



Migrasi database Microsoft SQL Server ke Cloud AWS

AWS Bimbingan Preskriptif



AWS Bimbingan Preskriptif: Migrasi database Microsoft SQL Server ke Cloud AWS

Copyright © 2024 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 para pelanggan, atau dengan cara apa pun yang menghina atau mendiskreditkan Amazon. Semua merek dagang lain yang tidak dimiliki oleh Amazon merupakan properti dari masing-masing pemilik, yang mungkin berafiliasi, terkait dengan, atau disponsori oleh Amazon, atau tidak.

Table of Contents

Pengantar	1
Gambaran Umum	1
Strategi migrasi	4
Memilih strategi migrasi yang tepat	5
Migrasi online dan offline	6
Metode migrasi	8
Pencadangan/pemulihan SQL Server asli	15
Pengiriman log	17
Pencerminan basis data	18
Selalu Aktif pada grup ketersediaan	19
Grup ketersediaan terdistribusi	20
Replikasi transaksional	21
Orkestrator AWS Migration Hub	23
AWS SnowballTepi	25
Migrasi basis data homogen	26
Amazon RDS for SQL Server	27
Kapan memilih Amazon RDS	28
Ketersediaan tinggi	29
Replika baca	30
Pemulihan bencana	33
Amazon RDS Custom for SQL Server	33
Kapan memilih Amazon RDS Custom untuk SQL Server	33
Cara kerjanya	34
Amazon EC2 untuk SQL Server	37
Kapan memilih Amazon EC2	37
Ketersediaan tinggi	38
Pemulihan bencana	46
VMware Cloud aktif AWS untuk SQL Server	47
Kapan memilih VMware Cloud di AWS	48
Migrasi database heterogen	49
Alat	51
AWS SCT	51
AWS DMS	52
Ikan babel	52

Skenario migrasi hibrida	55
Mencadangkan database Anda ke cloud	55
Memperluas ketersediaan tinggi dan solusi pemulihan bencana	55
Storage Gateway	56
Menggunakan AWS DMS dan AWS SCT	57
Memodernasikan basis data SQL Server Anda	58
Migrasikan beban kerja SQL Server Anda dari Windows ke Linux	58
Ketersediaan tinggi di Linux	59
AWS Launch Wizard	60
Praktik terbaik untuk bermigrasi ke Amazon RDS untuk SQL Server	64
Menyediakan basis data target Anda	64
Mencadangkan dari database sumber Anda	65
Mentransfer file dump data ke AWS	65
Memulihkan data ke database target Anda	65
Langkah pasca-migrasi	66
Menguji migrasi	66
Mengoperasikan dan mengoptimalkan database Amazon RDS Anda	67
Memilih antara Amazon EC2 dan Amazon RDS	68
Matriks keputusan	68
Tanggung Jawab Bersama	95
Pola migrasi database SQL Server	97
Mitra	98
Sumber daya tambahan	99
Ucapan Terima Kasih	100
Lampiran: Kuesioner migrasi basis data SQL Server	101
Informasi umum	101
infrastruktur	102
Backup database	102
Fitur database	102
Keamanan database	102
Ketersediaan basis data dan pemulihan bencana	103
Riwayat dokumen	104
Glosarium	107
#	107
A	108
B	111

C	112
D	115
E	119
F	121
G	122
H	123
I	124
L	127
M	128
O	131
P	133
Q	136
R	136
D	139
T	142
U	144
V	144
W	145
Z	146
.....	cxlvii

Migrasi database Microsoft SQL Server ke Cloud AWS

Sagar Patel, Amazon Web Services (AWS)

April 2024 ([riwayat dokumen](#))

Amazon Web Services (AWS) menyediakan serangkaian layanan dan alat yang komprehensif untuk menyebarkan database Microsoft SQL Server pada infrastruktur Cloud yang andal dan aman AWS. Manfaat menjalankan SQL Server AWS termasuk penghematan biaya, skalabilitas, ketersediaan tinggi dan pemulihan bencana, kinerja yang lebih baik, dan kemudahan manajemen. Untuk informasi selengkapnya, lihat [Pelajari AWS mengapa cloud terbaik untuk menjalankan beban kerja Microsoft Windows Server dan SQL Server](#) di blog AWS Compute.

Panduan ini menjelaskan opsi yang tersedia untuk memigrasikan database SQL Server dari lokal ke AWS Cloud, ke Amazon Relational Database Service (Amazon RDS), Amazon Elastic Compute Cloud (Amazon EC2), atau VMware Cloud on AWS. Ini menyelami praktik dan rekomendasi terbaik untuk menggunakan opsi migrasi ini. Ini juga menyediakan informasi tentang cara mengatur ketersediaan tinggi dan solusi pemulihan bencana antara lingkungan SQL Server lokal dan AWS, menggunakan fitur SQL Server asli seperti pengiriman log, replikasi, dan grup ketersediaan Selalu Aktif.

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

Gambaran Umum

Sebelum Anda memigrasikan database SQL Server ke 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 dan beban kerja database SQL Server Anda dengan memahami kompleksitas, kompatibilitas, dan biaya migrasi. Berikut adalah beberapa poin teratas yang harus Anda pertimbangkan ketika Anda berencana untuk bermigrasi:

- Ukuran basis data — Periksa ukuran saat ini dan pertumbuhan kapasitas keseluruhan database Anda. Misalnya, jika Anda berencana untuk memigrasikan database SQL Server ke Amazon RDS atau Amazon RDS Custom, Anda dapat membuat instans DB dengan penyimpanan hingga 16 TiB.

Anda dapat meminta lebih banyak penyimpanan dengan [membuka tiket dukungan dengan AWS Support](#). Untuk informasi terbaru, lihat [Penyimpanan instans Amazon RDS DB](#) di dokumentasi Amazon RDS.

- IOPS — Tentukan IOPS dan throughput database Anda. Jika Anda berencana untuk bermigrasi ke Amazon RDS, pertimbangkan kinerja [I/O instans Amazon RDS DB](#).
- Dependensi - Periksa dependensi database saat ini. Jika database Anda bergantung pada database lain, Anda dapat memigrasikan mereka bersama-sama atau membuat dependensi setelah Anda memigrasikan database utama Anda.

Jika database Anda mendukung aplikasi lama, kustom, atau paket, Amazon RDS Custom for SQL Server mungkin merupakan pilihan yang baik. Layanan ini memungkinkan Anda mempertahankan kontrol atas konfigurasi database, sistem file bersama, dan tambalan sistem operasi.

Inventarisasi semua dependensi SQL Server. Cari tahu server web mana (misalnya, server pelaporan atau server intelijen bisnis) antarmuka dengan SQL Server. Ketika tiba waktunya untuk bermigrasi, informasi ini membantu Anda menentukan apa yang akan terpengaruh dan bagaimana Anda dapat meminimalkan dampaknya.

- Kepatuhan - Tinjau arsitektur dan kebutuhan audit atau kepatuhan Anda saat ini, untuk memastikan Anda dapat memenuhi persyaratan ini setelah pindah ke Amazon RDS atau Amazon EC2.
- HA/DR - Apakah Anda memerlukan ketersediaan tinggi (HA) dan kemampuan failover otomatis? Jika Anda menjalankan beban kerja produksi, ketersediaan tinggi dan pemulihan bencana (DR) direkomendasikan praktik terbaik.

Pahami persyaratan HA/DR Anda untuk menentukan apakah Anda memerlukan arsitektur Multi-wilayah. Jika demikian, migrasi database SQL Server Anda ke Amazon EC2. Amazon RDS tidak mendukung konfigurasi Multi-wilayah.

- Dukungan versi - [Periksa versi dan edisi perangkat lunak SQL Server Anda jika Anda berencana untuk pindah ke Amazon RDS for SQL Server \(lihat versi yang saat ini didukung untuk Amazon RDS dan Amazon RDS\)](#).
- Konektivitas jaringan — 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.
- Waktu henti migrasi — Tentukan jumlah waktu henti yang tersedia untuk migrasi sehingga Anda dapat merencanakan pendekatan migrasi dan memutuskan apakah Anda ingin menggunakan migrasi online atau offline.

- Persyaratan RTO, RPO, SLA - Identifikasi persyaratan tujuan waktu pemulihan (RTO), tujuan titik pemulihan (RPO), dan perjanjian tingkat layanan (SLA) Anda untuk beban kerja database Anda yang ada.
- Lisensi — Memahami opsi lisensi Anda. Anda dapat memilih opsi yang disertakan lisensi di Amazon EC2 dan Amazon RDS, atau [memilih untuk membawa lisensi Anda](#) sendiri (BYOL) di Amazon EC2.
- Dukungan fitur - Identifikasi fitur database dan fungsionalitas yang digunakan aplikasi Anda, apakah itu dikembangkan sendiri atau perangkat lunak commercial-off-the-shelf (COTS). Informasi ini dapat membantu Anda menentukan apakah Anda dapat mengurangi biaya lisensi dengan beralih dari edisi SQL Server Enterprise ke edisi Standar. Namun, tinjau batasan sumber daya edisi Standar sebelum Anda beralih. Misalnya, edisi Standar hanya mendukung 128 GB RAM.

Apakah beban kerja Anda sesuai dengan fitur dan kemampuan yang ditawarkan oleh Amazon RDS for SQL Server? Untuk informasi selengkapnya, lihat [fitur SQL Server di Amazon RDS](#). Jika Anda memerlukan fitur yang tidak didukung, bermigrasi ke Amazon EC2 adalah pilihan.

Strategi migrasi database SQL Server

Pada tingkat tinggi, ada dua opsi untuk memigrasi database SQL Server dari tempat ke AWS Cloud: tetap menggunakan SQL Server ([migrasi homogen](#)) atau [memindahkan SQL Server \(migrasi heterogen\)](#). Dalam migrasi homogen, Anda tidak mengubah mesin database. Artinya, database target Anda juga merupakan database SQL Server. Dalam migrasi heterogen, Anda mengalihkan database SQL Server Anda ke mesin database sumber terbuka seperti MySQL, PostgreSQL, atau MariaDB, atau ke database AWS Cloud-native seperti Amazon Aurora, Amazon DynamoDB, atau Amazon Redshift.

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

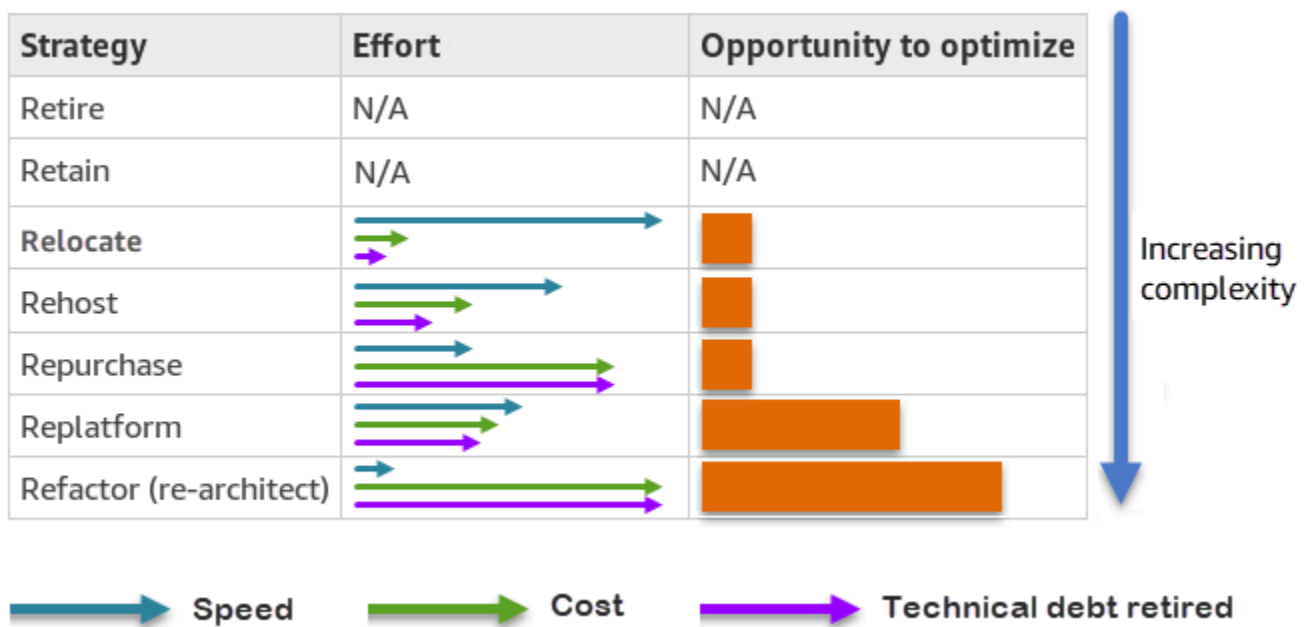
Strategi	Tipe	Kapan harus memilih	Contoh
Rehost	Homogen	Anda ingin memigrasi database SQL Server Anda sebagaimana adanya, dengan atau tanpa mengubah sistem operasi, perangkat lunak database, atau konfigurasi.	SQL Server ke Amazon EC2 (Telusuri pola rehost)
Replatform	Homogen	Anda ingin mengurangi waktu yang Anda habiskan untuk mengelola instance database dengan menggunakan penawaran database yang dikelola sepenuhnya.	SQL Server ke Amazon RDS untuk SQL Server (Telusuri pola Replatform)

Strategi	Tipe	Kapan harus memilih	Contoh
Re-arsitek (refactor)	Heterogen	Anda ingin merestrukturisasi, menulis ulang, dan merancang ulang database dan aplikasi Anda untuk memanfaatkan fitur database open-source dan cloud-native.	SQL Server ke Amazon Aurora PostgreSQL, MySQL, atau MariaDB Telusuri pola Re-arsitek)

Jika Anda mencoba memutuskan antara rehosting atau replatforming database SQL Server Anda, lihat [Memilih antara Amazon EC2 dan Amazon RDS](#) nanti dalam panduan ini untuk perbandingan fitur yang didukung. side-by-side

Memilih strategi migrasi yang tepat

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



Refactoring database SQL Server Anda dan migrasi ke database open-source atau AWS Cloud-native seperti Amazon Aurora PostgreSQL-compatible Edition atau Aurora MySQL Compatible Edition dapat membantu Anda memodernisasi dan mengoptimalkan database Anda. Dengan pindah ke database sumber terbuka, Anda dapat menghindari lisensi mahal (sehingga biaya lebih rendah), periode penguncian vendor, dan audit. Namun, tergantung pada kompleksitas beban kerja Anda, refactoring database SQL Server Anda dapat menjadi upaya yang rumit, memakan waktu, dan sumber daya intensif.

Untuk mengurangi kompleksitas, alih-alih memigrasi database Anda dalam satu langkah, Anda mungkin mempertimbangkan pendekatan bertahap. Pada tahap pertama, Anda dapat fokus pada fungsi database 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 SQL Server lokal dengan Aurora MySQL yang kompatibel, Anda dapat mempertimbangkan untuk menghosting ulang database Anda di Amazon EC2 atau replatforming database Anda di Amazon RDS untuk SQL Server pada fase pertama, dan kemudian refactor ke Aurora MySQL yang kompatibel pada fase berikutnya. Pendekatan ini membantu mengurangi biaya, sumber daya, dan risiko selama fase migrasi dan berfokus pada optimalisasi dan modernisasi pada fase kedua.

Migrasi online dan offline

Anda dapat menggunakan dua metode untuk memigrasi database SQL Server dari lingkungan cloud lokal atau lainnya ke AWS Cloud, berdasarkan timeline migrasi dan berapa banyak waktu henti yang dapat Anda izinkan: migrasi offline atau migrasi online.

- **Migrasi offline:** Metode ini digunakan saat aplikasi Anda mampu melakukan downtime yang direncanakan. Dalam migrasi offline, database sumber offline selama periode migrasi. Sementara database sumber offline, itu dimigrasi 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 dimigrasi 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 langsung. Selama cutover, aplikasi beralih koneksi ke database target pada AWS,

meninggalkan tidak ada koneksi ke database sumber. Anda dapat menggunakan AWS Database Migration Service (AWS DMS) atau alat yang tersedia dari [AWS Marketplace](#)(seperti Attunity) untuk menyinkronkan database sumber dan target.

Metode migrasi database SQL Server

Ada berbagai metode untuk memigrasikan database SQL Server Anda ke. AWS Anda dapat memilih dari AWS layanan dan fitur asli SQL Server berdasarkan penilaian dan persyaratan Anda. Bagian ini menjelaskan beberapa metode yang paling umum, yang dirangkum dalam dua tabel berikut. Diskusi terperinci tentang beberapa metode ini disertakan dalam bagian di Amazon EC2 dan Amazon RDS nanti dalam panduan ini.

