



Migrasi database Oracle ke Cloud AWS

# AWS Bimbingan Preskriptif



# AWS Bimbingan Preskriptif: Migrasi database Oracle ke Cloud AWS

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
Ikhtisar .....	1
Strategi migrasi database Oracle .....	3
Memilih Strategi Migrasi yang Tepat .....	4
Migrasi online dan offline .....	5
Migrasi basis data homogen .....	6
Amazon RDS for Oracle .....	7
Kapan memilih Amazon RDS .....	7
Ketersediaan tinggi .....	8
Replika baca .....	9
Menggunakan replika baca di Wilayah lain AWS .....	10
Amazon RDS Custom for Oracle .....	11
Kapan memilih Amazon RDS Custom untuk Oracle .....	12
Cara kerjanya .....	12
Amazon EC2 untuk Oracle .....	15
Kapan memilih Amazon EC2 .....	15
Ketersediaan tinggi .....	16
VMware Cloud on AWS untuk Oracle .....	18
Kapan memilih VMware Cloud di AWS .....	19
Alat .....	19
Pengembang Oracle SQL .....	21
Oracle SQL*Loader .....	22
Ekspor dan Impor Oracle .....	22
Oracle Data Pump .....	22
AWS DMS .....	23
Oracle GoldenGate .....	25
Oracle Data Guard .....	26
Oracle RMAN .....	27
VMware HCX .....	28
Opsi lisensi .....	28
Termasuk Lisensi .....	29
BYOL .....	29
Migrasi database heterogen .....	31
Alat untuk migrasi database heterogen .....	32

AWS SCT .....	33
AWS DMS .....	33
Praktik terbaik untuk bermigrasi ke Amazon RDS for Oracle .....	34
Menyediakan basis data target Anda .....	34
Mengekspor data dari database sumber Anda .....	34
Mentransfer file dump data ke AWS .....	35
Mengimpor data ke database target Anda .....	35
Langkah pasca-impor .....	36
Menguji migrasi .....	36
Mengoperasikan dan mengoptimalkan basis data Amazon RDS Anda .....	37
AWS Mitra .....	38
Sumber daya tambahan .....	39
Lampiran: Kuesioner migrasi Oracle .....	41
Informasi umum .....	41
Infrastruktur .....	42
Pencadangan basis data .....	42
Keamanan basis data .....	42
Database ketersediaan tinggi dan pemulihan bencana .....	43
Riwayat dokumen .....	44
Glosarium .....	46
# .....	46
A .....	47
B .....	50
C .....	52
D .....	55
E .....	59
F .....	61
G .....	63
H .....	64
I .....	65
L .....	68
M .....	69
O .....	73
P .....	76
Q .....	79
R .....	79

---

D .....	82
T .....	86
U .....	87
V .....	88
W .....	88
Z .....	89
.....	xcii

# Migrasi database Oracle ke Cloud AWS

Sagar Patel, Amazon Web Services (AWS)

Agustus 2024 ([sejarah dokumen](#))

Amazon Web Services (AWS) menyediakan seperangkat layanan dan alat yang komprehensif untuk menyebarkan Oracle Database pada infrastruktur yang andal dan aman AWS Cloud . Panduan ini menjelaskan opsi yang tersedia untuk memigrasikan database lokal Oracle Anda ke database lokal. AWS Cloud Ini juga menyelami praktik dan skenario terbaik untuk menggunakan opsi migrasi ini.

Panduan ini ditujukan untuk manajer program atau proyek, pemilik produk, administrator basis data, insinyur basis data, dan manajer operasi atau infrastruktur yang berencana untuk memigrasikan database Oracle lokal mereka. AWS

## Ikhtisar

Sebelum Anda memigrasikan database Oracle AWS, Anda harus memahami dan mengevaluasi strategi migrasi Anda dengan menggunakan kerangka kerja yang dibahas dalam Strategi [Migrasi untuk Database Relasional](#).

Langkah pertama adalah melakukan analisis aplikasi Anda dan beban kerja Oracle Database untuk memahami kompleksitas, kompatibilitas, dan biaya migrasi. Berikut adalah beberapa poin teratas yang harus Anda pertimbangkan ketika Anda berencana untuk bermigrasi:

- Periksa ukuran database saat ini dan pertumbuhan kapasitas keseluruhan. Misalnya, jika Anda berencana untuk memigrasikan database Oracle ke Amazon Relational Database Service (Amazon RDS) atau Amazon RDS Custom, Anda dapat membuat instans DB dengan penyimpanan hingga 64 TiB. Untuk informasi terbaru, lihat [Penyimpanan instans Amazon RDS DB](#) di dokumentasi Amazon RDS.
- Tinjau laporan Oracle Automatic Workload Repository (AWR) untuk memeriksa penggunaan sumber daya dan kesehatan database database lokal Anda.
- Periksa dependensi database saat ini pada database lain. Jika database Anda bergantung pada database lain, Anda dapat memigrasikan mereka bersama-sama atau membuat dependensi setelah Anda memigrasikan database utama Anda.
- Periksa dependensi aplikasi. Jika database Anda mendukung aplikasi lama, kustom, atau paket, Amazon RDS Custom for Oracle mungkin merupakan pilihan yang baik. Layanan ini

memungkinkan Anda mempertahankan kontrol atas konfigurasi database, sistem file bersama, dan tambalan sistem operasi.

- Tentukan IOPS dan throughput database Anda. Jika Anda berencana untuk bermigrasi ke Amazon RDS, pertimbangkan kinerja [I/O instans Amazon RDS DB](#).
- Tinjau arsitektur dan kebutuhan audit atau kepatuhan Anda saat ini, untuk memastikan Anda dapat memenuhi persyaratan ini setelah pindah ke Amazon RDS atau Amazon Elastic Compute Cloud (Amazon). EC2
- [Periksa versi dan edisi perangkat lunak Oracle Database Anda untuk memastikannya didukung jika Anda berencana untuk pindah ke Amazon RDS for Oracle \(lihat versi yang saat ini didukung untuk Amazon RDS dan Amazon RDS Custom\).](#)
- Periksa konektivitas jaringan antara lingkungan lokal Anda dan AWS, untuk memastikan bahwa itu menyediakan bandwidth yang cukup untuk transfer data yang cepat antara di tempat dan AWS.
- Tentukan jumlah waktu henti yang tersedia untuk migrasi sehingga Anda dapat merencanakan pendekatan migrasi dan memutuskan apakah Anda ingin menggunakan migrasi online atau offline.
- Identifikasi persyaratan tujuan waktu pemulihan (RTO), tujuan titik pemulihan (RPO), dan perjanjian tingkat layanan (SLA) Anda untuk beban kerja database Anda yang ada.
- Periksa platform endian chipset dari beban kerja database. AWS mendukung platform endian kecil x86-x64. Platform lain, seperti Sun SPARC, HP Tru64, atau platform big-endian berbasis IBM Zseries, memerlukan migrasi lintas platform.
- AWS mendukung Linux (32-bit dan 64-bit) dan sistem operasi Windows. Ini tidak mendukung sistem operasi Solaris, HP-UX, atau IBM AIX, yang biasanya digunakan untuk database Oracle. Migrasi database Oracle dari sistem operasi ini memerlukan konversi platform.

# Strategi migrasi database Oracle

Pada tingkat tinggi, ada dua opsi untuk memigrasikan database Oracle dari tempat ke lokasi AWS Cloud: tetap di Oracle (migrasi homogen) atau pindah dari Oracle (migrasi heterogen). Dalam migrasi homogen, Anda tidak mengubah mesin database (yaitu, database target Anda juga merupakan database Oracle). Dalam migrasi heterogen, Anda beralih ke mesin database sumber terbuka seperti MySQL, PostgreSQL, atau MariaDB, atau ke database Cloud-native seperti Amazon Aurora, Amazon DynamoDB, atau Amazon. AWS RedShift

Ada tiga strategi umum untuk memigrasikan database Oracle Anda ke AWS: rehost, replatform, dan re-architect (refactor). Ini adalah bagian dari [7 Rs strategi migrasi aplikasi](#) dan dijelaskan dalam tabel berikut.

Strategi	Jenis	Kapan harus memilih	Contoh
Rehost	Homogen	Anda ingin memigrasikan database Oracle Anda apa adanya, dengan atau tanpa mengubah sistem operasi, perangkat lunak database, atau konfigurasi.	Database Oracle ke Amazon EC2
Platform Ulang	Homogen	Anda ingin mengurangi waktu yang Anda habiskan untuk mengelola instance database dengan menggunakan penawaran database-as-a-service (DBaaS).	Database Oracle ke Amazon RDS for Oracle
Arsitek ulang (refactor)	Heterogen	Anda ingin merestrukturisasi, menulis ulang, dan merancang ulang database dan	Database Oracle ke Amazon Aurora PostgreSQL, MySQL, atau MariaDB

aplikasi Anda untuk memanfaatkan fitur database open-source dan cloud-native.

## Memilih Strategi Migrasi yang Tepat

Memilih strategi yang tepat tergantung pada kebutuhan bisnis Anda, kendala sumber daya Anda, jangka waktu migrasi Anda, dan pertimbangan biaya. Diagram berikut menunjukkan upaya dan kompleksitas yang terlibat dalam migrasi, termasuk enam strategi.

Strategy	Effort (time and cost)	Opportunity to optimize
Retire	N/A	N/A
Retain		N/A
Rehost		
Repurchase		
Replatform		
Refactor (re-architect)		


 Increasing complexity

Memfaktorkan ulang database Oracle Anda dan bermigrasi ke database open-source atau AWS cloud-native seperti Amazon Aurora PostgreSQL Compatible Edition atau Amazon Aurora MySQL Compatible Edition dapat membantu Anda memodernisasi dan mengoptimalkan database Anda. Dengan pindah ke database sumber terbuka, Anda dapat menghindari lisensi mahal (menghasilkan biaya lebih rendah), periode penguncian vendor, dan audit, dan Anda tidak perlu membayar biaya tambahan untuk fitur baru. Namun, tergantung pada kompleksitas beban kerja Anda, refactoring database Oracle Anda bisa menjadi upaya yang rumit, memakan waktu, dan intensif sumber daya.

Untuk mengurangi kompleksitas, alih-alih memigrasikan database Anda dalam satu langkah, Anda dapat mempertimbangkan pendekatan bertahap. Pada tahap pertama, Anda dapat fokus pada fungsionalitas basis data inti. Pada tahap berikutnya, Anda dapat mengintegrasikan AWS layanan tambahan ke lingkungan cloud Anda, untuk mengurangi biaya, dan untuk mengoptimalkan kinerja, produktivitas, dan kepatuhan. Misalnya, jika tujuan Anda adalah mengganti database Oracle lokal Anda dengan Aurora PostgreSQL yang kompatibel, Anda dapat mempertimbangkan untuk menghosting ulang database Anda di Amazon atau memplatform ulang database Anda di Amazon

RDS for Oracle pada fase EC2 pertama, lalu refactor ke Aurora PostgreSQL yang kompatibel dengan Aurora pada fase berikutnya. Pendekatan ini membantu mengurangi biaya, sumber daya, dan risiko selama fase migrasi dan berfokus pada optimalisasi dan modernisasi di fase kedua.

## Migrasi online dan offline

Anda dapat menggunakan dua metode untuk memigrasikan Oracle Database dari lingkungan lokal ke AWS Cloud, berdasarkan timeline migrasi dan berapa banyak waktu henti yang dapat Anda izinkan: migrasi online atau migrasi offline.

- **Migrasi offline:** Metode ini digunakan ketika aplikasi Anda mampu membayar downtime yang direncanakan. Dalam migrasi offline, database sumber offline selama periode migrasi. Sementara database sumber offline, itu dimigrasikan ke database target pada AWS. Setelah migrasi selesai, pemeriksaan validasi dan verifikasi dilakukan untuk memastikan konsistensi data dengan database sumber. Ketika database melewati semua pemeriksaan validasi, Anda melakukan cutover AWS dengan menghubungkan aplikasi Anda ke database target pada AWS.
- **Migrasi online:** Metode ini digunakan saat aplikasi Anda membutuhkan waktu henti mendekati nol hingga minimal. Dalam migrasi online, database sumber dimigrasikan dalam beberapa langkah ke AWS. Pada langkah awal, data dalam database sumber disalin ke database target saat database sumber masih berjalan. Pada langkah selanjutnya, semua perubahan dari database sumber disebarkan ke database target. Ketika database sumber dan target disinkronkan, mereka siap untuk dipotong. Selama cutover, aplikasi mengalihkan koneksinya ke database target AWS, tanpa meninggalkan koneksi ke database sumber. Anda dapat menggunakan AWS Database Migration Service (AWS DMS), Oracle, Quest GoldenGate SharePlex, atau alat yang tersedia dari [AWS Marketplace](#) (seperti Attunity) untuk menyinkronkan basis data sumber dan target.

# Migrasi database homogen untuk database Oracle

AWS menawarkan Anda kemampuan untuk menjalankan Oracle Database di lingkungan cloud. Untuk pengembang dan administrator database, menjalankan Oracle Database di AWS Cloud sangat mirip dengan menjalankan Oracle Database di pusat data. Bagian ini menjelaskan opsi untuk memigrasikan Oracle Database dari lingkungan lokal atau pusat data ke AWS Cloud

AWS menawarkan empat opsi untuk menjalankan Oracle Database AWS, seperti yang dijelaskan dalam tabel berikut.

Opsi	Sorotan	Informasi lebih lanjut
Database Oracle di Amazon RDS	Layanan terkelola, menyediakan an penyediaan dan perizinan yang mudah	<a href="#">Amazon RDS for Oracle bagian</a>
Database Oracle di Amazon RDS Kustom	Layanan terkelola, tetapi Anda mempertahankan hak administratif atas database dan sistem operasi yang mendasarinya	<a href="#">Amazon RDS Kustom untuk bagian Oracle</a>
Database Oracle di Amazon EC2	Dikelola sendiri, memberikan kontrol dan fleksibilitas penuh	<a href="#">Amazon EC2 untuk bagian Oracle</a>
Database Oracle di VMware Cloud di AWS	Gangguan minimal, mudah dikelola	<a href="#">VMware Cloud aktif AWS untuk bagian Oracle</a>

## Pemberitahuan

Per 30 April 2024, VMware Cloud on AWS tidak lagi dijual kembali oleh AWS atau mitra salurannya. Layanan ini akan terus tersedia melalui Broadcom. Kami mendorong Anda untuk menghubungi AWS perwakilan Anda untuk detailnya.

