



Panduan Pengguna

# AWS CloudShell



# AWS CloudShell: Panduan Pengguna

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 di antara 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 AWS CloudShell? .....	1
Fitur dari AWS CloudShell .....	1
AWS Command Line Interface .....	2
Kerang dan alat pengembangan .....	2
Penyimpanan tetap .....	2
CloudShell Lingkungan VPC .....	3
Keamanan .....	3
Opsi kustomisasi .....	4
Pemulihan sesi .....	4
Harga untuk AWS CloudShell .....	4
AWS CloudShell Topik utama .....	4
Memulai .....	6
Prasyarat .....	6
Daftar Isi .....	7
Langkah 1: Masuk ke Konsol Manajemen AWS .....	7
Langkah 2: Pilih Wilayah, luncurkan AWS CloudShell, dan pilih shell .....	8
Langkah 3: Unduh file dari AWS CloudShell .....	11
Langkah 4: Unggah file ke AWS CloudShell .....	12
Langkah 5: Hapus file dari AWS CloudShell .....	13
Langkah 6: Buat cadangan direktori home .....	13
Langkah 7: Mulai ulang sesi shell .....	15
Langkah 8: Hapus direktori home sesi shell .....	16
Langkah 9: Edit kode file Anda dan jalankan menggunakan baris perintah .....	17
Langkah 10: Gunakan AWS CLI untuk menambahkan file sebagai objek di bucket Amazon S3 .....	18
Topik terkait .....	20
Tutorial .....	21
Tutorial: Menyalin banyak file .....	21
Mengunggah dan mengunduh banyak file menggunakan Amazon S3 .....	22
Mengunggah dan mengunduh banyak file menggunakan folder zip .....	25
Tutorial: Membuat presigned URLs .....	27
Prasyarat .....	27
Langkah 1: Buat peran IAM untuk memberikan akses ke bucket Amazon S3 .....	27
Hasilkan URL yang telah ditentukan sebelumnya .....	29

Tutorial: Membangun wadah Docker di dalam CloudShell dan mendorong ke Amazon ECR .....	30
Prasyarat .....	30
Prosedur Tutorial .....	30
Bersihkan .....	33
Tutorial: Menyebarkan fungsi Lambda menggunakan AWS CDK .....	33
Prasyarat .....	33
Prosedur Tutorial .....	33
Bersihkan .....	36
AWS CloudShell Konsep .....	37
Menavigasi antarmuka AWS CloudShell .....	37
Bekerja di Wilayah AWS .....	39
Menentukan default Wilayah AWS Anda untuk AWS CLI .....	39
Bekerja dengan file dan penyimpanan .....	40
Akses CloudShell di Aplikasi Mobile Console .....	41
Bekerja dengan Docker .....	41
Fitur aksesibilitas .....	43
Navigasi keyboard di CloudShell .....	43
CloudShell fitur aksesibilitas terminal .....	43
Memilih ukuran font dan tema antarmuka di CloudShell .....	43
Kelola AWS layanan .....	45
AWS CLI contoh baris perintah untuk AWS layanan yang dipilih .....	45
DynamoDB .....	46
Amazon EC2 .....	46
Amazon Glacier .....	46
AWS CLI Elastic Beanstalk .....	47
Amazon ECS CLI .....	47
AWS SAM CLI .....	48
Kiro CLI di CloudShell .....	49
Menggunakan perintah obrolan Kiro di CloudShell .....	49
Menggunakan perintah terjemahan Kiro di CloudShell .....	49
Penyelesaian perintah CLI di CloudShell .....	49
Saran sebaris Kiro di CloudShell .....	50
Kebijakan berbasis identitas untuk Kiro CLI di CloudShell .....	50
Menjalankan perintah CloudShell dari AWS konsol Layanan .....	51
Menyesuaikan AWS CloudShell .....	53
Memisahkan tampilan baris perintah menjadi beberapa tab .....	53

Mengubah ukuran font .....	54
Mengubah tema antarmuka .....	54
Menggunakan Safe Paste untuk teks multiline .....	54
Penggunaan tmux ke pemulihan sesi .....	55
.....	55
Menggunakan AWS CloudShell di Amazon Virtual Private Cloud (Amazon VPC) .....	56
Kendala operasi .....	56
Menciptakan lingkungan CloudShell VPC .....	57
Izin IAM yang diperlukan untuk membuat dan menggunakan lingkungan VPC CloudShell .....	58
Kebijakan IAM memberikan CloudShell akses penuh termasuk akses ke VPC .....	59
Menggunakan kunci kondisi IAM untuk lingkungan VPC .....	62
Contoh kebijakan dengan kunci syarat untuk pengaturan VPC .....	63
Keamanan .....	3
Perlindungan data .....	68
Enkripsi data .....	69
Identity and Access Management .....	69
Audiens .....	70
Mengautentikasi dengan identitas .....	70
Mengelola akses menggunakan kebijakan .....	72
Bagaimana AWS CloudShell bekerja dengan IAM .....	73
Identity-based contoh kebijakan .....	79
Pemecahan masalah .....	82
Mengelola AWS CloudShell akses dan penggunaan dengan kebijakan IAM .....	84
Pencatatan log dan pemantauan .....	98
Memantau aktivitas dengan CloudTrail .....	98
AWS CloudShell di CloudTrail .....	99
Validasi kepatuhan .....	101
Ketahanan .....	106
Keamanan infrastruktur .....	107
Praktik terbaik keamanan .....	107
FAQ Keamanan .....	108
Apa AWS proses dan teknologi digunakan saat Anda meluncurkan CloudShell dan memulai sesi shell? .....	108
Apakah mungkin untuk membatasi akses jaringan? CloudShell .....	109
Dapatkah saya menyesuaikan CloudShell lingkungan saya? .....	109
Dimana saya \$ RUMAH direktori sebenarnya disimpan di AWS Cloud? .....	109

Apakah mungkin untuk mengenkripsi saya \$ RUMAH direktori? .....	110
Dapatkah saya menjalankan pemindaian virus pada saya \$ RUMAH direktori? .....	110
Dapatkah saya membatasi masuknya data atau keluar untuk saya? CloudShell .....	110
Bisakah saya membatasi CloudShell pengguna dari akses root di dalam wadah? .....	110
AWS CloudShell lingkungan komputasi .....	112
Menghitung sumber daya lingkungan .....	112
CloudShell persyaratan jaringan .....	112
Perangkat lunak pra-instal .....	113
Kerang .....	114
AWS antarmuka baris perintah (CLI) .....	114
Runtime dan AWS SDKs: Node.js dan Python 3 .....	118
Alat pengembangan dan utilitas shell .....	121
Menginstal AWS CLI ke direktori home Anda .....	130
Menginstal perangkat lunak pihak ketiga di lingkungan shell Anda .....	131
Memodifikasi shell Anda dengan skrip .....	132
Migrasi dari Amazon Linux 2 ke Amazon Linux 2023 .....	133
AWS CloudShell Migrasi FAQs .....	134
Pemecahan Masalah .....	136
Memecahkan masalah kesalahan .....	136
Akses ditolak .....	137
Izin tidak cukup .....	137
Tidak dapat mengakses AWS CloudShell baris perintah .....	137
Tidak dapat melakukan ping ke alamat IP eksternal .....	138
Ada beberapa masalah dalam mempersiapkan terminal Anda .....	138
Tombol panah tidak berfungsi dengan benar di PowerShell .....	138
Soket Web yang tidak didukung menyebabkan kegagalan untuk memulai sesi CloudShell ..	140
Tidak dapat mengimpor AWSPowerShell.NetCoremodul .....	141
Docker tidak berjalan saat menggunakan AWS CloudShell .....	142
Docker kehabisan ruang disk .....	142
docker push habis waktu dan terus mencoba lagi .....	142
Tidak dapat mengakses sumber daya dalam VPC dari saya AWS CloudShell Lingkungan VPC .....	143
ENI yang digunakan oleh AWS CloudShell karena lingkungan VPC saya tidak dibersihkan .	143
Pengguna dengan CreateEnvironmentizin hanya untuk lingkungan VPC juga memiliki akses ke publik AWS CloudShell lingkungan .....	144
Wilayah yang Didukung .....	145

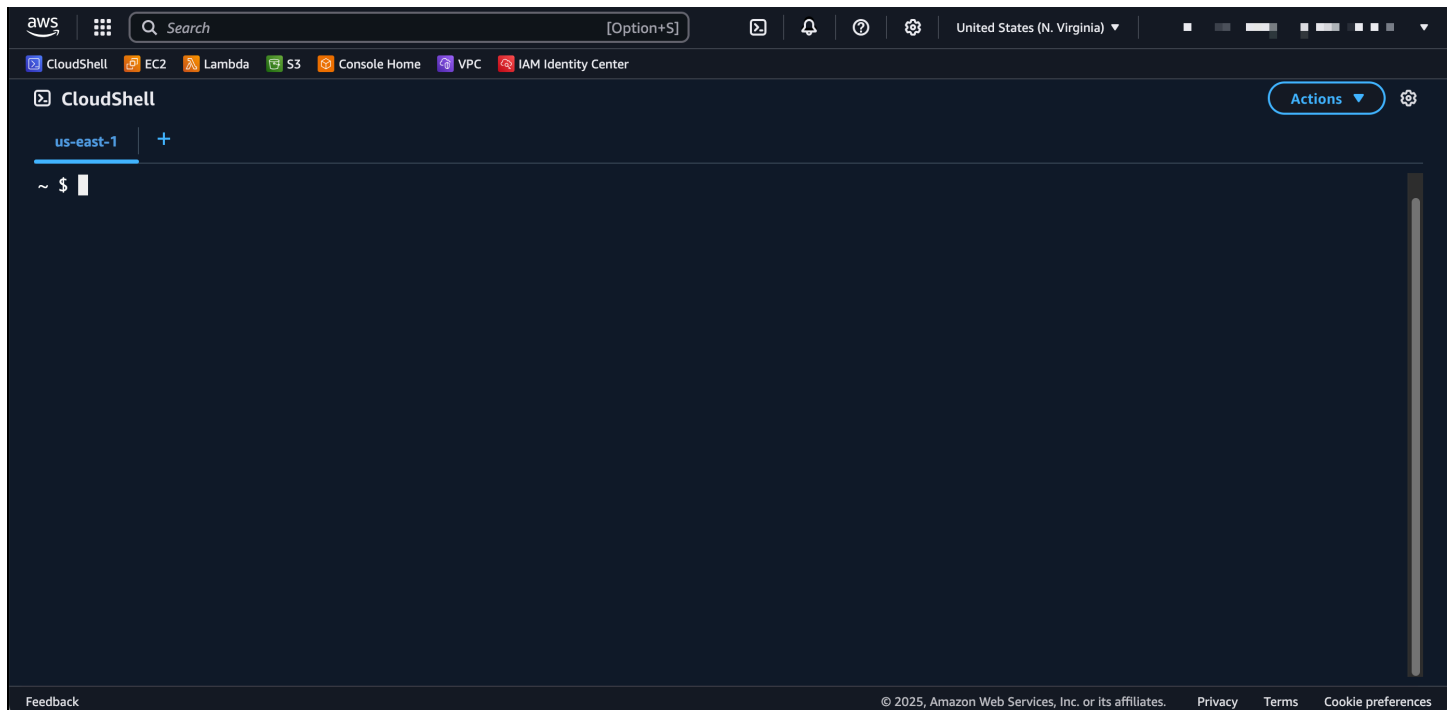
---

Kuota dan batasan layanan .....	146
Penyimpanan tetap .....	146
Penggunaan bulanan .....	147
Cangkang bersamaan .....	147
Ukuran perintah .....	148
Sesi Shell .....	148
Lingkungan VPC .....	148
Akses jaringan dan transfer data .....	149
Pembatasan pada file sistem dan pemuatan ulang halaman .....	149
Riwayat dokumen .....	150
.....	cliv

# Apa itu AWS CloudShell?

AWS CloudShell adalah shell pra-otentikasi berbasis browser yang dapat Anda luncurkan langsung dari file. Konsol Manajemen AWS Anda dapat menavigasi ke CloudShell dari Konsol Manajemen AWS beberapa cara yang berbeda. Untuk informasi selengkapnya, lihat [Memulai AWS CloudShell](#)

Anda dapat menjalankan AWS CLI perintah menggunakan shell pilihan Anda, seperti Bash, PowerShell, atau Z shell. Dan Anda dapat melakukan ini tanpa mengunduh atau menginstal alat baris perintah.



Saat Anda meluncurkan AWS CloudShell, [lingkungan komputasi](#) yang didasarkan pada Amazon Linux 2023 dibuat. Dalam lingkungan ini, Anda dapat mengakses [berbagai alat pengembangan pra-instal](#), opsi untuk [mengunggah](#) dan [mengunduh](#) file, dan [penyimpanan file yang bertahan](#) di antara sesi. Anda dapat menggunakan CloudShell di versi terbaru Google Chrome, Mozilla Firefox, Microsoft Edge, dan browser Apple Safari.

(Coba sekarang: [Memulai dengan AWS CloudShell](#))

## Fitur dari AWS CloudShell

AWS CloudShell menyediakan fitur-fitur berikut:

## AWS Command Line Interface

Anda dapat meluncurkan AWS CloudShell dari Konsol Manajemen AWS. AWS Kredensi yang Anda gunakan untuk masuk ke konsol secara otomatis tersedia di sesi shell baru. Karena AWS CloudShell pengguna telah diautentikasi sebelumnya, Anda tidak perlu mengonfigurasi kredensial saat berinteraksi dengan menggunakan versi 2. Layanan AWS CLI AWS CLI Ini sudah diinstal sebelumnya di lingkungan komputasi shell.

Untuk informasi selengkapnya tentang berinteraksi dengan Layanan AWS menggunakan antarmuka baris perintah, lihat [Kelola AWS layanan dari CLI di CloudShell](#).

## Kerang dan alat pengembangan

Dengan shell yang dibuat untuk AWS CloudShell sesi, Anda dapat beralih dengan mulus di antara shell baris perintah pilihan Anda. Lebih khusus lagi, Anda dapat beralih antara Bash, PowerShell, dan Z shell. Anda juga memiliki akses ke alat dan utilitas pra-instal. Ini termasuk git, make, pip, sudo, tar, tmux, vim, wget, dan zip.

Lingkungan shell sudah dikonfigurasi sebelumnya dengan dukungan untuk beberapa bahasa perangkat lunak utama terkemuka, seperti Node.js dan Python. Ini berarti, misalnya, Anda dapat menjalankan Node.js dan Python memproyeksikan tanpa terlebih dahulu melakukan instalasi runtime. PowerShell pengguna dapat menggunakan .NET Core runtime.

Untuk informasi selengkapnya, lihat [AWS CloudShell lingkungan komputasi: spesifikasi dan perangkat lunak](#).

## Penyimpanan tetap

Dengan AWS CloudShell, Anda dapat menggunakan hingga 1 GB penyimpanan persisten Wilayah AWS di masing-masing tanpa biaya tambahan. Penyimpanan persisten terletak di direktori home Anda (\$HOME) dan bersifat pribadi untuk Anda. Tidak seperti sumber daya lingkungan sementara yang didaur ulang setelah setiap sesi shell berakhir, data di direktori home Anda tetap ada di antara sesi.

Untuk informasi selengkapnya tentang retensi data dalam penyimpanan persisten, lihat [Penyimpanan tetap](#).

**Note**

CloudShell Lingkungan VPC tidak memiliki penyimpanan persisten. Direktori \$HOME dihapus ketika waktu lingkungan VPC Anda habis (setelah 20-30 menit tidak aktif), atau ketika Anda menghapus atau memulai ulang lingkungan Anda.

## CloudShell Lingkungan VPC

AWS CloudShell Virtual Private Cloud (VPC) memungkinkan Anda untuk menciptakan CloudShell lingkungan di VPC Anda. Untuk setiap lingkungan VPC, Anda dapat menetapkan VPC, menambahkan subnet, dan mengaitkan satu atau beberapa grup keamanan. AWS CloudShell mewarisi konfigurasi jaringan VPC dan memungkinkan Anda untuk AWS CloudShell menggunakan dengan aman dalam subnet yang sama dengan sumber daya lain di VPC.

## Keamanan

AWS CloudShell Lingkungan dan penggunaannya dilindungi oleh fitur keamanan tertentu. Ini termasuk fitur seperti manajemen izin IAM, pembatasan sesi shell, dan Safe Paste untuk input teks.

### Manajemen izin dengan IAM

Sebagai administrator, Anda dapat memberikan dan menolak izin kepada AWS CloudShell pengguna yang menggunakan kebijakan IAM. Anda juga dapat membuat kebijakan yang menentukan tindakan tertentu yang dapat dilakukan pengguna dengan lingkungan shell. Untuk informasi selengkapnya, lihat [Mengelola AWS CloudShell akses dan penggunaan dengan kebijakan IAM](#).

### Manajemen sesi Shell

Sesi yang tidak aktif dan berjalan lama secara otomatis dihentikan dan didaur ulang. Untuk informasi selengkapnya, lihat [Sesi Shell](#).

### Tempel Aman untuk masukan teks

Safe Paste diaktifkan secara default. Fitur keamanan ini mengharuskan Anda memverifikasi bahwa teks multiline yang ingin Anda tempel ke shell tidak mengandung skrip berbahaya. Untuk informasi selengkapnya, lihat [Menggunakan Safe Paste untuk teks multiline](#).

## Opsi kustomisasi

Anda dapat menyesuaikan AWS CloudShell pengalaman Anda dengan preferensi Anda yang tepat. Misalnya, Anda dapat mengubah tata letak layar (beberapa tab), ukuran teks yang ditampilkan, dan beralih di antara tema antarmuka terang dan gelap. Untuk informasi selengkapnya, lihat [Menyesuaikan AWS CloudShell pengalaman](#).

Anda juga dapat memperluas lingkungan shell Anda dengan [menginstal perangkat lunak Anda sendiri](#) dan [memodifikasi shell Anda dengan skrip](#).

## Pemulihan sesi

Fungsi pemulihan sesi mengembalikan sesi yang Anda jalankan di satu atau beberapa tab browser di CloudShell terminal. Jika Anda menyegarkan atau membuka kembali tab browser yang baru saja ditutup, fungsi ini melanjutkan sesi hingga shell dihentikan karena sesi tidak aktif. Untuk terus menggunakan CloudShell sesi Anda, tekan tombol apa saja di dalam jendela terminal. Untuk informasi selengkapnya tentang sesi Shell, lihat [sesi Shell](#).

Pemulihan sesi juga mengembalikan output terminal terbaru dan menjalankan proses di setiap tab terminal.

### Note

Pemulihan sesi tidak tersedia di aplikasi seluler.

## Harga untuk AWS CloudShell

AWS CloudShell adalah Layanan AWS yang tersedia tanpa biaya tambahan. Namun, Anda membayar AWS sumber daya lain yang Anda jalankan AWS CloudShell. Selain itu, [tarif transfer data standar](#) juga berlaku. Untuk informasi selengkapnya, lihat [harga AWS CloudShell](#).

Untuk informasi selengkapnya, lihat [Kuota layanan dan batasan untuk AWS CloudShell](#).

## AWS CloudShell Topik utama

- [Memulai dengan AWS CloudShell](#)
- [AWS CloudShell Konsep](#)

- [Kelola AWS layanan dari CLI di CloudShell](#)
- [Menyesuaikan AWS CloudShell pengalaman](#)
- [AWS CloudShell lingkungan komputasi: spesifikasi dan perangkat lunak](#)

# Memulai dengan AWS CloudShell

Tutorial pengantar ini menunjukkan kepada Anda cara meluncurkan AWS CloudShell dan melakukan tugas-tugas utama menggunakan antarmuka baris perintah shell.

Pertama, Anda masuk ke Konsol Manajemen AWS dan pilih file Wilayah AWS. Anda kemudian meluncurkan CloudShell di jendela browser baru dan jenis shell untuk bekerja dengan.

Selanjutnya, Anda membuat folder baru di direktori home Anda dan mengunggah file ke sana dari mesin lokal Anda. Anda mengerjakan file itu menggunakan editor pra-instal sebelum menjalankannya sebagai program dari baris perintah. Terakhir, Anda memanggil AWS CLI perintah untuk membuat bucket Amazon S3 dan menambahkan file Anda sebagai objek ke bucket.

## Prasyarat

### Izin IAM

Anda dapat memperoleh izin AWS CloudShell dengan melampirkan kebijakan AWS terkelola berikut ke identitas IAM Anda (seperti pengguna, peran, atau grup):

- `AWSCloudShellFullAccess`: Menyediakan pengguna dengan akses penuh ke AWS CloudShell dan fitur-fiturnya.

Untuk tutorial ini, Anda juga berinteraksi dengan Layanan AWS. Lebih khusus lagi, Anda berinteraksi dengan Amazon S3 dengan membuat bucket S3 dan menambahkan objek ke bucket itu. Identitas IAM Anda memerlukan kebijakan yang memberikan, setidaknya, izin `s3:CreateBucket` dan `s3:PutObject` izin.

Untuk informasi selengkapnya, lihat [Tindakan Amazon S3](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

### File latihan

Latihan ini juga melibatkan mengunggah dan mengedit file yang kemudian dijalankan sebagai program dari antarmuka baris perintah. Buka editor teks di mesin lokal Anda dan tambahkan cuplikan kode berikut.

```
import sys
```

```
x=int(sys.argv[1])
y=int(sys.argv[2])
sum=x+y
print("The sum is",sum)
```

Simpan file dengan nama `add_prog.py`.

## Daftar Isi

- [Langkah 1: Masuk ke Konsol Manajemen AWS](#)
- [Langkah 2: Pilih Wilayah, luncurkan AWS CloudShell, dan pilih shell](#)
- [Langkah 3: Unduh file dari AWS CloudShell](#)
- [Langkah 4: Unggah file ke AWS CloudShell](#)
- [Langkah 5: Hapus file dari AWS CloudShell](#)
- [Langkah 6: Buat cadangan direktori home](#)
- [Langkah 7: Mulai ulang sesi shell](#)
- [Langkah 8: Hapus direktori home sesi shell](#)
- [Langkah 9: Edit kode file Anda dan jalankan dari baris perintah](#)
- [Langkah 10: Gunakan AWS CLI untuk menambahkan file sebagai objek di bucket Amazon S3](#)

## Langkah 1: Masuk ke Konsol Manajemen AWS

Langkah ini melibatkan memasukkan informasi pengguna IAM Anda untuk mengakses. Konsol Manajemen AWS Jika Anda sudah berada di konsol, lewati ke [langkah 2](#).

- Anda dapat mengakses Konsol Manajemen AWS dengan menggunakan URL masuk pengguna IAM atau membuka halaman masuk utama.

### IAM user sign-in URL

- Buka browser dan masukkan URL masuk berikut. Ganti `account_alias_or_id` dengan alias akun atau ID akun yang diberikan administrator Anda.

```
https://account_alias_or_id.signin.aws.amazon.com/console/
```

- Masukkan kredensi masuk IAM Anda dan pilih Masuk.

## Main sign-in page

- Buka <https://aws.amazon.com/console/>.
- Jika sebelumnya Anda tidak masuk menggunakan browser ini, halaman masuk utama akan muncul. Pilih pengguna IAM, masukkan alias akun atau ID akun, dan pilih Berikutnya.
- Jika Anda sudah masuk sebagai pengguna IAM sebelumnya. Browser Anda mungkin mengingat alias akun atau ID akun untuk. Akun AWSJika demikian, masukkan kredensial masuk IAM Anda dan pilih Masuk.

### Note

Anda juga dapat masuk sebagai [pengguna root](#). Identitas ini memiliki akses lengkap ke semua Layanan AWS dan sumber daya di akun. Kami sangat menyarankan agar Anda tidak menggunakan pengguna root untuk tugas sehari-hari, bahkan yang administratif. Sebagai gantinya, patuhi praktik terbaik dalam menggunakan pengguna root saja untuk membuat pengguna IAM pertama Anda.

## Langkah 2: Pilih Wilayah, luncurkan AWS CloudShell, dan pilih shell

Pada langkah ini, Anda meluncurkan CloudShell dari antarmuka konsol, memilih yang tersedia Wilayah AWS, dan beralih ke shell pilihan Anda, seperti Bash, PowerShell, atau Z shell.

1. Untuk memilih Wilayah AWS untuk bekerja, buka menu Pilih Wilayah dan pilih [AWS Wilayah yang didukung](#) untuk bekerja. (Wilayah yang Tersedia disorot.)

### Important

Jika Anda beralih Wilayah, antarmuka menyegarkan dan nama yang dipilih Wilayah AWS ditampilkan di atas teks baris perintah. File apa pun yang Anda tambahkan ke penyimpanan persisten hanya tersedia dalam hal yang sama Wilayah AWS. Jika Anda mengubah Wilayah, penyimpanan dan file yang berbeda dapat diakses.

### Important

Jika CloudShell tidak tersedia di Wilayah yang dipilih saat Anda meluncurkan CloudShell di Console Toolbar, di kiri bawah konsol, maka Wilayah default disetel ke Wilayah yang paling dekat dengan Wilayah yang dipilih. Anda dapat menjalankan perintah yang menyediakan izin untuk mengelola sumber daya di Wilayah yang berbeda dari Region default. Untuk informasi lebih lanjut, lihat [Bekerja di Wilayah AWS](#).

### Example

#### Contoh

Jika Anda memilih Eropa (Spain-south-2) CloudShell tetapi tidak tersedia di Eropa (Spain-south-2), maka Wilayah default diatur ke Eropa (Ireland-west-1), yang paling dekat dengan Eropa (Spain). eu-south-2

Anda akan menggunakan kuota layanan untuk Wilayah default, Eropa (Ireland) eu-west-1 dan CloudShell sesi yang sama akan dipulihkan di semua Wilayah. Wilayah default dapat diubah dan Anda akan diberi tahu di jendela CloudShell browser.

2. Dari Konsol Manajemen AWS, Anda dapat meluncurkan CloudShell dengan memilih salah satu opsi berikut:
  1. Pada bilah navigasi, pilih CloudShell ikon.
  2. Di kotak Pencarian, ketik "CloudShell", lalu pilih CloudShell.
  3. Di widget yang baru dikunjungi, pilih CloudShell.
  4. Pilih CloudShell pada Console Toolbar, di kiri bawah konsol.
    - Anda dapat menyesuaikan ketinggian CloudShell sesi Anda dengan menyeret=.
    - Anda dapat mengalihkan CloudShell sesi Anda ke layar penuh dengan mengklik Buka di tab browser baru.

Ketika command prompt ditampilkan, shell siap untuk interaksi.

**Note**

Jika Anda mengalami masalah yang mencegah Anda berhasil meluncurkan atau berinteraksi AWS CloudShell, periksa informasi untuk mengidentifikasi dan mengatasi masalah tersebut. [Pemecahan Masalah AWS CloudShell](#)

3. Untuk memilih shell pra-instal untuk bekerja dengan, masukkan nama programnya pada prompt baris perintah.

**Bash**

```
bash
```

Jika Anda beralih keBash, simbol pada prompt perintah diperbarui ke\$.

**Note**

Bashadalah shell default yang berjalan saat Anda meluncurkan AWS CloudShell.

**PowerShell**

```
pwsh
```

Jika Anda beralih ke PowerShell, simbol pada prompt perintah diperbarui kePS>.

**Z shell**

```
zsh
```

Jika Anda beralih keZ shell, simbol pada prompt perintah diperbarui ke%.

Untuk informasi tentang versi yang telah diinstal sebelumnya di lingkungan shell Anda, lihat [tabel shell](#) di bagian [lingkungan CloudShell komputasi AWS](#).

## Langkah 3: Unduh file dari AWS CloudShell

### Note

Opsi ini tidak tersedia untuk lingkungan VPC.

Langkah ini memandu Anda melalui proses mengunduh file.

1. Untuk mengunduh file, buka Tindakan dan pilih Unduh file dari menu.

Kotak dialog Download file akan ditampilkan.

