*.mp4"
New-CFInvalidation -DistributionId "EXAMPLENSTXAXE" -
InvalidationBatch_CallerReference 20190515090000 -Paths_Item $Paths -
Paths_Quantity 3
```

Output:

Invalidation	Location
-----	-----
Amazon.CloudFront.Model.Invalidation	https://cloudfront.amazonaws.com/2018-11-05/distribution/EXAMPLENSTXAXE/invalidation/EXAMPLE8N0K9H

- Untuk detail API, lihat [CreateInvalidation](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V4).

Alat untuk PowerShell V5

Contoh 1: Contoh ini membuat pembatalan baru pada distribusi dengan ID EXAMPLENSTXAXE. CallerReference Ini adalah ID unik yang dipilih oleh pengguna; dalam hal ini, cap waktu yang mewakili 15 Mei 2019 pukul 9:00 pagi digunakan. Variabel `$Paths` menyimpan tiga jalur ke file gambar dan media yang tidak diinginkan pengguna sebagai bagian dari cache distribusi. Nilai parameter `-Paths_Quantity` adalah jumlah total jalur yang ditentukan dalam parameter `-Paths_Item`.

```
$Paths = "/images/*.gif", "/images/image1.jpg", "/videos/*.mp4"
New-CFInvalidation -DistributionId "EXAMPLENSTXAXE" -
InvalidationBatch_CallerReference 20190515090000 -Paths_Item $Paths -
Paths_Quantity 3
```

Output:

Invalidation	Location
-----	-----

Amazon.CloudFront.Model.Invalidation <https://cloudfront.amazonaws.com/2018-11-05/distribution/EXAMPLENSTXAXE/invalidation/EXAMPLE8N0K9H>

- Untuk detail API, lihat [CreateInvalidation](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V5).

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CreateKeyGroup** dengan AWS SDK

Contoh kode berikut menunjukkan cara menggunakan `CreateKeyGroup`.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Grup kunci memerlukan setidaknya satu kunci publik yang digunakan untuk memverifikasi tanda tangan URLs atau cookie.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.services.cloudfront.CloudFrontClient;

import java.util.UUID;

public class CreateKeyGroup {
    private static final Logger logger =
        LoggerFactory.getLogger(CreateKeyGroup.class);

    public static String createKeyGroup(CloudFrontClient cloudFrontClient, String
        publicKeyId) {
        String keyGroupId = cloudFrontClient.createKeyGroup(b ->
            b.keyGroupConfig(c -> c
```



```

        .items(publicKeyId)
        .name("JavaKeyGroup" + UUID.randomUUID()))
        .keyGroup().id();
    logger.info("KeyGroup created with ID: [{}]", keyGroupId);
    return keyGroupId;
}
}

```

- Untuk detail API, lihat [CreateKeyGroup](#) di Referensi AWS SDK for Java 2.x API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **CreatePublicKey** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `CreatePublicKey`.

CLI

AWS CLI

Untuk membuat kunci CloudFront publik

Contoh berikut membuat kunci CloudFront publik dengan menyediakan parameter dalam file JSON bernama `pub-key-config.json`. Sebelum Anda dapat menggunakan perintah ini, Anda harus memiliki kunci publik yang dikodekan PEM. Untuk informasi selengkapnya, lihat [Membuat Pasangan Kunci RSA](#) di Panduan CloudFront Pengembang Amazon.

```

aws cloudfront create-public-key \
  --public-key-config file://pub-key-config.json

```

File tersebut `pub-key-config.json` adalah dokumen JSON di folder saat ini yang berisi berikut ini. Perhatikan bahwa kunci publik dikodekan dalam format PEM.

```

{
  "CallerReference": "cli-example",
  "Name": "ExampleKey",
  "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAA0CAQ8AMIIBCgKCAQEAxPMbCA2Ks0lnd7IR+3pw

```

```

\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAe/1U0WcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAz03ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPNhinFJ8Wdy8YBXSPpy7r95y1UQd9LfYTBzVZYG2tSesplc0kjM3\n2Uu
+oMWxQAw1NINnSLPinMVsutJy6Zq1V3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nnq
+kGZ2NQ0FyIyT2eiLK0X5Rgb/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nnrwIDAQAB\n-----
END PUBLIC KEY-----\n",
    "Comment": "example public key"
}

```

Output:

```

{
  "Location": "https://cloudfront.amazonaws.com/2019-03-26/public-key/
KDFB19YGCR002",
  "ETag": "E2QWRUHEXAMPLE",
  "PublicKey": {
    "Id": "KDFB19YGCR002",
    "CreatedTime": "2019-12-05T18:51:43.781Z",
    "PublicKeyConfig": {
      "CallerReference": "cli-example",
      "Name": "ExampleKey",
      "EncodedKey": "-----BEGIN PUBLIC KEY-----
\nMIIBIjANBgkqhkiG9w0BAQEFAAA0CAQ8AMIIBCgKCAQEAPMCA2Ks0lnd7IR+3pw
\nwd3H/7jPGwj8bLUmore7bX+oeGpZ6QmLAe/1U0WcmZX2u70dYcSIzB1ofZtcn4cJ
\nenHBAz03ohBY/L1tQGJfS2A+omnN6H16VZE1JCK8XSJyfze7MDLcUyHZETdxuvRb
\nA9X343/vMAuQPNhinFJ8Wdy8YBXSPpy7r95y1UQd9LfYTBzVZYG2tSesplc0kjM3\n2Uu
+oMWxQAw1NINnSLPinMVsutJy6Zq1V3McWNWe4T+STGtWhrPNqJEn45sIcCx4\nnq
+kGZ2NQ0FyIyT2eiLK0X5Rgb/a36E/aMk4VoDsaenBQgG7WLTnstb9sr7MIhS6A\nnrwIDAQAB\n-----
END PUBLIC KEY-----\n",
      "Comment": "example public key"
    }
  }
}

```

- Untuk detail API, lihat [CreatePublicKey](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Contoh kode berikut dibaca dalam kunci publik dan mengunggahnya ke Amazon CloudFront.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.services.cloudfront.CloudFrontClient;
import software.amazon.awssdk.services.cloudfront.model.CreatePublicKeyResponse;
import software.amazon.awssdk.utils.IoUtils;

import java.io.IOException;
import java.io.InputStream;
import java.util.UUID;

public class CreatePublicKey {
    private static final Logger logger =
        LoggerFactory.getLogger(CreatePublicKey.class);

    public static String createPublicKey(CloudFrontClient cloudFrontClient,
        String publicKeyFileName) {
        try (InputStream is =
            CreatePublicKey.class.getClassLoader().getResourceAsStream(publicKeyFileName)) {
            String publicKeyString = IoUtils.toUtf8String(is);
            CreatePublicKeyResponse createPublicKeyResponse = cloudFrontClient
                .createPublicKey(b -> b.publicKeyConfig(c -> c
                    .name("JavaCreatedPublicKey" + UUID.randomUUID())
                    .encodedKey(publicKeyString)
                    .callerReference(UUID.randomUUID().toString())));
            String createdPublicKeyId = createPublicKeyResponse.publicKey().id();
            logger.info("Public key created with id: {}", createdPublicKeyId);
            return createdPublicKeyId;
        } catch (IOException e) {
            throw new RuntimeException(e);
        }
    }
}
```

```
}  
}
```

- Untuk detail API, lihat [CreatePublicKey](#) di Referensi AWS SDK for Java 2.x API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **DeleteDistribution** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `DeleteDistribution`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Memulai dengan CloudFront distribusi dasar](#)

CLI

AWS CLI

Untuk menghapus CloudFront distribusi

Contoh berikut menghapus CloudFront distribusi dengan ID `EDFDVBD6EXAMPLE`. Sebelum Anda dapat menghapus distribusi, Anda harus menonaktifkannya. Untuk menonaktifkan distribusi, gunakan perintah pembaruan-distribusi. Untuk informasi selengkapnya, lihat contoh distribusi pembaruan.

Ketika distribusi dinonaktifkan, Anda dapat menghapusnya. Untuk menghapus distribusi, Anda harus menggunakan `--if-match` opsi untuk menyediakan distribusiETag. Untuk mendapatkanETag, gunakan `get-distribusi` atau `get-distribution-config` perintah.

```
aws cloudfront delete-distribution \  
  --id EDFDVBD6EXAMPLE \  
  --if-match E2QWRUHEXAMPLE
```

Ketika berhasil, perintah ini tidak memiliki output.

- Untuk detail API, lihat [DeleteDistribution](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Contoh kode berikut memperbarui distribusi ke dinonaktifkan, menggunakan pelayan yang menunggu perubahan diterapkan, lalu menghapus distribusi.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.core.internal.waiters.ResponseOrException;
import software.amazon.awssdk.services.cloudfront.CloudFrontClient;
import
    software.amazon.awssdk.services.cloudfront.model.DeleteDistributionResponse;
import software.amazon.awssdk.services.cloudfront.model.DistributionConfig;
import software.amazon.awssdk.services.cloudfront.model.GetDistributionResponse;
import software.amazon.awssdk.services.cloudfront.waiters.CloudFrontWaiter;

public class DeleteDistribution {
    private static final Logger logger =
        LoggerFactory.getLogger(DeleteDistribution.class);

    public static void deleteDistribution(final CloudFrontClient
cloudFrontClient, final String distributionId) {
        // First, disable the distribution by updating it.
        GetDistributionResponse response =
cloudFrontClient.getDistribution(b -> b
            .id(distributionId));
        String etag = response.eTag();
        DistributionConfig distConfig =
response.distribution().distributionConfig();

        cloudFrontClient.updateDistribution(builder -> builder
            .id(distributionId)
            .distributionConfig(builder1 -> builder1
                .cacheBehaviors(distConfig.cacheBehaviors()))
```

```

        .defaultCacheBehavior(distConfig.defaultCacheBehavior())
            .enabled(false)
            .origins(distConfig.origins())
            .comment(distConfig.comment())

        .callerReference(distConfig.callerReference())

        .defaultCacheBehavior(distConfig.defaultCacheBehavior())

        .priceClass(distConfig.priceClass())
            .aliases(distConfig.aliases())
            .logging(distConfig.logging())

        .defaultRootObject(distConfig.defaultRootObject())

        .customErrorResponses(distConfig.customErrorResponses())

        .httpVersion(distConfig.httpVersion())

        .isIPV6Enabled(distConfig.isIPV6Enabled())

        .restrictions(distConfig.restrictions())

        .viewerCertificate(distConfig.viewerCertificate())
            .webACLId(distConfig.webACLId())

        .originGroups(distConfig.originGroups())
            .ifMatch(etag));

        logger.info("Distribution [{}] is DISABLED, waiting for
deployment before deleting ...",
            distributionId);
        GetDistributionResponse distributionResponse;
        try (CloudFrontWaiter cfWaiter =
CloudFrontWaiter.builder().client(cloudFrontClient).build()) {
            ResponseOrException<GetDistributionResponse>
responseOrException = cfWaiter
                .waitUntilDistributionDeployed(builder ->
builder.id(distributionId)).matched();
            distributionResponse = responseOrException.response()
                .orElseThrow(() -> new
RuntimeException("Could not disable distribution"));
        }

```

```

        DeleteDistributionResponse deleteDistributionResponse =
cloudFrontClient
                                .deleteDistribution(builder -> builder
                                                .id(distributionId)

                                .ifMatch(distributionResponse.eTag()));
        if (deleteDistributionResponse.sdkHttpResponse().isSuccessful())
        {
            logger.info("Distribution [{}] DELETED", distributionId);
        }
    }
}

```

- Untuk detail API, lihat [DeleteDistribution](#) di Referensi AWS SDK for Java 2.x API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **GetCloudFrontOriginAccessIdentity** dengan CLI

Contoh kode berikut menunjukkan cara menggunakan `GetCloudFrontOriginAccessIdentity`.

CLI

AWS CLI

Untuk mendapatkan identitas akses CloudFront asal

Contoh berikut mendapatkan identitas akses CloudFront asal (OAI) dengan IDE74FTE3AEXAMPLE, termasuk ID kanonik ETag dan S3 terkait. ID OAI dikembalikan dalam output perintah `-access-identity` dan `create-cloud-front-origin -access-identities`. `list-cloud-front-origin`

```
aws cloudfront get-cloud-front-origin-access-identity --id E74FTE3AEXAMPLE
```

Output:

```
{
```

```

    "ETag": "E2QWRUHEXAMPLE",
    "CloudFrontOriginAccessIdentity": {
      "Id": "E74FTE3AEXAMPLE",
      "S3CanonicalUserId":
"cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
      "CloudFrontOriginAccessIdentityConfig": {
        "CallerReference": "cli-example",
        "Comment": "Example OAI"
      }
    }
  }
}

```

- Untuk detail API, lihat [GetCloudFrontOriginAccessIdentity](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell V4

Contoh 1: Contoh ini mengembalikan identitas akses CloudFront asal Amazon tertentu, yang ditentukan oleh parameter `-Id`. Meskipun parameter `-Id` tidak diperlukan, jika Anda tidak menentukannya, tidak ada hasil yang dikembalikan.

```
Get-CFCloudFrontOriginAccessIdentity -Id E3XXXXXXXXXXRT
```

Output:

```

CloudFrontOriginAccessIdentityConfig    Id
S3CanonicalUserId
-----
Amazon.CloudFront.Model.CloudFrontOr... E3XXXXXXXXXXRT
4b6e...

```

- Untuk detail API, lihat [GetCloudFrontOriginAccessIdentity](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V4).

Alat untuk PowerShell V5

Contoh 1: Contoh ini mengembalikan identitas akses CloudFront asal Amazon tertentu, yang ditentukan oleh parameter `-Id`. Meskipun parameter `-Id` tidak diperlukan, jika Anda tidak menentukannya, tidak ada hasil yang dikembalikan.


```
Get-CFCloudFrontOriginAccessIdentity -Id E3XXXXXXXXXXRT
```

Output:

```
CloudFrontOriginAccessIdentityConfig    Id
S3CanonicalUserId
-----
Amazon.CloudFront.Model.CloudFrontOr... E3XXXXXXXXXXRT
4b6e...
```

- Untuk detail API, lihat [GetCloudFrontOriginAccessIdentity](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V5).

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **GetCloudFrontOriginAccessIdentityConfig** dengan CLI

Contoh kode berikut menunjukkan cara menggunakan `GetCloudFrontOriginAccessIdentityConfig`.

CLI

AWS CLI

Untuk mendapatkan konfigurasi identitas akses CloudFront asal

Contoh berikut mendapatkan metadata tentang identitas akses CloudFront asal (OAI) dengan `E74FTE3AEXAMPLE`, termasuk nya. ETag ID OAI dikembalikan dalam output perintah `-access-identity` dan `create-cloud-front-origin -access-identities`. `list-cloud-front-origin`

```
aws cloudfront get-cloud-front-origin-access-identity-config --id E74FTE3AEXAMPLE
```

Output:

```
{
  "ETag": "E2QWRUHEXAMPLE",
  "CloudFrontOriginAccessIdentityConfig": {
```

```

    "CallerReference": "cli-example",
    "Comment": "Example OAI"
  }
}

```

- Untuk detail API, lihat [GetCloudFrontOriginAccessIdentityConfig](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell V4

Contoh 1: Contoh ini mengembalikan informasi konfigurasi tentang identitas akses CloudFront asal Amazon tunggal, yang ditentukan oleh parameter -Id. Kesalahan terjadi jika tidak ada parameter -Id yang ditentukan..

```
Get-CFCloudFrontOriginAccessIdentityConfig -Id E3XXXXXXXXXXRT
```

Output:

CallerReference	Comment
-----	-----
mycallerreference: 2/1/2011 1:16:32 PM	Caller
reference: 2/1/2011 1:16:32 PM	

- Untuk detail API, lihat [GetCloudFrontOriginAccessIdentityConfig](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V4).

Alat untuk PowerShell V5

Contoh 1: Contoh ini mengembalikan informasi konfigurasi tentang identitas akses CloudFront asal Amazon tunggal, yang ditentukan oleh parameter -Id. Kesalahan terjadi jika tidak ada parameter -Id yang ditentukan..

```
Get-CFCloudFrontOriginAccessIdentityConfig -Id E3XXXXXXXXXXRT
```

Output:

CallerReference	Comment
-----	-----

```
mycallerreference: 2/1/2011 1:16:32 PM  
reference: 2/1/2011 1:16:32 PM
```

Caller

- Untuk detail API, lihat [GetCloudFrontOriginAccessIdentityConfig](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V5).

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **GetDistribution** dengan CLI

Contoh kode berikut menunjukkan cara menggunakan `GetDistribution`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Memulai dengan CloudFront distribusi dasar](#)

CLI

AWS CLI

Untuk mendapatkan CloudFront distribusi

`get-distribution` Contoh berikut mendapatkan CloudFront distribusi dengan `IDEDFDVBD6EXAMPLE`, termasuk `ETag`. ID distribusi dikembalikan dalam perintah `create-distribution` dan `list-distributions`.

```
aws cloudfront get-distribution \  
--id EDFDVBD6EXAMPLE
```

Output:

```
{  
  "ETag": "E2QWRUHEXAMPLE",  
  "Distribution": {  
    "Id": "EDFDVBD6EXAMPLE",  
    "ARN": "arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE",  
    "Status": "Deployed",  
    "LastModifiedTime": "2019-12-04T23:35:41.433Z",  
  }  
}
```

```

    "InProgressInvalidationBatches": 0,
    "DomainName": "d1111111abcdef8.cloudfront.net",
    "ActiveTrustedSigners": {
        "Enabled": false,
        "Quantity": 0
    },
    "DistributionConfig": {
        "CallerReference": "cli-example",
        "Aliases": {
            "Quantity": 0
        },
        "DefaultRootObject": "index.html",
        "Origins": {
            "Quantity": 1,
            "Items": [
                {
                    "Id": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
                    "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                    "OriginPath": "",
                    "CustomHeaders": {
                        "Quantity": 0
                    },
                    "S3OriginConfig": {
                        "OriginAccessIdentity": ""
                    }
                }
            ]
        },
        "OriginGroups": {
            "Quantity": 0
        },
        "DefaultCacheBehavior": {
            "TargetOriginId": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-
example",
            "ForwardedValues": {
                "QueryString": false,
                "Cookies": {
                    "Forward": "none"
                },
                "Headers": {
                    "Quantity": 0
                },
                "QueryStringCacheKeys": {
                    "Quantity": 0
                }
            }
        }
    }
}

```

```
    }
  },
  "TrustedSigners": {
    "Enabled": false,
    "Quantity": 0
  },
  "ViewerProtocolPolicy": "allow-all",
  "MinTTL": 0,
  "AllowedMethods": {
    "Quantity": 2,
    "Items": [
      "HEAD",
      "GET"
    ],
    "CachedMethods": {
      "Quantity": 2,
      "Items": [
        "HEAD",
        "GET"
      ]
    }
  },
  "SmoothStreaming": false,
  "DefaultTTL": 86400,
  "MaxTTL": 31536000,
  "Compress": false,
  "LambdaFunctionAssociations": {
    "Quantity": 0
  },
  "FieldLevelEncryptionId": ""
},
"CacheBehaviors": {
  "Quantity": 0
},
"CustomErrorResponses": {
  "Quantity": 0
},
"Comment": "",
"Logging": {
  "Enabled": false,
  "IncludeCookies": false,
  "Bucket": "",
  "Prefix": ""
},
}
```

```
    "PriceClass": "PriceClass_All",
    "Enabled": true,
    "ViewerCertificate": {
      "CloudFrontDefaultCertificate": true,
      "MinimumProtocolVersion": "TLSv1",
      "CertificateSource": "cloudfront"
    },
    "Restrictions": {
      "GeoRestriction": {
        "RestrictionType": "none",
        "Quantity": 0
      }
    },
    "WebACLId": "",
    "HttpVersion": "http2",
    "IsIPV6Enabled": true
  }
}
```

- Untuk detail API, lihat [GetDistribution](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell V4

Contoh 1: Mengambil informasi untuk distribusi tertentu.

```
Get-CFDistribution -Id EXAMPLE0000ID
```

- Untuk detail API, lihat [GetDistribution](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V4).