AWS layanan

Metode migrasi	Target	Fitur dan keterbatasan	Informasi lain
AWS DMS	Amazon EC2 Amazon RDS Kustom Amazon RDS Amazon Aurora	<ul style="list-style-type: none"> • Mendukung beban penuh dan CDC • Mendukung semua ukuran database 	Bagian AWS DMS
Orkestrator AWS Migration Hub	Amazon EC2 Amazon RDS	<ul style="list-style-type: none"> • Menyediakan templat step-by-step alur kerja yang telah ditentukan • Mengotomatiskan pencadangan dan pemulihan asli • Mendukung semua edisi dan versi SQL Server • Dapat diterapkan pada satu atau banyak database sekaligus • Mendukung semua ukuran database 	Bagian Orkestrator AWS Migration Hub

Metode migrasi	Target	Fitur dan keterbatasan	Informasi lain
AWS Application Migration Service	Amazon EC2	<ul style="list-style-type: none"> • lift-and-shift Solusi yang sangat otomatis • Replikasi tingkat blok berbasis agen 	Tidak tercakup dalam panduan ini (lihat dokumentasi Layanan Migrasi Aplikasi)
AWS SnowballTepi	Amazon EC2 Amazon RDS Kustom Amazon RDS	<ul style="list-style-type: none"> • Mendukung database yang sangat besar (hingga 210 TB) • Menggunakan Amazon Simple Storage Service (Amazon S3) untuk menyimpan dan memulihkan data 	Bagian Snowball Edge

Metode asli SQL Server

Metode migrasi	Target	Fitur dan keterbatasan	Informasi lain
Pencadangan dan pemulihan asli	Amazon EC2 Amazon RDS Kustom Amazon RDS	<ul style="list-style-type: none"> • Dapat diterapkan pada satu atau banyak database sekaligus • Membutuhkan downtime • Mendukung semua ukuran database 	Bagian pencadangan/pemulihan SQL Server asli (Anda dapat menggunakan Orkestrator AWS Migration Hub untuk mengotomatiskan pencadangan dan pemulihan asli)

Metode migrasi	Target	Fitur dan keterbatasan	Informasi lain
Pengiriman log	Amazon EC2 Amazon RDS Kustom Amazon RDS	<ul style="list-style-type: none">• Diterapkan per database• Bisa ditunda	Bagian pengiriman log
Pencerminan basis data	Amazon EC2	<ul style="list-style-type: none">• Diterapkan per database• Dapat sinkron atau asinkron, berdasarkan edisi SQL Server• Database sekunder tidak dapat dibaca; itu bertindak sebagai siaga• Mendukung failover otomatis dan manual	Bagian pencerminan basis data

Metode migrasi	Target	Fitur dan keterbatasan	Informasi lain
Selalu Aktif pada grup ketersediaan	Amazon EC2 Kustom Amazon RDS	<ul style="list-style-type: none">• Diterapkan ke satu set database pengguna• Bisa sinkron atau asinkron• Database sekunder dapat dibaca (edisi SQL Server Enterprise saja)• Mendukung failover otomatis dan manual• Failover dapat dimulai untuk beberapa database sekaligus, pada tingkat grup database	Selalu Di bagian grup ketersediaan

Metode migrasi	Target	Fitur dan keterbatasan	Informasi lain
Grup ketersediaan Basic Always On	Amazon EC2 Kustom Amazon RDS	<ul style="list-style-type: none">• Didukung dalam edisi Standar SQL Server• Diterapkan ke database pengguna tunggal per grup ketersediaan• Bisa sinkron atau asinkron• Mendukung failover otomatis dan manual• Failover dapat dimulai pada tingkat grup ketersediaan• Dapat digunakan sebagai lingkungan hibrida antara di tempat dan AWS	Tidak tercakup dalam panduan ini (lihat grup ketersediaan Dasar Selalu Aktif untuk satu database dalam dokumentasi Microsoft)

Metode migrasi	Target	Fitur dan keterbatasan	Informasi lain
Grup ketersediaan terdistribusi	Amazon EC2 Kustom Amazon RDS	<ul style="list-style-type: none">• Dapat digunakan untuk penyebaran SQL Server Multi-wilayah• Dapat gagal ke versi SQL Server yang lebih baru• Tidak memerlukan Windows Server Failover Clustering (WSFC) untuk diperluas ke lingkungan target AWS• Dapat digunakan antara database SQL Server berbasis Windows (sumber) dan berbasis Linux (target)• Dapat digunakan sebagai penyebaran SQL Server hybrid antara di tempat dan AWS	Bagian grup ketersediaan terdistribusi

Metode migrasi	Target	Fitur dan keterbatasan	Informasi lain
Replikasi transaksional	Amazon EC2 Amazon RDS Kustom Amazon RDS	<ul style="list-style-type: none"> • Mendukung migrasi satu set objek (tabel, tampilan, prosedur tersimpan) • Mendukung replikasi asinkron dengan data mendekati waktu nyata • Database pelanggan dapat dibaca • Memerlukan pemantauan ketat pekerjaan replikasi SQL Server yang melakukan replikasi 	Bagian replikasi transaksional
Program salinan massal (bcp)	Amazon EC2 Kustom Amazon RDS	<ul style="list-style-type: none"> • Mendukung database kecil • Membutuhkan downtime • Skema dibuat sebelumnya di tempat tujuan • Digunakan untuk memindahkan data, tetapi bukan metadata 	Tidak tercakup dalam panduan ini (lihat Mengimpor dan mengeksport data SQL Server menggunakan metode lain , bagian Salinan massal dalam dokumentasi Amazon RDS)

Metode migrasi	Target	Fitur dan keterbatasan	Informasi lain
Lepaskan dan pasang	Amazon EC2 Kustom Amazon RDS	<ul style="list-style-type: none"> • Tidak diperlukan cadangan • Membutuhkan downtime • Melibatkan menghentikan, melepaskan, menyalin file, dan melampirkan ke Amazon EC2 	Tidak tercakup dalam panduan ini (lihat Detach dan Lampirkan Database dalam dokumentasi Microsoft)
Impor/ekspor	Amazon EC2 Kustom Amazon RDS	<ul style="list-style-type: none"> • Mendukung database kecil • Membutuhkan downtime • Skema dibuat sebelumnya di tempat tujuan • Digunakan untuk memindahkan data, tetapi bukan metadata 	Tidak tercakup dalam panduan ini (lihat Mengimpor dan mengekspor data SQL Server menggunakan metode lain dalam dokumentasi Amazon RDS)

Pencadangan/pemulihan SQL Server asli

Amazon RDS mendukung operasi pencadangan dan pemulihan asli untuk database Microsoft SQL Server menggunakan file cadangan lengkap dan diferensial (.bak). Ini juga mendukung pemulihan diferensial dan opsi pemulihan log pada instans Amazon RDS for SQL Server DB atau instans Amazon EC2 SQL Server, untuk meminimalkan waktu henti aplikasi Anda.

Note

Anda dapat melakukan operasi pemulihan penuh, diferensial, dan log di Amazon RDS for SQL Server. Namun, Anda hanya dapat melakukan pencadangan penuh dan diferensial (bukan cadangan log) saat ini.

Menggunakan file.bak asli adalah cara paling sederhana untuk mencadangkan dan memulihkan database SQL Server. Anda dapat menggunakan metode ini untuk memigrasikan database ke atau dari Amazon RDS. Anda dapat mencadangkan dan memulihkan database tunggal alih-alih seluruh instans DB. Anda juga dapat memindahkan database antara instans Amazon RDS for SQL Server DB.

Saat Anda menggunakan Amazon RDS, Anda dapat menyimpan dan mentransfer file cadangan di Amazon Simple Storage Service (Amazon S3), untuk lapisan perlindungan tambahan untuk pemulihan bencana. Misalnya:

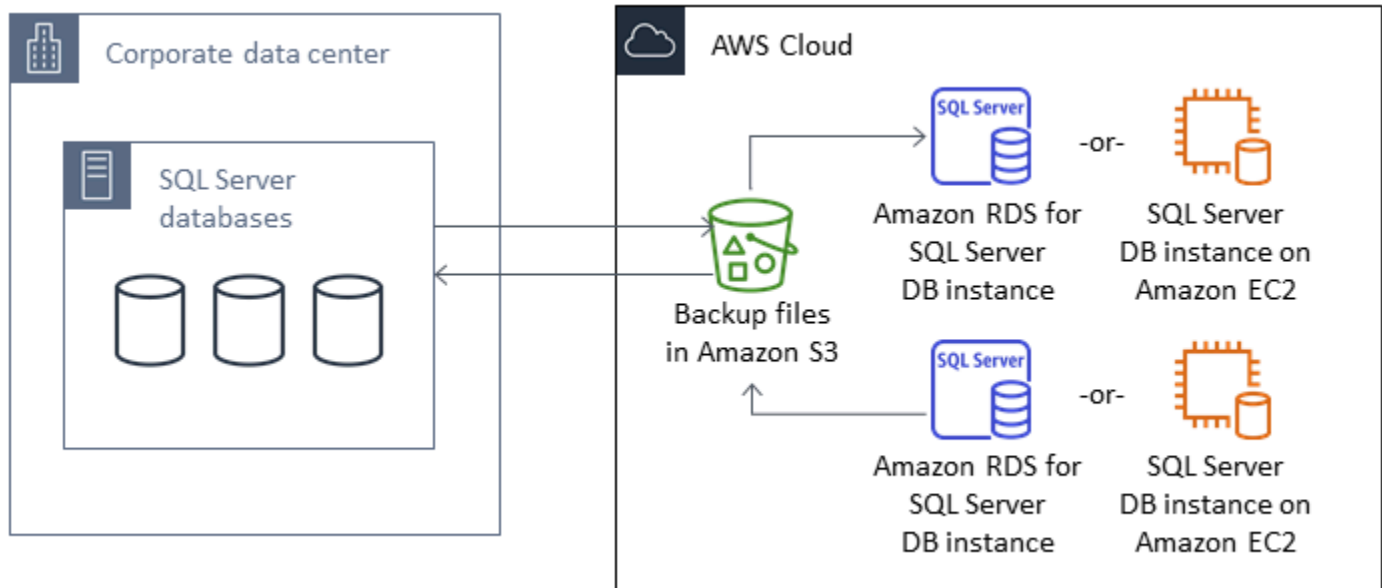
- Anda dapat membuat cadangan lengkap database Anda dari server lokal Anda, menyalinnya ke bucket S3, dan kemudian mengembalikannya ke instans Amazon RDS SQL Server DB yang ada.
- Anda dapat mengambil cadangan dari instans Amazon RDS for SQL Server DB, menyimpannya di Amazon S3, lalu memulihkannya di mana pun Anda inginkan.
- Anda dapat menerapkan aturan konfigurasi [Siklus Hidup Amazon S3](#) untuk mengarsipkan atau menghapus cadangan jangka panjang.

Amazon RDS for SQL Server mendukung pemulihan backup asli SQL Server ke instans SQL Server DB yang telah membaca replika yang dikonfigurasi. Ini berarti Anda tidak perlu menghapus replika baca sebelum memulihkan file cadangan asli ke instans Amazon RDS for SQL Server DB Anda.

Note

Anda dapat menggunakan Migration Hub Orchestrator untuk mengotomatiskan dan mengatur migrasi database SQL Server Anda ke Amazon EC2 atau Amazon RDS dengan menggunakan pencadangan dan pemulihan asli. Untuk informasi lebih lanjut, lihat [Orkestrator AWS Migration Hub bagian](#).

Diagram berikut menunjukkan proses pencadangan/pemulihan SQL Server asli. Anda dapat menggunakan Migration Hub Orchestrator untuk mengotomatiskan proses ini. Anda dapat menggunakan proses ini untuk mencadangkan dan memulihkan database SQL Server ke Amazon EC2 juga.



Untuk mengotomatiskan pencadangan dan pemulihan, lihat dokumentasi [Migration Hub Orchestrator](#).

[Untuk menyiapkan pencadangan/pemulihan asli menggunakan Amazon S3, lihat dokumentasi Amazon RDS.](#)

Untuk batasan saat menggunakan pencadangan dan pemulihan asli SQL Server, lihat [Batasan dan rekomendasi](#) dalam dokumentasi Amazon RDS.

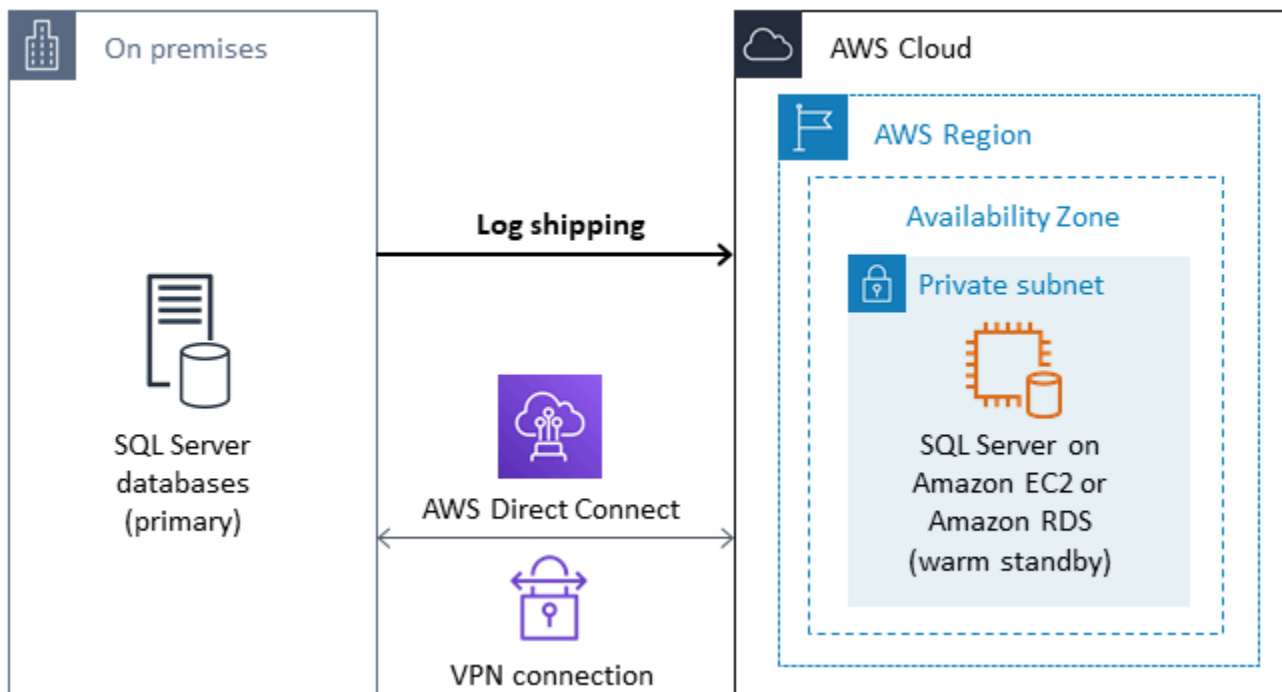
Pengiriman log

Anda dapat menggunakan pengiriman log untuk mengirim cadangan log transaksi dari database SQL Server utama dan lokal ke satu atau beberapa database SQL Server sekunder (siaga hangat) yang digunakan pada instans EC2 atau Amazon RDS for SQL Server DB instans di Cloud. AWS Untuk mengatur pengiriman log di Amazon RDS for SQL Server, Anda harus menggunakan skrip kustom Anda sendiri.

Dalam skenario ini, Anda mengonfigurasi database SQL Server siaga hangat pada instans EC2 atau instans Amazon RDS for SQL Server DB, dan mengirim cadangan log transaksi secara asinkron

antara database lokal dan server siaga hangat di Cloud. AWS Pencadangan log transaksi kemudian diterapkan ke database siaga hangat. Ketika semua log telah diterapkan, Anda dapat melakukan failover manual dan memotong ke cloud.

Opsi ini mendukung semua versi dan edisi SQL Server. Setelah memigrasikan database ke AWS Cloud, Anda dapat menambahkan replika sekunder menggunakan grup ketersediaan Selalu Aktif untuk tujuan ketersediaan dan ketahanan tinggi.



Untuk informasi selengkapnya tentang penggunaan metode ini untuk mencapai ketersediaan tinggi, perlindungan data, dan pemulihan bencana untuk database SQL Server Anda di Amazon EC2, [lihat Log pengiriman di bagian Amazon EC2 untuk SQL Server](#).

Pencerminan basis data

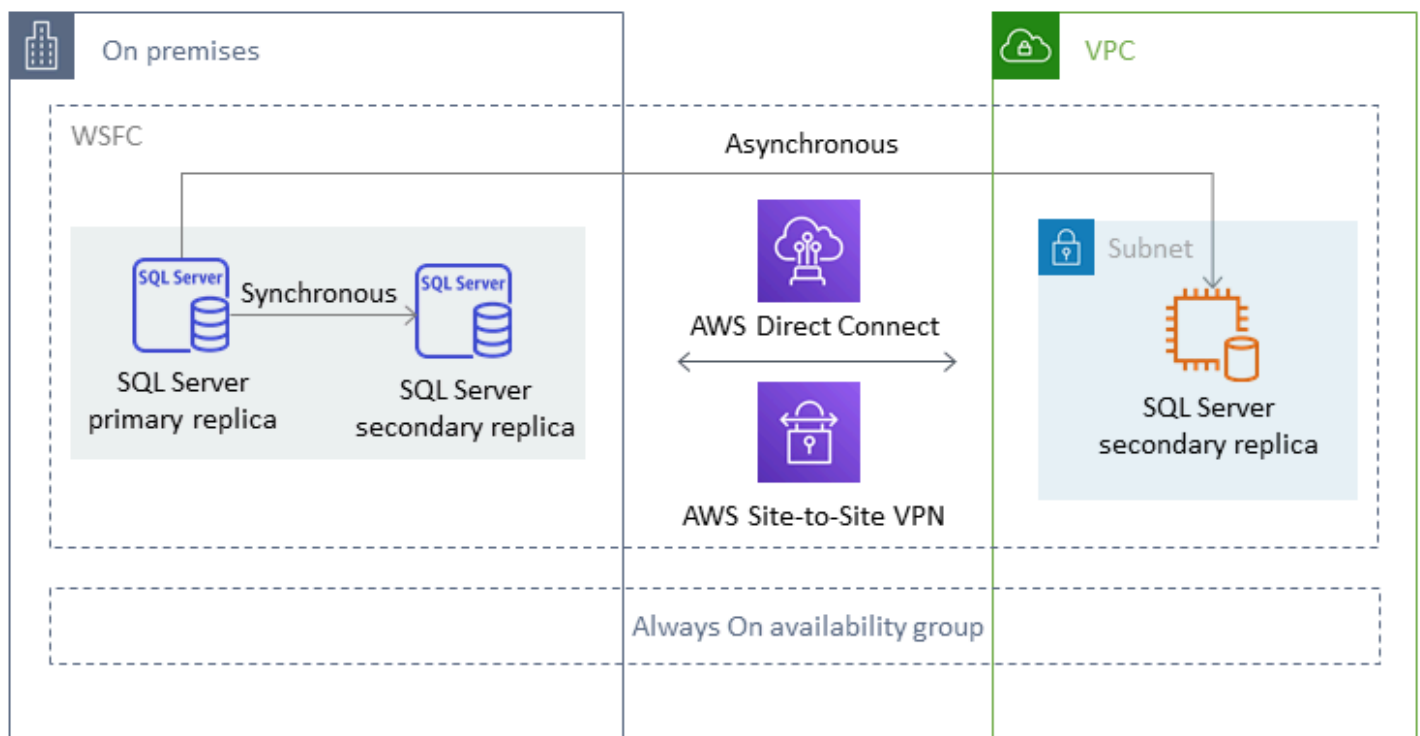
Anda dapat menggunakan mirroring database untuk menyiapkan lingkungan cloud hybrid untuk database SQL Server Anda. Opsi ini memerlukan edisi SQL Server Enterprise. Dalam skenario ini, database SQL Server utama Anda berjalan di tempat, dan Anda membuat siaga hangat di cloud. Anda mereplikasi data Anda secara asinkron, dan melakukan failover manual saat Anda siap untuk cutover. Setelah memigrasikan database ke AWS Cloud, Anda dapat menambahkan replika sekunder menggunakan grup ketersediaan Selalu Aktif untuk tujuan ketersediaan dan ketahanan tinggi.

Untuk informasi selengkapnya tentang penggunaan metode ini untuk mencapai ketersediaan tinggi, perlindungan data, dan pemulihan bencana untuk database SQL Server Anda di Amazon EC2, [lihat Pencerminan basis data](#) di bagian Amazon EC2 untuk SQL Server.

Selalu Aktif pada grup ketersediaan

Grup ketersediaan SQL Server Always On adalah fitur tingkat perusahaan canggih untuk menyediakan solusi ketersediaan dan pemulihan bencana yang tinggi. Fitur ini tersedia jika Anda menggunakan SQL Server 2014 dan versi yang lebih baru. Anda juga dapat menggunakan grup ketersediaan Selalu Aktif untuk memigrasikan database SQL Server lokal Anda ke Amazon EC2. AWS Pendekatan ini memungkinkan Anda untuk memigrasikan database Anda dengan minimal atau tanpa downtime.

Jika Anda memiliki penyebaran lokal grup ketersediaan SQL Server Always On, replika utama dan replika sekunder Anda akan mereplikasi data secara sinkron dalam grup ketersediaan. Jadi, untuk memigrasikan database Anda ke AWS Cloud, Anda dapat memperluas cluster Windows Server Failover Clustering (WSFC) Anda ke cloud. Ini bisa bersifat sementara, hanya untuk tujuan migrasi. Anda kemudian membuat replika sekunder di AWS Cloud dan menggunakan replikasi asinkron, seperti yang ditunjukkan pada diagram berikut. Setelah replika sekunder disinkronkan dengan database lokal utama, Anda dapat melakukan failover manual kapan pun Anda siap untuk cutover.



