



Buku pedoman migrasi untuk migrasi AWS besar

# AWS Panduan Preskriptif



# AWS Panduan Preskriptif: Buku pedoman migrasi untuk migrasi AWS besar

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 merendahkan 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

Pengantar .....	1
Panduan untuk migrasi besar .....	1
Tentang runbook, alat, dan templat .....	2
Tahap 1: Inisialisasi .....	4
Tugas 1: Memvalidasi pola migrasi dan metadata .....	4
Langkah 1: Validasi pola migrasi .....	5
Langkah 2: Validasi metadata migrasi dan rencana gelombang .....	6
Kriteria keluar tugas .....	8
Tugas 2: Membuat draf runbook migrasi .....	8
Langkah 1: Buat draf runbook migrasi untuk setiap pola .....	8
Langkah 2: Perbarui runbook migrasi dengan kebijakan dan proses .....	9
Kriteria keluar tugas .....	11
Tugas 3: Menganalisis dan menguji runbook migrasi Anda .....	11
Langkah 1: Lakukan penelusuran setiap runbook .....	12
Langkah 2: Lakukan POC yang menguji setiap pola migrasi .....	12
Langkah 3: Tinjau dan identifikasi kesenjangan dalam draf runbook migrasi saat ini .....	13
Kriteria keluar tugas .....	13
Tugas 4: Meningkatkan runbook migrasi Anda .....	13
Langkah 1: Perbarui runbook migrasi dan ulangi pengujian .....	14
Langkah 2: Otomatiskan tugas berulang .....	14
Langkah 3: Buat daftar tugas migrasi .....	15
Kriteria keluar tugas .....	15
Tahap 2: Implementasi .....	17
Tugas 1: Melakukan perencanaan sprint untuk gelombang terjadwal .....	18
Langkah 1: Tinjau backlog untuk gelombang yang dijadwalkan .....	18
Langkah 2: Tetapkan tugas dan tentukan tanggal jatuh tempo .....	18
Tugas 2: Melakukan tugas pra-migrasi dan migrasi .....	19
Tugas 3: Melakukan tugas cutover .....	19
Tugas 4: Meninjau dan meningkatkan runbook migrasi .....	20
Langkah 1: Tinjau gelombang yang telah selesai dan identifikasi celah di buku runbook migrasi saat ini .....	21
Langkah 2: Perbarui runbook migrasi dan selesaikan pengujian .....	21
Sumber daya .....	22
AWS migrasi besar .....	22

---

Referensi tambahan .....	22
Kontributor .....	23
Riwayat dokumen .....	24
Glosarium .....	25
# .....	25
A .....	26
B .....	29
C .....	31
D .....	34
E .....	38
F .....	40
G .....	42
H .....	43
I .....	45
L .....	47
M .....	49
O .....	53
P .....	56
Q .....	59
R .....	59
D .....	62
T .....	66
U .....	67
V .....	68
W .....	68
Z .....	69
.....	lxxi

# Buku pedoman migrasi untuk migrasi AWS besar

Amazon Web Services ([kontributor](#))

Februari 2022 ([riwayat dokumen](#))

Dalam migrasi besar, alur kerja migrasi menggunakan rencana gelombang dan metadata migrasi yang disediakan oleh alur kerja portofolio untuk memigrasikan beban kerja ke file. AWS Cloud Aliran kerja migrasi bertanggung jawab untuk mengirimkan permintaan perubahan, memigrasi aplikasi, mengoordinasikan pengujian aplikasi dengan pemilik aplikasi, melakukan cutover, dan memantau aplikasi melalui periode hypercare. Pada tahap pertama, menginisialisasi migrasi besar, Anda membuat buku runbook yang digunakan oleh aliran kerja migrasi untuk memigrasikan aplikasi dan server. Pada tahap kedua, menerapkan migrasi besar, alur kerja migrasi merencanakan sprint dan menggunakan runbook migrasi untuk memigrasi dan memotong aplikasi. Untuk informasi selengkapnya tentang alur kerja inti dan pendukung, lihat [Aliran kerja dalam migrasi besar di buku pedoman Foundation untuk migrasi besar](#). AWS

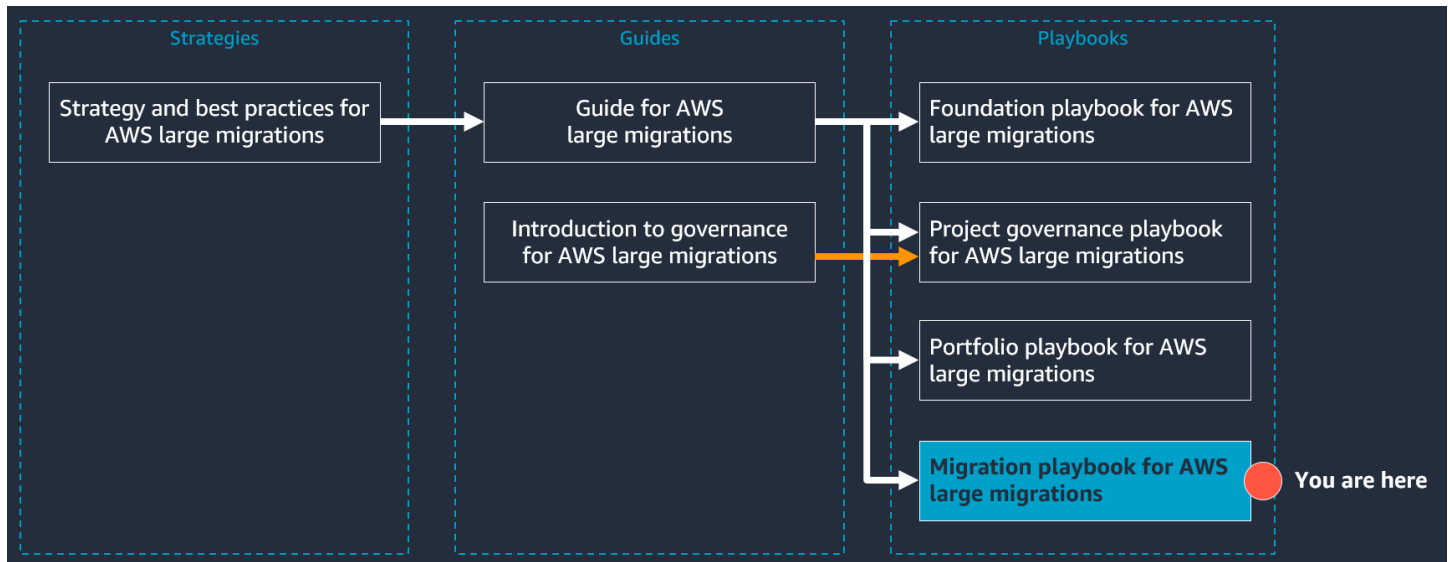
Buku pedoman migrasi ini menguraikan tugas alur kerja migrasi, yang mencakup kedua tahap migrasi, inisialisasi, dan implementasi besar:

- Pada tahap 1, inisialisasi, Anda menyusun, menguji, dan menyempurnakan runbook, lalu Anda mengotomatiskan tugas manual untuk setiap pola migrasi.
- Di tahap 2, implementasikan, Anda melakukan migrasi dengan runbook standar yang dibangun di tahap 1.

## Panduan untuk migrasi besar

Migrasi 300 atau lebih server dianggap sebagai migrasi besar. Tantangan orang, proses, dan teknologi dari proyek migrasi besar biasanya baru bagi sebagian besar perusahaan. Dokumen ini adalah bagian dari seri Panduan AWS Preskriptif tentang migrasi besar ke AWS Cloud. Seri ini dirancang untuk membantu Anda menerapkan strategi dan praktik terbaik yang benar sejak awal, untuk merampingkan perjalanan Anda ke cloud.

Gambar berikut menunjukkan dokumen lain dalam seri ini. Tinjau strategi terlebih dahulu, lalu panduannya, lalu lanjutkan ke buku pedoman. Untuk mengakses seri lengkap, lihat [Migrasi besar ke file. AWS Cloud](#)



## Tentang runbook, alat, dan templat

Sebaiknya gunakan [templat playbook migrasi](#) dan kemudian menyesuaikannya untuk portofolio, proses, dan lingkungan Anda. Template yang disediakan mencakup proses standar, proses cutover tipikal, dan placeholder untuk proses yang unik untuk lingkungan Anda. Petunjuk dalam buku pedoman ini memberi tahu Anda kapan dan bagaimana menyesuaikan masing-masing templat ini. Buku pedoman ini mencakup templat berikut:

- Templat runbook migrasi rehost
- Templat daftar tugas migrasi rehost

Untuk pola migrasi, dari mana Anda dapat membuat runbook Anda sendiri, lihat Pola migrasi [Panduan AWS Preskriptif](#).

Runbook migrasi memerlukan berbagai tingkat detail:

- Runbook terperinci — Runbook terperinci paling cocok untuk pola migrasi yang akan Anda ulangi berkali-kali. Untuk pola ini, sebaiknya mulai dengan template runbook migrasi Rehost (format Microsoft Word). Template ini menangkap detail sebanyak mungkin, termasuk tangkapan layar dan step-by-step instruksi, dan dirancang untuk membantu banyak orang melakukan tugas yang sama secara konsisten.
- Daftar Tugas - Untuk pola migrasi yang satu kali atau sangat sederhana, daftar tugas pendek adalah pilihan yang lebih baik. Untuk pola ini, sebaiknya mulai dengan templat daftar tugas migrasi Rehost (format Microsoft Excel). Template ini berisi daftar tugas tingkat tinggi dan biasanya

digunakan untuk melacak dan mengelola kepemilikan tugas. Anda juga dapat menggunakan daftar tugas untuk melacak status tugas yang didokumentasikan dalam runbook.

Apakah Anda menggunakan runbook terperinci atau daftar tugas pendek, verifikasi bahwa runbook Anda menjelaskan tugas secara berurutan. Untuk tugas kompleks, Anda dapat memberikan tautan ke dokumentasi eksternal.

# Tahap 1: Menginisialisasi migrasi besar

Pada tahap inisialisasi, tujuannya adalah untuk menentukan prosedur operasi standar (SOP) untuk migrasi besar, juga dikenal sebagai runbook. Anda membuat runbook khusus berdasarkan kebijakan dan proses perusahaan Anda. Jika anggota tim lain bertanggung jawab untuk menentukan runbook dalam proyek migrasi besar Anda, lewati ke [Stage 2: Menerapkan migrasi besar](#), di mana Anda akan menggunakan runbook untuk memprioritaskan aplikasi dan melakukan perencanaan gelombang.

Tahap 1 terdiri dari tugas dan langkah berikut:

- [Tugas 1: Memvalidasi pola migrasi dan metadata](#)
  - [Langkah 1: Validasi pola migrasi](#)
  - [Langkah 2: Validasi metadata migrasi dan rencana gelombang](#)
- [Tugas 2: Membuat draf runbook migrasi](#)
  - [Langkah 1: Buat draf runbook migrasi untuk setiap pola](#)
  - [Langkah 2: Perbarui runbook migrasi dengan kebijakan dan proses](#)
- [Tugas 3: Menganalisis dan menguji runbook migrasi Anda](#)
  - [Langkah 1: Lakukan penelusuran setiap runbook](#)
  - [Langkah 2: Lakukan POC yang menguji setiap pola migrasi](#)
  - [Langkah 3: Tinjau dan identifikasi kesenjangan dalam draf runbook migrasi saat ini](#)
- [Tugas 4: Meningkatkan runbook migrasi Anda](#)
  - [Langkah 1: Perbarui runbook migrasi dan ulangi pengujian](#)
  - [Langkah 2: Otomatiskan tugas berulang](#)
  - [Langkah 3: Buat daftar tugas migrasi](#)

Dengan runbook migrasi, pada tahap 2, tim migrasi mengikuti prosedur dan melakukan migrasi besar yang memiliki hasil yang dapat diprediksi dan terukur.

## Tugas 1: Memvalidasi pola migrasi dan metadata

Dalam tugas ini, Anda memvalidasi pola migrasi yang diidentifikasi dalam penilaian dan aktivitas perencanaan gelombang di alur kerja portofolio, lalu Anda memvalidasi sumber metadata migrasi. Tujuannya adalah untuk memverifikasi bahwa data yang cukup telah dikumpulkan untuk mendukung setiap pola migrasi.

Tugas ini terdiri dari langkah-langkah berikut:

- [Langkah 1: Validasi pola migrasi](#)
- [Langkah 2: Validasi metadata migrasi dan rencana gelombang](#)

## Langkah 1: Validasi pola migrasi

Dalam alur kerja portofolio, Anda melakukan penilaian awal terhadap portofolio aplikasi, strategi migrasi yang dipilih, dan pola migrasi yang diidentifikasi untuk setiap strategi. Informasi ini harus terkandung dalam runbook penilaian portofolio Anda. Untuk informasi selengkapnya, lihat [buku pedoman Portofolio untuk migrasi AWS besar](#).

Pada langkah ini, Anda meninjau strategi migrasi, memverifikasi bahwa Anda telah mengidentifikasi semua pola migrasi, dan mengonfirmasi bahwa Anda siap untuk menyusun runbook migrasi. Anda dapat mengulangi tugas ini di seluruh proyek, dan saat pemahaman Anda tentang portofolio matang, kemungkinan Anda akan mengidentifikasi pola migrasi tambahan di tahap migrasi selanjutnya.

### 1. Tinjau strategi migrasi untuk portofolio

Strategi migrasi adalah pendekatan yang digunakan untuk memigrasikan aplikasi lokal ke aplikasi. AWS Cloud Ada tujuh strategi migrasi untuk memindahkan aplikasi ke cloud, yang dikenal sebagai 7 Rs. Strategi umum untuk migrasi besar termasuk rehost, replatform, relokasi, dan pensiun. Refactor tidak disarankan untuk migrasi besar karena melibatkan modernisasi aplikasi selama migrasi. Ini adalah strategi migrasi yang paling kompleks, dan dapat menjadi rumit untuk mengelola sejumlah besar aplikasi. Sebagai gantinya, kami merekomendasikan rehosting, relokasi, atau replatforming aplikasi dan kemudian memodernisasi aplikasi setelah migrasi selesai. Untuk informasi lebih lanjut tentang 7 Rs, lihat [Panduan untuk migrasi AWS besar](#).

Berdasarkan hasil penilaian portofolio awal, Anda memiliki daftar semua strategi migrasi yang diperlukan untuk portofolio dan menentukan berapa banyak portofolio yang dialokasikan untuk setiap strategi. Contoh:

- Rehost — 70%
- Platform Ulang — 20%
- Pensiun — 10%

### 2. Verifikasi bahwa pola migrasi untuk portofolio

Pola migrasi adalah tugas migrasi berulang yang merinci strategi, tujuan, dan aplikasi atau layanan yang digunakan. Pada langkah ini, Anda memverifikasi bahwa pola migrasi menyertakan informasi terperinci, seperti alat mana yang akan digunakan dan AWS layanan mana yang ditargetkan.

Contoh:

- Rehost ke Amazon Elastic Compute Cloud (Amazon EC2)AWS menggunakan (MGN) atau Cloud AWS Transform MGN Migration Factory
- Replatform ke Amazon EC2 dengan AWS CloudFormation menggunakan template untuk membangun infrastruktur baru di AWS Cloud
- Replatform ke Amazon Relational Database Service (Amazon RDS) dengan AWS Database Migration Service menggunakan AWS (DMS) atau teknologi database asli

Di [buku pedoman Portofolio untuk migrasi AWS besar](#), Anda memetakan setiap pola migrasi ke strategi migrasi dan mendokumentasikan hasilnya dalam tabel seperti contoh berikut.

Strategi	Pola
Rehost	Rehost ke Amazon EC2 dengan menggunakan MGN atau Cloud Migration Factory
Platform Ulang	Replatform ke Amazon RDS dengan menggunakan AWS DMS atau teknologi database asli
Platform Ulang	Replatform ke Amazon EC2 dengan CloudFormation menggunakan template untuk membangun infrastruktur baru di AWS Cloud

## Langkah 2: Validasi metadata migrasi dan rencana gelombang

Pada langkah ini, Anda memvalidasi lokasi sumber metadata migrasi. Anda memeriksa apakah struktur data, seperti kolom yang tersedia dalam dokumen Excel, cocok untuk menyimpan metadata yang diperlukan, dan Anda memeriksa apakah semua metadata tersedia.

### 1. Validasi metadata migrasi untuk pola migrasi Anda

Setiap pola migrasi memerlukan satu set metadata migrasi yang berbeda untuk memigrasikan server dan aplikasi. Misalnya, migrasi rehost ke Amazon EC2 mengharuskan Anda memberikan spesifikasi untuk instans target, seperti subnet VPC, grup keamanan, dan informasi jenis instans. Namun, migrasi penyimpanan, migrasi database, atau migrasi replatform memerlukan kumpulan metadata migrasi yang berbeda. Anda biasanya menentukan persyaratan metadata migrasi dalam runbook penilaian portofolio, tetapi Anda perlu memastikan bahwa Anda memiliki metadata yang cukup untuk mendukung setiap pola migrasi Anda. Untuk informasi selengkapnya tentang identifikasi dan pengumpulan metadata, lihat [buku pedoman Portofolio untuk AWS migrasi](#) besar.