Alat untuk PowerShell V5

Contoh 1: Mengambil informasi untuk distribusi tertentu.

```
Get-CFDistribution -Id EXAMPLE0000ID
```

- Untuk detail API, lihat [GetDistribution](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V5).

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **GetDistributionConfig** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `GetDistributionConfig`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Memulai dengan CloudFront distribusi dasar](#)

CLI

AWS CLI

Untuk mendapatkan konfigurasi CloudFront distribusi

Contoh berikut mendapatkan metadata tentang CloudFront distribusi dengan `IDEDFDVBD6EXAMPLE`, termasuk nya. ETag ID distribusi dikembalikan dalam perintah `create-distribution` dan `list-distributions`.

```
aws cloudfront get-distribution-config \
  --id EDFDVBD6EXAMPLE
```

Output:

```
{
  "ETag": "E2QWRUHEXAMPLE",
  "DistributionConfig": {
    "CallerReference": "cli-example",
    "Aliases": {
      "Quantity": 0
    },
    "DefaultRootObject": "index.html",
    "Origins": {
      "Quantity": 1,
      "Items": [
        {
          "Id": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
          "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
```

```
        "OriginPath": "",
        "CustomHeaders": {
            "Quantity": 0
        },
        "S3OriginConfig": {
            "OriginAccessIdentity": ""
        }
    }
]
},
"OriginGroups": {
    "Quantity": 0
},
"DefaultCacheBehavior": {
    "TargetOriginId": "amzn-s3-demo-bucket.s3.amazonaws.com-cli-example",
    "ForwardedValues": {
        "QueryString": false,
        "Cookies": {
            "Forward": "none"
        },
        "Headers": {
            "Quantity": 0
        },
        "QueryStringCacheKeys": {
            "Quantity": 0
        }
    },
    "TrustedSigners": {
        "Enabled": false,
        "Quantity": 0
    },
    "ViewerProtocolPolicy": "allow-all",
    "MinTTL": 0,
    "AllowedMethods": {
        "Quantity": 2,
        "Items": [
            "HEAD",
            "GET"
        ],
        "CachedMethods": {
            "Quantity": 2,
            "Items": [
                "HEAD",
                "GET"
            ]
        }
    }
}
```



```

        ]
    },
    "SmoothStreaming": false,
    "DefaultTTL": 86400,
    "MaxTTL": 31536000,
    "Compress": false,
    "LambdaFunctionAssociations": {
        "Quantity": 0
    },
    "FieldLevelEncryptionId": ""
},
"CacheBehaviors": {
    "Quantity": 0
},
"CustomErrorResponses": {
    "Quantity": 0
},
"Comment": "",
"Logging": {
    "Enabled": false,
    "IncludeCookies": false,
    "Bucket": "",
    "Prefix": ""
},
"PriceClass": "PriceClass_All",
"Enabled": true,
"ViewerCertificate": {
    "CloudFrontDefaultCertificate": true,
    "MinimumProtocolVersion": "TLSv1",
    "CertificateSource": "cloudfront"
},
"Restrictions": {
    "GeoRestriction": {
        "RestrictionType": "none",
        "Quantity": 0
    }
},
"WebACLId": "",
"HttpVersion": "http2",
"IsIPV6Enabled": true
}
}

```

- Untuk detail API, lihat [GetDistributionConfig](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell V4

Contoh 1: Mengambil konfigurasi untuk distribusi tertentu.

```
Get-CFDistributionConfig -Id EXAMPLE0000ID
```

- Untuk detail API, lihat [GetDistributionConfig](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V4).

Alat untuk PowerShell V5

Contoh 1: Mengambil konfigurasi untuk distribusi tertentu.

```
Get-CFDistributionConfig -Id EXAMPLE0000ID
```

- Untuk detail API, lihat [GetDistributionConfig](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V5).

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
class CloudFrontWrapper:
    """Encapsulates Amazon CloudFront operations."""

    def __init__(self, cloudfront_client):
        """
        :param cloudfront_client: A Boto3 CloudFront client
        """
```

```
self.cloudfront_client = cloudfront_client

def update_distribution(self):
    distribution_id = input(
        "This script updates the comment for a CloudFront distribution.\n"
        "Enter a CloudFront distribution ID: "
    )

    distribution_config_response =
self.cloudfront_client.get_distribution_config(
    Id=distribution_id
)
    distribution_config = distribution_config_response["DistributionConfig"]
    distribution_etag = distribution_config_response["ETag"]

    distribution_config["Comment"] = input(
        f"\nThe current comment for distribution {distribution_id} is "
        f"'{distribution_config['Comment']}'.\n"
        f"Enter a new comment: "
    )
    self.cloudfront_client.update_distribution(
        DistributionConfig=distribution_config,
        Id=distribution_id,
        IfMatch=distribution_etag,
    )
    print("Done!")
```

- Untuk detail API, lihat [GetDistributionConfig](#) di AWS SDK for Python (Boto3) Referensi API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ListCloudFrontOriginAccessIdentities** dengan CLI

Contoh kode berikut menunjukkan cara menggunakan `ListCloudFrontOriginAccessIdentities`.

CLI

AWS CLI

Untuk mencantumkan identitas akses CloudFront asal

Contoh berikut mendapatkan daftar identitas akses CloudFront asal (OAIs) di AWS akun Anda:

```
aws cloudfront list-cloud-front-origin-access-identities
```

Output:

```
{
  "CloudFrontOriginAccessIdentityList": {
    "Items": [
      {
        "Id": "E74FTE3AEXAMPLE",
        "S3CanonicalUserId":
        "cd13868f797c227fbea2830611a26fe0a21ba1b826ab4bed9b7771c9aEXAMPLE",
        "Comment": "Example OAI"
      },
      {
        "Id": "EH1HDMBEXAMPLE",
        "S3CanonicalUserId":
        "1489f6f2e6faacaae7ff64c4c3e6956c24f78788abfc1718c3527c263bf7a17EXAMPLE",
        "Comment": "Test OAI"
      },
      {
        "Id": "E2X2C9TEXAMPLE",
        "S3CanonicalUserId":
        "cbfeebb915a64749f9be546a45b3fcfd3a31c779673c13c4dd460911ae402c2EXAMPLE",
        "Comment": "Example OAI #2"
      }
    ]
  }
}
```

- Untuk detail API, lihat [ListCloudFrontOriginAccessIdentities](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell V4

Contoh 1: Contoh ini mengembalikan daftar identitas akses CloudFront asal Amazon. Karena -MaxItem parameter menentukan nilai 2, hasilnya mencakup dua identitas.

```
Get-CFCloudFrontOriginAccessIdentityList -MaxItem 2
```

Output:

```
IsTruncated : True
Items       : {E326XXXXXXXXXT, E1YWXXXXXXXX9B}
Marker      :
MaxItems    : 2
NextMarker  : E1YXXXXXXXXXX9B
Quantity    : 2
```

- Untuk detail API, lihat [ListCloudFrontOriginAccessIdentities](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V4).

Alat untuk PowerShell V5

Contoh 1: Contoh ini mengembalikan daftar identitas akses CloudFront asal Amazon. Karena -MaxItem parameter menentukan nilai 2, hasilnya mencakup dua identitas.

```
Get-CFCloudFrontOriginAccessIdentityList -MaxItem 2
```

Output:

```
IsTruncated : True
Items       : {E326XXXXXXXXXT, E1YWXXXXXXXX9B}
Marker      :
MaxItems    : 2
NextMarker  : E1YXXXXXXXXXX9B
Quantity    : 2
```

- Untuk detail API, lihat [ListCloudFrontOriginAccessIdentities](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V5).

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **ListDistributions** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `ListDistributions`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Memulai dengan WAF](#)

CLI

AWS CLI

Untuk daftar CloudFront distribusi

Contoh berikut mendapatkan daftar CloudFront distribusi di AWS akun Anda.

```
aws cloudfront list-distributions
```

Output:

```
{
  "DistributionList": {
    "Items": [
      {
        "Id": "E23YS80EXAMPLE",
        "ARN": "arn:aws:cloudfront::123456789012:distribution/E23YS80EXAMPLE",
        "Status": "Deployed",
        "LastModifiedTime": "2024-08-05T18:23:40.375000+00:00",
        "DomainName": "abcdefgh12ijk.cloudfront.net",
        "Aliases": {
          "Quantity": 0
        },
        "Origins": {
          "Quantity": 1,
          "Items": [
            {
```

```

        "Id": "amzn-s3-demo-bucket.s3.us-
east-1.amazonaws.com",
        "DomainName": "amzn-s3-demo-bucket.s3.us-
east-1.amazonaws.com",
        "OriginPath": "",
        "CustomHeaders": {
            "Quantity": 0
        },
        "S3OriginConfig": {
            "OriginAccessIdentity": ""
        },
        "ConnectionAttempts": 3,
        "ConnectionTimeout": 10,
        "OriginShield": {
            "Enabled": false
        },
        "OriginAccessControlId": "EIAP8PEXAMPLE"
    }
]
},
"OriginGroups": {
    "Quantity": 0
},
"DefaultCacheBehavior": {
    "TargetOriginId": "amzn-s3-demo-bucket.s3.us-
east-1.amazonaws.com",
    "TrustedSigners": {
        "Enabled": false,
        "Quantity": 0
    },
    "TrustedKeyGroups": {
        "Enabled": false,
        "Quantity": 0
    },
    "ViewerProtocolPolicy": "allow-all",
    "AllowedMethods": {
        "Quantity": 2,
        "Items": [
            "HEAD",
            "GET"
        ],
        "CachedMethods": {
            "Quantity": 2,
            "Items": [

```

```

        "HEAD",
        "GET"
    ]
    },
    "SmoothStreaming": false,
    "Compress": true,
    "LambdaFunctionAssociations": {
        "Quantity": 0
    },
    "FunctionAssociations": {
        "Quantity": 0
    },
    "FieldLevelEncryptionId": "",
    "CachePolicyId": "658327ea-f89d-4fab-a63d-7e886EXAMPLE"
},
"CacheBehaviors": {
    "Quantity": 0
},
"CustomErrorResponses": {
    "Quantity": 0
},
"Comment": "",
"PriceClass": "PriceClass_All",
"Enabled": true,
"ViewerCertificate": {
    "CloudFrontDefaultCertificate": true,
    "SSLSupportMethod": "vip",
    "MinimumProtocolVersion": "TLSv1",
    "CertificateSource": "cloudfront"
},
"Restrictions": {
    "GeoRestriction": {
        "RestrictionType": "none",
        "Quantity": 0
    }
},
"WebACLId": "",
"HttpVersion": "HTTP2",
"IsIPV6Enabled": true,
"Staging": false
}
]
}

```



```
}
```

- Untuk detail API, lihat [ListDistributions](#) di Referensi AWS CLI Perintah.

PowerShell

Alat untuk PowerShell V4

Contoh 1: Mengembalikan distribusi.

```
Get-CFDistributionList
```

- Untuk detail API, lihat [ListDistributions](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V4).

Alat untuk PowerShell V5

Contoh 1: Mengembalikan distribusi.

```
Get-CFDistributionList
```

- Untuk detail API, lihat [ListDistributions](#) di Referensi Alat AWS untuk PowerShell Cmdlet (V5).

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
class CloudFrontWrapper:
    """Encapsulates Amazon CloudFront operations."""

    def __init__(self, cloudfront_client):
        """
        :param cloudfront_client: A Boto3 CloudFront client
        """
        self.cloudfront_client = cloudfront_client
```

```

def list_distributions(self):
    print("CloudFront distributions:\n")
    distributions = self.cloudfront_client.list_distributions()
    if distributions["DistributionList"]["Quantity"] > 0:
        for distribution in distributions["DistributionList"]["Items"]:
            print(f"Domain: {distribution['DomainName']}")
            print(f"Distribution Id: {distribution['Id']}")
            print(
                f"Certificate Source: "
                f"{distribution['ViewerCertificate']['CertificateSource']}"
            )
            if distribution["ViewerCertificate"]["CertificateSource"] ==
"acm":
                print(
                    f"Certificate: {distribution['ViewerCertificate']
['Certificate']}"
                )
                print("")
            else:
                print("No CloudFront distributions detected.")

```

- Untuk detail API, lihat [ListDistributions](#) di AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK for SAP ABAP

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

TRY.

```

oo_result = lo_fnt->listdistributions( ). " oo_result is returned for
testing purposes. "

```

```

MESSAGE 'Retrieved list of CloudFront distributions.' TYPE 'I'.

```

```

CATCH /aws1/cx_fntinvalidargument.

```

```

MESSAGE 'Invalid argument provided.' TYPE 'E'.

```

```
ENDTRY.
```

- Untuk detail API, lihat [ListDistributions](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan **UpdateDistribution** dengan AWS SDK atau CLI

Contoh kode berikut menunjukkan cara menggunakan `UpdateDistribution`.

Contoh tindakan adalah kutipan kode dari program yang lebih besar dan harus dijalankan dalam konteks. Anda dapat melihat tindakan ini dalam konteks dalam contoh kode berikut:

- [Memulai dengan CloudFront distribusi dasar](#)

CLI

AWS CLI

Contoh 1: Untuk memperbarui objek root default CloudFront distribusi

Contoh berikut memperbarui objek root default `index.html` untuk CloudFront distribusi dengan `IDEDFDVBD6EXAMPLE`.

```
aws cloudfront update-distribution \  
  --id EDFDVBD6EXAMPLE \  
  --default-root-object index.html
```

Output:

```
{  
  "ETag": "E2QWRUHEXAMPLE",  
  "Distribution": {  
    "Id": "EDFDVBD6EXAMPLE",  
    "ARN": "arn:aws:cloudfront::123456789012:distribution/EDFDVBD6EXAMPLE",  
    "Status": "InProgress",  
    "LastModifiedTime": "2019-12-06T18:55:39.870Z",  
    "InProgressInvalidationBatches": 0,  
  }  
}
```

```

"DomainName": "d111111abcdef8.cloudfront.net",
"ActiveTrustedSigners": {
  "Enabled": false,
  "Quantity": 0
},
"DistributionConfig": {
  "CallerReference": "6b10378d-49be-4c4b-a642-419ccaf8f3b5",
  "Aliases": {
    "Quantity": 0
  },
  "DefaultRootObject": "index.html",
  "Origins": {
    "Quantity": 1,
    "Items": [
      {
        "Id": "example-website",
        "DomainName": "www.example.com",
        "OriginPath": "",
        "CustomHeaders": {
          "Quantity": 0
        },
        "CustomOriginConfig": {
          "HTTPPort": 80,
          "HTTPSPort": 443,
          "OriginProtocolPolicy": "match-viewer",
          "OriginSslProtocols": {
            "Quantity": 2,
            "Items": [
              "SSLv3",
              "TLSv1"
            ]
          },
          "OriginReadTimeout": 30,
          "OriginKeepaliveTimeout": 5
        }
      }
    ]
  },
  "OriginGroups": {
    "Quantity": 0
  },
  "DefaultCacheBehavior": {
    "TargetOriginId": "example-website",
    "ForwardedValues": {

```

```
    "QueryString": false,
    "Cookies": {
      "Forward": "none"
    },
    "Headers": {
      "Quantity": 1,
      "Items": [
        "*"
      ]
    },
    "QueryStringCacheKeys": {
      "Quantity": 0
    }
  },
  "TrustedSigners": {
    "Enabled": false,
    "Quantity": 0
  },
  "ViewerProtocolPolicy": "allow-all",
  "MinTTL": 0,
  "AllowedMethods": {
    "Quantity": 2,
    "Items": [
      "HEAD",
      "GET"
    ],
    "CachedMethods": {
      "Quantity": 2,
      "Items": [
        "HEAD",
        "GET"
      ]
    }
  },
  "SmoothStreaming": false,
  "DefaultTTL": 86400,
  "MaxTTL": 31536000,
  "Compress": false,
  "LambdaFunctionAssociations": {
    "Quantity": 0
  },
  "FieldLevelEncryptionId": ""
},
"CacheBehaviors": {
```

```

        "Quantity": 0
    },
    "CustomErrorResponses": {
        "Quantity": 0
    },
    "Comment": "",
    "Logging": {
        "Enabled": false,
        "IncludeCookies": false,
        "Bucket": "",
        "Prefix": ""
    },
    "PriceClass": "PriceClass_All",
    "Enabled": true,
    "ViewerCertificate": {
        "CloudFrontDefaultCertificate": true,
        "MinimumProtocolVersion": "TLSv1",
        "CertificateSource": "cloudfront"
    },
    "Restrictions": {
        "GeoRestriction": {
            "RestrictionType": "none",
            "Quantity": 0
        }
    },
    "WebACLId": "",
    "HttpVersion": "http1.1",
    "IsIPV6Enabled": true
    }
}

```

Contoh 2: Untuk memperbarui CloudFront distribusi

Contoh berikut menonaktifkan CloudFront distribusi dengan ID EMLARXS9EXAMPLE dengan menyediakan konfigurasi distribusi dalam file JSON bernama `dist-config-disable.json`. Untuk memperbarui distribusi, Anda harus menggunakan `--if-match` opsi untuk menyediakan distribusiETag. Untuk mendapatkanETag, gunakan `get-distribusi` atau `get-distribution-config` perintah. Perhatikan bahwa `Enabled` bidang diatur ke `false` dalam file JSON.

Setelah Anda menggunakan contoh berikut untuk menonaktifkan distribusi, Anda dapat menggunakan perintah `hapus-distribusi` untuk menghapusnya.

```
aws cloudfront update-distribution \  
  --id EMLARXS9EXAMPLE \  
  --if-match E2QWRUHEXAMPLE \  
  --distribution-config file://dist-config-disable.json
```

Isi dari dist-config-disable.json:

```
{  
  "CallerReference": "cli-1574382155-496510",  
  "Aliases": {  
    "Quantity": 0  
  },  
  "DefaultRootObject": "index.html",  
  "Origins": {  
    "Quantity": 1,  
    "Items": [  
      {  
        "Id": "amzn-s3-demo-bucket.s3.amazonaws.com-1574382155-273939",  
        "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",  
        "OriginPath": "",  
        "CustomHeaders": {  
          "Quantity": 0  
        },  
        "S3OriginConfig": {  
          "OriginAccessIdentity": ""  
        }  
      }  
    ]  
  },  
  "OriginGroups": {  
    "Quantity": 0  
  },  
  "DefaultCacheBehavior": {  
    "TargetOriginId": "amzn-s3-demo-  
bucket.s3.amazonaws.com-1574382155-273939",  
    "ForwardedValues": {  
      "QueryString": false,  
      "Cookies": {  
        "Forward": "none"  
      },  
      "Headers": {  
        "Quantity": 0  
      },  
    },  
  },  
}
```

```
        "QueryStringCacheKeys": {
            "Quantity": 0
        },
    },
    "TrustedSigners": {
        "Enabled": false,
        "Quantity": 0
    },
    "ViewerProtocolPolicy": "allow-all",
    "MinTTL": 0,
    "AllowedMethods": {
        "Quantity": 2,
        "Items": [
            "HEAD",
            "GET"
        ],
        "CachedMethods": {
            "Quantity": 2,
            "Items": [
                "HEAD",
                "GET"
            ]
        }
    },
    "SmoothStreaming": false,
    "DefaultTTL": 86400,
    "MaxTTL": 31536000,
    "Compress": false,
    "LambdaFunctionAssociations": {
        "Quantity": 0
    },
    "FieldLevelEncryptionId": ""
},
"CacheBehaviors": {
    "Quantity": 0
},
"CustomErrorResponses": {
    "Quantity": 0
},
"Comment": "",
"Logging": {
    "Enabled": false,
    "IncludeCookies": false,
    "Bucket": "",
```



```

    "Prefix": ""
  },
  "PriceClass": "PriceClass_All",
  "Enabled": false,
  "ViewerCertificate": {
    "CloudFrontDefaultCertificate": true,
    "MinimumProtocolVersion": "TLSv1",
    "CertificateSource": "cloudfront"
  },
  "Restrictions": {
    "GeoRestriction": {
      "RestrictionType": "none",
      "Quantity": 0
    }
  },
  "WebACLId": "",
  "HttpVersion": "http2",
  "IsIPV6Enabled": true
}