Persyaratan aplikasi Anda, fitur database, fungsionalitas, kapasitas pertumbuhan, dan kompleksitas arsitektur secara keseluruhan akan menentukan opsi mana yang harus dipilih. Jika Anda

memigrasikan beberapa database Oracle ke AWS, beberapa di antaranya mungkin sangat cocok untuk Amazon RDS sedangkan yang lain mungkin lebih cocok untuk dijalankan langsung di Amazon EC2. Anda mungkin memiliki database yang berjalan di Oracle Enterprise Edition (EE) tetapi cocok untuk Oracle Standard Edition Two (SE2). SE2 Anda dapat menghemat biaya dan lisensi untuk database tersebut. Banyak AWS pelanggan menjalankan beberapa beban kerja Oracle Database di Amazon RDS, Amazon EC2, dan Cloud on. VMware AWS Jika Anda pindah ke Amazon RDS Custom, pastikan untuk meninjau [persyaratan dan batasan untuk Amazon RDS Custom for Oracle](#).

## Amazon RDS for Oracle

Amazon RDS for Oracle adalah layanan database terkelola yang menyederhanakan penyediaan dan pengelolaan Oracle Database. AWS Amazon RDS memudahkan pengaturan, pengoperasian, dan skala penyebaran Oracle Database di cloud. Anda dapat menerapkan database Anda dalam hitungan menit dan memilih penyimpanan Tujuan Umum (SSD) atau penyimpanan IOPS yang disediakan. (Untuk detailnya, lihat [jenis penyimpanan Amazon RDS](#) dalam AWS dokumentasi.)

Amazon RDS membebaskan Anda untuk fokus pada pengembangan aplikasi, karena mengelola tugas administrasi basis data yang memakan waktu, termasuk penyediaan, pencadangan, penambalan perangkat lunak, pemantauan, dan penskalaan perangkat keras. Amazon RDS for Oracle dengan mudah menyediakan replika baca dan database multi-AZ untuk meningkatkan ketersediaan, kinerja, dan keandalan untuk beban kerja produksi.

## Kapan memilih Amazon RDS

Amazon RDS for Oracle adalah opsi migrasi yang baik ketika:

- Anda ingin fokus pada bisnis dan aplikasi Anda, dan Anda AWS ingin mengurus tugas-tugas berat yang tidak terdiferensiasi seperti penyediaan database, manajemen tugas pencadangan dan pemulihan, manajemen patch keamanan, peningkatan versi Oracle minor, dan manajemen penyimpanan.
- Anda memerlukan solusi database yang sangat tersedia, dan Anda ingin memanfaatkan tombol tekan, replikasi Multi-AZ sinkron yang ditawarkan oleh Amazon RDS, tanpa harus mengatur dan memelihara database siaga secara manual.
- Anda ingin memiliki replikasi sinkron ke instance siaga, untuk menyediakan ketersediaan tinggi untuk database Oracle Database Standard Edition One (SE1) atau Standard Edition Two (SE2) Anda, daripada harus membayar Oracle Database Enterprise Edition (EE).

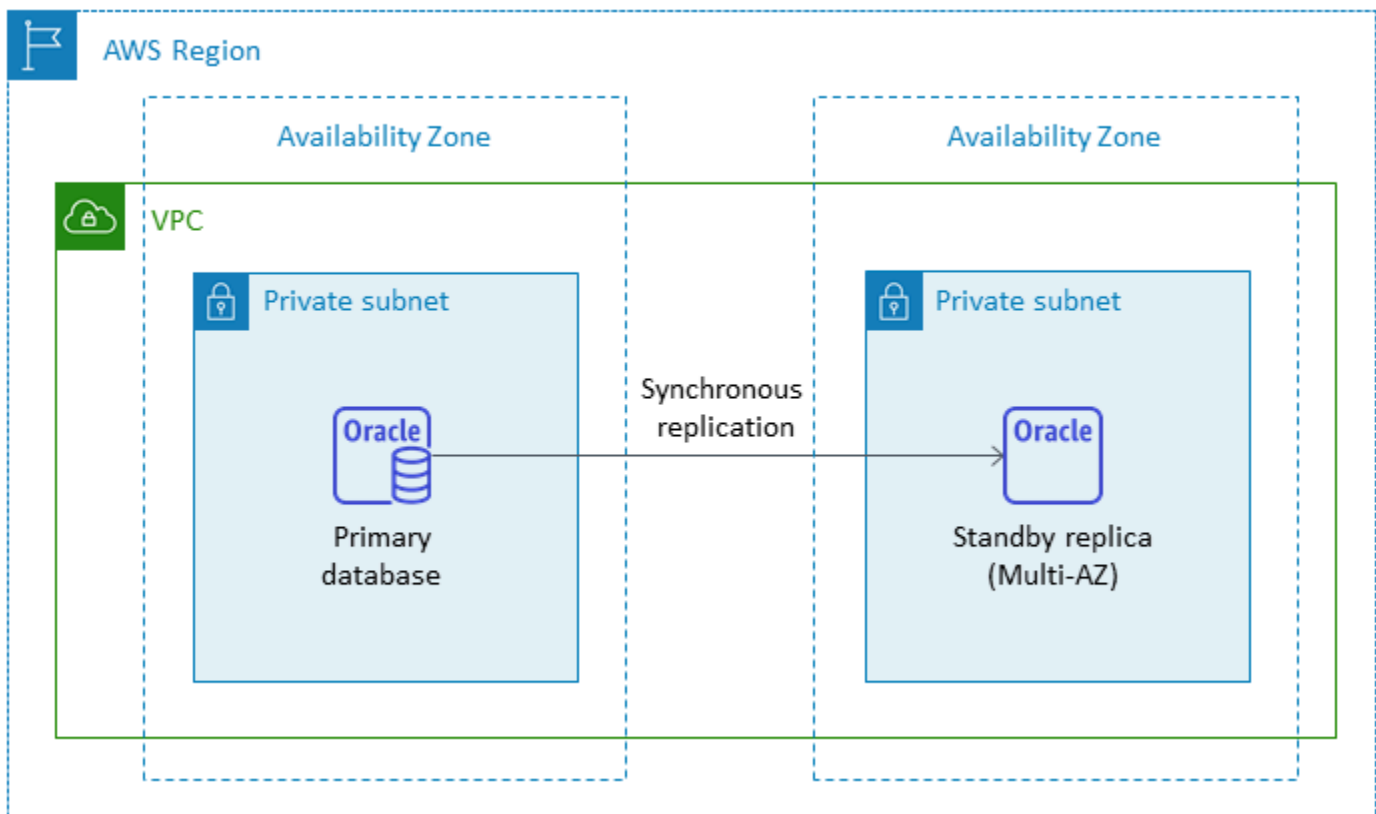
- Anda ingin membayar lisensi Oracle sebagai bagian dari biaya instans setiap jam daripada melakukan investasi di muka yang besar.
- Ukuran database dan kebutuhan IOPS Anda didukung oleh Amazon RDS for Oracle. Lihat [penyimpanan instans Amazon RDS DB](#) dalam AWS dokumentasi untuk batas maksimum saat ini.
- Anda tidak ingin mengelola backup atau point-in-time pemulihan database Anda.
- Anda lebih suka fokus pada tugas-tugas tingkat tinggi, seperti penyetelan kinerja dan pengoptimalan skema, daripada administrasi harian database.
- Anda ingin menskalakan jenis instans ke atas atau ke bawah berdasarkan pola beban kerja Anda tanpa khawatir tentang kompleksitas lisensi.

Setelah menilai database dan persyaratan proyek, jika Anda memutuskan untuk bermigrasi ke Amazon RDS for Oracle, lihat detail yang disediakan di bagian berikut, dan tinjau praktik terbaik migrasi yang akan dibahas nanti dalam panduan ini.

## Ketersediaan tinggi

Amazon RDS menyediakan ketersediaan tinggi dan dukungan failover untuk database yang digunakan dengan opsi Multi-AZ. Saat Anda menyediakan database Anda dengan opsi Multi-AZ, Amazon RDS secara otomatis menyediakan dan memelihara replika siaga sinkron di Availability Zone yang berbeda. Basis data utama secara sinkron mereplikasi data ke replika siaga di seluruh Availability Zones. Jika terjadi kegagalan infrastruktur atau gangguan Availability Zone, Amazon RDS melakukan failover otomatis ke replika siaga sehingga Anda dapat melanjutkan operasi database segera setelah failover selesai. Ini memberikan redundansi tinggi, daya tahan, dan peningkatan ketersediaan database utama Anda. Ini juga membongkar database utama Anda dengan mengambil cadangan otomatis dari replika siaga. Untuk informasi selengkapnya, lihat [Ketersediaan Tinggi \(Multi-AZ\) untuk Amazon RDS](#) dalam dokumentasi. AWS

Diagram berikut menggambarkan opsi penyebaran Amazon RDS for Oracle Multi-AZ. Aplikasi database dan pengguna terhubung ke database Oracle utama, dan semua perubahan secara sinkron direplikasi ke database sekunder, yang berada di Availability Zone yang berbeda. Database sekunder tidak tersedia untuk pengguna sampai failover selesai. Setelah failover, endpoint tetap sama, sehingga pengguna dan aplikasi database dapat melanjutkan operasi database tanpa intervensi manual.



## Replika baca

Replika baca adalah jenis khusus instans Amazon RDS for Oracle DB yang membantu mengurangi beban pada instans DB utama Anda. Pembaruan yang dilakukan pada instans DB utama Anda disalin secara asinkron ke replika baca, yang dapat Anda atur dalam hal yang sama atau yang lain. Wilayah AWS Wilayah AWS

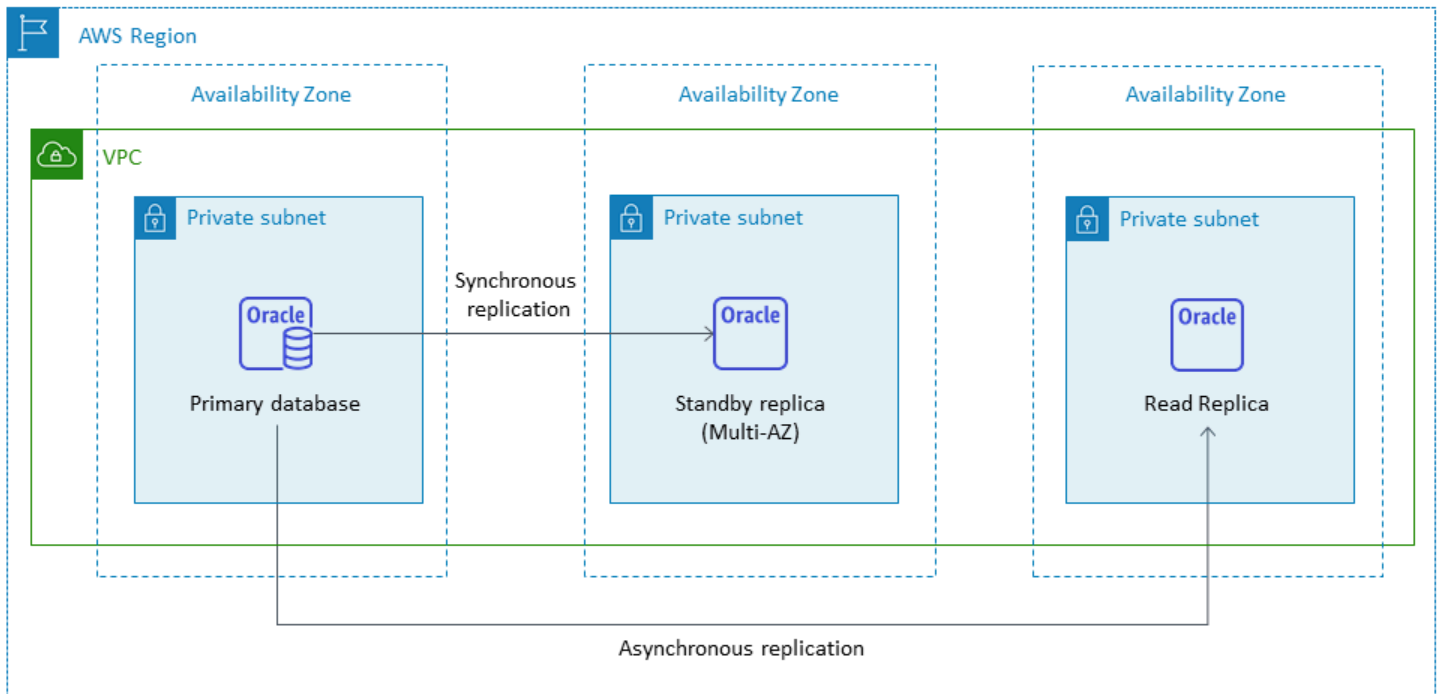
Anda dapat menyediakan database Amazon RDS for Oracle dengan replika baca menggunakan Oracle Active Data Guard untuk menurunkan beban kerja read-only Anda dari database Oracle utama. Oracle Active Data Guard mereplikasi perubahan database dari instance DB sumber ke replika baca. Fitur ini mendukung pemulihan bencana terkelola untuk database mission-critical dengan memungkinkan replika baca di database lain Wilayah AWS untuk dipromosikan sebagai database produksi baru yang mandiri. Anda dapat menyediakan hingga lima replika baca untuk database Amazon RDS for Oracle Anda.

Amazon RDS for Oracle memudahkan pembuatan replika baca dengan mengelola konfigurasi Active Data Guard dan menjaga koneksi jaringan yang aman antara instans DB primer dan replika bacanya. Untuk informasi selengkapnya, lihat [Bekerja dengan replika baca untuk Amazon RDS for Oracle dalam dokumentasi Amazon RDS](#).

Untuk menggunakan fitur replika baca, Anda harus menggunakan model Bring Your Own License (BYOL) dengan Oracle Database Enterprise Edition (EE) dan juga memiliki lisensi Active Data Guard.

### Menggunakan replika baca dalam hal yang sama Wilayah AWS

Diagram berikut mengilustrasikan instans Amazon RDS for Oracle DB di lingkungan multi-AZ dengan replika baca di Availability Zone lain dalam lingkungan yang sama. Wilayah AWS Tidak semua Wilayah AWS menawarkan lebih dari dua Availability Zone, jadi Anda harus [memeriksa Wilayah](#) yang Anda rencanakan untuk digunakan sebelum mengadopsi strategi ini.