Untuk informasi selengkapnya tentang penggunaan metode ini untuk mencapai ketersediaan tinggi, perlindungan data, dan pemulihan bencana untuk database SQL Server Anda di Amazon EC2, [lihat grup ketersediaan Selalu Aktif di bagian](#) Amazon EC2 untuk SQL Server.

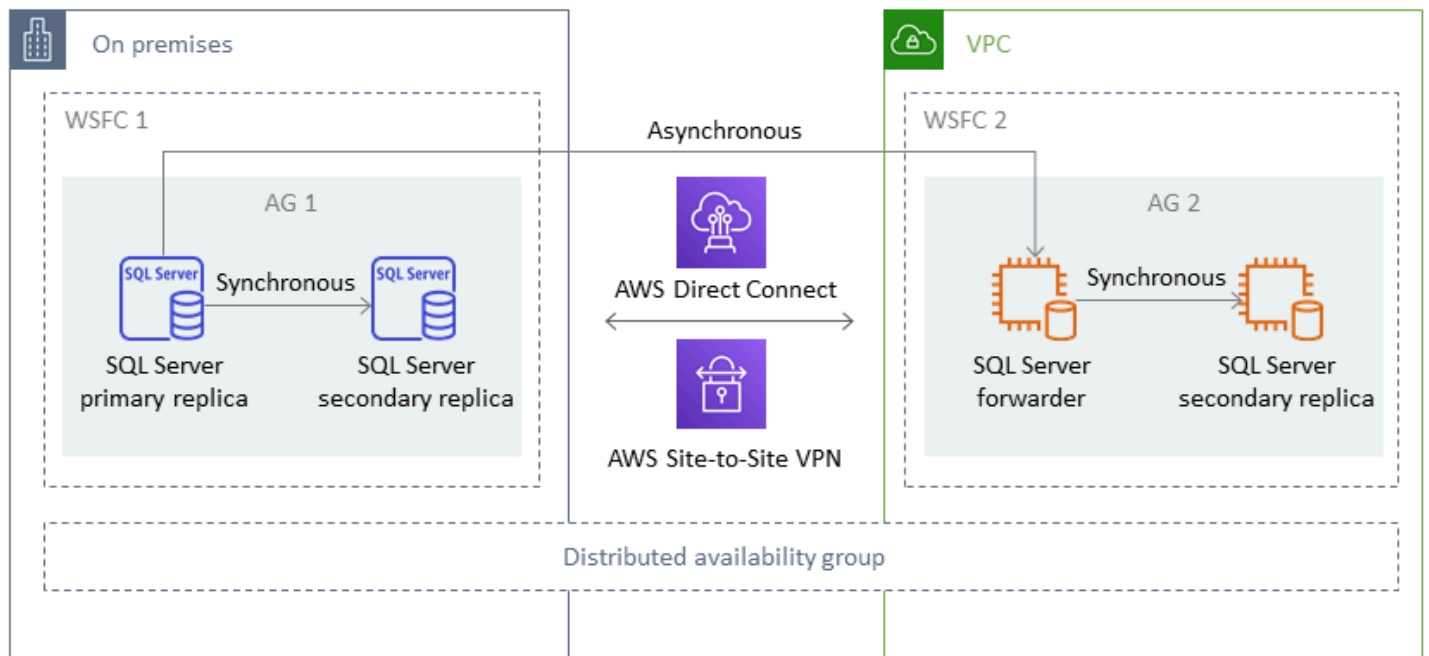
Grup ketersediaan terdistribusi

Grup ketersediaan terdistribusi mencakup dua grup ketersediaan terpisah. Anda dapat menganggapnya sebagai grup ketersediaan grup ketersediaan. Grup ketersediaan yang mendasarinya dikonfigurasi pada dua cluster WSFC yang berbeda. Grup ketersediaan yang berpartisipasi dalam grup ketersediaan terdistribusi tidak perlu berbagi lokasi yang sama. Mereka bisa fisik atau virtual, di tempat atau di cloud publik. Grup ketersediaan dalam grup ketersediaan terdistribusi tidak harus menjalankan versi SQL Server yang sama. Instans DB target dapat menjalankan versi SQL Server yang lebih baru daripada instance DB sumber.

Arsitektur grup ketersediaan terdistribusi memberi Anda cara yang fleksibel untuk meng-host ulang instance atau database SQL Server yang penting. AWS Ini memberikan solusi hibrida untuk mengangkat dan menggeser (atau mengangkat dan mengubah) basis data SQL Server kritis Anda. AWS

Menggunakan arsitektur grup ketersediaan terdistribusi lebih efisien daripada memperluas kluster WFSC lokal yang ada. AWS Data ditransfer hanya dari primer lokal ke salah satu AWS replika (forwarder). Forwarder bertanggung jawab untuk mengirim data ke replika baca sekunder lainnya di. AWS

Dalam diagram berikut, kluster WSFC pertama (WSFC 1) di-host di tempat dan memiliki grup ketersediaan lokal (AG 1). Cluster WSFC kedua (WSFC 2) di-host AWS dan memiliki grup AWS ketersediaan (AG 2). [AWS Direct Connect](#) digunakan sebagai koneksi jaringan khusus antara lingkungan lokal dan AWS. Grup ketersediaan lokal (AG 1) memiliki dua replika (node). Transfer data antar node sinkron, dengan failover otomatis. Demikian pula, grup AWS ketersediaan (AG 2) juga memiliki dua replika, dan transfer data di antara mereka sinkron dengan failover otomatis. Grup ketersediaan terdistribusi membuat database tetap sinkron secara asinkron. Data ditransfer dari replika utama SQL Server di AG 1 (yang ada di tempat) ke replika utama (forwarder) di AG 2 (yang aktif). AWS Forwarder bertanggung jawab untuk mengirim data ke replika baca lainnya AWS dan terus memperbaruinya. Setelah lokal dan AWS database disinkronkan, Anda dapat melakukan failover manual grup ketersediaan terdistribusi. AWS AWSDatabase menjadi database utama untuk akses baca/tulis dari aplikasi.



Note

Pada titik waktu tertentu, hanya ada satu database yang tersedia untuk operasi tulis. Anda dapat menggunakan replika sekunder yang tersisa untuk operasi baca. Untuk mengukur beban kerja baca Anda, Anda dapat menambahkan lebih banyak replika baca di beberapa Availability Zone. AWS

Untuk informasi selengkapnya tentang grup ketersediaan terdistribusi, lihat:

- [Dokumentasi Microsoft SQL Server](#)
- [Cara merancang solusi Microsoft SQL Server hybrid menggunakan grup ketersediaan terdistribusi di blog AWS Database](#)
- [Migrasikan SQL Server untuk AWS menggunakan grup ketersediaan terdistribusi](#) di situs web AWS Prescriptive Guidance

Replikasi transaksional

Replikasi transaksional adalah teknologi SQL Server yang digunakan untuk mereplikasi perubahan antara dua database. Perubahan ini dapat mencakup objek database seperti tabel (kunci utama diperlukan), prosedur tersimpan, tampilan, dan sebagainya, serta data. Proses replikasi melibatkan

penerbit (database utama yang menerbitkan data), pelanggan (database sekunder yang menerima data yang direplikasi), dan distributor (server yang menyimpan metadata dan transaksi untuk replikasi transaksional). Anda dapat menggunakan replikasi transaksional untuk SQL Server di Amazon EC2 dan Amazon RDS untuk instans DB Amazon RDS for SQL Server.

Replikasi transaksional membuat snapshot objek dan data dalam database lokal (publikasi) Anda dan mengirimkannya ke database pelanggan. Setelah snapshot diterapkan ke pelanggan, semua perubahan data berikutnya dan modifikasi skema yang dilakukan di penerbit dikirim ke pelanggan saat terjadi. Perubahan data kemudian terus diterapkan ke pelanggan dalam urutan yang sama seperti yang terjadi di penerbit.

Setelah sinkronisasi selesai, Anda melakukan validasi pada instans SQL Server DB target. Ketika kedua database disinkronkan, Anda menghentikan aktivitas pada database lokal, memastikan replikasi telah selesai, dan kemudian melakukan cutover ke instans SQL Server DB target. Anda kemudian dapat menghentikan langganan push, menghapusnya, dan mulai menggunakan Amazon RDS for SQL Server.

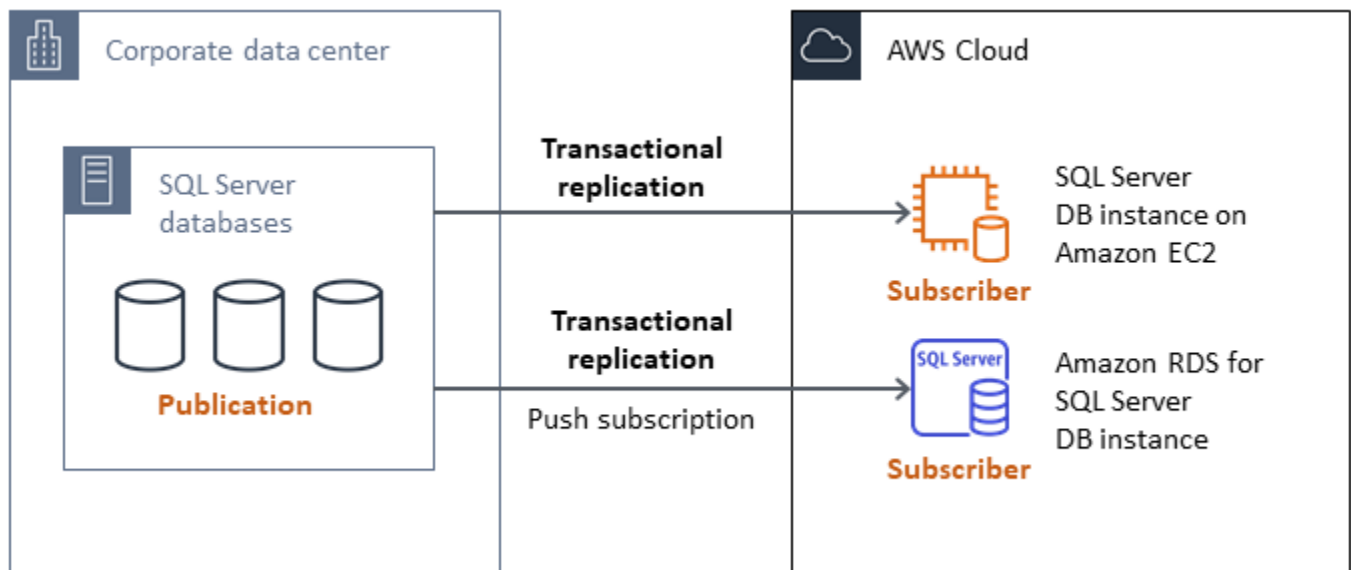
Database pelanggan juga dapat digunakan sebagai database read-only. Distributor, yang mencatat pekerjaan sinkronisasi, direkomendasikan untuk berada di server terpisah. Jika database target Anda ada di Amazon RDS for SQL Server, Anda dapat mengatur langganan push untuk menyebarkan perubahan ke pelanggan.

Kami menyarankan Anda menggunakan replikasi transaksional ketika Anda ingin:

- Lakukan migrasi satu kali data Anda ke Amazon RDS atau Amazon EC2.
- Migrasikan objek tingkat skema atau tingkat tabel ke AWS.
- Migrasikan sebagian database ke AWS.
- Migrasi dengan downtime minimal menggunakan strategi replikasi SQL Server yang ada dengan menambahkan pelanggan tambahan.

Jika Anda berencana menggunakan replikasi transaksional untuk migrasi satu kali data Anda ke Amazon RDS for SQL Server, kami sarankan Anda menyiapkan konfigurasi AZ tunggal untuk replikasi. Setelah proses replikasi selesai, Anda dapat mengubah lingkungan Anda menjadi arsitektur multi-AZ untuk ketersediaan tinggi.

Diagram berikut menunjukkan proses replikasi transaksional untuk database di Amazon RDS dan Amazon EC2.



Untuk informasi selengkapnya tentang replikasi transaksional, lihat [dokumentasi Microsoft SQL Server](#) dan postingan [Cara bermigrasi ke Amazon RDS for SQL Server menggunakan replikasi transaksional](#) di blog Database. AWS

Orkestrator AWS Migration Hub

Orkestrator AWS Migration Hub membantu Anda mengatur dan mengotomatiskan migrasi database SQL Server ke Amazon EC2 atau Amazon RDS. Fitur ini AWS Migration Hub membantu Anda memulai dengan cepat dengan menggunakan templat alur kerja yang telah ditentukan sebelumnya yang dibangun berdasarkan praktik terbaik. Migration Hub Orchestrator mengotomatiskan tugas manual rawan kesalahan yang terlibat dalam proses migrasi, seperti memeriksa kesiapan lingkungan dan koneksi. Anda juga dapat menggunakan Migration Hub Orchestrator untuk mengatur dan mempercepat migrasi untuk aplikasi.NET, beban kerja SAP, dan gambar mesin virtual, selain database SQL Server Anda. Anda dapat mengakses alat ini melalui konsol [Migration Hub Orchestrator](#).

Untuk migrasi SQL Server, Migration Hub Orchestrator mendukung tiga kasus penggunaan:

- Rehost SQL Server di Amazon EC2. Anda dapat memilih server SQL tertentu dan meng-host ulang server tersebut di Amazon EC2 dengan menggunakan backup dan restore native otomatis di Migration Hub Orchestrator. Untuk mempelajari selengkapnya, lihat [Rehost SQL server di Amazon EC2](#) dalam dokumentasi Migration Hub Orchestrator.
- Platform ulang SQL Server di Amazon RDS. Anda dapat memilih database SQL Server tertentu dan memplatformnya kembali di Amazon RDS dengan menggunakan pencadangan dan pemulihan

asli otomatis di Migration Hub Orchestrator. Untuk mempelajari selengkapnya, lihat [Memplatform ulang SQL server di Amazon RDS](#) dalam dokumentasi Migration Hub Orchestrator.

- Rehost aplikasi Windows dan SQL Server di Amazon EC2. Anda dapat mengangkat dan menggeser server Windows Anda yang menjalankan .NET dan SQL Server ke Amazon EC2 dengan menggunakan aplikasi Rehost pada template Amazon EC2. Untuk mempelajari selengkapnya, lihat [Rehost aplikasi di Amazon EC2](#) dalam dokumentasi Migration Hub Orchestrator.

Migration Hub Orchestrator membantu menghindari kelebihan jadwal dan anggaran dalam migrasi SQL Server Anda. Manfaat utama lainnya termasuk:

- Migrasikan aplikasi dengan menggunakan metodologi preskriptif. Anda dapat memulai dengan cepat dengan templat alur kerja yang telah ditentukan, yang didasarkan pada praktik terbaik migrasi yang telah terbukti. Anda juga dapat menyesuaikan alur kerja migrasi dengan menambahkan, menyusun ulang, dan menghapus langkah-langkah berdasarkan kebutuhan Anda. Misalnya, Anda dapat menambahkan langkah untuk persetujuan cutover.
- Otomatiskan langkah-langkah manual. Migration Hub Orchestrator mengotomatiskan tugas manual seperti menginstal agen, mengimpor gambar lokal, menyediakan lingkungan target, dan memverifikasi lingkungan sumber dan target. AWS Otomatisasi menghemat waktu dan biaya sekaligus mengurangi kesalahan.
- Mengatur alur kerja migrasi. Migration Hub Orchestrator mengatur alat yang digunakan dalam langkah migrasi dengan menggunakan kembali metadata inventaris, spesifikasi konfigurasi, dan konteks lingkungan untuk meminimalkan jumlah input yang dibutuhkan alat ini.

Untuk informasi tambahan, lihat sumber daya berikut:

- [Konsol Orkestrator Hub Migrasi](#)
- [Rehost aplikasi di Amazon EC2](#) (Panduan Pengguna Orkestrator Hub Migrasi)
- [Platform ulang server SQL di Amazon RDS \(Panduan Pengguna Orkestrator Hub Migrasi\)](#)
- [Alur kerja migrasi](#) (Panduan Pengguna Orkestrator Hub Migrasi)
- [Menggunakan Migration Hub Orchestrator untuk menyederhanakan dan mempercepat migrasi Microsoft SQL Server \(posting blog\)](#) AWS
- [Sederhanakan migrasi gambar Windows Server Anda dengan Orkestrator AWS Migration Hub](#) (AWSposting blog)

AWS SnowballTepi

Anda dapat menggunakan AWS Snowball Edge untuk memigrasi database yang sangat besar (berukuran hingga 210 TB). Snowball memiliki port Ethernet 10 Gb yang Anda colokkan ke server lokal dan menempatkan semua backup database atau data pada perangkat Snowball. Setelah data disalin ke Snowball, Anda mengirim alat AWS ke untuk ditempatkan di bucket S3 yang Anda tentukan. Anda kemudian dapat mengunduh cadangan dari Amazon S3 dan memulihkannya di SQL Server pada instans EC2, atau menjalankan `rds_restore_database` prosedur tersimpan untuk memulihkan database ke Amazon RDS. Anda juga dapat menggunakan [AWSSnowcone](#) untuk database berukuran hingga 8 TB. Untuk informasi selengkapnya, lihat [dokumentasi AWS Snowball Edge](#) dan [Mengimpor dan mengekspor database SQL Server, Memulihkan bagian database](#), di dokumentasi Amazon RDS.

Migrasi database homogen untuk SQL Server

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

AWS menawarkan tiga opsi untuk menjalankan SQL Server AWS, seperti yang dijelaskan dalam tabel berikut.

Opsi	Sorotan	Informasi lain
SQL Server di Amazon RDS	Layanan terkelola, menyediakan an penyediaan dan perizinan yang mudah, hemat biaya, mudah diatur, dikelola, dan dipelihara.	Amazon RDS untuk bagian SQL Server
SQL Server di Amazon RDS Kustom	Layanan terkelola, tetapi Anda mempertahankan hak administratif atas database dan sistem operasi yang mendasarinya.	Amazon RDS Kustom untuk bagian SQL Server
SQL Server di Amazon EC2	Dikelola sendiri, memberikan kontrol dan fleksibilitas penuh.	Amazon EC2 untuk bagian SQL Server
SQL Server di VMware Cloud di AWS	Siapkan, skala, dan operasikan beban kerja SQL Server Anda di VMware Cloud AWS dan integrasikan dengan AWS Directory Service, Active Directory Connector, dan Amazon S3.	VMware Cloud aktif AWS untuk bagian SQL Server

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 SQL Server 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 pada edisi SQL Server Enterprise tetapi cocok untuk edisi Standar SQL Server. Anda mungkin juga ingin memodernisasi database SQL Server Anda yang berjalan di Windows untuk berjalan pada sistem operasi Linux untuk menghemat biaya dan lisensi. Banyak AWS pelanggan menjalankan beberapa beban kerja database SQL Server di Amazon RDS, Amazon EC2, dan VMware Cloud on. AWS

Note

Anda dapat menggunakan Migration Hub Orchestrator untuk mengotomatiskan dan mengatur migrasi database SQL Server Anda ke Amazon EC2 atau Amazon RDS dengan menggunakan pencadangan dan pemulihan asli. Untuk informasi lebih lanjut, lihat [Orkestrator AWS Migration Hub bagian](#).