```

Output:

```

{
  "ETag": "E9LHASXEXAMPLE",
  "Distribution": {
    "Id": "EMLARXS9EXAMPLE",
    "ARN": "arn:aws:cloudfront::123456789012:distribution/EMLARXS9EXAMPLE",
    "Status": "InProgress",
    "LastModifiedTime": "2019-12-06T18:32:35.553Z",
    "InProgressInvalidationBatches": 0,
    "DomainName": "d1111111abcdef8.cloudfront.net",
    "ActiveTrustedSigners": {
      "Enabled": false,
      "Quantity": 0
    }
  },
  "DistributionConfig": {
    "CallerReference": "cli-1574382155-496510",
    "Aliases": {
      "Quantity": 0
    },
    "DefaultRootObject": "index.html",
    "Origins": {
      "Quantity": 1,

```

```

        "Items": [
            {
                "Id": "amzn-s3-demo-
bucket.s3.amazonaws.com-1574382155-273939",
                "DomainName": "amzn-s3-demo-bucket.s3.amazonaws.com",
                "OriginPath": "",
                "CustomHeaders": {
                    "Quantity": 0
                },
                "S3OriginConfig": {
                    "OriginAccessIdentity": ""
                }
            }
        ],
        "OriginGroups": {
            "Quantity": 0
        },
        "DefaultCacheBehavior": {
            "TargetOriginId": "amzn-s3-demo-
bucket.s3.amazonaws.com-1574382155-273939",
            "ForwardedValues": {
                "QueryString": false,
                "Cookies": {
                    "Forward": "none"
                },
                "Headers": {
                    "Quantity": 0
                },
                "QueryStringCacheKeys": {
                    "Quantity": 0
                }
            },
            "TrustedSigners": {
                "Enabled": false,
                "Quantity": 0
            },
            "ViewerProtocolPolicy": "allow-all",
            "MinTTL": 0,
            "AllowedMethods": {
                "Quantity": 2,
                "Items": [
                    "HEAD",
                    "GET"
                ]
            }
        }
    ]
}

```

```
    ],
    "CachedMethods": {
      "Quantity": 2,
      "Items": [
        "HEAD",
        "GET"
      ]
    }
  },
  "SmoothStreaming": false,
  "DefaultTTL": 86400,
  "MaxTTL": 31536000,
  "Compress": false,
  "LambdaFunctionAssociations": {
    "Quantity": 0
  },
  "FieldLevelEncryptionId": ""
},
"CacheBehaviors": {
  "Quantity": 0
},
"CustomErrorResponses": {
  "Quantity": 0
},
"Comment": "",
"Logging": {
  "Enabled": false,
  "IncludeCookies": false,
  "Bucket": "",
  "Prefix": ""
},
"PriceClass": "PriceClass_All",
"Enabled": false,
"ViewerCertificate": {
  "CloudFrontDefaultCertificate": true,
  "MinimumProtocolVersion": "TLSv1",
  "CertificateSource": "cloudfront"
},
"Restrictions": {
  "GeoRestriction": {
    "RestrictionType": "none",
    "Quantity": 0
  }
},
},
```

```

        "WebACLId": "",
        "HttpVersion": "http2",
        "IsIPV6Enabled": true
    }
}

```

- Untuk detail API, lihat [UpdateDistribution](#) di Referensi AWS CLI Perintah.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```

import software.amazon.awssdk.regions.Region;
import software.amazon.awssdk.services.cloudfront.CloudFrontClient;
import software.amazon.awssdk.services.cloudfront.model.GetDistributionRequest;
import software.amazon.awssdk.services.cloudfront.model.GetDistributionResponse;
import software.amazon.awssdk.services.cloudfront.model.Distribution;
import software.amazon.awssdk.services.cloudfront.model.DistributionConfig;
import
    software.amazon.awssdk.services.cloudfront.model.UpdateDistributionRequest;
import software.amazon.awssdk.services.cloudfront.model.CloudFrontException;

/**
 * Before running this Java V2 code example, set up your development
 * environment, including your credentials.
 *
 * For more information, see the following documentation topic:
 *
 * https://docs.aws.amazon.com/sdk-for-java/latest/developer-guide/get-
 * started.html
 */
public class ModifyDistribution {
    public static void main(String[] args) {
        final String usage = ""

```

```
Usage:
    <id>\s

Where:
    id - the id value of the distribution.\s
""";

if (args.length != 1) {
    System.out.println(usage);
    System.exit(1);
}

String id = args[0];
CloudFrontClient cloudFrontClient = CloudFrontClient.builder()
    .region(Region.AWS_GLOBAL)
    .build();

modDistribution(cloudFrontClient, id);
cloudFrontClient.close();
}

public static void modDistribution(CloudFrontClient cloudFrontClient, String
idVal) {
    try {
        // Get the Distribution to modify.
        GetDistributionRequest disRequest = GetDistributionRequest.builder()
            .id(idVal)
            .build();

        GetDistributionResponse response =
cloudFrontClient.getDistribution(disRequest);
        Distribution disObject = response.distribution();
        DistributionConfig config = disObject.distributionConfig();

        // Create a new DistributionConfig object and add new values to
comment and
        // aliases
        DistributionConfig config1 = DistributionConfig.builder()
            .aliases(config.aliases()) // You can pass in new values here
            .comment("New Comment")
            .cacheBehaviors(config.cacheBehaviors())
            .priceClass(config.priceClass())
            .defaultCacheBehavior(config.defaultCacheBehavior())
            .enabled(config.enabled())
```

```

        .callerReference(config.callerReference())
        .logging(config.logging())
        .originGroups(config.originGroups())
        .origins(config.origins())
        .restrictions(config.restrictions())
        .defaultRootObject(config.defaultRootObject())
        .webACLId(config.webACLId())
        .httpVersion(config.httpVersion())
        .viewerCertificate(config.viewerCertificate())
        .customErrorResponses(config.customErrorResponses())
        .build();

        UpdateDistributionRequest updateDistributionRequest =
UpdateDistributionRequest.builder()
        .distributionConfig(config1)
        .id(disObject.id())
        .ifMatch(response.eTag())
        .build();

        cloudFrontClient.updateDistribution(updateDistributionRequest);

    } catch (CloudFrontException e) {
        System.err.println(e.awsErrorDetails().errorMessage());
        System.exit(1);
    }
}
}

```

- Untuk detail API, lihat [UpdateDistribution](#) di Referensi AWS SDK for Java 2.x API.

Python

SDK untuk Python (Boto3)

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
class CloudFrontWrapper:
```

```
"""Encapsulates Amazon CloudFront operations."""

def __init__(self, cloudfront_client):
    """
    :param cloudfront_client: A Boto3 CloudFront client
    """
    self.cloudfront_client = cloudfront_client

def update_distribution(self):
    distribution_id = input(
        "This script updates the comment for a CloudFront distribution.\n"
        "Enter a CloudFront distribution ID: "
    )

    distribution_config_response =
self.cloudfront_client.get_distribution_config(
    Id=distribution_id
)
    distribution_config = distribution_config_response["DistributionConfig"]
    distribution_etag = distribution_config_response["ETag"]

    distribution_config["Comment"] = input(
        f"\nThe current comment for distribution {distribution_id} is "
        f"'{distribution_config['Comment']}'.\n"
        f"Enter a new comment: "
    )
    self.cloudfront_client.update_distribution(
        DistributionConfig=distribution_config,
        Id=distribution_id,
        IfMatch=distribution_etag,
    )
    print("Done!")
```

- Untuk detail API, lihat [UpdateDistribution](#) di AWS SDK for Python (Boto3) Referensi API.

SAP ABAP

SDK for SAP ABAP

 Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

```
TRY.
    " Get the current distribution configuration and ETag "
    DATA(lo_distribution_config_result) = lo_fnt-
>getdistributionconfig( iv_id = iv_distribution_id ).
    DATA(lo_old_config) = lo_distribution_config_result-
>get_distributionconfig( ).
    DATA(lv_etag) = lo_distribution_config_result->get_etag( ).

    " Create a new distribution config with the updated comment "
    " Since the config object is immutable, we need to create a new one with
all existing values "
    DATA(lo_new_config) = NEW /aws1/cl_fntdistributionconfig(
        iv_callerreference = lo_old_config->get_callerreference( )
        io_aliases = lo_old_config->get_aliases( )
        iv_defaultrootobject = lo_old_config->get_defaultrootobject( )
        io_origins = lo_old_config->get_origins( )
        io_origingroups = lo_old_config->get_origingroups( )
        io_defaultcachebehavior = lo_old_config->get_defaultcachebehavior( )
        io_cachebehaviors = lo_old_config->get_cachebehaviors( )
        io_customerrorresponses = lo_old_config->get_customerrorresponses( )
        iv_comment = iv_comment
        io_logging = lo_old_config->get_logging( )
        iv_priceclass = lo_old_config->get_priceclass( )
        iv_enabled = lo_old_config->get_enabled( )
        io_viewercertificate = lo_old_config->get_viewercertificate( )
        io_restrictions = lo_old_config->get_restrictions( )
        iv_webaclid = lo_old_config->get_webaclid( )
        iv_httpversion = lo_old_config->get_httpversion( )
        iv_isipv6enabled = lo_old_config->get_isipv6enabled( ) ).

    " Update the distribution with the modified configuration "
    lo_fnt->updatedistribution(
```



```
io_distributionconfig = lo_new_config
iv_id = iv_distribution_id
iv_ifmatch = lv_etag ).
MESSAGE 'CloudFront distribution updated successfully.' TYPE 'I'.
CATCH /aws1/cx_fntnosuchdistribution.
MESSAGE 'Distribution does not exist.' TYPE 'E'.
CATCH /aws1/cx_fntpreconditionfailed.
MESSAGE 'Precondition failed - ETag mismatch.' TYPE 'E'.
CATCH /aws1/cx_fntinvalidifmatchvrs.
MESSAGE 'Invalid If-Match version.' TYPE 'E'.
ENDTRY.
```

- Untuk detail API, lihat [UpdateDistribution](#) di AWS SDK untuk referensi SAP ABAP API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Skenario untuk CloudFront menggunakan AWS SDKs

Contoh kode berikut menunjukkan kepada Anda bagaimana menerapkan skenario umum CloudFront dengan AWS SDKs. Skenario ini menunjukkan kepada Anda bagaimana menyelesaikan tugas tertentu dengan memanggil beberapa fungsi di dalam CloudFront atau dikombinasikan dengan yang lain Layanan AWS. Setiap skenario menyertakan tautan ke kode sumber lengkap, di mana Anda dapat menemukan instruksi tentang cara mengatur dan menjalankan kode.

Skenario menargetkan tingkat pengalaman menengah untuk membantu Anda memahami tindakan layanan dalam konteks.

Contoh

- [Buat SDK sumber daya manajer SaaS AWS](#)
- [Hapus sumber CloudFront penandatanganan menggunakan AWS SDK](#)
- [Memulai dengan CloudFront distribusi dasar](#)
- [Memulai dengan WAF](#)
- [Buat cookie yang ditandatangani URLs dan menggunakan AWS SDK](#)

Buat SDK sumber daya manajer SaaS AWS

Contoh kode berikut menunjukkan cara membuat distribusi multi-tenant dan penyewa distribusi dengan berbagai konfigurasi.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Contoh berikut menunjukkan cara membuat distribusi multi-tenant dengan parameter dan sertifikat wildcard.

```
import software.amazon.awssdk.core.internal.waiters.ResponseOrException;
import software.amazon.awssdk.services.cloudfront.CloudFrontClient;
import software.amazon.awssdk.services.cloudfront.model.ConnectionMode;
import
    software.amazon.awssdk.services.cloudfront.model.CreateDistributionResponse;
import software.amazon.awssdk.services.cloudfront.model.Distribution;
import software.amazon.awssdk.services.cloudfront.model.GetDistributionResponse;
import software.amazon.awssdk.services.cloudfront.model.HttpVersion;
import software.amazon.awssdk.services.cloudfront.model.Method;
import software.amazon.awssdk.services.cloudfront.model.SSLSupportMethod;
import software.amazon.awssdk.services.cloudfront.model.ViewerProtocolPolicy;
import software.amazon.awssdk.services.cloudfront.waiters.CloudFrontWaiter;
import software.amazon.awssdk.services.s3.S3Client;

import java.time.Instant;

public class CreateMultiTenantDistribution {
    public static Distribution
    CreateMultiTenantDistributionWithCert(CloudFrontClient cloudFrontClient,
                                         S3Client
    s3Client,
                                         final String
    bucketName,
```

```

final String
certificateArn) {
    // fetch the origin info if necessary
    final String region = s3Client.headBucket(b ->
b.bucket(bucketName)).sdkHttpResponse().headers()
        .get("x-amz-bucket-region").get(0);
    final String originDomain = bucketName + ".s3." + region +
".amazonaws.com";
    String originId = originDomain; // Use the originDomain value for the
originId.

    CreateDistributionResponse createDistResponse =
cloudFrontClient.createDistribution(builder -> builder
        .distributionConfig(b1 -> b1
            .httpVersion(HttpVersion.HTTP2)
            .enabled(true)
            .comment("Template Distribution with cert built with
java")

            .connectionMode(ConnectionMode.TENANT_ONLY)
            .callerReference(Instant.now().toString())
            .viewerCertificate(certBuilder -> certBuilder
                .acmCertificateArn(certificateArn)
                .sslSupportMethod(SSLSupportMethod.SNI_ONLY))

            .origins(b2 -> b2
                .quantity(1)
                .items(b3 -> b3
                    .domainName(originDomain)
                    .id(originId)
                    .originPath("/{tenantName}")
                    .s3OriginConfig(builder4 -> builder4
                        .originAccessIdentity(
                            ""))))

            .tenantConfig(b5 -> b5
                .parameterDefinitions(b6 -> b6
                    .name("tenantName")
                    .definition(b7 -> b7
                        .stringSchema(b8 -> b8
                            .comment("tenantName
value")

                            .defaultValue("root")
                            .required(false))))

                .defaultCacheBehavior(b2 -> b2

            .viewerProtocolPolicy(ViewerProtocolPolicy.ALLOW_ALL)

```

```

        .targetOriginId(originId)
        .cachePolicyId("658327ea-f89d-4fab-
a63d-7e88639e58f6") // CachingOptimized Policy
        .allowedMethods(b4 -> b4
            .quantity(2)
            .items(Method.HEAD, Method.GET)))
    ));

    final Distribution distribution = createDistResponse.distribution();
    try (CloudFrontWaiter cfWaiter =
CloudFrontWaiter.builder().client(cloudFrontClient).build()) {
        ResponseOrException<GetDistributionResponse> responseOrException =
cfWaiter
            .waitUntilDistributionDeployed(builder ->
builder.id(distribution.id()))
            .matched();
        responseOrException.response()
            .orElseThrow(() -> new RuntimeException("Distribution not
created"));
    }
    return distribution;
}

public static Distribution
CreateMultiTenantDistributionNoCert(CloudFrontClient cloudFrontClient,
                                    S3Client s3Client,
                                    final String
bucketName) {
    // fetch the origin info if necessary
    final String region = s3Client.headBucket(b ->
b.bucket(bucketName)).sdkHttpResponse().headers()
        .get("x-amz-bucket-region").get(0);
    final String originDomain = bucketName + ".s3." + region +
".amazonaws.com";
    String originId = originDomain; // Use the originDomain value for the
originId.

    CreateDistributionResponse createDistResponse =
cloudFrontClient.createDistribution(builder -> builder
        .distributionConfig(b1 -> b1
            .httpVersion(HttpVersion.HTTP2)
            .enabled(true)
            .comment("Template Distribution with cert built with
java"))

```

```

        .connectionMode(ConnectionMode.TENANT_ONLY)
        .callerReference(Instant.now().toString())
        .origins(b2 -> b2
            .quantity(1)
            .items(b3 -> b3
                .domainName(originDomain)
                .id(originId)
                .originPath("/{tenantName}")
                .s3OriginConfig(builder4 -> builder4
                    .originAccessIdentity(
                        ""))))
        .tenantConfig(b5 -> b5
            .parameterDefinitions(b6 -> b6
                .name("tenantName")
                .definition(b7 -> b7
                    .stringSchema(b8 -> b8
                        .comment("tenantName
value")
                        .defaultValue("root")
                        .required(false))))
            .defaultCacheBehavior(b2 -> b2
                .viewerProtocolPolicy(ViewerProtocolPolicy.ALLOW_ALL)
                .targetOriginId(originId)
                .cachePolicyId("658327ea-f89d-4fab-
a63d-7e88639e58f6") // CachingOptimized Policy
                .allowedMethods(b4 -> b4
                    .quantity(2)
                    .items(Method.HEAD, Method.GET)))
        ));

    final Distribution distribution = createDistResponse.distribution();
    try (CloudFrontWaiter cfWaiter =
        CloudFrontWaiter.builder().client(cloudFrontClient).build()) {
        ResponseOrException<GetDistributionResponse> responseOrException =
            cfWaiter
                .waitUntilDistributionDeployed(builder ->
                    builder.id(distribution.id()))
                .matched();
        responseOrException.response()
            .orElseThrow(() -> new RuntimeException("Distribution not
created"));
    }
    return distribution;

```

```
    }
}
```

Contoh berikut menunjukkan cara membuat penyewa distribusi yang terkait dengan template itu, termasuk memanfaatkan parameter yang kami nyatakan di atas. Perhatikan bahwa kita tidak perlu menambahkan info sertifikat di sini karena domain kita sudah dicakup oleh template induk.

```
import software.amazon.awssdk.services.cloudfront.CloudFrontClient;
import
    software.amazon.awssdk.services.cloudfront.model.CreateConnectionGroupResponse;
import
    software.amazon.awssdk.services.cloudfront.model.CreateDistributionTenantResponse;
import software.amazon.awssdk.services.cloudfront.model.DistributionTenant;
import
    software.amazon.awssdk.services.cloudfront.model.GetConnectionGroupResponse;
import software.amazon.awssdk.services.cloudfront.model.ValidationTokenHost;
import software.amazon.awssdk.services.route53.Route53Client;
import software.amazon.awssdk.services.route53.model.RRType;

import java.time.Instant;

public class CreateDistributionTenant {

    public static DistributionTenant
    createDistributionTenantNoCert(CloudFrontClient cloudFrontClient,
                                Route53Client
                                route53Client,
                                String
                                distributionId,
                                String
                                domain,
                                String
                                hostedZoneId) {
        CreateDistributionTenantResponse createResponse =
        cloudFrontClient.createDistributionTenant(builder -> builder
            .distributionId(distributionId)
            .domains(b1 -> b1
                .domain(domain))
            .parameters(b2 -> b2
                .name("tenantName"))
```

```

        .value("myTenant"))
        .enabled(false)
        .name("no-cert-tenant")
    );

    final DistributionTenant distributionTenant =
createResponse.distributionTenant();

    // Then update the Route53 hosted zone to point your domain at the
distribution tenant
    // We fetch the RoutingEndpoint to point to via the default connection
group that was created for your tenant
    final GetConnectionGroupResponse fetchedConnectionGroup =
cloudFrontClient.getConnectionGroup(builder -> builder
        .identifier(distributionTenant.connectionGroupId()));

    route53Client.changeResourceRecordSets(builder -> builder
        .hostedZoneId(hostedZoneId)
        .changeBatch(b1 -> b1
            .comment("ChangeBatch comment")
            .changes(b2 -> b2
                .resourceRecordSet(b3 -> b3
                    .name(domain)
                    .type("CNAME")
                    .ttl(300L)
                    .resourceRecords(b4 -> b4
                        .value(fetchedConnectionGroup.connectionGroup().routingEndpoint()))
                    .action("CREATE"))
            ));
    return distributionTenant;
}
}