### Menggunakan replika baca di Wilayah lain AWS

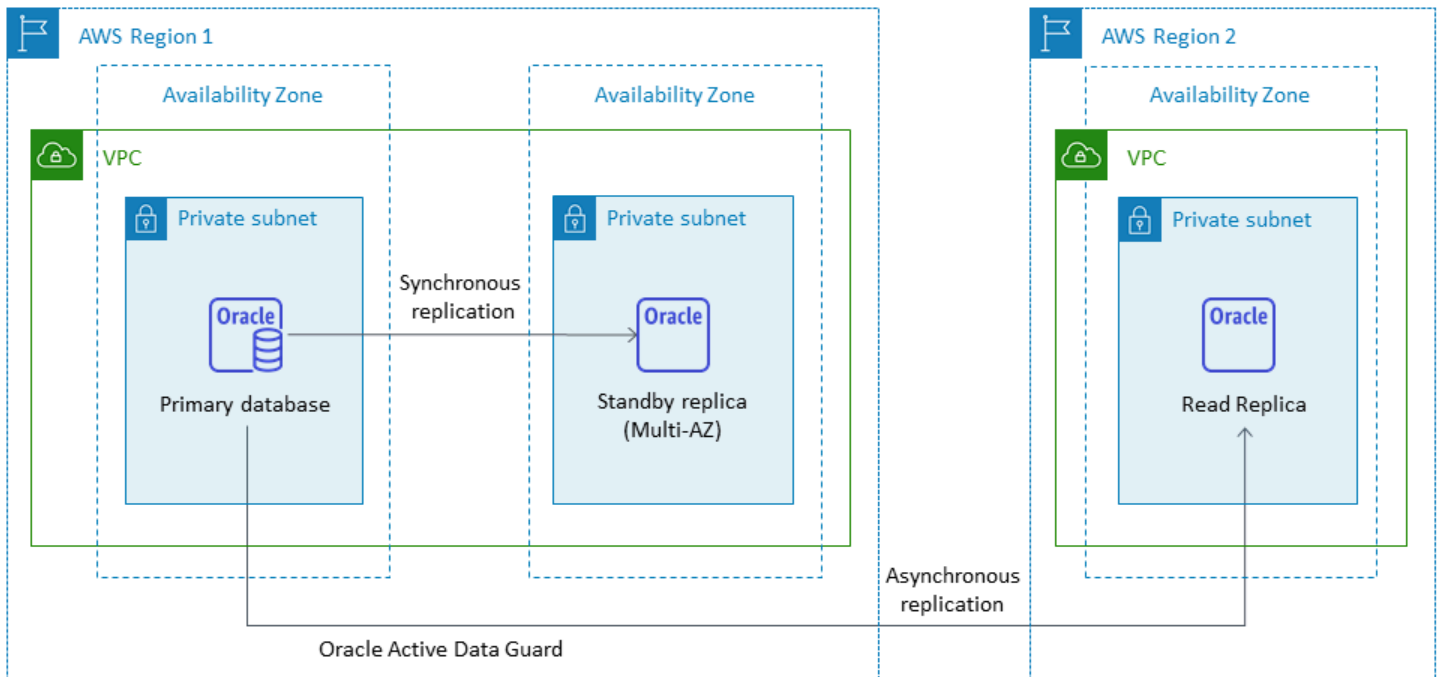
Amazon RDS for Oracle juga mendukung replika baca lintas wilayah. Ini menggunakan Oracle Active Data Guard untuk membuat dan mengelola konfigurasi instance DB siaga fisik yang berbeda Wilayah AWS dari instance DB primer. Ini mereplikasi data melalui koneksi jaringan yang aman antara instans DB primer dan replika bacanya di seluruh Wilayah.

Replika baca Lintas Wilayah menyediakan:

- Ketersediaan tinggi dan perlindungan data terhadap kegagalan Single-region.
- Kemampuan untuk menskalakan operasi baca ke operasi lain Wilayah AWS yang lebih dekat dengan pengguna aplikasi Anda.

Anda dapat mempromosikan replika baca Oracle ke instans DB mandiri secara eksplisit, atau Anda dapat mempromosikannya secara implisit dengan menghapus instans DB sumbernya. Saat Anda meningkatkan replika baca, instans DB akan di-boot ulang sebelum replika baca itu tersedia. Replika baca yang dipromosikan berperilaku sama seperti instans Oracle DB lainnya.

Diagram berikut menunjukkan konfigurasi replika baca Amazon RDS for Oracle Cross-region.



Data yang ditransfer untuk replikasi lintas Wilayah akan menimbulkan biaya transfer data Amazon RDS.

Untuk informasi selengkapnya tentang penggunaan replika baca, lihat [Bekerja dengan replika baca instans DB](#) dan [Bekerja dengan replika baca untuk Amazon RDS for Oracle](#) dalam dokumentasi.

AWS Untuk informasi selengkapnya tentang biaya transfer data, lihat [Harga Amazon RDS](#).

## Amazon RDS Custom for Oracle

Jika Anda tidak dapat pindah ke layanan yang dikelola sepenuhnya seperti Amazon RDS karena persyaratan penyesuaian, Anda dapat bermigrasi ke [Amazon RDS Custom](#) for Oracle. Dengan Amazon RDS Custom, Anda dapat mempertahankan hak administratif ke database dan sistem operasi yang mendasarinya.

## Kapan memilih Amazon RDS Custom untuk Oracle

Amazon RDS Custom for Oracle adalah opsi migrasi yang baik ketika:

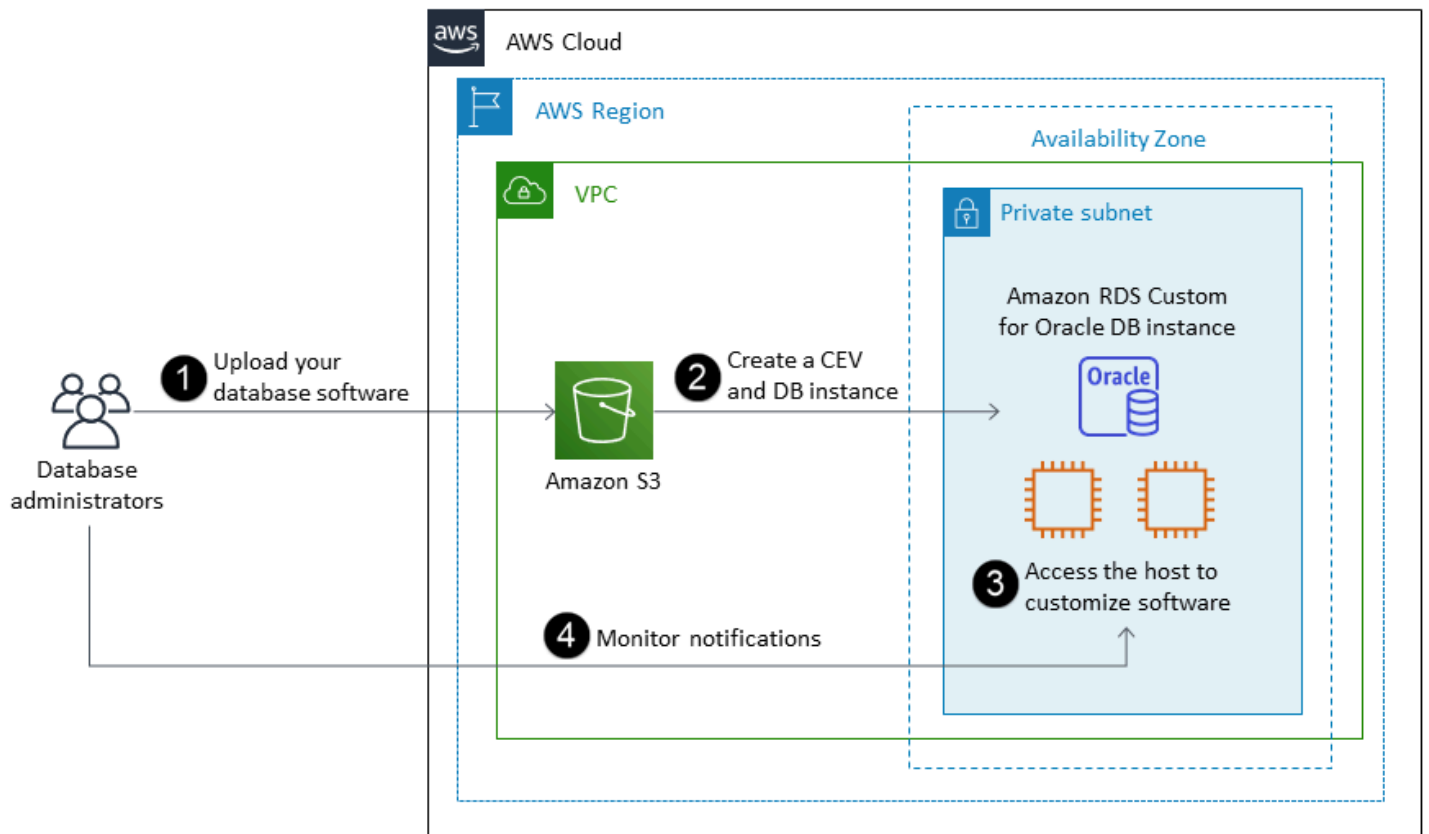
- Anda memiliki aplikasi lama, kustom, dan paket yang memerlukan akses ke sistem operasi dan lingkungan database yang mendasarinya.
- Anda memerlukan akses ke pengguna SYS atau SYSTEM untuk memenuhi persyaratan penerapan aplikasi berbasis vendor.
- Anda memerlukan akses ke sistem operasi yang mendasarinya untuk mengonfigurasi pengaturan, menginstal tambalan, dan mengaktifkan fitur asli untuk memenuhi persyaratan aplikasi dependen.
- Anda ingin mengakses dan menyesuaikan lingkungan database (dengan menerapkan patch database kustom atau memodifikasi paket sistem operasi) untuk memenuhi kebutuhan database dan aplikasi Anda.

## Cara kerjanya

Untuk menggunakan Amazon RDS Custom for Oracle, Anda mengikuti langkah-langkah ini, yang diilustrasikan dalam diagram berikut:

1. Unggah perangkat lunak database Anda ke bucket Amazon Simple Storage Service (Amazon S3).
2. Buat versi mesin khusus (CEV) dan instans DB.
3. Hubungkan aplikasi Anda ke titik akhir instans DB dan akses host untuk menyesuaikan perangkat lunak Anda.
4. Pantau notifikasi yang dihasilkan oleh otomatisasi Kustom Amazon RDS.

Untuk informasi selengkapnya tentang langkah-langkah ini, lihat [dokumentasi Kustom Amazon RDS](#).



Untuk menyediakan instans Amazon RDS Custom for Oracle DB, tinjau [persyaratan dalam dokumentasi](#) Amazon RDS Custom for Oracle.

Di Amazon RDS Custom for Oracle, Anda menggunakan media, patch, dan lisensi Oracle Anda sendiri. Saat Anda membuat [versi mesin khusus](#) (CEV), Amazon RDS Custom menginstal media yang Anda berikan. Anda memiliki akses ke instans EC2 yang mendasari yang menghosting mesin DB. Anda dapat mengakses instans EC2 dengan menggunakan Secure Shell (SSH) atau AWS Systems Manager dan melakukan penyesuaian Anda.

Anda juga dapat menginstal perangkat lunak untuk menjalankan aplikasi dan agen khusus. Karena Anda memiliki akses istimewa ke host, Anda dapat memodifikasi sistem file untuk mendukung aplikasi lama. Anda juga dapat menerapkan patch database kustom atau memodifikasi paket sistem operasi pada instans Amazon RDS Custom DB Anda.

Amazon RDS Custom secara otomatis menyediakan pemantauan, pencadangan, dan pemulihan instans, dan memastikan bahwa instans DB Anda menggunakan AWS infrastruktur, sistem operasi, dan database yang didukung. Jika ingin menyesuaikan instans, Anda dapat menjeda otomatisasi Amazon RDS Custom hingga 24 jam, lalu melanjutkannya saat pekerjaan penyesuaian Anda

selesai. Menjeda otomatisasi mencegah otomatisasi Amazon RDS secara langsung mengganggu penyesuaian Anda.

Saat Anda melanjutkan otomatisasi, [perimeter dukungan](#) menentukan apakah penyesuaian database atau lingkungan sistem operasi Anda mengganggu, atau merusak, otomatisasi Amazon RDS Custom. Amazon RDS Custom mendukung kustomisasi host dan lingkungan database selama perubahan Anda tidak menempatkan instans DB di luar perimeter dukungan. Pemeriksaan perimeter dukungan dilakukan setiap 30 menit secara default, dan juga terjadi setelah peristiwa seperti penghapusan snapshot atau menghapus instans Amazon RDS Custom, yang memantau instans DB. Agen Kustom Amazon RDS adalah komponen penting untuk memastikan fungsionalitas Amazon RDS Custom. Jika Anda menghapus instalasi agen, Amazon RDS Custom menjalankan pemeriksaan perimeter dukungan setelah satu menit dan memindahkan instans DB ke luar perimeter dukungan.

Amazon RDS Custom for Oracle tersedia di sistem operasi Oracle Linux dan mendukung Oracle Database Enterprise Edition dan Standard Edition pada model BYOL. Untuk lebih jelasnya, lihat [Ketersediaan fitur dan dukungan untuk RDS Custom for Oracle](#) dan [RDS Custom untuk persyaratan dan batasan Oracle](#) dalam dokumentasi. AWS

Untuk informasi tambahan, lihat sumber daya berikut:

- [Amazon RDS Kustom untuk Oracle - Kemampuan Kontrol Baru di Lingkungan Database \(BlogAWS berita\)](#)
- Menggunakan Amazon RDS for Oracle Cross-region backup otomatis untuk meningkatkan kemampuan DR Anda:
  - [Pemulihan bencana terkelola dengan Amazon RDS for Oracle Cadangan otomatis lintas wilayah - Bagian 1 \(Blog database\)AWS](#)
  - [Pemulihan bencana terkelola dengan Amazon RDS for Oracle Cadangan otomatis lintas wilayah - Bagian 2 \(Blog database\)AWS](#)
- Bermigrasi dari database Oracle lokal atau yang dikelola sendiri ke Amazon RDS Custom for Oracle dengan menggunakan alat bawaan:
  - [Migrasi fisik database Oracle ke Amazon RDS Custom menggunakan Data Guard \(AWS blog Database\)](#)
  - [Migrasi fisik database Oracle ke Amazon RDS Custom menggunakan duplikasi RMAN \(blog Database\)AWS](#)
- Mengintegrasikan sistem file bersama Amazon Elastic File System (Amazon EFS) dengan Amazon RDS for Oracle untuk berbagi file antara database dan server aplikasi atau sebagai lokasi

pementasan untuk menyimpan cadangan dan pemuatan data: [Integrasikan Amazon RDS Custom for Oracle dengan Amazon EFS](#) (Database blog)AWS

## Amazon EC2 untuk Oracle

Amazon EC2 mendukung database Oracle yang dikelola sendiri—yaitu, ini memberi Anda kontrol penuh atas penyiapan infrastruktur dan lingkungan database. Menjalankan database di Amazon EC2 sangat mirip dengan menjalankan database di server Anda sendiri. Anda memiliki kontrol penuh atas database dan akses tingkat sistem operasi, sehingga Anda dapat menggunakan alat pilihan Anda untuk mengelola sistem operasi, perangkat lunak database, tambalan, replikasi data, pencadangan, dan pemulihan. Opsi migrasi ini mengharuskan Anda menyiapkan, mengonfigurasi, mengelola, dan menyetel semua komponen, termasuk instans Amazon EC2, volume penyimpanan, skalabilitas, jaringan, dan keamanan, berdasarkan AWS praktik terbaik arsitektur.