## 2. Validasi lokasi sumber metadata migrasi dan rencana gelombang

Anda biasanya mendokumentasikan lokasi sumber metadata migrasi di buku runbook manajemen metadata Anda. Idealnya, lokasi bertindak sebagai sumber kebenaran tunggal, seperti spreadsheet perencanaan gelombang. Mungkin juga metadata masih di beberapa tempat, termasuk lokasi umum berikut:

Validasi berikut ini untuk lokasi sumber metadata:

- Alat penemuan
- Database manajemen konfigurasi (CMDB)
- Kuesioner pemilik aplikasi
- Spreadsheet perencanaan gelombang migrasi

Validasi berikut ini untuk lokasi sumber metadata:

- a. Apakah katalog sumber dikelola dengan lokasi semua sumber dan pemilik metadata?
- b. Apakah lokasi sumber (misalnya, spreadsheet perencanaan gelombang) memiliki semua metadata migrasi yang diperlukan?
- c. Apakah ada instruksi yang jelas untuk mengakses setiap sumber metadata?
- d. Jika tidak ada sumber tunggal, apakah setiap sumber metadata dipetakan dengan jelas ke atributnya?
- e. Apakah ada rencana gelombang yang jelas untuk server dan aplikasi, dan apakah setidaknya lima gelombang siap untuk aliran kerja migrasi?
- f. Apakah ada proses untuk memperbarui sumber? Jika demikian, bagaimana frekuensi dan proses notifikasi?

## Kriteria keluar tugas

Ketika Anda telah memenuhi kriteria keluar berikut, lanjutkan ke tugas berikutnya:

- Anda telah memvalidasi daftar pola migrasi yang didefinisikan dengan jelas.
- Lokasi sumber metadata migrasi memiliki semua metadata yang diperlukan untuk setiap pola, atau ada proses untuk menangkap metadata yang hilang.
- Anda telah memvalidasi rencana gelombang dan metadata migrasi untuk setidaknya lima gelombang, dan Anda telah menentukan proses untuk pemberitahuan dan pembaruan.

## Tugas 2: Membuat draf runbook migrasi

Dalam tugas ini, Anda menyusun dan meninjau buku runbook migrasi untuk setiap pola migrasi. Misalnya, Anda menyusun runbook migrasi untuk rehost ke Amazon EC2 dan runbook lain untuk replatform ke Amazon RDS. Anda mengulangi tugas ini sampai Anda telah menyusun buku runbook migrasi untuk setiap pola migrasi yang diidentifikasi dalam tugas sebelumnya.

Anda dapat menggunakan templat runbook yang disediakan yang tersedia di [templat buku pedoman migrasi](#) dan menyesuaikannya untuk lingkungan Anda. Untuk pola migrasi yang sering diulang, sebaiknya gunakan templat runbook migrasi Rehost (format Microsoft Word), dan untuk pola yang satu kali atau sangat sederhana, kami merekomendasikan templat daftar tugas migrasi Rehost (format Microsoft Excel). Anda juga dapat menggunakan daftar tugas untuk melacak status tugas yang didokumentasikan dalam runbook. Untuk informasi selengkapnya, lihat [Tentang runbook, alat, dan templat](#).

Tugas ini terdiri dari langkah-langkah berikut:

- [Langkah 1: Buat draf runbook migrasi untuk setiap pola](#)
- [Langkah 2: Perbarui runbook migrasi dengan kebijakan dan proses](#)

### Langkah 1: Buat draf runbook migrasi untuk setiap pola

Pada langkah ini, Anda menyusun runbook untuk setiap pola migrasi Anda. Runbook migrasi lengkap biasanya berisi petunjuk tentang cara menggunakan layanan atau alat migrasi yang dipilih, tugas apa pun yang unik untuk lingkungan Anda, dan instruksi pemotongan.

1. Buka templat runbook migrasi Rehost (format Microsoft Word), tersedia di templat buku [pedoman migrasi](#).
2. Perbarui bagian tugas Premigrasi, bagian Tugas migrasi, dan bagian Tugas potong dengan instruksi yang khusus untuk pola migrasi Anda. Bergantung pada kasus penggunaan Anda, Anda mungkin perlu memperbarui ketiga bagian tersebut. Sertakan hal berikut saat menyesuaikan tugas Anda:
  - Petunjuk migrasi standar untuk layanan yang dipilih — Anda biasanya dapat menemukan informasi yang diperlukan untuk melengkapi template Anda dalam AWS dokumentasi. Misalnya, lihat yang berikut ini:
    - [Cara menggunakan yang baru AWS Transform MGN untuk migrasi lift-and-shift](#)
    - [Memulai dengan AWS DataSync](#)
    - [AWS Database Migration Service penelusuran langkah demi langkah](#)
  - Tugas yang unik untuk lingkungan TI Anda - Rekam tugas yang unik untuk operasi dan lingkungan TI Anda. Tujuannya adalah agar orang baru yang bergabung dengan tim migrasi Anda dapat mengikuti runbook dengan kurva pembelajaran minimal. Misalnya, perangkat lunak pemantauan apa yang perlu Anda instal pada mesin target setelah cutover? Server Domain Name System (DNS) mana yang Anda gunakan untuk subnet itu? Bagaimana Anda mengirimkan permintaan untuk perubahan (RFC)?
  - Tugas cutover — Setiap lingkungan memiliki proses cutover yang sedikit berbeda. Penting untuk mendokumentasikan semua langkah untuk cutover di lingkungan Anda karena Anda ingin semua orang mengikuti proses yang sama. Mendokumentasikan langkah-langkah ini meminimalkan waktu yang dihabiskan di jendela cutover dan membantu Anda merencanakan jumlah waktu yang dibutuhkan untuk menyelesaikan cutover.

## Langkah 2: Perbarui runbook migrasi dengan kebijakan dan proses

Templat Runbook dan daftar tugas mencakup sebagian besar tugas migrasi, atau bagian dari proses yang standar. Tugas yang tersisa unik untuk lingkungan Anda, dan Anda harus menyesuaikan runbook yang sesuai. Misalnya, pertimbangkan apakah runbook Anda harus berisi tugas khusus untuk proses berikut di lingkungan Anda.

### Konektivitas

- Cara terhubung ke lingkungan VMware

- Cara terhubung ke server DNS dan memperbarui catatan DNS
- Cara terhubung ke server otomatisasi migrasi
- Cara terhubung ke lingkungan sumber
- Cara menghubungkan ke repositori dokumen, seperti SharePoint atau Confluence

#### Izin dan manajemen perubahan

- Cara mengirimkan RFC di lingkungan Anda
- Cara meninjau status RFC untuk setiap gelombang
- Cara memberikan akses untuk insinyur migrasi baru
- Cara meminta izin ke server sumber
- Cara meminta izin ke akun target AWS
- Siapa yang memiliki izin untuk terhubung ke server target setelah cutover

#### Implementasi migrasi dan cutover

- Perangkat lunak mana yang akan diinstal atau dihapus di server target
- Cara mengubah pengaturan infrastruktur, seperti firewall, routing, dan pengaturan load balancer
- Siapa yang dapat mengubah pengaturan infrastruktur
- Cara mengubah konfigurasi aplikasi selama cutover

- Cara melakukan pengujian aplikasi
- Cara menyelesaikan cutover dan go-live
- Cara menyelesaikan tugas yang terjadi setelah cutover, seperti mengonfigurasi pemantauan atau cadangan

Beberapa tugas ini mungkin terdengar sepele, tetapi pengetahuan dan izin bervariasi di lingkungan apa pun. Penting untuk mendokumentasikan tugas-tugas ini di runbook migrasi yang sama.

#### Tip

Kami sangat menyarankan menggunakan otomatisasi untuk mempercepat migrasi besar Anda. Menggunakan model pabrik migrasi menyederhanakan dan mengurangi jumlah masalah dengan tugas berulang, terutama untuk rehost dan replatform pola migrasi. [AWS Cloud Migration Factory Solution](#) dirancang untuk membantu pelanggan bermigrasi dalam skala besar dengan otomatisasi. Anda dapat menerapkan solusi dan menggunakan skrip otomatisasi yang telah ditentukan di runbook Anda.

## Kriteria keluar tugas

Ulangi tugas ini seperlunya, dan ketika Anda telah memenuhi kriteria keluar berikut, lanjutkan ke tugas berikutnya:

- Anda telah menyusun buku runbook untuk setiap pola migrasi.
- Setiap draf runbook berisi tiga bagian utama: tugas pra-migrasi, tugas migrasi, dan tugas cutover.
- Draft runbook Anda mencakup tugas-tugas yang unik untuk lingkungan Anda.
- Draft runbook terperinci Anda mencakup panduan langkah demi langkah dan tangkapan layar.

## Tugas 3: Menganalisis dan menguji runbook migrasi Anda

Dalam tugas ini, Anda menelusuri setiap runbook yang Anda buat di tugas sebelumnya, menganalisis kesenjangan yang diidentifikasi, melakukan bukti migrasi konsep (POC), dan meninjau catatan dan umpan balik.

Tugas ini terdiri dari langkah-langkah berikut:

- [Langkah 1: Lakukan penelusuran setiap runbook](#)
- [Langkah 2: Lakukan POC yang menguji setiap pola migrasi](#)
- [Langkah 3: Tinjau dan identifikasi kesenjangan dalam draf runbook migrasi saat ini](#)

## Langkah 1: Lakukan penelusuran setiap runbook

Pada langkah ini, tim migrasi menilai runbook dan urutan tugas seolah-olah mereka melakukannya secara nyata. Tim migrasi bertemu dan meninjau setiap langkah, dan anggota tim mengajukan pertanyaan dan membagikan umpan balik mereka. Proses penelusuran ini membantu tim mengidentifikasi langkah-langkah yang hilang dan masalah urutan. Lengkapi penelusuran sebagai berikut:

1. Kumpulkan tim migrasi yang bertanggung jawab untuk menyelesaikan tugas di runbook.
2. Berjalanlah melalui langkah-langkah di runbook satu per satu, seolah-olah ini adalah migrasi langsung. Saat Anda pergi, identifikasi dan catat kesenjangan atau masalah apa pun. Jangan melakukan migrasi atau tugas sebagai bagian dari panduan.
3. Perbarui draf runbook untuk mengatasi kesenjangan atau masalah yang diidentifikasi dalam panduan.

## Langkah 2: Lakukan POC yang menguji setiap pola migrasi

1. Pilih kandidat POC dari gelombang yang sudah disiapkan.
2. Buka draf runbook migrasi.
3. Selesaikan runbook langkah demi langkah untuk memigrasikan kandidat POC sebagai berikut:
  - Ikuti setiap langkah di runbook. Jangan membuat asumsi atau membuat keputusan sendiri.
  - Asumsikan orang yang menggunakan runbook tidak memiliki pengetahuan sebelumnya tentang migrasi atau lingkungan Anda.
  - Jika sebuah langkah tidak jelas tetapi Anda dapat melanjutkan, catat langkahnya dan lanjutkan.
  - Jika ada langkah yang hilang dan Anda tidak dapat melanjutkan, berhenti, dan sorot bagian dari mana Anda tidak dapat melanjutkan. Bekerja dengan pemilik runbook untuk mengklarifikasi langkah yang hilang sehingga Anda dapat melanjutkan dan menyelesaikan POC.

## Langkah 3: Tinjau dan identifikasi kesenjangan dalam draf runbook migrasi saat ini

1. Tinjau masalah atau kesenjangan yang diidentifikasi pada langkah sebelumnya.
2. Analisis kesenjangan dan pertimbangkan pertanyaan-pertanyaan berikut:
  - Apakah runbook memiliki langkah-langkah yang diperlukan untuk menyelesaikan migrasi dan pemotongan, dari ujung ke ujung?
  - Apakah runbook berisi tautan referensi untuk tugas-tugas yang telah ditentukan sebelumnya di lingkungan Anda?
  - Apakah runbook dengan jelas mendefinisikan siapa, apa, kapan, dan bagaimana menyelesaikan tugas?

### Kriteria keluar tugas

Ketika Anda telah memenuhi kriteria keluar berikut, lanjutkan ke tugas berikutnya:

- Anda telah meninjau dan menguji setiap runbook migrasi.
- Untuk setiap runbook, Anda telah menyelesaikan POC migrasi untuk setidaknya satu aplikasi dan untuk lebih dari dua varian sistem operasi (OS).
- Anda telah mengidentifikasi dan mendokumentasikan kesenjangan dan masalah yang diidentifikasi di setiap runbook.

## Tugas 4: Meningkatkan runbook migrasi Anda

Dalam tugas ini, Anda meningkatkan runbook dengan mengulangi POC beberapa kali. Dengan setiap gelombang, tes POC dan retrospektif, pertemuan di mana tim meninjau gelombang yang telah selesai, memberikan kesempatan untuk meningkatkan runbook. Anda juga meningkatkan runbook Anda dengan mengotomatiskan tugas berulang, yang meningkatkan kecepatan migrasi dan mengurangi risiko kesalahan konfigurasi manual.

Tugas ini terdiri dari langkah-langkah berikut:

- [Langkah 1: Perbarui runbook migrasi dan ulangi pengujian](#)

- [Langkah 2: Otomatiskan tugas berulang](#)
- [Langkah 3: Buat daftar tugas migrasi](#)

## Langkah 1: Perbarui runbook migrasi dan ulangi pengujian

1. Untuk masalah dan kesenjangan yang diidentifikasi dalam tugas sebelumnya, perbarui runbook dengan instruksi terperinci. Contoh:
  - Jika ada langkah yang hilang, tambahkan petunjuk langkah demi langkah
  - Jika langkah tidak jelas, pertimbangkan untuk memperbarui teks, menambahkan tangkapan layar, atau menambahkan tautan referensi
2. Ulangi tugas sebelumnya sampai Anda puas bahwa instruksinya lengkap dan jelas.
3. Uji draf akhir setiap runbook dengan meminta anggota tim migrasi baru, orang yang belum menguji runbook ini sebelumnya, untuk melakukan POC dan menyelesaikan runbook.

## Langkah 2: Otomatiskan tugas berulang

1. Tinjau setiap runbook dan identifikasi area otomatisasi untuk tugas manual. Pertimbangkan pertanyaan-pertanyaan menyelidik berikut:
  - Apakah ada tugas manual yang berulang untuk setiap server atau aplikasi di runbook?
  - Apakah ada tindakan yang Anda lakukan di setiap server atau aplikasi?
  - Apakah Anda perlu menginstal atau menghapus perangkat lunak pada server target?
  - Apakah Anda perlu mengubah pengaturan jaringan atau infrastruktur satu per satu untuk setiap server?
  - Apakah Anda perlu menyalin dan menempelkan data secara manual?
2. Buat skrip otomatisasi dan perbarui runbook.
3. Ulangi tugas 3 dan tugas 4 hingga Anda mendokumentasikan runbook dengan informasi yang jelas dan lengkap serta tugas migrasi berulang otomatis.

### Note

Untuk mengotomatiskan tugas migrasi, kami sangat menyarankan Anda membuat skrip baru atau menyesuaikan skrip yang ada di [AWS Cloud Migration Factory Solution](#).

## Langkah 3: Buat daftar tugas migrasi

Daftar tugas migrasi dapat membantu Anda mengelola status dan pemilik tugas. Anda membuat daftar tugas untuk setiap runbook migrasi, dan Anda menyertakan informasi tingkat tinggi dari runbook tanpa menyertakan detailnya. Daftar tugas biasanya berisi informasi berikut, dan Anda dapat menambahkan lebih banyak atribut sesuai kebutuhan:

- Nama deskriptif, seperti:
  - Periksa versi OS server
  - Instal agen
  - Mulai ulang server
  - Perbarui DNS
- Dependensi
- Urutan tugas
- Pemilik
- Estimasi waktu yang dibutuhkan untuk menyelesaikan setiap tugas
- Status

Ada banyak alat yang tersedia untuk membuat dan mengelola daftar tugas. Anda dapat menggunakan templat daftar tugas migrasi Rehost yang disediakan (format Microsoft Excel) yang tersedia di templat [buku pedoman migrasi](#). Anda juga dapat menggunakan alat manajemen proyek, seperti Jira atau papan Kanban.

### Note

Kami juga merekomendasikan menggunakan templat daftar tugas Excel untuk mendokumentasikan tugas kecil, dipahami dengan baik, atau tidak berulang, seperti memulai ulang server atau mendapatkan alamat IP. Tugas-tugas ini harus ditangkap dan dilacak tetapi tidak memerlukan langkah-langkah rinci dari template runbook Word.