```

Jika sertifikat penampil dihilangkan dari templat induk, Anda perlu menambahkan info sertifikat pada penyewa yang terkait dengannya. Contoh berikut menunjukkan bagaimana melakukannya melalui sertifikat ACM arn yang mencakup domain yang diperlukan untuk penyewa.

```
import software.amazon.awssdk.services.cloudfront.CloudFrontClient;
```

```

import
    software.amazon.awssdk.services.cloudfront.model.CreateConnectionGroupResponse;
import
    software.amazon.awssdk.services.cloudfront.model.CreateDistributionTenantResponse;
import software.amazon.awssdk.services.cloudfront.model.DistributionTenant;
import
    software.amazon.awssdk.services.cloudfront.model.GetConnectionGroupResponse;
import software.amazon.awssdk.services.cloudfront.model.ValidationTokenHost;
import software.amazon.awssdk.services.route53.Route53Client;
import software.amazon.awssdk.services.route53.model.RRType;

import java.time.Instant;

public class CreateDistributionTenant {

    public static DistributionTenant
    createDistributionTenantWithCert(CloudFrontClient cloudFrontClient,

    Route53Client route53Client,                                String
    distributionId,                                             String
    domain,                                                    String
    hostedZoneId,                                             String
    certificateArn) {
        CreateDistributionTenantResponse createResponse =
        cloudFrontClient.createDistributionTenant(builder -> builder
            .distributionId(distributionId)
            .domains(b1 -> b1
                .domain(domain))
            .enabled(false)
            .name("tenant-with-cert")
            .parameters(b2 -> b2
                .name("tenantName")
                .value("myTenant"))
            .customizations(b3 -> b3
                .certificate(b4 -> b4
                    .arn(certificateArn))) // NOTE: Cert must be in
            Us-East-1 and cover the domain provided in this request

        );
    }

```



```

        final DistributionTenant distributionTenant =
            createResponse.distributionTenant();

        // Then update the Route53 hosted zone to point your domain at the
        // distribution tenant
        // We fetch the RoutingEndpoint to point to via the default connection
        // group that was created for your tenant
        final GetConnectionGroupResponse fetchedConnectionGroup =
            cloudFrontClient.getConnectionGroup(builder -> builder
                .identifier(distributionTenant.connectionGroupId()));

        route53Client.changeResourceRecordSets(builder -> builder
            .hostedZoneId(hostedZoneId)
            .changeBatch(b1 -> b1
                .comment("ChangeBatch comment")
                .changes(b2 -> b2
                    .resourceRecordSet(b3 -> b3
                        .name(domain)
                        .type("CNAME")
                        .ttl(300L)
                        .resourceRecords(b4 -> b4
                            .value(fetchedConnectionGroup.connectionGroup().routingEndpoint()))
                            .action("CREATE"))
                    ));
        return distributionTenant;
    }
}

```

Contoh berikut menunjukkan cara melakukannya dengan permintaan sertifikat terkelola CloudFront -host. Ini sangat ideal jika Anda belum memiliki lalu lintas ke domain Anda. Dalam hal ini, kami membuat ConnectionGroup untuk menghasilkan a RoutingEndpoint. Kemudian kami menggunakannya RoutingEndpoint untuk membuat catatan DNS yang memverifikasi kepemilikan domain dan menunjuk ke CloudFront. CloudFront kemudian akan secara otomatis menyajikan token untuk memvalidasi kepemilikan domain dan membuat sertifikat terkelola.

```

import software.amazon.awssdk.services.cloudfront.CloudFrontClient;
import
    software.amazon.awssdk.services.cloudfront.model.CreateConnectionGroupResponse;

```

```
import
    software.amazon.awssdk.services.cloudfront.model.CreateDistributionTenantResponse;
import software.amazon.awssdk.services.cloudfront.model.DistributionTenant;
import
    software.amazon.awssdk.services.cloudfront.model.GetConnectionGroupResponse;
import software.amazon.awssdk.services.cloudfront.model.ValidationTokenHost;
import software.amazon.awssdk.services.route53.Route53Client;
import software.amazon.awssdk.services.route53.model.RRType;

import java.time.Instant;

public class CreateDistributionTenant {

    public static DistributionTenant
    createDistributionTenantCfHosted(CloudFrontClient cloudFrontClient,

    Route53Client route53Client,                                     String
    distributionId,                                                String
    domain,                                                         String
    hostedZoneId) throws InterruptedException {
        CreateConnectionGroupResponse createConnectionGroupResponse =
        cloudFrontClient.createConnectionGroup(builder -> builder
            .ipv6Enabled(true)
            .name("cf-hosted-connection-group")
            .enabled(true));

        route53Client.changeResourceRecordSets(builder -> builder
            .hostedZoneId(hostedZoneId)
            .changeBatch(b1 -> b1
                .comment("cf-hosted domain validation record")
                .changes(b2 -> b2
                    .resourceRecordSet(b3 -> b3
                        .name(domain)
                        .type(RRType.CNAME)
                        .ttl(300L)
                        .resourceRecords(b4 -> b4
                            .value(createConnectionGroupResponse.connectionGroup().routingEndpoint()))
                        .action("CREATE"))
                    ));
    });
```

```

        // Give the R53 record time to propagate, if it isn't being returned by
        servers yet, the following call will fail
        Thread.sleep(60000);

        CreateDistributionTenantResponse createResponse =
cloudFrontClient.createDistributionTenant(builder -> builder
            .distributionId(distributionId)
            .domains(b1 -> b1
                .domain(domain))

            .connectionGroupId(createConnectionGroupResponse.connectionGroup().id())
            .enabled(false)
            .name("cf-hosted-tenant")
            .parameters(b2 -> b2
                .name("tenantName")
                .value("myTenant"))
            .managedCertificateRequest(b3 -> b3
                .validationTokenHost(ValidationTokenHost.CLOUDFRONT)
            )
        );

        return createResponse.distributionTenant();
    }
}

```

Contoh berikut menunjukkan cara melakukannya dengan permintaan sertifikat terkelola yang dihosting sendiri. Ini sangat ideal jika Anda memiliki lalu lintas ke domain Anda dan tidak dapat mentolerir downtime selama migrasi. Pada akhir contoh ini, Penyewa akan dibuat dalam keadaan menunggu validasi domain dan penyiapan DNS. Ikuti langkah-langkah [di sini] (<https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/managed-cloudfront-certificates.html#complete-domain-ownership>) untuk menyelesaikan penyiapan saat Anda siap untuk memigrasikan lalu lintas.

```

import software.amazon.awssdk.services.cloudfront.CloudFrontClient;
import
    software.amazon.awssdk.services.cloudfront.model.CreateConnectionGroupResponse;
import
    software.amazon.awssdk.services.cloudfront.model.CreateDistributionTenantResponse;
import software.amazon.awssdk.services.cloudfront.model.DistributionTenant;

```

```

import
    software.amazon.awssdk.services.cloudfront.model.GetConnectionGroupResponse;
import software.amazon.awssdk.services.cloudfront.model.ValidationTokenHost;
import software.amazon.awssdk.services.route53.Route53Client;
import software.amazon.awssdk.services.route53.model.RRType;

import java.time.Instant;

public class CreateDistributionTenant {

    public static DistributionTenant
    createDistributionTenantSelfHosted(CloudFrontClient cloudFrontClient,
                                     String
    distributionId,
                                     String
    domain) {
        CreateDistributionTenantResponse createResponse =
        cloudFrontClient.createDistributionTenant(builder -> builder
            .distributionId(distributionId)
            .domains(b1 -> b1
                .domain(domain))
            .parameters(b2 -> b2
                .name("tenantName")
                .value("myTenant"))
            .enabled(false)
            .name("self-hosted-tenant")
            .managedCertificateRequest(b3 -> b3
                .validationTokenHost(ValidationTokenHost.SELF_HOSTED)
                .primaryDomainName(domain)
            )
        );

        return createResponse.distributionTenant();
    }
}

```

- Untuk detail API, lihat topik berikut di Referensi API AWS SDK for Java 2.x .
 - [CreateDistribution](#)
 - [CreateDistributionTenant](#)

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Hapus sumber CloudFront penandatanganan menggunakan AWS SDK

Contoh kode berikut menunjukkan cara menghapus sumber daya yang digunakan untuk mendapatkan akses ke konten terbatas di bucket Amazon Simple Storage Service (Amazon S3).

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkap dan pelajari cara menyiapkan dan menjalankan di [Repositori Contoh Kode AWS](#).

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.services.cloudfront.CloudFrontClient;
import software.amazon.awssdk.services.cloudfront.model.DeleteKeyGroupResponse;
import
    software.amazon.awssdk.services.cloudfront.model.DeleteOriginAccessControlResponse;
import software.amazon.awssdk.services.cloudfront.model.DeletePublicKeyResponse;
import software.amazon.awssdk.services.cloudfront.model.GetKeyGroupResponse;
import
    software.amazon.awssdk.services.cloudfront.model.GetOriginAccessControlResponse;
import software.amazon.awssdk.services.cloudfront.model.GetPublicKeyResponse;

public class DeleteSigningResources {
    private static final Logger logger =
        LoggerFactory.getLogger(DeleteSigningResources.class);

    public static void deleteOriginAccessControl(final CloudFrontClient
        cloudFrontClient,
        final String originAccessControlId) {
        GetOriginAccessControlResponse getResponse = cloudFrontClient
            .getOriginAccessControl(b -> b.id(originAccessControlId));
        DeleteOriginAccessControlResponse deleteResponse =
            cloudFrontClient.deleteOriginAccessControl(builder -> builder
```

```

        .id(originAccessControlId)
        .ifMatch(getResponse.eTag()));
    if (deleteResponse.sdkHttpResponse().isSuccessful()) {
        logger.info("Successfully deleted Origin Access Control [{}]",
originAccessControlId);
    }
}

    public static void deleteKeyGroup(final CloudFrontClient cloudFrontClient,
final String keyGroupId) {

        GetKeyGroupResponse getResponse = cloudFrontClient.getKeyGroup(b ->
b.id(keyGroupId));
        DeleteKeyGroupResponse deleteResponse =
cloudFrontClient.deleteKeyGroup(builder -> builder
            .id(keyGroupId)
            .ifMatch(getResponse.eTag()));
        if (deleteResponse.sdkHttpResponse().isSuccessful()) {
            logger.info("Successfully deleted Key Group [{}]", keyGroupId);
        }
    }

    public static void deletePublicKey(final CloudFrontClient cloudFrontClient,
final String publicKeyId) {
        GetPublicKeyResponse getResponse = cloudFrontClient.getPublicKey(b ->
b.id(publicKeyId));

        DeletePublicKeyResponse deleteResponse =
cloudFrontClient.deletePublicKey(builder -> builder
            .id(publicKeyId)
            .ifMatch(getResponse.eTag()));

        if (deleteResponse.sdkHttpResponse().isSuccessful()) {
            logger.info("Successfully deleted Public Key [{}]", publicKeyId);
        }
    }
}

```

- Untuk detail API, lihat topik berikut di Referensi API AWS SDK for Java 2.x .
 - [DeleteKeyGroup](#)
 - [DeleteOriginAccessControl](#)

- [DeletePublicKey](#)

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Memulai dengan CloudFront distribusi dasar

Contoh kode berikut ini menunjukkan cara untuk melakukan:

- Buat bucket Amazon S3.
- Unggah konten ke bucket
- Buat CloudFront distribusi dengan OAC
- Pembersihan sumber daya

Bash

AWS CLI dengan skrip Bash

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankan di repositori [tutorial pengembang Sample](#).

```
#!/bin/bash

# CloudFront Getting Started Tutorial Script
# This script creates an S3 bucket, uploads sample content, creates a CloudFront
# distribution with OAC,
# and demonstrates how to access content through CloudFront.

set -euo pipefail

# Set up logging
LOG_FILE="cloudfront-tutorial.log"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "Starting CloudFront Getting Started Tutorial at $(date)"
```

```

# Function to handle errors
handle_error() {
    echo "ERROR: $1" >&2
    echo "Resources created before error:"
    if [ -n "${BUCKET_NAME:-}" ]; then
        echo "- S3 Bucket: $BUCKET_NAME"
    fi
    if [ -n "${OAC_ID:-}" ]; then
        echo "- CloudFront Origin Access Control: $OAC_ID"
    fi
    if [ -n "${DISTRIBUTION_ID:-}" ]; then
        echo "- CloudFront Distribution: $DISTRIBUTION_ID"
    fi

    echo "Attempting to clean up resources..."
    cleanup
    exit 1
}

# Function to clean up resources
cleanup() {
    echo "Cleaning up resources..."

    if [ -n "${DISTRIBUTION_ID:-}" ]; then
        echo "Disabling CloudFront distribution $DISTRIBUTION_ID..."

        # Get the current configuration and ETag in one call
        DIST_CONFIG=$(aws cloudfront get-distribution-config --id
"$DISTRIBUTION_ID" 2>/dev/null) || {
            echo "Failed to get distribution config. Continuing with cleanup..."
            DIST_CONFIG=""
        }

        if [ -n "$DIST_CONFIG" ]; then
            ETAG=$(echo "$DIST_CONFIG" | jq -r '.ETag')

            # Modify and update distribution in one pipeline
            if echo "$DIST_CONFIG" | jq '.DistributionConfig.Enabled = false' | \
                aws cloudfront update-distribution \
                    --id "$DISTRIBUTION_ID" \
                    --distribution-config "$(cat)" \
                    --if-match "$ETAG" 2>/dev/null; then

```



```

        echo "Waiting for distribution to be disabled (this may take
several minutes)..."
        aws cloudfront wait distribution-deployed --id "$DISTRIBUTION_ID"
2>/dev/null || {
            echo "Distribution deployment wait timed out. Proceeding with
deletion..."
        }

        # Get fresh ETag for deletion
        DIST_CONFIG=$(aws cloudfront get-distribution-config --id
"$DISTRIBUTION_ID" 2>/dev/null) || {
            echo "Failed to get updated config. Skipping distribution
deletion..."
            DIST_CONFIG=""
        }

        if [ -n "$DIST_CONFIG" ]; then
            ETAG=$(echo "$DIST_CONFIG" | jq -r '.ETag')
            aws cloudfront delete-distribution --id "$DISTRIBUTION_ID" --
if-match "$ETAG" 2>/dev/null && \
                echo "CloudFront distribution deleted." || \
                echo "Failed to delete distribution. You may need to
delete it manually."
        fi
        else
            echo "Failed to disable distribution. Continuing with cleanup..."
        fi
    fi
fi

if [ -n "${OAC_ID:-}" ]; then
    echo "Deleting Origin Access Control $OAC_ID..."
    OAC_DATA=$(aws cloudfront get-origin-access-control --id "$OAC_ID" 2>/
dev/null) || {
        echo "Failed to get Origin Access Control. You may need to delete it
manually."
        OAC_DATA=""
    }

    if [ -n "$OAC_DATA" ]; then
        OAC_ETAG=$(echo "$OAC_DATA" | jq -r '.ETag')
        aws cloudfront delete-origin-access-control --id "$OAC_ID" --if-match
"$OAC_ETAG" 2>/dev/null && \
            echo "Origin Access Control deleted." || \

```

```
        echo "Failed to delete Origin Access Control. You may need to
delete it manually."
    fi
fi

if [ -n "${BUCKET_NAME:-}" ] && [ "$BUCKET_IS_SHARED" != "true" ]; then
    echo "Deleting S3 bucket $BUCKET_NAME and its contents..."
    aws s3 rm "s3://$BUCKET_NAME" --recursive 2>/dev/null || {
        echo "Failed to remove bucket contents. Continuing with bucket
deletion..."
    }

    aws s3 rb "s3://$BUCKET_NAME" 2>/dev/null && \
        echo "S3 bucket deleted." || \
        echo "Failed to delete bucket. You may need to delete it manually."
fi

# Clean up temporary files
rm -f temp_disabled_config.json distribution-config.json bucket-policy.json
2>/dev/null || true
rm -rf temp_content 2>/dev/null || true
}

# Trap errors and cleanup
trap 'handle_error "Script interrupted"' INT TERM

# Validate AWS CLI is available
if ! command -v aws &> /dev/null; then
    handle_error "AWS CLI is not installed or not in PATH"
fi

# Validate jq is available
if ! command -v jq &> /dev/null; then
    handle_error "jq is not installed or not in PATH"
fi

# Validate AWS credentials are configured
if ! aws sts get-caller-identity &> /dev/null; then
    handle_error "AWS credentials are not configured or invalid"
fi

# Initialize variables
BUCKET_NAME=""
OAC_ID=""
```

```
DISTRIBUTION_ID=""
BUCKET_IS_SHARED=false

# Generate a random identifier for the bucket name using secure random
RANDOM_ID=$(openssl rand -hex 6)
if [ -z "$RANDOM_ID" ]; then
    handle_error "Failed to generate random identifier"
fi

# Check for shared prereq bucket and get account ID in parallel calls
PREREQ_BUCKET=$(aws cloudformation describe-stacks --stack-name tutorial-prereqs-
bucket \
    --query 'Stacks[0].Outputs[?OutputKey==`BucketName`].OutputValue' --output
text 2>/dev/null) || PREREQ_BUCKET=""

if [ -n "$PREREQ_BUCKET" ] && [ "$PREREQ_BUCKET" != "None" ]; then
    BUCKET_NAME="$PREREQ_BUCKET"
    BUCKET_IS_SHARED=true
    echo "Using shared bucket: $BUCKET_NAME"
else
    BUCKET_IS_SHARED=false
    BUCKET_NAME="cloudfront-${RANDOM_ID}"
fi
echo "Using bucket name: $BUCKET_NAME"

# Get AWS account ID early
ACCOUNT_ID=$(aws sts get-caller-identity --query 'Account' --output text)
if [ $? -ne 0 ]; then
    handle_error "Failed to get AWS account ID"
fi

# Validate account ID format
if ! [[ "$ACCOUNT_ID" =~ ^[0-9]{12}$ ]]; then
    handle_error "Invalid AWS account ID format: $ACCOUNT_ID"
fi

# Create a temporary directory for content with restrictive permissions
TEMP_DIR="temp_content"
mkdir -p "$TEMP_DIR/css"
chmod 700 "$TEMP_DIR"
if [ $? -ne 0 ]; then
    handle_error "Failed to create temporary directory"
fi
```

```

# Step 1: Create an S3 bucket (only if not shared)
if [ "$BUCKET_IS_SHARED" != "true" ]; then
    echo "Creating S3 bucket: $BUCKET_NAME"
    aws s3 mb "s3://$BUCKET_NAME" --region us-east-1
    if [ $? -ne 0 ]; then
        handle_error "Failed to create S3 bucket"
    fi

    aws s3api put-bucket-tagging --bucket "$BUCKET_NAME" --tagging
    'TagSet=[{Key=project,Value=doc-smith},{Key=tutorial,Value=cloudfront-
gettingstarted}]'

    # Batch bucket configuration calls for efficiency
    aws s3api put-bucket-versioning --bucket "$BUCKET_NAME" --versioning-
configuration Status=Enabled &
    aws s3api put-public-access-block \
        --bucket "$BUCKET_NAME" \
        --public-access-block-configuration
    "BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy=true,RestrictPublicBuckets
    &
    aws s3api put-bucket-encryption \
        --bucket "$BUCKET_NAME" \
        --server-side-encryption-configuration '{
            "Rules": [
                {
                    "ApplyServerSideEncryptionByDefault": {
                        "SSEAlgorithm": "AES256"
                    }
                }
            ]
        }' &
    wait
fi

# Step 2: Create sample content
echo "Creating sample content..."
cat > "$TEMP_DIR/index.html" << 'EOF'
<!DOCTYPE html>
<html>
<head>
    <title>Hello World</title>
    <link rel="stylesheet" type="text/css" href="css/styles.css">
</head>
<body>

```

```
<h1>Hello world!</h1>
</body>
</html>
EOF

cat > "$TEMP_DIR/css/styles.css" << 'EOF'
body {
    font-family: Arial, sans-serif;
    margin: 40px;
    background-color: #f5f5f5;
}
h1 {
    color: #333;
    text-align: center;
}
EOF

# Set restrictive permissions on content files
chmod 600 "$TEMP_DIR/index.html" "$TEMP_DIR/css/styles.css"

# Step 3: Upload content to the S3 bucket with encryption and metadata
echo "Uploading content to S3 bucket..."
aws s3 cp "$TEMP_DIR/" "s3://$BUCKET_NAME/" --recursive \
    --sse AES256 \
    --metadata "Source=CloudFrontTutorial"
if [ $? -ne 0 ]; then
    handle_error "Failed to upload content to S3 bucket"
fi

# Step 4: Create Origin Access Control
echo "Creating Origin Access Control..."
OAC_RESPONSE=$(aws cloudfront create-origin-access-control \
    --origin-access-control-config Name="oac-for-$BUCKET_NAME",SigningProtocol=sigv4,SigningBehavior=always,OriginAccessControlOriginType=

if [ $? -ne 0 ]; then
    handle_error "Failed to create Origin Access Control"
fi