## Kapan memilih Amazon EC2

Amazon EC2 adalah opsi migrasi yang baik untuk database Oracle Anda saat:

- Anda memerlukan kontrol penuh atas database dan akses ke sistem operasi yang mendasarinya.
- Anda ingin mengontrol backup, replikasi, dan clustering Anda.
- Anda ingin menggunakan fitur dan opsi yang saat ini tidak didukung oleh Amazon RDS. Untuk detailnya, lihat [Oracle Database Feature Support](#) di dokumentasi Amazon RDS.
- Anda memerlukan versi Oracle Database tertentu yang tidak didukung oleh Amazon RDS. Untuk daftar versi dan edisi yang didukung, lihat [Amazon RDS for Oracle di dokumentasi Amazon RDS](#).
- Ukuran database dan kebutuhan kinerja Anda melebihi penawaran Amazon RDS. Untuk detailnya, lihat [Penyimpanan instans Amazon RDS DB](#) dalam dokumentasi Amazon RDS.
- Anda ingin menghindari patch perangkat lunak otomatis yang mungkin tidak sesuai dengan aplikasi Anda.
- Anda ingin mencapai IOPS dan kapasitas penyimpanan penyediaan yang lebih tinggi daripada batas saat ini. Untuk detailnya, lihat [Penyimpanan instans Amazon RDS DB](#) dalam dokumentasi Amazon RDS.

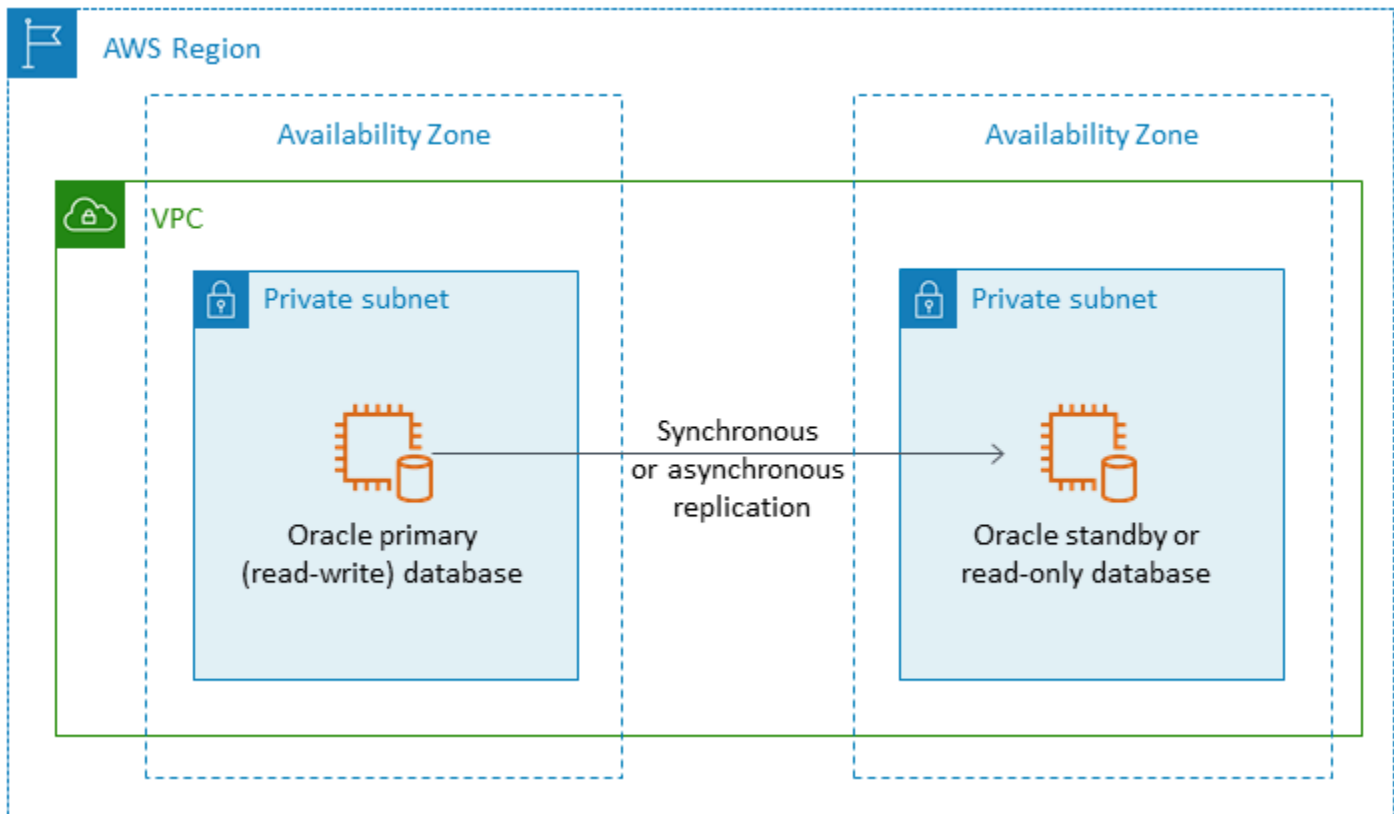
## Ketersediaan tinggi

Oracle Database di Amazon EC2 dapat bekerja dengan teknologi replikasi yang didukung Oracle untuk mencapai ketersediaan tinggi dan pemulihan bencana. Beberapa solusi umum adalah Oracle Data Guard, Oracle Active Data Guard, dan Oracle GoldenGate.

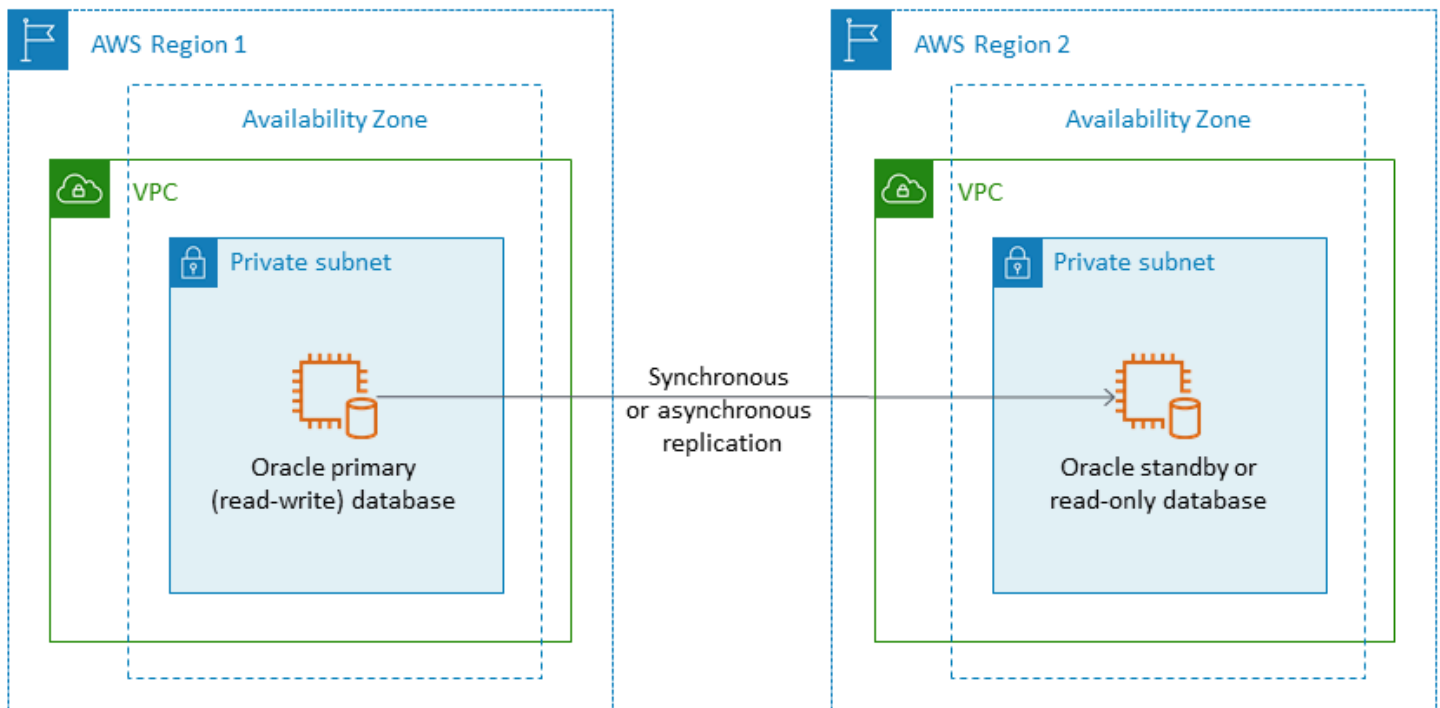
Database Oracle di Amazon EC2 menggunakan Oracle Data Guard atau Oracle Active Data Guard untuk mencapai ketersediaan tinggi, perlindungan data, dan pemulihan bencana.

- Oracle Data Guard menyediakan serangkaian layanan untuk membuat, memelihara, dan mengelola database siaga, untuk membantu melindungi database produksi Oracle terhadap bencana dan korupsi data. Oracle Data Guard secara otomatis memelihara setiap database siaga dengan mentransmisikan perubahan redo dari database utama, dan kemudian menerapkan redo ke database siaga. Jika database utama turun untuk pemadaman yang direncanakan atau tidak direncanakan, Anda dapat gagal ke database siaga dengan mengubahnya menjadi basis data baca-tulis utama. Oracle Data Guard disertakan dengan Oracle Database Enterprise Edition (EE) saja dan tidak memerlukan lisensi terpisah.
- Oracle Active Data Guard menyediakan akses read-only ke database siaga fisik untuk kueri, penyortiran, pelaporan, dan operasi baca lainnya sementara itu menerapkan perubahan redo terus menerus dari database utama. Oracle Active Data Guard memerlukan lisensi terpisah yang harus dibeli tambahan dengan Oracle Database EE. Fitur Oracle Active Data Guard meliputi Real-Time Query, Automatic Block Repair, Far Sync, Standby Block Change Tracking, Active Data Guard Rolling Upgrade, Layanan Database Global, dan Kontinuitas Aplikasi.

Diagram berikut menunjukkan bagaimana Anda dapat menggunakan Oracle Database di Amazon EC2 dalam dua Availability Zone dalam satu Wilayah AWS. Database utama adalah database baca-tulis, dan database siaga dikonfigurasi dengan Data Guard (siaga fisik tanpa akses baca) atau Active Data Guard. Semua data redo dari database utama ditransfer dan diterapkan ke database siaga secara asinkron secara default.



Anda juga dapat menggunakan Oracle Data Guard atau Oracle Active Data Guard untuk mengonfigurasi ketersediaan tinggi dan pemulihan bencana di beberapa Wilayah AWS, menggunakan Oracle Database di Amazon EC2 untuk database utama dan database siaga Anda, seperti yang diilustrasikan dalam diagram berikut.



## VMware Cloud on AWS untuk Oracle

### **i** Pemberitahuan

Per 30 April 2024, VMware Cloud on AWS tidak lagi dijual kembali oleh AWS atau mitra salurannya. Layanan ini akan terus tersedia melalui Broadcom. Kami mendorong Anda untuk menghubungi AWS perwakilan Anda untuk detailnya.

VMware Cloud on AWS adalah penawaran cloud terintegrasi yang dikembangkan bersama oleh AWS dan VMware. Saat Anda memigrasikan Oracle Database ke VMware Cloud on AWS, Anda memiliki kontrol penuh atas database dan akses tingkat sistem operasi, seperti Amazon EC2. Anda dapat menjalankan arsitektur canggih seperti Oracle Real Application Cluster (RAC) dan Oracle RAC extended cluster (di berbagai Availability Zone) di Cloud on. VMware AWS Anda dapat memilih dari sejumlah metode dan alat migrasi berdasarkan kebutuhan dan sistem yang ada.

Untuk migrasi online, VMware teknologi seperti VMware Hybrid Cloud Extension (VMware HCX) dan HCX vMotion membantu Anda memigrasikan beban kerja VM dari cluster lokal ke Cloud on. VMware VMware AWS Untuk migrasi offline beban kerja Oracle, Anda dapat menggunakan Oracle Recovery Manager (RMAN),,, AWS Snowball Edge atau HCX. AWS Storage Gateway VMware

## Kapan memilih VMware Cloud di AWS

VMware Cloud on AWS adalah pilihan yang baik untuk database Oracle Anda ketika:

- Database Oracle Anda sudah berjalan di pusat data lokal di lingkungan virtual vSphere.
- Anda perlu menjalankan Oracle RAC di cloud.
- Anda memiliki sejumlah besar database dan Anda memerlukan migrasi cepat (misalnya, hanya beberapa jam) ke cloud tanpa memerlukan pekerjaan tambahan dari tim migrasi.

Untuk informasi selengkapnya, lihat posting blog [Cara Memigrasi Beban Kerja Oracle ke VMware Cloud AWS](#) dan [Praktik Terbaik untuk Virtualisasi Oracle RAC dengan VMware Cloud AWS di blog Jaringan AWS Mitra \(APN\)](#).

## Alat untuk migrasi database homogen

### Pemberitahuan

Per 30 April 2024, VMware Cloud on AWS tidak lagi dijual kembali oleh AWS atau mitra salurannya. Layanan ini akan terus tersedia melalui Broadcom. Kami mendorong Anda untuk menghubungi AWS perwakilan Anda untuk detailnya.

Sejumlah alat dan teknologi tersedia untuk migrasi data. Anda dapat melakukan migrasi dalam satu fase atau dalam beberapa fase, berdasarkan ukuran database, konsistensi, bandwidth koneksi jaringan antara lingkungan lokal dan AWS, dan waktu yang diizinkan untuk migrasi database. Bagan berikut menyediakan daftar alat dan informasi untuk membantu Anda memilih opsi yang paling sesuai dengan kebutuhan Anda.