## Kriteria keluar tugas

Ulangi tugas ini seperlunya, dan ketika Anda telah memenuhi kriteria keluar berikut, lanjutkan ke tugas berikutnya:

- Anda telah mengidentifikasi peluang untuk otomatisasi dan telah mengembangkan skrip otomatisasi atau memiliki rencana untuk melakukannya.
- Tiga orang atau lebih memiliki peer-review setiap runbook.
- Dua orang atau lebih yang tidak berada di tim pengembangan untuk runbook telah mengujinya secara end-to-end.
- Menggunakan runbook terbaru, Anda telah memigrasikan 20 server atau lebih ke lebih dari satu AWS akun.
- Anda telah mengembangkan daftar tugas untuk membantu melacak dan mengelola kemajuan migrasi.

## Stage2: Menerapkan migrasi besar

Di tahap 1, Anda mengembangkan runbook migrasi untuk setiap pola migrasi. Pada tahap 2, Anda menggunakan runbook ini untuk memigrasikan server dan kemudian meningkatkan runbook untuk mempercepat kecepatan migrasi. Membangun dan memperbarui runbook bukanlah tugas satu kali. Anda mungkin perlu melakukan itu selama perjalanan migrasi besar Anda. Misalnya, Anda mungkin perlu membuat runbook baru jika cakupan meningkat dan Anda mengidentifikasi pola migrasi baru, atau Anda mungkin perlu meningkatkan runbook yang ada jika kecepatan migrasi di bawah target dan memperkenalkan lebih banyak otomatisasi akan mengurangi jumlah tugas manual dan mempercepat migrasi.

### Note

Rencana gelombang yang dikembangkan dalam alur kerja portofolio menentukan aktivitas dalam alur kerja migrasi. Sebelum memulai tahap 2, verifikasi bahwa Anda telah memvalidasi rencana gelombang Anda. Untuk petunjuk dan informasi selengkapnya tentang rencana gelombang, lihat [Buku pedoman portofolio untuk migrasi AWS besar](#).

Tahap 2 terdiri dari tugas dan langkah berikut:

- [Tugas 1: Melakukan perencanaan sprint untuk gelombang terjadwal](#)
  - [Langkah 1: Tinjau backlog untuk gelombang yang dijadwalkan](#)
  - [Langkah 2: Tetapkan tugas dan tentukan tanggal jatuh tempo](#)
- [Tugas 2: Melakukan tugas pra-migrasi dan migrasi](#)
- [Tugas 3: Melakukan tugas cutover](#)
- [Tugas 4: Meninjau dan meningkatkan runbook migrasi](#)
  - [Langkah 1: Tinjau gelombang yang telah selesai dan identifikasi celah di buku runbook migrasi saat ini](#)
  - [Langkah 2: Perbarui runbook migrasi dan selesaikan pengujian](#)

# Tugas 1: Melakukan perencanaan sprint untuk gelombang terjadwal

Dalam tugas ini, Anda menetapkan gelombang ke sprint, yang merupakan periode waktu tetap di mana tim migrasi bekerja pada semua gelombang dalam sprint itu. Jika setiap sprint berdurasi 2 minggu, setiap gelombang mencakup setidaknya dua sprint. Perencanaan sprint mengacu pada proses penugasan pemilik dan tanggal jatuh tempo untuk semua tugas dalam sprint itu.

Tugas ini terdiri dari langkah-langkah berikut:

- [Langkah 1: Tinjau backlog untuk gelombang yang dijadwalkan](#)
- [Langkah 2: Tetapkan tugas dan tentukan tanggal jatuh tempo](#)

## Langkah 1: Tinjau backlog untuk gelombang yang dijadwalkan

Pada langkah ini, Anda meninjau backlog yang ada, atau tugas saat ini dan yang tertunda, untuk semua gelombang bersamaan, dan Anda menggunakan alat dan mekanisme yang direkomendasikan untuk mengelola gelombang. Misalnya, Anda mungkin menggunakan papan Kanban dengan jalur renang untuk setiap gelombang, atau Anda dapat menggunakan Jira dan melacak ombak dengan cerita dan epos. Untuk informasi selengkapnya, lihat [buku pedoman tata kelola proyek untuk migrasi AWS besar](#).

## Langkah 2: Tetapkan tugas dan tentukan tanggal jatuh tempo

Pada langkah ini, untuk semua gelombang dalam sprint ini, Anda menetapkan pemilik untuk setiap tugas dan menetapkan tanggal jatuh tempo yang sesuai. Anda dapat menggunakan spreadsheet daftar tugas migrasi yang Anda buat di tahap 1 untuk mengelola kemajuan gelombang, kepemilikan tugas, dan tanggal jatuh tempo, dan tugas ditentukan secara rinci di buku runbook migrasi untuk setiap pola. Karena gelombang biasanya tumpang tindih, adalah umum untuk mengelola banyak tugas bersamaan dari gelombang yang berbeda pada saat yang sama. Selain itu, setiap gelombang dapat berkisar dari 3-6 minggu, tergantung pada proses internal Anda. Untuk contoh jadwal gelombang, lihat bagian [Tahap 2: Menerapkan migrasi besar](#) dari Panduan untuk migrasi AWS besar.

### Important

Jangan menambahkan tugas ke sprint tanpa memperbarui runbook atau daftar tugas.

Dokumen-dokumen yang Anda buat di tahap 1 ini harus menjadi sumber kebenaran untuk

semua aktivitas migrasi Anda. Jika ada langkah yang hilang atau salah, perbarui dan validasi runbook sebelum menambahkan tugas ke sprint.

## Tugas 2: Melakukan tugas pra-migrasi dan migrasi

Sekarang Anda melakukan tugas pra-migrasi dan migrasi dan mematuhi jadwal berdasarkan hasil perencanaan sprint Anda. Sprint backlog berisi daftar semua tugas dalam migrasi, untuk semua gelombang dalam sprint saat ini, dan mengatur tugas berdasarkan minggu. Untuk daftar tugas, lihat buku runbook migrasi untuk setiap pola migrasi, yang dibuat di tahap 1 buku pedoman ini. Untuk jadwal gelombang, lihat alat manajemen proyek Anda, yang ditetapkan dalam [buku pedoman tata kelola proyek untuk migrasi AWS besar](#). Lakukan tugas dalam minggu-minggu yang dijadwalkan. Berikut ini adalah contoh jadwal tugas migrasi rehost di mana ada tugas migrasi untuk gelombang yang berbeda di minggu yang sama.

Nama tugas	Gelombang	Kategori	Pemilik
Memverifikasi prasyarat	Gelombang 1	Build	Jane Doe
Instal agen replikasi	Gelombang 1	Build	Jane Doe
Validasi template peluncuran	Gelombang 2	Validasi	Jane Doe
Luncurkan contoh pengujian	Gelombang 3	Pengujian boot-up	Jane Doe

## Tugas 3: Melakukan tugas cutover

Pada titik ini, Anda telah menyelesaikan tugas migrasi dan menguji semua server dan aplikasi, dan Anda siap untuk cutover. Gunakan matriks RACI yang Anda buat di [buku pedoman Foundation untuk migrasi AWS besar](#) guna mengelola tugas dan kepemilikan setiap tugas cutover, dan gunakan runbook migrasi Anda untuk setiap pola guna melakukan aktivitas cutover. Tabel berikut adalah contoh bagaimana Anda dapat melacak dan mengelola kemajuan cutover. Adalah umum untuk memiliki beberapa pola migrasi dalam gelombang yang sama untuk aplikasi yang berbeda.

Nama tugas	Gelombang	Runbook migrasi	Pemilik	Status
Periksa replikasi	Gelombang 1	Rehost ke Amazon EC2	Jane Doe	Selesai
Luncurkan instance EC2 cutover	Gelombang 1	Rehost ke Amazon EC2	Jane Doe	Selesai
Validasi status instans EC2	Gelombang 1	Rehost ke Amazon EC2	Jane Doe	Sedang berlangsung
Luncurkan database di Amazon RDS	Gelombang 1	Replatform ke Amazon RDS	John Smith	Sedang berlangsung
Transfer data penyimpanan lengkap	Gelombang 1	Replatform ke Amazon Elastic File System (Amazon EFS)	John Smith	Tidak dimulai
Lakukan pengujian aplikasi	Gelombang 1	Semua	Jane Doe	Tidak dimulai
Keputusan penerimaan aplikasi	Gelombang 1	Semua	Jane Doe	Tidak dimulai

## Tugas 4: Meninjau dan meningkatkan runbook migrasi

Tugas ini terdiri dari langkah-langkah berikut:

- [Langkah 1: Tinjau gelombang yang telah selesai dan identifikasi celah di buku runbook migrasi saat ini](#)
- [Langkah 2: Perbarui runbook migrasi dan selesaikan pengujian](#)

## Langkah 1: Tinjau gelombang yang telah selesai dan identifikasi celah di buku runbook migrasi saat ini

Gagal cepat adalah filosofi yang menggunakan pengujian yang sering dan bertahap untuk mengurangi siklus hidup pengembangan, dan ini adalah bagian penting dari pendekatan tangkas untuk migrasi besar. Setelah setiap cutover, jadwalkan pertemuan retrospektif untuk meninjau setiap tugas dengan tim migrasi. Ajukan contoh pertanyaan menyelidik berikut. Anda juga dapat menambahkan pertanyaan Anda sendiri:

- Apakah cutover berhasil? Jika tidak, apa masalahnya?
- Apakah runbook migrasi mencakup semua tugas untuk melakukan migrasi dan cutover?
- Apakah ada tugas yang memakan waktu lebih lama dari yang diharapkan?
- Apakah Anda mengetahui adanya masalah teknis dengan tugas apa pun di runbook?
- Apakah ada tugas manual yang dapat diotomatisasi?
- Apakah ada masalah terkait proses dengan runbook atau cutover?

## Langkah 2: Perbarui runbook migrasi dan selesaikan pengujian

Setelah mengumpulkan data dari rapat retrospektif, perbarui runbook migrasi sebagai berikut:

- Tambahkan instruksi terperinci untuk setiap langkah yang hilang.
- Perbaiki atau perbarui langkah apa pun sesuai kebutuhan.
- Lakukan tes end-to-end migrasi dengan setidaknya satu Windows dan satu server Linux.
- Kirim buku runbook yang diperbarui ke tim migrasi untuk digunakan dalam gelombang berikutnya.

# Sumber daya

## AWS migrasi besar

Untuk mengakses seri Panduan AWS Preskriptif lengkap untuk migrasi besar, lihat Migrasi [besar ke](#).  
AWS Cloud

## Referensi tambahan

- [AWS Solusi Pabrik Migrasi Cloud](#)
- [AWS Pola migrasi Panduan Preskriptif](#)

# Kontributor

Individu-individu berikut berkontribusi pada dokumen ini:

- Chris Baker, Konsultan Migrasi Senior, Amazon Web Services
- Wally Lu, Konsultan Utama, Amazon Web Services

## Riwayat dokumen

Tabel berikut menjelaskan perubahan signifikan pada panduan ini. Jika Anda ingin diberi tahu tentang pembaruan masa depan, Anda dapat berlangganan umpan [RSS](#).

Perubahan	Deskripsi	Tanggal
<a href="#">Nama AWS solusi yang diperbarui</a>	Kami memperbarui nama AWS solusi yang direferensikan dari Pabrik CloudEndure Migrasi ke Pabrik Migrasi Cloud.	2 Mei 2022
<a href="#">Publikasi awal</a>	—	28 Februari 2022

# AWS Glosarium Panduan Preskriptif

Berikut ini adalah istilah yang umum digunakan dalam strategi, panduan, dan pola yang disediakan oleh Panduan AWS Preskriptif. Untuk menyarankan entri, silakan gunakan tautan Berikan umpan balik di akhir glosarium.

## Nomor

### 7 Rs

Tujuh strategi migrasi umum untuk memindahkan aplikasi ke cloud. Strategi ini dibangun di atas 5 Rs yang diidentifikasi Gartner pada tahun 2011 dan terdiri dari yang berikut:

- Refactor/re-architect — Pindahkan aplikasi dan modifikasi arsitekturnya dengan memanfaatkan sepenuhnya fitur cloud-native untuk meningkatkan kelincahan, kinerja, dan skalabilitas. Ini biasanya melibatkan porting sistem operasi dan database. Contoh: Migrasikan database Oracle lokal Anda ke Amazon Aurora Edition. PostgreSQL-Compatible
- Replatform (angkat dan bentuk ulang) — Pindahkan aplikasi ke cloud, dan perkenalkan beberapa tingkat pengoptimalan untuk memanfaatkan kemampuan cloud. Contoh: Memigrasikan database Oracle lokal Anda ke Amazon Relational Database Service (Amazon RDS) untuk Oracle di AWS Cloud
- Pembelian kembali (drop and shop) - Beralih ke produk yang berbeda, biasanya dengan beralih dari lisensi tradisional ke model SaaS. Contoh: Migrasikan sistem manajemen hubungan pelanggan (CRM) Anda ke Salesforce.com
- Rehost (lift dan shift) — Pindahkan aplikasi ke cloud tanpa membuat perubahan apa pun untuk memanfaatkan kemampuan cloud. Contoh: Migrasikan database Oracle lokal Anda ke Oracle pada instans EC2 di AWS Cloud
- Relokasi (hypervisor-level lift and shift) — Pindahkan infrastruktur ke cloud tanpa membeli perangkat keras baru, menulis ulang aplikasi, atau memodifikasi operasi yang ada. Anda memigrasikan server dari platform lokal ke layanan cloud untuk platform yang sama. Contoh: Migrasikan Microsoft Hyper-V aplikasi ke AWS.
- Pertahankan (kunjungi kembali) - Simpan aplikasi di lingkungan sumber Anda. Ini mungkin termasuk aplikasi yang memerlukan refactoring besar, dan Anda ingin menunda pekerjaan itu sampai nanti, dan aplikasi lama yang ingin Anda pertahankan, karena tidak ada pembenaran bisnis untuk memigrasikannya.

- Pensiun — Menonaktifkan atau menghapus aplikasi yang tidak lagi diperlukan di lingkungan sumber Anda.

## A

### A2A () Agent-to-Agent

Protokol stateful untuk kolaborasi agen-ke-agen yang mendukung delegasi tugas dan transfer negara.

### ABAC

Lihat [kontrol akses berbasis atribut](#).

### layanan abstrak

Lihat [layanan terkelola](#).

### ASAM

Lihat [atomisitas, konsistensi, isolasi, daya tahan](#).

### migrasi aktif-aktif

Metode migrasi database di mana database sumber dan target tetap sinkron (dengan menggunakan alat replikasi dua arah atau operasi penulisan ganda), dan kedua database menangani transaksi dari menghubungkan aplikasi selama migrasi. Metode ini mendukung migrasi dalam batch kecil yang terkontrol alih-alih memerlukan pemotongan satu kali. Ini lebih fleksibel tetapi membutuhkan lebih banyak pekerjaan daripada migrasi [aktif-pasif](#).

### migrasi aktif-pasif

Metode migrasi database di mana database sumber dan target disimpan dalam sinkron, tetapi hanya database sumber yang menangani transaksi dari menghubungkan aplikasi sementara data direplikasi ke database target. Basis data target tidak menerima transaksi apa pun selama migrasi.

### Agen

Sistem AI yang dapat secara mandiri bernalar, merencanakan, dan mengambil tindakan menggunakan alat untuk mencapai tujuan.

## Agen Ops

Praktik operasional untuk membangun, menguji, menyebarkan, dan menjalankan agen AI dalam produksi dalam skala besar.

## fungsi agregat

Fungsi SQL yang beroperasi pada sekelompok baris dan menghitung nilai pengembalian tunggal untuk grup. Contoh fungsi agregat meliputi SUM dan MAX.

## AI

Lihat [kecerdasan buatan](#).

## AIOps

Lihat [operasi kecerdasan buatan](#).

## anonimisasi

Proses menghapus informasi pribadi secara permanen dalam kumpulan data. Anonimisasi dapat membantu melindungi privasi pribadi. Data anonim tidak lagi dianggap sebagai data pribadi.

## anti-pola

Solusi yang sering digunakan untuk masalah berulang di mana solusinya kontra-produktif, tidak efektif, atau kurang efektif daripada alternatif.

## kontrol aplikasi

Pendekatan keamanan yang memungkinkan penggunaan hanya aplikasi yang disetujui untuk membantu melindungi sistem dari malware.

## portofolio aplikasi

Kumpulan informasi rinci tentang setiap aplikasi yang digunakan oleh organisasi, termasuk biaya untuk membangun dan memelihara aplikasi, dan nilai bisnisnya. Informasi ini adalah kunci untuk [penemuan portofolio dan proses analisis dan](#) membantu mengidentifikasi dan memprioritaskan aplikasi yang akan dimigrasi, dimodernisasi, dan dioptimalkan.