OAC_ID=$(echo "$OAC_RESPONSE" | jq -r '.OriginAccessControl.Id')
if [ -z "$OAC_ID" ] || [ "$OAC_ID" = "null" ]; then
    handle_error "Failed to extract OAC ID from response"
fi
```

```
# Validate OAC ID format (alphanumeric and hyphens)
if ! [[ "$OAC_ID" =~ ^[A-Z0-9]+$ ]]; then
    handle_error "Invalid OAC ID format: $OAC_ID"
fi

echo "Created Origin Access Control with ID: $OAC_ID"

# Step 5: Create CloudFront distribution
echo "Creating CloudFront distribution..."

# Validate bucket name format
if ! [[ "$BUCKET_NAME" =~ ^[a-z0-9][a-z0-9.-]*[a-z0-9]$ ]]; then
    handle_error "Invalid S3 bucket name format: $BUCKET_NAME"
fi

# Create distribution configuration with improved security settings
cat > distribution-config.json << EOF
{
    "CallerReference": "cli-tutorial-$(date +%s)",
    "Origins": {
        "Quantity": 1,
        "Items": [
            {
                "Id": "S3-$BUCKET_NAME",
                "DomainName": "$BUCKET_NAME.s3.amazonaws.com",
                "S3OriginConfig": {
                    "OriginAccessIdentity": ""
                },
                "OriginAccessControlId": "$OAC_ID"
            }
        ]
    },
    "DefaultCacheBehavior": {
        "TargetOriginId": "S3-$BUCKET_NAME",
        "ViewerProtocolPolicy": "redirect-to-https",
        "AllowedMethods": {
            "Quantity": 2,
            "Items": ["GET", "HEAD"],
            "CachedMethods": {
                "Quantity": 2,
                "Items": ["GET", "HEAD"]
            }
        }
    },
    "DefaultTTL": 86400,
```

```
    "MinTTL": 0,
    "MaxTTL": 31536000,
    "Compress": true,
    "ForwardedValues": {
        "QueryString": false,
        "Cookies": {
            "Forward": "none"
        }
    }
},
"Comment": "CloudFront distribution for tutorial",
"Enabled": true,
"WebACLId": "",
"HttpVersion": "http2and3"
}
EOF

# Set restrictive permissions on config file before passing credentials
chmod 600 distribution-config.json

DIST_RESPONSE=$(aws cloudfront create-distribution --distribution-config file://
distribution-config.json)
if [ $? -ne 0 ]; then
    handle_error "Failed to create CloudFront distribution"
fi

DISTRIBUTION_ID=$(echo "$DIST_RESPONSE" | jq -r '.Distribution.Id')
DOMAIN_NAME=$(echo "$DIST_RESPONSE" | jq -r '.Distribution.DomainName')

if [ -z "$DISTRIBUTION_ID" ] || [ "$DISTRIBUTION_ID" = "null" ] || [ -z
"$DOMAIN_NAME" ] || [ "$DOMAIN_NAME" = "null" ]; then
    handle_error "Failed to extract distribution ID or domain name from response"
fi

# Validate distribution ID format
if ! [[ "$DISTRIBUTION_ID" =~ ^[A-Z0-9]+$ ]]; then
    handle_error "Invalid distribution ID format: $DISTRIBUTION_ID"
fi

echo "Created CloudFront distribution with ID: $DISTRIBUTION_ID"
echo "CloudFront domain name: $DOMAIN_NAME"

# Tag the CloudFront distribution
```

```

aws cloudfront tag-resource --resource "arn:aws:cloudfront::
$ACCOUNT_ID:distribution/$DISTRIBUTION_ID" --tags 'Items=[{Key=project,Value=doc-
smith},{Key=tutorial,Value=cloudfront-gettingstarted}]'

# Step 6: Update S3 bucket policy
echo "Updating S3 bucket policy..."

cat > bucket-policy.json << EOF
{
  "Version":"2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCloudFrontServicePrincipal",
      "Effect": "Allow",
      "Principal": {
        "Service": "cloudfront.amazonaws.com"
      },
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::$BUCKET_NAME/*",
      "Condition": {
        "StringEquals": {
          "AWS:SourceArn": "arn:aws:cloudfront::
$ACCOUNT_ID:distribution/$DISTRIBUTION_ID"
        }
      }
    }
  ]
}
EOF

# Set restrictive permissions on policy file
chmod 600 bucket-policy.json

aws s3api put-bucket-policy --bucket "$BUCKET_NAME" --policy file://bucket-
policy.json
if [ $? -ne 0 ]; then
  handle_error "Failed to update S3 bucket policy"
fi

# Step 7: Wait for distribution to deploy
echo "Waiting for CloudFront distribution to deploy (this may take 5-10
minutes)..."
aws cloudfront wait distribution-deployed --id "$DISTRIBUTION_ID" 2>/dev/null ||
{

```



```
    echo "Warning: Distribution deployment wait timed out. The distribution may
    still be deploying."
}

echo "CloudFront distribution is now deployed."

# Step 8: Display access information
echo ""
echo "==== CloudFront Distribution Setup Complete ====="
echo "You can access your content at: https://$DOMAIN_NAME/index.html"
echo ""
echo "Resources created:"
echo "- S3 Bucket: $BUCKET_NAME"
echo "- CloudFront Origin Access Control: $OAC_ID"
echo "- CloudFront Distribution: $DISTRIBUTION_ID"
echo ""
echo "To clean up resources, run: cleanup"
echo ""

echo "Tutorial completed at $(date)"
```

- Untuk detail API, lihat topik berikut di Referensi Perintah AWS CLI .
 - [CreateDistribution](#)
 - [CreateOriginAccessControl](#)
 - [DeleteDistribution](#)
 - [DeleteOriginAccessControl](#)
 - [GetDistribution](#)
 - [GetDistributionConfig](#)
 - [GetOriginAccessControl](#)
 - [UpdateDistribution](#)
 - [WaitDistributionDeployed](#)

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Memulai dengan WAF

Contoh kode berikut ini menunjukkan cara untuk melakukan:

- Buat ACL web
- Tambahkan aturan pencocokan string
- Tambahkan aturan terkelola
- Mengonfigurasi pencatatan log
- Pembersihan sumber daya

Bash

AWS CLI dengan skrip Bash

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankan di repositori [tutorial pengembang Sample](#).

```
#!/bin/bash

# AWS WAF Getting Started Script
# This script creates a Web ACL with a string match rule and AWS Managed Rules,
# associates it with a CloudFront distribution, and then cleans up all resources.

set -euo pipefail

# Security: Restrict file permissions
umask 077

# Set up logging with secure file handling
LOG_FILE="waf-tutorial.log"
touch "$LOG_FILE"
chmod 600 "$LOG_FILE"
exec > >(tee -a "$LOG_FILE") 2>&1

echo "====="
echo "AWS WAF Getting Started Tutorial"
echo "====="
```

```
echo "This script will create AWS WAF resources and associate"
echo "them with a CloudFront distribution."
echo ""

# Maximum number of retries for operations
MAX_RETRIES=3

# Function to handle errors securely
handle_error() {
    local error_msg="$1"
    echo "ERROR: $error_msg" >&2
    echo "Check the log file for details: $LOG_FILE" >&2
    cleanup_resources
    exit 1
}

# Function to validate AWS CLI JSON output
validate_json() {
    local json_string="$1"
    if ! echo "$json_string" | jq empty 2>/dev/null; then
        return 1
    fi
    return 0
}

# Function to safely extract JSON values
extract_json_value() {
    local json_string="$1"
    local key_path="$2"

    if ! validate_json "$json_string"; then
        return 1
    fi

    echo "$json_string" | jq -r "$key_path" 2>/dev/null || return 1
}

# Function to clean up resources securely
cleanup_resources() {
    echo ""
    echo "=====
echo "CLEANING UP RESOURCES"
echo "=====
```

```

if [ -n "${DISTRIBUTION_ID:-}" ] && [ -n "${WEB_ACL_ARN:-}" ]; then
    echo "Disassociating Web ACL from CloudFront distribution..."
    local account_id
    account_id=$(aws sts get-caller-identity --query Account --output text
2>/dev/null) || account_id=""

    if [ -z "$account_id" ]; then
        echo "Warning: Could not retrieve AWS account ID"
        return
    fi

    local disassociate_result
    disassociate_result=$(aws wafv2 disassociate-web-acl \
        --resource-arn "arn:aws:cloudfront::${account_id}:distribution/
${DISTRIBUTION_ID}" \
        --region us-east-1 2>&1) || true

    if echo "$disassociate_result" | grep -qi "error"; then
        echo "Warning: Failed to disassociate Web ACL: $disassociate_result"
    else
        echo "Web ACL disassociated successfully."
    fi
fi

if [ -n "${WEB_ACL_ID:-}" ] && [ -n "${WEB_ACL_NAME:-}" ]; then
    echo "Deleting Web ACL..."

    local get_result
    get_result=$(aws wafv2 get-web-acl \
        --name "${WEB_ACL_NAME}" \
        --scope CLOUDFRONT \
        --id "${WEB_ACL_ID}" \
        --region us-east-1 2>&1) || true

    if echo "$get_result" | grep -qi "error"; then
        echo "Warning: Failed to get Web ACL for deletion: $get_result"
        echo "You may need to manually delete the Web ACL using the AWS
Console."
    else
        local latest_token
        latest_token=$(extract_json_value "$get_result" '.WebACL.LockToken'
2>/dev/null) || latest_token=""

        if [ -n "$latest_token" ]; then

```

```

        local delete_result
        delete_result=$(aws wafv2 delete-web-acl \
            --name "$WEB_ACL_NAME" \
            --scope CLOUDFRONT \
            --id "$WEB_ACL_ID" \
            --lock-token "$latest_token" \
            --region us-east-1 2>&1) || true

        if echo "$delete_result" | grep -qi "error"; then
            echo "Warning: Failed to delete Web ACL: $delete_result"
            echo "You may need to manually delete the Web ACL using the
AWS Console."
        else
            echo "Web ACL deleted successfully."
        fi
    else
        echo "Warning: Could not extract lock token for deletion. You may
need to manually delete the Web ACL."
    fi
fi

echo "Cleanup process completed."
}

# Security: Trap EXIT to ensure cleanup on any exit
trap cleanup_resources EXIT

# Generate a random identifier for resource names using secure method
RANDOM_ID=$(openssl rand -hex 4) || handle_error "Failed to generate random ID"
WEB_ACL_NAME="MyWebACL-${RANDOM_ID}"
METRIC_NAME="MyWebACLMetrics-${RANDOM_ID}"

echo "Using Web ACL name: $WEB_ACL_NAME"

# Step 1: Create a Web ACL
echo ""
echo "=====
echo "STEP 1: Creating Web ACL"
echo "=====

local create_result
create_result=$(aws wafv2 create-web-acl \
    --name "$WEB_ACL_NAME" \

```

```

--scope "CLOUDFRONT" \
--default-action Allow={} \
--visibility-config
"SampledRequestsEnabled=true,CloudWatchMetricsEnabled=true,MetricName=
$METRIC_NAME" \
--tags Key=project,Value=doc-smith Key=tutorial,Value=aws-waf-gs \
--region us-east-1 2>&1) || handle_error "Failed to create Web ACL"

if ! validate_json "$create_result"; then
    handle_error "Invalid JSON response from create-web-acl"
fi

# Extract Web ACL ID, ARN, and Lock Token from the response
WEB_ACL_ID=$(extract_json_value "$create_result" '.Summary.Id') || handle_error
"Failed to extract Web ACL ID"
WEB_ACL_ARN=$(extract_json_value "$create_result" '.Summary.ARN') || handle_error
"Failed to extract Web ACL ARN"
LOCK_TOKEN=$(extract_json_value "$create_result" '.Summary.LockToken') ||
    handle_error "Failed to extract Lock Token"

echo "Web ACL created successfully with ID: $WEB_ACL_ID"
echo "Lock Token: [REDACTED]"

# Step 2: Add a String Match Rule
echo ""
echo "====="
echo "STEP 2: Adding String Match Rule"
echo "====="

for ((i=1; i<=MAX_RETRIES; i++)); do
    echo "Attempt $i to add string match rule..."

    # Get the latest lock token before updating
    local get_result
    get_result=$(aws wafv2 get-web-acl \
        --name "$WEB_ACL_NAME" \
        --scope CLOUDFRONT \
        --id "$WEB_ACL_ID" \
        --region us-east-1 2>&1) || true

    if echo "$get_result" | grep -qi "error"; then
        echo "Warning: Failed to get Web ACL for update: $get_result"
        if [ "$i" -eq "$MAX_RETRIES" ]; then
            handle_error "Failed to get Web ACL after $MAX_RETRIES attempts"
        fi
    fi
done

```

```
        fi
        sleep 2
        continue
    fi

    if ! validate_json "$get_result"; then
        echo "Warning: Invalid JSON response from get-web-acl"
        if [ "$i" -eq "$MAX_RETRIES" ]; then
            handle_error "Invalid JSON response after $MAX_RETRIES attempts"
        fi
        sleep 2
        continue
    fi

    local latest_token
    latest_token=$(extract_json_value "$get_result" '.WebACL.LockToken' 2>/dev/
null) || true

    if [ -z "$latest_token" ]; then
        echo "Warning: Could not extract lock token for update"
        if [ "$i" -eq "$MAX_RETRIES" ]; then
            handle_error "Failed to extract lock token after $MAX_RETRIES
attempts"
        fi
        sleep 2
        continue
    fi

    local update_result
    update_result=$(aws wafv2 update-web-acl \
        --name "$WEB_ACL_NAME" \
        --scope "CLOUDFRONT" \
        --id "$WEB_ACL_ID" \
        --lock-token "$latest_token" \
        --default-action Allow={} \
        --rules '[{
            "Name": "UserAgentRule",
            "Priority": 0,
            "Statement": {
                "ByteMatchStatement": {
                    "SearchString": "MyAgent",
                    "FieldToMatch": {
                        "SingleHeader": {
                            "Name": "user-agent"
```

```

        }
    },
    "TextTransformations": [
        {
            "Priority": 0,
            "Type": "NONE"
        }
    ],
    "PositionalConstraint": "EXACTLY"
}
},
"Action": {
    "Count": {}
},
"VisibilityConfig": {
    "SampledRequestsEnabled": true,
    "CloudWatchMetricsEnabled": true,
    "MetricName": "UserAgentRuleMetric"
}
}]' \
--visibility-config
"SampledRequestsEnabled=true,CloudWatchMetricsEnabled=true,MetricName=
$METRIC_NAME" \
--region us-east-1 2>&1) || true

if echo "$update_result" | grep -qi "WAFOptimisticLockException"; then
    echo "Optimistic lock exception encountered. Will retry with new lock
token."
    if [ "$i" -eq "$MAX_RETRIES" ]; then
        handle_error "Failed to add string match rule after $MAX_RETRIES
attempts"
    fi
    sleep 2
    continue
elif echo "$update_result" | grep -qi "error"; then
    handle_error "Failed to add string match rule: $update_result"
else
    echo "String match rule added successfully."
    break
fi
done

# Step 3: Add AWS Managed Rules
echo ""

```



```

echo "=====
echo "STEP 3: Adding AWS Managed Rules"
echo "=====

for ((i=1; i<=MAX_RETRIES; i++)); do
    echo "Attempt $i to add AWS Managed Rules..."

    # Get the latest lock token before updating
    local get_result
    get_result=$(aws wafv2 get-web-acl \
        --name "$WEB_ACL_NAME" \
        --scope CLOUDFRONT \
        --id "$WEB_ACL_ID" \
        --region us-east-1 2>&1) || true

    if echo "$get_result" | grep -qi "error"; then
        echo "Warning: Failed to get Web ACL for update: $get_result"
        if [ "$i" -eq "$MAX_RETRIES" ]; then
            handle_error "Failed to get Web ACL after $MAX_RETRIES attempts"
        fi
        sleep 2
        continue
    fi

    if ! validate_json "$get_result"; then
        echo "Warning: Invalid JSON response from get-web-acl"
        if [ "$i" -eq "$MAX_RETRIES" ]; then
            handle_error "Invalid JSON response after $MAX_RETRIES attempts"
        fi
        sleep 2
        continue
    fi

    local latest_token
    latest_token=$(extract_json_value "$get_result" '.WebACL.LockToken' 2>/dev/
null) || true

    if [ -z "$latest_token" ]; then
        echo "Warning: Could not extract lock token for update"
        if [ "$i" -eq "$MAX_RETRIES" ]; then
            handle_error "Failed to extract lock token after $MAX_RETRIES
attempts"
        fi
        sleep 2

```

```

        continue
    fi

    local update_result
    update_result=$(aws wafv2 update-web-acl \
        --name "$WEB_ACL_NAME" \
        --scope "CLOUDFRONT" \
        --id "$WEB_ACL_ID" \
        --lock-token "$latest_token" \
        --default-action Allow={} \
        --rules '[{
            "Name": "UserAgentRule",
            "Priority": 0,
            "Statement": {
                "ByteMatchStatement": {
                    "SearchString": "MyAgent",
                    "FieldToMatch": {
                        "SingleHeader": {
                            "Name": "user-agent"
                        }
                    },
                    "TextTransformations": [
                        {
                            "Priority": 0,
                            "Type": "NONE"
                        }
                    ],
                    "PositionalConstraint": "EXACTLY"
                }
            },
            "Action": {
                "Count": {}
            },
            "VisibilityConfig": {
                "SampledRequestsEnabled": true,
                "CloudWatchMetricsEnabled": true,
                "MetricName": "UserAgentRuleMetric"
            }
        },
        {
            "Name": "AWS-AWSManagedRulesCommonRuleSet",
            "Priority": 1,
            "Statement": {
                "ManagedRuleGroupStatement": {

```

```

        "VendorName": "AWS",
        "Name": "AWSManagedRulesCommonRuleSet",
        "ExcludedRules": []
    },
    "OverrideAction": {
        "Count": {}
    },
    "VisibilityConfig": {
        "SampledRequestsEnabled": true,
        "CloudWatchMetricsEnabled": true,
        "MetricName": "AWS-AWSManagedRulesCommonRuleSet"
    }
}]' \
--visibility-config
"SampledRequestsEnabled=true,CloudWatchMetricsEnabled=true,MetricName=
$METRIC_NAME" \
--region us-east-1 2>&1) || true

if echo "$update_result" | grep -qi "WAFOptimisticLockException"; then
    echo "Optimistic lock exception encountered. Will retry with new lock
token."
    if [ "$i" -eq "$MAX_RETRIES" ]; then
        handle_error "Failed to add AWS Managed Rules after $MAX_RETRIES
attempts"
    fi
    sleep 2
    continue
elif echo "$update_result" | grep -qi "error"; then
    handle_error "Failed to add AWS Managed Rules: $update_result"
else
    echo "AWS Managed Rules added successfully."
    break
fi
done

# Step 4: List CloudFront distributions
echo ""
echo "=====
echo "STEP 4: Listing CloudFront Distributions"
echo "=====

local cf_result

```

```

cf_result=$(aws cloudfront list-distributions --query "DistributionList.Items[*].
{Id:Id,DomainName:DomainName}" --output table 2>&1) || cf_result=""

if echo "$cf_result" | grep -qi "error"; then
    echo "Warning: Failed to list CloudFront distributions: $cf_result"
    echo "Continuing without CloudFront association."
    DISTRIBUTION_ID=""
else
    echo "$cf_result"

    # Auto-select first CloudFront distribution if available
    echo ""
    echo "=====
    echo "STEP 5: Associate Web ACL with CloudFront Distribution"
    echo "=====

    local first_dist
    first_dist=$(aws cloudfront list-distributions --query
"DistributionList.Items[0].Id" --output text 2>&1) || first_dist=""

    if [ -n "$first_dist" ] && [ "$first_dist" != "None" ] && ! echo
"$first_dist" | grep -qi "error"; then
        DISTRIBUTION_ID="$first_dist"
        echo "Using CloudFront distribution: $DISTRIBUTION_ID"

        local account_id
        account_id=$(aws sts get-caller-identity --query Account --output text
2>/dev/null) || account_id=""

        if [ -z "$account_id" ]; then
            echo "Warning: Could not retrieve AWS account ID for association"
            DISTRIBUTION_ID=""
        else
            local associate_result
            associate_result=$(aws wafv2 associate-web-acl \
                --web-acl-arn "$WEB_ACL_ARN" \
                --resource-arn "arn:aws:cloudfront::${account_id}:distribution/
${DISTRIBUTION_ID}" \
                --region us-east-1 2>&1) || true

            if echo "$associate_result" | grep -qi "error"; then
                echo "Warning: Failed to associate Web ACL with CloudFront
distribution: $associate_result"
                echo "Continuing without CloudFront association."
            fi
        fi
    fi

```

```

        DISTRIBUTION_ID=""
    else
        echo "Web ACL associated with CloudFront distribution
successfully."
    fi
fi
else
    echo "No CloudFront distributions available. Skipping association."
    DISTRIBUTION_ID=""
fi
fi

# Display summary of created resources
echo ""
echo "=====
echo "RESOURCE SUMMARY"
echo "=====
echo "Web ACL Name: $WEB_ACL_NAME"
echo "Web ACL ID: $WEB_ACL_ID"
echo "Web ACL ARN: $WEB_ACL_ARN"
if [ -n "${DISTRIBUTION_ID:-}" ]; then
    echo "Associated CloudFront Distribution: $DISTRIBUTION_ID"
fi
echo ""

# Auto-confirm cleanup
echo "=====
echo "CLEANUP CONFIRMATION"
echo "=====
echo "Proceeding with automatic cleanup of all created resources..."

echo ""
echo "=====
echo "Tutorial completed!"
echo "=====
echo "Log file: $LOG_FILE"

```

- Untuk detail API, lihat topik berikut di Referensi Perintah AWS CLI .
 - [AssociateWebAcl](#)
 - [CreateWebAcl](#)
 - [DeleteWebAcl](#)
 - [DisassociateWebAcl](#)

- [GetCallerIdentity](#)
- [GetWebAcl](#)
- [ListDistributions](#)
- [UpdateWebAcl](#)

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Buat cookie yang ditandatangani URLs dan menggunakan AWS SDK

Contoh kode berikut menunjukkan cara membuat ditandatangani URLs dan cookie yang memungkinkan akses ke sumber daya terbatas.

Java

SDK untuk Java 2.x

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankannya di [Repositori Contoh Kode AWS](#).

Gunakan [CannedSignerRequest](#) kelas untuk menandatangani URLs atau cookie dengan kebijakan kalengan.