Amazon RDS for SQL Server

Amazon RDS for SQL Server adalah layanan database terkelola yang menyederhanakan penyediaan dan pengelolaan SQL Server aktif. AWS Amazon RDS memudahkan untuk mengatur, mengoperasikan, dan menskalakan penerapan SQL Server di cloud. Dengan Amazon RDS, Anda dapat menerapkan beberapa versi SQL Server (2014, 2016, 2017, 2019, dan 2022) dan edisi (termasuk Express, Web, Standard, dan Enterprise) dalam hitungan menit, dengan kapasitas komputasi yang hemat biaya dan dapat diubah ukurannya. Anda dapat menyediakan instans Amazon RDS for SQL Server DB dengan SSD Tujuan Umum atau penyimpanan SSD IOPS Tertentu. (Untuk detailnya, lihat [Jenis Penyimpanan Amazon RDS](#) dalam AWS dokumentasi.) IOPS SSD yang disediakan dirancang untuk memberikan kinerja I/O yang cepat, dapat diprediksi, dan konsisten, dan dioptimalkan untuk beban kerja database intensif I/O, transaksional (OLTP).

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 SQL Server juga menawarkan penerapan Multi-AZ dan replika baca (untuk edisi SQL Server Enterprise) untuk menyediakan ketersediaan, kinerja, skalabilitas, dan keandalan yang tinggi untuk beban kerja produksi.

Untuk informasi selengkapnya tentang migrasi dari SQL Server ke Amazon RDS, lihat [pola replatform](#) di situs web Prescriptive Guidance. AWS

Kapan memilih Amazon RDS

Amazon RDS for SQL Server adalah opsi migrasi 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 SQL Server kecil, 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 pencerminan database secara manual, cluster failover, atau grup ketersediaan Selalu Aktif.
- Anda ingin membayar lisensi SQL Server sebagai bagian dari biaya instans setiap jam alih-alih melakukan investasi di muka yang besar.
- Ukuran database dan kebutuhan IOPS Anda didukung oleh Amazon RDS for SQL Server. 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 ingin fokus pada tugas tingkat tinggi, seperti penyetelan kinerja dan pengoptimalan skema, alih-alih administrasi harian database.
- Anda ingin menskalakan jenis instans ke atas atau ke bawah berdasarkan pola beban kerja Anda tanpa khawatir tentang kompleksitas lisensi.

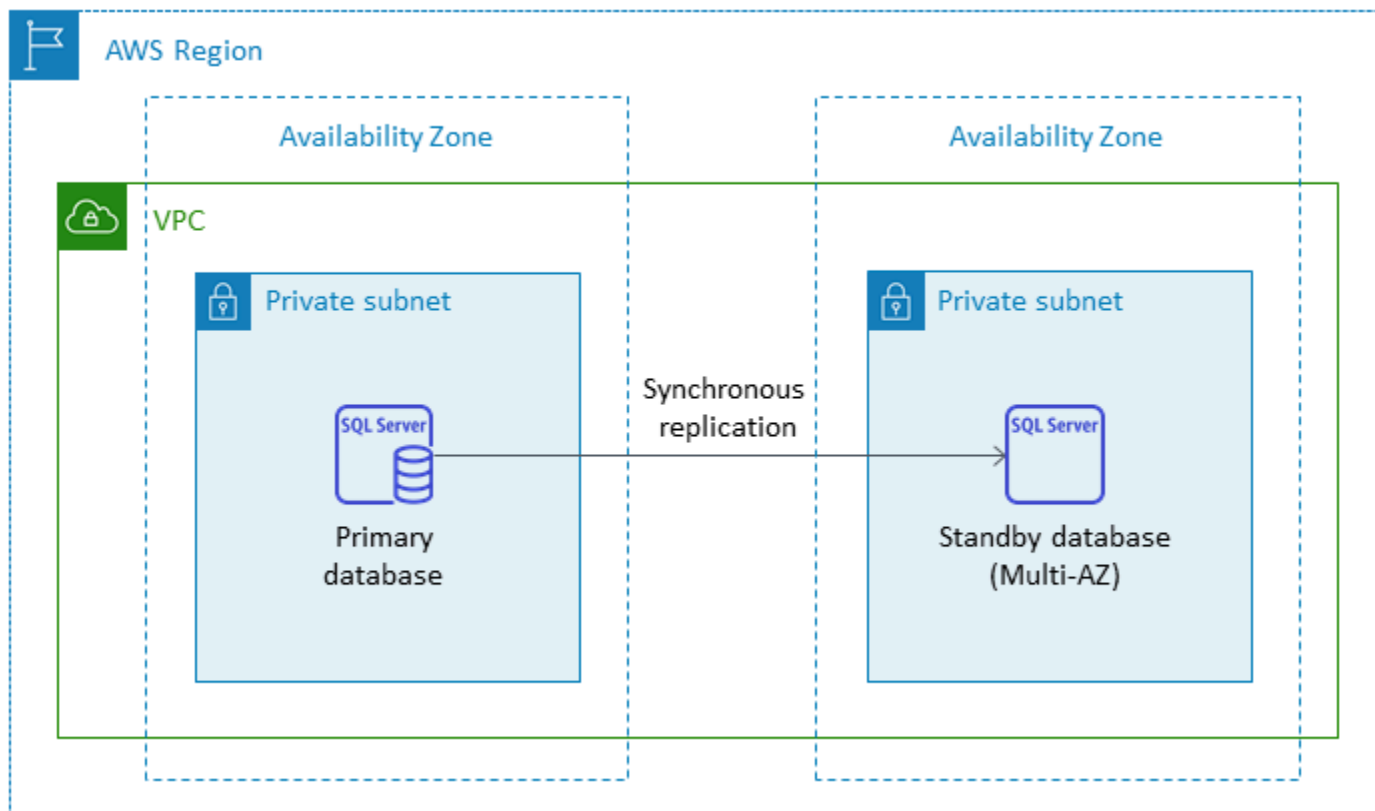
Setelah menilai persyaratan database dan proyek, jika Anda memutuskan untuk bermigrasi ke Amazon RDS for SQL Server, lihat detail yang disediakan di bagian berikut, dan tinjau praktik [terbaik migrasi yang akan dibahas nanti](#) dalam panduan ini.

Untuk fitur, versi, dan opsi SQL Server yang didukung saat ini, lihat fitur [Amazon RDS for SQL Server AWS](#) di [situs web, Memilih antara Amazon EC2 dan Amazon RDS nanti dalam panduan ini, dan Microsoft SQL Server](#) di Amazon RDS dalam dokumentasi. AWS Jika Anda pindah ke Amazon RDS Custom, pastikan untuk meninjau [persyaratan dan batasan untuk Amazon RDS Custom for SQL Server](#).

Ketersediaan tinggi

Amazon RDS menyediakan ketersediaan tinggi dan dukungan failover untuk database yang digunakan dengan opsi Multi-AZ. Ketika Anda menyediakan database Anda dengan opsi Multi-AZ, Amazon RDS secara otomatis menyediakan dan memelihara instans siaga sinkron di Availability Zone yang berbeda. Database utama secara sinkron mereplikasi data ke instance siaga. Jika terjadi masalah, Amazon RDS secara otomatis memperbaiki instans yang tidak sehat dan menetapkan kembali sinkronisasi. Jika terjadi kegagalan infrastruktur atau gangguan Availability Zone, Amazon RDS melakukan failover otomatis ke instans siaga. Failover terjadi hanya jika database siaga dan primer sepenuhnya disinkronkan. Karena titik akhir tetap sama untuk instance primer dan siaga, Anda dapat melanjutkan operasi database segera setelah failover selesai, tanpa melakukan intervensi manual. Waktu failover tergantung pada waktu yang diperlukan untuk menyelesaikan proses pemulihan. Transaksi besar akan meningkatkan waktu failover.

Diagram berikut menggambarkan opsi penyebaran Amazon RDS for SQL Server Multi-AZ.



Saat Anda menyiapkan SQL Server dalam konfigurasi Multi-AZ, Amazon RDS secara otomatis mengonfigurasi instans database siaga menggunakan pencerminan database atau grup ketersediaan

Selalu Aktif, berdasarkan versi SQL Server yang Anda gunakan. Versi dan edisi SQL Server tertentu tercantum dalam dokumentasi [Amazon RDS](#).

Dalam penerapan multi-AZ, operasi seperti penskalaan instance atau peningkatan sistem seperti patching sistem operasi (OS) diterapkan terlebih dahulu pada instance siaga, sebelum failover otomatis instans utama, untuk meningkatkan ketersediaan.

Karena optimasi failover SQL Server, beban kerja tertentu dapat menghasilkan beban I/O yang lebih besar pada instance siaga daripada yang mereka lakukan pada instance utama, terutama dalam penerapan mirroring database. Fungsionalitas ini dapat menghasilkan IOPS yang lebih tinggi pada instance siaga. Kami menyarankan Anda mempertimbangkan kebutuhan IOPS maksimum dari instans primer dan siaga saat Anda menyediakan jenis penyimpanan dan IOPS instans Amazon RDS for SQL Server DB Anda. Anda juga dapat menentukan `MultiSubnetFailover=True`, jika driver klien Anda mendukungnya, untuk secara signifikan mengurangi waktu failover.

Batasan

- Opsi Multi-AZ tidak tersedia untuk edisi SQL Server Express dan Web. Ini hanya tersedia untuk edisi SQL Server Standard dan Enterprise.
- Anda tidak dapat mengonfigurasi instans DB siaga untuk menerima aktivitas baca database.
- Cross-Region Multi-AZ tidak didukung.
- Di Amazon RDS, Anda dapat mengeluarkan perintah stop ke instans DB mandiri dan menyimpan instance dalam status berhenti untuk menghindari biaya komputasi. Anda tidak dapat menghentikan instans DB Amazon RDS for SQL Server dalam konfigurasi Multi-AZ. Sebagai gantinya, Anda dapat menghentikan instance, mengambil snapshot terakhir sebelum penghentian, dan membuat ulang instance Amazon RDS baru dari snapshot saat Anda membutuhkannya. Atau, Anda dapat menghapus konfigurasi Multi-AZ terlebih dahulu dan kemudian menghentikan instance. Setelah tujuh hari, instans berhenti Anda akan dimulai ulang sehingga pemeliharaan yang tertunda dapat diterapkan.

Untuk batasan tambahan, lihat [Catatan dan rekomendasi penerapan Multi-AZ Microsoft SQL Server](#) dalam dokumentasi Amazon RDS.

Replika baca

Replika baca memberikan skalabilitas dan penyeimbangan beban. Replika baca SQL Server adalah salinan fisik dari instans Amazon RDS for SQL Server DB yang digunakan untuk tujuan hanya-baca.

Amazon RDS membantu mengurangi beban pada instans DB primer dengan membongkar beban kerja hanya-baca ke instans DB replika baca. Pembaruan yang dilakukan pada instans DB utama Anda disalin secara asinkron ke instance replika baca.

Saat Anda meminta replika baca, Amazon RDS mengambil snapshot dari instans DB sumber, dan snapshot ini menjadi replika baca. Tidak ada pemadaman saat membuat dan menghapus replika baca. Amazon RDS for SQL Server memutakhirkan database utama segera setelah memutakhirkan replika baca, terlepas dari jendela pemeliharaannya. Setiap replika baca dilengkapi dengan titik akhir terpisah yang Anda gunakan untuk terhubung ke database replika baca.

Amazon RDS for SQL Server memudahkan pembuatan replika baca dengan mengonfigurasi grup ketersediaan Selalu Aktif, dan menjaga koneksi jaringan yang aman antara instans DB primer dan replika bacanya.

Anda dapat mengatur replika baca di AWS Wilayah yang sama dengan database utama Anda. Amazon RDS for SQL Server tidak mendukung replika baca lintas wilayah. Anda dapat membuat hingga lima replika baca untuk satu instans DB sumber.

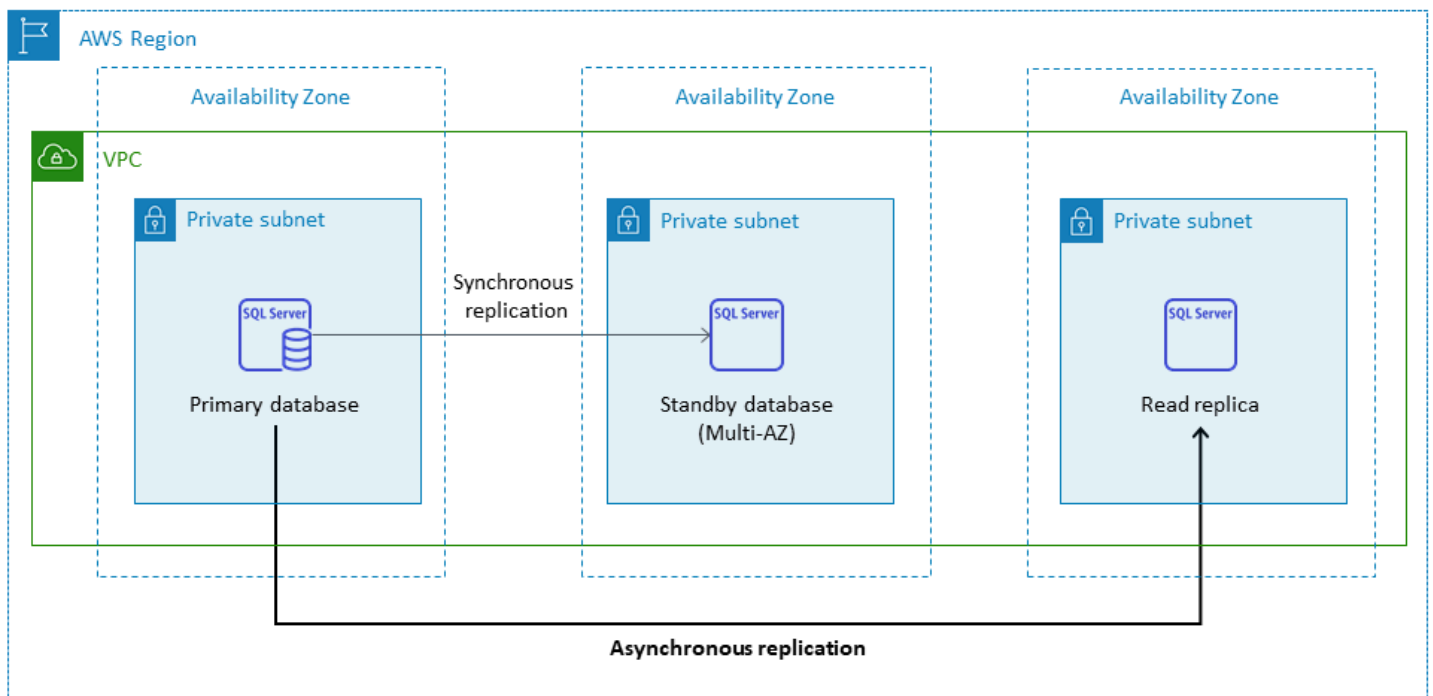
Note

Replika baca hanya tersedia dengan versi dan edisi SQL Server berikut:

- SQL Server 2017 Enterprise edisi 14.00.3049.1 atau yang lebih baru
- SQL Server 2016 Enterprise edisi 13.00.5216.0 atau yang lebih baru

Versi dan edisi SQL Server yang mendukung pencerminan basis data untuk lingkungan multi-AZ tidak menawarkan replika baca.

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



Replika baca SQL Server tidak mengizinkan operasi penulisan. Namun, Anda dapat mempromosikan replika baca untuk membuatnya dapat ditulis. Setelah Anda mempromosikannya, Anda tidak dapat mengembalikannya kembali ke replika baca. Ini akan menjadi instance DB tunggal mandiri yang tidak memiliki hubungan dengan instance database primer aslinya. Data dalam replika baca yang dipromosikan akan cocok dengan data dalam instans DB sumber hingga permintaan dibuat untuk mempromosikannya. Versi mesin SQL Server DB dari instans DB sumber dan semua replika bacanya akan sama.

Untuk replikasi yang efisien, kami merekomendasikan yang berikut:

- Siapkan setiap replika baca dengan sumber daya komputasi dan penyimpanan yang sama dengan instans DB sumber.
- Anda harus mengaktifkan pencadangan otomatis pada instans DB sumber dengan mengatur periode retensi cadangan ke nilai selain 0 (noI).
- Instans DB sumber harus diterapkan di lingkungan multi-AZ dengan grup ketersediaan Selalu Aktif.

Untuk dukungan, edisi, dan batasan versi SQL Server, lihat [Membaca batasan replika dengan SQL Server dalam dokumentasi](#) Amazon RDS.

Untuk informasi selengkapnya tentang menggunakan replika baca, lihat [Bekerja dengan replika baca](#) dan [Bekerja dengan SQL Server membaca replika untuk Amazon RDS](#) dalam dokumentasi AWS. Untuk informasi selengkapnya tentang biaya transfer data, lihat [Harga Amazon RDS](#).

Pemulihan bencana

Dengan Amazon RDS for SQL Server, Anda dapat membuat strategi pemulihan bencana lintas wilayah (DR) yang andal. Alasan utama untuk menciptakan solusi DR adalah kelangsungan bisnis dan kepatuhan:

- Strategi DR yang efektif membantu Anda menjaga sistem Anda tetap aktif dan berjalan dengan gangguan minimal atau tanpa gangguan selama peristiwa bencana. Strategi DR lintas wilayah yang andal dan efektif membuat bisnis Anda tetap beroperasi bahkan jika seluruh Wilayah offline.
- Solusi DR lintas wilayah membantu Anda memenuhi persyaratan audit dan kepatuhan.

Tujuan titik pemulihan (RPO), tujuan waktu pemulihan (RTO), dan biaya adalah tiga metrik utama yang perlu dipertimbangkan saat mengembangkan strategi DR Anda. Untuk opsi lain untuk menyediakan replika lintas wilayah, lihat [AWS Marketplace](#). Untuk informasi selengkapnya tentang pendekatan ini, lihat [Pemulihan bencana Lintas Wilayah Amazon RDS for SQL Server AWS](#) di blog Database.

Amazon RDS Custom for SQL Server

Jika Anda tidak dapat pindah ke layanan yang dikelola sepenuhnya seperti Amazon RDS karena persyaratan penyesuaian untuk aplikasi pihak ketiga, Anda dapat bermigrasi ke Amazon RDS Custom for SQL Server. Dengan Amazon RDS Custom, Anda dapat mempertahankan hak administratif ke database dan sistem operasi dasarnya untuk mengaktifkan aplikasi dependen.

Kapan memilih Amazon RDS Custom untuk SQL Server

Amazon RDS Kustom untuk SQL Server 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 pengguna administratif 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 OS) untuk memenuhi kebutuhan database dan aplikasi Anda.

Cara kerjanya

Untuk menggunakan Amazon RDS Custom for SQL Server, tinjau [persyaratan dalam dokumentasi Amazon RDS Custom for SQL Server](#). Anda harus terlebih dahulu menyiapkan lingkungan Anda untuk Amazon RDS Custom for SQL Server, seperti yang dijelaskan dalam dokumentasi [Amazon RDS](#). Setelah lingkungan diatur, ikuti langkah-langkah ini, yang diilustrasikan dalam diagram berikut:

1. Buat instans Amazon RDS Custom untuk SQL Server DB dari versi mesin SQL Server yang ditawarkan oleh Amazon RDS Custom.