2. Di kotak dialog Unduh file, masukkan jalur untuk file yang akan diunduh.

### Note

Anda dapat menggunakan jalur absolut atau relatif saat menentukan file untuk diunduh. Dengan nama jalur relatif, `/home/cloudshell-user/` ditambahkan secara otomatis ke awal secara default. Jadi, untuk mengunduh file bernama `mydownload-file`, kedua hal berikut ini adalah jalur yang valid:

- Jalur absolut: `/home/cloudshell-user/subfolder/mydownloadfile.txt`
- Jalur relatif: `subfolder/mydownloadfile.txt`

3. Pilih Unduh.

Jika jalur file benar, kotak dialog akan ditampilkan. Anda dapat menggunakan kotak dialog ini untuk membuka file dengan aplikasi default. Atau, Anda dapat menyimpan file ke folder di mesin lokal Anda.

### Note

Opsi Unduh tidak tersedia saat Anda meluncurkan CloudShell di fileConsole Toolbar. Anda dapat mengunduh file dari CloudShell konsol atau menggunakan browser web Chrome.

## Langkah 4: Unggah file ke AWS CloudShell

### Note

Opsi ini tidak tersedia untuk lingkungan VPC.

Langkah ini menjelaskan cara mengunggah file dan kemudian memindahkannya ke direktori baru di direktori home Anda.

1. Untuk memeriksa direktori kerja Anda saat ini, pada prompt masukkan perintah berikut:

```
pwd
```

Ketika Anda menekan Enter, shell mengembalikan direktori kerja Anda saat ini (misalnya, /home/cloudshell-user).

2. Untuk mengunggah file ke direktori ini, buka Tindakan dan pilih Unggah file dari menu.

Kotak dialog Unggah file ditampilkan.

3. Pilih Telusuri.
4. Di kotak dialog File upload sistem Anda, pilih file teks yang Anda buat untuk tutorial ini (add\_prog.py) dan pilih Buka.
5. Di kotak dialog Unggah file, pilih Unggah.

Bilah kemajuan melacak unggahan. Jika unggahan berhasil, pesan mengonfirmasi bahwa add\_prog.py telah ditambahkan ke root direktori home Anda.

6. Untuk membuat direktori untuk file, masukkan perintah make directory: `mkdir mysub_dir`.
7. Untuk memindahkan file yang diunggah dari root direktori home Anda ke direktori baru, gunakan `mv` perintah:

```
mv add_prog.py mysub_dir.
```

8. Untuk mengubah direktori kerja Anda ke direktori baru, masukkan `cd mysub_dir`.

Prompt perintah diperbarui untuk menunjukkan bahwa Anda telah mengubah direktori kerja Anda.

9. Untuk melihat isi direktori saat ini `mysub_dir`, masukkan `ls` perintah.

Isi direktori kerja terdaftar. Ini termasuk file yang baru saja Anda unggah.

## Langkah 5: Hapus file dari AWS CloudShell

Langkah ini menjelaskan cara menghapus file dari AWS CloudShell.

1. Untuk menghapus file dari AWS CloudShell, gunakan perintah shell standar seperti `rm` (hapus).

```
rm my-file-for-removal
```

2. Untuk menghapus beberapa file yang memenuhi kriteria tertentu, jalankan `find` perintah.

Contoh berikut menghapus semua file yang menyertakan akhiran “.pdf” dalam nama mereka.

```
find -type f -name '*.pdf' -delete
```

### Note

Misalkan Anda berhenti menggunakan AWS CloudShell secara spesifik Wilayah AWS. Kemudian, data yang ada di penyimpanan persisten Wilayah tersebut dihapus secara otomatis setelah periode tertentu. Untuk informasi selengkapnya, lihat [Penyimpanan Persisten](#).

## Langkah 6: Buat cadangan direktori home

Langkah ini menjelaskan cara membuat cadangan direktori home.

1. Buat file cadangan

Buat folder sementara di luar direktori home.

```
HOME_BACKUP_DIR=$(mktemp --directory)
```

Anda dapat menggunakan salah satu opsi berikut untuk membuat cadangan:

- a. Buat file cadangan menggunakan `tar`

Untuk membuat file cadangan menggunakan `tar`, masukkan perintah berikut:

```
tar \
```

```
--create \  
--gzip \  
--verbose \  
--file=${HOME_BACKUP_DIR}/home.tar.gz \  
[--exclude ${HOME}/.cache] \ // Optional  
${HOME}/  
echo "Home directory backed up to this file: ${HOME_BACKUP_DIR}/home.tar.gz"
```

b. Buat file cadangan menggunakan zip

Untuk membuat file cadangan menggunakan zip, masukkan perintah berikut:

```
zip \  
--recurse-paths \  
${HOME_BACKUP_DIR}/home.zip \  
${HOME} \  
[--exclude ${HOME}/.cache/\*] // Optional  
echo "Home directory backed up to this file: ${HOME_BACKUP_DIR}/home.zip"
```

## 2. Transfer file cadangan di luar CloudShell

Anda dapat menggunakan salah satu opsi berikut untuk mentransfer file cadangan di luar CloudShell:

a. Unduh file cadangan di mesin lokal Anda

Anda dapat mengunduh file yang dibuat pada langkah sebelumnya. Untuk informasi selengkapnya tentang cara mengunduh file CloudShell, lihat [Mengunduh file dari AWS CloudShell](#).

Di kotak dialog file unduhan, masukkan jalur untuk file yang akan diunduh (misalnya, /tmp/tmp.iA99tD9L98/home.tar.gz).

b. Transfer file cadangan ke S3

Untuk menghasilkan bucket, masukkan perintah berikut:

```
aws s3 mb s3://${BUCKET_NAME}
```

Gunakan AWS CLI untuk menyalin file ke bucket S3:

```
aws s3 cp ${HOME_BACKUP_DIR}/home.tar.gz s3://${BUCKET_NAME}
```

**Note**

Biaya transfer data mungkin berlaku.

### 3. Backup langsung ke bucket S3

Untuk membuat cadangan langsung ke bucket S3, masukkan perintah berikut:

```
aws s3 cp \  
  ${HOME}/ \  
  s3://${BUCKET_NAME} \  
  --recursive \  
  [--exclude .cache/\*] // Optional
```

## Langkah 7: Mulai ulang sesi shell

Langkah ini menjelaskan cara memulai ulang sesi shell.

**Note**


Sebagai tindakan pengamanan, jika Anda tidak berinteraksi dengan shell menggunakan keyboard atau pointer untuk waktu yang lama, sesi berhenti secara otomatis. Sesi yang berjalan lama juga dihentikan secara otomatis. Untuk informasi selengkapnya, lihat [Sesi Shell](#).

#### 1. Untuk memulai ulang sesi shell, pilih Actions, Restart.

Anda diberi tahu bahwa memulai ulang AWS CloudShell menghentikan semua sesi aktif saat ini. Wilayah AWS

#### 2. Untuk mengonfirmasi, pilih Restart.


Antarmuka menampilkan pesan bahwa lingkungan CloudShell komputasi berhenti. Setelah lingkungan berhenti dan dimulai ulang, Anda dapat mulai bekerja dengan baris perintah di sesi baru.

 Note


Dalam beberapa kasus, mungkin perlu beberapa menit bagi lingkungan Anda untuk memulai kembali.

## Langkah 8: Hapus direktori home sesi shell

Langkah ini menjelaskan cara menghapus sesi shell.

 Note

Opsi ini tidak tersedia untuk lingkungan VPC. Saat Anda me-restart lingkungan VPC, direktori home nya dihapus.

 Warning


Menghapus direktori home Anda adalah tindakan yang tidak dapat diubah di mana semua data yang disimpan di direktori home Anda dihapus secara permanen. Namun, Anda mungkin ingin mempertimbangkan opsi ini dalam situasi berikut:

- Anda salah memodifikasi file dan tidak dapat mengakses lingkungan AWS CloudShell komputasi. Menghapus direktori home Anda kembali AWS CloudShell ke pengaturan default.
- Anda ingin AWS CloudShell segera menghapus semua data Anda. Jika Anda berhenti menggunakan AWS CloudShell di AWS Wilayah, penyimpanan persisten secara [otomatis dihapus pada akhir periode penyimpanan](#) kecuali Anda meluncurkan AWS CloudShell lagi di Wilayah.

Jika Anda memerlukan penyimpanan jangka panjang untuk file Anda, pertimbangkan layanan seperti Amazon S3.

1. Untuk menghapus sesi shell, pilih Tindakan, Hapus.

Anda diberi tahu bahwa menghapus direktori AWS CloudShell home akan menghapus semua data yang saat ini disimpan di lingkungan Anda. AWS CloudShell

 Note

Anda tidak dapat membatalkan tindakan ini.

2. Untuk mengonfirmasi penghapusan, masukkan hapus di bidang input teks, lalu pilih Hapus.

AWS CloudShell menghentikan semua sesi aktif saat ini Wilayah AWS. Anda dapat membuat lingkungan baru atau mengatur lingkungan CloudShell VPC.

3. Untuk membuat lingkungan baru, pilih Buka tab.
4. Untuk membuat lingkungan CloudShell VPC, pilih Buat lingkungan VPC.

Keluar dari sesi shell secara manual

Dengan baris perintah, Anda dapat meninggalkan sesi shell dan keluar menggunakan `exit` perintah. Anda kemudian dapat menekan tombol apa saja untuk menyambung kembali dan terus menggunakan AWS CloudShell.

## Langkah 9: Edit kode file Anda dan jalankan menggunakan baris perintah

Langkah ini menunjukkan cara menggunakan Vim editor pra-instal untuk bekerja dengan file. Anda kemudian menjalankan file itu sebagai program dari baris perintah.

1. Untuk mengedit file yang Anda unggah pada langkah sebelumnya, masukkan perintah berikut:

```
vim add_prog.py
```

Antarmuka shell menyegarkan untuk menampilkan editor. Vim

2. Untuk mengedit fileVim, tekan I tombol. Sekarang edit isinya sehingga program menambahkan tiga angka, bukan dua.

```
import sys
x=int(sys.argv[1])
y=int(sys.argv[2])
```

```
z=int(sys.argv[3])
sum=x+y+z
print("The sum is",sum)
```

### Note

Jika Anda menempelkan teks ke editor dan mengaktifkan [fitur Tempel Aman](#), peringatan akan ditampilkan. Teks multiline yang disalin dapat berisi skrip berbahaya. Dengan fitur Safe Paste, Anda dapat memverifikasi teks lengkap sebelum ditempelkan. Jika Anda puas bahwa teksnya aman, pilih Tempel.

3. Setelah Anda mengedit program, tekan Esc untuk masuk ke mode Vim perintah. Kemudian, masukkan `:wq` perintah untuk menyimpan file dan keluar dari editor.

### Note

Jika Anda baru mengenal mode Vim perintah, Anda mungkin awalnya merasa sulit untuk beralih antara mode perintah dan mode sisipkan. Mode perintah digunakan saat menyimpan file dan keluar dari aplikasi. Mode sisipkan digunakan saat memasukkan teks baru. Untuk masuk ke mode sisipkan, tekan `dan`, untuk masuk ke mode perintah, tekan `Esc`. Untuk informasi selengkapnya tentang Vim dan alat lain yang tersedia di AWS CloudShell, lihat [Alat pengembangan dan utilitas shell](#).

4. Pada antarmuka baris perintah utama, jalankan program berikut dan tentukan tiga angka untuk input. Sintaksnya adalah sebagai berikut.

```
python3 add_prog.py 4 5 6
```

Baris perintah menampilkan output program: `The sum is 15.`

## Langkah 10: Gunakan AWS CLI untuk menambahkan file sebagai objek di bucket Amazon S3

Pada langkah ini, Anda membuat bucket Amazon S3 dan kemudian menggunakan `PutObject` metode untuk menambahkan file kode Anda sebagai objek di bucket itu.

**Note**

Tutorial ini menunjukkan bagaimana Anda dapat menggunakannya AWS CLI AWS CloudShell untuk berinteraksi dengan layanan AWS lainnya. Dengan menggunakan metode ini, Anda tidak perlu mengunduh atau menginstal sumber daya tambahan apa pun. Selain itu, karena Anda sudah diautentikasi di dalam shell, Anda tidak perlu mengonfigurasi kredensial sebelum melakukan panggilan.

1. Untuk membuat bucket dalam yang ditentukan Wilayah AWS, masukkan perintah berikut:

```
aws s3api create-bucket --bucket insert-unique-bucket-name-here --region us-east-1
```

**Note**

Jika Anda membuat bucket di luar us-east-1 Region, tambahkan `create-bucket-configuration LocationConstraint` parameter untuk menentukan Region. Berikut ini adalah contoh sintaks.

```
$ aws s3api create-bucket --bucket my-bucket --region eu-west-1 --create-bucket-configuration LocationConstraint=eu-west-1
```

Jika panggilan berhasil, baris perintah menampilkan respons dari layanan yang mirip dengan output berikut.

```
{
  "Location": "/insert-unique-bucket-name-here"
}
```

**Note**

Jika Anda tidak mematuhi [aturan penamaan bucket](#), kesalahan berikut akan ditampilkan: Terjadi kesalahan (InvalidBucketName) saat memanggil CreateBucket operasi: Bucket yang ditentukan tidak valid.

2. Untuk mengunggah file dan menambahkan file sebagai objek ke bucket yang baru saja Anda buat, panggil PutObject metode tersebut.

```
aws s3api put-object --bucket insert-unique-bucket-name-here --key add_prog --body  
add_prog.py
```

Setelah objek diunggah ke bucket Amazon S3, baris perintah menampilkan respons dari layanan yang mirip dengan output berikut:

```
{"ETag": "\"ab123c1:w:wad4a567d8bfd9a1234ebee56\""}
```

ETag itu adalah hash dari objek yang disimpan. Anda dapat menggunakan hash ini untuk [memeriksa integritas objek yang diunggah ke Amazon S3](#).

## Topik terkait

- [Kelola AWS layanan dari CLI di CloudShell](#)
- [Menyalin beberapa file antara mesin lokal Anda dan CloudShell](#)
- [AWS CloudShell Konsep](#)
- [Menyesuaikan AWS CloudShell pengalaman](#)

# AWS CloudShell tutorial

Tutorial berikut menunjukkan cara bereksperimen, dan menguji berbagai fungsi dan integrasi saat menggunakan AWS CloudShell

Gambaran umum tutorial	Pelajari selengkapnya
Menyalin banyak file	<a href="#">the section called “Tutorial: Menyalin banyak file”</a>
Membuat presigned URLs	<a href="#">???</a>
Membangun wadah Docker di dalam AWS CloudShell dan mendorong ke Amazon ECR	<a href="#">???</a>
Menerapkan fungsi Lambda menggunakan AWS CDK	<a href="#">???</a>

## Menyalin beberapa file antara mesin lokal Anda dan CloudShell

Tutorial ini menunjukkan cara menyalin beberapa file antara mesin lokal Anda dan CloudShell.

Dengan menggunakan AWS CloudShell antarmuka, Anda dapat mengunggah atau mengunduh satu file antara mesin lokal Anda dan lingkungan shell sekaligus. Untuk menyalin beberapa file antara CloudShell dan mesin lokal Anda secara bersamaan, gunakan salah satu opsi berikut:

- Amazon S3: Gunakan bucket S3 sebagai perantara saat menyalin file antara mesin lokal Anda dan CloudShell
- File zip: Kompres beberapa file dalam satu folder zip yang dapat diunggah atau diunduh menggunakan antarmuka CloudShell

### Note

Karena CloudShell tidak mengizinkan lalu lintas internet masuk, saat ini tidak mungkin untuk menggunakan perintah seperti `scp` atau `rsync` untuk menyalin beberapa file antara mesin lokal dan lingkungan CloudShell komputasi.

## Mengunggah dan mengunduh banyak file menggunakan Amazon S3

Langkah ini menjelaskan cara mengunggah dan mengunduh banyak file menggunakan Amazon S3.

### Prasyarat

Untuk bekerja dengan bucket dan objek, Anda memerlukan kebijakan IAM yang memberikan izin untuk melakukan tindakan API Amazon S3 berikut:

- `s3:CreateBucket`
- `s3:PutObject`
- `s3:GetObject`
- `s3:ListBucket`

Untuk daftar lengkap tindakan Amazon S3, lihat [Tindakan](#) di Referensi API Layanan Penyimpanan Sederhana Amazon.

### Unggah beberapa file untuk AWS CloudShell menggunakan Amazon S3

Langkah ini menjelaskan cara mengunggah banyak file menggunakan Amazon S3.

1. Di AWS CloudShell, buat bucket S3 dengan menjalankan `s3` perintah berikut:

```
aws s3api create-bucket --bucket your-bucket-name --region us-east-1
```

Jika panggilan berhasil, baris perintah menampilkan respons dari layanan S3:

```
{
  "Location": "/your-bucket-name"
}
```

2. Unggah file dalam direktori dari mesin lokal Anda ke bucket. Pilih salah satu opsi berikut untuk mengunggah file:
  - **Konsol Manajemen AWS:** Gunakan drag-and-drop untuk mengunggah file dan folder ke ember.
  - **AWS CLI:** Dengan versi alat yang diinstal pada mesin lokal Anda, gunakan baris perintah untuk mengunggah file dan folder ke ember.

## Using the console

- Buka konsol Amazon S3 di <https://s3.console.aws.amazon.com/s3/>

(Jika Anda menggunakan AWS CloudShell, Anda seharusnya sudah masuk ke konsol.)

- Di panel navigasi kiri, pilih Bucket, lalu pilih nama bucket tempat Anda ingin mengunggah folder atau file. Anda juga dapat membuat ember pilihan Anda dengan memilih Buat ember.
- Untuk memilih file dan folder yang ingin Anda unggah, pilih Unggah. Kemudian, seret dan lepas file dan folder yang dipilih ke jendela konsol yang mencantumkan objek di bucket tujuan, atau pilih Tambahkan file, atau Tambahkan folder.

File yang Anda pilih tercantum di Unggah yang baru.

- Pilih kotak centang untuk menunjukkan file yang akan ditambahkan.
- Untuk menambahkan file yang dipilih ke bucket, pilih Unggah.

### Note

Untuk informasi tentang berbagai opsi konfigurasi saat menggunakan konsol, lihat [Bagaimana cara mengunggah file dan folder ke bucket S3?](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

## Using AWS CLI

### Note

Untuk opsi ini, Anda harus memiliki AWS CLI alat yang diinstal pada mesin lokal Anda dan memiliki kredensial Anda dikonfigurasi untuk panggilan ke AWS layanan. Untuk informasi selengkapnya, silakan lihat [Panduan Pengguna AWS Command Line Interface](#).

- Luncurkan AWS CLI alat dan jalankan `aws s3` perintah berikut untuk menyinkronkan bucket yang ditentukan dengan isi direktori saat ini di mesin lokal Anda:

```
aws s3 sync folder-path s3://your-bucket-name
```

Jika sinkronisasi berhasil, pesan upload akan ditampilkan untuk setiap objek yang ditambahkan ke bucket.

3. Kembali ke baris CloudShell perintah dan masukkan perintah berikut untuk menyinkronkan direktori di lingkungan shell dengan isi bucket S3:

```
aws s3 sync s3://your-bucket-name folder-path
```

#### Note

Anda juga dapat menambahkan `--exclude "<value>"` dan `--include "<value>"` parameter ke `sync` perintah untuk melakukan pencocokan pola untuk mengecualikan atau menyertakan file atau objek tertentu.

Untuk informasi selengkapnya, lihat [Penggunaan Kecualikan dan Sertakan Filter](#) di Referensi AWS CLI Perintah.

Jika sinkronisasi berhasil, pesan unduhan ditampilkan untuk setiap file yang diunduh dari bucket ke direktori.

#### Note

Dengan perintah sinkronisasi, hanya file baru dan yang diperbarui yang disalin secara rekursif dari direktori sumber ke tujuan.

### Unduh beberapa file dari AWS CloudShell menggunakan Amazon S3

Langkah ini menjelaskan cara mengunduh banyak file menggunakan Amazon S3.

1. Menggunakan baris AWS CloudShell perintah, masukkan `aws s3` perintah berikut untuk menyinkronkan bucket S3 dengan isi direktori saat ini di lingkungan shell:

```
aws s3 sync folder-path s3://your-bucket-name
```

**Note**

Anda juga dapat menambahkan `--exclude "<value>"` dan `--include "<value>"` parameter ke `sync` perintah untuk melakukan pencocokan pola untuk mengecualikan atau menyertakan file atau objek tertentu.

Untuk informasi selengkapnya, lihat [Penggunaan Kecualikan dan Sertakan Filter](#) di Referensi AWS CLI Perintah.

Jika sinkronisasi berhasil, pesan upload akan ditampilkan untuk setiap objek yang ditambahkan ke bucket.

2. Unduh isi ember ke mesin lokal Anda. Karena konsol Amazon S3 tidak mendukung pengunduhan beberapa objek, Anda perlu menggunakan AWS CLI alat yang diinstal pada mesin lokal Anda.

Dari baris perintah AWS CLI alat, jalankan perintah berikut:

```
aws s3 sync s3://your-bucket-name folder-path
```

Jika sinkronisasi berhasil, baris perintah menampilkan pesan unduhan untuk setiap file yang diperbarui atau ditambahkan di direktori tujuan.

**Note**

Untuk opsi ini, Anda harus memiliki AWS CLI alat yang diinstal pada mesin lokal Anda dan memiliki kredensial Anda dikonfigurasi untuk panggilan ke AWS layanan. Untuk informasi selengkapnya, silakan lihat [Panduan Pengguna AWS Command Line Interface](#).

## Mengunggah dan mengunduh banyak file menggunakan folder zip

Langkah ini menjelaskan cara mengunggah dan mengunduh banyak file menggunakan folder zip.

Dengan `zip/unzip` utilitas, Anda dapat mengompres beberapa file dalam arsip yang dapat diperlakukan sebagai satu file. Utilitas sudah diinstal sebelumnya di lingkungan CloudShell komputasi.

Untuk informasi selengkapnya tentang alat pra-instal, lihat [Alat pengembangan dan utilitas shell](#).

Unggah beberapa file untuk AWS CloudShell menggunakan folder zip

Langkah ini menjelaskan cara mengunggah beberapa file menggunakan folder zip.

1. Di komputer lokal Anda, tambahkan file yang akan diunggah ke folder zip.
2. Luncurkan CloudShell, lalu pilih Tindakan, Unggah file.
3. Dalam kotak dialog Unggah file, pilih Pilih file, lalu pilih folder zip yang baru saja Anda buat.
4. Dalam kotak dialog Unggah file, pilih Unggah untuk menambahkan file yang dipilih ke lingkungan shell.
5. Di baris CloudShell perintah, jalankan perintah berikut untuk unzip isi arsip zip ke direktori tertentu:

```
unzip zipped-files.zip -d my-unzipped-folder
```

Unduh beberapa file dari AWS CloudShell menggunakan folder zip

Langkah ini menjelaskan cara mengunduh banyak file menggunakan folder zip.

1. Di baris CloudShell perintah, jalankan perintah berikut untuk menambahkan semua file di direktori saat ini ke folder zip:

```
zip -r zipped-archive.zip *
```

2. Pilih Tindakan, Unduh file.
3. Di kotak dialog Unduh file, masukkan jalur untuk folder zip (`/home/cloudshell-user/zip-folder/zipped-archive.zip`, misalnya), lalu pilih Unduh.

Jika jalurnya benar, dialog browser menawarkan pilihan untuk membuka folder zip atau menyimpannya ke mesin lokal Anda.

4. Di mesin lokal Anda, Anda sekarang dapat membuka zip konten folder zip yang diunduh.

# Membuat URL presigned untuk objek Amazon S3 menggunakan CloudShell

Tutorial ini menunjukkan cara membuat URL presigned untuk berbagi objek Amazon S3 dengan orang lain. Karena pemilik objek menentukan kredensi keamanan mereka sendiri saat berbagi, siapa pun yang menerima URL presigned dapat mengakses objek untuk waktu yang terbatas.

## Prasyarat

- Pengguna IAM dengan izin akses yang disediakan oleh kebijakan. `AWSCloudShellFullAccess`
- Untuk izin IAM yang diperlukan untuk membuat URL yang telah ditetapkan sebelumnya, lihat [Berbagi objek dengan orang lain](#) di Panduan Pengguna Layanan Penyimpanan Sederhana Amazon.

## Langkah 1: Buat peran IAM untuk memberikan akses ke bucket Amazon S3

Langkah ini menjelaskan cara membuat peran IAM untuk memberikan akses ke bucket Amazon S3.

1. Untuk mendapatkan detail IAM Anda yang dapat dibagikan, hubungi `get-caller-identity` perintah dari AWS CloudShell.

```
aws sts get-caller-identity
```

Jika panggilan berhasil, baris perintah menampilkan respons yang mirip dengan yang berikut ini.

```
{
  "Account": "123456789012",
  "UserId": "AROAXX0ZUU0TTWDCVIDZ2:redirect_session",
  "Arn": "arn:aws:sts::531421766567:assumed-role/Feder08/redirect_session"
}
```

2. Ambil informasi pengguna yang Anda peroleh di langkah sebelumnya, dan tambahkan ke CloudFormation template. Template ini menciptakan peran IAM. Peran ini memberikan izin hak istimewa paling sedikit kepada kolaborator Anda untuk sumber daya bersama.

```
Resources:
  CollaboratorRole:
    Type: AWS::IAM::Role
```

```
Properties:
  AssumeRolePolicyDocument:
    Version: 2012-10-17
    Statement:
      - Effect: Allow
        Principal:
          AWS: "arn:aws:iam::531421766567:role/Feder08"
        Action: "sts:AssumeRole"
    Description: Role used by my collaborators
    MaxSessionDuration: 7200
  CollaboratorPolicy:
    Type: AWS::IAM::Policy
    Properties:
      PolicyDocument:
        Version: 2012-10-17
        Statement:
          - Effect: Allow
            Action:
              - 's3:*'
            Resource: 'arn:aws:s3:::<YOUR_BUCKET_FOR_FILE_TRANSFER>'
            Condition:
              StringEquals:
                s3:prefix:
                  - "myfolder/*"
      PolicyName: S3ReadSpecificFolder
    Roles:
      - !Ref CollaboratorRole
Outputs:
  CollaboratorRoleArn:
    Description: Arn for the Collaborator's Role
    Value: !GetAtt CollaboratorRole.Arn
```

3. Simpan CloudFormation template dalam file yang diberi nama `template.yaml`.
4. Gunakan template untuk menyebarkan tumpukan dan membuat peran IAM dengan memanggil perintah `deploy`

```
aws cloudformation deploy --template-file ./template.yaml --stack-name
CollaboratorRole --capabilities CAPABILITY_IAM
```

## Hasilkan URL yang telah ditentukan sebelumnya

Langkah ini menjelaskan cara menghasilkan URL yang telah ditetapkan sebelumnya.

1. Dengan menggunakan editor Anda AWS CloudShell, tambahkan kode berikut. Kode ini membuat URL yang menyediakan pengguna federasi dengan akses langsung ke file. Konsol Manajemen AWS

```
import urllib, json, sys
import requests
import boto3
import os

def main():
    sts_client = boto3.client('sts')
    assume_role_response = sts_client.assume_role(
        RoleArn=os.environ.get(ROLE_ARN),
        RoleSessionName="collaborator-session"
    )
    credentials = assume_role_response['Credentials']
    url_credentials = {}
    url_credentials['sessionId'] = credentials.get('AccessKeyId')
    url_credentials['sessionKey'] = credentials.get('SecretAccessKey')
    url_credentials['sessionToken'] = credentials.get('SessionToken')
    json_string_with_temp_credentials = json.dumps(url_credentials)
    print(f"json string {json_string_with_temp_credentials}")

    request_parameters = f"?
Action=getSignInToken&Session={urllib.parse.quote(json_string_with_temp_credentials)}"
    request_url = "https://signin.aws.amazon.com/federation" + request_parameters
    r = requests.get(request_url)
    signin_token = json.loads(r.text)
    request_parameters = "?Action=login"
    request_parameters += "&Issuer=Example.org"
    request_parameters += "&Destination=" + urllib.parse.quote("https://us-
west-2.console.aws.amazon.com/cloudshell")
    request_parameters += "&SignInToken=" + signin_token["SignInToken"]
    request_url = "https://signin.aws.amazon.com/federation" + request_parameters

    # Send final URL to stdout
    print (request_url)

if __name__ == "__main__":
```

```
main()
```

2. Simpan kode dalam file bernama `share.py`.
3. Jalankan yang berikut ini dari baris perintah untuk mengambil Amazon Resource Name (ARN) dari peran IAM dari CloudFormation. Kemudian, gunakan dalam Python skrip untuk mendapatkan kredensial keamanan sementara.

```
ROLE_ARN=$(aws cloudformation describe-stacks --stack-name CollaboratorRole --query "Stacks[*].Outputs[?OutputKey=='CollaboratorRoleArn'].OutputValue" --output text)
python3 ./share.py
```

Skrip mengembalikan URL yang dapat diklik kolaborator untuk membawanya AWS CloudShell masuk Konsol Manajemen AWS. Kolaborator memiliki kontrol penuh atas `myfolder/` folder di bucket Amazon S3 selama 3.600 detik berikutnya (1 jam). Kredensialnya kedaluwarsa setelah satu jam. Setelah waktu ini, kolaborator tidak dapat lagi mengakses bucket.