## kecerdasan buatan (AI)

Bidang ilmu komputer yang didedikasikan untuk menggunakan teknologi komputasi untuk melakukan fungsi kognitif yang biasanya terkait dengan manusia, seperti belajar, memecahkan masalah, dan mengenali pola. Untuk informasi lebih lanjut, lihat [Apa itu Kecerdasan Buatan?](#)

## operasi kecerdasan buatan (AIOps)

Proses menggunakan teknik pembelajaran mesin untuk memecahkan masalah operasional, mengurangi insiden operasional dan intervensi manusia, dan meningkatkan kualitas layanan. Untuk informasi selengkapnya tentang cara AIOps digunakan dalam strategi AWS migrasi, lihat [panduan integrasi operasi](#).

## enkripsi asimetris

Algoritma enkripsi yang menggunakan sepasang kunci, kunci publik untuk enkripsi dan kunci pribadi untuk dekripsi. Anda dapat berbagi kunci publik karena tidak digunakan untuk dekripsi, tetapi akses ke kunci pribadi harus sangat dibatasi.

## atomisitas, konsistensi, isolasi, daya tahan (ACID)

Satu set properti perangkat lunak yang menjamin validitas data dan keandalan operasional database, bahkan dalam kasus kesalahan, kegagalan daya, atau masalah lainnya.

## kontrol akses berbasis atribut (ABAC)

Praktik membuat izin berbutir halus berdasarkan atribut pengguna, seperti departemen, peran pekerjaan, dan nama tim. Untuk informasi selengkapnya, lihat [ABAC untuk AWS](#) dokumentasi AWS Identity and Access Management (IAM).

## sumber data otoritatif

Lokasi di mana Anda menyimpan versi utama data, yang dianggap sebagai sumber informasi yang paling dapat diandalkan. Anda dapat menyalin data dari sumber data otoritatif ke lokasi lain untuk tujuan memproses atau memodifikasi data, seperti menganonimkan, menyunting, atau membuat nama samaran.

## Zona Ketersediaan

Lokasi berbeda di dalam Wilayah AWS yang terisolasi dari kegagalan di Availability Zone lainnya dan menyediakan konektivitas jaringan latensi rendah yang murah ke Availability Zone lainnya di Wilayah yang sama.

## AWS Kerangka Adopsi Cloud (AWS CAF)

Kerangka pedoman dan praktik terbaik AWS untuk membantu organisasi mengembangkan rencana yang efisien dan efektif untuk bergerak dengan sukses ke cloud. AWS CAF mengatur panduan ke dalam enam area fokus yang disebut perspektif: bisnis, orang, tata kelola, platform, keamanan, dan operasi. Perspektif bisnis, orang, dan tata kelola fokus pada keterampilan dan proses bisnis; perspektif platform, keamanan, dan operasi fokus pada keterampilan dan proses teknis. Misalnya, perspektif masyarakat menargetkan pemangku kepentingan yang menangani

sumber daya manusia (SDM), fungsi kepegawaian, dan manajemen orang. Untuk perspektif ini, AWS CAF memberikan panduan untuk pengembangan, pelatihan, dan komunikasi orang untuk membantu mempersiapkan organisasi untuk adopsi cloud yang sukses. Untuk informasi lebih lanjut, lihat [situs web AWS CAF](#) dan [whitepaper AWS CAF](#).

## AWS Kerangka Kualifikasi Beban Kerja (AWS WQF)

Alat yang mengevaluasi beban kerja migrasi database, merekomendasikan strategi migrasi, dan memberikan perkiraan kerja. AWS WQF disertakan dengan AWS Schema Conversion Tool (AWS SCT). Ini menganalisis skema database dan objek kode, kode aplikasi, dependensi, dan karakteristik kinerja, dan memberikan laporan penilaian.

## B

### bot buruk

[Bot](#) yang dimaksudkan untuk mengganggu atau menyebabkan kerugian bagi individu atau organisasi.

### BCP

Lihat [perencanaan kontinuitas bisnis](#).

### grafik perilaku

Pandangan interaktif yang terpadu tentang perilaku dan interaksi sumber daya dari waktu ke waktu. Anda dapat menggunakan grafik perilaku dengan Amazon Detective untuk memeriksa upaya logon yang gagal, panggilan API yang mencurigakan, dan tindakan serupa. Untuk informasi selengkapnya, lihat [Data dalam grafik perilaku](#) di dokumentasi Detektif.

### sistem big-endian

Sistem yang menyimpan byte paling signifikan terlebih dahulu. Lihat juga [endianness](#).

### klasifikasi biner

Sebuah proses yang memprediksi hasil biner (salah satu dari dua kelas yang mungkin). Misalnya, model ML Anda mungkin perlu memprediksi masalah seperti “Apakah email ini spam atau bukan spam?” atau “Apakah produk ini buku atau mobil?”

### filter mekar

Struktur data probabilistik dan efisien memori yang digunakan untuk menguji apakah suatu elemen adalah anggota dari suatu himpunan.

## blue/green penyebaran

Strategi penyebaran tempat Anda membuat dua lingkungan yang terpisah namun identik. Anda menjalankan versi aplikasi saat ini di satu lingkungan (biru) dan versi aplikasi baru di lingkungan lain (hijau). Strategi ini membantu Anda dengan cepat memutar kembali dengan dampak minimal.

## bot

Aplikasi perangkat lunak yang menjalankan tugas otomatis melalui internet dan mensimulasikan aktivitas atau interaksi manusia. Beberapa bot berguna atau bermanfaat, seperti perayap web yang mengindeks informasi di internet. Beberapa bot lain, yang dikenal sebagai bot buruk, dimaksudkan untuk mengganggu atau membahayakan individu atau organisasi.

## botnet

Jaringan [bot](#) yang terinfeksi oleh [malware](#) dan berada di bawah kendali satu pihak, yang dikenal sebagai bot herder atau operator bot. Botnet adalah mekanisme paling terkenal untuk skala bot dan dampaknya.

## cabang

Area berisi repositori kode. Cabang pertama yang dibuat dalam repositori adalah cabang utama. Anda dapat membuat cabang baru dari cabang yang ada, dan Anda kemudian dapat mengembangkan fitur atau memperbaiki bug di cabang baru. Cabang yang Anda buat untuk membangun fitur biasanya disebut sebagai cabang fitur. Saat fitur siap dirilis, Anda menggabungkan cabang fitur kembali ke cabang utama. Untuk informasi selengkapnya, lihat [Tentang cabang](#) (GitHub dokumentasi).

## akses break-glass

Dalam keadaan luar biasa dan melalui proses yang disetujui, cara cepat bagi pengguna untuk mendapatkan akses ke Akun AWS yang biasanya tidak memiliki izin untuk mengaksesnya. Untuk informasi lebih lanjut, lihat indikator [Implementasikan prosedur break-glass](#) dalam panduan. AWS Well-Architected

## strategi brownfield

Infrastruktur yang ada di lingkungan Anda. Saat mengadopsi strategi brownfield untuk arsitektur sistem, Anda merancang arsitektur di sekitar kendala sistem dan infrastruktur saat ini. Jika Anda memperluas infrastruktur yang ada, Anda dapat memadukan strategi brownfield dan [greenfield](#).

## cache penyangga

Area memori tempat data yang paling sering diakses disimpan.

## kemampuan bisnis

Apa yang dilakukan bisnis untuk menghasilkan nilai (misalnya, penjualan, layanan pelanggan, atau pemasaran). Arsitektur layanan mikro dan keputusan pengembangan dapat didorong oleh kemampuan bisnis. Untuk informasi selengkapnya, lihat bagian [Terorganisir di sekitar kemampuan bisnis](#) dari [Menjalankan layanan mikro kontainer](#) di whitepaper. AWS

## perencanaan kelangsungan bisnis (BCP)

Rencana yang membahas dampak potensial dari peristiwa yang mengganggu, seperti migrasi skala besar, pada operasi dan memungkinkan bisnis untuk melanjutkan operasi dengan cepat.

# C

## KAFE

Lihat [Kerangka Adopsi AWS Cloud](#).

## penyebaran kenari

Rilis versi yang lambat dan bertahap untuk pengguna akhir. Ketika Anda yakin, Anda menyebarkan versi baru dan mengganti versi saat ini secara keseluruhan.

## CCoE

Lihat [Cloud Center of Excellence](#).

## CDC

Lihat [mengubah pengambilan data](#).

## ubah pengambilan data (CDC)

Proses melacak perubahan ke sumber data, seperti tabel database, dan merekam metadata tentang perubahan tersebut. Anda dapat menggunakan CDC untuk berbagai tujuan, seperti mengaudit atau mereplikasi perubahan dalam sistem target untuk mempertahankan sinkronisasi.

## rekayasa kekacauan

Sengaja memperkenalkan kegagalan atau peristiwa yang mengganggu untuk menguji ketahanan sistem. Anda dapat menggunakan [AWS Fault Injection Service \(AWS FIS\)](#) untuk melakukan eksperimen yang menekankan AWS beban kerja Anda dan mengevaluasi responsnya.

## CI/CD

Lihat [integrasi berkelanjutan dan pengiriman berkelanjutan](#).

## klasifikasi

Proses kategorisasi yang membantu menghasilkan prediksi. Model ML untuk masalah klasifikasi memprediksi nilai diskrit. Nilai diskrit selalu berbeda satu sama lain. Misalnya, model mungkin perlu mengevaluasi apakah ada mobil dalam gambar atau tidak.

## Pengembang Warga

Pengguna bisnis yang membuat aplikasi AI menggunakan platform tanpa code/low kode tanpa keterampilan teknis khusus.

## Enkripsi sisi klien

Enkripsi data secara lokal, sebelum target Layanan AWS menerimanya.

## Cloud Center of Excellence (CCoE)

Tim multi-disiplin yang mendorong upaya adopsi cloud di seluruh organisasi, termasuk mengembangkan praktik terbaik cloud, memobilisasi sumber daya, menetapkan jadwal migrasi, dan memimpin organisasi melalui transformasi skala besar. Untuk informasi selengkapnya, lihat [posting CCoE](#) di Blog Strategi AWS Cloud Perusahaan.

## komputasi cloud

Teknologi cloud yang biasanya digunakan untuk penyimpanan data jarak jauh dan manajemen perangkat IoT. Cloud computing umumnya terhubung ke teknologi [edge computing](#).

## model operasi cloud

Dalam organisasi TI, model operasi yang digunakan untuk membangun, mematangkan, dan mengoptimalkan satu atau lebih lingkungan cloud. Untuk informasi selengkapnya, lihat [Membangun Model Operasi Cloud Anda](#).

## tahap adopsi cloud

Empat fase yang biasanya dilalui organisasi ketika mereka bermigrasi ke AWS Cloud:

- Proyek — Menjalankan beberapa proyek terkait cloud untuk bukti konsep dan tujuan pembelajaran
- Foundation — Melakukan investasi dasar untuk meningkatkan adopsi cloud Anda (misalnya, membuat landing zone, mendefinisikan CCoE, membuat model operasi)
- Migrasi — Migrasi aplikasi individual
- Re-invention — Mengoptimalkan produk dan layanan, dan berinovasi di cloud

Tahapan ini didefinisikan oleh Stephen Orban dalam posting blog [The Journey Toward Cloud-First & the Stages of Adoption](#) di blog Strategi AWS Cloud Perusahaan. Untuk informasi tentang bagaimana kaitannya dengan strategi AWS migrasi, lihat [panduan kesiapan migrasi](#).

## CMDB

Lihat [database manajemen konfigurasi](#).

## repositori kode

Lokasi di mana kode sumber dan aset lainnya, seperti dokumentasi, sampel, dan skrip, disimpan dan diperbarui melalui proses kontrol versi. Repositori cloud umum termasuk GitHub atau Bitbucket Cloud. Setiap versi kode disebut cabang. Dalam struktur layanan mikro, setiap repositori dikhususkan untuk satu bagian fungsionalitas. Satu CI/CD pipa dapat menggunakan beberapa repositori.

## cache dingin

Cache buffer yang kosong, tidak terisi dengan baik, atau berisi data basi atau tidak relevan. Ini mempengaruhi kinerja karena instance database harus membaca dari memori utama atau disk, yang lebih lambat daripada membaca dari cache buffer.

## data dingin

Data yang jarang diakses dan biasanya historis. Saat menanyakan jenis data ini, kueri lambat biasanya dapat diterima. Memindahkan data ini ke tingkat penyimpanan atau kelas yang berkinerja lebih rendah dan lebih murah dapat mengurangi biaya.

## visi komputer (CV)

Bidang [AI](#) yang menggunakan pembelajaran mesin untuk menganalisis dan mengekstrak informasi dari format visual seperti gambar dan video digital. Misalnya, Amazon SageMaker AI menyediakan algoritma pemrosesan gambar untuk CV.

## konfigurasi drift

Untuk beban kerja, konfigurasi berubah dari status yang diharapkan. Ini dapat menyebabkan beban kerja menjadi tidak patuh, dan biasanya bertahap dan tidak disengaja.

## database manajemen konfigurasi (CMDB)

Repositori yang menyimpan dan mengelola informasi tentang database dan lingkungan TI, termasuk komponen perangkat keras dan perangkat lunak dan konfigurasinya. Anda biasanya menggunakan data dari CMDB dalam penemuan portofolio dan tahap analisis migrasi.

## paket kesesuaian

Kumpulan AWS Config aturan dan tindakan remediasi yang dapat Anda kumpulkan untuk menyesuaikan kepatuhan dan pemeriksaan keamanan Anda. Anda dapat menerapkan paket kesesuaian sebagai entitas tunggal di Akun AWS dan Region, atau di seluruh organisasi, dengan menggunakan templat YAMM. Untuk informasi selengkapnya, lihat [Paket kesesuaian dalam dokumentasi](#). AWS Config

## integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD)

Proses mengotomatiskan sumber, membangun, menguji, pementasan, dan tahap produksi dari proses rilis perangkat lunak. CI/CD biasanya digambarkan sebagai pipa. CI/CD dapat membantu Anda mengotomatiskan proses, meningkatkan produktivitas, meningkatkan kualitas kode, dan memberikan lebih cepat. Untuk informasi lebih lanjut, lihat [Manfaat pengiriman berkelanjutan](#). CD juga dapat berarti penerapan berkelanjutan. Untuk informasi selengkapnya, lihat [Continuous Delivery vs Continuous Deployment](#).

## CV

Lihat [visi komputer](#).

## D

### data saat istirahat

Data yang stasioner di jaringan Anda, seperti data yang ada di penyimpanan.

### klasifikasi data

Proses untuk mengidentifikasi dan mengkategorikan data dalam jaringan Anda berdasarkan kekritisannya dan sensitivitasnya. Ini adalah komponen penting dari setiap strategi manajemen risiko keamanan siber karena membantu Anda menentukan perlindungan dan kontrol retensi yang tepat untuk data. Klasifikasi data adalah komponen pilar keamanan dalam AWS Well-Architected Framework. Untuk informasi selengkapnya, lihat [Klasifikasi data](#).

### penyimpangan data

Variasi yang berarti antara data produksi dan data yang digunakan untuk melatih model ML, atau perubahan yang berarti dalam data input dari waktu ke waktu. Penyimpangan data dapat mengurangi kualitas, akurasi, dan keadilan keseluruhan dalam prediksi model ML.

## data dalam transit

Data yang aktif bergerak melalui jaringan Anda, seperti antara sumber daya jaringan.

## jala data

Kerangka arsitektur yang menyediakan kepemilikan data terdistribusi dan terdesentralisasi dengan manajemen dan tata kelola terpusat.

## minimalisasi data

Prinsip pengumpulan dan pemrosesan hanya data yang sangat diperlukan. Mempraktikkan minimalisasi data di dalamnya AWS Cloud dapat mengurangi risiko privasi, biaya, dan jejak karbon analitik Anda.

## perimeter data

Satu set pagar pembatas pencegahan di AWS lingkungan Anda yang membantu memastikan bahwa hanya identitas tepercaya yang mengakses sumber daya tepercaya dari jaringan yang diharapkan. Untuk informasi selengkapnya, lihat [Membangun perimeter data pada AWS](#).

## prapemrosesan data

Untuk mengubah data mentah menjadi format yang mudah diuraikan oleh model ML Anda. Preprocessing data dapat berarti menghapus kolom atau baris tertentu dan menangani nilai yang hilang, tidak konsisten, atau duplikat.

## asal data

Proses melacak asal dan riwayat data sepanjang siklus hidupnya, seperti bagaimana data dihasilkan, ditransmisikan, dan disimpan.

## subjek data

Individu yang datanya dikumpulkan dan diproses.

## gudang data

Sistem manajemen data yang mendukung intelijen bisnis, seperti analitik. Gudang data biasanya berisi sejumlah besar data historis, dan biasanya digunakan untuk kueri dan analisis.