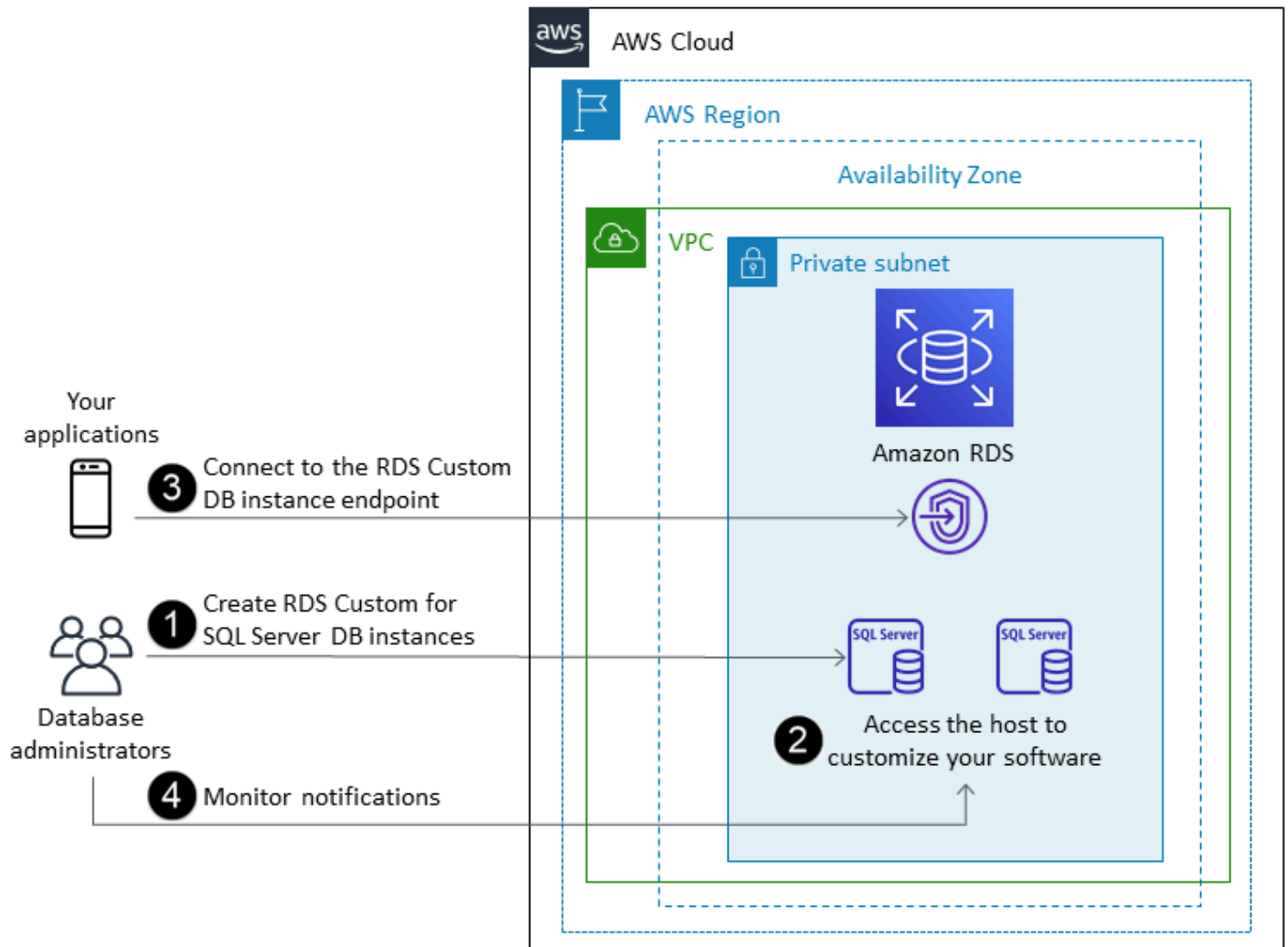
Amazon RDS Custom for SQL Server saat ini mendukung SQL Server 2022 atau 2019 Enterprise, Standard, atau Developer Edition dengan [class instans DB yang didukung](#) yang tercantum dalam dokumentasi. Untuk informasi selengkapnya, lihat [Membuat RDS Custom for SQL Server DB instance](#).

2. Hubungkan aplikasi Anda ke titik akhir instans Amazon RDS Custom DB.

Untuk informasi selengkapnya, lihat [Menghubungkan ke instans DB Kustom RDS menggunakan AWS Systems Manager dan Menghubungkan ke instans DB Kustom RDS menggunakan RDP](#).

3. (Opsional) Akses host untuk menyesuaikan perangkat lunak Anda.
4. Pantau pemberitahuan dan pesan yang dihasilkan oleh Amazon RDS Otomatisasi kustom.

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



Di Amazon RDS Custom for SQL Server, Anda 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 OS 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 menunda otomatisasi Amazon RDS Custom hingga 24 jam, lalu lanjutkannya saat pekerjaan penyesuaian Anda selesai. Menunda 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.

Saat Anda menyiapkan instans Amazon RDS Custom untuk SQL Server DB, lisensi perangkat lunak disertakan. Artinya, Anda tidak perlu membeli lisensi SQL Server secara terpisah. Untuk informasi selengkapnya tentang perizinan, lihat bagian 10.5 dalam ketentuan [AWS layanan](#). Jika Anda memiliki akun Dukungan AWS Premium yang aktif, Anda dapat menghubungi Dukungan AWS Premium untuk Amazon RDS Custom untuk masalah khusus SQL Server.

Amazon RDS Custom for SQL Server didukung dalam pilihan terbatas Wilayah AWS dan dengan kelas instans DB terbatas. Untuk batasan ini dan lainnya, lihat halaman [persyaratan dan batasan](#) di dokumentasi Amazon RDS Custom for SQL Server.

Jika Anda memiliki database SQL Server lokal, Anda dapat mengikuti proses yang dijelaskan dalam [dokumentasi Amazon RDS untuk memigrasikannya ke Amazon RDS Custom for SQL Server](#) dengan menggunakan utilitas pencadangan dan pemulihan bawaan.

Untuk informasi tambahan, lihat sumber daya berikut:

- [Baru - Amazon RDS Kustom untuk SQL Server Umumnya Tersedia \(BlogAWS berita\)](#)
- [Konfigurasi replikasi SQL Server antara Amazon RDS Kustom untuk SQL Server dan Amazon RDS for SQL Server \(blog Database\)AWS](#)
- [Otomatisasi Amazon EC2 SQL Server lokal atau Amazon ke Amazon RDS untuk migrasi SQL Server menggunakan pengiriman log kustom \(Blog database\)AWS](#)
- [Konfigurasi ketersediaan tinggi dengan Grup Ketersediaan Selalu Aktif di Amazon RDS Kustom untuk SQL Server \(blog AWS Database\)](#)
- [Memulai Amazon RDS Custom untuk SQL Server menggunakan AWS CloudFormation template \(Pengaturan jaringan\) \(BlogAWS database\)](#)
- [Memigrasikan beban kerja SQL Server lokal ke Amazon RDS Kustom untuk SQL Server menggunakan grup ketersediaan terdistribusi \(Blog database\)AWS](#)
- [Optimalkan biaya SQL Server Anda dengan menggunakan bawa media Anda sendiri \(BYOM\) di Amazon RDS Kustom untuk SQL Server \(Database blog\)AWS](#)

Amazon EC2 untuk SQL Server

Amazon EC2 mendukung database SQL Server yang dikelola sendiri. Artinya, ini memberi Anda kontrol penuh atas pengaturan 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 EC2, volume penyimpanan, skalabilitas, jaringan, dan keamanan, berdasarkan praktik terbaik AWS arsitektur. Anda bertanggung jawab atas replikasi dan pemulihan data di seluruh instans Anda di Wilayah yang sama atau berbeda AWS .

Untuk informasi selengkapnya tentang migrasi dari SQL Server ke Amazon EC2, lihat pola [rehost di situs web Panduan Preskriptif. AWS](#)

Kapan memilih Amazon EC2

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

- Anda memerlukan kontrol penuh atas database dan akses ke sistem operasi yang mendasarinya, instalasi database, dan konfigurasi.
- Anda ingin mengelola database Anda, termasuk backup dan recovery, patch sistem operasi dan database, tuning sistem operasi dan parameter database, mengelola keamanan, dan mengkonfigurasi ketersediaan tinggi atau replikasi.
- Anda ingin menggunakan fitur dan opsi yang saat ini tidak didukung oleh Amazon RDS. Untuk detailnya, lihat [Fitur yang tidak didukung dan fitur dengan dukungan terbatas](#) dalam dokumentasi Amazon RDS.
- Anda memerlukan versi SQL Server tertentu yang tidak didukung oleh Amazon RDS. Untuk daftar versi dan edisi yang didukung, lihat [versi SQL Server di Amazon RDS dalam dokumentasi Amazon RDS](#).
- Ukuran database dan kebutuhan kinerja Anda melebihi penawaran Amazon RDS for SQL Server saat ini. 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.

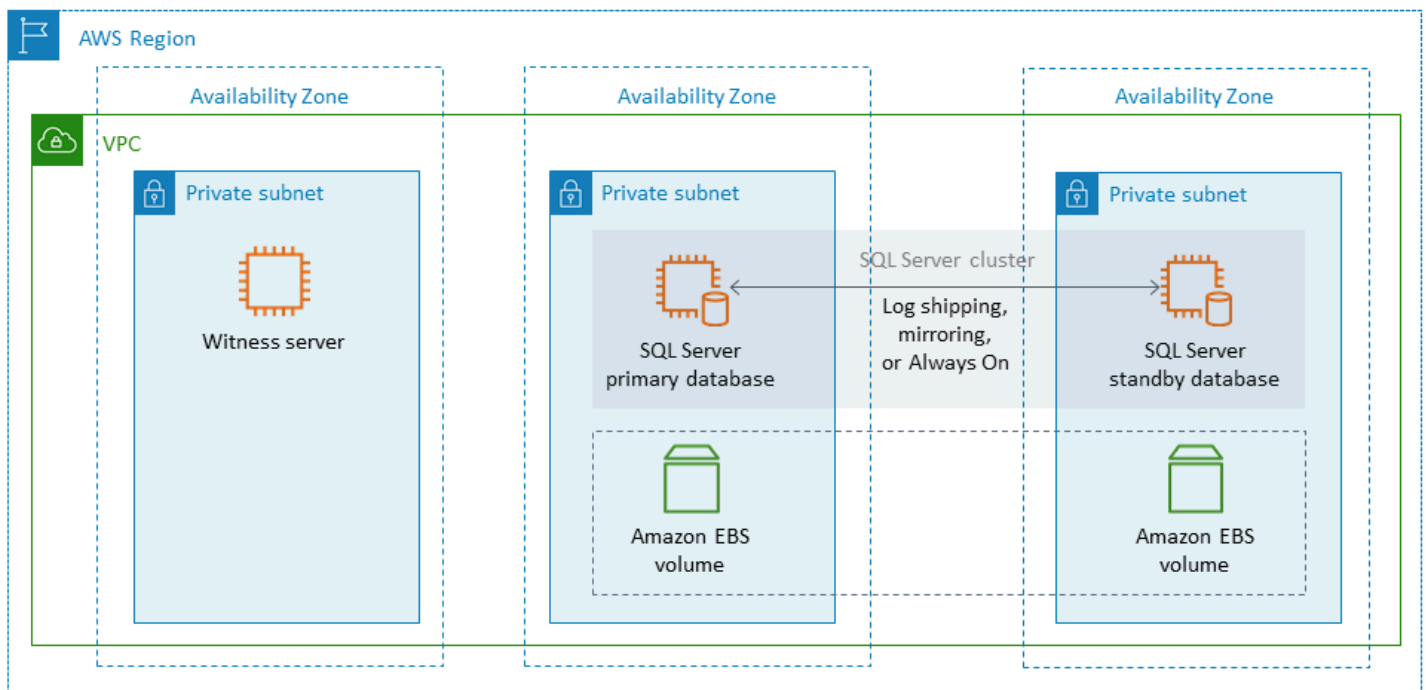
- Anda ingin membawa lisensi Anda sendiri alih-alih menggunakan Amazon RDS for SQL Server model yang disertakan lisensi.
- Anda ingin mencapai IOPS dan kapasitas penyimpanan yang lebih tinggi daripada batas saat ini. Untuk detailnya, lihat [Penyimpanan instans Amazon RDS DB](#) dalam dokumentasi Amazon RDS.

Untuk daftar fitur dan versi SQL Server yang saat ini didukung di Amazon EC2, [lihat Memilih antara Amazon EC2 dan Amazon RDS](#) nanti dalam panduan ini.

Ketersediaan tinggi

Anda dapat menggunakan teknologi replikasi yang didukung SQL Server dengan database SQL Server Anda di Amazon EC2 untuk mencapai ketersediaan tinggi, perlindungan data, dan pemulihan bencana. Beberapa solusi umum adalah pengiriman log, pencerminan basis data, grup ketersediaan Always On, dan Instans Cluster Always On Failover.

Diagram berikut menunjukkan bagaimana Anda dapat menggunakan SQL Server di Amazon EC2 di beberapa Availability Zone dalam satu AWS Region. Database utama adalah database baca-tulis, dan database sekunder dikonfigurasi dengan pengiriman log, pencerminan database, atau grup ketersediaan Always On untuk ketersediaan tinggi. Semua data transaksi dari database utama ditransfer dan dapat diterapkan ke database sekunder secara asinkron untuk pengiriman log, dan secara asinkron untuk grup ketersediaan Always On dan mirroring.



Pengiriman log

Pengiriman log memungkinkan Anda secara otomatis mengirim cadangan log transaksi dari instance database utama ke satu atau lebih database sekunder (juga dikenal sebagai siaga hangat) pada instance DB terpisah. Pengiriman log menggunakan pekerjaan SQL Server Agent untuk mengotomatiskan proses pencadangan, penyalinan, dan penerapan cadangan log transaksi. Meskipun pengiriman log biasanya dianggap sebagai fitur pemulihan bencana, itu juga dapat memberikan ketersediaan tinggi dengan memungkinkan instans DB sekunder untuk dipromosikan jika instans DB primer gagal. Jika RTO dan RPO Anda fleksibel, atau database Anda tidak dianggap sangat penting, pertimbangkan untuk menggunakan pengiriman log untuk menyediakan ketersediaan yang lebih baik untuk database SQL Server Anda.

Pengiriman log meningkatkan ketersediaan database dengan menyediakan akses ke database sekunder untuk digunakan sebagai salinan read-only dari database utama bila diperlukan. Anda dapat mengonfigurasi penundaan jeda (waktu tunda yang lebih lama) di mana Anda dapat memulihkan data yang diubah secara tidak sengaja pada database utama sebelum perubahan ini dikirim ke database sekunder.

Sebaiknya jalankan instans DB primer dan sekunder di Availability Zone terpisah, dan menerapkan instance monitor untuk melacak semua detail pengiriman log. Pencadangan, salin, pemulihan, dan peristiwa kegagalan untuk grup pengiriman log tersedia dari instance monitor. Konfigurasi pengiriman log tidak secara otomatis gagal dari server utama ke server sekunder. Namun, salah satu database sekunder dapat dibawa online secara manual jika database utama menjadi tidak tersedia.

Pengiriman log sering digunakan sebagai solusi pemulihan bencana tetapi juga dapat digunakan sebagai solusi ketersediaan tinggi, tergantung pada kebutuhan aplikasi Anda. Gunakan pengiriman log saat:

- Anda memiliki persyaratan RTO dan RPO yang fleksibel. Pengiriman log menyediakan RPO menit, dan RTO menit hingga jam.
- Anda tidak memerlukan failover otomatis ke database sekunder.
- Anda ingin membaca dari database sekunder, tetapi Anda tidak memerlukan keterbacaan selama operasi pemulihan.

Untuk informasi selengkapnya tentang pengiriman log, lihat [dokumentasi Microsoft SQL Server](#).

Pencerminan basis data

Pencerminan database mengambil database yang ada pada instance EC2 dan menyediakan salinan read-only (mirror) yang lengkap atau hampir lengkap pada instance DB terpisah. Amazon RDS menggunakan mirroring database untuk memberikan dukungan Multi-AZ untuk Amazon RDS for SQL Server. Fitur ini meningkatkan ketersediaan dan perlindungan database, dan menyediakan mekanisme untuk menjaga database tetap tersedia selama peningkatan.

Note

Menurut [dokumentasi Microsoft](#), mirroring database akan dihapus dalam versi SQL Server yang akan datang. Anda harus berencana untuk menggunakan grup ketersediaan Always On sebagai gantinya.

Dalam mirroring database, server SQL dapat mengambil salah satu dari tiga peran:

- Server utama, yang menghosting versi baca/tulis utama database.
- Server cermin, yang menampung salinan database utama.
- Server saksi opsional. Server ini hanya tersedia dalam mode keamanan tinggi. Ini memonitor keadaan cermin database dan mengotomatiskan failover dari database utama ke database mirror.

Sesi mirroring dibuat antara server utama dan server cermin. Selama mirroring, semua perubahan database yang dilakukan dalam database utama juga dilakukan pada database mirror. Pencerminan database dapat berupa operasi sinkron atau asinkron. Ini ditentukan oleh dua mode operasi mirroring: mode keamanan tinggi dan mode kinerja tinggi.

- Mode keamanan tinggi: Mode ini menggunakan operasi sinkron. Dalam mode ini, sesi pencerminan database menyinkronkan operasi penyisipan, pembaruan, dan penghapusan dari database utama ke database cermin secepat mungkin. Segera setelah database disinkronkan, transaksi dilakukan di database prinsipal dan cermin. Kami menyarankan Anda menggunakan mode operasi ini ketika database cermin berada di Availability Zone yang sama atau berbeda, tetapi dihosting dalam AWS Wilayah yang sama.
- Mode kinerja tinggi: Mode ini menggunakan operasi asinkron. Dalam mode ini, sesi pencerminan database menyinkronkan operasi penyisipan, pembaruan, dan penghapusan dari database utama ke database cermin, tetapi mungkin ada jeda antara waktu database utama melakukan transaksi

dan waktu database mirror melakukan transaksi. Kami menyarankan Anda menggunakan mode ini ketika database cermin berada di AWS Wilayah yang berbeda.

Gunakan pencerminan basis data saat:

- Anda memiliki persyaratan RTO dan RPO yang ketat, dan tidak dapat memiliki penundaan antara database primer dan sekunder. Pencerminan basis data menyediakan RPO nol detik (dengan komit sinkron) dan RTO detik hingga menit.
- Anda tidak memiliki persyaratan untuk membaca dari database sekunder.
- Anda ingin melakukan failover otomatis ketika Anda memiliki server saksi yang dikonfigurasi dalam mode sinkronisasi.
- Anda tidak dapat menggunakan grup ketersediaan Selalu Aktif, yang merupakan opsi yang lebih disukai.

Pembatasan:

- Hanya one-to-one failover yang didukung. Anda tidak dapat memiliki beberapa tujuan database yang disinkronkan dengan database utama.

Untuk informasi selengkapnya tentang mirroring, lihat dokumentasi [Microsoft SQL Server](#).