## Membangun wadah Docker di dalam CloudShell dan mendorongnya ke repositori Amazon ECR

Tutorial ini menunjukkan cara mendefinisikan dan membangun wadah Docker AWS CloudShell dan mendorongnya ke repositori Amazon ECR.

### Prasyarat

- Anda harus memiliki izin yang diperlukan untuk membuat dan mendorong ke repositori Amazon ECR. Untuk informasi selengkapnya tentang repositori dengan Amazon ECR, lihat [repositori pribadi Amazon ECR](#) di Panduan Pengguna Amazon ECR. Untuk informasi selengkapnya tentang izin yang diperlukan untuk mendorong gambar dengan Amazon ECR, lihat [izin IAM yang diperlukan untuk mendorong gambar di Panduan Pengguna](#) Amazon ECR.

### Prosedur Tutorial

Tutorial berikut menguraikan cara menggunakan CloudShell antarmuka untuk membangun wadah Docker dan mendorongnya ke repositori Amazon ECR.

1. Buat folder baru di direktori home Anda.

```
mkdir ~/docker-cli-tutorial
```

2. Arahkan ke folder yang Anda buat.

```
cd ~/docker-cli-tutorial
```

3. Buat Dockerfile kosong.

```
touch Dockerfile
```

4. Menggunakan editor teks, misalnanano Dockerfile, buka file dan tempel konten berikut ke dalamnya.

```
# Dockerfile

# Base this container on the latest Amazon Linux version
FROM public.ecr.aws/amazonlinux/amazonlinux:latest

# Install the cowsay binary
RUN dnf install --assumeyes cowsay

# Default entrypoint binary
ENTRYPOINT [ "cowsay" ]

# Default argument for the cowsay entrypoint
CMD [ "Hello, World!" ]
```

5. Dockerfile sekarang siap dibangun. Bangun wadah dengan menjalankandocker build. Tandai wadah dengan easy-to-type nama untuk digunakan dalam perintah future.

```
docker build --tag test-container .
```

Pastikan untuk memasukkan trailing period (.).

Gambar perintah docker build berjalan di dalamnya. AWS CloudShell

6. Anda sekarang dapat menguji wadah untuk memeriksa apakah penampung berjalan dengan benar AWS CloudShell.

```
docker container run test-container
```

Gambar perintah docker container run di dalamnya AWS CloudShell

7. Sekarang setelah Anda memiliki wadah Docker yang berfungsi, Anda perlu mendorongnya ke repositori Amazon ECR. Jika Anda memiliki repositori Amazon ECR yang ada, Anda dapat melewati langkah ini.

Jalankan perintah berikut untuk membuat repositori Amazon ECR untuk tutorial ini.

```
ECR_REPO_NAME=docker-tutorial-repo
aws ecr create-repository --repository-name ${ECR_REPO_NAME}
```

Gambar perintah yang digunakan untuk membuat repositori Amazon ECR di dalamnya AWS CloudShell

8. Setelah Anda membuat repositori Amazon ECR, Anda dapat mendorong wadah Docker ke sana.

Jalankan perintah berikut untuk mendapatkan kredensial masuk Amazon ECR untuk Docker.

```
AWS_ACCOUNT_ID=$(aws sts get-caller-identity --query "Account" --output text)
ECR_URL=${AWS_ACCOUNT_ID}.dkr.ecr.${AWS_REGION}.amazonaws.com
aws ecr get-login-password | docker login --username AWS --password-stdin
${ECR_URL}
```

Gambar perintah yang digunakan untuk mendapatkan kredensial masuk Amazon ECR untuk Docker.

#### Note

Jika variabel `AWS_REGION` lingkungan tidak diatur dalam Anda CloudShell atau Anda ingin berinteraksi dengan sumber daya di lain Wilayah AWS, jalankan perintah berikut:

```
AWS_REGION=<your-desired-region>
```

9. Tandai gambar dengan repositori Amazon ECR target dan kemudian dorong ke repositori itu.

```
docker tag test-container ${ECR_URL}/${ECR_REPO_NAME}
docker push ${ECR_URL}/${ECR_REPO_NAME}
```

Gambar perintah yang digunakan untuk menandai gambar dengan repositori Amazon ECR target.

Jika Anda mengalami kesalahan atau mengalami masalah saat mencoba menyelesaikan tutorial ini, lihat bagian [Pemecahan Masalah](#) dari panduan ini untuk mendapatkan bantuan.

## Bersihkan

Anda sekarang telah berhasil menerapkan wadah Docker Anda ke repositori Amazon ECR Anda. Untuk menghapus file yang Anda buat dalam tutorial ini dari AWS CloudShell lingkungan Anda, jalankan perintah berikut.

- ```
cd ~  
rm -rf ~/docker-cli-tutorial
```

- Hapus repositori Amazon ECR.

```
aws ecr delete-repository --force --repository-name ${ECR_REPO_NAME}
```

## Menerapkan fungsi Lambda menggunakan in AWS CDK CloudShell

Tutorial ini menunjukkan cara menerapkan fungsi Lambda ke akun Anda menggunakan AWS Cloud Development Kit (AWS CDK) in. CloudShell

### Prasyarat

- Bootstrap akun Anda untuk digunakan dengan file AWS CDK. Untuk informasi tentang bootstrap dengan AWS CDK, lihat [Bootstrapping di Panduan Pengembang v2.AWS CDK](#). Jika Anda belum melakukan bootstrap akun, Anda dapat menjalankannya. `cdk bootstrap` CloudShell
- Pastikan Anda memiliki izin yang sesuai untuk menyebarkan sumber daya ke akun Anda. Izin administrator direkomendasikan.

## Prosedur Tutorial

Tutorial berikut menguraikan cara menerapkan fungsi Lambda berbasis kontainer Docker menggunakan in. AWS CDK CloudShell

1. Buat folder baru di direktori home Anda.

```
mkdir ~/docker-cdk-tutorial
```

2. Arahkan ke folder yang Anda buat.

```
cd ~/docker-cdk-tutorial
```

3. Instal AWS CDK dependensi secara lokal.

```
npm install aws-cdk aws-cdk-lib
```

Gambar perintah yang digunakan untuk menginstal AWS CDK dependensi.

4. Buat AWS CDK proyek kerangka di folder yang Anda buat.

```
touch cdk.json
mkdir lib
touch lib/docker-tutorial.js lib/Dockerfile lib/hello.js
```

5. Menggunakan editor teks, misal `nano cdk.json`, buka file dan tempel konten berikut ke dalamnya.

```
{
  "app": "node lib/docker-tutorial.js"
}
```

6. Buka `lib/docker-tutorial.js` file dan tempel konten berikut ke dalamnya.

```
// this file defines the CDK constructs we want to deploy

const { App, Stack } = require('aws-cdk-lib');
const { DockerImageFunction, DockerImageCode } = require('aws-cdk-lib/aws-lambda');
const path = require('path');

// create an application
const app = new App();

// define stack
class DockerTutorialStack extends Stack {
  constructor(scope, id, props) {
    super(scope, id, props);
  }
}
```

```
// define lambda that uses a Docker container
const dockerfileDir = path.join(__dirname);
new DockerImageFunction(this, 'DockerTutorialFunction', {
  code: DockerImageCode.fromImageAsset(dockerfileDir),
  functionName: 'DockerTutorialFunction',
});
}
}

// instantiate stack
new DockerTutorialStack(app, 'DockerTutorialStack');
```

7. Buka `lib/Dockerfile` dan tempel konten berikut ke dalamnya.

```
# Use a NodeJS 20.x runtime
FROM public.ecr.aws/lambda/nodejs:20

# Copy the function code to the LAMBDA_TASK_ROOT directory
# This environment variable is provided by the lambda base image
COPY hello.js ${LAMBDA_TASK_ROOT}

# Set the CMD to the function handler
CMD [ "hello.handler" ]
```

8. Buka `lib/hello.js` file dan tempel konten berikut ke dalamnya.

```
// define the handler
exports.handler = async (event) => {
  // simply return a friendly success response
  const response = {
    statusCode: 200,
    body: JSON.stringify('Hello, World!'),
  };
  return response;
};
```

9. Gunakan AWS CDK CLI untuk mensintesis proyek dan menyebarkan sumber daya. Anda harus bootstrap akun Anda.

```
npx cdk synth
npx cdk deploy --require-approval never
```

Gambar perintah untuk menggunakan AWS CDK CLI untuk mensintesis proyek dan menyebarkan sumber daya.

10. Panggil fungsi Lambda untuk mengonfirmasi dan memverifikasinya.

```
aws lambda invoke --function-name DockerTutorialFunction out.json  
jq . out.json
```

Gambar perintah yang digunakan untuk memanggil fungsi Lambda.

Anda sekarang telah berhasil menerapkan fungsi Lambda berbasis kontainer Docker menggunakan AWS CDK. Untuk informasi selengkapnya AWS CDK, lihat [Panduan Pengembang AWS CDK v2](#). Jika Anda mengalami kesalahan atau mengalami masalah saat mencoba menyelesaikan tutorial ini, lihat bagian [Pemecahan Masalah](#) dari panduan ini untuk mendapatkan bantuan.

## Bersihkan

Anda sekarang telah berhasil menerapkan fungsi Lambda berbasis kontainer Docker menggunakan AWS CDK. Di dalam AWS CDK proyek, jalankan perintah berikut untuk menghapus sumber daya terkait. Anda akan diminta untuk mengkonfirmasi penghapusan.

- ```
npx cdk destroy DockerTutorialStack
```
- Untuk menghapus file dan sumber daya yang Anda buat dalam tutorial ini dari AWS CloudShell lingkungan Anda, jalankan perintah berikut.

```
cd ~  
rm -rf ~/docker-cli-tutorial
```

# AWS CloudShell Konsep

Bagian ini menjelaskan cara berinteraksi dengan AWS CloudShell dan melakukan tindakan spesifik dengan aplikasi yang didukung.

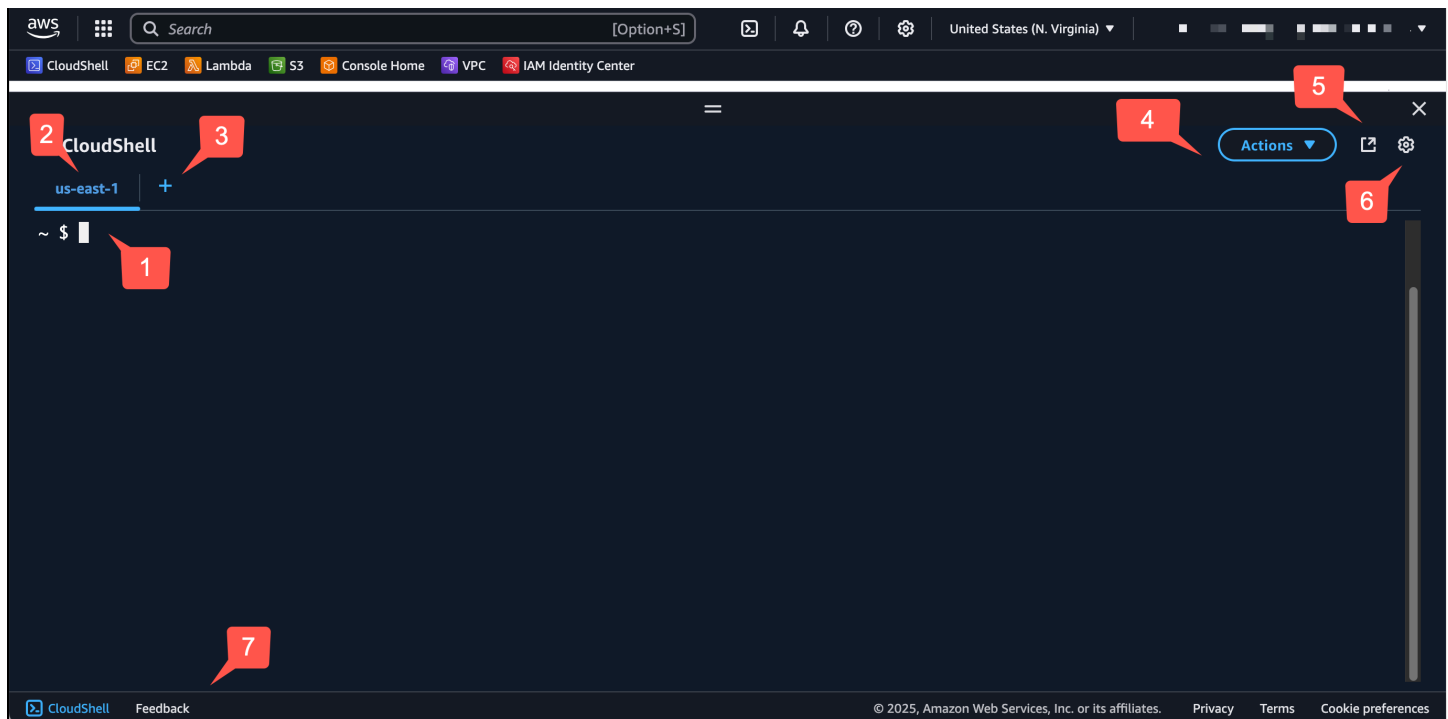
Topik

- [Menavigasi antarmuka AWS CloudShell](#)
- [Bekerja di Wilayah AWS](#)
- [Bekerja dengan file dan penyimpanan](#)
- [Akses CloudShell di Aplikasi Mobile Console](#)
- [Bekerja dengan Docker](#)


## Menavigasi antarmuka AWS CloudShell

Anda dapat menavigasi fitur CloudShell antarmuka dari Konsol Manajemen AWS dan Console Toolbar.

Tangkapan layar berikut menunjukkan beberapa fitur AWS CloudShell antarmuka utama.




1. AWS CloudShell antarmuka baris perintah yang Anda gunakan untuk menjalankan perintah dengan menggunakan [shell pilihan Anda](#). Jenis shell saat ini ditunjukkan oleh command prompt.
2. Tab terminal, yang menggunakan Wilayah AWS where saat AWS CloudShell ini berjalan.
3. Ikon + adalah menu tarik-turun yang mencakup opsi untuk membuat, memulai ulang, dan menghapus lingkungan.
4. Menu Tindakan, yang menyediakan opsi untuk [mengubah tata letak layar](#), [mengunduh](#) dan [mengunggah](#) file, [memulai ulang AWS CloudShell](#), dan [menghapus direktori home Anda](#). AWS CloudShell

 Note

Opsi Unduh tidak tersedia saat Anda meluncurkan CloudShell di fileConsole Toolbar.

5. Buka di tab browser baru, yang menyediakan opsi untuk mengakses CloudShell sesi Anda dalam layar penuh.
6. Opsi Preferensi, yang dapat Anda gunakan untuk [menyesuaikan pengalaman shell Anda](#).
7. Bilah bawah, yang menyediakan opsi berikut untuk:
  - Luncurkan CloudShell dari CloudShellikon.
  - Berikan umpan balik dari ikon Umpan Balik. Pilih jenis umpan balik yang ingin Anda kirimkan, tambahkan komentar Anda, lalu pilih Kirim.
  - Untuk mengirimkan umpan balik CloudShell, pilih salah satu opsi berikut:
    - Dari konsol, luncurkan CloudShell, dan pilih Umpan Balik. Tambahkan komentar Anda, lalu pilih Kirim.
    - Pilih CloudShellpadaConsole Toolbar, di kiri bawah konsol, dan kemudian pilih Buka di ikon tab browser baru, Umpan balik. Tambahkan komentar Anda, lalu pilih Kirim.

 Note

Opsi Umpan Balik tidak tersedia saat Anda meluncurkan CloudShell diConsole Toolbar.

- Pelajari tentang kebijakan privasi dan ketentuan penggunaan kami, dan sesuaikan preferensi cookie.

# Bekerja di Wilayah AWS

Arus Wilayah AWS yang Anda jalankan ditampilkan sebagai tab.

Anda dapat memilih Wilayah AWS untuk bekerja dengan memilih Wilayah tertentu menggunakan pemilih Wilayah. Setelah Anda mengubah Regions, antarmuka akan diperbarui saat sesi shell Anda terhubung ke lingkungan komputasi berbeda yang berjalan di Wilayah yang dipilih.

## Important

- Anda dapat menggunakan hingga 1 GB penyimpanan persisten di masing-masing Wilayah AWS. Penyimpanan persisten disimpan di direktori home Anda (\$HOME). Ini berarti bahwa setiap file pribadi, direktori, program, atau skrip yang disimpan di direktori home Anda semuanya terletak dalam satu. Wilayah AWS Selain itu, mereka berbeda dari yang terletak di direktori home dan menyimpan Wilayah yang berbeda.

Retensi jangka panjang file dalam penyimpanan persisten juga dikelola berdasarkan Perwilayah. Untuk informasi selengkapnya, lihat [Penyimpanan tetap](#).

- Penyimpanan persisten tidak tersedia untuk lingkungan AWS CloudShell VPC.

## Menentukan default Wilayah AWS Anda untuk AWS CLI

Anda dapat menggunakan [variabel lingkungan](#) untuk menentukan opsi konfigurasi dan kredensial yang diperlukan untuk mengakses Layanan AWS menggunakan. AWS CLI Variabel lingkungan yang menentukan default Wilayah AWS untuk sesi shell Anda diatur baik ketika Anda memulai AWS CloudShell dari Wilayah tertentu di Konsol Manajemen AWS atau ketika Anda memilih opsi di pemilih Region.

[Variabel lingkungan lebih diutamakan daripada file AWS CLI kredensial yang diperbarui oleh file.](#)

`aws configure` Jadi, Anda tidak dapat menjalankan `aws configure` perintah untuk mengubah Wilayah yang ditentukan oleh variabel lingkungan. Sebagai gantinya, untuk mengubah Region default untuk AWS CLI perintah, tetapkan nilai ke variabel `AWS_REGION` lingkungan. Dalam contoh berikut, ganti `us-east-1` dengan Wilayah tempat Anda berada.

Bash or Zsh

```
$ export AWS_REGION=us-east-1
```

Menyetel variabel lingkungan mengubah nilai yang digunakan hingga di akhir sesi shell Anda atau saat Anda menyetel variabel ke nilai yang berbeda. Anda dapat mengatur variabel dalam skrip startup shell Anda untuk membuat variabel persisten di seluruh sesi masa depan.

## PowerShell

```
PS C:\> $Env:AWS_REGION="us-east-1"
```

Jika Anda menetapkan variabel lingkungan pada PowerShell prompt, variabel lingkungan menyimpan nilai hanya untuk durasi sesi saat ini. Atau, Anda dapat mengatur variabel untuk semua PowerShell sesi future dengan menambahkan variabel ke PowerShell profil Anda. Untuk informasi selengkapnya tentang menyimpan variabel lingkungan, lihat [PowerShell dokumentasi](#).

Untuk mengonfirmasi bahwa Anda telah mengubah Region default, jalankan `aws configure list` perintah untuk menampilkan data AWS CLI konfigurasi saat ini.

### Note

Untuk AWS CLI perintah tertentu, Anda dapat mengganti Region default menggunakan opsi `--region` baris perintah. Untuk informasi selengkapnya, lihat [Opsis baris perintah](#) di Panduan AWS Command Line Interface Pengguna.

## Bekerja dengan file dan penyimpanan

Menggunakan AWS CloudShell antarmuka, Anda dapat mengunggah file ke dan mengunduh file dari lingkungan shell. Untuk informasi selengkapnya tentang mengunduh dan mengunggah file, lihat [Memulai AWS CloudShell](#).

Untuk memastikan file apa pun yang Anda tambahkan tersedia setelah sesi Anda berakhir, Anda harus mengetahui perbedaan antara penyimpanan persisten dan sementara.

- Penyimpanan persisten: Anda memiliki 1 GB penyimpanan persisten untuk masing-masing Wilayah AWS. Penyimpanan persisten ada di direktori home Anda.
- Penyimpanan sementara: Penyimpanan sementara didaur ulang pada akhir sesi. Penyimpanan sementara ada di direktori yang berada di luar direktori home Anda.

### Important

Pastikan untuk meninggalkan file yang ingin Anda simpan dan gunakan untuk sesi shell future di direktori home Anda. Misalnya, Anda memindahkan file dari direktori home Anda dengan menjalankan `mv` perintah. Kemudian, file itu didaur ulang ketika sesi shell saat ini berakhir.

## Akses CloudShell di Aplikasi Mobile Console

Anda dapat mengakses CloudShell AWS Console Mobile Application dari layar beranda. Dari layar beranda, Anda dapat melihat informasi tentang CloudShell dan AWS layanan lainnya. Untuk informasi selengkapnya, lihat [Memulai dengan AWS Console Mobile Application](#). Untuk meluncurkan CloudShell di AWS Console Mobile Application, pilih salah satu opsi berikut:

- Pilih CloudShell ikon di bagian bawah bilah navigasi.
- Pilih CloudShell pada menu Layanan.

Anda dapat keluar CloudShell kapan saja dengan memilih X.

Untuk informasi selengkapnya tentang mengakses CloudShell di Console Mobile Application, lihat [Akses AWS CloudShell](#).

### Note

Saat ini, Anda tidak dapat membuat atau meluncurkan lingkungan VPC di. AWS Console Mobile Application

## Bekerja dengan Docker

AWS CloudShell sepenuhnya mendukung Docker tanpa instalasi atau konfigurasi. Anda dapat menentukan, membangun, dan menjalankan kontainer Docker di dalamnya AWS CloudShell. Anda dapat menerapkan sumber daya berbasis Docker, seperti fungsi Lambda berdasarkan kontainer Docker, melalui AWS CDK Toolkit serta membangun kontainer Docker dan mendorongnya ke repositori Amazon ECR melalui CLI Docker. Untuk langkah-langkah mendetail tentang cara menjalankan kedua penerapan ini, lihat tutorial berikut:

- [Tutorial: Menyebarkan fungsi Lambda menggunakan AWS CDK](#)
- [Tutorial: Membangun wadah Docker di dalam AWS CloudShell dan mendorongnya ke repositori Amazon ECR](#)

Ada batasan dan batasan tertentu dengan menggunakan Docker dengan AWS CloudShell:

- Docker memiliki ruang terbatas di suatu lingkungan. Jika Anda memiliki gambar individual yang besar, atau terlalu banyak gambar Docker yang sudah ada sebelumnya, ini dapat menyebabkan masalah yang mungkin mencegah Anda menarik, membangun, atau menjalankan gambar tambahan. Untuk informasi selengkapnya tentang Docker, lihat panduan [Dokumentasi Docker](#).
- Docker tersedia di semua AWS Wilayah, kecuali Wilayah AWS GovCloud (AS). Untuk daftar Wilayah di mana Docker tersedia, lihat [AWS Wilayah yang Didukung untuk AWS CloudShell](#).
- Jika Anda mengalami masalah saat menggunakan Docker AWS CloudShell, lihat bagian [Pemecahan Masalah](#) di panduan ini untuk informasi tentang cara mengatasi masalah ini secara potensial.

# Fitur aksesibilitas untuk AWS CloudShell

Topik ini menjelaskan cara menggunakan fitur aksesibilitas untuk CloudShell. Anda dapat menggunakan keyboard untuk menavigasi melalui elemen yang dapat difokuskan pada halaman. Anda juga dapat menyesuaikan tampilan CloudShell, termasuk ukuran font dan tema antarmuka.

## Navigasi keyboard di CloudShell

Untuk menavigasi melalui elemen yang dapat difokuskan pada halaman, tekan. Tab

## CloudShell fitur aksesibilitas terminal

Anda dapat menggunakan Tab tombol dalam mode berikut:

- Mode terminal (Default) — Dalam mode ini, terminal menangkap entri Tab kunci Anda. Setelah fokus pada terminal, tekan Tab untuk mengakses hanya fungsionalitas terminal.
- Mode navigasi — Dalam mode ini, terminal tidak menangkap entri Tab kunci Anda. Tekan Tab untuk menavigasi melalui elemen yang dapat difokuskan pada halaman.

Untuk beralih antara mode terminal dan mode navigasi, tekan Ctrl +M. Setelah Anda beralih kembali, Tab: navigasi muncul di header, dan Anda dapat menggunakan Tab tombol untuk menavigasi halaman.

Untuk kembali ke mode terminal, tekan Ctrl +M. Atau, pilih X di sebelah Tab: navigasi.

### Note

Saat ini, fitur aksesibilitas CloudShell terminal tidak tersedia di perangkat seluler.

## Memilih ukuran font dan tema antarmuka di CloudShell

Anda dapat menyesuaikan tampilan CloudShell untuk mengakomodasi preferensi visual Anda.

- Ukuran font - Pilih dari ukuran font Terkecil, Kecil, Sedang, Besar, dan Terbesar di terminal. Untuk informasi selengkapnya tentang mengubah ukuran font, lihat [the section called “Mengubah ukuran font”](#).

- Tema - Pilih antara tema antarmuka Terang dan Gelap. Untuk informasi selengkapnya tentang mengubah tema antarmuka, lihat [the section called “Mengubah tema antarmuka”](#).

# Kelola AWS layanan dari CLI di CloudShell

Manfaat utama AWS CloudShell adalah Anda dapat menggunakannya untuk mengelola AWS layanan Anda dari antarmuka baris perintah. Ini berarti Anda tidak perlu mengunduh dan menginstal alat atau mengonfigurasi kredensial Anda secara lokal sebelumnya. Saat Anda meluncurkan AWS CloudShell, lingkungan komputasi dibuat yang memiliki alat baris AWS perintah berikut yang sudah diinstal:

- [AWS CLI](#)
- [AWS Elastic Beanstalk CLI](#)
- [Amazon ECS CLI](#)
- [AWS SAM](#)

Dan karena Anda sudah masuk AWS, tidak ada persyaratan untuk mengonfigurasi kredensial Anda secara lokal sebelum menggunakan layanan. Kredensial yang Anda gunakan untuk masuk diteruskan ke Konsol Manajemen AWS . AWS CloudShell

Jika Anda ingin mengubah AWS Wilayah default yang digunakan AWS CLI, Anda dapat mengubah nilai yang ditetapkan ke variabel `AWS_REGION` lingkungan. (Untuk informasi selengkapnya, lihat [Menentukan default Wilayah AWS Anda untuk AWS CLI.](#))

Sisa topik ini menunjukkan bagaimana Anda dapat mulai menggunakan AWS CloudShell untuk berinteraksi dengan AWS layanan yang dipilih dari baris perintah.

## AWS CLI contoh baris perintah untuk AWS layanan yang dipilih

Contoh berikut hanya mewakili beberapa dari banyak AWS layanan yang dapat Anda gunakan menggunakan perintah yang tersedia dari AWS CLI Versi 2. Untuk daftar lengkap, lihat [Referensi Perintah AWS CLI.](#)

- [DynamoDB](#)
- [Amazon EC2](#)
- [Amazon Glacier](#)

## DynamoDB

DynamoDB adalah layanan basis data NoSQL terkelola penuh yang memberikan performa yang cepat dan dapat diprediksi dengan skalabilitas sempurna. Implementasi layanan ini dari mode NoSQL mendukung nilai kunci dan struktur data dokumen.

`create-table`Perintah berikut membuat tabel gaya NoSQL yang dinamai `MusicCollection` di akun Anda. AWS

```
aws dynamodb create-table \  
  --table-name MusicCollection \  
  --attribute-definitions AttributeName=Artist,AttributeType=S \  
  AttributeName=SongTitle,AttributeType=S \  
  --key-schema AttributeName=Artist,KeyType=HASH \  
  AttributeName=SongTitle,KeyType=RANGE \  
  --provisioned-throughput ReadCapacityUnits=5,WriteCapacityUnits=5 \  
  --tags Key=Owner,Value=blueTeam
```

Untuk informasi selengkapnya, lihat [Menggunakan DynamoDB dengan AWS CLI di AWS Command Line Interface Panduan Pengguna](#).