```
import software.amazon.awssdk.services.cloudfront.model.CannedSignerRequest;

import java.net.URL;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.time.Instant;
import java.time.temporal.ChronoUnit;

public class CreateCannedPolicyRequest {

    public static CannedSignerRequest createRequestForCannedPolicy(String
distributionDomainName,
```

```

        String fileNameToUpload,
        String privateKeyFullPath, String publicKeyId) throws Exception {
    String protocol = "https";
    String resourcePath = "/" + fileNameToUpload;

    String cloudFrontUrl = new URL(protocol, distributionDomainName,
resourcePath).toString();
    Instant expirationDate = Instant.now().plus(7, ChronoUnit.DAYS);
    Path path = Paths.get(privateKeyFullPath);

    return CannedSignerRequest.builder()
        .resourceUrl(cloudFrontUrl)
        .privateKey(path)
        .keyPairId(publicKeyId)
        .expirationDate(expirationDate)
        .build();
    }
}

```

Gunakan [CustomSignerRequest](#) kelas untuk menandatangani URLs atau cookie dengan kebijakan khusus. Metode `activeDate` dan `ipRange` merupakan metode opsional.

```

import software.amazon.awssdk.services.cloudfront.model.CustomSignerRequest;

import java.net.URL;
import java.nio.file.Path;
import java.nio.file.Paths;
import java.time.Instant;
import java.time.temporal.ChronoUnit;

public class CreateCustomPolicyRequest {

    public static CustomSignerRequest createRequestForCustomPolicy(String
distributionDomainName,
        String fileNameToUpload,
        String privateKeyFullPath, String publicKeyId) throws Exception {
    String protocol = "https";
    String resourcePath = "/" + fileNameToUpload;

    String cloudFrontUrl = new URL(protocol, distributionDomainName,
resourcePath).toString();
    Instant expireDate = Instant.now().plus(7, ChronoUnit.DAYS);

```

```
// URL will be accessible tomorrow using the signed URL.
Instant activeDate = Instant.now().plus(1, ChronoUnit.DAYS);
Path path = Paths.get(privateKeyFullPath);

return CustomSignerRequest.builder()
    .resourceUrl(cloudFrontUrl)
    // .resourceUrlPattern("https://*.example.com/*") // Optional.
    .privateKey(path)
    .keyPairId(publicKeyId)
    .expirationDate(expireDate)
    .activeDate(activeDate) // Optional.
    // .ipRange("192.168.0.1/24") // Optional.
    .build();
}
}
```

Contoh berikut menunjukkan penggunaan [CloudFrontUtilities](#) kelas untuk menghasilkan cookie ditandatangani dan URLs. [Lihat](#) contoh kode ini di GitHub.

```
import org.slf4j.Logger;
import org.slf4j.LoggerFactory;
import software.amazon.awssdk.services.cloudfront.CloudFrontUtilities;
import software.amazon.awssdk.services.cloudfront.cookie.CookiesForCannedPolicy;
import software.amazon.awssdk.services.cloudfront.cookie.CookiesForCustomPolicy;
import software.amazon.awssdk.services.cloudfront.model.CannedSignerRequest;
import software.amazon.awssdk.services.cloudfront.model.CustomSignerRequest;
import software.amazon.awssdk.services.cloudfront.url.SignedUrl;

public class SigningUtilities {
    private static final Logger logger =
        LoggerFactory.getLogger(SigningUtilities.class);
    private static final CloudFrontUtilities cloudFrontUtilities =
        CloudFrontUtilities.create();

    public static SignedUrl signUrlForCannedPolicy(CannedSignerRequest
        cannedSignerRequest) {
        SignedUrl signedUrl =
            cloudFrontUtilities.getSignedUrlWithCannedPolicy(cannedSignerRequest);
        logger.info("Signed URL: [{}]", signedUrl.url());
        return signedUrl;
    }
}
```



```

    public static SignedUrl signUrlForCustomPolicy(CustomSignerRequest
customSignerRequest) {
        SignedUrl signedUrl =
cloudFrontUtilities.getSignedUrlWithCustomPolicy(customSignerRequest);
        logger.info("Signed URL: [{}]", signedUrl.url());
        return signedUrl;
    }

    public static CookiesForCannedPolicy
getCookiesForCannedPolicy(CannedSignerRequest cannedSignerRequest) {
        CookiesForCannedPolicy cookiesForCannedPolicy = cloudFrontUtilities
            .getCookiesForCannedPolicy(cannedSignerRequest);
        logger.info("Cookie EXPIRES header [{}]",
cookiesForCannedPolicy.expiresHeaderValue());
        logger.info("Cookie KEYPAIR header [{}]",
cookiesForCannedPolicy.keyPairIdHeaderValue());
        logger.info("Cookie SIGNATURE header [{}]",
cookiesForCannedPolicy.signatureHeaderValue());
        return cookiesForCannedPolicy;
    }

    public static CookiesForCustomPolicy
getCookiesForCustomPolicy(CustomSignerRequest customSignerRequest) {
        CookiesForCustomPolicy cookiesForCustomPolicy = cloudFrontUtilities
            .getCookiesForCustomPolicy(customSignerRequest);
        logger.info("Cookie POLICY header [{}]",
cookiesForCustomPolicy.policyHeaderValue());
        logger.info("Cookie KEYPAIR header [{}]",
cookiesForCustomPolicy.keyPairIdHeaderValue());
        logger.info("Cookie SIGNATURE header [{}]",
cookiesForCustomPolicy.signatureHeaderValue());
        return cookiesForCustomPolicy;
    }
}

```

- Untuk detail API, lihat [CloudFrontUtilities](#) di Referensi AWS SDK for Java 2.x API.

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

CloudFront Contoh fungsi untuk CloudFront

Contoh kode berikut menunjukkan cara menggunakan CloudFront dengan AWS SDKs.

Contoh

- [Tambahkan header keamanan HTTP ke acara respons penampil CloudFront Fungsi](#)
- [Tambahkan header CORS ke acara respons penampil CloudFront Fungsi](#)
- [Tambahkan header kontrol cache ke acara respons penampil CloudFront Fungsi](#)
- [Tambahkan header IP klien sejati ke acara permintaan penampil CloudFront Fungsi](#)
- [Tambahkan header asal ke acara permintaan penampil CloudFront Fungsi](#)
- [Tambahkan index.html untuk meminta URLs tanpa nama file dalam acara permintaan penampil CloudFront Fungsi](#)
- [Menormalkan parameter string kueri dalam permintaan penampil CloudFront Fungsi](#)
- [Mengalihkan ke URL baru dalam acara permintaan penampil CloudFront Fungsi](#)
- [Menulis ulang URI permintaan berdasarkan KeyValueStore konfigurasi untuk acara permintaan penampil CloudFront Fungsi](#)
- [Rutekan permintaan ke asal yang lebih dekat dengan penampil dalam acara permintaan penampil CloudFront Fungsi](#)
- [Gunakan pasangan nilai kunci dalam permintaan penampil CloudFront Fungsi](#)
- [Validasi token sederhana dalam permintaan penampil CloudFront Fungsi](#)

Tambahkan header keamanan HTTP ke acara respons penampil CloudFront Fungsi

Contoh kode berikut menunjukkan cara menambahkan header keamanan HTTP ke acara respons penampil CloudFront Fungsi.

JavaScript

JavaScript runtime 2.0 untuk Fungsi CloudFront

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankan di repositori [contoh CloudFront Fungsi](#).

```
async function handler(event) {
    var response = event.response;
    var headers = response.headers;

    // Set HTTP security headers
    // Since JavaScript doesn't allow for hyphens in variable names, we use the
    dict["key"] notation
    headers['strict-transport-security'] = { value: 'max-age=63072000;
includeSubdomains; preload'};
    headers['content-security-policy'] = { value: "default-src 'none'; img-src
'self'; script-src 'self'; style-src 'self'; object-src 'none'; frame-ancestors
'none'"};
    headers['x-content-type-options'] = { value: 'nosniff'};
    headers['x-frame-options'] = {value: 'DENY'};
    headers['x-xss-protection'] = {value: '1; mode=block'};
    headers['referrer-policy'] = {value: 'same-origin'};

    // Return the response to viewers
    return response;
}
```

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Tambahkan header CORS ke acara respons penampil CloudFront Fungsi

Contoh kode berikut menunjukkan cara menambahkan header CORS ke acara respons penampil CloudFront Functions.

JavaScript

JavaScript runtime 2.0 untuk Fungsi CloudFront

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankan di repositori [contoh CloudFront Fungsi](#).

```
async function handler(event) {
    var request = event.request;
    var response = event.response;

    // If Access-Control-Allow-Origin CORS header is missing, add it.
    // Since JavaScript doesn't allow for hyphens in variable names, we use the
    dict["key"] notation.
    if (!response.headers['access-control-allow-origin'] &&
        request.headers['origin']) {
        response.headers['access-control-allow-origin'] = {value:
            request.headers['origin'].value};
        console.log("Access-Control-Allow-Origin was missing, adding it now.");
    }

    return response;
}
```

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Tambahkan header kontrol cache ke acara respons penampil CloudFront Fungsi

Contoh kode berikut menunjukkan cara menambahkan header kontrol cache ke acara respons penampil CloudFront Fungsi.

JavaScript

JavaScript runtime 2.0 untuk Fungsi CloudFront

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankan di repositori [contoh CloudFront Fungsi](#).

```
async function handler(event) {
  var response = event.response;
  var headers = response.headers;

  if (response.statusCode >= 200 && response.statusCode < 400) {
    // Set the cache-control header
    headers['cache-control'] = {value: 'public, max-age=63072000'};
  }

  // Return response to viewers
  return response;
}
```

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Tambahkan header IP klien sejati ke acara permintaan penampil CloudFront Fungsi

Contoh kode berikut menunjukkan cara menambahkan header IP klien sejati ke acara permintaan penampil CloudFront Fungsi.

JavaScript

JavaScript runtime 2.0 untuk Fungsi CloudFront

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankan di repositori [contoh CloudFront Fungsi](#).

```
async function handler(event) {  
    var request = event.request;  
    var clientIP = event.viewer.ip;  
  
    //Add the true-client-ip header to the incoming request  
    request.headers['true-client-ip'] = {value: clientIP};  
  
    return request;  
}
```

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Tambahkan header asal ke acara permintaan penampil CloudFront Fungsi

Contoh kode berikut menunjukkan cara menambahkan header asal ke acara permintaan penampil CloudFront Fungsi.

JavaScript

JavaScript runtime 2.0 untuk Fungsi CloudFront

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankan di repositori [contoh CloudFront Fungsi](#).

```
async function handler(event) {
  var request = event.request;
  var headers = request.headers;
  var host = request.headers.host.value;

  // If origin header is missing, set it equal to the host header.
  if (!headers.origin)
    headers.origin = {value: `https://${host}`};

  return request;
}
```

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Tambahkan index.html untuk meminta URLs tanpa nama file dalam acara permintaan penampil CloudFront Fungsi

Contoh kode berikut menunjukkan cara menambahkan index.html untuk meminta URLs tanpa nama file dalam acara permintaan penampil CloudFront Fungsi.

JavaScript

JavaScript runtime 2.0 untuk Fungsi CloudFront

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankan di repositori [contoh CloudFront Fungsi](#).

```
async function handler(event) {
  var request = event.request;
  var uri = request.uri;

  // Check whether the URI is missing a file name.
  if (uri.endsWith('/')) {
```

```
        request.uri += 'index.html';
    }
    // Check whether the URI is missing a file extension.
    else if (!uri.includes('.')) {
        request.uri += '/index.html';
    }

    return request;
}
```

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Menormalkan parameter string kueri dalam permintaan penampil CloudFront Fungsi

Contoh kode berikut menunjukkan cara menormalkan parameter string kueri dalam permintaan penampil CloudFront Fungsi.

JavaScript

JavaScript runtime 2.0 untuk Fungsi CloudFront

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankan di repositori [contoh CloudFront Fungsi](#).

```
function handler(event) {
    var qs=[];
    for (var key in event.request.querystring) {
        if (event.request.querystring[key].multiValue) {
            event.request.querystring[key].multiValue.forEach((mv) =>
            {qs.push(key + "=" + mv.value)});
        } else {
            qs.push(key + "=" + event.request.querystring[key].value);
        }
    }
}
```



```
};

event.request.querystring = qs.sort().join('&');

return event.request;
}
```

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Mengalihkan ke URL baru dalam acara permintaan penampil CloudFront Fungsi

Contoh kode berikut menunjukkan cara mengalihkan ke URL baru dalam acara permintaan penampil CloudFront Fungsi.

JavaScript

JavaScript runtime 2.0 untuk Fungsi CloudFront

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankan di repositori [contoh CloudFront Fungsi](#).

```
async function handler(event) {
    var request = event.request;
    var headers = request.headers;
    var host = request.headers.host.value;
    var country = 'DE' // Choose a country code
    var newurl = `https://${host}/de/index.html`; // Change the redirect URL to
    your choice

    if (headers['cloudfront-viewer-country']) {
        var countryCode = headers['cloudfront-viewer-country'].value;
        if (countryCode === country) {
```

```
        var response = {
            statusCode: 302,
            statusDescription: 'Found',
            headers:
                { "location": { "value": newurl } }
        }

        return response;
    }
}
return request;
}
```

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Menulis ulang URI permintaan berdasarkan KeyValueStore konfigurasi untuk acara permintaan penampil CloudFront Fungsi

Contoh kode berikut menunjukkan cara menulis ulang URI permintaan berdasarkan KeyValueStore konfigurasi untuk acara permintaan penampil CloudFront Fungsi.

JavaScript

JavaScript runtime 2.0 untuk Fungsi CloudFront

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankan di repositori [contoh CloudFront Fungsi](#).

```
import cf from 'cloudfront';

// (Optional) Replace KVS_ID with actual KVS ID
const kvsId = "KVS_ID";
// enable stickiness by setting a cookie from origin or using another edge
function
```

```
const stickinessCookieName = "appversion";
// set to true to enable console logging
const loggingEnabled = false;

// function rewrites the request uri based on configuration in KVS
// example config in KVS in key:value format
// "latest": {"a_weightage": .8, "a_url": "v1", "b_url": "v2"}
// given above key and value in KVS the request uri will be rewritten
// for example http(s)://domain/latest/something/else will be rewritten as
// http(s)://domain/v1/something/else or http(s)://domain/v2/something/else
// depending on weightage
// if no configuration is found, then the request is returned as is
async function handler(event) {
  // NOTE: This example function is for a viewer request event trigger.
  // Choose viewer request for event trigger when you associate this function
  // with a distribution.
  const request = event.request;
  const pathSegments = request.uri.split('/');
  const key = pathSegments[1];

  // if empty path segment or if there is valid stickiness cookie
  // then skip call to KVS and let the request continue.
  if (!key || isValidStickinessCookie(request.cookies[stickinessCookieName],
    key)) {
    return event.request;
  }

  try {
    // get the prefix replacement from KVS
    const replacement = await getPathPrefixByWeightage(key);
    if (!replacement) {
      return event.request;
    }
    // Replace the first path with the replacement
    pathSegments[1] = replacement;
    log(`using prefix ${pathSegments[1]}`);
    const newUri = pathSegments.join('/');
    log(`${request.uri} -> ${newUri}`);
    request.uri = newUri;

    return request;
  } catch (err) {
    // No change to the path if the key is not found or any other error
    log(`request uri: ${request.uri}, error: ${err}`);
  }
}
```

```
    }
    // no change to path - return request
    return event.request;
}

// function to get the prefix from KVS
async function getPathPrefixByWeightage(key) {
    const kvsHandle = cf.kvs(kvsId);
    // get the weightage config from KVS
    const kvsResponse = await kvsHandle.get(key);
    const weightageConfig = JSON.parse(kvsResponse);
    // no configuration - return null
    if (!weightageConfig || !isFinite(weightageConfig.a_weightage)) {
        return null;
    }
    // return the url based on weightage
    // return null if no url is configured
    if (Math.random() <= weightageConfig.a_weightage) {
        return weightageConfig.a_url ? weightageConfig.a_url : null;
    } else {
        return weightageConfig.b_url ? weightageConfig.b_url : null;
    }
}

// function to check if the stickiness cookie is valid
function hasValidStickinessCookie(stickinessCookie, pathSegment) {
    // if the value exists and it matches pathSegment
    return (stickinessCookie && stickinessCookie.value === pathSegment)
}

function log(message) {
    if (loggingEnabled) {
        console.log(message);
    }
}
}
```

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Rutekan permintaan ke asal yang lebih dekat dengan penampil dalam acara permintaan penampil CloudFront Fungsi

Contoh kode berikut menunjukkan cara merutekan permintaan ke asal yang lebih dekat dengan penampil dalam acara permintaan penampil CloudFront Fungsi.

JavaScript

JavaScript runtime 2.0 untuk Fungsi CloudFront

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankan di repositori [contoh CloudFront Fungsi](#).

```
import cf from 'cloudfront';

function handler(event) {
  const request = event.request;
  const headers = request.headers;
  const country = headers['cloudfront-viewer-country'] &&
    headers['cloudfront-viewer-country'].value;

  //List of Regions with S3 buckets containing content
  const countryToRegion = {
    'DE': 'eu-central-1',
    'IE': 'eu-west-1',
    'GB': 'eu-west-2',
    'FR': 'eu-west-3',
    'JP': 'ap-northeast-1',
    'IN': 'ap-south-1'
  };

  const DEFAULT_REGION = 'us-east-1';

  const selectedRegion = (country && countryToRegion[country]) ||
    DEFAULT_REGION;

  const domainName =
```

```
`cloudfront-functions-demo-bucket-in-${selectedRegion}.s3.
${selectedRegion}.amazonaws.com`;

cf.updateRequestOrigin({
  "domainName": domainName,
  "originAccessControlConfig": {
    "enabled": true,
    "region": selectedRegion,
    "signingBehavior": "always",
    "signingProtocol": "sigv4",
    "originType": "s3"
  },
});

return request;
}
```

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Gunakan pasangan nilai kunci dalam permintaan penampil CloudFront Fungsi

Contoh kode berikut menunjukkan cara menggunakan pasangan kunci-nilai dalam permintaan penampil CloudFront Fungsi.

JavaScript

JavaScript runtime 2.0 untuk Fungsi CloudFront

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankan di repositori [contoh CloudFront Fungsi](#).