Alat migrasi	Ukuran basis data	Mendukung	Direkomendasikan untuk
<a href="#">Pengembang Oracle SQL (fitur Salin Database)</a>	Hingga 200 MB	Amazon RDS Amazon EC2	Database kecil dengan sejumlah objek.
<a href="#">Oracle SQL* Pemuat</a>	Hingga 10 GB	Amazon RDS Amazon EC2	Database berukuran kecil hingga

			menengah dengan jumlah objek terbatas.
<a href="#"><u>Utilitas Ekspor dan Impor Oracle</u></a>	Hingga 10 GB	Amazon RDS Amazon EC2	Database berukuran kecil hingga sedang dengan sejumlah besar objek.
<a href="#"><u>Pompa Data Oracle</u></a>	Hingga 20 TB	Amazon RDS Amazon EC2	Metode yang disukai untuk database apa pun yang berukuran 10 GB - 20 TB.
<a href="#"><u>AWS DMS</u></a>	Ukuran apa pun	Amazon RDS Amazon EC2	Migrasi downtime minimal. Ukuran database dibatasi oleh bandwidth. Anda dapat menggunakan an AWS DMS Oracle Data Pump untuk migrasi database besar.
<a href="#"><u>Oracle GoldenGate</u></a>	Ukuran apa pun	Amazon RDS Amazon VMware EC2 Cloud aktif AWS	Migrasi downtime minimal. Digunakan dengan Oracle Data Pump untuk migrasi database besar.
<a href="#"><u>Penjaga Data Oracle</u></a>	Ukuran apa pun	Amazon RDS Kustom Amazon VMware EC2 Cloud aktif AWS	Migrasi downtime minimal. Digunakan dengan Oracle RMAN untuk mereplikasi perubahan setelah transfer data awal.

<a href="#">Oracle RMAN</a>	Ukuran apa pun	Amazon RDS Kustom Amazon EC2  VMware Awan di AWS	Database lebih dari 2 TB, atau jika cadangan database sudah ada di Amazon Simple Storage Service (Amazon S3).
AWS Application Migration Service	Ukuran apa pun	Amazon EC2	Replikasi cepat dengan downtime minimal selama cutover. Untuk informasi selengkapnya, lihat <a href="#">dokumentasi Layanan Migrasi Aplikasi</a> .
<a href="#">VMware HCX</a>	Ukuran apa pun	VMware Awan di AWS	HCX vMotion menyediakan migrasi online atau offline dari mesin virtual tunggal (VM) pada satu waktu tanpa downtime.

Subbagian berikut memberikan informasi lebih lanjut tentang setiap alat.

## Pengembang Oracle SQL

[Oracle SQL Developer](#) adalah alat GUI gratis dari Oracle untuk manipulasi data, administrasi, pengembangan, dan manajemen. Alat berbasis Java ini tersedia untuk Microsoft Windows, Linux, atau macOS. Anda dapat menggunakan fitur Salin Database untuk memigrasikan database kecil ke AWS, di mana ukuran total data Anda di bawah 200 MB. Transfer data antara sumber dan basis data target dilakukan langsung melalui jaringan. Untuk menggunakan opsi ini, Anda memerlukan koneksi jaringan yang andal antara sumber dan basis data target. Selain itu, perlu diingat bahwa metode ini tidak mengenkripsi data selama transfer.

Oracle SQL Developer mendukung database Amazon RDS for Oracle dan Oracle di Amazon EC2.

## Oracle SQL\*Loader

[Oracle SQL\\* Loader](#) adalah utilitas pemuatan data massal yang tersedia dari Oracle untuk memuat data dari file eksternal ke dalam database. SQL\* Loader disertakan dengan binari klien Oracle Database lengkap. Anda dapat menggunakan SQL\* Loader untuk database berukuran kecil hingga menengah di bawah 10 GB yang berisi sejumlah objek terbatas. Karena ini adalah metode berbasis skema, ini melibatkan mengeksport skema tertentu secara individual dari database sumber dan memuatnya ke dalam database target. Jika Anda memiliki beberapa skema dalam database, Anda harus mengulangi proses untuk setiap skema.

Oracle SQL\* Loader mendukung database Amazon RDS for Oracle dan Oracle di Amazon EC2.

## Ekspor dan Impor Oracle

[Utilitas Ekspor dan Impor Oracle](#) membantu Anda memigrasikan database yang lebih kecil dari 10 GB dan tidak menyertakan tipe data float biner dan ganda. Proses impor membuat objek skema, jadi Anda tidak perlu menjalankan skrip untuk membuatnya sebelumnya. Ini membuat proses ini cocok untuk database yang memiliki sejumlah besar tabel kecil.

Anda dapat menggunakan alat ini untuk database Amazon RDS for Oracle dan Oracle di Amazon EC2.

## Oracle Data Pump

[Oracle Data Pump](#) adalah versi yang disempurnakan dari Oracle Export and Import. Utilitas ini digunakan untuk mengeksport dan mengimpor data dan metadata dari atau ke database Oracle. Anda dapat menjalankan Data Pump export/import pada seluruh database, skema selektif, tablespace, atau objek database. Data Pump adalah alat yang direkomendasikan untuk memigrasikan data ke AWS, untuk database besar yang berukuran mulai dari 10 GB hingga 20 TB. Ini memungkinkan paralelisme tingkat tinggi, opsi ekstraksi data yang fleksibel, dan operasi yang dapat diskalakan, yang memungkinkan pergerakan data dan metadata berkecepatan tinggi dari basis data sumber ke basis data target. Oracle Data Pump juga mendukung enkripsi dan kompresi saat mengeksport data Anda ke file dump data.

Anda dapat menggunakan alat ini untuk database Amazon RDS for Oracle dan Oracle di Amazon EC2. Anda juga dapat menggunakan Oracle Data Pump with AWS DMS dan Oracle GoldenGate untuk menangani transfer data awal untuk database besar.

Untuk Amazon RDS for Oracle, setelah data diekspor ke file dump menggunakan utilitas ekspor Oracle Data Pump, utilitas impor Oracle Data Pump mengharuskan file data tersedia di instance

server database untuk mengimpornya ke database. Anda tidak dapat mengakses sistem file di instans Amazon RDS DB secara langsung, jadi Anda perlu mentransfer file dump ke Amazon RDS menggunakan salah satu opsi ini:

- Gunakan link database antara dua database. Proses ini menggunakan Oracle Data Pump dan paket Oracle [DBMS\\_FILE\\_TRANSFER](#). Ini membuat link database antara sumber (lokal) database Oracle dan target Amazon RDS for Oracle database. Opsi ini membutuhkan konektivitas bandwidth yang lebih tinggi antara basis data sumber dan target; kami sarankan Anda menggunakannya [AWS Direct Connect](#). Opsi ini direkomendasikan hanya untuk database kecil. Untuk informasi selengkapnya, lihat [Mengimpor data dengan Oracle Data Pump dan tautan database](#) dalam dokumentasi Amazon RDS.
- Gunakan bucket Amazon S3. Amazon RDS for Oracle mendukung integrasi Amazon S3. Opsi ini direkomendasikan ketika Anda memiliki file dump data besar dan ukuran database Anda dalam terabyte. Anda kemudian dapat menyalin file dump data dari lokasi ke bucket S3 Anda dengan menggunakan AWS Direct Connect (jika ukuran data Anda dari 10 GB hingga 5 TB) atau AWS Snowball (jika ukuran data Anda lebih dari 5 TB) tergantung pada waktu migrasi yang diperlukan untuk database Anda.

Setelah file pompa data diunggah ke Amazon S3, Anda dapat mengunduhnya ke direktori DATA\_PUMP\_DIR pada instans Amazon RDS for Oracle DB target, lalu mengimpor data ke instans DB. Untuk informasi selengkapnya, lihat [Mengimpor data dengan Oracle Data Pump dan bucket Amazon S3 dalam dokumentasi](#) Amazon RDS.

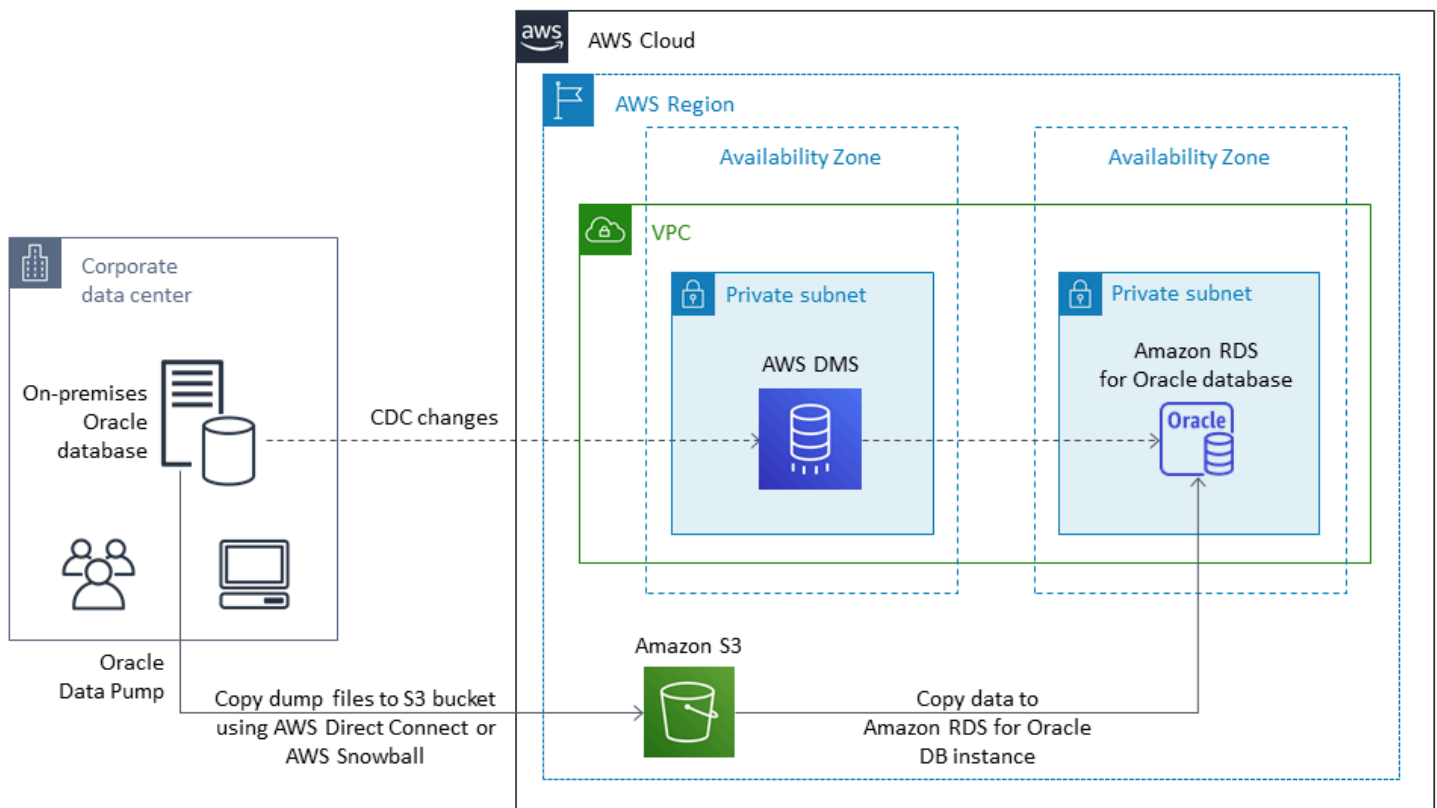
Dengan Oracle Data Pump, Anda dapat memigrasikan database yang lebih besar secara bertahap, secara bertahap. schema-by-schema Anda dapat bermigrasi ke versi perangkat lunak Oracle Database yang berbeda dan juga bermigrasi ke platform yang memiliki konfigurasi perangkat keras dan perangkat lunak yang berbeda.

## AWS DMS

[AWS Database Migration Service \(AWS DMS\)](#) adalah layanan terkelola yang membantu Anda memindahkan data ke dan dari AWS dengan mudah dan aman. AWS DMS mendukung sebagian besar database komersial dan open-source, dan memfasilitasi migrasi homogen dan heterogen. AWS DMS menawarkan teknologi salinan database penuh satu kali dan mengubah pengambilan data (CDC) untuk menjaga sumber dan basis data target tetap sinkron dan untuk meminimalkan waktu henti selama migrasi.

AWS DMS dapat melakukan salinan lengkap skema database Oracle Anda untuk database berukuran kecil (10-20 GB) hingga menengah (100-200 GB). Untuk database yang sangat besar, Anda dapat memigrasikan data ke Amazon RDS atau Amazon EC2 dengan menggunakan Oracle Data Pump, dan kemudian menggunakan fitur AWS DMS CDC untuk replikasi berkelanjutan dengan waktu henti minimal. Saat data disinkronkan, Anda dapat memotong ke database target.

Diagram berikut menunjukkan bagaimana Anda dapat menggunakan Oracle Data Pump dan AWS DMS bersama-sama untuk memigrasikan database lokal ke Amazon RDS for Oracle dengan waktu henti minimal. Utilitas ekspor Oracle Data Pump mengekspor skema ke file dump database, dan kemudian mentransfer file tersebut ke Amazon S3 dengan menggunakan salah satu AWS Direct Connect atau AWS Snowball (tergantung pada ukuran database, bandwidth jaringan, dan waktu migrasi yang diizinkan). Setelah file dump dimuat ke Amazon S3, Anda dapat mengunggah file ke instans Amazon RDS for Oracle DB. Utilitas impor Oracle Data Pump kemudian mengimpor data ke Amazon RDS for Oracle, AWS DMS dan CDC mereplikasi semua perubahan dari database sumber ke target Amazon RDS for Oracle database.