## Amazon EC2

Amazon Elastic Compute Cloud (Amazon EC2) adalah layanan web yang menyediakan kapasitas komputasi yang aman dan dapat diubah ukurannya di cloud. Ini dirancang untuk membuat komputasi awan skala web lebih mudah dan lebih mudah diakses.

`run-instances`Perintah berikut meluncurkan instance `t2.micro` di subnet yang ditentukan dari VPC:

```
aws ec2 run-instances --image-id ami-xxxxxxx --count 1 --instance-type t2.micro --key-name MyKeyPair --security-group-ids sg-903004f8 --subnet-id subnet-6e7f829e
```

Untuk informasi selengkapnya, lihat [Menggunakan Amazon EC2 dengan AWS CLI di AWS Command Line Interface Panduan Pengguna](#).

## Amazon Glacier

Amazon Glacier dan Amazon Glacier Deep Archive adalah kelas penyimpanan cloud Amazon S3 yang aman, tahan lama, dan sangat murah untuk pengarsipan data dan pencadangan jangka panjang.

`create-vault`Perintah berikut membuat vault—wadah untuk menyimpan arsip:

```
aws glacier create-vault --vault-name my-vault --account-id -
```

Untuk informasi selengkapnya, lihat [Menggunakan Amazon Glacier dengan](#) di AWS Command Line Interface Panduan AWS CLI Pengguna.

## AWS CLI Elastic Beanstalk

AWS Elastic Beanstalk CLI menyediakan antarmuka baris perintah yang dibuat untuk menyederhanakan pembuatan, pembaruan, dan pemantauan lingkungan dari repositori lokal. Dalam konteks ini, lingkungan mengacu pada kumpulan AWS sumber daya yang menjalankan versi aplikasi.

`create`Perintah berikut membuat lingkungan baru di Amazon Virtual Private Cloud (VPC) kustom.

```
$ eb create dev-vpc --vpc.id vpc-0ce8dd99 --vpc.elbsubnets subnet-  
b356d7c6,subnet-02f74b0c --vpc.ec2subnets subnet-0bb7f0cd,subnet-3b6697c1 --  
vpc.securitygroup sg-70cff265
```

Untuk informasi selengkapnya, lihat referensi [perintah EB CLI](#) di Panduan AWS Elastic Beanstalk Pengembang.

## Amazon ECS CLI

Antarmuka baris perintah Amazon Elastic Container Service (Amazon ECS) (CLI) menyediakan beberapa perintah tingkat tinggi. Ini dirancang untuk menyederhanakan proses pembuatan, pembaruan, dan pemantauan cluster dan tugas dari lingkungan pembangunan lokal. (Cluster Amazon ECS adalah pengelompokan tugas atau layanan yang logis.)

`configure`Perintah berikut mengkonfigurasi Amazon ECS CLI untuk membuat konfigurasi cluster bernama `ecs-cli-demo`. Konfigurasi cluster ini digunakan FARGATE sebagai tipe peluncuran default untuk `ecs-cli-demo` cluster di `us-east-1` region.

```
ecs-cli configure --region us-east-1 --cluster ecs-cli-demo --default-launch-type  
FARGATE --config-name ecs-cli-demo
```

Untuk informasi selengkapnya, lihat [Referensi Baris Perintah Amazon ECS](#) di Panduan Pengembang Layanan Kontainer Elastis Amazon.

# AWS SAM CLI

AWS SAM CLI adalah alat baris perintah yang beroperasi pada AWS Serverless Application Model template dan kode aplikasi. Anda dapat melakukan beberapa tugas menggunakannya. Ini termasuk menjalankan fungsi Lambda secara lokal, membuat paket penerapan untuk aplikasi tanpa server Anda, dan menerapkan aplikasi tanpa server Anda ke Cloud. AWS

`init` Perintah berikut menginisialisasi proyek SAM baru dengan parameter yang diperlukan diteruskan sebagai parameter:

```
sam init --runtime python3.9 --dependency-manager pip --app-template hello-world --name sam-app
```

Untuk informasi selengkapnya, lihat [referensi perintah AWS SAM CLI di Panduan AWS Serverless Application Model](#) Pengembang.

## Menggunakan Kiro CLI di CloudShell

Kiro CLI adalah antarmuka baris perintah yang memungkinkan Anda berinteraksi dengan Kiro. Untuk informasi lebih lanjut, lihat [Fitur Inti Kiro CLI](#) di Panduan Pengguna Kiro.

Kiro CLI CloudShell in memungkinkan Anda untuk berinteraksi dalam percakapan bahasa alami, mengajukan pertanyaan, dan menerima tanggapan dari Kiro semua dari terminal Anda. Anda bisa mendapatkan perintah shell terkait yang mengurangi kebutuhan untuk mencari, mengingat sintaks, dan menerima saran perintah saat Anda mengetik di terminal.

Jika Anda tidak melihat fitur Kiro CLI CloudShell di, hubungi administrator Anda untuk memberi Anda izin IAM. Untuk informasi selengkapnya, lihat [contoh kebijakan berbasis identitas untuk Pengembang Kiro di Panduan](#) Pengguna Kiro.

Bab ini menjelaskan bagaimana Anda dapat menggunakan fitur Kiro CLI di. CloudShell

Beberapa fitur CLI Kiro memerlukan otentikasi. Untuk informasi selengkapnya, lihat [Otentikasi](#) di Panduan Pengguna Kiro.

## Menggunakan perintah obrolan Kiro di CloudShell

`kiro-cli` Perintah ini memungkinkan Anda untuk mengajukan pertanyaan dan menerima tanggapan dari Kiro semua dari terminal Anda. Untuk memulai percakapan dengan Kiro, jalankan `kiro-cli` perintah di terminal. CloudShell Untuk informasi lebih lanjut, lihat [Mengobrol dengan Kiro di CLI](#) di Panduan Pengguna Kiro.

## Menggunakan perintah terjemahan Kiro di CloudShell

`kiro-cli translate` Perintah ini memungkinkan Anda untuk menulis instruksi bahasa alami. Untuk menerjemahkan dengan Kiro, jalankan `kiro-cli translate` perintah di CloudShell terminal. Untuk informasi selengkapnya, lihat [Menerjemahkan dari bahasa alami ke bash](#) di Panduan Pengguna Kiro.

## Penyelesaian perintah CLI di CloudShell

Penyelesaian CLI di CloudShell memberikan saran untuk perintah dan opsi saat Anda mengetik di terminal. Untuk informasi selengkapnya, lihat [Menghasilkan penyelesaian baris perintah](#) di Panduan Pengguna Kiro.

## Menggunakan saran sebaris Kiro di CloudShell

Saran sebaris Kiro CloudShell memberi Anda saran perintah saat Anda mengetik di terminal. Untuk informasi lebih lanjut, lihat [Kiro inline pada baris perintah di Panduan Pengguna Kiro](#).

Untuk menggunakan saran sebaris Kiro di CloudShell

1. Dari Konsol Manajemen AWS, Pilih CloudShell.
2. Pada CloudShell terminal, beralih ke Z shell, dan mulai mengetik. Untuk beralih ke shell Z, `zsh` ketik terminal, lalu tekan Enter.

### Note

Saat ini, Kiro inline hanya didukung di Z shell.

Ketika Anda mulai mengetik perintah Anda, Kiro akan membuat saran berdasarkan input Anda saat ini dan perintah sebelumnya. Saran sebaris diaktifkan secara otomatis.

Untuk menonaktifkan saran sebaris, jalankan perintah berikut:

```
kiro-cli inline disable
```

Untuk mengaktifkan saran sebaris, jalankan perintah berikut:

```
kiro-cli inline enable
```

## Kebijakan berbasis identitas untuk Kiro CLI di CloudShell

Untuk menggunakan Kiro CLI CloudShell di, pastikan Anda memiliki izin IAM yang diperlukan.

Untuk informasi selengkapnya, lihat [contoh kebijakan berbasis identitas untuk Pengembang Kiro di Panduan Pengguna Kiro](#).

# Menjalankan perintah CloudShell dari AWS konsol Layanan

Anda dapat menjalankan perintah di CloudShell terminal melalui [Amazon ElastiCache dan Amazon DocumentDB \(dengan kompatibilitas MongoDB\)](#) konsol di. Konsol Manajemen AWS

Untuk menjalankan perintah CloudShell dari konsol AWS Layanan lain, kebijakan IAM yang ditetapkan ke peran Anda harus menyertakan `cloudshell:approveCommand` izin.

CloudShell terbuka di Console Toolbar dan Run command pop-up muncul di. CloudShell Pada pop-up perintah Run, perintah muncul di kotak perintah.

Untuk menjalankan perintah di CloudShell terminal, pilih salah satu langkah berikut:

1. Masukkan nama di kotak Nama lingkungan baru jika Anda belum membuat lingkungan VPC di. CloudShell

Anda dapat melihat detail lingkungan VPC yang didasarkan pada detail VPC sumber daya Anda.

- a. Pilih Buat dan jalankan.

Langkah ini akan membuat lingkungan CloudShell VPC baru dan menjalankan perintah di terminal. CloudShell

2. Anda dapat melihat nama CloudShell lingkungan jika Anda telah membuat lingkungan CloudShell VPC.

## Note

Jika Anda sudah memiliki lingkungan CloudShell VPC, Anda tidak dapat membuat lingkungan VPC baru.

- a. Pilih Jalankan.

Langkah ini akan menjalankan perintah di CloudShell terminal di lingkungan CloudShell VPC yang dipilih.

## Note

Jika Anda tidak memiliki izin untuk melihat lingkungan VPC yang dibuat, hubungi administrator Anda untuk menambahkan izin. `cloudshell:describeEnvironments`

Untuk informasi selengkapnya, lihat [Mengelola AWS CloudShell akses dan penggunaan dengan kebijakan IAM](#).

Anda dapat terus menjalankan perintah di CloudShell terminal.

# Menyesuaikan AWS CloudShell pengalaman

Anda dapat menyesuaikan aspek-aspek berikut dari AWS CloudShell pengalaman Anda:

- [Tata letak tab](#): Pisahkan antarmuka baris perintah menjadi beberapa kolom dan baris.
- [Ukuran font](#): Sesuaikan ukuran teks baris perintah.
- [Tema warna](#): Beralih antara tema terang dan gelap.
- [Tempel Aman](#): Aktifkan atau nonaktifkan fitur yang mengharuskan Anda memverifikasi teks multiline sebelum ditempelkan.
- [Tmux ke pemulihan sesi: Menggunakan tmux mengembalikan](#) sesi Anda hingga sesi menjadi tidak aktif.

Anda juga dapat memperluas lingkungan shell Anda dengan [menginstal perangkat lunak Anda sendiri](#) dan [memodifikasi shell Anda dengan skrip](#).

## Memisahkan tampilan baris perintah menjadi beberapa tab

Jalankan beberapa perintah dengan memisahkan antarmuka baris perintah Anda menjadi beberapa panel.

### Note

Setelah membuka beberapa tab, Anda dapat memilih salah satu yang ingin Anda kerjakan dengan mengklik di mana saja di panel pilihan Anda. Anda dapat menutup tab dengan memilih simbol x, yang berada di sebelah nama Wilayah.

- Pilih Tindakan dan salah satu opsi berikut dari tata letak Tab:
  - Tab baru: Tambahkan tab baru yang berada di sebelah tab yang sedang aktif.
  - Dibagi menjadi baris: Tambahkan tab baru dalam satu baris yang berada di bawah tab yang sedang aktif.
  - Dibagi menjadi kolom: Tambahkan tab baru di kolom yang berada di sebelah kolom yang sedang aktif.

Jika tidak ada cukup ruang untuk menampilkan setiap tab sepenuhnya, gulir untuk melihat seluruh tab. Anda juga dapat memilih bilah terpisah yang memisahkan panel dan menyeretnya dengan menggunakan penunjuk untuk menambah atau mengurangi ukuran panel.

## Mengubah ukuran font

Menambah atau mengurangi ukuran teks yang ditampilkan di antarmuka baris perintah.

1. Untuk mengubah pengaturan AWS CloudShell terminal, buka Pengaturan, Preferensi.
2. Pilih ukuran teks. Pilihan Anda adalah Terkecil, Kecil, Sedang, Besar, dan Terbesar.

## Mengubah tema antarmuka

Beralih antara tema terang dan gelap untuk antarmuka baris perintah.

1. Untuk mengubah AWS CloudShell tema, buka Pengaturan, Preferensi.
2. Pilih Terang atau Gelap.

## Menggunakan Safe Paste untuk teks multiline

Safe Paste adalah fitur keamanan yang meminta Anda untuk memverifikasi bahwa teks multiline yang akan Anda tempel ke dalam shell tidak mengandung skrip berbahaya. Teks yang disalin dari situs pihak ketiga dapat berisi kode tersembunyi yang memicu perilaku tak terduga di lingkungan shell Anda.

Dialog Safe Paste menampilkan teks lengkap yang Anda salin ke clipboard Anda. Jika Anda puas bahwa tidak ada risiko keamanan, pilih Tempel.

**Warning: Pasting multiline text into AWS CloudShell**

Text that's copied from external sources can contain malicious scripts. Verify the text below before pasting.

```
import sys
x=int(sys.argv[1])
y=int(sys.argv[2])
z=int(sys.argv[3])
total=x+y+z
print("The total is",total)
```

Always ask before pasting multiline code

Cancel

Paste

Kami menyarankan Anda mengaktifkan Safe Paste untuk menangkap potensi risiko keamanan dalam skrip. Anda dapat mengaktifkan atau menonaktifkan fitur ini dengan memilih Preferensi, Aktifkan Tempel Aman, dan Nonaktifkan Tempel Aman.

## Penggunaan tmux ke pemulihan sesi

AWS CloudShell menggunakan tmux untuk memulihkan sesi di satu atau beberapa tab browser. Jika Anda me-refresh tab browser, itu melanjutkan sesi Anda sampai sesi menjadi tidak aktif. Untuk informasi selengkapnya, lihat [Pemulihan sesi](#).

# Penggunaan AWS CloudShell di Amazon VPC

AWS CloudShell Virtual Private Cloud (VPC) memungkinkan Anda untuk menciptakan CloudShell lingkungan di VPC Anda. Untuk setiap lingkungan VPC, Anda dapat menetapkan VPC, menambahkan subnet, dan mengaitkan hingga lima grup keamanan. AWS CloudShell mewarisi konfigurasi jaringan VPC dan memungkinkan Anda untuk AWS CloudShell menggunakan dengan aman dalam subnet yang sama dengan sumber daya lain di VPC dan terhubung ke mereka.

Dengan Amazon VPC, Anda dapat meluncurkan AWS sumber daya di jaringan virtual yang terisolasi secara logis yang telah Anda tentukan. Jaringan virtual ini sangat mirip dengan jaringan konvensional yang akan Anda operasikan di pusat data Anda sendiri dengan manfaatnya, yaitu menggunakan infrastruktur AWS yang dapat diskalakan. Untuk informasi selengkapnya tentang VPC, lihat [Amazon Virtual Private Cloud](#).

## Kendala operasi

AWS CloudShell Lingkungan VPC memiliki kendala berikut:

- Anda dapat membuat maksimal dua lingkungan VPC per prinsipal IAM.
- Anda dapat menetapkan maksimal lima grup keamanan untuk lingkungan VPC.
- Anda tidak dapat menggunakan opsi CloudShell unggah dan unduh di menu Tindakan untuk lingkungan VPC.

### Note

Dimungkinkan untuk mengunggah atau mengunduh file dari lingkungan VPC yang memiliki akses ke internet ingress/egress melalui alat CLI lainnya.

- Lingkungan VPC tidak mendukung penyimpanan persisten. Penyimpanan bersifat fana. Data dan direktori home dihapus ketika sesi lingkungan aktif berakhir.
- AWS CloudShell Lingkungan Anda hanya dapat terhubung ke internet jika berada di subnet VPC pribadi.

### Note

Alamat IP publik tidak dialokasikan ke lingkungan CloudShell VPC secara default. Lingkungan VPC yang dibuat dalam subnet publik dengan tabel routing yang dikonfigurasi

untuk merutekan semua lalu lintas ke Internet Gateway tidak akan memiliki akses ke internet publik, tetapi subnet pribadi yang dikonfigurasi dengan Network Address Translation (NAT) memiliki akses ke internet publik. Lingkungan VPC yang dibuat dalam subnet pribadi tersebut akan memiliki akses ke internet publik.

- AWS CloudShell Lingkungan Anda tidak akan dikonfigurasi secara default untuk menggunakan zona DNS lokal di VPC Anda.
- Untuk menyediakan CloudShell lingkungan terkelola untuk akun Anda, AWS mungkin menyediakan akses jaringan ke layanan berikut untuk host komputasi yang mendasarinya:
  - Amazon S3
  - Titik akhir VPC
    - com.amazonaws. <region>.ssmmessages
    - com.amazonaws. <region>.log
    - com.amazonaws. <region>.kms
    - com.amazonaws. <region>.mengeksekusi api
    - com.amazonaws. <region>.ecs-telemetry
    - com.amazonaws. <region>.ecs-agen
    - com.amazonaws. <region>.ecs
    - com.amazonaws. <region>.ecr.dkr
    - com.amazonaws. <region>.ecr.api
    - com.amazonaws. <region>.codecatalyst.packages
    - com.amazonaws. <region>.codecatalyst.git
    - aws.api.global.codecatalyst

Anda tidak dapat membatasi akses ke titik akhir ini dengan memodifikasi konfigurasi VPC Anda.

CloudShell VPC tersedia di semua AWS Wilayah dan GovCloud Wilayah. Untuk mengetahui daftar Wilayah di mana CloudShell VPC tersedia, lihat [AWS Wilayah yang Didukung](#) untuk AWS CloudShell

## Menciptakan lingkungan CloudShell VPC

[Topik ini memandu Anda melalui langkah-langkah untuk membuat lingkungan VPC. CloudShell](#)

## Prasyarat

Administrator Anda harus memberikan izin IAM yang diperlukan agar Anda dapat membuat lingkungan VPC. Untuk informasi selengkapnya tentang mengaktifkan izin untuk membuat lingkungan CloudShell VPC, lihat [the section called “Izin IAM yang diperlukan untuk membuat dan menggunakan lingkungan VPC CloudShell”](#)

Untuk membuat lingkungan CloudShell VPC

1. Di halaman CloudShell konsol, pilih ikon + lalu pilih Buat lingkungan VPC dari menu tarik-turun.
2. Pada halaman Buat lingkungan VPC, masukkan nama untuk lingkungan VPC Anda di kotak Nama.
3. Dari daftar dropdown Virtual Private Cloud (VPC), pilih VPC.
4. Dari daftar dropdown Subnet, pilih subnet.
5. Dari daftar tarik-turun grup Keamanan, pilih satu atau beberapa grup keamanan yang ingin Anda tetapkan ke lingkungan VPC Anda.

### Note

Anda dapat memilih maksimal lima grup keamanan.

6. Pilih Buat untuk membuat lingkungan VPC Anda.
7. (Opsional) Pilih Tindakan, lalu pilih Lihat detail untuk meninjau detail lingkungan VPC yang baru dibuat. Alamat IP lingkungan VPC Anda ditampilkan di prompt baris perintah.

Untuk informasi tentang penggunaan lingkungan VPC, lihat [Memulai](#)

## Izin IAM yang diperlukan untuk membuat dan menggunakan lingkungan VPC CloudShell

Untuk membuat dan menggunakan lingkungan CloudShell VPC, administrator IAM harus mengaktifkan akses ke izin Amazon EC2 khusus VPC. Bagian ini mencantumkan izin Amazon EC2 yang diperlukan untuk membuat dan menggunakan lingkungan VPC.

Untuk membuat lingkungan VPC, kebijakan IAM yang ditetapkan ke peran Anda harus menyertakan izin Amazon EC2 berikut:

- `ec2:DescribeVpcs`
- `ec2:DescribeSubnets`
- `ec2:DescribeSecurityGroups`
- `ec2:DescribeDhcpOptions`
- `ec2:DescribeNetworkInterfaces`
  
- `ec2:CreateTags`
- `ec2:CreateNetworkInterface`
- `ec2:CreateNetworkInterfacePermission`

Kami merekomendasikan untuk memasukkan:

- `ec2:DeleteNetworkInterface`

#### Note

Izin ini tidak wajib, tetapi ini diperlukan CloudShell untuk membersihkan sumber daya ENI (ENI yang dibuat untuk lingkungan CloudShell VPC ditandai `ManagedByCloudShell` dengan kunci) yang dibuat olehnya. Jika izin ini tidak diaktifkan, Anda harus membersihkan sumber daya ENI secara manual setelah setiap penggunaan lingkungan CloudShell VPC.

## Kebijakan IAM memberikan CloudShell akses penuh termasuk akses ke VPC

Contoh berikut menampilkan cara mengaktifkan izin penuh, termasuk akses ke VPC, untuk CloudShell

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCloudShellOperations",
```

```
"Effect": "Allow",
"Action": [
  "cloudshell:*"
],
"Resource": "*"
},
{
  "Sid": "AllowDescribeVPC",
  "Effect": "Allow",
  "Action": [
    "ec2:DescribeSubnets",
    "ec2:DescribeSecurityGroups",
    "ec2:DescribeVpcs"
  ],
  "Resource": "*"
},
{
  "Sid": "AllowInspectVPCConfigurationViaCloudShell",
  "Effect": "Allow",
  "Action": [
    "ec2:DescribeDhcpOptions",
    "ec2:DescribeNetworkInterfaces"
  ],
  "Resource": "*",
  "Condition": {
    "ForAnyValue:StringEquals": {
      "aws:CalledVia": "cloudshell.amazonaws.com"
    }
  }
},
{
  "Sid": "AllowCreateTagWithCloudShellKeyViaCloudShell",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateTags"
  ],
  "Resource": "arn:aws:ec2:*:*:network-interface/*",
  "Condition": {
    "StringEquals": {
      "ec2:CreateAction": "CreateNetworkInterface"
    }
  },
  "ForAnyValue:StringEquals": {
    "aws:TagKeys": "ManagedByCloudShell",
    "aws:CalledVia": "cloudshell.amazonaws.com"
  }
}
```

```
    }
  }
},
{
  "Sid": "AllowCreateNetworkInterfaceWithSubnetsAndSGViaCloudShell",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterface"
  ],
  "Resource": [
    "arn:aws:ec2:*:*:subnet/*",
    "arn:aws:ec2:*:*:security-group/*"
  ],
  "Condition": {
    "ForAnyValue:StringEquals": {
      "aws:CalledVia": "cloudshell.amazonaws.com"
    }
  }
},
{
  "Sid": "AllowCreateNetworkInterfaceWithCloudShellTagViaCloudShell",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterface"
  ],
  "Resource": "arn:aws:ec2:*:*:network-interface/*",
  "Condition": {
    "ForAnyValue:StringEquals": {
      "aws:TagKeys": "ManagedByCloudShell",
      "aws:CalledVia": "cloudshell.amazonaws.com"
    }
  }
},
{
  "Sid": "AllowCreateNetworkInterfacePermissionWithCloudShellTagViaCloudShell",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterfacePermission"
  ],
  "Resource": "arn:aws:ec2:*:*:network-interface/*",
  "Condition": {
    "StringEquals": {
      "aws:ResourceTag/ManagedByCloudShell": ""
    }
  }
},
```

```

    "ForAnyValue:StringEquals": {
      "aws:CalledVia": "cloudshell.amazonaws.com"
    }
  },
  {
    "Sid": "AllowDeleteNetworkInterfaceWithCloudShellTagViaCloudShell",
    "Effect": "Allow",
    "Action": [
      "ec2:DeleteNetworkInterface"
    ],
    "Resource": "arn:aws:ec2:*:*:network-interface/*",
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/ManagedByCloudShell": ""
      },
      "ForAnyValue:StringEquals": {
        "aws:CalledVia": "cloudshell.amazonaws.com"
      }
    }
  }
]
}

```

## Menggunakan kunci kondisi IAM untuk lingkungan VPC

Anda dapat menggunakan tombol CloudShell-specific kondisi untuk pengaturan VPC untuk memberikan kontrol izin tambahan untuk lingkungan VPC Anda. Anda juga dapat menentukan subnet dan grup keamanan yang dapat dan tidak dapat digunakan oleh lingkungan VPC.

CloudShell mendukung kunci kondisi berikut dalam kebijakan IAM:

- `CloudShell:VpcIds`— Izinkan atau tolak satu atau lebih VPC
- `CloudShell:SubnetIds`— Izinkan atau tolak satu atau lebih subnet
- `CloudShell:SecurityGroupIds`— Izinkan atau tolak satu atau lebih grup keamanan

### Note

Jika izin untuk pengguna dengan akses ke CloudShell lingkungan publik dimodifikasi untuk menambahkan pembatasan pada `cloudshell:createEnvironment` tindakan,

mereka masih dapat mengakses lingkungan publik yang ada. Namun, jika Anda ingin mengubah kebijakan IAM dengan pembatasan ini dan menonaktifkan akses mereka ke lingkungan publik yang ada, Anda harus terlebih dahulu memperbarui kebijakan IAM dengan pembatasan, dan kemudian memastikan bahwa setiap CloudShell pengguna di akun Anda secara manual menghapus lingkungan publik yang ada menggunakan antarmuka pengguna CloudShell web (Actions → Delete environment). CloudShell

## Contoh kebijakan dengan kunci syarat untuk pengaturan VPC

Contoh-contoh berikut ini menunjukkan cara menggunakan kunci syarat untuk pengaturan VPC. Setelah Anda membuat pernyataan kebijakan dengan batasan yang diinginkan, tambahkan pernyataan kebijakan untuk pengguna atau peran target.

Pastikan bahwa pengguna hanya membuat lingkungan VPC dan menolak penciptaan lingkungan publik

Untuk memastikan bahwa pengguna hanya dapat membuat lingkungan VPC, gunakan izin tolak seperti yang ditunjukkan pada contoh berikut:

```
{
  "Statement": [
    {
      "Sid": "DenyCloudShellNonVpcEnvironments",
      "Action": [
        "cloudshell:CreateEnvironment"
      ],
      "Effect": "Deny",
      "Resource": "*",
      "Condition": {
        "Null": {
          "cloudshell:VpcIds": "true"
        }
      }
    }
  ]
}
```

## Menolak akses pengguna ke VPC, subnet, atau grup keamanan tertentu

Untuk menolak akses pengguna ke VPC tertentu, gunakan `StringEquals` untuk memeriksa nilai syarat `cloudshell:VpcIds`. Contoh berikut menolak akses pengguna ke `vpc-1` dan `vpc-2`:

Untuk menolak akses pengguna ke VPC tertentu, gunakan `StringEquals` untuk memeriksa nilai syarat `cloudshell:SubnetIds`. Contoh berikut menolak akses pengguna ke `subnet-1` dan `subnet-2`:

Untuk menolak akses pengguna ke VPC tertentu, gunakan `StringEquals` untuk memeriksa nilai syarat `cloudshell:SecurityGroupIds`. Contoh berikut menolak akses pengguna ke `sg-1` dan `sg-2`:

### JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EnforceOutOfSecurityGroups",
      "Action": [
        "cloudshell:CreateEnvironment"
      ],
      "Effect": "Deny",
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "cloudshell:SecurityGroupIds": [
            "sg-1",
            "sg-2"
          ]
        }
      }
    }
  ]
}
```

## Izinkan pengguna membuat lingkungan dengan konfigurasi VPC tertentu

Untuk memungkinkan pengguna mengakses VPC tertentu, gunakan `StringEquals` untuk memeriksa nilai `cloudshell:VpcIds` kondisi. Contoh berikut memungkinkan pengguna mengakses `vpc-1` dan `vpc-2`:

Untuk memungkinkan pengguna mengakses VPC tertentu, gunakan `StringEquals` untuk memeriksa nilai `cloudshell:SubnetIds` kondisi. Contoh berikut memungkinkan pengguna mengakses `subnet-1` dan `subnet-2`:

### JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EnforceStayInSpecificSubnets",
      "Action": [
        "cloudshell:CreateEnvironment"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Condition": {
        "ForAllValues:StringEquals": {
          "cloudshell:SubnetIds": [
            "subnet-1",
            "subnet-2"
          ]
        }
      }
    }
  ]
}
```

Untuk memungkinkan pengguna mengakses VPC tertentu, gunakan `StringEquals` untuk memeriksa nilai `cloudshell:SecurityGroupIds` kondisi. Contoh berikut memungkinkan pengguna mengakses `sg-1` dan `sg-2`:

## JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EnforceStayInSpecificSecurityGroup",
      "Action": [
        "cloudshell:CreateEnvironment"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Condition": {
        "ForAllValues:StringEquals": {
          "cloudshell:SecurityGroupIds": [
            "sg-1",
            "sg-2"
          ]
        }
      }
    }
  ]
}
```

# Keamanan untuk AWS CloudShell

Keamanan cloud di Amazon Web Services (AWS) merupakan 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 ini sebagai Keamanan dari Cloud dan Keamanan dalam Cloud.