## bahasa definisi database (DDL)

Pernyataan atau perintah untuk membuat atau memodifikasi struktur tabel dan objek dalam database.

## bahasa manipulasi basis data (DHTML)

Pernyataan atau perintah untuk memodifikasi (memasukkan, memperbarui, dan menghapus) informasi dalam database.

## DDL

Lihat [bahasa definisi database](#).

## ansambel yang dalam

Untuk menggabungkan beberapa model pembelajaran mendalam untuk prediksi. Anda dapat menggunakan ansambel dalam untuk mendapatkan prediksi yang lebih akurat atau untuk memperkirakan ketidakpastian dalam prediksi.

## pembelajaran mendalam

Subbidang ML yang menggunakan beberapa lapisan jaringan saraf tiruan untuk mengidentifikasi pemetaan antara data input dan variabel target yang diinginkan.

## pertahanan-mendalam

Pendekatan keamanan informasi di mana serangkaian mekanisme dan kontrol keamanan dilapisi dengan cermat di seluruh jaringan komputer untuk melindungi kerahasiaan, integritas, dan ketersediaan jaringan dan data di dalamnya. Saat Anda mengadopsi strategi ini AWS, Anda menambahkan beberapa kontrol pada lapisan AWS Organizations struktur yang berbeda untuk membantu mengamankan sumber daya. Misalnya, pendekatan defense-in-depth mungkin menggabungkan otentikasi multi-faktor, segmentasi jaringan, dan enkripsi.

## administrator yang didelegasikan

Di AWS Organizations, layanan yang kompatibel dapat mendaftarkan akun AWS anggota untuk mengelola akun organisasi dan mengelola izin untuk layanan tersebut. Akun ini disebut administrator yang didelegasikan untuk layanan itu. Untuk informasi selengkapnya dan daftar layanan yang kompatibel, lihat [Layanan yang berfungsi dengan AWS Organizations](#) AWS Organizations dokumentasi.

## deployment

Proses pembuatan aplikasi, fitur baru, atau perbaikan kode tersedia di lingkungan target. Deployment melibatkan penerapan perubahan dalam basis kode dan kemudian membangun dan menjalankan basis kode itu di lingkungan aplikasi.

## lingkungan pengembangan

Lihat [lingkungan](#).

## kontrol detektif

Kontrol keamanan yang dirancang untuk mendeteksi, mencatat, dan memperingatkan setelah suatu peristiwa terjadi. Kontrol ini adalah garis pertahanan kedua, memperingatkan Anda tentang peristiwa keamanan yang melewati kontrol pencegahan yang ada. Untuk informasi selengkapnya, lihat Kontrol [Detektif dalam Menerapkan kontrol](#) keamanan pada. AWS

## pemetaan aliran nilai pengembangan (DVSM)

Sebuah proses yang digunakan untuk mengidentifikasi dan memprioritaskan kendala yang mempengaruhi kecepatan dan kualitas dalam siklus hidup pengembangan perangkat lunak. DVSM memperluas proses pemetaan aliran nilai yang awalnya dirancang untuk praktik manufaktur ramping. Ini berfokus pada langkah-langkah dan tim yang diperlukan untuk menciptakan dan memindahkan nilai melalui proses pengembangan perangkat lunak.

## kembar digital

Representasi virtual dari sistem dunia nyata, seperti bangunan, pabrik, peralatan industri, atau jalur produksi. Kembar digital mendukung pemeliharaan prediktif, pemantauan jarak jauh, dan optimalisasi produksi.

## tabel dimensi

Dalam [skema bintang](#), tabel yang lebih kecil yang berisi atribut data tentang data kuantitatif dalam tabel fakta. Atribut tabel dimensi biasanya bidang teks atau angka diskrit yang berperilaku seperti teks. Atribut ini biasanya digunakan untuk pembatasan kueri, pemfilteran, dan pelabelan set hasil.

## musibah

Peristiwa yang mencegah beban kerja atau sistem memenuhi tujuan bisnisnya di lokasi utama yang digunakan. Peristiwa ini dapat berupa bencana alam, kegagalan teknis, atau akibat dari tindakan manusia, seperti kesalahan konfigurasi yang tidak disengaja atau serangan malware.

## pemulihan bencana (DR)

Strategi dan proses yang Anda gunakan untuk meminimalkan downtime dan kehilangan data yang disebabkan oleh [bencana](#). Untuk informasi selengkapnya, lihat [Disaster Recovery of Workloads on AWS: Recovery in the Cloud](#) in the AWS Well-Architected Framework.

## DML~

Lihat [bahasa manipulasi database](#).

## desain berbasis domain

Pendekatan untuk mengembangkan sistem perangkat lunak yang kompleks dengan menghubungkan komponennya ke domain yang berkembang, atau tujuan bisnis inti, yang dilayani setiap komponen. Konsep ini diperkenalkan oleh Eric Evans dalam bukunya, *Domain-Driven Design: Tackling Complexity in the Heart of Software* (Boston: Addison-Wesley Professional, 2003). Untuk informasi tentang cara menggunakan desain berbasis domain dengan pola gambar pencekik, lihat Memodernisasi layanan [web Microsoft ASP.NET \(ASMX\) lama](#) secara bertahap menggunakan container dan Amazon API Gateway.

## DR

Lihat [pemulihan bencana](#).

## deteksi drift

Melacak penyimpangan dari konfigurasi dasar. Misalnya, Anda dapat menggunakan AWS CloudFormation untuk [mendeteksi penyimpangan dalam sumber daya sistem](#), atau Anda dapat menggunakannya AWS Control Tower untuk [mendeteksi perubahan di landing zone](#) yang mungkin memengaruhi kepatuhan terhadap persyaratan tata kelola.

## DVSM

Lihat [pemetaan aliran nilai pengembangan](#).

## E

### EDA

Lihat [analisis data eksplorasi](#).

### EDI

Lihat [pertukaran data elektronik](#).

## komputasi tepi

Teknologi yang meningkatkan daya komputasi untuk perangkat pintar di tepi jaringan IoT. Jika dibandingkan dengan [komputasi awan](#), komputasi tepi dapat mengurangi latensi komunikasi dan meningkatkan waktu respons.

## pertukaran data elektronik (EDI)

Pertukaran otomatis dokumen bisnis antar organisasi. Untuk informasi selengkapnya, lihat [Apa itu Pertukaran Data Elektronik](#).

## enkripsi

Proses komputasi yang mengubah data plaintext, yang dapat dibaca manusia, menjadi ciphertext.

### kunci enkripsi

String kriptografi dari bit acak yang dihasilkan oleh algoritma enkripsi. Panjang kunci dapat bervariasi, dan setiap kunci dirancang agar tidak dapat diprediksi dan unik.

### endianness

Urutan byte disimpan dalam memori komputer. Big-endian sistem menyimpan byte paling signifikan terlebih dahulu. Little-endian sistem menyimpan byte paling tidak signifikan terlebih dahulu.

### titik akhir

Lihat [titik akhir layanan](#).

### layanan endpoint

Layanan yang dapat Anda host di cloud pribadi virtual (VPC) untuk dibagikan dengan pengguna lain. Anda dapat membuat layanan endpoint dengan AWS PrivateLink dan memberikan izin kepada prinsipal lain Akun AWS atau ke AWS Identity and Access Management (IAM). Akun atau prinsipal ini dapat terhubung ke layanan endpoint Anda secara pribadi dengan membuat titik akhir VPC antarmuka. Untuk informasi selengkapnya, lihat [Membuat layanan titik akhir](#) di dokumentasi Amazon Virtual Private Cloud (Amazon VPC).

### perencanaan sumber daya perusahaan (ERP)

Sistem yang mengotomatiskan dan mengelola proses bisnis utama (seperti akuntansi, [MES](#), dan manajemen proyek) untuk suatu perusahaan.

### enkripsi amplop

Proses mengenkripsi kunci enkripsi dengan kunci enkripsi lain. Untuk informasi selengkapnya, lihat [Enkripsi amplop](#) dalam dokumentasi AWS Key Management Service (AWS KMS).

### lingkungan

Sebuah contoh dari aplikasi yang sedang berjalan. Berikut ini adalah jenis lingkungan yang umum dalam komputasi awan:

- **Development Environment** — Sebuah contoh dari aplikasi yang berjalan yang hanya tersedia untuk tim inti yang bertanggung jawab untuk memelihara aplikasi. Lingkungan pengembangan digunakan untuk menguji perubahan sebelum mempromosikannya ke lingkungan atas. Jenis lingkungan ini kadang-kadang disebut sebagai lingkungan pengujian.

- lingkungan yang lebih rendah — Semua lingkungan pengembangan untuk aplikasi, seperti yang digunakan untuk build awal dan pengujian.
- lingkungan produksi — Sebuah contoh dari aplikasi yang berjalan yang dapat diakses oleh pengguna akhir. Dalam sebuah CI/CD pipeline, lingkungan produksi adalah lingkungan penyebaran terakhir.
- lingkungan atas — Semua lingkungan yang dapat diakses oleh pengguna selain tim pengembangan inti. Ini dapat mencakup lingkungan produksi, lingkungan praproduksi, dan lingkungan untuk pengujian penerimaan pengguna.

## epik

Dalam metodologi tangkas, kategori fungsional yang membantu mengatur dan memprioritaskan pekerjaan Anda. Epik memberikan deskripsi tingkat tinggi tentang persyaratan dan tugas implementasi. Misalnya, epos keamanan AWS CAF mencakup manajemen identitas dan akses, kontrol detektif, keamanan infrastruktur, perlindungan data, dan respons insiden. Untuk informasi selengkapnya tentang epos dalam strategi AWS migrasi, lihat [panduan implementasi program](#).

## ERP

Lihat [perencanaan sumber daya perusahaan](#).

## analisis data eksplorasi (EDA)

Proses menganalisis dataset untuk memahami karakteristik utamanya. Anda mengumpulkan atau mengumpulkan data dan kemudian melakukan penyelidikan awal untuk menemukan pola, mendeteksi anomali, dan memeriksa asumsi. EDA dilakukan dengan menghitung statistik ringkasan dan membuat visualisasi data.

## F

### tabel fakta

Tabel tengah dalam [skema bintang](#). Ini menyimpan data kuantitatif tentang operasi bisnis. Biasanya, tabel fakta berisi dua jenis kolom: kolom yang berisi ukuran dan yang berisi kunci asing ke tabel dimensi.

### gagal cepat

Filosofi yang menggunakan pengujian yang sering dan bertahap untuk mengurangi siklus hidup pengembangan. Ini adalah bagian penting dari pendekatan tangkas.

## batas isolasi kesalahan

Dalam AWS Cloud, batas seperti Availability Zone, Wilayah AWS, control plane, atau data plane yang membatasi efek kegagalan dan membantu meningkatkan ketahanan beban kerja. Untuk informasi selengkapnya, lihat [Batas Isolasi AWS Kesalahan](#).

## cabang fitur

Lihat [cabang](#).

## fitur

Data input yang Anda gunakan untuk membuat prediksi. Misalnya, dalam konteks manufaktur, fitur bisa berupa gambar yang diambil secara berkala dari lini manufaktur.

## pentingnya fitur

Seberapa signifikan fitur untuk prediksi model. Ini biasanya dinyatakan sebagai skor numerik yang dapat dihitung melalui berbagai teknik, seperti Shapley Additive Explanations (SHAP) dan gradien terintegrasi. Untuk informasi lebih lanjut, lihat [Interpretabilitas model pembelajaran mesin](#) dengan AWS

## transformasi fitur

Untuk mengoptimalkan data untuk proses ML, termasuk memperkaya data dengan sumber tambahan, menskalakan nilai, atau mengekstrak beberapa set informasi dari satu bidang data. Hal ini memungkinkan model ML untuk mendapatkan keuntungan dari data. Misalnya, jika Anda memecah tanggal "2021-05-27 00:15:37" menjadi "2021", "Mei", "Kamis", dan "15", Anda dapat membantu algoritme pembelajaran mempelajari pola bernuansa yang terkait dengan komponen data yang berbeda.

## beberapa tembakan mendorong

Menyediakan [LLM](#) dengan sejumlah kecil contoh yang menunjukkan tugas dan output yang diinginkan sebelum memintanya untuk melakukan tugas serupa. Teknik ini adalah aplikasi pembelajaran dalam konteks, di mana model belajar dari contoh (bidikan) yang tertanam dalam petunjuk. Few-shot prompt bisa efektif untuk tugas-tugas yang membutuhkan pemformatan, penalaran, atau pengetahuan domain tertentu. Lihat juga [bidikan nol](#).

## FGAC

Lihat kontrol [akses berbutir halus](#).

## kontrol akses berbutir halus (FGAC)

Penggunaan beberapa kondisi untuk mengizinkan atau menolak permintaan akses.

## migrasi flash-cut

Metode migrasi database yang menggunakan replikasi data berkelanjutan melalui [pengambilan data perubahan](#) untuk memigrasikan data dalam waktu sesingkat mungkin, alih-alih menggunakan pendekatan bertahap. Tujuannya adalah untuk menjaga downtime seminimal mungkin.

## FM

Lihat [model pondasi](#).

## model pondasi (FM)

Jaringan saraf pembelajaran mendalam yang besar yang telah melatih kumpulan data besar-besaran data umum dan tidak berlabel. FM mampu melakukan berbagai tugas umum, seperti memahami bahasa, menghasilkan teks dan gambar, dan berbicara dalam bahasa alami. Untuk informasi selengkapnya, lihat [Apa itu Model Foundation](#).

## Gerbang FM

[Perantara terpusat yang mengontrol dan menormalkan akses ke model pondasi](#). Juga dikenal sebagai gateway LLM.

## G

### AI generatif

Subset model [AI](#) yang telah dilatih pada sejumlah besar data dan yang dapat menggunakan prompt teks sederhana untuk membuat konten dan artefak baru, seperti gambar, video, teks, dan audio. Untuk informasi lebih lanjut, lihat [Apa itu AI Generatif](#).

### pemblokiran geografis

Lihat [pembatasan geografis](#).

### pembatasan geografis (pemblokiran geografis)

Di Amazon CloudFront, opsi untuk mencegah pengguna di negara tertentu mengakses distribusi konten. Anda dapat menggunakan daftar izinkan atau daftar blokir untuk menentukan negara yang disetujui dan dilarang. Untuk informasi selengkapnya, lihat [Membatasi distribusi geografis konten Anda](#) dalam dokumentasi CloudFront

## Alur kerja Gitflow

Pendekatan di mana lingkungan bawah dan atas menggunakan cabang yang berbeda dalam repositori kode sumber. Alur kerja Gitflow dianggap warisan, dan [alur kerja berbasis batang](#) adalah pendekatan modern yang lebih disukai.

### gambar emas

Sebuah snapshot dari sistem atau perangkat lunak yang digunakan sebagai template untuk menyebarkan instance baru dari sistem atau perangkat lunak itu. Misalnya, di bidang manufaktur, gambar emas dapat digunakan untuk menyediakan perangkat lunak pada beberapa perangkat dan membantu meningkatkan kecepatan, skalabilitas, dan produktivitas dalam operasi manufaktur perangkat.

### strategi greenfield

Tidak adanya infrastruktur yang ada di lingkungan baru. [Saat mengadopsi strategi greenfield untuk arsitektur sistem, Anda dapat memilih semua teknologi baru tanpa batasan kompatibilitas dengan infrastruktur yang ada, juga dikenal sebagai brownfield.](#) Jika Anda memperluas infrastruktur yang ada, Anda dapat memadukan strategi brownfield dan greenfield.

### pagar pembatas

Aturan tingkat tinggi yang membantu mengatur sumber daya, kebijakan, dan kepatuhan di seluruh unit organisasi (OU). Pagar pembatas preventif menegakkan kebijakan untuk memastikan keselarasan dengan standar kepatuhan. Mereka diimplementasikan dengan menggunakan kebijakan kontrol layanan dan batas izin IAM. Detective guardrails mendeteksi pelanggaran kebijakan dan masalah kepatuhan, dan menghasilkan peringatan untuk remediasi. Mereka diimplementasikan dengan menggunakan AWS Config, AWS Security Hub CSPM, Amazon GuardDuty AWS Trusted Advisor, Amazon Inspector, dan pemeriksaan khusus AWS Lambda .

### pagar pembatas (AI)

Mekanisme keamanan yang menyaring, memvalidasi, dan membatasi input dan output [agen](#) untuk membantu memastikan perilaku AI yang bertanggung jawab dan aman.

## H

### HA

Lihat [ketersediaan tinggi](#).

## migrasi database heterogen

Memigrasi database sumber Anda ke database target yang menggunakan mesin database yang berbeda (misalnya, Oracle ke Amazon Aurora). Migrasi heterogen biasanya merupakan bagian dari upaya arsitektur ulang, dan mengubah skema dapat menjadi tugas yang kompleks. [AWS menyediakan AWS SCT](#) yang membantu dengan konversi skema.