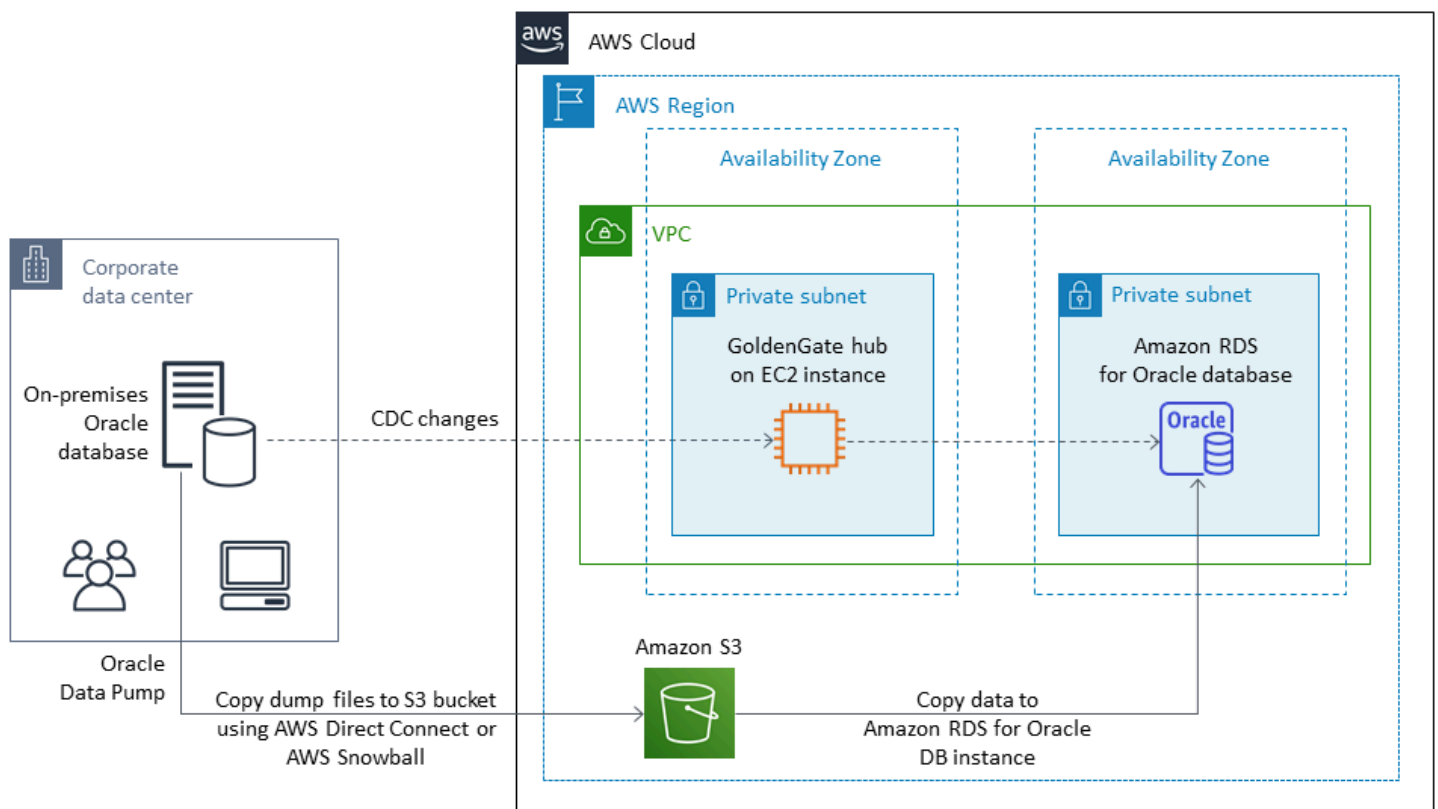
Untuk informasi selengkapnya tentang penggunaan AWS DMS untuk memigrasi database sumber Oracle, lihat [Menggunakan database Oracle sebagai sumber dalam dokumentasi](#). AWS DMS AWS

## Oracle GoldenGate

[Oracle GoldenGate](#) adalah alat untuk mereplikasi data antara database sumber dan satu atau lebih database tujuan dengan downtime minimal. Anda dapat menggunakannya untuk membangun arsitektur ketersediaan tinggi, dan untuk melakukan integrasi data real-time, pengambilan data perubahan transaksional, replikasi di lingkungan heterogen, dan replikasi data berkelanjutan.

Anda dapat menjalankan Oracle GoldenGate dari server lokal Anda di lingkungan sumber Anda. Namun, kami menyarankan Anda menginstal dan menjalankan alat ini dari instans EC2, yang berfungsi sebagai GoldenGate hub, AWS untuk kinerja yang lebih baik. Anda dapat memiliki beberapa GoldenGate hub, terutama jika Anda memigrasikan data dari satu database sumber ke beberapa tujuan. Anda dapat menggunakan GoldenGate Amazon RDS untuk replikasi database Aktif-Aktif, migrasi dan peningkatan waktu henti tanpa henti, pemulihan bencana, perlindungan data, serta replikasi dalam wilayah dan lintas wilayah. Untuk detailnya, lihat [Menggunakan Oracle GoldenGate dengan Amazon RDS](#) dalam dokumentasi. AWS

Diagram berikut menunjukkan cara menggunakan Oracle Data Pump dan Oracle GoldenGate bersama-sama untuk memigrasikan database Oracle lokal ke Amazon RDS for Oracle.



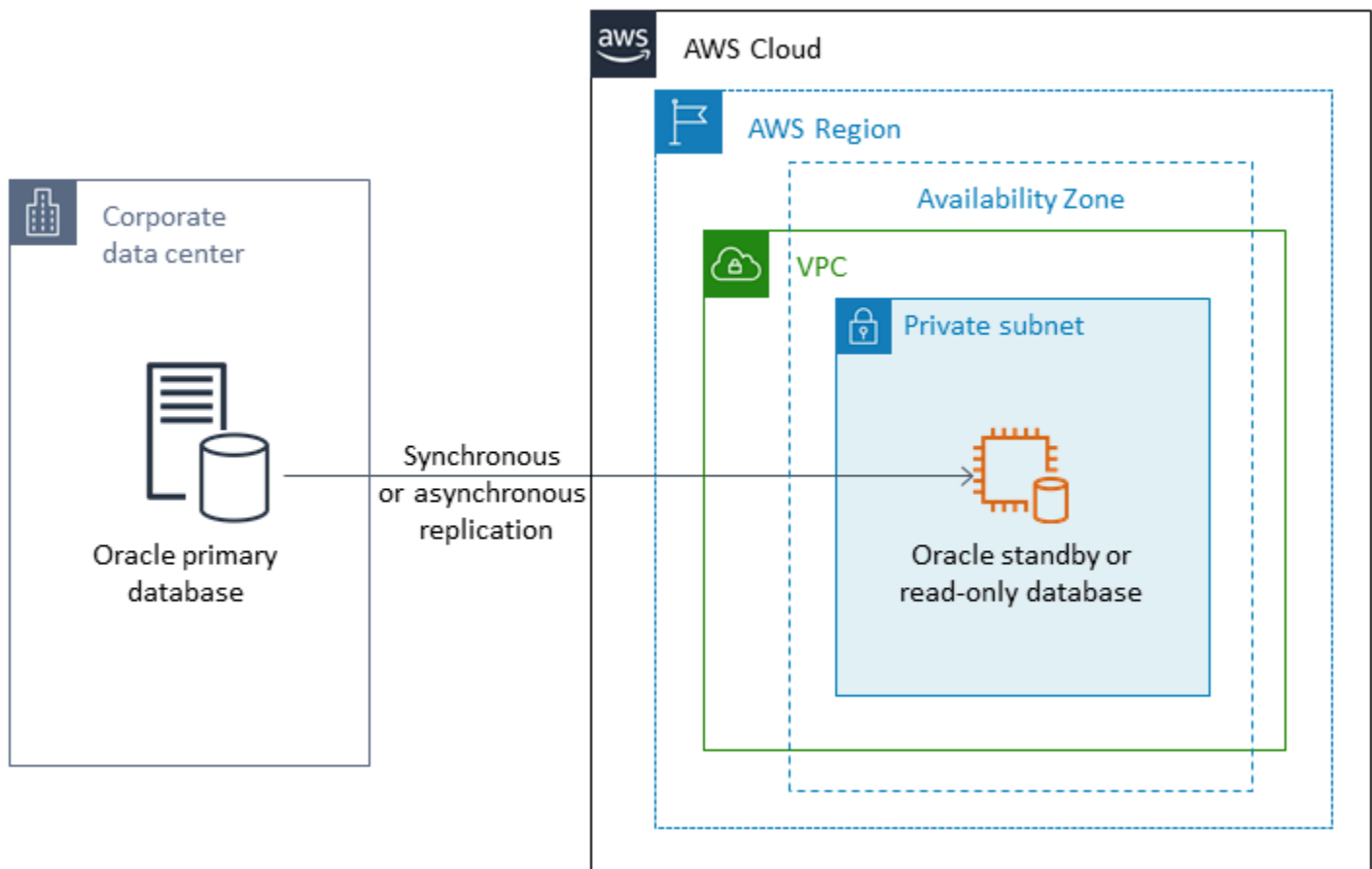
Oracle GoldenGate membutuhkan lisensi terpisah dari Oracle.

Oracle GoldenGate mendukung database Amazon RDS for Oracle dan Oracle yang berjalan di Amazon EC2 atau Cloud on. VMware AWS

## Oracle Data Guard

[Oracle Data Guard](#) menyediakan serangkaian layanan untuk membuat, memelihara, memantau, dan mengelola database siaga Oracle. Anda dapat memigrasikan seluruh database Oracle dari lokasi ke Amazon EC2 dengan waktu henti minimal menggunakan Oracle Recovery Manager (RMAN) dan Oracle Data Guard. Dengan RMAN, Anda memulihkan database utama Anda ke database siaga target di Amazon EC2, menggunakan backup/restore salah satu atau metode database duplikat. Anda kemudian mengkonfigurasi database target sebagai database siaga fisik dengan Oracle Data Guard, memungkinkan semua perubahan transaction/redo data dari database lokal utama ke database siaga.

Ketika database Oracle lokal utama disinkronkan dengan database siaga target pada instans EC2, Anda dapat beralih ke database target, yang akan mengubahnya menjadi database baca-tulis. Anda kemudian dapat mengarahkan koneksi aplikasi Anda ke database utama yang baru. Dengan opsi ini, Anda dapat mencapai waktu henti minimum dan mendapatkan salinan fisik persis dari database Anda. AWS Migrasi diilustrasikan dalam diagram berikut.



Oracle Data Guard mendukung database Oracle yang berjalan di Amazon EC2, Amazon RDS Custom, dan Cloud on. VMware AWS

## Oracle RMAN

[Oracle Recovery Manager \(RMAN\)](#) adalah alat yang disediakan oleh Oracle untuk melakukan dan mengelola backup dan restorasi database Oracle. Anda dapat menggunakan RMAN untuk mencadangkan database Oracle Anda dari tempat atau dari pusat data Anda, dan mengembalikannya ke database Oracle pada instans EC2. Gunakan metode ini jika Anda berencana untuk memindahkan seluruh database Anda ke database Oracle yang dikelola sendiri pada instans EC2. Basis data dapat berukuran berapa pun, dan Anda dapat menggunakan paralelisme, kompresi, dan enkripsi dalam cadangan Anda.

Anda dapat menempatkan cadangan Oracle RMAN dari database Oracle lokal Anda secara langsung di bucket S3 dengan menggunakan modul Cloud Oracle Secure Backup (OSB), atau. AWS Storage Gateway AWS DataSync Anda kemudian dapat menggunakan peran AWS Identity and Access Management (IAM) untuk memberikan akses bucket S3 ke database Oracle target Anda pada instans EC2, dan memulihkan database dengan menggunakan file cadangan RMAN. Anda dapat mengambil

cadangan tambahan dari database Oracle lokal Anda dan menerapkannya ke database Oracle target pada instans EC2 hingga database lokal dan target disinkronkan. Anda kemudian dapat melakukan peralihan pada waktu yang tepat.

Oracle RMAN mendukung Amazon EC2, Amazon RDS Custom VMware , dan Cloud pada migrasi. AWS Ini adalah pendekatan yang disarankan ketika Anda dapat mengizinkan waktu henti yang cukup untuk memigrasikan data Anda ke. AWS

## VMware HCX

[VMware Hybrid Cloud Extension \(HCX\)](#) memungkinkan Anda untuk memigrasikan database Oracle lokal Anda AWS tanpa harus memperbaiki infrastruktur Anda. VMware Ini mencakup beberapa metode migrasi yang dirinci dalam posting blog [Cara Migrasi Oracle Workloads ke VMware Cloud on AWS dan Migrating Workloads to Cloud on AWS dengan Hybrid VMware Cloud Extension \(HCX\)](#). Salah satu metode ini, HCX vMotion, menyediakan migrasi langsung dari VM tunggal tanpa downtime dan ketersediaan tinggi.

HCX tersedia gratis untuk VMware Cloud pada AWS pelanggan.

## Opsi lisensi

Lisensi Oracle Database AWS didasarkan pada ukuran instance tempat database diinstal. Banyak beban kerja Oracle Database membutuhkan memori tinggi, penyimpanan, dan bandwidth I/O, tetapi tidak terikat CPU, sehingga Anda dapat mengurangi jumlah virtual CPUs (vCPUs) dalam penyebaran Anda tanpa mempengaruhi kinerja.

AWS menawarkan opsi CPU berikut untuk mengoptimalkan instans Amazon RDS dan EC2 Anda untuk beban kerja atau kebutuhan bisnis tertentu:

- Jumlah inti CPU: Anda dapat menyesuaikan jumlah inti CPU untuk instans.
- Thread per inti: Anda dapat menonaktifkan multithreading dengan menentukan satu thread per inti CPU.

Untuk informasi selengkapnya, lihat [Mengoptimalkan opsi CPU](#) di dokumentasi Amazon EC2 [dan Memperkenalkan CPUs Optimalkan untuk Amazon RDS for Oracle](#) di situs web. AWS

Anda dapat menjalankan Oracle Database di AWS bawah dua model lisensi yang berbeda:

- Termasuk Lisensi
- Bawa Lisensi Sendiri (BYOL)

## Termasuk Lisensi

Dalam model Lisensi Termasuk, lisensi perangkat lunak Oracle Database disediakan oleh AWS, sehingga Anda tidak perlu membeli lisensi Oracle Anda sendiri secara terpisah. Harga model Termasuk Lisensi mencakup perangkat lunak, sumber daya perangkat keras yang mendasarinya, dan kemampuan manajemen Amazon RDS untuk Amazon RDS for Oracle. Anda membayar kapasitas komputasi per jam instans DB Anda berjalan, tanpa komitmen jangka panjang. Ini membebaskan Anda dari biaya dan kompleksitas perencanaan, pembelian, dan pemeliharaan perangkat keras.

Untuk penerapan Single-AZ dan multi-AZ, harga digunakan per jam instans DB, dari saat Anda meluncurkan instans DB hingga Anda menghentikan atau menghapus instans.

Model Termasuk Lisensi mendukung Edisi Standar Dua (SE2). Untuk informasi harga, lihat harga [AWS Amazon RDS for Oracle](#) di situs web.

## BYOL

Model Bring Your Own License (BYOL) ditujukan untuk pelanggan yang lebih suka menggunakan lisensi Oracle Database yang ada atau membeli lisensi Oracle baru. Jika Anda sudah memiliki lisensi database Oracle, Anda dapat menggunakan model BYOL untuk menjalankan database Oracle Anda di Amazon RDS. Jika Anda memigrasikan database Oracle Anda ke Amazon EC2 atau VMware ke Cloud AWS on, Anda harus menggunakan lisensi Oracle Anda sendiri.