```
import cf from 'cloudfront';

// This fails if there is no key value store associated with the function
```

```
const kvsHandle = cf.kvs();

// Remember to associate the KVS with your function before referencing KVS in
// your code.
// https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/kvs-with-
// functions-associate.html
async function handler(event) {
  const request = event.request;
  // Use the first segment of the pathname as key
  // For example http(s)://domain/<key>/something/else
  const pathSegments = request.uri.split('/')
  const key = pathSegments[1]
  try {
    // Replace the first path of the pathname with the value of the key
    // For example http(s)://domain/<value>/something/else
    pathSegments[1] = await kvsHandle.get(key);
    const newUri = pathSegments.join('/');
    console.log(`${request.uri} -> ${newUri}`)
    request.uri = newUri;
  } catch (err) {
    // No change to the pathname if the key is not found
    console.log(`${request.uri} | ${err}`);
  }
  return request;
}
```

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Validasi token sederhana dalam permintaan penampil CloudFront Fungsi

Contoh kode berikut menunjukkan cara memvalidasi token sederhana dalam permintaan penampil CloudFront Fungsi.

JavaScript

JavaScript runtime 2.0 untuk Fungsi CloudFront

Note

Ada lebih banyak tentang GitHub. Temukan contoh lengkapnya dan pelajari cara mengatur dan menjalankan di repositori [contoh CloudFront Fungsi](#).

```
import crypto from 'crypto';
import cf from 'cloudfront';

//Response when JWT is not valid.
const response401 = {
  statusCode: 401,
  statusDescription: 'Unauthorized'
};

// Remember to associate the KVS with your function before calling the const
kvsKey = 'jwt.secret'.
// https://docs.aws.amazon.com/AmazonCloudFront/latest/DeveloperGuide/kvs-with-
functions-associate.html
const kvsKey = 'jwt.secret';
// set to true to enable console logging
const loggingEnabled = false;

function jwt_decode(token, key, noVerify, algorithm) {
  // check token
  if (!token) {
    throw new Error('No token supplied');
  }
  // check segments
  const segments = token.split('.');
  if (segments.length !== 3) {
    throw new Error('Not enough or too many segments');
  }

  // All segment should be base64
  const headerSeg = segments[0];
  const payloadSeg = segments[1];
```



```
const signatureSeg = segments[2];

// base64 decode and parse JSON
const payload = JSON.parse(_base64urlDecode(payloadSeg));

if (!noVerify) {
  const signingMethod = 'sha256';
  const signingType = 'hmac';

  // Verify signature. `sign` will return base64 string.
  const signingInput = [headerSeg, payloadSeg].join('.');

  if (!_verify(signingInput, key, signingMethod, signingType,
signatureSeg)) {
    throw new Error('Signature verification failed');
  }

  // Support for nbf and exp claims.
  // According to the RFC, they should be in seconds.
  if (payload.nbf && Date.now() < payload.nbf*1000) {
    throw new Error('Token not yet active');
  }

  if (payload.exp && Date.now() > payload.exp*1000) {
    throw new Error('Token expired');
  }
}

return payload;
}

//Function to ensure a constant time comparison to prevent
//timing side channels.
function _constantTimeEquals(a, b) {
  if (a.length !== b.length) {
    return false;
  }

  let xor = 0;
  for (let i = 0; i < a.length; i++) {
    xor |= (a.charCodeAt(i) ^ b.charCodeAt(i));
  }

  return 0 === xor;
}
```

```
}

function _verify(input, key, method, type, signature) {
  if(type === "hmac") {
    return _constantTimeEquals(signature, _sign(input, key, method));
  }
  else {
    throw new Error('Algorithm type not recognized');
  }
}

function _sign(input, key, method) {
  return crypto.createHmac(method, key).update(input).digest('base64url');
}

function _base64urlDecode(str) {
  return Buffer.from(str, 'base64url')
}

async function handler(event) {
  let request = event.request;

  //Secret key used to verify JWT token.
  //Update with your own key.
  const secret_key = await getSecret()

  if(!secret_key) {
    return response401;
  }

  // If no JWT token, then generate HTTP redirect 401 response.
  if(!request.querystring.jwt) {
    log("Error: No JWT in the querystring");
    return response401;
  }

  const jwtToken = request.querystring.jwt.value;

  try{
    jwt_decode(jwtToken, secret_key);
  }
  catch(e) {
    log(e);
    return response401;
  }
}
```

```
    }

    //Remove the JWT from the query string if valid and return.
    delete request.querystring.jwt;
    log("Valid JWT token");
    return request;
}

// get secret from key value store
async function getSecret() {
    // initialize cloudfront kv store and get the key value
    try {
        const kvsHandle = cf.kvs();
        return await kvsHandle.get(kvsKey);
    } catch (err) {
        log(`Error reading value for key: ${kvsKey}, error: ${err}`);
        return null;
    }
}