Security of the Cloud - AWS bertanggung jawab untuk melindungi infrastruktur yang menjalankan semua layanan yang ditawarkan di AWS Cloud dan memberi Anda layanan yang dapat Anda gunakan dengan aman. Tanggung jawab keamanan kami adalah prioritas tertinggi di AWS, dan efektivitas keamanan kami secara teratur diuji dan diverifikasi oleh auditor pihak ketiga sebagai bagian dari [Program AWS Kepatuhan](#).

Keamanan di Cloud — Tanggung jawab Anda ditentukan oleh AWS layanan yang Anda gunakan, dan faktor-faktor lain termasuk sensitivitas data Anda, persyaratan organisasi Anda, serta undang-undang dan peraturan yang berlaku.

AWS CloudShell mengikuti [model tanggung jawab bersama](#) melalui AWS layanan khusus yang didukungnya. Untuk informasi keamanan AWS layanan, lihat [halaman dokumentasi keamanan AWS layanan](#) dan [AWS layanan yang berada dalam lingkup upaya AWS kepatuhan oleh program kepatuhan](#).

Topik berikut menunjukkan cara mengonfigurasi AWS CloudShell untuk memenuhi tujuan keamanan dan kepatuhan Anda.

Topik

- [Perlindungan data di AWS CloudShell](#)
- [Identity and Access Management untuk AWS CloudShell](#)
- [Penebangan dan pemantauan di AWS CloudShell](#)
- [Validasi kepatuhan untuk AWS CloudShell](#)
- [Ketahanan di AWS CloudShell](#)
- [Keamanan infrastruktur di AWS CloudShell](#)
- [Praktik terbaik keamanan untuk AWS CloudShell](#)
- [AWS CloudShell FAQ Keamanan](#)

# Perlindungan data di AWS CloudShell

[Model tanggung jawab bersama](#) AWS berlaku untuk perlindungan data di AWS CloudShell. 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 logging 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 AWS CloudShell 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.

## Enkripsi data

Enkripsi data mengacu pada melindungi data saat diam saat disimpan di dalam AWS CloudShell dan saat transit, ia melakukan perjalanan antara AWS CloudShell dan titik akhir layanan.

### Enkripsi saat istirahat menggunakan AWS KMS

Enkripsi saat istirahat didefinisikan sebagai perlindungan data dari akses tidak sah dengan mengenkripsi data saat disimpan. Saat menggunakan AWS CloudShell, Anda memiliki penyimpanan persisten 1 GB per AWS Wilayah tanpa biaya. Penyimpanan persisten terletak di direktori home Anda (\$HOME) dan bersifat pribadi untuk Anda. Tidak seperti sumber daya lingkungan sementara yang didaur ulang setelah setiap sesi shell berakhir, data di direktori home Anda tetap ada.

Enkripsi data yang disimpan AWS CloudShell diimplementasikan menggunakan kunci kriptografi yang disediakan oleh AWS Key Management Service (AWS KMS). Ini adalah AWS layanan terkelola untuk membuat dan AWS KMS keys mengendalikan kunci enkripsi yang digunakan untuk mengenkripsi data pelanggan yang disimpan di AWS CloudShell lingkungan. AWS CloudShell menghasilkan dan mengelola kunci kriptografi untuk mengenkripsi data atas nama pelanggan.

### Enkripsi saat bergerak

Enkripsi dalam transit didefinisikan sebagai perlindungan data dari intersepsi saat data ditransfer antar-endpoint komunikasi.

Secara default, semua komunikasi data antara komputer browser web klien dan berbasis cloud AWS CloudShell dienkripsi dengan mengirimkan semuanya melalui koneksi. HTTPS/TLS

Anda tidak perlu melakukan apa pun untuk memungkinkan penggunaan HTTPS/TLS komunikasi.

## Identity and Access Management untuk AWS CloudShell

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. CloudShell IAM adalah Layanan AWS yang dapat Anda gunakan tanpa biaya tambahan.

## Topik

- [Audiens](#)
- [Mengautentikasi dengan identitas](#)
- [Mengelola akses menggunakan kebijakan](#)
- [Bagaimana AWS CloudShell bekerja dengan IAM](#)
- [Identity-based contoh kebijakan untuk AWS CloudShell](#)
- [Memecahkan masalah CloudShell identitas dan akses AWS](#)
- [Mengelola AWS CloudShell akses dan penggunaan dengan kebijakan IAM](#)

## 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 CloudShell identitas dan akses AWS](#))
- Administrator layanan - tentukan akses pengguna dan mengirimkan permintaan izin (lihat [Bagaimana AWS CloudShell bekerja dengan IAM](#))
- Administrator IAM - tulis kebijakan untuk mengelola akses (lihat [Identity-based contoh kebijakan untuk AWS CloudShell](#))

## Mengautentikasi dengan identitas

Otentikasi adalah cara Anda masuk AWS menggunakan kredensial 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 kredensial sementara.

Identitas federasi adalah pengguna dari direktori perusahaan Anda, penyedia identitas web, atau Directory Service yang mengakses Layanan AWS menggunakan kredensial 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 kredensial 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.

### Identity-based kebijakan

Identity-based kebijakan 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.

Identity-based kebijakan dapat berupa kebijakan inline (disematkan langsung ke dalam satu identitas) atau kebijakan terkelola (kebijakan mandiri yang dilampirkan pada beberapa identitas). Untuk mempelajari cara memilih antara kebijakan terkelola dan kebijakan inline, lihat [Pilih antara kebijakan terkelola dan kebijakan inline](#) dalam Panduan Pengguna IAM.

### Resource-based kebijakan

Resource-based kebijakan 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.

Resource-based kebijakan adalah 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 (SCP) – Menentukan 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 (RCP) – Menetapkan izin maksimum yang tersedia untuk sumber daya di akun Anda. Untuk informasi selengkapnya, lihat [Kebijakan kontrol sumber daya \(RCP\)](#) dalam Panduan Pengguna AWS Organizations .
- 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 AWS CloudShell bekerja dengan IAM

Sebelum Anda menggunakan IAM untuk mengelola akses CloudShell, pelajari fitur IAM yang tersedia untuk digunakan. CloudShell

## Fitur IAM yang dapat Anda gunakan dengan AWS CloudShell

Fitur IAM	CloudShell dukungan
<a href="#">Identity-based kebijakan</a>	Ya
<a href="#">Resource-based kebijakan</a>	Tidak
<a href="#">Tindakan kebijakan</a>	Ya
<a href="#">Sumber daya kebijakan</a>	Ya
<a href="#">kunci-kunci persyaratan kebijakan (spesifik layanan)</a>	Ya
<a href="#">ACL</a>	Tidak
<a href="#">ABAC (tanda dalam kebijakan)</a>	Tidak
<a href="#">Kredensial sementara</a>	Ya
<a href="#">Sesi akses teruskan (FAS)</a>	Tidak
<a href="#">Peran layanan</a>	Tidak
<a href="#">Service-linked peran</a>	Tidak

Untuk mendapatkan tampilan tingkat tinggi tentang cara CloudShell dan AWS layanan lain bekerja dengan sebagian besar fitur IAM, lihat [AWS layanan yang bekerja dengan IAM di Panduan Pengguna IAM](#).

### Identity-based kebijakan untuk CloudShell

Mendukung kebijakan berbasis identitas: Ya

Identity-based kebijakan adalah dokumen kebijakan izin JSON yang dapat Anda lampirkan ke identitas, seperti pengguna IAM, grup pengguna, atau peran. 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.

### Identity-based contoh kebijakan untuk CloudShell

Untuk melihat contoh kebijakan CloudShell berbasis identitas, lihat. [Identity-based contoh kebijakan untuk AWS CloudShell](#)

### Resource-based kebijakan dalam CloudShell

Mendukung kebijakan berbasis sumber daya: Tidak

Resource-based kebijakan 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 CloudShell

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 CloudShell tindakan, lihat [Tindakan yang ditentukan oleh AWS CloudShell](#) di Referensi Otorisasi Layanan. Beberapa tindakan mungkin memiliki lebih dari satu API.

Tindakan kebijakan CloudShell menggunakan awalan berikut sebelum tindakan:

```
cloudshell
```

Untuk menetapkan secara spesifik beberapa tindakan dalam satu pernyataan, pisahkan tindakan tersebut dengan koma.

```
"Action": [  
  "cloudshell:action1",  
  "cloudshell:action2"  
]
```

Untuk melihat contoh kebijakan CloudShell berbasis identitas, lihat [Identity-based contoh kebijakan untuk AWS CloudShell](#)

## Sumber daya kebijakan untuk CloudShell

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 CloudShell sumber daya dan ARNnya, lihat Sumber [daya yang ditentukan oleh AWS CloudShell](#) di Referensi Otorisasi Layanan. Untuk mempelajari tindakan mana yang dapat Anda tentukan ARN dari setiap sumber daya, lihat [Tindakan yang ditentukan oleh AWS](#). CloudShell

Untuk melihat contoh kebijakan CloudShell berbasis identitas, lihat [Identity-based contoh kebijakan untuk AWS CloudShell](#)

## Kunci kondisi kebijakan untuk CloudShell

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 CloudShell kondisi, lihat [Kunci kondisi untuk AWS CloudShell](#) di Referensi Otorisasi Layanan. Untuk mempelajari tindakan dan sumber daya yang dapat Anda gunakan kunci kondisi, lihat [Tindakan yang ditentukan oleh AWS CloudShell](#).

Untuk melihat contoh kebijakan CloudShell berbasis identitas, lihat [Identity-based contoh kebijakan untuk AWS CloudShell](#)

## ACL di CloudShell

Mendukung ACL: Tidak

Daftar kontrol akses (ACL) mengendalikan principal mana (anggota akun, pengguna, atau peran) yang memiliki izin untuk mengakses sumber daya. ACL serupa dengan kebijakan berbasis sumber daya, meskipun kebijakan tersebut tidak menggunakan format dokumen kebijakan JSON.

## ABAC dengan CloudShell

Mendukung ABAC (tag dalam kebijakan): Tidak

Attribute-based Access Control (ABAC) adalah strategi otorisasi yang mendefinisikan izin berdasarkan atribut yang disebut tag. 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.

## Menggunakan kredensi sementara dengan CloudShell

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 menghasilkan kredensial sementara secara dinamis 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.

Ketika Anda beralih peran, Anda akan menggunakan lingkungan yang berbeda. Anda tidak dapat beralih peran dalam AWS CloudShell lingkungan yang sama.

## Teruskan sesi akses untuk CloudShell

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 CloudShell

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 CloudShell fungsionalitas. Edit peran layanan hanya jika CloudShell memberikan panduan untuk melakukannya.

## Service-linked peran untuk CloudShell

Mendukung peran terkait layanan: Tidak

Peran terkait layanan adalah jenis peran layanan yang ditautkan ke. Layanan AWS Layanan dapat mengambil peran untuk melakukan tindakan atas nama Anda. Service-linked peran muncul di Akun AWS dan dimiliki oleh layanan. Administrator IAM dapat melihat, tetapi tidak dapat mengedit izin untuk peran terkait layanan.

## Identity-based contoh kebijakan untuk AWS CloudShell

Secara default, pengguna dan peran tidak memiliki izin untuk membuat atau mengubah sumber daya CloudShell. 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 CloudShell, termasuk format ARN untuk setiap jenis sumber daya, lihat [Kunci tindakan, sumber daya, dan kondisi untuk AWS CloudShell](#) di Referensi Otorisasi Layanan.

### Topik

- [Praktik terbaik kebijakan](#)
- [Menggunakan CloudShell konsol](#)
- [Mengizinkan pengguna melihat izin mereka sendiri](#)

## Praktik terbaik kebijakan

Identity-based kebijakan menentukan apakah seseorang dapat membuat, mengakses, atau menghapus CloudShell 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.

## Menggunakan CloudShell konsol

Untuk mengakses CloudShell konsol AWS, Anda harus memiliki set izin minimum. Izin ini harus memungkinkan Anda untuk membuat daftar dan melihat detail tentang CloudShell sumber daya di Anda Akun AWS. Jika Anda membuat kebijakan berbasis identitas yang lebih ketat daripada izin minimum yang diperlukan, konsol tidak akan berfungsi sebagaimana mestinya untuk entitas (pengguna atau peran) dengan kebijakan tersebut.

Anda tidak perlu mengizinkan izin konsol minimum untuk pengguna yang melakukan panggilan hanya ke AWS CLI atau AWS API. Sebagai gantinya, izinkan akses hanya ke tindakan yang sesuai dengan operasi API yang coba mereka lakukan.

Untuk memastikan bahwa pengguna dan peran masih dapat menggunakan CloudShell konsol, lampirkan juga kebijakan CloudShell *ConsoleAccess* atau *ReadOnly* AWS terkelola ke entitas. Untuk informasi selengkapnya, lihat [Menambah izin untuk pengguna](#) 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": "*"
  }
]
```

## Memecahkan masalah CloudShell identitas dan akses AWS

Gunakan informasi berikut untuk membantu Anda mendiagnosis dan memperbaiki masalah umum yang mungkin Anda temui saat bekerja dengan CloudShell dan IAM.

### Topik

- [Saya tidak berwenang untuk melakukan tindakan di CloudShell](#)
- [Saya tidak berwenang untuk melakukan iam: PassRole](#)
- [Saya ingin mengizinkan orang di luar saya Akun AWS untuk mengakses CloudShell sumber daya saya](#)

### Saya tidak berwenang untuk melakukan tindakan di CloudShell

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 `aws:GetWidget` rekaan.

```
User: arn:aws:iam::123456789012:user/mateojackson is not authorized to perform:
aws:GetWidget on resource: my-example-widget
```

Dalam hal ini, kebijakan untuk pengguna `mateojackson` harus diperbarui untuk mengizinkan akses ke sumber daya `my-example-widget` dengan menggunakan tindakan awes: `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 CloudShell.

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 CloudShell. 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 CloudShell 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 (ACL), Anda dapat menggunakan kebijakan tersebut untuk memberi orang akses ke sumber daya Anda.

Untuk mempelajari selengkapnya, periksa referensi berikut:

- Untuk mempelajari apakah CloudShell mendukung fitur ini, lihat [Bagaimana AWS CloudShell 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](#).


## Mengelola AWS CloudShell akses dan penggunaan dengan kebijakan IAM

Dengan sumber daya manajemen akses yang dapat disediakan oleh AWS Identity and Access Management, administrator dapat memberikan izin kepada pengguna IAM. Dengan begitu, pengguna ini dapat mengakses AWS CloudShell dan menggunakan fitur lingkungan. Administrator juga dapat membuat kebijakan yang menentukan pada tingkat terperinci tindakan apa yang dapat dilakukan pengguna tersebut dengan lingkungan shell.

Cara tercepat bagi administrator untuk memberikan akses ke pengguna adalah melalui kebijakan AWS terkelola. [Kebijakan terkelola AWS](#) adalah kebijakan mandiri yang dibuat dan dikelola oleh AWS. Kebijakan AWS terkelola berikut ini AWS CloudShell dapat dilampirkan ke identitas IAM:

- **AWS CloudShellFullAccess**: Memberikan izin untuk menggunakan AWS CloudShell dengan akses penuh ke semua fitur.

**AWS CloudShellFullAccess** Kebijakan ini menggunakan karakter wildcard (\*) untuk memberikan identitas IAM (pengguna, peran, atau grup) akses penuh ke CloudShell dan fitur. Untuk informasi selengkapnya tentang kebijakan ini, lihat [AWS CloudShellFullAccess](#) di Panduan Pengguna Kebijakan AWS Terkelola.

 Note

Identitas IAM dengan kebijakan AWS terkelola berikut juga dapat diluncurkan. CloudShell Namun, kebijakan ini memberikan izin ekstensif. Jadi, kami menyarankan Anda hanya memberikan kebijakan ini jika kebijakan tersebut penting untuk peran pekerjaan pengguna IAM.


- [Administrator](#): Menyediakan pengguna IAM dengan akses penuh dan memungkinkan mereka untuk mendelegasikan izin ke setiap layanan dan sumber daya di AWS
- [Pengguna daya pengembang](#): Memungkinkan pengguna IAM untuk melakukan tugas pengembangan aplikasi dan membuat serta mengonfigurasi sumber daya dan layanan yang mendukung pengembangan aplikasi yang AWS sadar.

Untuk informasi selengkapnya tentang melampirkan kebijakan terkelola, lihat [Menambahkan izin identitas IAM \(konsol\) di Panduan Pengguna IAM](#).

## Mengelola tindakan yang diijinkan di AWS CloudShell menggunakan kebijakan khusus

Untuk mengelola tindakan yang dapat dilakukan pengguna IAM CloudShell, buat kebijakan kustom yang menggunakan kebijakan CloudShellPolicy terkelola sebagai templat. Atau, edit [kebijakan inline](#) yang disematkan dalam identitas IAM yang relevan (pengguna, grup, atau peran).

Misalnya, Anda dapat mengizinkan pengguna IAM untuk mengakses CloudShell, tetapi mencegah mereka meneruskan kredensi CloudShell lingkungan yang digunakan untuk masuk. Konsol Manajemen AWS

 Important

Untuk memulai AWS CloudShell dari Konsol Manajemen AWS, pengguna IAM memerlukan izin untuk tindakan berikut:

- `CreateEnvironment`
- `CreateSession`
- `GetEnvironmentStatus`

- `StartEnvironment`

Jika salah satu tindakan ini tidak diizinkan secara eksplisit oleh kebijakan terlampir, kesalahan izin IAM akan ditampilkan saat Anda mencoba meluncurkan. CloudShell

## AWS CloudShell izin

Nama	Deskripsi izin yang diberikan	Diperlukan untuk meluncurkan CloudShell?
<code>cloudshell:CreateEnvironment</code>	Menciptakan CloudShell lingkungan, mengambil tata letak di awal CloudShell sesi, dan menyimpan tata letak saat ini dari aplikasi web di backend. Izin ini hanya mengharapkan * sebagai nilai untuk Resource sebagaimana diuraikan dalam <a href="#">the section called “Contoh kebijakan IAM untuk CloudShell”</a>	Ya
<code>cloudshell:CreateSession</code>	Terhubung ke CloudShell lingkungan dari Konsol Manajemen AWS.	Ya
<code>cloudshell:GetEnvironmentStatus</code>	Baca status CloudShell lingkungan.	Ya
<code>cloudshell&gt;DeleteEnvironment</code>	Menghapus CloudShell lingkungan.	Tidak
<code>cloudshell:GetFileDownloadURLs</code>	Menghasilkan URL Amazon S3 yang telah	Tidak

Nama	Deskripsi izin yang diberikan	Diperlukan untuk meluncurkan CloudShell?
	ditandatangani sebelumnya yang digunakan untuk mengunduh file CloudShell melalui menggunakan CloudShell antarmuka web. Ini tidak tersedia untuk lingkungan VPC.	
<code>cloudshell:GetFileUploadUrls</code>	Menghasilkan URL Amazon S3 yang telah ditandatangani sebelumnya yang digunakan untuk mengunggah file CloudShell melalui menggunakan CloudShell antarmuka web. Ini tidak tersedia untuk lingkungan VPC.	Tidak
<code>cloudshell:DescribeEnvironments</code>	Menjelaskan lingkungan.	Tidak
<code>cloudshell:PutCredentials</code>	Meneruskan kredensial yang digunakan untuk masuk ke ke. Konsol Manajemen AWS CloudShell	Tidak
<code>cloudshell:StartEnvironment</code>	Memulai CloudShell lingkungan yang dihentikan.	Ya
<code>cloudshell:StopEnvironment</code>	Menghentikan CloudShell lingkungan yang sedang berjalan.	Tidak

Nama	Deskripsi izin yang diberikan	Diperlukan untuk meluncurkan CloudShell?
<code>cloudshell:ApproveCommand</code>	Menyetujui perintah yang dikirim CloudShell dari konsol AWS Layanan lain.	Tidak

### Contoh kebijakan IAM untuk CloudShell

Contoh berikut menunjukkan bagaimana kebijakan dapat dibuat untuk membatasi siapa yang dapat mengakses CloudShell. Contoh juga menunjukkan tindakan yang dapat dilakukan di lingkungan shell.

Kebijakan berikut ini memberlakukan penolakan lengkap atas akses CloudShell dan fitur-fiturnya.

#### JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "DenyCloudShell",
      "Effect": "Deny",
      "Action": [
        "cloudshell:*"
      ],
      "Resource": "*"
    }
  ]
}
```

Kebijakan berikut ini memungkinkan pengguna IAM untuk mengakses CloudShell tetapi memblokir mereka dari membuat URL yang telah ditandatangani sebelumnya untuk mengunggah dan mengunduh file. Pengguna masih dapat mentransfer file ke dan dari lingkungan, menggunakan klien seperti `wget` misalnya.

#### JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```
{
  "Sid": "AllowUsingCloudshell",
  "Effect": "Allow",
  "Action": [
    "cloudshell:*"
  ],
  "Resource": "*"
},
{
  "Sid": "DenyUploadDownload",
  "Effect": "Deny",
  "Action": [
    "cloudshell:GetFileDownloadUrls",
    "cloudshell:GetFileUploadUrls"
  ],
  "Resource": "*"
}]
}
```

Kebijakan berikut memungkinkan pengguna IAM untuk mengakses CloudShell. Namun, kebijakan ini mencegah kredensial yang Anda gunakan untuk masuk ke Konsol Manajemen AWS agar tidak diteruskan ke lingkungan. CloudShell Pengguna IAM dengan kebijakan ini perlu mengonfigurasi kredensialnya secara manual di dalamnya. CloudShell

## JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowUsingCloudshell",
      "Effect": "Allow",
      "Action": [
        "cloudshell:*"
      ],
      "Resource": "*"
    },
    {
      "Sid": "DenyCredentialForwarding",
      "Effect": "Deny",
      "Action": [
```

```

        "cloudshell:PutCredentials"
    ],
    "Resource": "*"
}]
}

```

Kebijakan berikut memungkinkan pengguna IAM untuk membuat AWS CloudShell lingkungan.

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [{
    "Sid": "CloudShellUser",
    "Effect": "Allow",
    "Action": [
      "cloudshell:CreateEnvironment",
      "cloudshell:CreateSession",
      "cloudshell:GetEnvironmentStatus",
      "cloudshell:StartEnvironment"
    ],
    "Resource": "*"
  }]
}

```

## Izin IAM yang diperlukan untuk membuat dan menggunakan lingkungan VPC CloudShell

Untuk membuat dan menggunakan lingkungan CloudShell VPC, administrator IAM harus mengaktifkan akses ke izin Amazon EC2 khusus VPC. Bagian ini mencantumkan izin Amazon EC2 yang diperlukan untuk membuat dan menggunakan lingkungan VPC.

Untuk membuat lingkungan VPC, kebijakan IAM yang ditetapkan ke peran Anda harus menyertakan izin Amazon EC2 berikut:

- `ec2:DescribeVpcs`
- `ec2:DescribeSubnets`
- `ec2:DescribeSecurityGroups`

- ec2:DescribeDhcpOptions
- ec2:DescribeNetworkInterfaces
  
- ec2:CreateTags
- ec2:CreateNetworkInterface
- ec2:CreateNetworkInterfacePermission

Kami merekomendasikan juga termasuk:

- ec2>DeleteNetworkInterface

#### Note

Izin ini tidak wajib, tetapi ini diperlukan CloudShell untuk membersihkan sumber daya ENI (ENI yang dibuat untuk lingkungan CloudShell VPC ditandai ManagedByCloudShell dengan kunci) yang dibuat olehnya. Jika izin ini tidak diaktifkan, Anda harus membersihkan sumber daya ENI secara manual setelah setiap penggunaan lingkungan CloudShell VPC.

Kebijakan IAM memberikan CloudShell akses penuh termasuk akses ke VPC

Contoh berikut menampilkan cara mengaktifkan izin penuh, termasuk akses ke VPC, untuk: CloudShell

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "AllowCloudShellOperations",
      "Effect": "Allow",
      "Action": [
        "cloudshell:*"
      ],
      "Resource": "*"
    }
  ],
  {
```

```
"Sid": "AllowDescribeVPC",
"Effect": "Allow",
"Action": [
  "ec2:DescribeDhcpOptions",
  "ec2:DescribeNetworkInterfaces",
  "ec2:DescribeSubnets",
  "ec2:DescribeSecurityGroups",
  "ec2:DescribeVpcs"
],
"Resource": "*"
},
{
  "Sid": "AllowCreateTagWithCloudShellKey",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateTags"
  ],
  "Resource": "arn:aws:ec2:*:*:network-interface/*",
  "Condition": {
    "StringEquals": {
      "ec2:CreateAction": "CreateNetworkInterface"
    },
    "ForAnyValue:StringEquals": {
      "aws:TagKeys": "ManagedByCloudShell"
    }
  }
},
{
  "Sid": "AllowCreateNetworkInterfaceWithSubnetsAndSG",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterface"
  ],
  "Resource": [
    "arn:aws:ec2:*:*:subnet/*",
    "arn:aws:ec2:*:*:security-group*"
  ]
},
{
  "Sid": "AllowCreateNetworkInterfaceWithCloudShellTag",
  "Effect": "Allow",
  "Action": [
    "ec2:CreateNetworkInterface"
  ],
```

```

    "Resource": "arn:aws:ec2:*:*:network-interface/*",
    "Condition": {
      "ForAnyValue:StringEquals": {
        "aws:TagKeys": "ManagedByCloudShell"
      }
    },
  },
  {
    "Sid": "AllowCreateNetworkInterfacePermissionWithCloudShellTag",
    "Effect": "Allow",
    "Action": [
      "ec2:CreateNetworkInterfacePermission"
    ],
    "Resource": "arn:aws:ec2:*:*:network-interface/*",
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/ManagedByCloudShell": ""
      }
    }
  },
  {
    "Sid": "AllowDeleteNetworkInterfaceWithCloudShellTag",
    "Effect": "Allow",
    "Action": [
      "ec2>DeleteNetworkInterface"
    ],
    "Resource": "arn:aws:ec2:*:*:network-interface/*",
    "Condition": {
      "StringEquals": {
        "aws:ResourceTag/ManagedByCloudShell": ""
      }
    }
  }
]
}

```

## Menggunakan kunci kondisi IAM untuk lingkungan VPC

Anda dapat menggunakan tombol CloudShell-specific kondisi untuk pengaturan VPC untuk memberikan kontrol izin tambahan untuk lingkungan VPC Anda. Anda juga dapat menentukan subnet dan grup keamanan yang dapat dan tidak dapat digunakan oleh lingkungan VPC.

CloudShell mendukung kunci kondisi berikut dalam kebijakan IAM:

- `CloudShell:VpcIds`— Izinkan atau tolak satu atau lebih VPC
- `CloudShell:SubnetIds`— Izinkan atau tolak satu atau lebih subnet
- `CloudShell:SecurityGroupIds`— Izinkan atau tolak satu atau lebih grup keamanan

#### Note

Jika izin untuk pengguna dengan akses ke CloudShell lingkungan publik dimodifikasi untuk menambahkan pembatasan pada `cloudshell:createEnvironment` tindakan, mereka masih dapat mengakses lingkungan publik yang ada. Namun, jika Anda ingin mengubah kebijakan IAM dengan pembatasan ini dan menonaktifkan akses mereka ke lingkungan publik yang ada, Anda harus terlebih dahulu memperbarui kebijakan IAM dengan pembatasan, dan kemudian memastikan bahwa setiap CloudShell pengguna di akun Anda secara manual menghapus lingkungan publik yang ada menggunakan antarmuka pengguna CloudShell web (Actions → Delete environment). CloudShell

Contoh kebijakan dengan kunci syarat untuk pengaturan VPC

Contoh-contoh berikut ini menunjukkan cara menggunakan kunci syarat untuk pengaturan VPC. Setelah Anda membuat pernyataan kebijakan dengan batasan yang diinginkan, tambahkan pernyataan kebijakan untuk pengguna atau peran target.