### Pemberitahuan

Per 30 April 2024, VMware Cloud on AWS tidak lagi dijual kembali oleh AWS atau mitra salurannya. Layanan ini akan terus tersedia melalui Broadcom. Kami mendorong Anda untuk menghubungi AWS perwakilan Anda untuk detailnya.

Untuk menjalankan instance DB di bawah model BYOL, Anda harus memiliki lisensi Oracle Database yang sesuai untuk kelas instans DB dan edisi Oracle Database yang ingin Anda jalankan. Anda juga harus mengikuti kebijakan Oracle untuk lisensi perangkat lunak Oracle Database di lingkungan komputasi cloud.

Jika Anda menggunakan model BYOL, Anda harus memiliki lisensi untuk instans DB primer dan instans DB siaga dalam penerapan Multi-AZ. Amazon RDS mendukung deployment Multi-AZ untuk Oracle sebagai solusi failover dengan ketersediaan tinggi. Sebaiknya gunakan Multi-AZ untuk beban kerja produksi. Untuk informasi selengkapnya, lihat [Mengonfigurasi dan mengelola penerapan Multi-AZ](#) di dokumentasi Amazon RDS.

Model BYOL mendukung Oracle Database Enterprise Edition (EE) dan Standard Edition Two (SE2).

Untuk informasi selengkapnya tentang opsi lisensi untuk Amazon RDS for Oracle, [lihat Oracle Licensing dan Amazon RDS for Oracle](#) di situs web. FAQs AWS

## Migrasi database heterogen

Karena inovasi dan peningkatan dalam database open-source dan platform komputasi awan seperti AWS, banyak organisasi beralih dari mesin database proprietary (pemrosesan transaksi online atau OLTP) seperti Oracle ke mesin open-source. Database Oracle adalah sistem misi penting untuk organisasi mana pun, tetapi dikunci ke vendor tertentu adalah situasi yang berisiko dan mahal. Biaya operasional yang rendah dan tidak ada biaya lisensi adalah alasan kuat untuk mempertimbangkan mengalihkan teknologi basis data yang mendasarinya ke database open-source atau cloud-native. AWS

Alasan lain untuk bermigrasi dari Oracle adalah periode penguncian vendor, audit lisensi, lisensi mahal, dan biaya. Daftar harga Oracle didasarkan pada model per-inti dengan biaya tambahan untuk fitur seperti partisi dan ketersediaan tinggi. Untuk alasan ini, banyak organisasi memilih untuk memigrasikan database Oracle mereka ke database sumber terbuka (seperti PostgreSQL, MySQL, atau MariaDB) atau database Cloud-native (seperti Amazon Aurora atau Amazon DynamoDB) saat mereka bermigrasi. AWS AWS

Anda juga dapat memigrasikan database gudang data Oracle Anda ke Amazon Redshift, yang merupakan gudang data cloud yang cepat dan terkelola sepenuhnya. Amazon Redshift terintegrasi dengan data lake Anda, menawarkan kinerja hingga tiga kali lebih cepat daripada gudang data lainnya, dan biaya hingga 75 persen lebih murah daripada gudang data cloud lainnya. Untuk informasi selengkapnya, lihat [Migrasi dari Oracle ke Amazon Redshift di](#) situs web. AWS

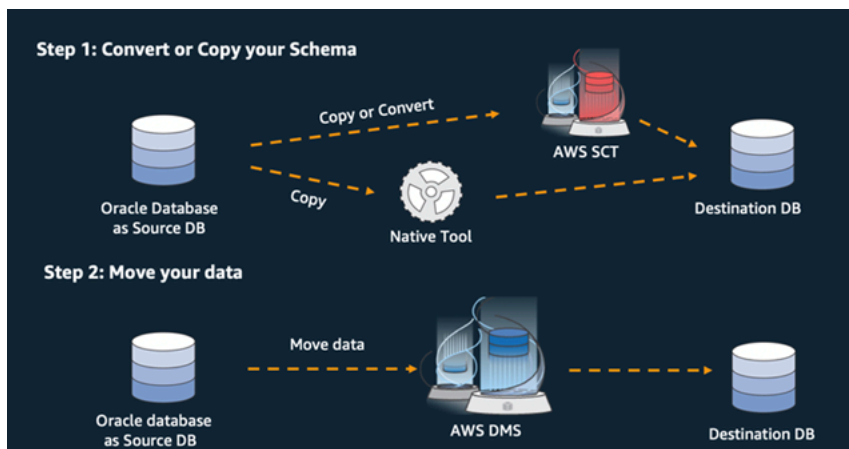
Untuk bermigrasi ke database open-source atau AWS-native, pilih database yang tepat tergantung pada jenis data yang Anda miliki, model akses, skalabilitas, kepraktisan aplikasi, dan kompleksitas. Misalnya, database PostgreSQL telah menjadi sangat populer dalam beberapa tahun terakhir karena fungsionalitasnya yang kuat dan tingkat kompatibilitas yang tinggi dengan database komersial, dan mereka adalah target migrasi paling umum bagi pengguna yang melakukan refactoring database Oracle mereka. Tetapi bermigrasi dari Oracle ke PostgreSQL dan ke database open-source lainnya seringkali sulit dan memakan waktu, dan membutuhkan penilaian, perencanaan, dan pengujian yang cermat.

Proses ini menjadi lebih mudah dengan layanan seperti AWS DMS dan AWS Schema Conversion Tool (AWS SCT), yang membantu Anda memigrasikan database komersial Anda ke database sumber terbuka AWS dengan waktu henti minimal.

Dalam migrasi database heterogen, mesin database sumber dan target berbeda, seperti pada migrasi Oracle ke Amazon Aurora, atau Oracle ke PostgreSQL, MySQL, atau MariaDB. Struktur

skema, tipe data, dan kode database dalam basis data sumber dan target bisa sangat berbeda, sehingga skema dan kode harus diubah sebelum migrasi data dimulai. Untuk alasan ini, migrasi heterogen adalah proses dua langkah:

- Langkah 1. Ubah skema sumber dan kode agar sesuai dengan basis data target. Anda dapat menggunakan AWS SCT untuk konversi ini.
- Langkah 2. Migrasikan data dari database sumber ke database target. Anda dapat menggunakan AWS DMS untuk proses ini.



AWS DMS menangani semua konversi tipe data yang diperlukan secara otomatis selama migrasi. Database sumber dapat ditempatkan di tempat Anda sendiri di luar AWS, dapat berupa database yang berjalan pada instans EC2, atau dapat berupa database Amazon RDS (lihat [Sumber untuk migrasi data](#) dalam AWS DMS dokumentasi). Targetnya bisa berupa database di Amazon EC2, Amazon RDS, atau Amazon Aurora.

## Alat untuk migrasi database heterogen

Bagan berikut menyediakan daftar alat yang dapat Anda gunakan untuk bermigrasi dari Oracle Database ke mesin database lain.

Alat migrasi	Dukungan basis data target	Digunakan untuk
<a href="#">AWS SCT</a>	Amazon RDS for MySQL Amazon RDS for PostgreSQL Amazon Aurora MySQL	Konversi skema

	Amazon Aurora PostgreSQL	
<a href="#">AWS DMS</a>	Amazon RDS for MySQL	Migrasi data
	Amazon RDS for PostgreSQL	
	Amazon Aurora MySQL	
	Amazon Aurora PostgreSQL	

Subbagian berikut memberikan informasi lebih lanjut tentang setiap alat.

## AWS SCT

[AWS Schema Conversion Tool \(AWS SCT\)](#) mengonversi skema basis data komersial Anda yang ada menjadi mesin sumber terbuka atau ke database Cloud-native. AWS SCT membuat migrasi database heterogen dapat diprediksi dengan secara otomatis mengonversi skema basis data sumber dan sebagian besar objek kode database, termasuk tampilan, prosedur tersimpan, dan fungsi, ke format yang kompatibel dengan database target. Objek apa pun yang tidak dapat dikonversi secara otomatis ditandai dengan jelas untuk konversi manual. AWS SCT juga dapat memindai kode sumber aplikasi Anda untuk pernyataan SQL tertanam dan mengonversinya sebagai bagian dari proyek konversi skema database.

## AWS DMS

[AWS Database Migration Service \(AWS DMS\)](#) memigrasikan data Anda dengan cepat dan aman ke AWS. Selama migrasi, database sumber tetap beroperasi penuh, meminimalkan waktu henti aplikasi. AWS DMS mendukung migrasi homogen seperti Oracle ke Oracle serta migrasi heterogen antara platform database yang berbeda, seperti Oracle ke database open-source atau ke database cloud-native. AWS DMS mengelola kompleksitas proses migrasi, termasuk secara otomatis mereplikasi perubahan data yang terjadi dalam database sumber ke database target. Setelah migrasi database selesai, database target tetap disinkronkan dengan database sumber selama yang Anda pilih, dan Anda dapat beralih ke database target pada waktu yang tepat.

# Praktik terbaik untuk bermigrasi ke Amazon RDS for Oracle

Berdasarkan penilaian database Anda dan persyaratan proyek Anda, jika tujuan Anda adalah untuk bermigrasi ke Amazon RDS for Oracle, ikuti praktik terbaik di bagian ini untuk menyediakan database target Anda, melakukan migrasi, dan menguji, mengoperasikan, dan mengoptimalkan database Amazon RDS for Oracle Anda.

## Important

Pastikan Anda memiliki paket rollback sebelum memigrasikan database Anda.

## Menyediakan basis data target Anda

Setelah Anda selesai menilai, merencanakan, dan menyiapkan strategi migrasi database, ikuti praktik terbaik berikut saat menyediakan database Amazon RDS for Oracle Anda:

- Ukuran tepat instans Amazon RDS for Oracle DB berdasarkan kebutuhan Anda untuk CPU, memori, IOPS, dan jenis penyimpanan.
- Atur zona waktu dan set karakter yang benar.
- Pastikan untuk meluncurkan Amazon RDS di cloud pribadi virtual (VPC) yang benar.
- Buat grup keamanan dengan port dan alamat IP yang benar.
- Menyediakan database Amazon RDS Anda di subnet pribadi untuk keamanan.
- Jika memungkinkan, berikan instans DB dengan menggunakan versi Oracle Database terbaru, yang saat ini 19c. Versi sebelumnya mendekati akhir dukungan. Untuk informasi selengkapnya, lihat [dukungan Amazon RDS untuk Oracle Database 19c](#).
- Jika Anda ingin menggunakan enkripsi, selalu aktifkan saat Anda menyediakan database Amazon RDS.
- Buat grup opsi dan grup parameter terpisah untuk setiap database Amazon RDS.

## Mengekspor data dari database sumber Anda

Ada banyak alat untuk memigrasikan database Oracle ke database Amazon RDS for Oracle. Alat yang paling umum digunakan adalah Oracle Data Pump. Sebelum Anda mengekspor database Oracle sumber Anda, periksa hal berikut untuk memfasilitasi proses ekspor:

- Periksa ukuran database, untuk melihat apakah Anda dapat memigrasikan skema berdasarkan skema, alih-alih memigrasikan database lengkap. Migrasi skema satu per satu kurang rawan kesalahan dan lebih mudah dikelola daripada memigrasikan semuanya sekaligus.
- Ekspor data dalam mode paralel, dengan menggunakan parameter Paralel Oracle Data Pump, untuk kinerja yang lebih baik.
- Periksa apakah tabel memiliki objek besar (LOBs). Jika Anda memiliki tabel besar LOBs, kami sarankan Anda mengekspor tabel tersebut secara terpisah.
- Selama proses ekspor, hindari menjalankan transaksi database yang panjang pada database sumber Anda untuk menghindari kesalahan inkonsistensi baca Oracle.
- Jika Anda menggunakan alat replikasi seperti AWS DMS, Oracle GoldenGate, atau Quest SharePlex, pastikan Anda memiliki cukup ruang di server lokal untuk menyimpan log arsip selama 24-72 jam, tergantung pada berapa lama migrasi berlangsung.

## Mentransfer file dump data ke AWS

[Jika Anda menggunakan AWS Direct Connect, yang menyediakan konektivitas bandwidth tinggi antara lingkungan lokal dan AWS, Anda dapat menyalin file Pompa Data dengan menggunakan utilitas Oracle DBMS\\_FILE\\_TRANSFER atau fitur integrasi Amazon S3.](#) Jika Anda tidak memiliki bandwidth tinggi AWS Direct Connect, gunakan AWS Snowball untuk mentransfer file dump ekspor database besar.

## Mengimpor data ke database target Anda

- Jika Anda memigrasikan database yang sangat besar, sebaiknya Anda menyediakan [jenis instans Amazon RDS](#) yang lebih besar pada awalnya, selama durasi migrasi, untuk pemuatan data yang lebih cepat. Setelah migrasi selesai, Anda dapat mengubah instans DB ke jenis instans berukuran tepat.
- Tingkatkan ukuran file log ulang, batalkan ruang tabel, dan ruang tabel sementara untuk meningkatkan kinerja selama migrasi, jika diperlukan.
- Nonaktifkan opsi Multi-AZ selama proses impor, dan aktifkan setelah migrasi selesai.
- Nonaktifkan pembuatan log arsip dengan mengatur retensi cadangan ke nol untuk mencapai pemuatan data yang lebih cepat.
- Siapkan database target dengan membuat ruang tabel, pengguna, peran, profil, dan skema terlebih dahulu.

- Jika Anda memiliki tabel besar LOBs, impor setiap tabel LOB secara terpisah.

## Langkah pasca-impor

- Periksa file log impor untuk kesalahan, dan perbaiki kesalahan apa pun setelah impor selesai.
- Periksa objek yang tidak valid. Jika Anda menemukannya, kompilasi dan perbaiki.
- Beberapa prosedur mungkin tidak dikompilasi karena kurangnya izin pada objek SYS yang tidak diizinkan atau didukung di Amazon RDS. Prosedur-prosedur ini harus ditulis ulang.
- Jika Anda menggunakan urutan, validasi nilai urutan terhadap database sumber untuk menghindari inkonsistensi urutan.
- Pastikan jumlah objek di database Amazon RDS Anda sama dengan di database sumber. Validasi tabel, indeks, prosedur, pemicu, fungsi, paket, kendala, dan objek lainnya.
- Jika database sumber Anda memiliki tautan basis data ke database lain, uji konektivitas untuk mengonfirmasi bahwa tautan masih berfungsi.
- Kumpulkan statistik tingkat kamus dan tingkat skema untuk kinerja optimal.