Selalu Aktif pada grup ketersediaan

Grup ketersediaan SQL Server Always On menyediakan ketersediaan tinggi dan solusi pemulihan bencana untuk database SQL Server. Grup ketersediaan terdiri dari satu set database pengguna yang gagal bersama-sama. Ini mencakup satu set database baca/tulis primer dan beberapa (satu hingga delapan) set database sekunder terkait. Anda dapat membuat database sekunder tersedia untuk tingkat aplikasi sebagai salinan hanya-baca dari database utama (edisi SQL Server Enterprise saja), untuk menyediakan arsitektur scale-out untuk beban kerja baca. Anda juga dapat menggunakan database sekunder untuk operasi pencadangan.

Grup ketersediaan SQL Server Always On mendukung mode komit sinkron dan asinkron. Dalam mode sinkron, replika utama melakukan transaksi database setelah perubahan dilakukan atau ditulis ke log replika sekunder. Dengan menggunakan mode ini, Anda dapat melakukan failover manual yang direncanakan dan failover otomatis jika replika disinkronkan. Anda dapat menggunakan mode komit sinkron antara instance SQL Server dalam lingkungan yang sama (misalnya, jika semua instance berada di lokasi atau semua instance berada di). AWS

Dalam mode komit asinkron, replika utama melakukan transaksi database tanpa menunggu replika sekunder. Anda dapat menggunakan mode komit asinkron antara instance SQL Server yang berada di lingkungan yang berbeda (misalnya, jika Anda memiliki instance di tempat dan di). AWS

Anda dapat menggunakan grup ketersediaan Selalu Aktif untuk ketersediaan tinggi atau pemulihan bencana. Gunakan metode ini ketika:

- Anda memiliki persyaratan RTO dan RPO yang ketat. Grup ketersediaan Selalu Aktif menyediakan RPO detik, dan RTO detik hingga menit.
- Anda ingin mengelola dan gagal melalui sekelompok database. Grup ketersediaan Selalu Aktif mendukung 0-4 replika sekunder dalam mode komit sinkron untuk SQL Server 2019.
- Anda ingin menggunakan failover otomatis dalam mode komit sinkron, dan Anda tidak memerlukan server saksi.
- Anda ingin membaca dari database sekunder.
- Anda ingin menyinkronkan beberapa tujuan database dengan database utama Anda.

Dimulai dengan SQL Server 2016 SP1, edisi Standar SQL Server menyediakan ketersediaan tinggi dasar untuk database sekunder tunggal yang tidak dapat dibaca dan pendengar per grup ketersediaan. Ini juga mendukung maksimal dua node per grup ketersediaan.

Selalu Pada Instans Failover Cluster

SQL Server Always On Failover Cluster Instances (FCI) menggunakan Windows Server Failover Clustering (WSFC) untuk menyediakan ketersediaan tinggi di tingkat instans server. FCI adalah contoh tunggal SQL Server yang diinstal di seluruh node WSFC untuk menyediakan ketersediaan tinggi untuk seluruh instalasi SQL Server. Jika node yang mendasari mengalami kegagalan perangkat keras, sistem operasi, aplikasi, atau layanan, semua yang ada di dalam instance SQL Server dipindahkan ke node WSFC lain. Ini termasuk database sistem, login SQL Server, pekerjaan Agen SQL Server, dan sertifikat.

FCI umumnya lebih disukai daripada grup ketersediaan Selalu Aktif ketika:

- Anda menggunakan edisi Standar SQL Server bukan edisi Enterprise.
- Anda memiliki sejumlah besar database kecil per instance.
- Anda terus-menerus memodifikasi objek tingkat instance seperti pekerjaan SQL Server Agent, login, dan sebagainya.

- Anda ingin mengoptimalkan biaya (baik biaya infrastruktur Amazon EC2 dan biaya lisensi SQL Server) untuk menjalankan instance SQL Server. Saat menggunakan FCI, Anda menghapus ketergantungan instans EC2 host SQL Server di throughput Amazon Elastic Block Store (Amazon EBS). Akibatnya, Anda dapat mengukur dengan benar (yaitu, tanpa perlu penyediaan berlebih) instans SQL Server EC2 dan mengurangi biaya lisensi Amazon EC2 dan SQL Server Anda, tanpa memengaruhi kinerja I/O.
- Anda ingin mengoptimalkan kinerja database pada instance SQL Server. Menggunakan FCI memberikan kinerja sekitar 30 persen lebih baik untuk beban kerja tipikal, karena replikasi data dilakukan pada tingkat blok, bukan tingkat database.

FCI memerlukan beberapa bentuk penyimpanan bersama — volume IOPS SSD (io2) Amazon EBS Provisioned IOPS, disk pada jaringan area penyimpanan (SAN), berbagi file di Blok Pesan Server (SMB), atau penyimpanan yang terpasang secara lokal dengan Storage Spaces Direct (S2D), SIOS Datakeeper, atau Amazon FSX — untuk memberikan ketahanan dan ketersediaan tinggi.

Ada empat opsi untuk menerapkan FCI di: AWS

- Amazon EBS Multi-Lampirkan dengan reservasi persisten
- Amazon FSx for Windows File Server
- Amazon FSx untuk ONTAP NetApp
- Solusi dari AWS Mitra

Menggunakan Amazon EBS Multi-Attach dengan reservasi persisten

[Amazon EBS Multi-Attach dengan reservasi NVMe](#) mendukung pembuatan SQL Server FCI dengan volume Amazon io2 EBS sebagai penyimpanan bersama pada cluster failover Windows Server. Fitur ini menyederhanakan proses penyiapan kluster failover dengan memungkinkan Anda membangun kluster failover dengan menggunakan volume Amazon EBS. io2 Volume ini hanya dapat dilampirkan ke instance yang berada di Availability Zone yang sama. Untuk menyebarkan cluster failover Windows Server dengan menggunakan io2 volume Amazon EBS, Anda harus menggunakan driver NVMe terbaru. AWS

Volume Amazon EBS dan volume penyimpanan instans ditampilkan sebagai perangkat blok NVMe di [instans berbasis Nitro](#). Anda harus menginstal [driver AWS NVMe](#) dengan [fitur reservasi persisten SCSI](#) yang dikonfigurasi saat Anda menggunakan io2 volume Amazon EBS untuk membentuk WSFC dan SQL Server FCI.

Untuk informasi selengkapnya tentang fitur ini, lihat posting AWS blog [Cara menerapkan cluster failover SQL Server dengan Amazon EBS Multi-Attach](#) di Windows Server.

Menggunakan Amazon FSx for Windows File Server

[Amazon FSx for Windows File Server](#) menyediakan penyimpanan file bersama yang dikelola sepenuhnya. Ini secara otomatis mereplikasi penyimpanan secara sinkron di dua Availability Zone untuk memberikan ketersediaan tinggi. Menggunakan FSx for Windows File Server untuk penyimpanan file membantu menyederhanakan dan mengoptimalkan penerapan ketersediaan tinggi SQL Server di Amazon EC2.

Dengan Microsoft SQL Server, ketersediaan tinggi biasanya digunakan di beberapa node database dalam WSFC, dan setiap node memiliki akses ke penyimpanan file bersama. Anda dapat menggunakan FSx for Windows File Server sebagai penyimpanan bersama untuk penyebaran ketersediaan tinggi SQL Server dalam dua cara: sebagai penyimpanan untuk file data aktif dan sebagai saksi berbagi file SMB.

Untuk informasi tentang bagaimana Anda dapat mengurangi kompleksitas dan biaya menjalankan penerapan SQL Server FCI dengan menggunakan FSx for Windows File Server, lihat posting blog Menyederhanakan [penerapan ketersediaan tinggi Microsoft SQL Server Anda menggunakan Amazon FSx for Windows File Server](#). Posting blog juga memberikan step-by-step instruksi untuk menyebarkan SQL Server FCI dengan menggunakan sistem file Amazon FSx Multi-AZ sebagai solusi penyimpanan bersama. Untuk informasi selengkapnya, lihat [dokumentasi Amazon FSx for Windows File Server](#).

Menggunakan Amazon FSx untuk ONTAP NetApp

Amazon FSx untuk NetApp ONTAP adalah layanan terkelola penuh yang menyediakan penyimpanan file yang sangat andal, terukur, berkinerja tinggi, dan kaya fitur yang dibangun di atas sistem file ONTAP. NetApp FSx untuk ONTAP menggabungkan fitur, kinerja, kemampuan, dan operasi API sistem NetApp file yang sudah dikenal dengan kelincahan, skalabilitas, dan kesederhanaan layanan yang dikelola sepenuhnya. AWS

fsX untuk ONTAP menyediakan akses multi-protokol ke data melalui protokol NFS, SMB, dan iSCSI untuk sistem Windows dan Linux. Anda dapat membangun arsitektur SQL Server Always On FCI yang sangat tersedia, seperti yang dijelaskan secara rinci dalam posting blog [SQL Server High Availability Deployment Menggunakan Amazon FSx untuk ONTAP](#). NetApp FSx untuk ONTAP juga dapat menyediakan cara cepat untuk gagal di lingkungan SQL Server Anda ke yang berbeda Wilayah AWS untuk memenuhi persyaratan tujuan waktu pemulihan (RTO) dan tujuan titik pemulihan (RPO).

Untuk informasi lebih lanjut, lihat posting blog [Menerapkan HA dan DR untuk SQL Server Always-On Failover Cluster Instance menggunakan FSx](#) untuk ONTAP.

Anda juga dapat menggunakan AWS Launch Wizard untuk menerapkan solusi SQL Server AWS, dengan dukungan untuk Always On Availability Groups dan single-node deployment. Launch Wizard mendukung penerapan SQL Server Always on FCI di Amazon EC2 dengan FSx untuk ONTAP sebagai penyimpanan bersama. Layanan ini menghemat waktu dan tenaga Anda dengan mengganti proses penerapan manual yang kompleks dengan panduan berbasis konsol terpandu yang mempercepat migrasi beban kerja SQL Server lokal yang bergantung pada penyimpanan bersama. Untuk informasi selengkapnya tentang bagaimana Launch Wizard dapat membantu Anda menyediakan dan mengkonfigurasi SQL Server FCI dalam hitungan jam, lihat posting blog [Sederhanakan SQL Server Always On deployment](#) with dan Amazon FSx. AWS Launch Wizard Launch Wizard juga mendukung penerapan untuk SQL Server Always On FCI dengan menggunakan [Amazon FSx for Windows File Server](#) sebagai solusi penyimpanan bersama.

Menggunakan solusi dari AWS Mitra

- [SIOS DataKeeper](#) menyediakan dukungan failover cluster ketersediaan tinggi di seluruh Wilayah AWS dan Availability Zones. SIOS DataKeeper tersedia di [AWS Marketplace](#).
- [DxEnterprise](#) dari DH2i memungkinkan failover otomatis dari Grup Ketersediaan SQL Server di Kubernetes dan failover instance terpadu untuk Windows dan Linux. D2HI tersedia di [AWS Marketplace](#)

FSx for Windows File Server

FSx for Windows File Server menyediakan penyimpanan file yang dikelola sepenuhnya, sangat andal, dan terukur yang dapat diakses dengan menggunakan protokol Server Message Block (SMB). Ini dibangun di atas Windows Server dan memberikan berbagai fitur administratif seperti kuota pengguna, pemulihan file pengguna akhir, dan integrasi Microsoft Active Directory (AD). Ini menawarkan opsi penyebaran Single-AZ dan multi-AZ, pencadangan yang dikelola sepenuhnya, dan enkripsi data saat istirahat dan dalam perjalanan. Anda dapat mengoptimalkan biaya dan kinerja untuk beban kerja Anda dengan opsi penyimpanan solid-state drive (SSD) dan hard disk drive (HDD), dan Anda dapat menskalakan penyimpanan dan mengubah kinerja throughput sistem file Anda kapan saja. Penyimpanan file Amazon FSx dapat diakses dari Windows, instans komputasi Linux yang berjalan di AWS, dan di tempat.

Amazon FSx mempermudah penerapan penyimpanan Windows bersama untuk penyebaran SQL Server dengan ketersediaan tinggi melalui dukungannya untuk berbagi file (CA) yang tersedia secara terus menerus dan sistem file yang lebih kecil. Opsi ini cocok untuk kasus penggunaan ini:

- Sebagai penyimpanan bersama yang digunakan oleh node SQL Server dalam instance WSFC.
- Sebagai saksi berbagi file SMB yang dapat digunakan dengan cluster SQL Server apa pun dengan WSFC.

Amazon FSx memberikan kinerja cepat dengan throughput dasar hingga 2 GB/detik per sistem file, ratusan ribu IOPS, dan latensi sub-milidetik yang konsisten.

Untuk memberikan kinerja yang tepat untuk instans SQL Anda, Anda dapat memilih tingkat throughput yang independen dari ukuran sistem file Anda. Tingkat kapasitas throughput yang lebih tinggi juga datang dengan tingkat IOPS yang lebih tinggi yang dapat disajikan oleh server file ke instance SQL Server yang mengaksesnya.

Kapasitas penyimpanan tidak hanya menentukan berapa banyak data yang dapat Anda simpan, tetapi juga berapa banyak IOPS yang dapat Anda lakukan pada penyimpanan. Setiap gigabyte penyimpanan menyediakan 3 IOPS. Anda dapat menyediakan setiap sistem file berukuran hingga 64 TB.

Untuk informasi tentang mengonfigurasi dan menggunakan Amazon FSx untuk mengurangi kompleksitas dan biaya penerapan ketersediaan tinggi SQL Server Anda, lihat [Menyederhanakan penerapan ketersediaan tinggi Microsoft SQL Server menggunakan FSx for Windows File Server](#) di blog Penyimpanan. AWS Untuk mempelajari selengkapnya tentang membuat berbagi CA baru, lihat dokumentasi [FSx for Windows File Server](#).

Pemulihan bencana

Banyak organisasi menerapkan ketersediaan tinggi untuk database SQL Server mereka, tetapi itu tidak cukup untuk organisasi yang membutuhkan ketahanan TI sejati. Kami menyarankan Anda menerapkan solusi pemulihan bencana untuk menghindari kehilangan data dan downtime database mission-critical. Mengadopsi arsitektur pemulihan bencana Multi-wilayah untuk penerapan SQL Server Anda membantu Anda:

- Mencapai kelangsungan bisnis
- Tingkatkan latensi untuk basis pelanggan Anda yang didistribusikan secara geografis
- Memenuhi persyaratan audit dan peraturan Anda

Opsi untuk pemulihan bencana termasuk [pengiriman log](#), [grup ketersediaan Selalu Aktif](#), [snapshot Amazon EBS](#) yang disimpan di Amazon S3 dan direplikasi di AWS seluruh Wilayah, [Instans Kluster Failover Selalu Aktif \(FCI\)](#) yang dikombinasikan dengan grup ketersediaan Selalu Aktif, dan grup ketersediaan terdistribusi.

Grup ketersediaan terdistribusi

Arsitektur dengan grup ketersediaan terdistribusi adalah pendekatan optimal untuk penyebaran SQL Server Multi-wilayah. Grup ketersediaan terdistribusi adalah jenis grup ketersediaan khusus yang mencakup dua grup ketersediaan terpisah. Anda dapat menganggapnya sebagai grup ketersediaan grup ketersediaan. Grup ketersediaan yang mendasarinya dikonfigurasi pada dua cluster WSFC yang berbeda.

Grup ketersediaan terdistribusi digabungkan secara longgar, yang berarti bahwa mereka tidak memerlukan satu cluster WSFC dan dikelola oleh SQL Server. Karena cluster WSFC dipertahankan secara individual dan transmisi terutama tidak sinkron antara dua kelompok ketersediaan, lebih mudah untuk mengkonfigurasi pemulihan bencana di situs lain. Replika utama di setiap grup ketersediaan menyinkronkan replika sekunder mereka sendiri.

Grup ketersediaan terdistribusi hanya mendukung failover manual saat ini. Untuk memastikan bahwa tidak ada data yang hilang, hentikan semua transaksi pada database primer global (yaitu, pada database grup ketersediaan primer). Kemudian atur grup ketersediaan terdistribusi ke komit sinkron.

VMware Cloud aktif AWS untuk SQL Server

[VMware Cloud on AWS](#) adalah penawaran cloud terintegrasi yang dikembangkan bersama oleh AWS dan VMware. SQL Server dengan mudah terintegrasi dengan VMware Cloud on. AWS Opsi migrasi ini memungkinkan Anda untuk membangun investasi yang ada dalam virtualisasi.

Anda dapat mengakses VMware Cloud setiap jam, berdasarkan permintaan, atau dalam bentuk berlangganan. AWS Ini mencakup teknologi VMware inti yang sama yang Anda jalankan di pusat data Anda, termasuk vSphere Hypervisor (ESXi), Virtual SAN (vSAN), dan platform virtualisasi jaringan NSX, dan dirancang untuk memberikan pengalaman yang efisien dan mulus untuk mengelola database SQL Server Anda. Anda dapat menskalakan penyimpanan, komputasi, dan memori database SQL Server Anda di VMware Cloud dalam beberapa menit. AWS

VMware Cloud AWS berjalan langsung pada perangkat keras fisik, tetapi memanfaatkan fitur jaringan dan perangkat keras yang dirancang untuk mendukung model infrastruktur AWS keamanan

pertama. Ini berarti bahwa tumpukan virtualisasi VMware berjalan pada AWS infrastruktur tanpa harus menggunakan virtualisasi bersarang.

VMware Cloud on AWS memudahkan untuk mengatur, menskalakan, dan mengoperasikan beban kerja database SQL Server Anda. AWS ini menyediakan solusi ketersediaan tinggi, terintegrasi dengan Active Directory lokal, dan menyediakan akses ke AWS layanan seperti AWS Directory Service for Microsoft Active Directory dan AD Connector, Amazon Route 53, Amazon CloudWatch, dan Amazon S3. Anda dapat menyimpan cadangan Anda di Amazon S3, dan memodernisasi serta menyederhanakan proses pemulihan bencana Anda.

Kapan memilih VMware Cloud di AWS

VMware Cloud on AWS adalah opsi untuk database SQL Server Anda ketika:

- Database SQL Server Anda sudah berjalan di pusat data lokal di lingkungan virtual vSphere.
- Anda memiliki sejumlah besar database dan Anda memerlukan migrasi cepat (misalnya, hanya beberapa jam) ke cloud karena salah satu alasan berikut, tanpa memerlukan pekerjaan tambahan dari tim migrasi:
 - Ekstensi pusat data. Anda memerlukan kapasitas sesuai permintaan untuk menjalankan desktop tervirtualisasi, untuk mempublikasikan aplikasi, atau untuk menyediakan lingkungan pengembangan/pengujian.
 - Pemulihan bencana. Anda ingin menyiapkan sistem pemulihan bencana baru atau mengganti sistem yang ada.
 - Migrasi awan. Anda ingin memigrasikan seluruh pusat data Anda ke cloud, atau menyegarkan infrastruktur Anda.

Jika database SQL Server Anda membutuhkan lebih dari 80K IOPS, Anda dapat menggunakan vSAN.