Pastikan bahwa pengguna hanya membuat lingkungan VPC dan menolak penciptaan lingkungan publik

Untuk memastikan bahwa pengguna hanya dapat membuat lingkungan VPC, gunakan izin tolak seperti yang ditunjukkan pada contoh berikut:

```
{
  "Statement": [
    {
      "Sid": "DenyCloudShellNonVpcEnvironments",
      "Action": [
        "cloudshell:CreateEnvironment"
      ],
      "Effect": "Deny",
```

```

    "Resource": "*",
    "Condition": {
      "Null": {
        "cloudshell:VpcIds": "true"
      }
    }
  }
]
}

```

Menolak akses pengguna ke VPC, subnet, atau grup keamanan tertentu

Untuk menolak akses pengguna ke VPC tertentu, gunakan `StringEquals` untuk memeriksa nilai syarat `cloudshell:VpcIds`. Contoh berikut menolak akses pengguna ke `vpc-1` dan `vpc-2`:

Untuk menolak akses pengguna ke VPC tertentu, gunakan `StringEquals` untuk memeriksa nilai syarat `cloudshell:SubnetIds`. Contoh berikut menolak akses pengguna ke `subnet-1` dan `subnet-2`:

Untuk menolak akses pengguna ke VPC tertentu, gunakan `StringEquals` untuk memeriksa nilai syarat `cloudshell:SecurityGroupIds`. Contoh berikut menolak akses pengguna ke `sg-1` dan `sg-2`:

JSON

```

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EnforceOutOfSecurityGroups",
      "Action": [
        "cloudshell:CreateEnvironment"
      ],
      "Effect": "Deny",
      "Resource": "*",
      "Condition": {
        "ForAnyValue:StringEquals": {
          "cloudshell:SecurityGroupIds": [
            "sg-1",
            "sg-2"
          ]
        }
      }
    }
  ]
}

```

```
}
  }
}
]
```

Izinkan pengguna membuat lingkungan dengan konfigurasi VPC tertentu

Untuk memungkinkan pengguna mengakses VPC tertentu, gunakan `StringEquals` untuk memeriksa nilai `cloudshell:VpcIds` kondisi. Contoh berikut memungkinkan pengguna mengakses `vpc-1` dan `vpc-2`:

Untuk memungkinkan pengguna mengakses VPC tertentu, gunakan `StringEquals` untuk memeriksa nilai `cloudshell:SubnetIds` kondisi. Contoh berikut memungkinkan pengguna mengakses `subnet-1` dan `subnet-2`:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EnforceStayInSpecificSubnets",
      "Action": [
        "cloudshell:CreateEnvironment"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Condition": {
        "ForAllValues:StringEquals": {
          "cloudshell:SubnetIds": [
            "subnet-1",
            "subnet-2"
          ]
        }
      }
    }
  ]
}
```

Untuk memungkinkan pengguna mengakses VPC tertentu, gunakan `StringEquals` untuk memeriksa nilai `cloudshell:SecurityGroupIds` kondisi. Contoh berikut memungkinkan pengguna mengakses `sg-1` dan `sg-2`:

JSON

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "EnforceStayInSpecificSecurityGroup",
      "Action": [
        "cloudshell:CreateEnvironment"
      ],
      "Effect": "Allow",
      "Resource": "*",
      "Condition": {
        "ForAllValues:StringEquals": {
          "cloudshell:SecurityGroupIds": [
            "sg-1",
            "sg-2"
          ]
        }
      }
    }
  ]
}
```

## Izin untuk mengakses Layanan AWS

CloudShell menggunakan kredensial IAM yang Anda gunakan untuk masuk ke Konsol Manajemen AWS

### Note

Untuk menggunakan kredensial IAM yang Anda gunakan untuk masuk Konsol Manajemen AWS, Anda harus memiliki izin `cloudshell:PutCredentials`

Fitur pra-otentikasi ini CloudShell membuatnya nyaman untuk digunakan. AWS CLI Namun, pengguna IAM masih memerlukan izin eksplisit untuk Layanan AWS yang dipanggil dari baris perintah.

Misalnya, pengguna IAM diharuskan membuat bucket Amazon S3 dan mengunggah file sebagai objek kepada mereka. Anda dapat membuat kebijakan yang secara eksplisit mengizinkan tindakan tersebut. Konsol IAM menyediakan [editor visual](#) interaktif yang memandu melalui proses membangun dokumen JSON-formatted kebijakan. Setelah kebijakan dibuat, Anda dapat melampirkannya ke identitas IAM yang relevan (pengguna, grup, atau peran).

Untuk informasi selengkapnya tentang melampirkan kebijakan terkelola, lihat [Menambahkan izin identitas IAM \(konsol\) di Panduan Pengguna IAM](#).

## Izin untuk mengakses fitur Amazon Q CLI di CloudShell

Untuk menggunakan fitur Amazon Q CLI CloudShell, seperti saran sebaris, obrolan, dan terjemahkan, pastikan Anda memiliki izin IAM yang diperlukan. Jika Anda tidak dapat mengakses fitur Amazon Q CLI CloudShell, hubungi administrator Anda untuk memberi Anda izin IAM yang diperlukan. Untuk informasi selengkapnya, lihat [contoh Identity-based kebijakan untuk Pengembang Amazon Q](#) di Panduan Pengguna Pengembang Amazon Q.

## Penebangan dan pemantauan di AWS CloudShell

Topik ini menjelaskan bagaimana Anda dapat mencatat dan memantau AWS CloudShell aktivitas dan kinerja dengan CloudTrail.

### Memantau aktivitas dengan CloudTrail

AWS CloudShell terintegrasi dengan AWS CloudTrail, layanan yang menyediakan catatan tindakan yang diambil oleh pengguna, peran, atau Layanan AWS dalam AWS CloudShell. CloudTrail menangkap semua panggilan API untuk AWS CloudShell sebagai peristiwa. Panggilan yang diambil termasuk panggilan dari AWS CloudShell konsol dan panggilan kode ke AWS CloudShell API.

Jika Anda membuat jejak, Anda dapat mengaktifkan pengiriman CloudTrail acara secara berkelanjutan ke bucket Amazon Simple Storage Service (Amazon S3). Ini termasuk acara untuk AWS CloudShell.

Jika Anda tidak membuat konfigurasi jejak, Anda masih dapat melihat kejadian terbaru dalam konsol CloudTrail di Riwayat peristiwa. Dengan menggunakan informasi yang dikumpulkan oleh CloudTrail, Anda dapat menemukan berbagai informasi tentang permintaan. Misalnya, Anda dapat menentukan

permintaan yang dibuat ke AWS CloudShell, Anda dapat mempelajari alamat IP tempat permintaan dibuat, siapa yang membuat permintaan, dan kapan permintaan itu dibuat.

## AWS CloudShell di CloudTrail

Tabel berikut mencantumkan AWS CloudShell peristiwa yang disimpan dalam file CloudTrail log.

### Note

AWS CloudShell acara yang meliputi:

- \*menunjukkan bahwa ini adalah panggilan API yang tidak bermutasi (hanya-baca).
- Kata tersebut `Environment` berkaitan dengan siklus hidup lingkungan komputasi yang menampung pengalaman shell.
- Kata `Layout` mengembalikan semua tab browser di terminal. CloudShell

### CloudShell Acara di CloudTrail

Nama peristiwa	Deskripsi
<code>createEnvironment</code>	Terjadi ketika CloudShell lingkungan dibuat.
<code>createSession</code>	Terjadi ketika CloudShell lingkungan terhubung dari Konsol Manajemen AWS.
<code>deleteEnvironment</code>	Terjadi ketika CloudShell lingkungan dihapus.
<code>deleteSession</code>	Terjadi ketika sesi di CloudShell tab yang berjalan di tab browser saat ini dihapus.
<code>getEnvironmentStatus*</code>	Terjadi ketika status CloudShell lingkungan diambil.
<code>getFileDownloadUrls*</code>	Terjadi ketika URL Amazon S3 yang telah ditandatangani sebelumnya yang digunakan untuk mengunduh file CloudShell melalui

Nama peristiwa	Deskripsi
	menggunakan antarmuka web CloudShell dihasilkan.
getFileUploadUrls*	Terjadi ketika URL Amazon S3 yang telah ditandatangani sebelumnya yang digunakan untuk mengunggah file CloudShell melalui menggunakan antarmuka web CloudShell dihasilkan.
cloudshell:DescribeEnvironments	Menjelaskan lingkungan.
getLayout*	Terjadi ketika CloudShell tata letak di awal sesi diambil.
putCredentials	Terjadi ketika kredensial yang digunakan untuk masuk ke Konsol Manajemen AWS to CloudShell diteruskan.
redeemCode*	Terjadi ketika alur kerja untuk mengambil token penyegaran di CloudShell lingkungan dimulai. Anda nantinya dapat menggunakan token ini dalam putCredentials perintah untuk mengakses CloudShell lingkungan.
sendHeartBeat	Terjadi untuk mengkonfirmasi bahwa CloudShell sesi aktif.
startEnvironment	Terjadi ketika CloudShell lingkungan dimulai.
stopEnvironment	Terjadi ketika CloudShell lingkungan berjalan dihentikan.
updateLayout	Terjadi ketika tata letak saat ini dari aplikasi web di backend disimpan.

Acara yang menyertakan kata “Layout” mengembalikan semua tab browser di CloudShell terminal.

## EventBridge aturan untuk AWS CloudShell tindakan

Dengan EventBridge aturan, Anda menentukan tindakan target yang akan diambil saat EventBridge menerima acara yang cocok dengan aturan. Anda dapat menentukan aturan yang menentukan tindakan target yang akan diambil berdasarkan AWS CloudShell tindakan yang direkam sebagai peristiwa dalam file CloudTrail log.

Misalnya, Anda dapat [membuat EventBridge aturan dengan AWS CLI](#) menggunakan `put-rule` perintah. `put-rule` Panggilan harus berisi setidaknya satu EventPattern atau ScheduleExpression. Aturan dengan EventPatterns dipicu ketika peristiwa yang cocok diamati. EventPattern Untuk AWS CloudShell acara:

```
{ "source": [ "aws.cloudshell" ], "detail-type": [ "AWS API Call via CloudTrail" ],
  "detail": { "eventSource": [ "cloudshell.amazonaws.com" ] } }
```

Untuk informasi selengkapnya, lihat [Peristiwa dan Pola Peristiwa EventBridge di](#) Panduan EventBridge Pengguna Amazon.

## Validasi kepatuhan untuk AWS CloudShell

Third-party auditor menilai keamanan dan kepatuhan AWS layanan sebagai bagian dari beberapa program AWS kepatuhan.

AWS CloudShell berada dalam lingkup dengan program kepatuhan berikut:

### SOC

AWS Laporan Sistem dan Kontrol Organisasi (SOC) adalah laporan pemeriksaan pihak ketiga independen yang menunjukkan bagaimana AWS mencapai kontrol dan tujuan kepatuhan utama.

Layanan	SDK	<a href="#">SOC 1,2,3</a>
AWS CloudShell	CloudShell	✓

### PCI

Standar Keamanan Data Industri Kartu Pembayaran (PCI DSS) adalah standar keamanan informasi eksklusif yang dikelola oleh Dewan Standar Keamanan PCI, yang didirikan oleh American Express, Discover Financial Services, JCB International, Worldwide dan Visa Inc. MasterCard

Layanan	SDK	<a href="#">PCI</a>
AWS CloudShell	CloudShell	✓

## Sertifikasi dan Layanan ISO dan CSA STAR

AWS memiliki sertifikasi untuk kepatuhan dengan ISO/IEC 27001:2013, 27017:2015, 27018:2019, 27701:2019, 22301:2019, 9001:2015, dan CSA STAR CCM v4.0.

Layanan	SDK	<a href="#">Sertifikasi dan Layanan ISO dan CSA STAR</a>
AWS CloudShell	CloudShell	✓

## FedRamp

Federal Risk and Authorization Management Program (FedRAMP) adalah program pemerintah AS yang memberikan pendekatan standar untuk penilaian keamanan, otorisasi, dan pemantauan berkelanjutan untuk produk dan layanan cloud.

Layanan	SDK	<a href="#">FedRAMP Sedang () East/West</a>	<a href="#">FedRAMP Tinggi () GovCloud</a>
AWS CloudShell	CloudShell	✓	✓

## DoD CC SRG

The Department of Defense (DoD) Cloud Computing Security Requirements Guide (SRG) menyediakan penilaian standar dan proses otorisasi untuk penyedia layanan cloud (CSP) untuk mendapatkan otorisasi sementara DoD, sehingga mereka dapat melayani pelanggan DoD.

Layanan yang melalui penilaian dan otorisasi DoD CC SRG akan memiliki status sebagai berikut:

- **Third-Party Penilaian Organisasi Penilaian (3PAO):** Layanan ini saat ini sedang menjalani penilaian oleh penilai pihak ketiga kami.

- Tinjauan Joint Authorization Board (JAB): Layanan ini saat ini sedang menjalani tinjauan JAB.
- Tinjauan Badan Sistem Informasi Pertahanan (DISA): Layanan ini saat ini sedang menjalani tinjauan DISA.

Layanan	SDK	<a href="#">DoD CC SRG IL2 () East/West</a>	<a href="#">DoD CC SRG IL2 () GovCloud</a>	<a href="#">DoD CC SRG IL4 () GovCloud</a>	<a href="#">DoD CC SRG IL5 () GovCloud</a>	<a href="#">DoD CC SRG IL6 (Wilayah Rahasia)A WS</a>
AWS CloudShell	CloudShell	✓	✓	✓	✓	N/A

## HIPAA BAA

Undang-Undang Portabilitas dan Akuntabilitas Asuransi Kesehatan 1996 (HIPAA) adalah undang-undang federal yang mewajibkan pembuatan standar nasional untuk melindungi informasi kesehatan pasien yang sensitif agar tidak diungkapkan tanpa persetujuan atau sepengetahuan pasien.

AWS memungkinkan entitas yang dilindungi dan rekan bisnis mereka yang tunduk pada HIPAA untuk memproses, menyimpan, dan mengirimkan informasi kesehatan yang dilindungi (PHI) dengan aman. Selain itu, pada Juli 2013, AWS menawarkan Adendum Asosiasi Bisnis standar (BAA) untuk pelanggan tersebut.

Layanan	SDK	<a href="#">HIPAA BAA</a>
AWS CloudShell	CloudShell	✓

## IRAP

Program Penilai Terdaftar Keamanan Informasi (IRAP) memungkinkan pelanggan Pemerintah Australia untuk memvalidasi bahwa kontrol yang sesuai telah ada dan menentukan model tanggung jawab yang sesuai untuk memenuhi persyaratan Manual Keamanan Informasi Pemerintah Australia (ISM) yang diproduksi oleh Australian Cyber Security Centre (ACSC).

Layanan	Ruang nama*	<a href="#">IRAP dilindungi</a>
AWS CloudShell	N/A	✓

\* Ruang nama membantu Anda mengidentifikasi layanan di seluruh lingkungan Anda. AWS Misalnya, saat Anda membuat kebijakan IAM, bekerja dengan Amazon Resource Names (ARN), dan membaca AWS CloudTrail log.

## MTCS

Multi-Tier Cloud Security (MTCS) adalah Standar manajemen keamanan operasional Singapura (SPRING SS 584), berdasarkan standar ISO 27001/02 Information Security Management System (ISMS).

Layanan	SDK	US-East(O hio)	US-East(N .Virginia)	US-West(O regon)	US-West(N .Californ ia)	Singapura	Seoul
AWS CloudShell	CloudShell	✓	✓	✓	N/A	N/A	N/A

## C5

Cloud Computing Compliance Controls Catalog (C5) adalah skema Government-backed pengesahan Jerman yang diperkenalkan di Jerman oleh Kantor Federal untuk Keamanan Informasi (BSI) untuk membantu organisasi menunjukkan keamanan operasional terhadap serangan cyber umum saat menggunakan layanan cloud dalam konteks “Rekomendasi Keamanan untuk Penyedia Cloud” Pemerintah Jerman.

Layanan	SDK	<a href="#">C5</a>
AWS CloudShell	CloudShell	✓

## ENS High

Skema akreditasi ENS (Esquema Nacional de Seguridad) telah dikembangkan oleh Kementerian Keuangan dan Administrasi Publik dan CCN (Pusat Kriptologi Nasional). Ini terdiri dari prinsip-prinsip dasar dan persyaratan minimum yang diperlukan untuk perlindungan informasi yang memadai.

Layanan	SDK	<a href="#">ENS Tinggi</a>
AWS CloudShell	CloudShell	✓

## FINMA

Otoritas Pengawas Pasar Keuangan Swiss (FINMA) adalah regulator pasar keuangan independen Swiss. AWS Keselarasan dengan persyaratan FINMA menunjukkan komitmen berkelanjutan kami untuk memenuhi harapan yang meningkat bagi penyedia layanan cloud yang ditetapkan oleh regulator dan pelanggan layanan keuangan Swiss.

Layanan	SDK	<a href="#">FINMA</a>
AWS CloudShell	CloudShell	✓

## PiTuKri

AWS Keselarasan dengan PiTuKri persyaratan menunjukkan komitmen berkelanjutan kami untuk memenuhi harapan yang meningkat untuk penyedia layanan cloud yang ditetapkan oleh Badan Transportasi dan Komunikasi Finlandia, Traficom.

Layanan	SDK	<a href="#">PiTuKri</a>
AWS CloudShell	CloudShell	✓

Untuk daftar AWS layanan yang berada dalam cakupan program kepatuhan tertentu, lihat [AWS Services in Scope by Compliance Program](#) . Untuk informasi umum, lihat [Program AWS Kepatuhan Program AWS](#) .

Anda dapat mengunduh laporan audit pihak ketiga dengan menggunakan AWS Artifact. Untuk informasi selengkapnya, lihat [Pengunduhan Laporan dalam AWS Artifact](#).

Tanggung jawab kepatuhan Anda saat menggunakan AWS CloudShell 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 Memulai](#) penerapan ini membahas pertimbangan arsitektur dan memberikan langkah-langkah untuk menerapkan lingkungan dasar yang berfokus pada keamanan dan berfokus pada kepatuhan. AWS
- [Arsitektur untuk HIPAA Security and Compliance Whitepaper](#) — Whitepaper ini menjelaskan bagaimana perusahaan dapat menggunakan untuk membuat aplikasi. AWS HIPAA-compliant
- [AWS Sumber Daya AWS](#) — Kumpulan buku kerja dan panduan ini mungkin berlaku untuk industri dan lokasi Anda.
- [Mengevaluasi Sumber Daya dengan Aturan](#) dalam Panduan AWS Config Pengembang — AWS Config Layanan menilai seberapa baik konfigurasi sumber daya Anda mematuhi praktik internal, pedoman industri, dan peraturan.
- [AWS Security Hub CSPM](#)— AWS Layanan ini memberikan pandangan komprehensif tentang keadaan keamanan Anda di dalamnya AWS yang membantu Anda memeriksa kepatuhan Anda terhadap standar industri keamanan dan praktik terbaik.

## Ketahanan di AWS CloudShell

Infrastruktur AWS global dibangun di sekitar AWS Wilayah dan Zona Ketersediaan. AWS Wilayah menyediakan beberapa Availability Zone yang terpisah secara fisik dan terisolasi, yang terhubung dengan latensi rendah, throughput tinggi, dan jaringan yang sangat redundan. Dengan Availability Zone, Anda dapat mendesain dan mengoperasikan aplikasi dan basis data yang secara otomatis mengalami kegagalan di antara zona tanpa gangguan. Zona Ketersediaan memiliki ketersediaan dan toleransi kesalahan yang lebih baik, dan dapat diskalakan dibandingkan infrastruktur pusat data tunggal atau multi tradisional.

Untuk informasi selengkapnya tentang AWS Wilayah dan Availability Zone, lihat [Infrastruktur AWS Global](#).

Selain infrastruktur AWS global, AWS CloudShell mendukung fitur berikut untuk mendukung ketahanan data dan kebutuhan cadangan Anda:

- Gunakan AWS CLI panggilan untuk menentukan file di direktori home Anda AWS CloudShell dan menambahkannya sebagai objek di bucket Amazon S3. Sebagai contoh, lihat [Memulai dengan AWS CloudShell](#).

## Keamanan infrastruktur di AWS CloudShell

Sebagai layanan terkelola, AWS CloudShell 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 AWS CloudShell melalui jaringan. Klien harus mendukung hal-hal berikut:

- Keamanan Lapisan Pengangkutan (TLS). Kami mensyaratkan TLS 1.2 dan menganjurkan TLS 1.3.
- Cipher suite dengan perfect forward secrecy (PFS) seperti DHE (Ephemeral) atau ECDHE (Elliptic Curve Ephemeral Diffie-Hellman). Diffie-Hellman Sebagian besar sistem modern seperti Java 7 dan versi lebih baru mendukung mode-mode ini.

### Note

Secara default, instal patch keamanan AWS CloudShell secara otomatis untuk paket sistem lingkungan komputasi Anda.

## Praktik terbaik keamanan untuk AWS CloudShell

Praktik terbaik berikut adalah pedoman umum dan tidak mewakili solusi keamanan yang lengkap. Karena praktik terbaik ini mungkin tidak sesuai atau cukup untuk lingkungan Anda, kami sarankan Anda memperlakukannya sebagai pertimbangan yang bermanfaat alih-alih resep.

Beberapa praktik terbaik keamanan untuk AWS CloudShell

- Gunakan izin dan kebijakan IAM untuk mengontrol akses ke AWS CloudShell dan memastikan pengguna hanya dapat melakukan tindakan tersebut (misalnya, mengunduh dan mengunggah

file) yang diperlukan oleh peran mereka. Untuk informasi selengkapnya, lihat [Mengelola AWS CloudShell akses dan penggunaan dengan kebijakan IAM](#).

- Jangan sertakan data sensitif dalam entitas IAM Anda seperti pengguna, peran, atau nama sesi.
- Fitur Keep Safe Paste diaktifkan untuk menangkap potensi risiko keamanan dalam teks yang telah Anda salin dari sumber eksternal. Safe Paste diaktifkan secara default. Untuk informasi selengkapnya tentang penggunaan tempel aman untuk teks multiline, lihat [Menggunakan Tempel Aman untuk teks multiline](#).
- Biasakan dengan [Model Tanggung Jawab Keamanan Bersama](#) jika Anda menginstal aplikasi pihak ketiga ke lingkungan komputasi. AWS CloudShell
- Siapkan mekanisme rollback sebelum mengedit skrip shell yang memengaruhi pengalaman shell pengguna. Untuk informasi selengkapnya tentang memodifikasi lingkungan shell default, lihat [Memodifikasi shell Anda dengan skrip](#).
- Simpan kode Anda dengan aman di sistem kontrol versi.

## AWS CloudShell FAQ Keamanan

Berikut ini adalah jawaban atas pertanyaan yang sering diajukan tentang keamanan untuk CloudShell.

- [Apa AWS proses dan teknologi digunakan saat Anda meluncurkan CloudShell dan memulai sesi shell?](#)
- [Apakah mungkin untuk membatasi akses jaringan? CloudShell](#)
- [Dapatkan saya menyesuaikan CloudShell lingkungan saya?](#)
- [Dimana saya \\$ RUMAH direktori sebenarnya disimpan di AWS Cloud?](#)
- [Apakah mungkin untuk mengenkripsi saya \\$ RUMAH direktori?](#)
- [Dapatkan saya menjalankan pemindaian virus pada saya \\$ RUMAH direktori?](#)
- [Bisakah saya membatasi CloudShell pengguna dari akses root di dalam wadah?](#)

### Apa AWS proses dan teknologi digunakan saat Anda meluncurkan CloudShell dan memulai sesi shell?

Saat masuk Konsol Manajemen AWS, Anda memasukkan kredensial pengguna IAM Anda. Dan, saat Anda meluncurkan CloudShell dari antarmuka konsol, kredensial ini digunakan dalam panggilan ke CloudShell API yang membuat lingkungan komputasi untuk layanan tersebut. AWS Systems

Manager Sesi kemudian dibuat untuk lingkungan komputasi, dan CloudShell mengirimkan perintah ke sesi itu.

[Kembali ke daftar FAQ keamanan](#)

## Apakah mungkin untuk membatasi akses jaringan? CloudShell

Untuk lingkungan publik, tidak mungkin membatasi akses jaringan. Jika Anda ingin membatasi akses jaringan, Anda harus mengaktifkan izin untuk membuat hanya lingkungan VPC dan menolak pembuatan lingkungan publik.

Untuk informasi selengkapnya, lihat [Memastikan bahwa pengguna hanya membuat lingkungan VPC dan menolak pembuatan lingkungan publik](#).

Untuk lingkungan CloudShell VPC, pengaturan jaringan diwarisi dari VPC Anda. Menggunakan CloudShell dalam VPC memungkinkan Anda untuk mengontrol akses jaringan lingkungan CloudShell VPC Anda.

[Kembali ke daftar FAQ keamanan](#)

## Dapatkah saya menyesuaikan CloudShell lingkungan saya?

Anda dapat mengunduh dan menginstal utilitas dan perangkat lunak pihak ketiga lainnya untuk CloudShell lingkungan Anda. Hanya perangkat lunak yang diinstal di \$HOME direktori Anda yang bertahan di antara sesi.

Sebagaimana didefinisikan oleh [model tanggung jawab AWS bersama](#), Anda bertanggung jawab atas konfigurasi dan pengelolaan aplikasi yang diperlukan yang Anda instal.

[Kembali ke daftar FAQ keamanan](#)

## Dimana saya \$ **RUMAH** direktori sebenarnya disimpan di AWS Cloud?

Untuk lingkungan Publik, infrastruktur untuk menyimpan data di Anda \$HOME disediakan oleh Amazon S3.

Untuk lingkungan VPC, \$HOME direktori Anda dihapus ketika waktu lingkungan VPC Anda habis (setelah 20-30 menit tidak aktif), atau ketika Anda menghapus atau memulai ulang lingkungan Anda.

[Kembali ke daftar FAQ keamanan](#)

## Apakah mungkin untuk mengenkripsi saya \$ RUMAH direktori?

Tidak, tidak mungkin mengenkripsi \$HOME direktori Anda dengan kunci Anda sendiri. Tetapi CloudShell mengenkripsi konten \$HOME direktori Anda saat menyimpannya di Amazon S3.

[Kembali ke daftar FAQ keamanan](#)

## Dapatkah saya menjalankan pemindaian virus pada saya \$ RUMAH direktori?

Saat ini, tidak mungkin menjalankan pemindaian virus \$HOME direktori Anda. Support untuk fitur ini sedang ditinjau.

[Kembali ke daftar FAQ keamanan](#)

## Dapatkah saya membatasi masuknya data atau keluar untuk saya? CloudShell

Untuk membatasi masuknya atau keluar, kami sarankan Anda menggunakan lingkungan VPC. CloudShell \$HOMEDirektori lingkungan VPC dihapus ketika waktu lingkungan VPC Anda habis (setelah 20-30 menit tidak aktif), atau ketika Anda menghapus atau memulai ulang lingkungan Anda. Di menu Tindakan, opsi unggah dan unduh tidak tersedia untuk lingkungan VPC.

[Kembali ke daftar FAQ keamanan](#)

## Bisakah saya membatasi CloudShell pengguna dari akses root di dalam wadah?

Tidak. AWS CloudShell menyediakan akses root dalam wadah komputasi berdasarkan desain. Wadah dalam CloudShell berfungsi sebagai pengemasan kode dan kenyamanan operasional. Mereka bukan batas keamanan.

CloudShell mengelola kontrol akses melalui AWS Identity and Access Management (IAM). Setiap CloudShell sesi menerima kredensial IAM sementara yang diputar secara teratur yang mencakup izin pengguna. Kredensial ini adalah batas keamanan, bukan wadah itu sendiri.