## Menguji migrasi

Kami merekomendasikan tes berikut untuk memvalidasi aplikasi Anda terhadap database Amazon RDS for Oracle baru Anda:

- Anda mungkin harus memutakhirkan perangkat lunak klien Oracle atau perangkat lunak JDBC Anda berdasarkan versi database Amazon RDS for Oracle. Jika Anda telah bermigrasi ke versi Oracle Database yang lebih baru, mungkin tidak mendukung versi perangkat lunak klien Oracle yang lebih lama.
- Lakukan pengujian fungsional.
- Bandingkan kinerja kueri SQL di basis data sumber dan target Anda, dan sesuaikan kueri sesuai kebutuhan. Beberapa query mungkin bekerja lebih lambat dalam database target, jadi kami sarankan Anda menangkap baseline dari query SQL dalam database sumber.
- Ketika tim aplikasi selesai menguji dan mengonfirmasi bahwa database Amazon RDS Anda berfungsi dengan baik, Anda dapat:
  - Ukuran instans Amazon RDS DB yang tepat berdasarkan penilaian Anda.
  - Aktifkan retensi cadangan.
  - Aktifkan log arsip.

- Setel ulang ukuran file log ulang.
- Aktifkan opsi Multi-AZ.
- Buat CloudWatch alarm Amazon dan atur topik Amazon Simple Notification Service (Amazon SNS) untuk peringatan.

Untuk validasi tambahan selama fase proof-of-concept (POC), kami merekomendasikan tes tambahan berikut:

- Jalankan tes kinerja untuk memastikan bahwa mereka memenuhi harapan bisnis Anda.
- Uji failover database, pemulihan, dan pemulihan untuk memastikan bahwa Anda memenuhi persyaratan RPO dan RTO.
- Buat daftar semua pekerjaan dan laporan penting, dan jalankan di Amazon RDS untuk mengevaluasi kinerjanya berdasarkan perjanjian tingkat layanan Anda (). SLAs

## Mengoperasikan dan mengoptimalkan basis data Amazon RDS Anda

Saat database aktif AWS, pastikan Anda mengikuti praktik terbaik di berbagai bidang seperti pemantauan, peringatan, cadangan, dan ketersediaan tinggi di cloud. Contoh:

- Siapkan CloudWatch pemantauan, dan aktifkan pemantauan terperinci.
- Gunakan [Amazon RDS Performance](#) Insights dan Agen [Manajemen Oracle Enterprise Manager \(OEM\)](#) untuk memantau database Anda.
- Siapkan peringatan dengan menggunakan topik SNS.
- Siapkan pencadangan otomatis menggunakan [AWS Backup](#) Anda juga dapat menggunakan cadangan Oracle Data Pump atau mengambil snapshot manual.
- Untuk ketersediaan tinggi, siapkan fitur Amazon RDS Multi-AZ.
- Jika Anda membutuhkan database [hanya-baca, siapkan Read Replica dalam yang](#) sama atau Wilayah AWS sesuai dengan kebutuhan Anda.

## AWS Mitra

Migrasi database dapat menjadi proyek yang menantang yang membutuhkan keahlian dan alat. Anda dapat mempercepat migrasi dan waktu menuju hasil melalui kemitraan. [AWS Database Migration Service Mitra Pengiriman](#) memiliki keahlian yang diperlukan untuk membantu pelanggan bermigrasi ke cloud dengan mudah dan aman. Mitra ini memiliki keahlian untuk migrasi homogen, seperti Oracle ke Oracle, dan migrasi heterogen antara platform database yang berbeda, seperti Oracle ke Amazon Aurora atau Microsoft SQL Server ke MySQL.

Berdasarkan persyaratan dan preferensi Anda, Anda dapat menggunakan AWS Partner untuk menangani migrasi lengkap atau untuk membantu hanya beberapa aspek migrasi. Selain itu, Anda dapat menggunakan alat dan solusi yang disediakan oleh AWS Mitra untuk membantu migrasi.

# Sumber daya tambahan

## Posting blog

- [Migrasi Database — Apa yang Perlu Anda Ketahui Sebelum Memulai?](#)
- [Migrasi database Oracle dengan downtime mendekati nol menggunakan AWS DMS](#)
- [Cara Migrasi Database Oracle Anda ke PostgreSQL](#)
- [Cara Memigrasi Database Oracle Anda ke Amazon Aurora](#)
- [Cara Migrasi Oracle Workloads ke Cloud VMware AWS](#)
- [Praktik Terbaik untuk Virtualisasi Oracle RAC dengan Cloud on VMware AWS](#)

## AWS dokumentasi

- [Amazon Aurora](#)
- [Amazon EC2](#)
- [Amazon RDS](#)
- [Amazon RDS Kustom](#)
- [Amazon Redshift](#)
- [AWS DMS](#)
- [AWS SCT](#)
- [Menggunakan Oracle GoldenGate dengan Amazon RDS](#)
- [Lisensi Oracle](#)

## Informasi tambahan

- [Pompa Data Oracle](#)
- [Penjaga Data Oracle](#)
- [Ekspor dan Impor Oracle](#)
- [Oracle GoldenGate](#)
- [Oracle RMAN](#)
- [Pengembang Oracle SQL](#)
- [Oracle SQL\\* Pemuat](#)

- [Melisensikan Perangkat Lunak Oracle di Lingkungan Cloud Computing](#)
- [VMware HCX](#)

# Lampiran: Kuesioner migrasi Oracle

Gunakan kuesioner di bagian ini sebagai titik awal untuk mengumpulkan informasi untuk fase penilaian dan perencanaan proyek migrasi Anda. Anda dapat [mengunduh kuesioner ini](#) dalam format Microsoft Excel dan menggunakannya untuk merekam informasi Anda.

## Informasi umum

1. Apa nama database Oracle Anda?
2. Apa versi database Oracle Anda?
3. Apa edisi database: Standar atau Perusahaan?
4. Berapa ukuran database Anda?
5. Apa set karakter database?
6. Apa zona waktu database?
7. Berapa rata-rata dan maksimum transaksi I/O per detik (TPS)?
8. Apa IOPS (rata-rata dan maksimum) untuk database ini untuk operasi baca/tulis?
9. Berapa pembuatan redo log per jam (rata-rata dan maksimum) per hari?
10. Berapa banyak skema yang Anda rencanakan untuk bermigrasi?
11. Berapa ukuran masing-masing skema?
12. Berapa banyak tabel besar (lebih dari 100 GB) yang Anda miliki per skema?
13. Bisakah Anda mengarsipkan tabel yang tidak perlu dimigrasi?
14. Berapa ukuran sistem global area (SGAs) dan program global area (PGAs) atau penggunaan Automatic Memory Management (AMM), dalam megabyte?
15. Berapa banyak tabel yang dimiliki LOBs? Berapa ukuran maksimum LOBs?
16. Apakah semua tabel Anda LOBs memiliki kunci utama?
17. Apakah Anda memiliki tautan basis data yang mengarah ke database lain?
18. Apa persyaratan SLA untuk database Anda?
19. Apa persyaratan RTO dan RPO untuk database Anda?
20. Berapa banyak downtime database yang dapat Anda izinkan untuk tujuan migrasi?
21. Apakah Anda memiliki persyaratan kepatuhan, peraturan, atau audit?

## Infrastruktur

1. Apa nama host dari database?
2. Apa sistem operasi yang digunakan untuk database ini?
3. Berapa banyak core CPU yang dimiliki server?
4. Berapa ukuran memori di server?
5. Apakah Anda menggunakan penyimpanan lokal?
6. Apakah Anda menggunakan jenis penyimpanan network-attached storage (NAS) atau storage area network (SAN)?
7. Apakah Anda memiliki database RAC? Jika ya, berapa banyak node yang dimilikinya?
8. Apakah Anda menggunakan fitur partisi?
9. Apakah Anda menggunakan Oracle Spatial?
10. Apakah Anda memiliki database multi-tenant?

## Pencadangan basis data

1. Bagaimana Anda membuat cadangan database Anda? Seberapa sering?
2. Berapa periode retensi Anda untuk log arsip dan cadangan?
3. Apakah Anda menggunakan backup untuk mengkloning database Anda?
4. Di mana Anda menyimpan cadangan Anda?

## Keamanan basis data

1. Apakah Anda menggunakan Oracle Database Vault?
2. Apakah Anda menggunakan data masking?
3. Apakah Anda menggunakan Secure Sockets Layer (SSL)?
4. Apakah Anda menggunakan fitur Oracle Advanced Security seperti Transparent Data Encryption (TDE)?
5. Apakah Anda menggunakan Oracle Advanced Compression?

## Database ketersediaan tinggi dan pemulihan bencana

1. Apa persyaratan ketersediaan tinggi Anda?
2. Apakah Anda menggunakan Oracle Data Guard? Di mana wilayah basis data utama dan siaga Anda?
3. Apakah Anda menggunakan Oracle Active Data Guard?
4. Apakah Anda menggunakan alias Domain Name System (DNS) untuk konektivitas database?
5. Apakah Anda menggunakan alat replikasi seperti Oracle GoldenGate, Quest SharePlex, atau Oracle Streams?

## 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="#">Informasi yang diperbarui</a>	Memperbarui edisi Oracle yang didukung oleh <a href="#">Amazon RDS Custom</a> untuk Oracle.	Agustus 16, 2024
<a href="#">Bagian yang dihapus</a>	Menghapus informasi tentang Kerangka Kualifikasi AWS Beban Kerja (AWS WQF).	Juli 20, 2023
<a href="#">Bagian yang dihapus</a>	Menghapus informasi tentang CloudEndure Migrasi, yang sedang dihentikan. <a href="#">AWS Application Migration Service</a> adalah layanan migrasi utama yang direkomendasikan untuk lift-and-shift migrasi ke AWS Cloud	September 23, 2022
<a href="#">Bagian yang ditambahkan</a>	Menambahkan informasi tentang <a href="#">memigrasi database Oracle ke Amazon RDS Custom</a> .	30 Juni 2022
<a href="#">Bagian yang diperbarui</a>	Memperbarui bagian CloudEndure Migrasi dengan informasi terbaru tentang ketersediaan produk.	10 Mei 2022
<a href="#">Informasi AWS WQF yang diperbarui</a>	Memperbarui bagian AWS WQF dengan dukungan	16 Oktober 2020

terbaru dan informasi  
ketersediaan.

Bagian yang ditambahkan

Memperbarui [strategi migrasi database Oracle](#) dengan informasi tambahan, menambahkan [praktik terbaik untuk bermigrasi ke Amazon RDS](#), dan menambahkan [kuesioner](#) untuk penilaian dan perencanaan migrasi.

16 Maret 2020

Publikasi awal

—

24 Februari 2020

# 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 — Memindahkan aplikasi dan memodifikasi 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 PostgreSQL Compatible Edition.
- Replatform (angkat dan bentuk ulang) — Pindahkan aplikasi ke cloud, dan perkenalkan beberapa tingkat pengoptimalan untuk memanfaatkan kemampuan cloud. Contoh: Migrasikan 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

### 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.

### 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.

deployment biru/hijau

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 Well-Architected AWS .

## 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.

Enkripsi sisi klien

Enkripsi data secara lokal, sebelum target Layanan AWS menerimanya.

## Pusat Keunggulan Cloud (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 CCo E](#) 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 CCo E, 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 Perusahaan. AWS Cloud 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. Pipa CI/CD tunggal 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 Wilayah, 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.

## defense-in-depth

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, defense-in-depth pendekatan 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 basis data](#).

## desain berbasis domain

Pendekatan untuk mengembangkan sistem perangkat lunak yang kompleks dengan menghubungkan komponennya ke domain yang berkembang, atau tujuan bisnis inti, yang dilayani oleh 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. Sistem big-endian menyimpan byte paling signifikan terlebih dahulu. Sistem little-endian 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. Beberapa bidikan dapat efektif untuk tugas-tugas yang memerlukan 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. FMs 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](#).

## 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 .

# 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.

#### 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.

|

#### 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](#) di AWS Well-Architected Framework.

## 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 inbound, outbound, dan inspeksi VPCs 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 lebih lanjut, 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 VPCs (dalam yang sama atau berbeda Wilayah AWS), internet, dan jaringan lokal. [Arsitektur Referensi AWS Keamanan](#) merekomendasikan pengaturan akun Jaringan Anda dengan inbound, outbound, dan inspeksi VPCs 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 dalam bahasa lain, dan menyelesaikan kalimat. Untuk informasi lebih lanjut, lihat [Apa itu LLMs](#).

### 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 dan pembelajaran pola. 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](#).

## 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 lebih lanjut, lihat [Membangun mekanisme](#) di AWS Well-Architected Framework.

## 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 ringan machine-to-machine \(M2M\), berdasarkan pola terbitkan/berlangganan, untuk perangkat IoT yang dibatasi sumber daya.](#)

## layanan mikro

Layanan kecil dan independen yang berkomunikasi dengan jelas APIs 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 ringan. APIs 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 dan pelajaran terbaik 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

Tim lintas fungsi 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 [Mengurai 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 machine-to-machine (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 lebih lanjut, lihat [Ulasan Kesiapan Operasional \(ORR\)](#) dalam Kerangka Kerja Well-Architected AWS .

## 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 dan permintaan ke bucket S3. PUT DELETE

## 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 inbound, outbound, dan inspeksi VPCs 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 di organisasi \(lihat kebijakan kontrol layanan\). AWS Organizations](#)

## ketekunan 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 yang dihosting pribadi

Container yang menyimpan informasi tentang bagaimana Anda ingin Amazon Route 53 merespons kueri DNS untuk domain dan subdomainnya dalam satu atau lebih VPCs. 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 pengidentifikasi 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). Tipe 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](#) merujuk sumber data otoritatif yang berada di luar sumber data pelatihannya sebelum menghasilkan respons. Misalnya, model RAG mungkin melakukan

penemuan 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 SCPs menentukan pagar pembatas atau menetapkan batasan pada tindakan yang dapat didelegasikan oleh administrator kepada pengguna atau peran. Anda dapat menggunakan SCPs 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](#).

## 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](#).

## split-and-seed model

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 Microsoft ASP.NET \(ASMX\) 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

Pasangan nilai kunci yang bertindak sebagai metadata untuk mengatur sumber daya Anda. AWS Tanda membantu Anda mengelola, mengidentifikasi, mengatur, dan memfilter sumber daya. Untuk informasi selengkapnya, lihat [Menandai AWS sumber daya Anda](#).

## 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.

### gerbang transit

Hub transit jaringan yang dapat Anda gunakan untuk menghubungkan jaringan Anda VPCs dan lokal. 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. Untuk informasi lebih lanjut, lihat panduan [Mengukur ketidakpastian dalam sistem pembelajaran mendalam](#).

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 VPCs yang memungkinkan Anda untuk 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.

## bidikan zero-shot

Memberikan [LLM](#) dengan instruksi untuk melakukan tugas tetapi tidak ada contoh (tembakan) 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.