Untuk informasi selengkapnya, lihat [Di Works — VMware Cloud AWS di](#) blog AWS Berita, dan [Menyebarkan Microsoft SQL Server di VMware Cloud di](#) situs web. AWS AWS

Migrasi database heterogen untuk SQL Server

Karena inovasi dan peningkatan dalam database open-source dan platform komputasi awan seperti AWS, banyak organisasi bergerak dari mesin database proprietary (pemrosesan transaksi online atau OLTP) seperti SQL Server ke mesin open-source. Database SQL Server adalah sistem mission-critical 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 untuk mengalihkan teknologi database yang mendasarinya ke database open-source atau cloud-native. AWS

Alasan lain untuk migrasi dari SQL Server adalah periode penguncian vendor, audit lisensi, lisensi mahal, dan biaya. Untuk alasan ini, banyak organisasi memilih untuk memigrasi database SQL Server mereka ke database sumber terbuka (seperti PostgreSQL, MySQL, atau MariaDB) atau database AWS Cloud-native (seperti Amazon Aurora atau Amazon DynamoDB) saat mereka bermigrasi ke. AWS

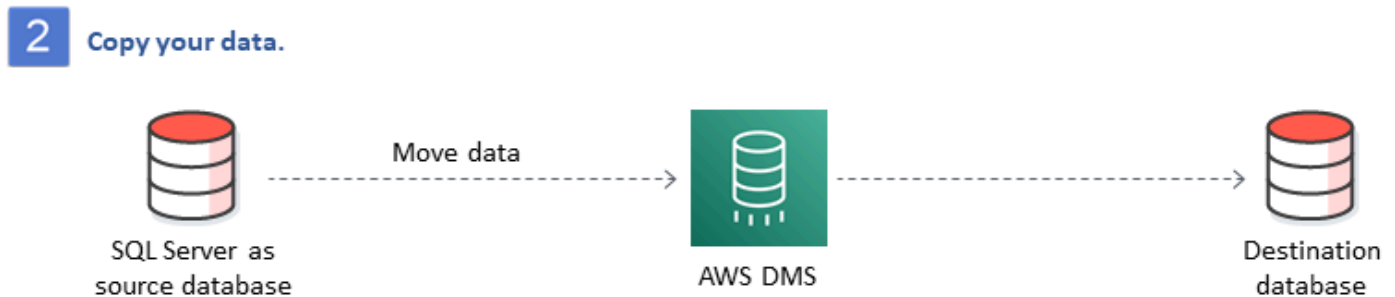
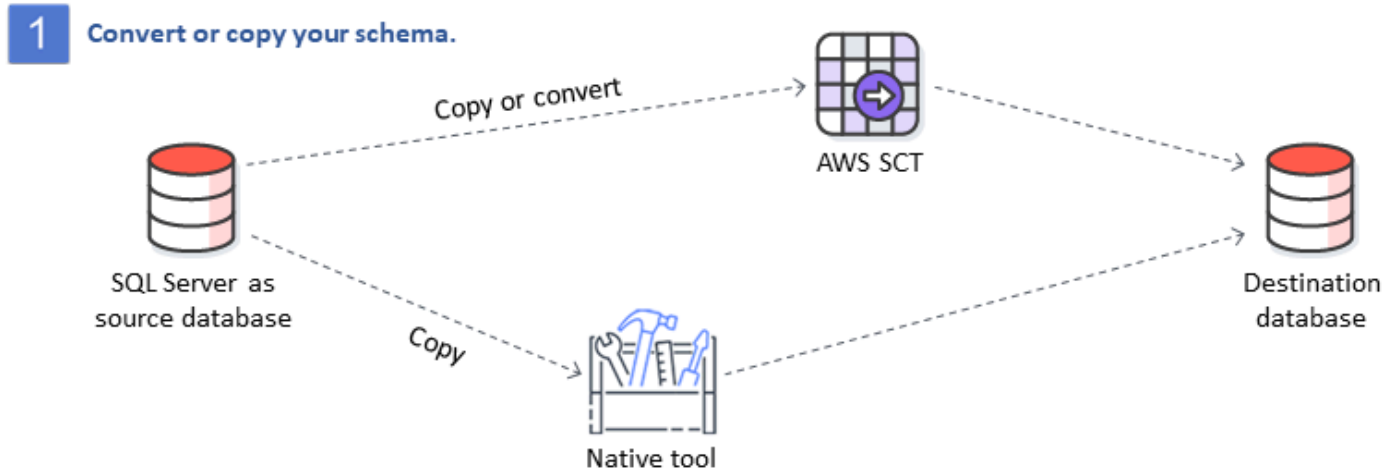
Anda juga dapat memigrasi database gudang data SQL Server Anda ke Amazon Redshift, yang merupakan gudang data cloud yang cepat dan terkelola sepenuhnya. Amazon Redshift terintegrasi dengan data lake Anda, menawarkan kinerja hingga tiga kali lebih cepat daripada gudang data lainnya, dan harganya hingga 75 persen lebih murah daripada gudang data cloud lainnya. Untuk informasi selengkapnya, lihat pola [Memigrasikan database Microsoft SQL Server lokal ke Amazon Redshift menggunakan AWS DMS](#) situs web Panduan AWS Preskriptif.

Untuk bermigrasi ke database open-source atau AWS cloud-native, pilih database yang tepat tergantung pada jenis data yang Anda miliki, model akses, skalabilitas, kepraktisan aplikasi, dan kompleksitas. Migrasi dari SQL Server ke PostgreSQL dan ke database open-source lainnya seringkali sulit dan memakan waktu, dan memerlukan penilaian, perencanaan, dan pengujian yang cermat.

Proses ini menjadi lebih mudah dengan layanan seperti AWS Database Migration Service (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 di SQL Server ke Aurora, atau migrasi SQL Server ke MariaDB. Struktur skema, tipe data, dan kode database di database 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. Mengkonversi skema sumber dan kode untuk mencocokkan yang dari database 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 konversi tipe data utama secara otomatis selama migrasi. Database sumber dapat ditemukan 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 Aurora. Untuk informasi tentang menggunakan MySQL sebagai database target, lihat [Migrasi Database SQL Server ke Mesin Database MySQL yang kompatibel dengan MySQL di blog Database. AWS](#)

Untuk informasi selengkapnya tentang refactoring database SQL Server Anda AWS, lihat [pola arsitek ulang di situs web Panduan Preskriptif. AWS](#)

Alat untuk migrasi database heterogen

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

Alat migrasi	Dukungan basis data target	Digunakan untuk
AWS SCT	Amazon RDS for MySQL Amazon RDS for PostgreSQL Amazon Aurora MySQL Amazon Aurora PostgreSQL	Konversi skema
AWS DMS	Amazon RDS for MySQL Amazon RDS for PostgreSQL Amazon Aurora MySQL Amazon Aurora PostgreSQL	Migrasi data
Ikan Babel	Amazon Aurora PostgreSQL	Akses data dan migrasi

Subbagian berikut memberikan informasi lebih lanjut tentang setiap alat.

AWS SCT

[AWS Schema Conversion Tool\(AWS SCT\)](#) mengubah skema database komersial Anda yang ada menjadi mesin sumber terbuka atau ke database Cloud-native. AWS SCT membuat migrasi database heterogen dapat diprediksi dengan secara otomatis mengubah skema database sumber dan sebagian besar objek kode database, termasuk tampilan, prosedur yang disimpan, dan fungsi, ke format yang kompatibel dengan database target.

Ketika Anda mengkonversi skema database Anda dari satu mesin ke mesin lain, Anda juga perlu memperbarui kode SQL dalam aplikasi Anda untuk berinteraksi dengan mesin database baru bukan yang lama. AWS SCT juga mengkonversi kode SQL di C ++, C #, Java atau kode aplikasi lainnya. 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 mengubahnya sebagai bagian dari proyek konversi skema database. Untuk informasi selengkapnya, lihat [Menggunakan Microsoft SQL Server sebagai sumber untuk AWS SCT](#) AWS dokumentasi.

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 migrasi data dari satu database SQL Server yang lain. Ini juga mendukung migrasi heterogen antara platform database yang berbeda, seperti memigrasi database SQL Server Anda ke database sumber terbuka 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. Untuk informasi selengkapnya, lihat [Menggunakan database Microsoft SQL Server sebagai sumber untuk AWS DMS](#) AWS dokumentasi.

Ikan babel

Babelfish adalah kemampuan bawaan Amazon Aurora. Babelfish untuk Aurora PostgreSQL memungkinkan database Edisi yang kompatibel dengan Aurora PostgreSQL Anda untuk memahami perintah dari aplikasi yang ditulis untuk Microsoft SQL Server. Memodifikasi aplikasi SQL Server yang memiliki kode database SQL Server ditulis dalam Transact-SQL (T-SQL), SQL dialek milik SQL Server, membutuhkan usaha dan memakan waktu. [Babelfish untuk Aurora PostgreSQL membuat proses ini lebih sederhana](#) dan lebih mudah. Menggunakan Babelfish, Anda tidak perlu membuat perubahan pada kode aplikasi Anda. Sebagai gantinya, Anda dapat menggunakan Babelfish untuk Aurora PostgreSQL untuk memigrasi database SQL Server ke kluster DB yang kompatibel dengan Aurora PostgreSQL.

Dengan Babelfish, Aurora PostgreSQL memahami T-SQL dan mendukung protokol komunikasi yang sama, sehingga Anda tidak perlu mengganti driver database atau menulis ulang kueri aplikasi Anda. Aplikasi Anda yang awalnya ditulis untuk SQL Server sekarang dapat bekerja dengan Aurora dengan perubahan kode yang lebih sedikit. Ini mengurangi upaya yang diperlukan untuk memodifikasi dan memindahkan aplikasi yang berjalan di SQL Server atau yang lebih baru ke Aurora, yang mengarah ke migrasi yang lebih cepat, berisiko rendah, dan lebih hemat biaya.

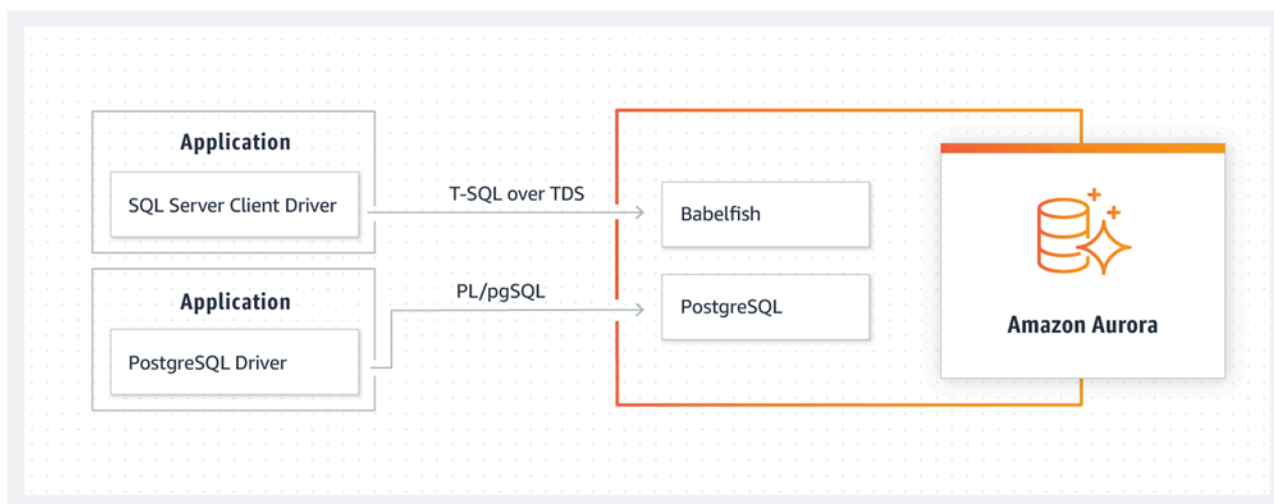
Jika Anda bermigrasi dari database SQL Server lama, Anda dapat menggunakan Babelfish untuk menjalankan kode SQL Server berdampingan dengan fungsionalitas baru yang Anda buat

menggunakan API PostgreSQL asli. Babelfish memungkinkan Aurora PostgreSQL bekerja dengan alat, perintah, dan driver SQL Server yang umum digunakan.

Babelfish juga menyediakan akses ke data dengan menggunakan koneksi PostgreSQL asli. Secara default, kedua dialek SQL yang didukung oleh Babelfish tersedia melalui protokol kawat asli mereka di port berikut:

- Untuk dialek SQL Server (T-SQL), sambungkan ke port 1433.
- Untuk dialek PostgreSQL (PL/pgSQL), sambungkan ke port 5432.

Babelfish memungkinkan aplikasi SQL Server lama Anda untuk berkomunikasi dengan Aurora tanpa penulisan ulang kode ekstensif, dengan menyediakan koneksi dari SQL Server atau port PostgreSQL. Diagram berikut menggambarkan arsitektur ini.



Anda dapat mengaktifkan Babelfish di kluster Aurora Anda dari konsol manajemen Amazon RDS. Untuk petunjuk, lihat [Membuat kluster Babelfish for Aurora PostgreSQL DB](#) dalam dokumentasi Amazon RDS.

Untuk informasi selengkapnya tentang migrasi, lihat [Memigrasi database SQL Server ke Babelfish untuk Aurora PostgreSQL dalam dokumentasi Aurora](#).

Untuk informasi tambahan, lihat sumber daya berikut:

- [Memulai dengan Babelfish untuk Aurora PostgreSQL](#) (Database blog) AWS
- [Migrasi dari SQL Server ke Amazon Aurora menggunakan Babelfish](#) (Blog database) AWS
- [Migrasi dari SQL Server ke Aurora PostgreSQL menggunakan SSIS dan Babelfish](#) (Blog database) AWS

- [Memodifikasi paket SSIS dari SQL Server ke Babelfish untuk Aurora PostgreSQL](#) (Database blog) AWS
- [Jalankan laporan Layanan Pelaporan SQL Server terhadap Babelfish untuk Aurora PostgreSQL](#) (Blog database) AWS
- [Mempersiapkan migrasi Babelfish dengan laporan AWS SCT penilaian](#) (Blog AWS database)

Skenario migrasi hibrida untuk SQL Server

Anda juga dapat menjalankan beban kerja SQL Server di lingkungan hybrid yang mencakup AWS. Misalnya, Anda mungkin sudah menjalankan SQL Server di pusat data lokal atau lokasi bersama namun ingin menggunakan AWS Cloud untuk meningkatkan arsitektur Anda untuk memberikan ketersediaan tinggi atau solusi pemulihan bencana. Anda juga dapat menggunakan solusi hibrida untuk menyimpan backup SQL Server jangka panjang pada AWS, untuk memutar kembali migrasi Anda, jika terjadi masalah, atau menjalankan replika sekunder menggunakan grup ketersediaan SQL Server Always On AWS Cloud SQL Server memiliki beberapa teknologi replikasi yang menawarkan ketersediaan tinggi dan solusi pemulihan bencana.

Mencadangkan database SQL Server Anda ke AWS Cloud

Amazon Simple Storage Service (Amazon S3) memungkinkan Anda memanfaatkan fleksibilitas dan harga penyimpanan cloud. Ini memberi Anda kemampuan untuk mencadangkan database SQL Server Anda ke sistem penyimpanan yang aman, sangat tersedia, sangat tahan lama, dan andal. Anda dapat menyimpan cadangan SQL Server Anda dengan aman di Amazon S3 Anda juga dapat menggunakan kebijakan Siklus Hidup Amazon S3 untuk menyimpan cadangan Anda dalam jangka panjang. Amazon S3 memungkinkan Anda menyimpan data dalam jumlah besar dengan biaya yang sangat rendah. Anda dapat menggunakan [AWS DataSync](#) untuk mentransfer file cadangan ke Amazon S3

Anda dapat menggunakan Storage Gateway untuk menyimpan cadangan SQL Server lokal dan mengarsipkan data di Amazon S3 atau Amazon S3 Glacier. Anda dapat membuat volume penyimpanan cache dan memasangnya sebagai perangkat Internet Small Computer System Interface (iSCSI) dari server aplikasi cadangan lokal Anda. Semua data ditransfer dengan aman ke AWS melalui SSL dan disimpan dalam format terenkripsi di Amazon S3 Menggunakan volume cache gateway menghemat biaya di muka pemeliharaan dan penskalaan perangkat keras penyimpanan mahal di tempat. Jika Anda ingin menyimpan data utama atau cadangan di lokasi, Anda dapat menggunakan volume yang tersimpan gateway untuk menyimpan data ini secara lokal, dan mencadangkan data di luar situs ke Amazon S3.

Memperluas ketersediaan tinggi dan solusi pemulihan bencana

Anda dapat memperluas praktik ketersediaan tinggi di tempat yang ada dan memberikan solusi pemulihan bencana AWS dengan menggunakan fitur pengiriman log asli di SQL Server. Anda dapat

mentransfer log transaksi SQL Server dari pusat data lokal atau lokasi bersama ke instans SQL Server yang berjalan pada instans EC2 atau instans Amazon RDS for SQL Server DB di awan pribadi virtual (VPC). Anda dapat mengirimkan data ini dengan aman melalui koneksi jaringan khusus dengan menggunakan AWS Direct Connect, atau mengirimkannya melalui terowongan VPN yang aman. Pencadangan log transaksi dikirim ke instans EC2, dan mereka diterapkan ke instans database sekunder.

Anda dapat menggunakan AWS Cloud menyediakan tingkat ketersediaan tinggi dan pemulihan bencana yang lebih tinggi dengan menggunakan grup ketersediaan SQL Server Always On antara pusat data lokal Anda dan Amazon EC2. Hal ini dapat dilakukan dengan memperluas pusat data Anda ke VPC pada AWS dengan menggunakan koneksi jaringan khusus seperti AWS Direct Connect, atau dengan menetapkan terowongan VPN yang aman di antara kedua lingkungan ini.