Karena container dan instance yang mendasarinya memiliki cakupan kredensi IAM yang sama, akses di luar batas penampung tidak memberikan izin tambahan. AWS

---

[Kembali ke daftar FAQ keamanan](#)

# AWS CloudShell lingkungan komputasi: spesifikasi dan perangkat lunak

Saat Anda meluncurkan AWS CloudShell, lingkungan komputasi yang didasarkan pada [Amazon Linux 2023](#) dibuat untuk meng-host pengalaman shell. Lingkungan dikonfigurasi dengan [sumber daya komputasi \(vCPU dan memori\)](#) dan menyediakan berbagai perangkat lunak [pra-instal](#) yang dapat diakses dari antarmuka baris perintah. Pastikan perangkat lunak apa pun yang Anda instal di lingkungan komputasi ditambah dan diperbarui. Anda juga dapat mengonfigurasi lingkungan default Anda dengan menginstal perangkat lunak dan memodifikasi skrip shell.

## Menghitung sumber daya lingkungan

Setiap lingkungan AWS CloudShell komputasi diberi CPU dan sumber daya memori berikut:

- 1 vCPU (unit pemrosesan pusat virtual)
- RAM 2-GiB

Dan, lingkungan disediakan dengan konfigurasi penyimpanan berikut:

- Penyimpanan persisten 1-GB (penyimpanan tetap ada setelah sesi berakhir)

Untuk informasi selengkapnya, lihat [Penyimpanan tetap](#).

## CloudShell persyaratan jaringan

### WebSockets

CloudShell tergantung pada WebSocket protokol, yang memungkinkan komunikasi interaktif dua arah antara browser web pengguna dan CloudShell layanan di AWS Cloud. Jika Anda menggunakan browser di jaringan pribadi, akses aman ke internet mungkin difasilitasi oleh server proxy dan firewall. WebSocket Komunikasi biasanya dapat melintasi server proxy tanpa masalah. Tetapi dalam beberapa kasus, server proxy WebSockets mencegah bekerja dengan benar. Jika masalah ini terjadi, CloudShell antarmuka Anda melaporkan kesalahan berikut: `Failed to open sessions : Timed out while opening the session.`

Jika kesalahan ini terjadi berulang kali, lihat dokumentasi untuk server proxy Anda untuk memastikan bahwa itu dikonfigurasi untuk mengizinkan WebSockets. Atau, Anda dapat menghubungi administrator sistem jaringan Anda.

#### Note

Jika Anda ingin menentukan izin granular dengan daftar izin tertentu URLs, Anda dapat menambahkan bagian dari URL yang digunakan AWS Systems Manager sesi untuk membuka WebSocket koneksi untuk mengirim input dan menerima output. (AWS CloudShell Perintah Anda dikirim ke sesi Systems Manager tersebut.)

Format untuk ini StreamUrl digunakan oleh Systems Manager adalah `wss://ssmmessages.region.amazonaws.com/v1/data-channel/session-id?stream=(input|output)`.

Wilayah ini mewakili pengenal Wilayah untuk AWS Wilayah yang didukung oleh AWS Systems Manager, seperti `us-east-2` untuk Wilayah Timur AS (Ohio).

Karena `session-id` dibuat setelah sesi Systems Manager tertentu berhasil dimulai, Anda hanya dapat menentukan `wss://ssmmessages.region.amazonaws.com` saat memperbarui daftar izin URL Anda. Untuk informasi selengkapnya, lihat [StartSession](#) operasi di Referensi AWS Systems Manager API.

## Perangkat lunak pra-instal

#### Note

Karena lingkungan AWS CloudShell pengembangan diperbarui secara berkala untuk menyediakan akses ke perangkat lunak terbaru, kami tidak memberikan nomor versi tertentu dalam dokumentasi ini. Sebagai gantinya, kami menjelaskan bagaimana Anda dapat memeriksa versi mana yang diinstal. Untuk memeriksa versi yang diinstal, masukkan nama program diikuti dengan `--version` opsi (misalnya, `git --version`).

## Kerang

### Cangkang pra-instal

Nama	Deskripsi	Informasi versi
Bash	Shell Bash adalah aplikasi shell default untuk AWS CloudShell.	<code>bash --version</code>
PowerShell (pwsh)	Menawarkan antarmuka baris perintah dan dukungan bahasa scripting, PowerShell dibangun di atas Microsoft .NET Command Language Runtime. PowerShell menggunakan perintah ringan cmdlets yang disebut yang menerima dan mengembalikan objek .NET.	<code>pwsh --version</code>
Z Shell (zsh)	Z Shell, juga dikenal sebagai zsh, adalah versi diperpanjang dari Bourne Shell yang menawarkan dukungan penyesuaian yang ditingkatkan untuk tema dan plugin.	<code>zsh --version</code>

## AWS antarmuka baris perintah (CLI)

### CLI

Nama	Deskripsi	Informasi versi
AWS CDK CLI Toolkit	AWS CDK Toolkit, perintah CLI <code>cdk</code> , adalah alat utama yang berinteraksi dengan	<code>cdk --version</code>

Nama	Deskripsi	Informasi versi
	<p>aplikasi Anda. AWS CDK Ini mengeksekusi aplikasi Anda, menginterogasi model aplikasi yang Anda tetapkan, dan menghasilkan serta menerapkan AWS CloudFormation template yang dihasilkan oleh. AWS CDK</p> <p>Untuk informasi selengkapnya, lihat <a href="#">AWS CDK Toolkit</a>.</p>	
AWS CLI	<p>AWS CLI Ini adalah antarmuka baris perintah yang dapat Anda gunakan untuk mengelola beberapa AWS layanan dari baris perintah dan mengotomatiskannya menggunakan skrip. Untuk informasi selengkapnya, lihat <a href="#">Kelola AWS layanan dari CLI di CloudShell</a>.</p> <p>Untuk informasi tentang cara memastikan bahwa Anda menggunakan up-to-date versi paling banyak AWS CLI versi 2, lihat <a href="#">Menginstal AWS CLI ke direktori home Anda</a>.</p>	<pre>aws --version</pre>

Nama	Deskripsi	Informasi versi
EB CLI	<p>AWS Elastic Beanstalk CLI menyediakan antarmuka baris perintah untuk menyederhanakan pembuatan, pembaruan, dan pemantauan lingkungan dari repositori lokal.</p> <p>Untuk informasi selengkapnya, lihat <a href="#">Menggunakan antarmuka baris perintah Elastic Beanstalk (EB CLI)</a> di Panduan Pengembang AWS Elastic Beanstalk</p>	<code>eb --version</code>
Amazon ECS CLI	<p>Antarmuka baris perintah Amazon Elastic Container Service (Amazon ECS) (CLI) menyediakan perintah tingkat tinggi untuk menyederhanakan pembuatan, pembaruan, dan pemantauan cluster dan tugas.</p> <p>Untuk informasi selengkapnya, lihat <a href="#">Menggunakan Antarmuka Baris Perintah Amazon ECS</a> di Panduan Pengembang Layanan Amazon Elastic Container.</p>	<code>ecs-cli --version</code>

Nama	Deskripsi	Informasi versi
AWS SAM CLI	<p>AWS SAM CLI adalah alat baris perintah yang beroperasi pada AWS Serverless Application Model template dan kode aplikasi. Anda dapat melakukan beberapa tugas. Ini termasuk menjalankan fungsi Lambda secara lokal, membuat paket penerapan untuk aplikasi tanpa server Anda, dan menerapkan aplikasi tanpa server Anda ke Cloud. AWS</p> <p>Untuk informasi selengkapnya, lihat <a href="#">referensi perintah AWS SAM CLI di Panduan AWS Serverless Application Model Pengembang</a>.</p>	<code>sam --version</code>

Nama	Deskripsi	Informasi versi
Alat AWS untuk PowerShell	<p>Alat AWS untuk PowerShell Ini adalah PowerShell modul yang dibangun di atas fungsionalitas yang diekspos oleh SDK untuk .NET. Dengan Alat AWS untuk PowerShell, Anda dapat membuat skrip operasi pada AWS sumber daya Anda dari baris PowerShell perintah.</p> <p>AWS CloudShell pra-instal versi termodulasi (AWS.tools) dari. Alat AWS untuk PowerShell</p> <p>Untuk informasi selengkapnya, lihat <a href="#">Menggunakan Alat AWS untuk PowerShell</a> di Panduan Alat AWS untuk PowerShell Pengguna.</p>	<pre>psh --Command 'Get-AWSPowerShell Version'</pre>

## Runtime dan AWS SDKs: Node.js dan Python 3

### Runtime dan AWS SDKs

Nama	Deskripsi	Informasi versi
Node.js (dengan npm)	<p>Node.js adalah JavaScript runtime yang dirancang untuk memudahkan penerapan teknik pemrograman asinkron. Untuk informasi selengkapnya, lihat <a href="#">dokumentasi di situs resmi Node.js</a>.</p>	<ul style="list-style-type: none"> <li>• Node.js: <code>node --version</code></li> <li>• npm: <code>npm --version</code></li> </ul>

Nama	Deskripsi	Informasi versi
	<p>npm adalah manajer paket yang menyediakan akses ke registri JavaScript modul online. Untuk informasi lebih lanjut, lihat <a href="#">dokumentasi di situs resmi npm</a>.</p>	
SDK untuk JavaScript di Node.js	<p>Kit pengembangan perangkat lunak (SDK) membantu menyederhanakan pengembangan dengan menyediakan JavaScript objek untuk layanan AWS termasuk Amazon S3, Amazon EC2, DynamoDB, dan Amazon SWF. Lihat informasi selengkapnya di <a href="#">Panduan Developer AWS SDK untuk JavaScript</a>.</p>	<pre>npm -g ls --depth 0 2&gt;/dev/null   grep aws-sdk</pre>

Nama	Deskripsi	Informasi versi
Python	<p>Python 3 siap digunakan di lingkungan shell. Python 3 sekarang dianggap sebagai versi default dari bahasa pemrograman (dukungan untuk Python 2 berakhir pada Januari 2020). Untuk informasi selengkapnya, lihat <a href="#">dokumentasi di situs resmi Python</a>.</p> <p>Juga, pra-instal adalah pip, penginstal paket untuk Python. Anda dapat menggunakan program baris perintah ini untuk menginstal paket Python dari indeks online seperti Indeks Paket Python. Untuk informasi selengkapnya, lihat <a href="#">dokumentasi yang disediakan oleh Otoritas Kemasan Python</a>.</p>	<ul style="list-style-type: none"><li>• Python 3: <code>python3 --version</code></li><li>• pip: <code>pip3 --version</code></li></ul>

Nama	Deskripsi	Informasi versi
SDK untuk Python (Boto3)	<p>Boto adalah perangkat pengembangan perangkat lunak (SDK) yang digunakan pengembang Python untuk membuat, mengkonfigurasi, dan mengelola Layanan AWS, seperti Amazon EC2 dan Amazon S3. SDK menyediakan API berorientasi objek easy-to-use, serta akses tingkat rendah ke Layanan AWS</p> <p>Untuk informasi lebih lanjut, lihat dokumentasi <a href="#">Boto3</a>.</p>	<code>pip3 list   grep boto3</code>

## Alat pengembangan dan utilitas shell

### Alat pengembangan dan utilitas shell

Nama	Deskripsi	Informasi versi
penyelesaian bash-	<p>bash-completion adalah kumpulan fungsi shell yang memungkinkan pelengkapan otomatis perintah atau argumen yang diketik sebagian dengan menekan tombol Tab. Anda dapat menemukan paket yang didukung bash-completion.</p> <p><code>/usr/share/bash-completion/completions</code></p>	<code>dnf info bash-completion</code>

Nama	Deskripsi	Informasi versi
	<p>Untuk mengatur pelengkapan otomatis untuk perintah paket, file program harus bersumber . Misalnya, untuk menyiapkan pelengkapan otomatis untuk perintah Git, tambahkan baris berikut .bashrc agar fitur tersedia setiap kali AWS CloudShell sesi Anda dimulai:</p> <pre>source /usr/share/ bash-completion/ completions/git</pre> <p>Jika Anda ingin menggunakan skrip penyelesaian kustom, tambahkan skrip tersebut ke direktori home persisten (\$HOME) Anda dan masukkan langsung ke dalam .bashrc.</p> <p>Untuk informasi selengkapnya, lihat halaman <a href="#">README</a> proyek di GitHub.</p>	

Nama	Deskripsi	Informasi versi
ekspansi cqlsh	<p>cqlsh-expansion adalah toolkit yang mencakup cqlsh dan pembantu yang telah dikonfigurasi sebelumnya untuk Amazon Keyspaces sambil mempertahankan kompatibilitas penuh dengan Apache Cassandra. Untuk informasi selengkapnya, lihat <a href="#">Menggunakan cqlsh untuk terhubung ke Amazon Keyspaces di Amazon Keyspaces</a> (untuk Apache Cassandra) Panduan Pengembang.</p>	<pre>cqlsh-expansion --version</pre>

Nama	Deskripsi	Informasi versi
Docker	<p><a href="#">Docker</a> adalah platform terbuka untuk mengembankan, mengirim, dan menjalankan aplikasi. Docker memungkinkan Anda untuk memisahkan aplikasi Anda dari infrastruktur Anda sehingga Anda dapat mengirimkan perangkat lunak dengan cepat. Ini memungkinkan Anda untuk membangun Dockerfiles di dalam AWS CloudShell, dan membangun aset Docker dengan CDK. Untuk informasi tentang AWS Wilayah mana yang didukung dengan Docker, lihat <a href="#">AWS Wilayah yang Didukung untuk AWS CloudShell</a>. Anda harus menyadari bahwa Docker memiliki ruang terbatas di lingkungan. Jika Anda memiliki gambar individual yang besar, atau terlalu banyak gambar Docker yang sudah ada sebelumnya, itu dapat menyebabkan masalah. Untuk informasi selengkapnya tentang Docker, lihat panduan <a href="#">Dokumentasi Docker</a>.</p>	<code>docker --version</code>

Nama	Deskripsi	Informasi versi
Git	Git adalah sistem kontrol versi terdistribusi yang mendukung praktik pengembangan perangkat lunak modern melalui alur kerja cabang dan pementasan konten. Untuk informasi selengkapnya, lihat <a href="#">halaman dokumentasi di situs resmi Git</a> .	<code>git --version</code>
iputil	Paket iputils berisi utilitas untuk jaringan Linux. Untuk informasi lebih lanjut tentang utilitas yang disediakan, lihat repositori <a href="#">iputils</a> di GitHub	Contoh untuk alat iputils: <code>arping -V</code>
jq	Utilitas jq mem-parsing data berformat JSON untuk menghasilkan output yang dimodifikasi oleh filter baris perintah. Untuk informasi lebih lanjut, lihat <a href="#">manual jq yang dihosting di GitHub</a> .	<code>jq --version</code>
kubectl	kubectl adalah alat baris perintah untuk berkomunikasi dengan control plane kluster Kubernetes, menggunakan API Kubernetes.	<code>kubectl --version</code>

Nama	Deskripsi	Informasi versi
membuat	Utilitas make digunakan <code>makefiles</code> untuk mengotomatiskan set tugas dan mengatur kompilasi kode. Untuk informasi selengkapnya, lihat <a href="#">dokumentasi GNU Make</a> .	<code>make --version</code>
pria	Perintah <code>man</code> menyediakan halaman manual untuk utilitas dan alat baris perintah. Misalnya, <code>man ls</code> mengembalikan halaman manual untuk <code>ls</code> perintah yang mencantumkan isi direktori. Untuk informasi lebih lanjut, lihat <a href="#">entri Wikipedia di halaman manual</a> .	<code>man --version</code>
nano	<code>nano</code> adalah editor kecil dan user-friendly untuk antarmuka berbasis teks. Untuk informasi lebih lanjut, lihat dokumentasi <a href="#">GNU nano</a> .	<code>nano --version</code>

Nama	Deskripsi	Informasi versi
OpenJDK 21	<p><a href="#">Amazon Corretto 21</a> adalah <a href="#">distribusi Jangka Panjang yang Didukung (LTS)</a> dari <a href="#">OpenJDK 21</a>. Amazon Corretto adalah distribusi tanpa biaya, multiplatform, siap produksi dari Open Java Development Kit (OpenJDK) . Untuk informasi lebih lanjut, lihat <a href="#">Apa itu Amazon Corretto 21?</a> di Panduan Pengguna Corretto 21.</p>	java -version
procs	<p>procs adalah utilitas administrasi sistem yang dapat Anda gunakan untuk memantau dan menghentikan proses yang sedang berjalan. Untuk informasi selengkapnya, lihat <a href="#">file README yang mencantumkan program yang dapat dijalankan dengan procs</a>.</p>	ps --version
psql	<p>PostgreSQL adalah sistem database open source yang kuat yang menggunakan kemampuan SQL standar sambil menyediakan fitur yang kuat untuk mengelola dan menskalakan operasi data yang kompleks dengan aman. Untuk informasi lebih lanjut, lihat <a href="#">Apa itu PostgreSQL</a>.</p>	psql --version

Nama	Deskripsi	Informasi versi
Klien SSH	Klien SSH menggunakan protokol shell aman untuk komunikasi terenkripsi dengan komputer jarak jauh. OpenSSH adalah klien SSH yang sudah diinstal sebelumnya. Untuk informasi selengkapnya, lihat situs <a href="#">OpenSSH yang dikelola oleh OpenBSD</a> .	<code>ssh -V</code>
sudo	Dengan utilitas sudo, pengguna dapat menjalankan program dengan izin keamanan pengguna lain, biasanya superuser. Sudo berguna ketika Anda perlu menginstal aplikasi sebagai administrator sistem. Untuk informasi lebih lanjut, lihat <a href="#">Manual Sudo</a> .	<code>sudo --version</code>
tar	tar adalah utilitas baris perintah yang dapat Anda gunakan untuk mengelompokkan beberapa file dalam satu file arsip (sering disebut tarball). Untuk informasi lebih lanjut, lihat <a href="#">dokumentasi tar GNU</a> .	<code>tar --version</code>

Nama	Deskripsi	Informasi versi
tmux	tmux adalah multiplexer terminal yang dapat Anda gunakan untuk menjalankan berbagai program secara simultan di beberapa jendela. Untuk informasi lebih lanjut, lihat <a href="#">blog yang menyediakan pengantar singkat untuk tmux</a> .	tmux -V
vim	vim adalah editor yang dapat disesuaikan yang dapat Anda berinteraksi melalui antarmuka berbasis teks. Untuk informasi selengkapnya, lihat <a href="#">sumber dokumentasi yang disediakan di vim.org</a> .	vim --version
wget	wget adalah program komputer yang digunakan untuk mengambil konten dari server web yang ditentukan oleh titik akhir di baris perintah. Untuk informasi lebih lanjut, lihat dokumentasi <a href="#">GNU Wget</a> .	wget --version

Nama	Deskripsi	Informasi versi
zip/unzip	zip/unzip Utilitas menggunakan format file arsip yang memberikan kompresi data lossless tanpa kehilangan data. Panggil perintah zip untuk mengelompokkan dan mengompres file dalam satu arsip. Gunakan unzip untuk mengekstrak file dari arsip ke direktori tertentu.	<code>unzip --version</code> <code>zip --version</code>

## Menginstal AWS CLI ke direktori home Anda

Seperti perangkat lunak lainnya yang sudah diinstal sebelumnya di CloudShell lingkungan Anda, AWS CLI alat ini diperbarui secara otomatis dengan peningkatan terjadwal dan tambalan keamanan. Jika Anda ingin memastikan bahwa Anda memiliki up-to-date versi terbanyak AWS CLI, Anda dapat memilih untuk menginstal alat secara manual di direktori home shell.

### Important

Anda perlu menginstal salinan Anda secara manual AWS CLI di direktori home sehingga tersedia saat berikutnya Anda memulai CloudShell sesi. Instalasi ini diperlukan karena file yang ditambahkan ke direktori di luar \$HOME dihapus setelah Anda menyelesaikan sesi shell. Juga, setelah Anda menginstal salinan ini AWS CLI, itu tidak diperbarui secara otomatis. Dengan kata lain, Anda bertanggung jawab untuk mengelola pembaruan dan patch keamanan.

Untuk informasi selengkapnya tentang Model Tanggung Jawab AWS Bersama, lihat [Perlindungan data di AWS CloudShell](#).

Untuk menginstal AWS CLI

1. Di baris CloudShell perintah, gunakan `curl` perintah untuk mentransfer salinan zip yang AWS CLI diinstal ke shell:

```
curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"
```

2. Buka zip folder zip:

```
unzip awscliv2.zip
```

3. Untuk menambahkan alat ke folder tertentu, jalankan AWS CLI installer:

```
sudo ./aws/install --install-dir /home/cloudshell-user/usr/local/aws-cli --bin-dir /home/cloudshell-user/usr/local/bin
```

Jika berhasil diinstal, baris perintah menampilkan pesan berikut:

```
You can now run: /home/cloudshell-user/usr/local/bin/aws --version
```

4. Untuk kenyamanan Anda sendiri, kami sarankan Anda juga memperbarui variabel PATH lingkungan sehingga Anda tidak perlu menentukan jalur ke instalasi alat saat menjalankan aws perintah:

```
export PATH=/home/cloudshell-user/usr/local/bin:$PATH
```

#### Note

Jika Anda membatalkan perubahan iniPATH, aws perintah yang tidak menampilkan jalur tertentu menggunakan versi pra-instal secara default AWS CLI .

## Menginstal perangkat lunak pihak ketiga di lingkungan shell Anda

#### Note

Kami menyarankan Anda meninjau [Model Tanggung Jawab Keamanan Bersama](#) sebelum Anda menginstal aplikasi pihak ketiga ke AWS CloudShell lingkungan komputasi.

Secara default, semua AWS CloudShell pengguna memiliki izin sudo. Oleh karena itu, Anda dapat menggunakan sudo perintah untuk menginstal perangkat lunak yang belum tersedia di lingkungan

komputasi shell. Misalnya, Anda dapat menggunakan sudo dengan utilitas manajemen paket DNF untuk menginstal cowsay, yang menghasilkan gambar seni ASCII sapi dengan pesan:

```
sudo dnf install cowsay
```

Anda kemudian dapat meluncurkan program yang baru diinstal dengan mengetikkan "Welcome to AWS CloudShell" | cowsay.

#### Important

Package mengelola utilitas seperti dnf install program di direktori `/usr/bin` (, misalnya), yang didaur ulang ketika sesi shell Anda berakhir. Ini berarti perangkat lunak tambahan diinstal dan digunakan berdasarkan per sesi.

## Memodifikasi shell Anda dengan skrip

Jika Anda ingin memodifikasi lingkungan shell default, Anda dapat mengedit skrip shell yang berjalan setiap kali lingkungan shell dimulai. `.bashrc` Skrip berjalan setiap kali shell bash default dimulai.

#### Warning

Jika Anda salah memodifikasi `.bashrc` file Anda, Anda mungkin tidak dapat mengakses lingkungan shell Anda sesudahnya. Ini praktik yang baik untuk membuat salinan file sebelum mengedit. Anda juga dapat mengurangi risiko dengan membuka dua shell saat mengedit. `.bashrc` Jika Anda kehilangan akses di satu shell, Anda masih masuk ke shell lain dan dapat memutar kembali perubahan apa pun.

Jika Anda kehilangan akses setelah salah memodifikasi `.bashrc` atau file lainnya, Anda dapat kembali AWS CloudShell ke pengaturan default dengan [menghapus direktori home Anda](#).

Dalam prosedurnya, Anda akan memodifikasi `.bashrc` skrip sehingga lingkungan shell Anda beralih secara otomatis untuk menjalankan shell Z.

1. Buka `.bashrc` menggunakan editor teks (Vim, misalnya):

```
vim .bashrc
```

2. Di antarmuka editor, tekan tombol I untuk mulai mengedit, lalu tambahkan yang berikut ini:

```
zsh
```

3. Untuk keluar dan menyimpan `.bashrc` file yang diedit, tekan Esc untuk masuk ke mode perintah Vim dan masukkan yang berikut:

```
:wq
```

4. Gunakan `source` perintah untuk memuat ulang `.bashrc` file:

```
source .bashrc
```

Ketika antarmuka baris perintah tersedia lagi, simbol prompt telah berubah `%` untuk menunjukkan bahwa Anda sekarang menggunakan shell Z.

## AWS CloudShell bermigrasi dari ke AL2 AL2023

AWS CloudShell, yang didasarkan pada Amazon Linux 2 (AL2), telah bermigrasi ke Amazon Linux 2023 (AL2023). Untuk informasi selengkapnya AL2023, lihat [Apa itu Amazon Linux 2023 \(AL2023\)](#) di Panduan Pengguna Amazon Linux 2023.

Dengan AL2023, Anda dapat terus mengakses CloudShell lingkungan yang ada dengan semua alat yang disediakan oleh CloudShell. Untuk informasi selengkapnya tentang alat yang tersedia, lihat [Perangkat lunak pra-instal](#).

AL2023 menyediakan beberapa perbaikan pada alat pengembangan, termasuk versi paket yang lebih baru Node seperti.js 18 dan Python 3.9.

### Note

Di AL2023, Python 2 tidak lagi dikirim dengan CloudShell lingkungan Anda.

Untuk informasi selengkapnya tentang perbedaan utama antara AL2 dan AL2023, lihat [Membandingkan Amazon Linux 2 dan Amazon Linux 2023](#) di Panduan Pengguna Amazon Linux 2023.

Jika Anda memiliki pertanyaan, hubungi [Dukungan](#). Anda juga dapat mencari jawaban dan memposting pertanyaan di [AWS re:Post](#). Saat Anda masuk AWS re:Post, Anda mungkin diminta untuk masuk AWS.

## AWS CloudShell Migrasi FAQs

Berikut ini adalah jawaban atas beberapa pertanyaan umum tentang migrasi dari AL2 ke AL2023 dengan AWS CloudShell.

- [Apakah migrasi akan AL2023 memengaruhi AWS sumber daya saya yang lain, seperti instans Amazon EC2 yang sedang berjalan? AL2](#)
- [Paket apa yang akan diubah dengan migrasi ke AL2023?](#)
- [Bisakah saya memilih keluar dari migrasi?](#)
- [Bisakah saya membuat cadangan AWS CloudShell lingkungan saya?](#)

### Apakah migrasi akan AL2023 memengaruhi AWS sumber daya saya yang lain, seperti instans Amazon EC2 yang sedang berjalan? AL2

Tidak ada layanan atau sumber daya selain AWS CloudShell lingkungan Anda yang terpengaruh oleh migrasi ini. Ini termasuk sumber daya yang mungkin telah Anda buat atau akses dari dalam AWS CloudShell. Misalnya, jika Anda telah membuat instans Amazon EC2 yang berjalan, AL2 ini tidak akan dimigrasikan ke AL2023.

### Apa paket yang telah diubah dengan migrasi ke AL2023?

AWS CloudShell lingkungan saat ini termasuk perangkat lunak pra-instal. Untuk mempelajari tentang daftar lengkap perangkat lunak pra-instal, lihat Perangkat lunak [pra-instal](#). AWS CloudShell akan terus mengirimkan paket-paket ini, dengan pengecualian Python 2. Untuk perbedaan lengkap antara paket yang disediakan oleh AL2 dan AL2023, lihat [Membandingkan AL2 dan AL2023](#). Untuk pelanggan dengan persyaratan paket dan versi tertentu yang tidak lagi terpenuhi setelah migrasi ke AL2023, kami sarankan menghubungi AWS Support untuk mengirimkan permintaan.

### Bisakah saya memilih keluar dari migrasi?

Tidak, Anda tidak dapat memilih keluar dari migrasi. AWS CloudShell lingkungan dikelola oleh AWS, oleh karena itu, semua lingkungan telah ditingkatkan ke AL2023.

## Bisakah saya membuat cadangan AWS CloudShell lingkungan saya?

AWS CloudShell akan terus mempertahankan direktori home pengguna. Untuk informasi selengkapnya, lihat [Kuota dan batasan layanan untuk AWS CloudShell](#). Jika Anda memiliki file atau konfigurasi yang disimpan di folder rumah Anda dan jika Anda ingin membuat cadangan untuk hal yang sama, selesaikan [Langkah 6: Buat cadangan direktori home](#).

# Pemecahan Masalah AWS CloudShell

Saat menggunakan AWS CloudShell, Anda mungkin mengalami masalah, seperti ketika Anda meluncurkan CloudShell atau melakukan tugas-tugas utama menggunakan antarmuka baris perintah shell. Informasi yang tercakup dalam Bab ini mencakup cara memecahkan masalah beberapa masalah umum yang mungkin Anda temui.