## ketersediaan tinggi (HA)

Kemampuan beban kerja untuk beroperasi terus menerus, tanpa intervensi, jika terjadi tantangan atau bencana. Sistem HA dirancang untuk gagal secara otomatis, secara konsisten memberikan kinerja berkualitas tinggi, dan menangani beban dan kegagalan yang berbeda dengan dampak kinerja minimal.

## modernisasi sejarawan

Pendekatan yang digunakan untuk memodernisasi dan meningkatkan sistem teknologi operasional (OT) untuk melayani kebutuhan industri manufaktur dengan lebih baik. Sejarawan adalah jenis database yang digunakan untuk mengumpulkan dan menyimpan data dari berbagai sumber di pabrik.

## data penahanan

Sebagian dari data historis berlabel yang ditahan dari kumpulan data yang digunakan untuk melatih model pembelajaran [mesin](#). Anda dapat menggunakan data penahanan untuk mengevaluasi kinerja model dengan membandingkan prediksi model dengan data penahanan.

## manusia-dalam-lingkaran (HiTL)

Pola alur kerja di mana eksekusi [agen](#) berhenti untuk peninjauan dan persetujuan manusia pada titik keputusan kritis.

## migrasi database homogen

Memigrasi database sumber Anda ke database target yang berbagi mesin database yang sama (misalnya, Microsoft SQL Server ke Amazon RDS for SQL Server). Migrasi homogen biasanya merupakan bagian dari upaya rehosting atau replatforming. Anda dapat menggunakan utilitas database asli untuk memigrasi skema.

## data panas

Data yang sering diakses, seperti data real-time atau data translasi terbaru. Data ini biasanya memerlukan tingkat atau kelas penyimpanan berkinerja tinggi untuk memberikan respons kueri yang cepat.

## perbaikan terbaru

Perbaikan mendesak untuk masalah kritis dalam lingkungan produksi. Karena urgensinya, perbaikan terbaru biasanya dibuat di luar alur kerja DevOps rilis biasa.

## periode hypercare

Segera setelah cutover, periode waktu ketika tim migrasi mengelola dan memantau aplikasi yang dimigrasi di cloud untuk mengatasi masalah apa pun. Biasanya, periode ini panjangnya 1-4 hari. Pada akhir periode hypercare, tim migrasi biasanya mentransfer tanggung jawab untuk aplikasi ke tim operasi cloud.

## I

### IAC

Lihat [infrastruktur sebagai kode](#).

### kebijakan berbasis identitas

Kebijakan yang dilampirkan pada satu atau beberapa prinsip IAM yang mendefinisikan izin mereka dalam lingkungan. AWS Cloud

### aplikasi idle

Aplikasi yang memiliki penggunaan CPU dan memori rata-rata antara 5 dan 20 persen selama periode 90 hari. Dalam proyek migrasi, adalah umum untuk menghentikan aplikasi ini atau mempertahankannya di tempat.

### IIoT

Lihat [Internet of Things industri](#).

### infrastruktur yang tidak dapat diubah

Model yang menyebarkan infrastruktur baru untuk beban kerja produksi alih-alih memperbarui, menambal, atau memodifikasi infrastruktur yang ada. [Infrastruktur yang tidak dapat diubah secara inheren lebih konsisten, andal, dan dapat diprediksi daripada infrastruktur yang dapat berubah.](#)

Untuk informasi selengkapnya, lihat praktik terbaik [Deploy using immutable infrastructure](#) in the Framework. AWS Well-Architected

### masuk (masuknya) VPC

Dalam arsitektur AWS multi-akun, VPC yang menerima, memeriksa, dan merutekan koneksi jaringan dari luar aplikasi. [Arsitektur Referensi AWS Keamanan](#) merekomendasikan pengaturan

akun Jaringan Anda dengan VPC masuk, keluar, dan inspeksi untuk melindungi antarmuka dua arah antara aplikasi Anda dan internet yang lebih luas.

## migrasi inkremental

Strategi cutover di mana Anda memigrasikan aplikasi Anda dalam bagian-bagian kecil alih-alih melakukan satu cutover penuh. Misalnya, Anda mungkin hanya memindahkan beberapa layanan mikro atau pengguna ke sistem baru pada awalnya. Setelah Anda memverifikasi bahwa semuanya berfungsi dengan baik, Anda dapat secara bertahap memindahkan layanan mikro atau pengguna tambahan hingga Anda dapat menonaktifkan sistem lama Anda. Strategi ini mengurangi risiko yang terkait dengan migrasi besar.

## Industri 4.0

Sebuah istilah yang diperkenalkan oleh [Klaus Schwab](#) pada tahun 2016 untuk merujuk pada modernisasi proses manufaktur melalui kemajuan dalam konektivitas, data real-time, otomatisasi, analitik, dan AI/ML

## infrastruktur

Semua sumber daya dan aset yang terkandung dalam lingkungan aplikasi.

## infrastruktur sebagai kode (IAC)

Proses penyediaan dan pengelolaan infrastruktur aplikasi melalui satu set file konfigurasi. IAC dirancang untuk membantu Anda memusatkan manajemen infrastruktur, menstandarisasi sumber daya, dan menskalakan dengan cepat sehingga lingkungan baru dapat diulang, andal, dan konsisten.

## Internet of Things industri (IIoT)

Penggunaan sensor dan perangkat yang terhubung ke internet di sektor industri, seperti manufaktur, energi, otomotif, perawatan kesehatan, ilmu kehidupan, dan pertanian. Untuk informasi selengkapnya, lihat [Membangun strategi transformasi digital Internet of Things \(IIoT\) industri](#).

## inspeksi VPC

Dalam arsitektur AWS multi-akun, VPC terpusat yang mengelola inspeksi lalu lintas jaringan antara VPC (dalam hal yang sama atau berbeda Wilayah AWS), internet, dan jaringan lokal. [Arsitektur Referensi AWS Keamanan](#) merekomendasikan pengaturan akun Jaringan Anda dengan VPC masuk, keluar, dan inspeksi untuk melindungi antarmuka dua arah antara aplikasi Anda dan internet yang lebih luas.

## Internet of Things (IoT)

Jaringan objek fisik yang terhubung dengan sensor atau prosesor tertanam yang berkomunikasi dengan perangkat dan sistem lain melalui internet atau melalui jaringan komunikasi lokal. Untuk informasi selengkapnya, lihat [Apa itu IoT?](#)

## interpretabilitas

Karakteristik model pembelajaran mesin yang menggambarkan sejauh mana manusia dapat memahami bagaimana prediksi model bergantung pada inputnya. Untuk informasi lebih lanjut, lihat [Interpretabilitas model pembelajaran mesin](#) dengan AWS

## IoT

Lihat [Internet of Things](#).

## Perpustakaan informasi TI (ITIL)

Serangkaian praktik terbaik untuk memberikan layanan TI dan menyelaraskan layanan ini dengan persyaratan bisnis. ITIL menyediakan dasar untuk ITSM.

## Manajemen layanan TI (ITSM)

Kegiatan yang terkait dengan merancang, menerapkan, mengelola, dan mendukung layanan TI untuk suatu organisasi. Untuk informasi tentang mengintegrasikan operasi cloud dengan alat ITSM, lihat panduan [integrasi operasi](#).

## ITIL

Lihat [perpustakaan informasi TI](#).

## ITSM

Lihat [manajemen layanan TI](#).

## L

## kontrol akses berbasis label (LBAC)

Implementasi kontrol akses wajib (MAC) di mana pengguna dan data itu sendiri masing-masing secara eksplisit diberi nilai label keamanan. Persimpangan antara label keamanan pengguna dan label keamanan data menentukan baris dan kolom mana yang dapat dilihat oleh pengguna.

## landing zone

Landing zone adalah AWS lingkungan multi-akun yang dirancang dengan baik yang dapat diskalakan dan aman. Ini adalah titik awal dari mana organisasi Anda dapat dengan cepat meluncurkan dan menyebarkan beban kerja dan aplikasi dengan percaya diri dalam lingkungan keamanan dan infrastruktur mereka. Untuk informasi selengkapnya tentang zona pendaratan, lihat [Menyiapkan lingkungan multi-akun AWS yang aman dan dapat diskalakan](#).

## model bahasa besar (LLM)

Model [AI](#) pembelajaran mendalam yang dilatih sebelumnya pada sejumlah besar data. LLM dapat melakukan beberapa tugas, seperti menjawab pertanyaan, meringkas dokumen, menerjemahkan teks ke bahasa lain, dan menyelesaikan kalimat. Untuk informasi lebih lanjut, lihat [Apa itu LLM](#).

## migrasi besar

Migrasi 300 atau lebih server.

## LBAC

Lihat [kontrol akses berbasis label](#).

## hak istimewa paling sedikit

Praktik keamanan terbaik untuk memberikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi selengkapnya, lihat [Menerapkan izin hak istimewa terkecil dalam dokumentasi IAM](#).

## angkat dan geser

Lihat [7 Rs](#).

## sistem endian kecil

Sebuah sistem yang menyimpan byte paling tidak signifikan terlebih dahulu. Lihat juga [endianness](#).

## LLM

Lihat [model bahasa besar](#).

## lingkungan yang lebih rendah

Lihat [lingkungan](#).

# M

## pembelajaran mesin (ML)

Jenis kecerdasan buatan yang menggunakan algoritma dan teknik untuk pengenalan pola dan pembelajaran. ML menganalisis dan belajar dari data yang direkam, seperti data Internet of Things (IoT), untuk menghasilkan model statistik berdasarkan pola. Untuk informasi selengkapnya, lihat [Machine Learning](#).

## cabang utama

Lihat [cabang](#).

## malware

Perangkat lunak yang dirancang untuk membahayakan keamanan atau privasi komputer. Malware dapat mengganggu sistem komputer, membocorkan informasi sensitif, atau mendapatkan akses yang tidak sah. Contoh malware termasuk virus, worm, ransomware, Trojan horse, spyware, dan keyloggers.

## layanan terkelola

Layanan AWS yang AWS mengoperasikan lapisan infrastruktur, sistem operasi, dan platform, dan Anda mengakses titik akhir untuk menyimpan dan mengambil data. Amazon Simple Storage Service (Amazon S3) dan Amazon DynamoDB adalah contoh layanan terkelola. Ini juga dikenal sebagai layanan abstrak.

## sistem eksekusi manufaktur (MES)

Sistem perangkat lunak untuk melacak, memantau, mendokumentasikan, dan mengendalikan proses produksi yang mengubah bahan baku menjadi produk jadi di lantai toko.

## PETA

Lihat [Program Percepatan Migrasi](#).

## MCP

Lihat [Protokol Konteks Model](#).

## Protokol Konteks Model (MCP)

Protokol stateless untuk komunikasi [agen](#) -to- [alat](#).

## Server MCP

Layanan yang mengekspos satu atau lebih [alat](#) melalui [Protokol Konteks Model](#).

## mekanisme

Proses lengkap di mana Anda membuat alat, mendorong adopsi alat, dan kemudian memeriksa hasilnya untuk melakukan penyesuaian. Mekanisme adalah siklus yang memperkuat dan meningkatkan dirinya sendiri saat beroperasi. Untuk informasi selengkapnya, lihat [Membangun mekanisme](#) dalam AWS Well-Architected Kerangka Kerja.

## akun anggota

Semua Akun AWS selain akun manajemen yang merupakan bagian dari organisasi di AWS Organizations. Akun dapat menjadi anggota dari hanya satu organisasi pada suatu waktu.

## MES

Lihat [sistem eksekusi manufaktur](#).

## Transportasi Telemetri Antrian Pesan (MQTT)

[Protokol komunikasi mesin-ke-mesin \(M2M\) yang ringan, berdasarkan pola publish/subscribe, untuk perangkat IoT yang dibatasi sumber daya.](#)

## layanan mikro

Layanan kecil dan independen yang berkomunikasi melalui API yang terdefinisi dengan baik dan biasanya dimiliki oleh tim kecil yang mandiri. Misalnya, sistem asuransi mungkin mencakup layanan mikro yang memetakan kemampuan bisnis, seperti penjualan atau pemasaran, atau subdomain, seperti pembelian, klaim, atau analitik. Manfaat layanan mikro termasuk kelincahan, penskalaan yang fleksibel, penyebaran yang mudah, kode yang dapat digunakan kembali, dan ketahanan. Untuk informasi selengkapnya, lihat [Mengintegrasikan layanan mikro dengan menggunakan layanan tanpa AWS server](#).

## arsitektur microservices

Pendekatan untuk membangun aplikasi dengan komponen independen yang menjalankan setiap proses aplikasi sebagai layanan mikro. Layanan mikro ini berkomunikasi melalui antarmuka yang terdefinisi dengan baik dengan menggunakan API ringan. Setiap layanan mikro dalam arsitektur ini dapat diperbarui, digunakan, dan diskalakan untuk memenuhi permintaan fungsi tertentu dari suatu aplikasi. Untuk informasi selengkapnya, lihat [Menerapkan layanan mikro di AWS](#).

## Program Percepatan Migrasi (MAP)

AWS Program yang menyediakan dukungan konsultasi, pelatihan, dan layanan untuk membantu organisasi membangun fondasi operasional yang kuat untuk pindah ke cloud, dan untuk membantu mengimbangi biaya awal migrasi. MAP mencakup metodologi migrasi untuk

mengeksekusi migrasi lama dengan cara metodis dan seperangkat alat untuk mengotomatisasi dan mempercepat skenario migrasi umum.

## migrasi dalam skala

Proses memindahkan sebagian besar portofolio aplikasi ke cloud dalam gelombang, dengan lebih banyak aplikasi bergerak pada tingkat yang lebih cepat di setiap gelombang. Fase ini menggunakan praktik terbaik dan pelajaran yang dipetik dari fase sebelumnya untuk mengimplementasikan pabrik migrasi tim, alat, dan proses untuk merampingkan migrasi beban kerja melalui otomatisasi dan pengiriman tangkas. Ini adalah fase ketiga dari [strategi AWS migrasi](#).

## pabrik migrasi

Cross-functional tim yang merampingkan migrasi beban kerja melalui pendekatan otomatis dan gesit. Tim pabrik migrasi biasanya mencakup operasi, analis dan pemilik bisnis, insinyur migrasi, pengembang, dan DevOps profesional yang bekerja di sprint. Antara 20 dan 50 persen portofolio aplikasi perusahaan terdiri dari pola berulang yang dapat dioptimalkan dengan pendekatan pabrik. Untuk informasi selengkapnya, lihat [diskusi tentang pabrik migrasi](#) dan [panduan Pabrik Migrasi Cloud](#) di kumpulan konten ini.

## metadata migrasi

Informasi tentang aplikasi dan server yang diperlukan untuk menyelesaikan migrasi. Setiap pola migrasi memerlukan satu set metadata migrasi yang berbeda. Contoh metadata migrasi termasuk subnet target, grup keamanan, dan akun. AWS

## pola migrasi

Tugas migrasi berulang yang merinci strategi migrasi, tujuan migrasi, dan aplikasi atau layanan migrasi yang digunakan. Contoh: Rehost migrasi ke Amazon EC2 AWS dengan Layanan Migrasi Aplikasi.

## Penilaian Portofolio Migrasi (MPA)

Alat online yang menyediakan informasi untuk memvalidasi kasus bisnis untuk bermigrasi ke. AWS Cloud MPA menyediakan penilaian portofolio terperinci (ukuran kanan server, harga, perbandingan TCO, analisis biaya migrasi) serta perencanaan migrasi (analisis data aplikasi dan pengumpulan data, pengelompokan aplikasi, prioritas migrasi, dan perencanaan gelombang). [Alat MPA](#) (memerlukan login) tersedia gratis untuk semua AWS konsultan dan konsultan APN Partner.

## Penilaian Kesiapan Migrasi (MRA)

Proses mendapatkan wawasan tentang status kesiapan cloud organisasi, mengidentifikasi kekuatan dan kelemahan, dan membangun rencana aksi untuk menutup kesenjangan yang diidentifikasi, menggunakan CAF. AWS Untuk informasi selengkapnya, lihat [panduan kesiapan migrasi](#). MRA adalah tahap pertama dari [strategi AWS migrasi](#).

### strategi migrasi

Pendekatan yang digunakan untuk memigrasikan beban kerja ke. AWS Cloud Untuk informasi lebih lanjut, lihat entri [7 Rs](#) di glosarium ini dan lihat [Memobilisasi organisasi Anda untuk mempercepat](#) migrasi skala besar.

### ML

Lihat [pembelajaran mesin](#).