function log(message) {
    if (loggingEnabled) {
        console.log(message);
    }
}
```

Untuk daftar lengkap panduan pengembang AWS SDK dan contoh kode, lihat [Menggunakan CloudFront dengan AWS SDK](#). Topik ini juga mencakup informasi tentang memulai dan detail tentang versi SDK sebelumnya.

Riwayat dokumen

Tabel berikut menjelaskan perubahan penting yang dibuat pada CloudFront dokumentasi. Untuk pemberitahuan pembaruan, Anda dapat [berlangganan umpan RSS](#).

Perubahan	Deskripsi	Tanggal
Ditambahkan TLS timbal balik (penampil)	CloudFront mendukung TLS timbal balik (penampil).	November 24, 2025
Ditambahkan bidang log untuk log akses	Menambahkan connectio n-id bidang untuk log akses (log standar) dan log akses real-time.	November 24, 2025
Menambahkan log koneksi	Menambahkan log koneksi sebagai fitur logging baru untuk TLS bersama (penampil).	November 24, 2025
Menambahkan Fungsi Koneksi	CloudFront mendukung Fungsi Koneksi untuk TLS bersama (penampil).	November 24, 2025
Tambahkan dukungan untuk membawa IP Anda sendiri untuk CloudFront menggunakan IPAM	CloudFront mendukung membawa IPv4 alamat Anda sendiri melalui BYOIP IPAM untuk layanan global.	November 24, 2025
AWS pembaruan kebijakan terkelola	Kebijakan CloudFron tReadOnlyAccess dan CloudFrontFullAccess IAM sekarang mendukung ec2:DescribeIpamPools dan ec2:GetIpamPoolCidrs bertindak.	November 24, 2025

Pembaruan untuk CloudFront Fungsi	Rilis ini menambahkan parameter untuk metode pembantu modifikasi asal di CloudFront Functions. Anda dapat menggunakan <code>hostHeader</code> , <code>sni</code> , <code>allowedCertificateNames</code> , dan <code>originOverrides</code> parameter.	November 20, 2025
Pembaruan untuk CloudFront Fungsi	Menambahkan dukungan CBOR Web Tokens (CWT) untuk Fungsi. CloudFront	November 20, 2025
Pembaruan untuk CloudFront Fungsi	Menambahkan metode pembantu umum untuk CloudFront Fungsi.	November 20, 2025
Kebijakan AWS terkelola yang diperbarui	Diperbarui CloudFront <code>FullAccess</code> untuk mengizinkan izin membuat sumber daya ACL AWS WAF Web dan membuat, memperbarui, menghapus, dan membaca akses ke AWS Pricing Plan Manager.	November 18, 2025
Kebijakan AWS terkelola yang diperbarui	Diperbarui CloudFront <code>ReadOnlyAccess</code> untuk mengizinkan izin baca saja ke AWS Pricing Plan Manager.	November 18, 2025

Kebijakan AWS terkelola yang diperbarui	Diperbarui CloudFrontFullAccess untuk mengizinkan izin membuat sumber daya ACL AWS WAF Web dan membuat, memperbarui, menghapus, dan membaca akses ke AWS Pricing Plan Manager.	November 18, 2025
Kebijakan AWS terkelola yang diperbarui	Diperbarui CloudFrontReadOnlyAccess untuk mengizinkan izin baca saja ke AWS Pricing Plan Manager.	November 18, 2025
CloudFront mendukung rencana penetapan harga flat-rate	Anda sekarang dapat berlangganan distribusi Anda ke paket harga CloudFront flat-rate.	November 18, 2025
Anycast statis IPs	Anda sekarang dapat memilih antara hanya IPv4 alamat atau keduanya IPv4 dan IPv6 alamat (dualstack).	November 5, 2025

[Menambahkan dukungan untuk berbagi asal-usul VPC di seluruh Akun AWS](#)

Anda dapat membuat berbagi sumber daya dan menambahkan asal VPC ke dalamnya. Ini memungkinkan Anda menyimpan asal dan CloudFront distribusi VPC Anda secara terpisah Akun AWS, memungkinkan organisasi Anda untuk mempertahankan persyaratan multi-akun. Lainnya Akun AWS dapat mengaitkan asal VPC bersama untuk CloudFront distribusinya.

November 5, 2025

[Ditambahkan kebijakan keamanan penampil](#)

Menambahkan kebijakan TLSv1 keamanan.2_2025.

Oktober 10, 2025

- [Protokol dan cipher yang didukung antara pemirsa dan CloudFront](#)
- [Kebijakan keamanan \(SSL/ TLS versi minimum\)](#)

[Menambahkan algoritma pertukaran kunci pasca-kuantum \(PQ\)](#)

Menambahkan dua algoritma pertukaran kunci PQ baru ke kebijakan CloudFront TLS.

September 4, 2025

IPv6 permintaan asal	Bila Anda menggunakan custom origin (tidak termasuk Amazon S3 dan asal VPC), Anda dapat menyesuaikan pengaturan asal untuk distribusi Anda untuk memilih CloudFront cara menghubungkan ke asal Anda menggunakan atau alamat. IPv4 IPv6	September 3, 2025
Menambahkan kebijakan permintaan asal terkelola baru	Menambahkan kebijakan permintaan asal terkelola baru <code>HostHeaderOnly</code> .	Agustus 29, 2025
CloudFront titik akhir publik sekarang mendukung IPv6	Lihat CloudFront titik akhir dan kuota Amazon di Referensi Umum AWS dan dukungan Layanan AWS itu IPv6 di Panduan Pengguna Amazon VPC.	Agustus 21, 2025
Ditambahkan kebijakan keamanan penampil	TLSv1Ditambahkan.3_2025, kebijakan keamanan khusus TLS 1.3 yang baru.	Agustus 7, 2025
Menambahkan pengaturan batas waktu asal baru	Menambahkan batas waktu penyelesaian respons untuk semua asal, dan menambahkan batas waktu respons (batas waktu baca asal) untuk asal S3.	Juli 30, 2025
Ditambahkan pengaturan distribusi standar yang telah dikonfigurasi	Menambahkan pengaturan yang telah dikonfigurasi untuk distribusi standar.	Juni 17, 2025

Menambahkan alur kerja konsol baru untuk pengaturan domain distribusi standar	Menambahkan alur kerja konsol baru untuk pengaturan domain distribusi standar.	Juni 17, 2025
Ditambahkan parameter contoh	Menambahkan contoh untuk menggunakan parameter dengan nama domain dan jalur asal di penyewa distribusi.	Juni 17, 2025
Menambahkan dukungan CloudFront Fungsi untuk CloudFront SaaS Manager	Ditambahkan fungsi pembantu dan endpoint bidang untuk context objek.	2 Mei 2025
Pembaruan untuk logging standar (v2)	Ditambahkan variabel {distributionid} partisi untuk mendukung pengiriman log akses ke AWS Glue.	1 Mei 2025
Pembaruan kebijakan CloudFront terkelola	Menambahkan izin ACM ke CloudFrontReadOnlyAccess dan kebijakan CloudFrontFullAccess terkelola.	April 28, 2025

[Menambahkan dukungan untuk penyewa distribusi dan distribusi multi-tenant](#)

Anda dapat membuat distribusi multi-tenant untuk mengatur pengaturan distribusi umum berdasarkan tipe asal Anda. Kemudian, Anda dapat menggunakan kembali distribusi multi-tenant untuk membuat beberapa penyewa distribusi yang berbagi pengaturan tersebut. Anda kemudian dapat menyesuaikan penyewa distribusi tertentu saat Anda menambahkan situs web atau aplikasi tambahan.

April 28, 2025

[Pembaruan untuk fungsi Lambda @Edge](#)

Fungsi Lambda @Edge sekarang mendukung kontrol logging lanjutan dan menyesuaikan nama grup CloudWatch log.

April 7, 2025

[Anycast statis IPs](#)

Anda dapat menggunakan Anycast static IPs untuk mengaktifkan routing domain apex langsung ke distribusi Anda. CloudFront

April 4, 2025

[Menambahkan metode pembantu tambahan untuk modifikasi asal](#)

Menambahkan metode pembantu `selectRequestOriginById()` dan `createRequestOriginGroup()` CloudFront Fungsi.

April 2, 2025

Pembaruan untuk logging standar (v2)	Menambahkan variabel {accountid} partisi dan contoh jalur sufiks untuk pengiriman log akses ke Amazon S3.	Februari 14, 2025
Menambahkan bidang log akses real-time tambahan untuk logging standar (v2)	Anda dapat menentukan c-country dan bidang log akses cache-behavior-path-pattern real-time saat Anda mengaktifkan pencatatan standar (v2).	Januari 31, 2025
Lambda @Edge mendukung versi runtime yang lebih baru	Lambda @Edge sekarang mendukung fungsi Lambda dengan runtime Node.js 22.	November 22, 2024
Dukungan ketahanan sadar kualitas media untuk CloudFront	Anda dapat menggunakan fitur media quality-aware resiliency (MQAR) sehingga CloudFront secara otomatis memilih asal dalam grup asal dengan skor kualitas media tertinggi.	November 21, 2024
Metode pembantu untuk modifikasi asal	Menambahkan metode pembantu CloudFront Fungsi baru untuk modifikasi asal.	November 21, 2024
Asal VPC	Gunakan asal CloudFront VPC untuk membatasi akses ke Application Load Balancer, Network Load Balancer, atau asal instans EC2.	November 20, 2024
Pembaruan kebijakan terkelola	Kebijakan terkelola yang diperbarui CloudFrontFullAccess .	November 20, 2024

Anycast statis IPs	Anda dapat meminta Anycast statis IPs dari CloudFront untuk digunakan dengan distribusi Anda.	November 20, 2024
Ditambahkan dukungan untuk logging standar	CloudFront mendukung pencatatan standar (v2) dan mengirim log Anda ke Amazon CloudWatch Logs, Amazon Data Firehose, dan Amazon Simple Storage Service (Amazon S3).	November 20, 2024
Menambahkan dukungan untuk gRPC	CloudFront sekarang mendukung permintaan gRPC untuk distribusi Anda.	November 20, 2024
Menambahkan kebijakan terkelola baru untuk asal VPC	Menambahkan kebijakan terkelola baru AWS CloudFront VPC Origin Service Role Policy .	November 20, 2024
Lambda @Edge mendukung versi runtime yang lebih baru	Lambda @Edge sekarang mendukung fungsi Lambda dengan runtime Python 3.13.	November 13, 2024
Evaluasi dengan AWS Config Aturan	CloudFront Evaluasi konfigurasi Anda dengan AWS Config Aturan.	September 20, 2024
Menambahkan lebih banyak konten pemecahan masalah	Menambahkan lebih banyak konten pemecahan masalah untuk kode status respons kesalahan HTTP 4xx dan 5xx.	Agustus 26, 2024

Menambahkan kebijakan cache terkelola baru	Menambahkan kebijakan cache terkelola baru UseOriginCacheControlHeaders danUseOriginCacheControlHeaders-QueryString .	24 Mei 2024
Menambahkan dukungan kontrol akses asal	Anda sekarang dapat membuat kontrol akses asal (OAC) untuk AWS Elemental MediaPackage V2 dan URL AWS Lambda fungsi.	April 11, 2024
Bidang log akses real-time untuk CMCD	Menambahkan 18 bidang data klien media umum (CMCD) untuk log akses real-time.	April 9, 2024
Memulai dengan distribusi CloudFront standar	Tutorial yang diperbarui untuk distribusi standar yang menggunakan asal Amazon S3 dengan kontrol akses asal (OAC).	Maret 18, 2024
Contoh kode untuk CloudFront menggunakan AWS SDKs	Contoh kode yang ditambahkan yang menunjukkan cara menggunakan CloudFront kit pengembangan AWS perangkat lunak (SDK). Contoh dibagi menjadi kutipan kode yang menunjukkan cara memanggil fungsi layanan individual dan contoh yang menunjukkan cara menyelesaikan tugas tertentu dengan memanggil beberapa fungsi dalam layanan yang sama.	Februari 16, 2024

AWS kebijakan terkelola	Kebijakan CloudFront <code>ReadOnlyAccess</code> dan <code>CloudFrontFullAccess</code> IAM sekarang mendukung <code>KeyValueStore</code> operasi.	Desember 19, 2023
JavaScript runtime 2.0	Menambahkan fitur JavaScript runtime 2.0 untuk CloudFront Fungsi.	21 November 2023
CloudFront KeyValueStore	Amazon CloudFront sekarang mendukung CloudFront <code>KeyValueStore</code> . Fitur ini adalah datastore nilai kunci latensi rendah yang aman, global, dan memungkinkan akses baca dari dalam Fungsi. CloudFront Anda dapat mengaktifkan logika lanjutan yang dapat disesuaikan di lokasi CloudFront tepi.	21 November 2023
Lambda @Edge mendukung versi runtime yang lebih baru	Lambda @Edge sekarang mendukung fungsi Lambda dengan runtime Node.js 20.	15 November 2023
Dasbor keamanan	CloudFront membuat dasbor keamanan saat Anda membuat distribusi. Aktifkan AWS WAF, kelola pembatasan geografis, dan lihat data tingkat tinggi untuk permintaan, bot, dan log.	8 November 2023

Menyortir string kueri dalam fungsi	CloudFront sekarang mendukung query string sorting menggunakan CloudFront Functions.	3 Oktober 2023
AWS WAF rekomendasi keamanan	Amazon CloudFront sekarang menampilkan rekomendasi AWS WAF keamanan di CloudFront konsol.	26 September 2023
Support untuk menyajikan konten cache basi (kedaluwarsa)	CloudFront mendukung arahan kontrol Stale-While-Revalidate dan Stale-If-Error cache.	15 Mei 2023
Aktifkan AWS WAF perlindungan dengan satu klik	Metode yang disederhanakan untuk menambahkan perlindungan AWS WAF keamanan ke CloudFront distribusi.	10 Mei 2023
Aktifkan ACLs untuk bucket S3 baru yang digunakan untuk log standar	Menambahkan catatan dan tautan untuk mengatasi pengaturan ACL default untuk bucket S3 baru.	11 April 2023
Buat asal menggunakan Amazon S3 Object Lambda	Anda dapat menggunakan alias Titik Akses Lambda Objek Amazon S3 sebagai asal distribusi Anda.	31 Maret 2023
Sesuaikan status dan badan HTTP menggunakan CloudFront Fungsi	Anda dapat menggunakan CloudFront Fungsi untuk memperbarui kode status respons penampil dan mengganti atau menghapus badan respons.	29 Maret 2023

[Menambahkan opsi wildcard header CORS untuk port](#)

Anda sekarang dapat menyertakan konfigurasi wildcard untuk port di header kontrol akses CORS.

20 Maret 2023

[Menambahkan tautan baru untuk Panduan AWS Security Hub CSPM Pengguna](#)

Bahasa yang diperbarui dan tautan yang ditambahkan ke CloudFront kontrol Amazon yang direorganisasi di Panduan AWS Security Hub CSPM Pengguna.

9 Maret 2023

[CloudFront sekarang mendukung daftar blokir \(“semua kecuali”\) dalam kebijakan permintaan asal](#)

Gunakan daftar blokir dalam kebijakan permintaan asal untuk menyertakan semua string kueri, header HTTP, atau cookie, kecuali yang ditentukan, dalam permintaan yang CloudFront dikirim ke asal.

22 Februari 2023

[CloudFront menambahkan kebijakan permintaan asal terkelola baru untuk meneruskan semua header penampil kecuali header Host](#)

Gunakan CloudFront kebijakan permintaan asal terkelola yang baru untuk menyertakan semua header dari permintaan penampil, kecuali Host header, dalam permintaan yang CloudFront dikirim ke asal.

22 Februari 2023

[Pembatasan yang diperbarui pada Lambda @Edge](#)

Lambda @Edge mendukung konfigurasi manajemen runtime Lambda yang disetel ke Otomatis.

16 Februari 2023

Memperbarui panduan IAM untuk CloudFront	Memperbarui panduan untuk menyelaraskan dengan praktik terbaik IAM. Untuk informasi lebih lanjut, lihat Praktik terbaik keamanan di IAM .	15 Februari 2023
Keamanan yang ditingkatkan dengan kontrol akses asal	Anda sekarang dapat mengamankan MediaStore asal dengan mengizinkan akses hanya ke distribusi yang ditunjuk CloudFront .	9 Februari 2023
Header baru untuk menentukan struktur header penampil	Anda sekarang dapat menambahkan urutan header dan jumlah header untuk membantu mengidentifikasi penampil berdasarkan header yang dikirimkannya.	13 Januari 2023
Lambda @Edge mendukung versi runtime yang lebih baru	Lambda @Edge sekarang mendukung fungsi Lambda dengan runtime Node.js 18.	Januari 12, 2023
Hapus header respons menggunakan kebijakan header respons	Sekarang Anda dapat menggunakan kebijakan header CloudFront respons untuk menghapus header yang CloudFront diterima dalam respons dari asal. Header yang ditentukan tidak termasuk dalam respons yang CloudFront dikirim ke pemirsa.	Januari 3, 2023

Penerapan berkelanjutan untuk menguji perubahan konfigurasi dengan aman	Anda sekarang dapat menerapkan perubahan pada konfigurasi CDN Anda dengan menguji dengan subset lalu lintas produksi.	18 November 2022
Pelepasan CloudFront-Viewer-JA3-Fingerprint header	Anda sekarang dapat menggunakan JA3 sidik jari untuk membantu menentukan apakah permintaan tersebut berasal dari klien yang dikenal.	16 November 2022
Ditambahkan CORS header pilihan wildcard	Anda sekarang dapat menggunakan berbagai konfigurasi wildcard di beberapa header kontrol akses CORS.	11 November 2022
Metrik tambahan untuk distribusi CloudFront	Support untuk MonitoringSubscription di CloudFront API dan CloudFormation.	3 Oktober 2022
Keamanan yang ditingkatkan dengan kontrol akses asal	Anda sekarang dapat mengamankan asal Amazon S3 dengan mengizinkan akses hanya ke distribusi yang ditunjuk. CloudFront	Agustus 24, 2022
Dukungan HTTP/3 untuk distribusi CloudFront	Sekarang Anda dapat memilih HTTP/3 untuk distribusi Anda CloudFront .	Agustus 15, 2022
Tambahkan detail jabat tangan ke header CloudFront-Viewer-TLS	Anda dapat melihat informasi baru tentang SSL/TLS jabat tangan yang digunakan.	Juni 27, 2022

Metrik baru di header Server-Timing	Menambahkan cdn-downstream-fbl metrik baru ke Server-Timing header.	Juni 13, 2022
Header baru untuk mendapatkan informasi tentang versi TLS dan cipher	Anda sekarang dapat menggunakan CloudFront-Viewer-TLS header untuk mendapatkan informasi tentang versi TLS (atau SSL) dan cipher yang digunakan untuk koneksi antara penampil dan CloudFront	23 Mei 2022
FunctionThrottlesMetrik baru untuk CloudFront Fungsi	Dengan Amazon CloudWatch, Anda sekarang dapat memantau berapa kali CloudFront Fungsi dibatasi dalam periode waktu tertentu.	4 Mei, 2022
CloudFront mendukung fungsi Lambda URLs	Jika Anda membangun aplikasi web tanpa server dengan menggunakan fungsi Lambda dengan fungsi URLs, Anda sekarang dapat menambahkan berbagai CloudFront manfaat.	April 6, 2022
Header Server-Timing dalam tanggapan HTTP	Anda sekarang dapat mengaktifkan Server-Timing header dalam respons HTTP yang dikirim dari CloudFront untuk melihat metrik yang dapat membantu Anda mendapatkan wawasan tentang perilaku dan kinerja CloudFront	Maret 30, 2022

[Gunakan daftar awalan AWS-managed untuk membatasi lalu lintas masuk](#)

Anda sekarang dapat membatasi lalu lintas HTTP dan HTTPS masuk ke asal Anda hanya dari alamat IP CloudFront milik server yang menghadap asal.

Februari 7, 2022

[Fitur baru](#)

CloudFront menambahkan dukungan untuk kebijakan header respons, yang memungkinkan Anda menentukan header HTTP yang CloudFront menambah respons HTTP yang dikirimkan ke pemirsa (browser web atau klien lain). Anda dapat menentukan header yang diinginkan (dan nilainya) tanpa membuat perubahan apa pun pada asal atau menulis kode apa pun. Untuk informasi selengkapnya, lihat [Menambahkan atau menghapus header HTTP dalam CloudFront tanggapan](#).

2 November 2021

[Header CloudFront-Viewer-Address permintaan baru](#)

CloudFront menambahkan dukungan untuk header baru, CloudFront-Viewer-Address, yang berisi alamat IP penampil yang mengirim permintaan HTTP ke CloudFront. Untuk informasi selengkapnya, lihat [Menambahkan header CloudFront permintaan](#).

25 Oktober 2021

[Lambda @Edge mendukung versi runtime baru](#)

Lambda @Edge sekarang mendukung fungsi Lambda dengan runtime Python 3.9. Untuk informasi selengkapnya, lihat [Runtime yang didukung](#).

22 September 2021

[AWS pembaruan kebijakan terkelola](#)

CloudFront memperbarui CloudFrontReadOnlyAccess kebijakan. Untuk informasi selengkapnya, lihat [CloudFront pembaruan kebijakan AWS terkelola](#).

8 September 2021

[Fitur baru](#)

CloudFront sekarang mendukung sertifikat ECDSA untuk koneksi HTTPS yang menghadap pemirsa. Untuk informasi selengkapnya, lihat [Protokol dan sandi yang didukung antara pemirsa dan CloudFront dan Persyaratan untuk menggunakan sertifikat. SSL/TLS CloudFront](#)

14 Juli 2021

[Fitur baru](#)

CloudFront sekarang mendukung lebih banyak cara untuk memindahkan nama domain alternatif dari satu distribusi ke distribusi lainnya, tanpa menghubungi Dukungan. Untuk informasi selengkapnya, lihat [Memindahkan nama domain alternatif ke distribusi yang berbeda](#).

7 Juli 2021

Kebijakan keamanan baru

CloudFront sekarang mendukung kebijakan keamanan baru, TLSv1.2_2021, dengan serangkaian cipher yang didukung yang lebih kecil. Untuk informasi selengkapnya, lihat [Protokol dan sandi yang didukung antara pemirsa dan CloudFront](#).

23 Juni 2021

Fitur baru

Amazon CloudFront sekarang mendukung CloudFront Functions, fitur asli CloudFront yang memungkinkan Anda menulis fungsi ringan untuk penyesuaian CDN JavaScript skala tinggi yang sensitif terhadap latensi. Untuk informasi selengkapnya, lihat [Menyesuaikan di tepi dengan CloudFront Fungsi](#).

3 Mei 2021

Lambda @Edge mendukung versi runtime yang lebih baru

Lambda@Edge kini mendukung fungsi Lambda dengan waktu aktif Node.js 14. Untuk informasi selengkapnya, lihat [Runtime yang didukung](#).

29 April 2021

[Hapus dokumentasi untuk distribusi RTMP](#)

[Amazon CloudFront menghentikan distribusi real-time messaging protocol \(RTMP\) pada 31 Desember 2020.](#) Dokumentasi untuk distribusi RTMP sekarang dihapus dari Panduan CloudFront Pengembang Amazon.

10 Februari 2021

[Opsi harga baru](#)

Amazon CloudFront memperkenalkan bundel tabungan CloudFront keamanan, cara sederhana untuk menghemat hingga 30% pada CloudFront AWS tagihan Anda. Untuk informasi selengkapnya, lihat Paket Tabungan [FAQs](#).

5 Februari 2021

[Tutorial baru](#)

Panduan CloudFront Pengembang Amazon sekarang menyertakan tutorial untuk menggunakan Amazon untuk membatasi akses CloudFront ke Application Load Balancer di Elastic Load Balancing. Untuk informasi selengkapnya, lihat [Membatasi akses ke Application Load Balancers](#).

18 Desember 2020

[Opsi baru untuk manajemen kunci publik](#)

CloudFront sekarang mendukung manajemen kunci publik untuk cookie yang ditandatangani URLs dan ditandatangani melalui CloudFront konsol dan API, tanpa memerlukan akses ke pengguna Akun AWS root. Untuk informasi selengkapnya, lihat [Menentukan penandatangan yang dapat membuat cookie yang ditandatangani URLs dan ditandatangani](#).

22 Oktober 2020

[Fitur baru - Origin Shield](#)

CloudFront sekarang mendukung CloudFront Origin Shield, lapisan tambahan dalam infrastruktur CloudFront caching yang membantu meminimalkan beban asal Anda, meningkatkan ketersediaannya, dan mengurangi biaya operasinya. Untuk informasi selengkapnya, lihat [Menggunakan Amazon CloudFront Origin Shield](#).

20 Oktober 2020

Format kompresi baru

CloudFront sekarang mendukung formasi kompresi Brotli ketika Anda mengonfigurasi CloudFront untuk mengompres objek di lokasi CloudFront tepi. Anda juga dapat mengkonfigurasi CloudFront untuk menyimpan objek Brotli menggunakan header yang dinormalisasi `Accept-Encoding`. Untuk informasi selengkapnya, lihat [Menyajikan file terkompresi](#) dan [Dukungan kompresi](#).

14 September 2020

Protokol TLS baru

CloudFront sekarang mendukung protokol TLS 1.3 untuk koneksi HTTPS antara pemirsa dan CloudFront distribusi. TLS 1.3 diaktifkan secara default di semua kebijakan CloudFront keamanan. Untuk informasi selengkapnya, lihat [Protokol dan sandi yang didukung antara pemirsa dan CloudFront](#).

3 September 2020

[Log akses real-time baru](#)

CloudFront sekarang mendukung log akses real-time yang dapat dikonfigurasi. Dengan log akses real-time, Anda bisa mendapatkan informasi tentang permintaan yang dibuat ke distribusi secara real time. Anda dapat menggunakan log akses real-time untuk memantau, menganalisis, dan mengambil tindakan berdasarkan kinerja pengiriman konten. Untuk informasi selengkapnya, lihat [Log waktu nyata](#).

31 Agustus 2020

[Dukungan API untuk metrik tambahan](#)

CloudFront sekarang mendukung mengaktifkan delapan metrik real-time tambahan dengan API. CloudFront Untuk informasi selengkapnya, lihat [Mengaktifkan metrik tambahan](#).

28 Agustus 2020

[Header CloudFront HTTP baru](#)

CloudFront menambahkan header HTTP tambahan untuk menentukan informasi tentang penampil seperti jenis perangkat, lokasi geografis, dan banyak lagi. Untuk informasi selengkapnya, lihat [Menambahkan header CloudFront permintaan](#).

23 Juli 2020

Fitur baru	CloudFront sekarang mendukung kebijakan cache dan kebijakan permintaan asal, yang memberi Anda kontrol lebih terperinci atas kunci cache dan permintaan asal untuk distribusi Anda CloudFront . Untuk informasi selengkapnya, lihat Mengontrol kunci cache dan Mengontrol permintaan asal .	22 Juli 2020
Kebijakan keamanan baru	CloudFront sekarang mendukung kebijakan keamanan baru, TLSv1.2_2019, dengan serangkaian cipher yang didukung yang lebih kecil. Untuk informasi selengkapnya, lihat Protokol dan sandi yang didukung antara pemirsa dan CloudFront .	8 Juli 2020
Pengaturan baru untuk mengontrol batas waktu dan upaya asal	CloudFront menambahkan pengaturan baru yang mengontrol batas waktu dan upaya asal. Untuk informasi selengkapnya, lihat Mengontrol batas waktu dan upaya asal .	5 Juni 2020

[Dokumentasi baru untuk memulai CloudFront dengan membuat situs web statis yang aman](#)

Mulailah CloudFront dengan membuat situs web statis aman menggunakan Amazon S3,, CloudFront Lambda @Edge, dan lainnya, semuanya digunakan . CloudFormation Untuk informasi selengkapnya, lihat [Memulai dengan situs web statis yang aman](#).

2 Juni 2020

[Lambda @Edge mendukung versi runtime yang lebih baru](#)

Lambda@Edge kini mendukung fungsi Lambda dengan masa aktif Node.js 12 dan Python 3.8. Untuk informasi selengkapnya, lihat [Runtime yang didukung](#).

27 Februari 2020

[Metrik real-time baru di CloudWatch](#)

Amazon CloudFrontnow menawarkan delapan metrik real-time tambahan di Amazon CloudWatch. Untuk informasi selengkapnya, lihat [Mengaktifkan metrik CloudFront distribusi tambahan](#).

19 Desember 2019

[Bidang baru di log akses](#)

CloudFront menambahkan tujuh bidang baru untuk mengakses log. Untuk informasi selengkapnya, lihat [Bidang file log standar](#).

12 Desember 2019

[AWS WordPress plugin](#)

Anda dapat menggunakan an AWS WordPress plugin untuk memberikan pengunjung ke WordPress situs web Anda pengalaman menonton yang dipercepat menggunakan CloudFront. (Pembaruan: per 30 September 2022, WordPress plugin AWS for tidak digunakan lagi.)

30 Oktober 2019

[Kebijakan izin IAM berbasis tag dan tingkat sumber daya](#)

CloudFront sekarang mendukung dua cara tambahan untuk menentukan kebijakan izin IAM: izin kebijakan berbasis tag dan tingkat sumber daya. Untuk informasi lebih lanjut, lihat [Mengelola Akses ke Sumber Daya](#).

8 Agustus 2019

[Support untuk bahasa pemrograman Python](#)

Sekarang Anda dapat menggunakan bahasa pemrograman Python untuk mengembangkan fungsi di Lambda@Edge, selain Node.js. Misalnya, fungsi yang mencakup berbagai skenario, lihat [Fungsi Contoh Lambda@Edge](#).

1 Agustus 2019

[Grafik pemantauan yang diperbarui](#)

Pembaruan konten untuk menjelaskan cara baru bagi Anda untuk memantau fungsi Lambda yang terkait dengan CloudFront distribusi Anda langsung dari CloudFront konsol untuk lebih mudah melacak dan men-debug kesalahan. Untuk informasi lebih lanjut, lihat [Monitoring CloudFront](#).

20 Juni 2019

[Konten keamanan terkonsolidasi](#)

Bab Keamanan baru mengkonsolidasikan informasi tentang CloudFront fitur di sekitar dan implementasi perlindungan data, IAM, logging, kepatuhan, dan banyak lagi. Untuk informasi lebih lanjut, lihat [Keamanan](#).

24 Mei 2019

[Validasi domain sekarang diperlukan](#)

CloudFront sekarang mengharuskan Anda menggunakan sertifikat SSL untuk memverifikasi bahwa Anda memiliki izin untuk menggunakan nama domain alternatif dengan distribusi. Untuk informasi lebih lanjut, lihat [Menggunakan Nama Domain Alternatif dan HTTPS](#).

9 April 2019

[Nama file PDF yang diperbarui](#)

Nama file baru untuk Panduan CloudFront Pengembang Amazon adalah: AmazonCloudFront_. DevGuide Nama sebelumnya adalah: cf-dg.

7 Januari 2019

Fitur baru

CloudFront sekarang mendukung WebSocket, protokol berbasis TCP yang berguna ketika Anda membutuhkan koneksi jangka panjang antara klien dan server. Anda juga sekarang dapat mengatur CloudFront dengan failover asal untuk skenario yang membutuhkan ketersediaan tinggi. Untuk informasi selengkapnya, lihat [Menggunakan WebSocket dengan CloudFront Distribusi](#) dan [Mengoptimalkan Ketersediaan Tinggi dengan CloudFront Origin Failover](#).

20 November 2018

Fitur baru

CloudFront sekarang mendukung pencatatan kesalahan terperinci untuk permintaan HTTP yang menjalankan fungsi Lambda. Anda dapat menyimpan log CloudWatch dan menggunakannya untuk membantu memecahkan masalah kesalahan HTTP 5xx saat fungsi Anda mengembalikan respons yang tidak valid. Untuk informasi selengkapnya, lihat [CloudWatch Metrik dan CloudWatch Log untuk Fungsi Lambda](#).

8 Oktober 2018

Fitur baru

Sekarang Anda dapat memilih untuk memasukkan Lambda@Edge mengekspos tubuh dalam permintaan metode HTTP yang dapat ditulis (POST, PUT, DELETE, dan seterusnya), sehingga Anda dapat mengaksesnya dalam fungsi Lambda Anda. Anda dapat memilih akses hanya-baca, atau Anda dapat menentukan bahwa Anda akan mengganti isi. Untuk informasi lebih lanjut, lihat [Mengakses Badan Permintaan dengan Memilih Opsi Sertakan Tubuh](#).

14 Agustus 2018

Fitur baru

CloudFront sekarang mendukung penyajian konten yang dikompresi dengan menggunakan brotli atau algoritma kompresi lainnya, selain atau bukan gzip. Untuk informasi lebih lanjut, lihat [Menyajikan File Bertekanan](#).

25 Juli 2018

Reorganisasi

Panduan CloudFront Pengembang Amazon telah direorganisasi untuk menyederhanakan menemukan konten terkait, dan untuk meningkatkan pemindaian dan navigasi.

28 Juni 2018

Fitur Baru

Lambda @Edge sekarang memungkinkan Anda untuk lebih menyesuaikan pengiriman konten yang disimpan dalam bucket Amazon S3, dengan memungkinkan Anda mengakses header tambahan, termasuk header khusus, dalam peristiwa yang menghadap ke asal. Untuk informasi lebih lanjut, lihat contoh ini yang menunjukkan personalisasi konten berdasarkan [lokasi penampil](#) dan [jenis perangkat penampil](#).

20 Maret 2018

Fitur Baru

Anda sekarang dapat menggunakan Amazon CloudFront untuk menegosiasikan koneksi HTTPS ke asal menggunakan Elliptic Curve Digital Signature Algorithm (ECDSA). ECDSA menggunakan kunci lebih kecil yang lebih cepat, namun, sama amannya, seperti algoritma RSA yang lebih lama. Untuk informasi selengkapnya, lihat [SSL/TLS Protokol dan Cipher yang Didukung untuk Komunikasi Antara dan Asal Anda CloudFront dan Tentang Cipher RSA](#) dan ECDSA.

15 Maret 2018

[Fitur Baru](#)

Lambda @Edge memungkinkan Anda untuk menyesuaikan respons kesalahan dari asal Anda, dengan memungkinkan Anda menjalankan fungsi Lambda sebagai respons terhadap kesalahan HTTP yang Amazon CloudFront receives dari asal Anda. Untuk informasi lebih lanjut, lihat contoh ini yang menunjukkan [mengalihkan ke lokasi lain](#) dan [pembuatan respons dengan 200 kode status \(OK\)](#).

21 Desember 2017

[Fitur Baru](#)

CloudFront Kemampuan baru, enkripsi tingkat lapangan, membantu Anda untuk lebih meningkatkan keamanan data sensitif, seperti nomor kartu kredit atau informasi identitas pribadi (PII) seperti nomor jaminan sosial. Untuk informasi selengkapnya, lihat [Menggunakan enkripsi tingkat lapangan untuk membantu melindungi](#) data sensitif.

14 Desember 2017

[Riwayat dokumen diarsipkan](#)

Riwayat dok lama diarsipkan.

1 Desember 2017

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.