Untuk jawaban atas berbagai pertanyaan tentang CloudShell, lihat [AWS CloudShell FAQ](#). Anda juga dapat mencari jawaban dan memposting pertanyaan di [Forum AWS CloudShell Diskusi](#). Ketika Anda memasuki forum ini, Anda mungkin diminta untuk masuk AWS. Anda juga dapat [kontak kami](#) secara langsung.

## Memecahkan masalah kesalahan

Ketika Anda menemukan salah satu kesalahan terindeks berikut, Anda dapat menggunakan solusi berikut untuk mengatasi kesalahan ini.

### Topik

- [Akses ditolak](#)
- [Izin tidak cukup](#)
- [Tidak dapat mengakses AWS CloudShell baris perintah](#)
- [Tidak dapat melakukan ping ke alamat IP eksternal](#)
- [Ada beberapa masalah dalam mempersiapkan terminal Anda](#)
- [Tombol panah tidak berfungsi dengan benar di PowerShell](#)
- [Soket Web yang tidak didukung menyebabkan kegagalan untuk memulai sesi CloudShell](#)
- [Tidak dapat mengimpor AWSPowerShell.NetCore modul](#)
- [Docker tidak berjalan saat menggunakan AWS CloudShell](#)
- [Docker kehabisan ruang disk](#)
- [docker push habis waktu dan terus mencoba lagi](#)
- [Tidak dapat mengakses sumber daya dalam VPC dari saya AWS CloudShell Lingkungan VPC](#)
- [ENI yang digunakan oleh AWS CloudShell karena lingkungan VPC saya tidak dibersihkan](#)
- [Pengguna dengan CreateEnvironment izin hanya untuk lingkungan VPC juga memiliki akses ke publik AWS CloudShell lingkungan](#)

## Akses ditolak

Masalah: Ketika Anda mencoba meluncurkan CloudShell dari Konsol Manajemen AWS, Anda mendapatkan pesan "Tidak dapat memulai lingkungan. Untuk mencoba lagi, segarkan browser atau restart dengan memilih Actions, Restart AWS CloudShell". Anda ditolak akses bahkan setelah Anda memiliki izin yang diperlukan dari administrator IAM Anda dan Anda telah menyegarkan browser Anda atau memulai ulang. CloudShell

Solusi: Hubungi [AWS Support](#).

[\(kembali ke atas\)](#)

## Izin tidak cukup

Masalah: Saat Anda mencoba meluncurkan CloudShell dari Konsol Manajemen AWS, Anda mendapatkan pesan "Tidak dapat memulai lingkungan. Anda tidak memiliki izin yang diperlukan. Minta administrator IAM Anda untuk memberikan akses ke AWS CloudShell". Anda ditolak akses dan diberi tahu bahwa Anda tidak memiliki izin yang diperlukan.

Penyebab: Identitas IAM yang Anda gunakan untuk mengakses AWS CloudShell tidak memiliki izin IAM yang diperlukan.

Solusi: Minta administrator IAM Anda untuk memberi Anda izin yang diperlukan. Mereka dapat melakukan ini baik dengan menambahkan kebijakan AWS terkelola terlampir (AWSCloudShellFullAccess) atau kebijakan inline yang disematkan. Untuk informasi selengkapnya, lihat [Mengelola AWS CloudShell akses dan penggunaan dengan kebijakan IAM](#).

[\(kembali ke atas\)](#)

## Tidak dapat mengakses AWS CloudShell baris perintah

Masalah: Setelah memodifikasi file yang digunakan lingkungan komputasi, Anda tidak dapat mengakses baris perintah. AWS CloudShell

Solusi: Jika Anda kehilangan akses setelah salah memodifikasi `.bashrc` atau file lainnya, Anda dapat kembali AWS CloudShell ke pengaturan default dengan [menghapus direktori home Anda](#).

[\(kembali ke atas\)](#)

## Tidak dapat melakukan ping ke alamat IP eksternal

Masalah: Ketika Anda menjalankan perintah ping dari baris perintah (misalnya, `ping amazon.com`), Anda menerima pesan berikut.

```
ping: socket: Operation not permitted
```

Penyebab: Utilitas ping menggunakan Internet Control Message Protocol (ICMP) untuk mengirim paket permintaan gema ke host target. Ia menunggu gema untuk membalas dari target. Karena protokol ICMP tidak diaktifkan AWS CloudShell, utilitas ping tidak beroperasi di lingkungan komputasi shell.

Solusi: Karena ICMP tidak didukung AWS CloudShell, Anda dapat menjalankan perintah berikut untuk menginstal Netcat. Netcat adalah utilitas jaringan komputer untuk membaca dari, dan menulis ke, koneksi jaringan menggunakan TCP atau UDP.

```
sudo yum install nc
nc -zv www.amazon.com 443
```

[\(kembali ke atas\)](#)

## Ada beberapa masalah dalam mempersiapkan terminal Anda

Masalah: Saat mencoba mengakses AWS CloudShell menggunakan browser Microsoft Edge, Anda tidak dapat memulai sesi shell, dan browser menampilkan pesan kesalahan.

Penyebab: AWS CloudShell tidak kompatibel dengan versi Microsoft Edge sebelumnya. Anda dapat mengakses AWS CloudShell menggunakan empat versi utama terbaru dari browser yang didukung.

Solusi: Instal versi terbaru browser Edge dari [situs Microsoft](#).

[\(kembali ke atas\)](#)

## Tombol panah tidak berfungsi dengan benar di PowerShell

Masalah: Dalam operasi normal, Anda dapat menggunakan tombol panah untuk menavigasi antarmuka baris perintah dan memindai mundur dan maju melalui riwayat perintah Anda. Tapi, ketika Anda menekan tombol panah dalam versi tertentu dari PowerShell on AWS CloudShell, huruf mungkin salah dikeluarkan.

**Penyebab:** Situasi di mana tombol panah salah menampilkan huruf adalah masalah yang diketahui dengan versi PowerShell 7.2.x yang berjalan di Linux.

**Solusi:** Untuk menghapus urutan escape yang mengubah perilaku tombol panah, edit file PowerShell profil dan atur `$PSStyle` variabel ke `PlainText`.

1. Di baris AWS CloudShell perintah, masukkan perintah berikut untuk membuka file profil.

```
vim ~/.config/powershell/Microsoft.PowerShell_profile.ps1
```

#### Note

Jika Anda sudah masuk PowerShell, Anda juga dapat membuka file profil di editor dengan perintah berikut.

```
vim $PROFILE
```

2. Di editor, pergi ke akhir teks file yang ada, tekan `i` untuk masuk ke mode Sisipkan, dan kemudian tambahkan pernyataan berikut.

```
$PSStyle.OutputRendering = 'PlainText'
```

3. Setelah Anda melakukan pengeditan, tekan `Esc` untuk masuk ke mode perintah. Selanjutnya, masukkan perintah berikut untuk menyimpan file dan keluar dari editor.

```
:wq
```

#### Note

Perubahan Anda akan berlaku saat berikutnya Anda memulai PowerShell.

[\(kembali ke atas\)](#)

## Soket Web yang tidak didukung menyebabkan kegagalan untuk memulai sesi CloudShell

Masalah: Saat Anda mencoba memulai AWS CloudShell, Anda berulang kali menerima pesan berikut: `Failed to open sessions : Timed out while opening the session.`

Penyebab: CloudShell tergantung pada WebSocket protokol, yang memungkinkan komunikasi interaktif dua arah antara browser web Anda dan AWS CloudShell. Jika Anda menggunakan browser di jaringan pribadi, akses aman ke internet mungkin difasilitasi oleh server proxy dan firewall. WebSocket Komunikasi biasanya dapat melintasi server proxy tanpa masalah. Namun, dalam beberapa kasus, server proxy WebSockets mencegah agar tidak berfungsi dengan benar. Jika masalah ini terjadi, tidak CloudShell dapat memulai sesi shell dan upaya untuk terhubung akhirnya habis.

Solusi: Batas waktu koneksi mungkin disebabkan oleh masalah selain tidak WebSockets didukung. Jika ini masalahnya, pertama-tama segarkan jendela browser tempat antarmuka baris CloudShell perintah berada.

Jika Anda masih mendapatkan kesalahan batas waktu setelah penyegaran, lihat dokumentasi untuk server proxy Anda. Dan, pastikan bahwa server proxy Anda dikonfigurasi untuk mengizinkan Web Sockets. Atau, hubungi administrator sistem jaringan Anda.

### Note

Katakanlah Anda ingin menentukan izin granular dengan mengizinkan daftar URL tertentu. Anda dapat menambahkan bagian dari URL yang digunakan AWS Systems Manager sesi untuk membuka WebSocket koneksi untuk mengirim input dan menerima output. AWS CloudShell Perintah Anda dikirim ke sesi Systems Manager tersebut.

Format untuk ini StreamUrl yang digunakan oleh Systems Manager adalah `wss://ssmmessages.region.amazonaws.com/v1/data-channel/session-id?stream=(input|output)`.

Wilayah mewakili pengenal Wilayah untuk Wilayah AWS yang didukung oleh AWS Systems Manager. Misalnya, `us-east-2` adalah pengenal Wilayah untuk Wilayah Timur AS (Ohio). Karena `session-id` dibuat setelah sesi Systems Manager tertentu berhasil dimulai, Anda hanya dapat menentukan `wss://ssmmessages.region.amazonaws.com` kapan Anda memperbarui daftar izin URL Anda. Untuk informasi selengkapnya, lihat [StartSession](#) operasi di Referensi AWS Systems Manager API.

[\(kembali ke atas\)](#)

## Tidak dapat mengimpor **AWSPowerShell.NetCore** modul

Masalah: Saat Anda mengimpor `AWSPowerShell.NetCore` `Import-Module -Name AWSPowerShell.NetCore`, Anda menerima pesan galat berikut: PowerShell

Import-Module: Modul yang ditentukan 'AWSPowerShell.NetCore' tidak dimuat karena tidak ada file modul yang valid ditemukan di direktori modul apa pun.

Penyebab: `AWSPowerShell.NetCore` Modul digantikan oleh modul `AWS.Tools` per layanan di AWS CloudShell

Solusi: Setiap pernyataan impor eksplisit mungkin tidak lagi diperlukan atau perlu diubah ke modul `.Tools` per layanan terkait AWS.

Example

Example

- Untuk kebanyakan kasus, selama tidak ada jenis `.Net` yang digunakan, Anda tidak memerlukan pernyataan impor eksplisit. Berikut ini adalah contoh pernyataan impor.
  - `Get-S3Bucket`
  - `(Get-EC2Instance).Instances`
- Jika jenis `.Net` digunakan, impor modul tingkat layanan `()AWS.Tools.<Service>`. Berikut ini adalah contoh sintaks.

```
Import-Module -Name AWS.Tools.EC2
$instanceTag = [Amazon.EC2.Model.Tag]::new("Environment","Dev")
```

```
Import-Module -Name AWS.Tools.S3
$lifecycleRule = [Amazon.S3.Model.LifecycleRule]::new()
```

Untuk informasi lebih lanjut, lihat [pengumuman versi 4](#) untuk Alat AWS untuk PowerShell.

[\(kembali ke atas\)](#)

## Docker tidak berjalan saat menggunakan AWS CloudShell

Masalah: Docker tidak berjalan dengan baik saat menggunakan AWS CloudShell. Anda menerima pesan galat berikut:`docker: Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?`

Solusi: Coba mulai ulang lingkungan Anda. Pesan kesalahan ini dapat terjadi ketika Anda menjalankan Docker di AWS CloudShell dalam GovCloud Wilayah. Pastikan Anda menjalankan Docker di AWS Wilayah yang didukung. Untuk daftar Wilayah di mana Docker tersedia, lihat [AWS Wilayah yang Didukung untuk AWS CloudShell](#).

## Docker kehabisan ruang disk

Masalah: Anda menerima pesan kesalahan berikut:`ERROR: failed to solve: failed to register layer: write [...]: no space left on device.`

Penyebab: Dockerfile melebihi ruang disk yang tersedia di AWS CloudShell. Ini dapat disebabkan karena gambar individu yang besar atau terlalu banyak gambar Docker yang sudah ada sebelumnya.

Solusi: Jalankan `df -h` untuk menemukan penggunaan disk. Jalankan `sudo du -sh /folder/folder1` untuk menilai ukuran folder tertentu yang menurut Anda mungkin besar dan pertimbangkan untuk menghapus file lain untuk mengosongkan ruang. Salah satu opsi adalah mempertimbangkan untuk menghapus gambar Docker yang tidak digunakan dengan menjalankan `docker rmi`. Anda harus menyadari bahwa Docker memiliki ruang terbatas di lingkungan, untuk informasi lebih lanjut tentang Docker, lihat panduan Dokumentasi [Docker](#).

## **docker push** habis waktu dan terus mencoba lagi

Masalah: Ketika Anda `docker push` menjalankannya adalah waktu habis dan terus mencoba lagi tanpa hasil.

Penyebab: Ini dapat disebabkan sebagai akibat dari izin yang hilang, mendorong ke repositori yang salah atau kurangnya otentikasi.

Solusi: Untuk mencoba dan menyelesaikan masalah ini, pastikan Anda mendorong ke repositori yang benar. Jalankan `docker login` untuk mengautentikasi dengan benar. Pastikan Anda memiliki semua izin yang diperlukan untuk mendorong ke repositori Amazon ECR.

## Tidak dapat mengakses sumber daya dalam VPC dari saya AWS CloudShell Lingkungan VPC

Masalah: Tidak dapat mengakses sumber daya dalam VPC saat menggunakan lingkungan VPC saya AWS CloudShell .

Penyebab: Lingkungan AWS CloudShell VPC Anda mewarisi pengaturan jaringan VPC Anda.

Solusi: Untuk mengatasi masalah ini, pastikan VPC Anda diatur dengan benar untuk mengakses sumber daya Anda. [Untuk informasi selengkapnya, lihat dokumentasi VPC Menghubungkan VPC Anda ke jaringan lain dan dokumentasi Network Access Analyzer Network Access Analyzer.](#) Anda dapat menemukan alamat IPv4 yang digunakan lingkungan AWS CloudShell VPC, dengan menjalankan perintah `ip -a` di dalam lingkungan Anda di prompt baris perintah, atau di halaman Konsol VPC.

## ENI yang digunakan oleh AWS CloudShell karena lingkungan VPC saya tidak dibersihkan

Masalah: Tidak dapat membersihkan ENI yang digunakan oleh AWS CloudShell untuk lingkungan VPC saya.

Penyebab: `ec2:DeleteNetworkInterface` izin tidak diaktifkan untuk peran Anda.

Solusi: Untuk mengatasi masalah ini, pastikan `ec2:DeleteNetworkInterface` izin diaktifkan untuk peran Anda seperti yang ditunjukkan dalam skrip contoh berikut:

```
{
  "Effect": "Allow",
  "Action": [
    "ec2:DeleteNetworkInterface"
  ],
  "Condition": {
    "StringEquals": {
      "aws:ResourceTag/ManagedByCloudShell": ""
    }
  },
  "Resource": "arn:aws:ec2:*:*:network-interface/*"
}
```

## Pengguna dengan **CreateEnvironment** izin hanya untuk lingkungan VPC juga memiliki akses ke publik AWS CloudShell lingkungan

Masalah: Pengguna dibatasi dengan `CreateEnvironment` izin hanya untuk lingkungan VPC juga dapat mengakses lingkungan publik AWS CloudShell .

Penyebab: Ketika Anda membatasi `CreateEnvironment` izin untuk pembuatan lingkungan VPC saja dan jika Anda telah membuat lingkungan publik, Anda akan menjaga akses Anda ke lingkungan CloudShell publik yang ada sampai lingkungan ini dihapus menggunakan antarmuka pengguna web. Tetapi jika Anda belum pernah menggunakan CloudShell sebelumnya, Anda tidak akan memiliki akses ke lingkungan publik.

Solusi: Untuk membatasi akses ke AWS CloudShell lingkungan publik, administrator IAM harus terlebih dahulu memperbarui kebijakan IAM dengan pembatasan, dan kemudian pengguna harus secara manual menghapus lingkungan publik yang ada menggunakan antarmuka pengguna web. AWS CloudShell (Tindakan → Hapus CloudShell lingkungan).

## Didukung AWS Daerah untuk AWS CloudShell

Untuk informasi tentang AWS Wilayah dan titik akhir layanan yang AWS CloudShell mendukung, lihat sumber daya berikut:

- [AWS Daerah](#)
- [Titik akhir layanan](#)

# Kuota layanan dan batasan untuk AWS CloudShell

Halaman ini menjelaskan kuota Layanan dan pembatasan yang berlaku untuk area berikut:

- [Penyimpanan persisten](#)
- [Penggunaan bulanan](#)
- [Cangkang bersamaan](#)
- [Ukuran perintah](#)
- [Sesi Shell](#)
- [Lingkungan VPC](#)
- [Akses jaringan dan transfer data](#)
- [File sistem dan memuat ulang halaman](#)

## Penyimpanan tetap

Dengan AWS CloudShell, Anda memiliki penyimpanan persisten 1 GB untuk masing-masing tanpa Wilayah AWS biaya. Penyimpanan persisten terletak di direktori home Anda (\$HOME) dan bersifat pribadi untuk Anda. Tidak seperti sumber daya lingkungan sementara yang didaur ulang setelah setiap sesi shell berakhir, data di direktori home Anda tetap ada di antara sesi.

### Note

CloudShell Lingkungan VPC tidak memiliki penyimpanan persisten. Direktori \$HOME dihapus ketika waktu lingkungan VPC Anda habis (setelah 20-30 menit tidak aktif), atau ketika Anda menghapus lingkungan Anda.

Jika Anda berhenti menggunakan AWS CloudShell Wilayah AWS, data disimpan dalam penyimpanan persisten Wilayah tersebut selama 120 hari setelah akhir sesi terakhir Anda. Setelah 120 hari, kecuali Anda mengambil tindakan, data Anda secara otomatis dihapus dari penyimpanan persisten Wilayah tersebut. Anda dapat mencegah penghapusan dengan meluncurkan AWS CloudShell lagi di dalamnya Wilayah AWS. Untuk informasi selengkapnya, lihat [Langkah 2: Pilih Wilayah, luncurkan AWS CloudShell, dan pilih shell](#).

**Note****Skenario penggunaan**

Márcia telah digunakan AWS CloudShell untuk menyimpan file di direktori rumahnya di dua Wilayah AWS: AS Timur (Virginia N.) dan Eropa (Irlandia). Dia kemudian mulai menggunakan AWS CloudShell secara eksklusif di Eropa (Irlandia) dan berhenti meluncurkan sesi shell di AS Timur (Virginia N.).

Sebelum batas waktu untuk menghapus data di AS Timur (Virginia N.), Márcia memutuskan untuk mencegah direktori rumahnya didaur ulang dengan meluncurkan AWS CloudShell dan memilih Wilayah AS Timur (Virginia N.) lagi. Karena dia terus menggunakan Eropa (Irlandia) untuk sesi shell, penyimpanan persistennya di Wilayah itu tidak terpengaruh.

## Penggunaan bulanan

Masing-masing Wilayah AWS di Anda Akun AWS memiliki kuota pemakaian bulanan untuk AWS CloudShell. Kuota ini menggabungkan total waktu yang dihabiskan CloudShell oleh semua kepala sekolah IAM di Wilayah tersebut. Jika Anda mencoba mengakses CloudShell setelah mencapai kuota bulanan untuk Wilayah tersebut, sebuah pesan akan ditampilkan untuk menjelaskan mengapa lingkungan shell tidak dapat dimulai.

Untuk meminta peningkatan menggunakan konsol Service Quotas

Anda dapat meminta peningkatan kuota penggunaan bulanan Anda dengan membuka konsol [Service Quotas](#). Untuk informasi selengkapnya, lihat [Meminta peningkatan kuota](#) di Panduan Pengguna Service Quotas.

## Cangkang bersamaan

Anda dapat menjalankan hingga 10 shell pada saat yang sama di masing-masing Wilayah AWS untuk akun Anda.

Untuk meminta peningkatan menggunakan konsol Service Quotas

Anda dapat meminta peningkatan kuota untuk setiap Wilayah dengan membuka konsol [Service Quotas](#). Untuk informasi selengkapnya, lihat [Meminta peningkatan kuota](#) di Panduan Pengguna Service Quotas.

## Ukuran perintah

Ukuran perintah tidak boleh melebihi 65412 karakter.

### Note

Jika Anda bermaksud menjalankan perintah yang melebihi 65412 karakter, maka buat skrip dengan bahasa pilihan Anda, lalu jalankan dari antarmuka baris perintah. Untuk informasi selengkapnya tentang berbagai perangkat lunak pra-instal yang dapat diakses dari antarmuka baris perintah, lihat Perangkat lunak [pra-instal](#).

Untuk melihat sebagai contoh cara membuat skrip, dan kemudian menjalankannya dari antarmuka baris perintah, lihat [Tutorial: Memulai dengan AWS CloudShell](#).

## Sesi Shell

- Sesi tidak aktif: AWS CloudShell adalah lingkungan shell interaktif — jika Anda tidak berinteraksi dengannya menggunakan keyboard atau pointer selama 20-30 menit, sesi shell Anda berakhir. Proses yang berjalan tidak dihitung sebagai interaksi.

Jika Anda ingin melakukan tugas berbasis terminal menggunakan layanan AWS dengan batas waktu yang lebih fleksibel, sebaiknya luncurkan dan [sambungkan ke instans Amazon EC2](#).

- Sesi jangka panjang: Sesi shell yang berjalan terus menerus selama kurang lebih 12 jam secara otomatis berakhir meskipun pengguna secara teratur berinteraksi dengannya selama periode tersebut.

## Lingkungan VPC

Anda hanya dapat membuat hingga dua lingkungan VPC per prinsipal IAM.

### Note

Tidak ada biaya untuk terhubung ke VPC pribadi Anda dan mengakses sumber daya di dalamnya. Transfer data dalam VPC Pribadi Anda sudah termasuk dalam penagihan VPC Anda, dan transfer data antar pelanggan Anda CloudShell dibebankan dengan VPCs biaya yang sama dengan saat ini. CloudShell

## Akses jaringan dan transfer data

Pembatasan berikut berlaku untuk lalu lintas masuk dan keluar lingkungan Anda AWS CloudShell :

- Outbound: Anda dapat mengakses internet publik.
- Inbound: Anda tidak dapat mengakses port masuk. Tidak ada alamat IP publik yang tersedia.

### Warning

Dengan akses ke internet publik, ada risiko bahwa pengguna tertentu mungkin mengekspor data dari AWS CloudShell lingkungan. Kami menyarankan agar administrator IAM mengelola daftar izinkan AWS CloudShell pengguna tepercaya melalui alat IAM. Untuk informasi tentang bagaimana pengguna tertentu dapat secara eksplisit ditolak aksesnya, lihat. [Mengelola tindakan yang diijinkan di AWS CloudShell menggunakan kebijakan khusus](#)

Transfer data: Mengunggah dan mengunduh file ke dan dari AWS CloudShell mungkin lambat untuk file besar. Atau, Anda dapat mentransfer file ke lingkungan Anda dari bucket Amazon S3 menggunakan antarmuka baris perintah shell.

## Pembatasan pada file sistem dan pemuatan ulang halaman

- File sistem: Jika Anda salah memodifikasi file yang diperlukan oleh lingkungan komputasi, Anda mungkin mengalami masalah saat mengakses atau menggunakan lingkungan. AWS CloudShell Jika ini terjadi, Anda mungkin perlu [menghapus direktori home Anda](#) untuk mendapatkan kembali akses.
- Memuat ulang halaman: Untuk memuat ulang AWS CloudShell antarmuka, gunakan tombol refresh di browser Anda alih-alih urutan tombol pintasan default untuk sistem operasi Anda.

# Riwayat dokumen untuk Panduan AWS CloudShell Pengguna

Update terbaru

Tabel berikut menjelaskan perubahan penting pada Panduan AWS CloudShell Pengguna.

Perubahan	Deskripsi	Tanggal
<a href="#">Amazon Q CLI di AWS CloudShell</a>	Menambahkan dukungan untuk menggunakan fitur Amazon Q CLI di AWS CloudShell	Oktober 2, 2024
<a href="#">Dukungan Amazon VPC untuk AWS CloudShell di Wilayah tertentu</a>	Menambahkan dukungan untuk membuat dan menggunakan lingkungan AWS CloudShell VPC di Wilayah tertentu.	Juni 13, 2024
<a href="#">Tutorial baru telah ditambahkan ke Panduan AWS CloudShell Pengguna</a>	Dua tutorial baru telah ditambahkan yang merinci cara membangun wadah Docker di dalamnya AWS CloudShell dan mendorongnya ke repositori Amazon ECR, dan cara menerapkan fungsi Lambda melalui AWS CDK	27 Desember 2023
<a href="#">Kontainer Docker didukung AWS CloudShell di Wilayah tertentu</a>	Support untuk kontainer Docker dengan AWS CloudShell telah ditambahkan di Wilayah tertentu.	27 Desember 2023
<a href="#">AWS CloudShell telah bermigrasi untuk sekarang</a>	AWS CloudShell sekarang menggunakan AL2023 dan	Desember 4, 2023

[menggunakan Amazon Linux 2023 \(\) AL2023](#)

telah bermigrasi dari Amazon Linux 2.

[Wilayah AWS Baru untuk AWS CloudShell](#)

AWS CloudShell sekarang umumnya tersedia di AWS Wilayah berikut:

Juni 16, 2023

- AS Barat (California Utara)
- Afrika (Cape Town)
- Asia Pasifik (Hong Kong)
- Asia Pasifik (Osaka)
- Asia Pasifik (Seoul)
- Asia Pasifik (Jakarta)
- Asia Pasifik (Singapura)
- Eropa (Paris)
- Eropa (Stockholm)
- Europe (Milan)
- Timur Tengah (Bahrain)
- Middle East (UAE)

[Peluncuran AWS CloudShell di Console Toolbar](#)

Luncurkan CloudShell di Console Toolbar, di kiri bawah konsol dengan memilih CloudShell.

Maret 28, 2023

[AWS Wilayah Baru untuk AWS CloudShell](#)

AWS CloudShell sekarang tersedia di AWS Wilayah berikut:

6 Oktober 2022

- Kanada (Pusat)
- Eropa (London)
- Amerika Selatan (Sao Paulo)

---

<a href="#">AWS CloudShell didukung di AWS AS GovCloud</a>	AWS CloudShell sekarang didukung di Wilayah AWS GovCloud (AS).	Juni 29, 2022
<a href="#">Keamanan FAQs</a>	Tambahan FAQs berfokus pada masalah keamanan.	April 14, 2022
<a href="#">Soket Web</a>	Menambahkan bagian ke persyaratan jaringan CloudShell yang menjelaskan penggunaan WebSocket protokol.	Maret 21, 2022
<a href="#">Memecahkan masalah tombol panah di PowerShell</a>	Ikuti langkah-langkah untuk memperbaiki tombol panah yang salah menampilkan huruf saat ditekan.	Februari 7, 2022
<a href="#">Pelengkapan otomatis tombol tab</a>	Dokumentasi baru yang menjelaskan cara menggunakan bash-completion, yang memungkinkan pelengkapan otomatis perintah atau argumen yang diketik sebagian dengan menekan tombol Tab.	24 September 2021
<a href="#">Menentukan Wilayah AWS</a>	Dokumentasi tentang menentukan default Wilayah AWS untuk AWS CLI perintah.	11 Mei 2021
<a href="#">Memformat dalam versi PDF dan Kindle</a>	Ukuran gambar tetap dan teks dalam sel tabel.	10 Maret 2021

[Rilis ketersediaan umum \(GA\)](#)  
[AWS CloudShell di AWS](#)  
[Wilayah tertentu](#)

AWS CloudShell sekarang  
umumnya tersedia di AWS  
Wilayah berikut:

15 Desember 2020

- AS Timur (Ohio)
- AS Timur (Virginia Utara)
- AS Barat (Oregon)
- Asia Pasifik (Tokyo)
- Eropa (Irlandia)
- Asia Pasifik (Mumbai)
- Asia Pasifik (Sydney)
- Eropa (Frankfurt)

Terjemahan disediakan oleh mesin penerjemah. Jika konten terjemahan yang diberikan bertentangan dengan versi bahasa Inggris aslinya, utamakan versi bahasa Inggris.