### modernisasi

Mengubah aplikasi usang (warisan atau monolitik) dan infrastrukturnya menjadi sistem yang gesit, elastis, dan sangat tersedia di cloud untuk mengurangi biaya, mendapatkan efisiensi, dan memanfaatkan inovasi. Untuk informasi selengkapnya, lihat [Strategi untuk memodernisasi aplikasi di. AWS Cloud](#)

### penilaian kesiapan modernisasi

Evaluasi yang membantu menentukan kesiapan modernisasi aplikasi organisasi; mengidentifikasi manfaat, risiko, dan dependensi; dan menentukan seberapa baik organisasi dapat mendukung keadaan masa depan aplikasi tersebut. Hasil penilaian adalah cetak biru arsitektur target, peta jalan yang merinci fase pengembangan dan tonggak untuk proses modernisasi, dan rencana aksi untuk mengatasi kesenjangan yang diidentifikasi. Untuk informasi lebih lanjut, lihat [Mengevaluasi kesiapan modernisasi untuk](#) aplikasi di. AWS Cloud

### aplikasi monolitik (monolit)

Aplikasi yang berjalan sebagai layanan tunggal dengan proses yang digabungkan secara ketat. Aplikasi monolitik memiliki beberapa kelemahan. Jika satu fitur aplikasi mengalami lonjakan permintaan, seluruh arsitektur harus diskalakan. Menambahkan atau meningkatkan fitur aplikasi monolitik juga menjadi lebih kompleks ketika basis kode tumbuh. Untuk mengatasi masalah ini, Anda dapat menggunakan arsitektur microservices. Untuk informasi lebih lanjut, lihat [Menguraikan monolit](#) menjadi layanan mikro.

### MPA

Lihat [Penilaian Portofolio Migrasi](#).

## MQTT

Lihat [Transportasi Telemetri Antrian Pesan](#).

## klasifikasi multiclass

Sebuah proses yang membantu menghasilkan prediksi untuk beberapa kelas (memprediksi satu dari lebih dari dua hasil). Misalnya, model ML mungkin bertanya “Apakah produk ini buku, mobil, atau telepon?” atau “Kategori produk mana yang paling menarik bagi pelanggan ini?”

## infrastruktur yang bisa berubah

Model yang memperbarui dan memodifikasi infrastruktur yang ada untuk beban kerja produksi. Untuk meningkatkan konsistensi, keandalan, dan prediktabilitas, AWS Well-Architected Framework merekomendasikan penggunaan [infrastruktur yang tidak dapat diubah](#) sebagai praktik terbaik.

## O

### OAC

Lihat [kontrol akses asal](#).

### OAI

Lihat [identitas akses asal](#).

### OCM

Lihat [manajemen perubahan organisasi](#).

## migrasi offline

Metode migrasi di mana beban kerja sumber diturunkan selama proses migrasi. Metode ini melibatkan waktu henti yang diperpanjang dan biasanya digunakan untuk beban kerja kecil dan tidak kritis.

### OI

Lihat [integrasi operasi](#).

### OLA

Lihat [perjanjian tingkat operasional](#).

## migrasi online

Metode migrasi di mana beban kerja sumber disalin ke sistem target tanpa diambil offline. Aplikasi yang terhubung ke beban kerja dapat terus berfungsi selama migrasi. Metode ini melibatkan waktu henti nol hingga minimal dan biasanya digunakan untuk beban kerja produksi yang kritis.

## OPC-UA

Lihat [Komunikasi Proses Terbuka - Arsitektur Terpadu](#).

## Komunikasi Proses Terbuka - Arsitektur Terpadu () OPC-UA

Protokol komunikasi mesin-ke-mesin (M2M) untuk otomasi industri. OPC-UA menyediakan standar interoperabilitas dengan enkripsi data, otentikasi, dan skema otorisasi.

## perjanjian tingkat operasional (OLA)

Perjanjian yang menjelaskan apa yang dijanjikan kelompok TI fungsional untuk diberikan satu sama lain, untuk mendukung perjanjian tingkat layanan (SLA).

## Tinjauan Kesiapan Operasional (ORR)

Daftar pertanyaan dan praktik terbaik terkait yang membantu Anda memahami, mengevaluasi, mencegah, atau mengurangi ruang lingkup insiden dan kemungkinan kegagalan. Untuk informasi selengkapnya, lihat [Ulasan Kesiapan Operasional \(ORR\) dalam Kerangka Kerja AWS Well-Architected](#)

## teknologi operasional (OT)

Sistem perangkat keras dan perangkat lunak yang bekerja dengan lingkungan fisik untuk mengendalikan operasi industri, peralatan, dan infrastruktur. Di bidang manufaktur, integrasi sistem OT dan teknologi informasi (TI) adalah fokus utama untuk transformasi [Industri 4.0](#).

## integrasi operasi (OI)

Proses modernisasi operasi di cloud, yang melibatkan perencanaan kesiapan, otomatisasi, dan integrasi. Untuk informasi selengkapnya, lihat [panduan integrasi operasi](#).

## jejak organisasi

Jejak yang dibuat oleh AWS CloudTrail itu mencatat semua peristiwa untuk semua Akun AWS dalam organisasi di AWS Organizations. Jejak ini dibuat di setiap Akun AWS bagian organisasi dan melacak aktivitas di setiap akun. Untuk informasi selengkapnya, lihat [Membuat jejak untuk organisasi](#) dalam CloudTrail dokumentasi.

## manajemen perubahan organisasi (OCM)

Kerangka kerja untuk mengelola transformasi bisnis utama yang mengganggu dari perspektif orang, budaya, dan kepemimpinan. OCM membantu organisasi mempersiapkan, dan transisi ke, sistem dan strategi baru dengan mempercepat adopsi perubahan, mengatasi masalah transisi, dan mendorong perubahan budaya dan organisasi. Dalam strategi AWS migrasi, kerangka kerja ini disebut percepatan orang, karena kecepatan perubahan yang diperlukan dalam proyek adopsi cloud. Untuk informasi lebih lanjut, lihat [panduan OCM](#).

## kontrol akses asal (OAC)

Di CloudFront, opsi yang disempurnakan untuk membatasi akses untuk mengamankan konten Amazon Simple Storage Service (Amazon S3) Anda. OAC mendukung semua bucket S3 di semua Wilayah AWS, enkripsi sisi server dengan AWS KMS (SSE-KMS), dan dinamis PUT dan DELETE permintaan ke bucket S3.

## identitas akses asal (OAI)

Di CloudFront, opsi untuk membatasi akses untuk mengamankan konten Amazon S3 Anda. Saat Anda menggunakan OAI, CloudFront buat prinsipal yang dapat diautentikasi oleh Amazon S3. Prinsipal yang diautentikasi dapat mengakses konten dalam bucket S3 hanya melalui distribusi tertentu. CloudFront Lihat juga [OAC](#), yang menyediakan kontrol akses yang lebih terperinci dan ditingkatkan.

## ORR

Lihat [tinjauan kesiapan operasional](#).

## OT

Lihat [teknologi operasional](#).

## keluar (jalan keluar) VPC

Dalam arsitektur AWS multi-akun, VPC yang menangani koneksi jaringan yang dimulai dari dalam aplikasi. [Arsitektur Referensi AWS Keamanan](#) merekomendasikan pengaturan akun Jaringan Anda dengan VPC masuk, keluar, dan inspeksi untuk melindungi antarmuka dua arah antara aplikasi Anda dan internet yang lebih luas.

# P

## batas izin

Kebijakan manajemen IAM yang dilampirkan pada prinsipal IAM untuk menetapkan izin maksimum yang dapat dimiliki pengguna atau peran. Untuk informasi selengkapnya, lihat [Batas izin](#) dalam dokumentasi IAM.

## Informasi Identifikasi Pribadi (PII)

Informasi yang, jika dilihat secara langsung atau dipasangkan dengan data terkait lainnya, dapat digunakan untuk menyimpulkan identitas individu secara wajar. Contoh PII termasuk nama, alamat, dan informasi kontak.

## PII

Lihat informasi yang [dapat diidentifikasi secara pribadi](#).

## buku pedoman

Serangkaian langkah yang telah ditentukan sebelumnya yang menangkap pekerjaan yang terkait dengan migrasi, seperti mengirimkan fungsi operasi inti di cloud. Buku pedoman dapat berupa skrip, runbook otomatis, atau ringkasan proses atau langkah-langkah yang diperlukan untuk mengoperasikan lingkungan modern Anda.

## PLC

Lihat [pengontrol logika yang dapat diprogram](#).

## PLM

Lihat [manajemen siklus hidup produk](#).

## kebijakan

[Objek yang dapat menentukan izin \(lihat kebijakan berbasis identitas\), menentukan kondisi akses \(lihat kebijakan berbasis sumber daya\), atau menentukan izin maksimum untuk semua akun dalam organisasi di \(lihat kebijakan kontrol layanan\). AWS Organizations](#)

## persistensi poliglot

Secara independen memilih teknologi penyimpanan data microservice berdasarkan pola akses data dan persyaratan lainnya. Jika layanan mikro Anda memiliki teknologi penyimpanan data yang sama, mereka dapat menghadapi tantangan implementasi atau mengalami kinerja yang buruk. Layanan mikro lebih mudah diimplementasikan dan mencapai kinerja dan skalabilitas yang lebih baik jika mereka menggunakan penyimpanan data yang paling sesuai dengan kebutuhan mereka.

## penilaian portofolio

Proses menemukan, menganalisis, dan memprioritaskan portofolio aplikasi untuk merencanakan migrasi. Untuk informasi selengkapnya, lihat [Mengevaluasi kesiapan migrasi](#).

## predikat

Kondisi kueri yang mengembalikan `true` atau `false`, biasanya terletak di `WHERE` klausa.

## predikat pushdown

Teknik pengoptimalan kueri database yang menyaring data dalam kueri sebelum transfer. Ini mengurangi jumlah data yang harus diambil dan diproses dari database relasional, dan meningkatkan kinerja kueri.

## kontrol preventif

Kontrol keamanan yang dirancang untuk mencegah suatu peristiwa terjadi. Kontrol ini adalah garis pertahanan pertama untuk membantu mencegah akses tidak sah atau perubahan yang tidak diinginkan ke jaringan Anda. Untuk informasi selengkapnya, lihat [Kontrol pencegahan dalam Menerapkan kontrol](#) keamanan pada. AWS

## principal

Entitas AWS yang dapat melakukan tindakan dan mengakses sumber daya. Entitas ini biasanya merupakan pengguna root untuk Akun AWS, peran IAM, atau pengguna. Untuk informasi selengkapnya, lihat Prinsip dalam [istilah dan konsep Peran](#) dalam dokumentasi IAM.

## privasi berdasarkan desain

Pendekatan rekayasa sistem yang memperhitungkan privasi melalui seluruh proses pengembangan.

## zona host pribadi

Container yang menyimpan informasi tentang bagaimana Anda ingin Amazon Route 53 merespons kueri DNS untuk domain dan subdomainnya dalam satu atau beberapa VPC. Untuk informasi selengkapnya, lihat [Bekerja dengan zona yang dihosting pribadi](#) di dokumentasi Route 53.

## kontrol proaktif

[Kontrol keamanan](#) yang dirancang untuk mencegah penyebaran sumber daya yang tidak sesuai. Kontrol ini memindai sumber daya sebelum disediakan. Jika sumber daya tidak sesuai dengan kontrol, maka itu tidak disediakan. Untuk informasi selengkapnya, lihat [panduan referensi Kontrol](#)

dalam AWS Control Tower dokumentasi dan lihat [Kontrol proaktif](#) dalam Menerapkan kontrol keamanan pada AWS.

### manajemen siklus hidup produk (PLM)

Manajemen data dan proses untuk suatu produk di seluruh siklus hidupnya, mulai dari desain, pengembangan, dan peluncuran, melalui pertumbuhan dan kematangan, hingga penurunan dan penghapusan.

### lingkungan produksi

Lihat [lingkungan](#).

### pengontrol logika yang dapat diprogram (PLC)

Di bidang manufaktur, komputer yang sangat andal dan mudah beradaptasi yang memantau mesin dan mengotomatiskan proses manufaktur.

### rantai cepat

Menggunakan output dari satu prompt [LLM](#) sebagai input untuk prompt berikutnya untuk menghasilkan respons yang lebih baik. Teknik ini digunakan untuk memecah tugas yang kompleks menjadi subtugas, atau untuk secara iteratif memperbaiki atau memperluas respons awal. Ini membantu meningkatkan akurasi dan relevansi respons model dan memungkinkan hasil yang lebih terperinci dan dipersonalisasi.

### pseudonimisasi

Proses penggantian pengenal pribadi dalam kumpulan data dengan nilai placeholder. Pseudonimisasi dapat membantu melindungi privasi pribadi. Data pseudonim masih dianggap sebagai data pribadi.

### publish/subscribe (pub/sub)

Pola yang memungkinkan komunikasi asinkron antara layanan mikro untuk meningkatkan skalabilitas dan daya tanggap. Misalnya, dalam [MES](#) berbasis layanan mikro, layanan mikro dapat mempublikasikan pesan peristiwa ke saluran yang dapat berlangganan layanan mikro lainnya. Sistem dapat menambahkan layanan mikro baru tanpa mengubah layanan penerbitan.

## Q

### rencana kueri

Serangkaian langkah, seperti instruksi, yang digunakan untuk mengakses data dalam sistem database relasional SQL.

### regresi rencana kueri

Ketika pengoptimal layanan database memilih rencana yang kurang optimal daripada sebelum perubahan yang diberikan ke lingkungan database. Hal ini dapat disebabkan oleh perubahan statistik, kendala, pengaturan lingkungan, pengikatan parameter kueri, dan pembaruan ke mesin database.

## R

### Matriks RACI

Lihat [bertanggung jawab, akuntabel, dikonsultasikan, diinformasikan \(RACI\)](#).

### LAP

Lihat [Retrieval Augmented Generation](#).

### ransomware

Perangkat lunak berbahaya yang dirancang untuk memblokir akses ke sistem komputer atau data sampai pembayaran dilakukan.

### Matriks RASCI

Lihat [bertanggung jawab, akuntabel, dikonsultasikan, diinformasikan \(RACI\)](#).

### RCAC

Lihat [kontrol akses baris dan kolom](#).

### replika baca

Salinan database yang digunakan untuk tujuan read-only. Anda dapat merutekan kueri ke replika baca untuk mengurangi beban pada database utama Anda.

### arsitek ulang

Lihat [7 Rs](#).

## tujuan titik pemulihan (RPO)

Jumlah waktu maksimum yang dapat diterima sejak titik pemulihan data terakhir. Ini menentukan apa yang dianggap sebagai kehilangan data yang dapat diterima antara titik pemulihan terakhir dan gangguan layanan.

## tujuan waktu pemulihan (RTO)

Penundaan maksimum yang dapat diterima antara gangguan layanan dan pemulihan layanan.

## refactor

Lihat [7 Rs](#).

## Region

Kumpulan AWS sumber daya di wilayah geografis. Masing-masing Wilayah AWS terisolasi dan independen dari yang lain untuk memberikan toleransi kesalahan, stabilitas, dan ketahanan. Untuk informasi selengkapnya, lihat [Menentukan Wilayah AWS akun yang dapat digunakan](#).

## regresi

Teknik ML yang memprediksi nilai numerik. Misalnya, untuk memecahkan masalah “Berapa harga rumah ini akan dijual?” Model ML dapat menggunakan model regresi linier untuk memprediksi harga jual rumah berdasarkan fakta yang diketahui tentang rumah (misalnya, luas persegi).

## rehost

Lihat [7 Rs](#).

## melepaskan

Dalam proses penyebaran, tindakan mempromosikan perubahan pada lingkungan produksi.

## memindahkan

Lihat [7 Rs](#).

## memplatform ulang

Lihat [7 Rs](#).

## pembelian kembali

Lihat [7 Rs](#).

## ketahanan

Kemampuan aplikasi untuk melawan atau pulih dari gangguan. [Ketersediaan tinggi](#) dan [pemulihan bencana](#) adalah pertimbangan umum ketika merencanakan ketahanan di AWS Cloud. Untuk informasi lebih lanjut, lihat [AWS Cloud Ketahanan](#).

## kebijakan berbasis sumber daya

Kebijakan yang dilampirkan ke sumber daya, seperti bucket Amazon S3, titik akhir, atau kunci enkripsi. Jenis kebijakan ini menentukan prinsipal mana yang diizinkan mengakses, tindakan yang didukung, dan kondisi lain yang harus dipenuhi.

## matriks yang bertanggung jawab, akuntabel, dikonsultasikan, diinformasikan (RACI)

Matriks yang mendefinisikan peran dan tanggung jawab untuk semua pihak yang terlibat dalam kegiatan migrasi dan operasi cloud. Nama matriks berasal dari jenis tanggung jawab yang didefinisikan dalam matriks: bertanggung jawab (R), akuntabel (A), dikonsultasikan (C), dan diinformasikan (I). Jenis dukungan (S) adalah opsional. Jika Anda menyertakan dukungan, matriks disebut matriks RASCI, dan jika Anda mengecualikannya, itu disebut matriks RACI.

## kontrol responsif

Kontrol keamanan yang dirancang untuk mendorong remediasi efek samping atau penyimpangan dari garis dasar keamanan Anda. Untuk informasi selengkapnya, lihat [Kontrol responsif](#) dalam Menerapkan kontrol keamanan pada AWS.

## melestarikan

Lihat [7 Rs](#).

## pensiun

Lihat [7 Rs](#).

## Retrieval Augmented Generation (RAG)

Teknologi [AI generatif](#) di mana [LLM](#) mereferensikan sumber data otoritatif yang berada di luar sumber data pelatihannya sebelum menghasilkan respons. Misalnya, model RAG mungkin melakukan pencarian semantik dari basis pengetahuan organisasi atau data kustom. Untuk informasi lebih lanjut, lihat [Apa itu RAG](#).

## rotasi

Proses memperbarui [rahasia](#) secara berkala untuk membuatnya lebih sulit bagi penyerang untuk mengakses kredensial.

## kontrol akses baris dan kolom (RCAC)

Penggunaan ekspresi SQL dasar dan fleksibel yang telah menetapkan aturan akses. RCAC terdiri dari izin baris dan topeng kolom.

## RPO

Lihat [tujuan titik pemulihan](#).

## RTO

Lihat [tujuan waktu pemulihan](#).

## buku runbook

Satu set prosedur manual atau otomatis yang diperlukan untuk melakukan tugas tertentu. Ini biasanya dibangun untuk merampingkan operasi berulang atau prosedur dengan tingkat kesalahan yang tinggi.

## D

### SAML 2.0

Standar terbuka yang digunakan oleh banyak penyedia identitas (IdPs). Fitur ini memungkinkan sistem masuk tunggal gabungan (SSO), sehingga pengguna dapat masuk ke Konsol Manajemen AWS atau memanggil operasi AWS API tanpa Anda harus membuat pengguna di IAM untuk semua orang di organisasi Anda. Untuk informasi lebih lanjut tentang federasi berbasis SAMP 2.0, lihat [Tentang federasi berbasis SAMP 2.0](#) dalam dokumentasi IAM.

### SCADA

Lihat [kontrol pengawasan dan akuisisi data](#).

### SCP

Lihat [kebijakan kontrol layanan](#).

### Rahasia

Dalam AWS Secrets Manager, informasi rahasia atau terbatas, seperti kata sandi atau kredensial pengguna, yang Anda simpan dalam bentuk terenkripsi. Ini terdiri dari nilai rahasia dan metadatanya. Nilai rahasia dapat berupa biner, string tunggal, atau beberapa string. Untuk informasi selengkapnya, lihat [Apa yang ada di rahasia Secrets Manager?](#) dalam dokumentasi Secrets Manager.

## keamanan dengan desain

Pendekatan rekayasa sistem yang memperhitungkan keamanan melalui seluruh proses pengembangan.

## kontrol keamanan

Pagar pembatas teknis atau administratif yang mencegah, mendeteksi, atau mengurangi kemampuan pelaku ancaman untuk mengeksploitasi kerentanan keamanan. [Ada empat jenis kontrol keamanan utama: preventif, detektif, responsif, dan proaktif.](#)

## pengerasan keamanan

Proses mengurangi permukaan serangan untuk membuatnya lebih tahan terhadap serangan. Ini dapat mencakup tindakan seperti menghapus sumber daya yang tidak lagi diperlukan, menerapkan praktik keamanan terbaik untuk memberikan hak istimewa paling sedikit, atau menonaktifkan fitur yang tidak perlu dalam file konfigurasi.

## sistem informasi keamanan dan manajemen acara (SIEM)

Alat dan layanan yang menggabungkan sistem manajemen informasi keamanan (SIM) dan manajemen acara keamanan (SEM). Sistem SIEM mengumpulkan, memantau, dan menganalisis data dari server, jaringan, perangkat, dan sumber lain untuk mendeteksi ancaman dan pelanggaran keamanan, dan untuk menghasilkan peringatan.

## otomatisasi respons keamanan

Tindakan yang telah ditentukan dan diprogram yang dirancang untuk secara otomatis merespons atau memulihkan peristiwa keamanan. Otomatisasi ini berfungsi sebagai kontrol keamanan [detektif](#) atau [responsif](#) yang membantu Anda menerapkan praktik terbaik AWS keamanan. Contoh tindakan respons otomatis termasuk memodifikasi grup keamanan VPC, menambal instans Amazon EC2, atau memutar kredensial.

## enkripsi sisi server

Enkripsi data di tujuannya, oleh Layanan AWS yang menerimanya.

## kebijakan kontrol layanan (SCP)

Kebijakan yang menyediakan kontrol terpusat atas izin untuk semua akun di organisasi. AWS Organizations SCP menentukan pagar pembatas atau menetapkan batasan pada tindakan yang dapat didelegasikan oleh administrator kepada pengguna atau peran. Anda dapat menggunakan SCP sebagai daftar izin atau daftar penolakan, untuk menentukan layanan atau tindakan mana

yang diizinkan atau dilarang. Untuk informasi selengkapnya, lihat [Kebijakan kontrol layanan](#) dalam AWS Organizations dokumentasi.

#### titik akhir layanan

URL titik masuk untuk file Layanan AWS. Anda dapat menggunakan endpoint untuk terhubung secara terprogram ke layanan target. Untuk informasi selengkapnya, lihat [Layanan AWS titik akhir](#) di Referensi Umum AWS.

#### perjanjian tingkat layanan (SLA)

Perjanjian yang menjelaskan apa yang dijanjikan tim TI untuk diberikan kepada pelanggan mereka, seperti waktu kerja dan kinerja layanan.

#### indikator tingkat layanan (SLI)

Pengukuran aspek kinerja layanan, seperti tingkat kesalahan, ketersediaan, atau throughputnya.

#### tujuan tingkat layanan (SLO)

Metrik target yang mewakili kesehatan layanan, yang diukur dengan indikator [tingkat layanan](#).

#### model tanggung jawab bersama

Model yang menjelaskan tanggung jawab yang Anda bagikan AWS untuk keamanan dan kepatuhan cloud. AWS bertanggung jawab atas keamanan cloud, sedangkan Anda bertanggung jawab atas keamanan di cloud. Untuk informasi selengkapnya, lihat [Model tanggung jawab bersama](#).

#### Bayangan AI

Aplikasi [AI](#) yang tidak sah dibuat atau digunakan di luar saluran yang diatur dalam suatu organisasi.

#### SIEM

Lihat [informasi keamanan dan sistem manajemen acara](#).

#### titik kegagalan tunggal (SPOF)

Kegagalan dalam satu komponen penting dari aplikasi yang dapat mengganggu sistem.

#### SLA

Lihat [perjanjian tingkat layanan](#).

#### SLI

Lihat [indikator tingkat layanan](#).

## SLO

Lihat [tujuan tingkat layanan](#).

### model split-and-lead

Pola untuk menskalakan dan mempercepat proyek modernisasi. Ketika fitur baru dan rilis produk didefinisikan, tim inti berpisah untuk membuat tim produk baru. Ini membantu meningkatkan kemampuan dan layanan organisasi Anda, meningkatkan produktivitas pengembang, dan mendukung inovasi yang cepat. Untuk informasi lebih lanjut, lihat [Pendekatan bertahap untuk memodernisasi aplikasi](#) di AWS Cloud

## SPOF

Lihat [satu titik kegagalan](#).

### skema bintang

Struktur organisasi database yang menggunakan satu tabel fakta besar untuk menyimpan data transaksional atau terukur dan menggunakan satu atau lebih tabel dimensi yang lebih kecil untuk menyimpan atribut data. Struktur ini dirancang untuk digunakan dalam [gudang data](#) atau untuk tujuan intelijen bisnis.

### pola ara pencekik

Pendekatan untuk memodernisasi sistem monolitik dengan menulis ulang secara bertahap dan mengganti fungsionalitas sistem sampai sistem warisan dapat dinonaktifkan. Pola ini menggunakan analogi pohon ara yang tumbuh menjadi pohon yang sudah mapan dan akhirnya mengatasi dan menggantikan inangnya. Pola ini [diperkenalkan oleh Martin Fowler](#) sebagai cara untuk mengelola risiko saat menulis ulang sistem monolitik. Untuk contoh cara menerapkan pola ini, lihat [Memodernisasi layanan web ASP.NET Microsoft \(ASMX\) lama secara bertahap menggunakan container dan Amazon API Gateway](#).

## subnet

Rentang alamat IP dalam VPC Anda. Subnet harus berada di Availability Zone tunggal.

## kontrol pengawasan dan akuisisi data (SCADA)

Di bidang manufaktur, sistem yang menggunakan perangkat keras dan perangkat lunak untuk memantau aset fisik dan operasi produksi.

## enkripsi simetris

Algoritma enkripsi yang menggunakan kunci yang sama untuk mengenkripsi dan mendekripsi data.

## pengujian sintetis

Menguji sistem dengan cara yang mensimulasikan interaksi pengguna untuk mendeteksi potensi masalah atau untuk memantau kinerja. Anda dapat menggunakan [Amazon CloudWatch Synthetics](#) untuk membuat tes ini.

## sistem prompt

Teknik untuk memberikan konteks, instruksi, atau pedoman ke [LLM](#) untuk mengarahkan perilakunya. Permintaan sistem membantu mengatur konteks dan menetapkan aturan untuk interaksi dengan pengguna.

# T

## tag

Key-value pasangan yang bertindak sebagai metadata untuk mengatur sumber daya Anda AWS . Tag membantu Anda mengelola, mengidentifikasi, mengatur, dan memfilter sumber daya. Untuk informasi selengkapnya, lihat [Menandai sumber daya AWS](#).

## variabel target

Nilai yang Anda coba prediksi dalam ML yang diawasi. Ini juga disebut sebagai variabel hasil. Misalnya, dalam pengaturan manufaktur, variabel target bisa menjadi cacat produk.

## daftar tugas

Alat yang digunakan untuk melacak kemajuan melalui runbook. Daftar tugas berisi ikhtisar runbook dan daftar tugas umum yang harus diselesaikan. Untuk setiap tugas umum, itu termasuk perkiraan jumlah waktu yang dibutuhkan, pemilik, dan kemajuan.

## lingkungan uji

Lihat [lingkungan](#).

## pelatihan

Untuk menyediakan data bagi model ML Anda untuk dipelajari. Data pelatihan harus berisi jawaban yang benar. Algoritma pembelajaran menemukan pola dalam data pelatihan yang memetakan atribut data input ke target (jawaban yang ingin Anda prediksi). Ini menghasilkan model ML yang menangkap pola-pola ini. Anda kemudian dapat menggunakan model ML untuk membuat prediksi pada data baru yang Anda tidak tahu targetnya.

## alat

Fungsi atau API yang dapat [dipanggil agen](#) untuk melakukan operasi di sistem eksternal.

## gerbang transit

Hub transit jaringan yang dapat Anda gunakan untuk menghubungkan VPC dan jaringan lokal Anda. Untuk informasi selengkapnya, lihat [Apa itu gateway transit](#) dalam AWS Transit Gateway dokumentasi.

## alur kerja berbasis batang

Pendekatan di mana pengembang membangun dan menguji fitur secara lokal di cabang fitur dan kemudian menggabungkan perubahan tersebut ke cabang utama. Cabang utama kemudian dibangun untuk pengembangan, praproduksi, dan lingkungan produksi, secara berurutan.

## akses tepercaya

Memberikan izin ke layanan yang Anda tentukan untuk melakukan tugas di organisasi Anda di dalam AWS Organizations dan di akunnya atas nama Anda. Layanan tepercaya menciptakan peran terkait layanan di setiap akun, ketika peran itu diperlukan, untuk melakukan tugas manajemen untuk Anda. Untuk informasi selengkapnya, lihat [Menggunakan AWS Organizations dengan AWS layanan lain](#) dalam AWS Organizations dokumentasi.

## penyetelan

Untuk mengubah aspek proses pelatihan Anda untuk meningkatkan akurasi model ML. Misalnya, Anda dapat melatih model ML dengan membuat set pelabelan, menambahkan label, dan kemudian mengulangi langkah-langkah ini beberapa kali di bawah pengaturan yang berbeda untuk mengoptimalkan model.

## tim dua pizza

Sebuah DevOps tim kecil yang bisa Anda beri makan dengan dua pizza. Ukuran tim dua pizza memastikan peluang terbaik untuk berkolaborasi dalam pengembangan perangkat lunak.

## U

### waswas

Sebuah konsep yang mengacu pada informasi yang tidak tepat, tidak lengkap, atau tidak diketahui yang dapat merusak keandalan model ML prediktif. Ada dua jenis ketidakpastian:

ketidakpastian epistemik disebabkan oleh data yang terbatas dan tidak lengkap, sedangkan ketidakpastian aleatorik disebabkan oleh kebisingan dan keacakan yang melekat dalam data.

tugas yang tidak terdiferensiasi

Juga dikenal sebagai angkat berat, pekerjaan yang diperlukan untuk membuat dan mengoperasikan aplikasi tetapi itu tidak memberikan nilai langsung kepada pengguna akhir atau memberikan keunggulan kompetitif. Contoh tugas yang tidak terdiferensiasi termasuk pengadaan, pemeliharaan, dan perencanaan kapasitas.

lingkungan atas

Lihat [lingkungan](#).

## V

menyedot debu

Operasi pemeliharaan database yang melibatkan pembersihan setelah pembaruan tambahan untuk merebut kembali penyimpanan dan meningkatkan kinerja.

kendali versi

Proses dan alat yang melacak perubahan, seperti perubahan kode sumber dalam repositori.

Peering VPC

Koneksi antara dua VPC yang memungkinkan Anda merutekan lalu lintas dengan menggunakan alamat IP pribadi. Untuk informasi selengkapnya, lihat [Apa itu peering VPC](#) di dokumentasi VPC Amazon.

kerentanan

Kelemahan perangkat lunak atau perangkat keras yang membahayakan keamanan sistem.

## W

cache hangat

Cache buffer yang berisi data terkini dan relevan yang sering diakses. Instance database dapat membaca dari cache buffer, yang lebih cepat daripada membaca dari memori utama atau disk.

## data hangat

Data yang jarang diakses. Saat menanyakan jenis data ini, kueri yang cukup lambat biasanya dapat diterima.

## fungsi jendela

Fungsi SQL yang melakukan perhitungan pada sekelompok baris yang berhubungan dengan catatan saat ini. Fungsi jendela berguna untuk memproses tugas, seperti menghitung rata-rata bergerak atau mengakses nilai baris berdasarkan posisi relatif dari baris saat ini.

## beban kerja

Kumpulan sumber daya dan kode yang memberikan nilai bisnis, seperti aplikasi yang dihadapi pelanggan atau proses backend.

## aliran kerja

Grup fungsional dalam proyek migrasi yang bertanggung jawab atas serangkaian tugas tertentu. Setiap alur kerja independen tetapi mendukung alur kerja lain dalam proyek. Misalnya, alur kerja portofolio bertanggung jawab untuk memprioritaskan aplikasi, perencanaan gelombang, dan mengumpulkan metadata migrasi. Alur kerja portofolio mengirimkan aset ini ke alur kerja migrasi, yang kemudian memigrasikan server dan aplikasi.

## CACING

Lihat [menulis sekali, baca banyak](#).

## WQF

Lihat [AWS Kerangka Kualifikasi Beban Kerja](#).

## tulis sekali, baca banyak (WORM)

Model penyimpanan yang menulis data satu kali dan mencegah data dihapus atau dimodifikasi. Pengguna yang berwenang dapat membaca data sebanyak yang diperlukan, tetapi mereka tidak dapat mengubahnya. Infrastruktur penyimpanan data ini dianggap [tidak dapat diubah](#).

## Z

### eksploitasi zero-day

Serangan, biasanya malware, yang memanfaatkan kerentanan [zero-day](#).

## kerentanan zero-day

Cacat atau kerentanan yang tak tanggung-tanggung dalam sistem produksi. Aktor ancaman dapat menggunakan jenis kerentanan ini untuk menyerang sistem. Pengembang sering menyadari kerentanan sebagai akibat dari serangan tersebut.

## bisikan zero-shot

Memberikan [LLM](#) dengan instruksi untuk melakukan tugas tetapi tidak ada contoh (tembak) yang dapat membantu membimbingnya. LLM harus menggunakan pengetahuan pra-terlatih untuk menangani tugas. Efektivitas bidikan nol tergantung pada kompleksitas tugas dan kualitas prompt. Lihat juga beberapa [bidikan yang diminta](#).

## aplikasi zombie

Aplikasi yang memiliki CPU rata-rata dan penggunaan memori di bawah 5 persen. Dalam proyek migrasi, adalah umum untuk menghentikan aplikasi ini.

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