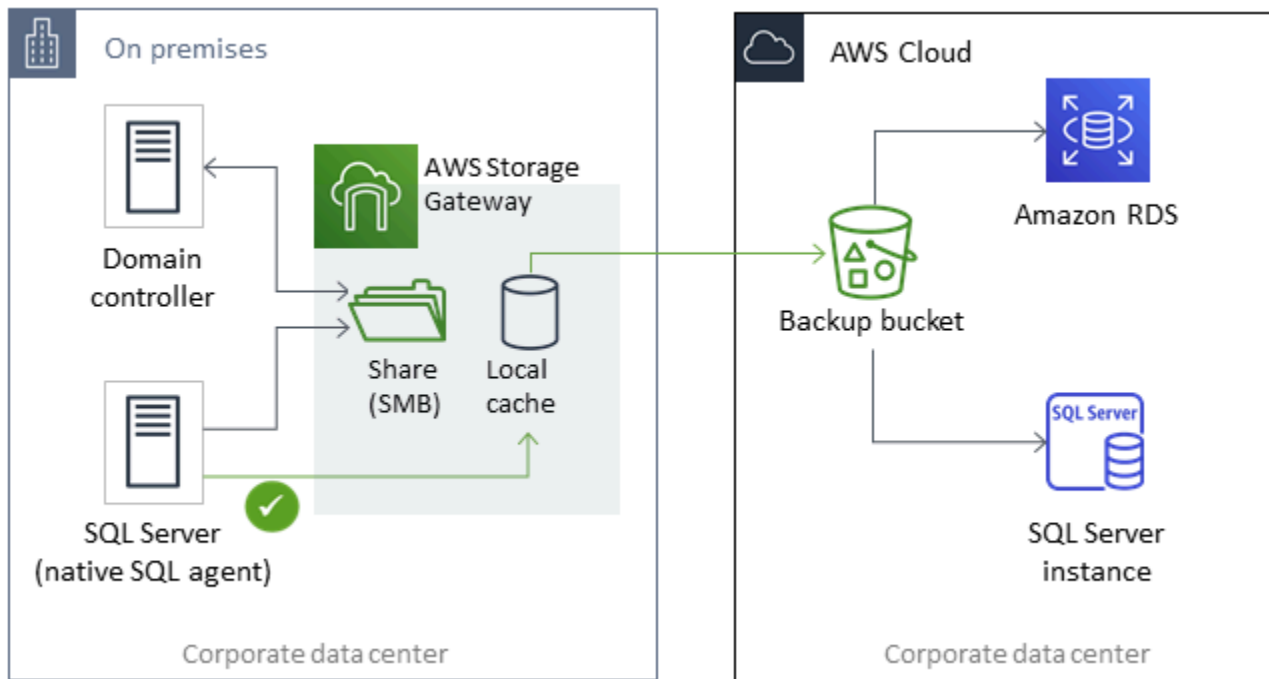
Berikut adalah beberapa hal yang perlu dipertimbangkan ketika merencanakan implementasi hibrida SQL Server Selalu Pada kelompok ketersediaan:

- Buat koneksi jaringan yang aman, andal, dan konsisten antara lingkungan lokal Anda dan AWS lewat AWS Direct Connect atau VPN
- Buat VPC menggunakan layanan Amazon Virtual Private Cloud (Amazon VPC) Gunakan tabel rute dan grup keamanan Amazon VPC untuk mengaktifkan komunikasi yang sesuai antara kedua lingkungan tersebut.
- Perluas domain Active Directory ke VPC dengan menerapkan kontroler domain sebagai instans EC2, atau dengan menggunakan AWS Directory Service for Microsoft Active Directory. Anda juga dapat menggunakan AWS Managed Microsoft AD untuk Amazon RDS for SQL Server Untuk informasi selengkapnya, lihat [Dokumentasi Amazon RDS](#).

Storage Gateway

Storage Gateway memungkinkan Anda untuk menyimpan dan mengambil file dengan menggunakan Server Message Block (SMB) berbagi untuk Windows. Anda dapat bergabung dengan gateway penyimpanan ke domain Active Directory on premise. Dengan memiliki database SQL Server dan gateway penyimpanan di domain yang sama, Anda dapat mengambil backup langsung ke berbagi jaringan SMB alih-alih menyimpannya secara lokal dan kemudian mengunggahnya ke berbagi jaringan. Gateway penyimpanan dikonfigurasi untuk menggunakan bucket S3, sehingga semua cadangan Anda akan tersedia di bucket S3 pada AWS. Anda dapat memulihkan database dengan mengunduh file cadangan ke SQL Server pada instans EC2, atau memulihkan database langsung ke Amazon RDS.

Diagram berikut menunjukkan cara menyimpan dan mengakses cadangan dengan menggunakan Storage Gateway dan Amazon S3. Untuk informasi selengkapnya, lihat [Dokumentasi Storage Gateway](#).



Menggunakan AWS DMS dan AWS SCT

Anda dapat menggunakan AWS DMS di lingkungan SQL Server hybrid, untuk memigrasikan data dari database lokal ke cloud, atau sebaliknya. Anda dapat memigrasi database SQL Server Anda ke MySQL atau PostgreSQL dengan menggunakan AWS DMS bersama AWS SCT. Untuk langkah-langkah migrasi, lihat [AWS SCT dokumentasi](#). Sebelum memigrasikan data, Anda dapat menjalankan [laporan penilaian migrasi](#) yang menandai setiap pekerjaan manual tambahan yang mungkin diperlukan.

Anda juga dapat menggunakan AWS DMS untuk replikasi yang sedang berlangsung (perubahan data capture atau CDC). Untuk informasi selengkapnya, lihat [Menggunakan replikasi berkelanjutan \(CDC\) dari sumber SQL Server di AWS DMS dokumentasi](#)

Memodernasikan basis data SQL Server Anda

Bagian ini menjelaskan cara memodernisasi beban kerja SQL Server Anda AWS dengan beralih dari sistem operasi Windows ke Linux. Perubahan ini memungkinkan Anda memanfaatkan teknologi sumber terbuka dan menghemat biaya lisensi Windows tanpa mengubah arsitektur sistem Anda secara drastis atau melatih ulang pengguna Anda.

Migrasikan beban kerja SQL Server Anda dari Windows ke Linux

Dimulai dengan SQL Server 2017, SQL Server tersedia untuk dijalankan pada sistem operasi Linux. Memindahkan beban kerja SQL Server Anda ke Linux menyediakan penghematan biaya dan peningkatan kinerja.

Hampir semua fungsi SQL Server, aplikasi, pernyataan, dan skrip yang Anda gunakan di Microsoft Windows juga didukung di Linux. Anda juga dapat menggunakan alat seperti SQL Server Management Studio (SSMS), SQL Server Data Tools (SSDT), dan modul PowerShell (sqlps) untuk mengelola SQL Server di Linux dari instance Windows.

Anda dapat menggunakan salah satu dari tiga opsi ini untuk memigrasikan beban kerja SQL Server Anda ke Linux:

- Asli SQL Server backup dan restore fitur (lihat [Dokumentasi Microsoft SQL Server](#))
- Grup ketersediaan terdistribusi (untuk mengubah sistem operasi Anda saat Anda bermigrasi ke AWS)
- Parameter AWS replatforming asisten, yang merupakan alat scripting berbasis PowerShell

Parameter AWS asisten replatform membantu Anda bermigrasi dari beban kerja SQL Server yang ada dari Windows ke sistem operasi Linux. Saat Anda menjalankan skrip PowerShell untuk asisten replatform pada basis data SQL Server sumber, instans Windows mencadangkan basis data ke bucket penyimpanan Amazon S3 yang dienkrpsi. Kemudian mengembalikan cadangan ke database SQL Server baru atau yang sudah ada pada instance EC2 Linux. Anda dapat mereplikasi database Anda dan menguji aplikasi Anda sementara database SQL Server sumber Anda tetap online. Setelah pengujian, Anda dapat menjadwalkan waktu henti aplikasi dan menjalankan kembali skrip cadangan PowerShell untuk melakukan cutover terakhir Anda.

Untuk informasi selengkapnya tentang cara menggunakan asisten replatform, lihat [Migrasikan Beban Kerja SQL Server on-premise Anda ke Amazon EC2 Linux](#) pada [AWS Database blog](#), dan [Dokumentasi Amazon EC2](#).

Ketersediaan tinggi di Linux

SQL Server 2017 mendukung Selalu Pada kelompok ketersediaan antara Windows dan Linux untuk membuat beban kerja skala baca tanpa ketersediaan tinggi. Sayangnya, Anda tidak dapat mencapai ketersediaan tinggi antara Windows dan Linux, karena tidak ada solusi berkelompok yang dapat mengelola konfigurasi cross-platform tersebut.

Untuk menggunakan ketersediaan tinggi dengan grup ketersediaan Always On, pertimbangkan untuk menggunakan Windows Server Failover Cluster (WSFC) atau Pacemaker di Linux. Solusi ini cocok untuk jalur migrasi dari SQL Server di Windows ke Linux dan sebaliknya, atau untuk pemulihan bencana menggunakan failover manual. Untuk informasi selengkapnya tentang skenario ini, lihat [Menyebarkan grup ketersediaan Always On antara instans Amazon EC2 Windows dan Amazon Linux 2](#) pada [AWS Blog](#) basis data.

AWSLaunch Wizard untuk SQL Server

AWSLaunch Wizard adalah layanan yang memandu Anda melalui ukuran, konfigurasi, dan penyebaran Microsoft SQL Server di Amazon EC2. Ini mendukung penerapan instans tunggal SQL Server dan ketersediaan tinggi (HA) di Amazon EC2.

Launch Wizard adalah layanan gratis. Anda hanya membayar untuk AWS sumber daya yang disediakan untuk menjalankan aplikasi Anda, seperti sumber daya Amazon EC2, Amazon EBS, dan Amazon VPC.

Anda memasukkan persyaratan aplikasi, termasuk kinerja, jumlah node, dan konektivitas, pada konsol Launch Wizard. Launch Wizard mengidentifikasi hak AWS sumber daya untuk menyebarkan dan menjalankan aplikasi SQL Server Anda. Ini juga menyediakan perkiraan biaya penyebaran, dan Anda dapat memodifikasi sumber daya Anda dan langsung melihat penilaian biaya yang diperbarui. Ketika Anda mengkonfirmasi pilihan Anda dan memulai penyebaran, Luncurkan ketentuan Wizard dan mengkonfigurasi sumber daya yang dipilih dalam beberapa jam untuk membuat aplikasi SQL Server produksi yang berfungsi penuh. Anda dapat mengakses aplikasi SQL Server yang digunakan dari konsol Amazon EC2.










Berikut adalah beberapa manfaat menggunakan Launch Wizard untuk SQL Server:

- **Deployment sederhana**— Anda dapat menyederhanakan penyediaan sumber daya SQL Server Anda di AWS dengan menjawab pertanyaan berdasarkan kebutuhan Anda. Penyebaran Launch Wizard lebih cepat daripada penggunaan manual, sehingga menghilangkan waktu untuk menyediakan dan mengkonfigurasi aplikasi Anda di AWS.
- **Ukuran otomatis dan estimasi biaya**— Launch Wizard menyediakan built-in pilihan contoh berdasarkan kebutuhan Anda. Ini memilih jenis instans, volume EBS, dan sumber daya lain yang paling sesuai dengan persyaratan SQL Server Anda. Launch Wizard juga menyediakan Anda dengan perkiraan biaya sebelum ketentuan AWS sumber daya.
- **Penghematan waktu dengan template otomatisasi berulang**— Anda dapat redeploy SQL Server dengan dapat digunakan kembali AWS CloudFormation template yang dibuat oleh Launch Wizard. Template ini berfungsi sebagai dasar dan menghemat waktu Anda.

Launch Wizard mendukung sistem operasi berikut, versi SQL Server, dan fitur. Untuk informasi terbaru, lihat [AWS Luncurkan dokumentasi Wizard](#).

Kategori	Gunakan kasus atau fitur	Luncurkan dukungan Wizard	Mulai cepat dukungan	Dukungan konsol Amazon EC2
Deployment pada Windows	Penyebaran simpul SQL tunggal			
	Deployment: Selalu Pada kelompok ketersediaan			
	Deployment: Selalu Pada Instans Kluster Failover (FCIs) dengan FSx for Windows File Server			
	Deployment: Host Khusus			
	Template kode yang dapat digunakan kembali			
	Deployment di Linux	Penyebaran simpul SQL tunggal		
	Penyebaran HA di Ubuntu			

Kategori	Gunakan kasus atau fitur	Luncurkan dukungan Wizard	Mulai cepat dukungan	Dukungan konsol Amazon EC2	
	Template kode yang dapat digunakan kembali				Tidak
Ukuran	Rekomendasi jenis instans				Tidak
	Perkiraan biaya				Tidak
Konfigurasi	Dibuat secara otomatisA WSKelompok sumber daya Systems Manager				Tidak
	Pemberitahuan Amazon SNS				Tidak
	Pemantauan Amazon CloudWatch				Tidak

Kategori	Gunakan kasus atau fitur	Luncurkan dukungan Wizard	Mulai cepat dukungan	Dukungan konsol Amazon EC2	
	Menghubungkan ke Active Directory yang ada (baik lokal maupun terkelola)		y: 	T 	Tidak
	Validasi input awal		y: 	T 	Tidak
	Kebijakan IAM terkelola		y: 	T 	Tidak

Untuk informasi selengkapnya tentang Launch Wizard untuk SQL Server, lihat yang berikut ini:

- [AWSLaunch Wizard untuk dokumentasi SQL Server](#)
- [Menyederhanakan SQL Server Selalu Pada penyebaran denganAWSLaunch Wizard dan Amazon FSxposting blog](#)
- [Mempercepat SQL Server Selalu Pada Deployment denganAWSLaunch Wizardposting blog](#)

Praktik terbaik untuk bermigrasi ke Amazon RDS untuk SQL Server

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

Important

Pastikan Anda memiliki paket rollback sebelum memigrasi database.

Note

Anda dapat menggunakan Migration Hub Orchestrator untuk mengotomatiskan dan mengatur migrasi database SQL Server Anda ke Amazon EC2 atau Amazon RDS dengan menggunakan pencadangan dan pemulihan asli. Untuk informasi lebih lanjut, lihat [Orkestrator AWS Migration Hub bagian](#).

Menyediakan basis data target Anda

Setelah Anda selesai menilai, merencanakan, dan menyiapkan strategi migrasi database, ikuti praktik terbaik ini saat menyediakan database Amazon RDS untuk SQL Server:

- Ukuran kanan instans Amazon RDS for SQL Server DB berdasarkan kebutuhan Anda untuk CPU, memori, IOPS, dan jenis penyimpanan. (Jika Anda menggunakan SQL Server Standard edition, menyediakan CPU dan memori dalam batasan edisi Standard.)
- Atur zona waktu dan pemeriksaan 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, menyediakan contoh SQL Server dengan versi terbaru dari SQL Server.

- Buat grup opsi dan grup parameter terpisah untuk setiap database Amazon RDS.
- Kumpulkan dan ekstrak login, pengguna, dan peran untuk migrasi.
- Tinjau pekerjaan Agen SQL Server untuk pemeliharaan dan aplikasi yang perlu dimigrasi.

Mencadangkan dari database sumber Anda

Ada banyak alat untuk memigrasi database SQL Server ke database Amazon RDS untuk SQL Server. Metode yang paling umum digunakan adalah menggunakan cadangan dan pemulihan asli SQL Server jika kebutuhan Anda mengizinkan waktu henti.

Jika Anda memiliki waktu henti terbatas, Anda dapat menggunakan cadangan/pemulihan SQL Server asli dengan cadangan diferensial dan cadangan log. Atau Anda dapat menggunakan AWS DMS, yang menyediakan tiga opsi: full-load, full-load dan CDC, atau CDC saja.

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 cadangan SQL Server ke Amazon S3 dan mengatur integrasi Amazon S3.](#)
- Jika Anda tidak memiliki bandwidth tinggi AWS Direct Connect, gunakan AWS Snowball untuk mentransfer file cadangan database besar. Anda juga dapat menggunakan AWS DMS untuk mentransfer data saat replikasi diperlukan.

Memulihkan data ke database target Anda

- Jika Anda memigrasi database yang sangat besar, sebaiknya Anda menyediakan [jenis instans Amazon RDS](#) yang lebih besar pada awalnya, selama durasi migrasi, untuk memuat data yang lebih cepat.
- Nonaktifkan Multi-AZ. (Ini dapat diaktifkan kembali setelah migrasi.)
- Nonaktifkan retensi cadangan. (Ini dapat diaktifkan kembali setelah migrasi.)
- Kembalikan database dengan menggunakan perintah pemulihan SQL Server asli.
- Buat login dan pengguna, dan perbaiki pengguna yatim piatu, jika diperlukan.
- Buat pekerjaan Agen SQL Server dan tinjau jadwal, sesuai kebutuhan.

Langkah pasca-migrasi

Setelah migrasi selesai, Anda dapat:

- Ubah instans DB ke tipe instans berukuran kanan.
- Aktifkan Multi-AZ dan retensi cadangan.
- Pastikan bahwa semua pekerjaan dibuat pada node sekunder (untuk konfigurasi Multi-AZ).
- Publikasikan error SQL Server dan log agen ke Amazon CloudWatch Logs, dan gunakan CloudWatch untuk melihat metrik dan membuat alarm. Untuk informasi selengkapnya, lihat [dokumentasi Amazon RDS](#).
- Aktifkan [pemantauan yang disempurnakan](#) untuk mendapatkan metrik instans DB Anda secara real time.
- Siapkan topik Amazon Simple Notification Service (Amazon SNS) untuk peringatan.

Menguji migrasi

Kami merekomendasikan pengujian berikut untuk memvalidasi aplikasi Anda terhadap database Amazon RDS for SQL Server baru Anda:

- Lakukan pengujian fungsional.
- Bandingkan kinerja kueri SQL di database sumber dan target Anda, dan sesuaikan kueri sesuai kebutuhan. Beberapa kueri mungkin bekerja lebih lambat dalam database target, jadi sebaiknya Anda menangkap baseline kueri SQL di database sumber.

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, pemulihan, dan pemulihan database untuk memastikan bahwa Anda memenuhi persyaratan RPO dan RTO.
- Cantumkan semua pekerjaan dan laporan penting, dan jalankan di Amazon RDS untuk mengevaluasi kinerja mereka terhadap perjanjian tingkat layanan (SLA) Anda.

Mengoperasikan dan mengoptimalkan database Amazon RDS Anda

Saat database Anda aktif AWS, pastikan Anda mengikuti praktik terbaik di area seperti pemantauan, peringatan, pencadangan, dan ketersediaan tinggi di cloud. Misalnya:

- Siapkan CloudWatch pemantauan, dan aktifkan pemantauan terperinci.
- Gunakan [Amazon RDS Performance Insights](#) dan solusi pemantauan pihak ketiga lainnya seperti [SentryOne](#) atau [Foglight for SQL Server untuk](#) memantau database Anda.
- Siapkan peringatan dengan menggunakan topik SNS.
- Siapkan cadangan otomatis dengan menggunakan [AWS Backup](#) atau cadangan SQL Server asli, dan salin ke Amazon S3.
- Untuk ketersediaan tinggi, siapkan fitur Multi-AZ Amazon RDS.
- Jika Anda memerlukan basis data hanya-baca, [siapkan replika baca dalam wilayah yang](#) sama atau di seluruh AWS Wilayah sesuai dengan kebutuhan Anda.

Memilih antara Amazon EC2 dan Amazon RDS





Amazon EC2 dan Amazon RDS menawarkan manfaat unik yang mungkin bermanfaat untuk kasus penggunaan spesifik Anda. Anda memiliki fleksibilitas untuk menggunakan salah satu atau kedua layanan untuk database SQL Server Anda, tergantung pada kebutuhan Anda. Bagian ini memberikan informasi terperinci untuk membantu pilihan Anda.










Matriks keputusan







Tabel berikut menyediakan side-by-side perbandingan fitur SQL Server yang didukung di Amazon RDS, Amazon RDS Custom untuk SQL Server, dan Amazon EC2. Gunakan informasi ini untuk memahami perbedaan mereka dan untuk memilih pendekatan terbaik untuk kasus penggunaan Anda.


Untuk informasi terbaru untuk Amazon RDS, lihat [Microsoft SQL Server di Amazon RDS dalam dokumentasi](#). AWS








Development










Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
Ekstensi kolom penyangga		T 	Y 	Jika fitur ini sangat penting untuk beban kerja Anda, pertimbangkan untuk memilih Amazon RDS Custom atau Amazon EC2.
BULK INSERT		Y 	Y 	Lihat Mengintegrasikan instans Amazon RDS for SQL Server DB dengan




Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
				Amazon S3 dalam dokumentasi Amazon RDS.
Change Data Capture (CDC)	 (Edisi Perusahaan: semua versi; Edisi Standar: 2016 SP1 dan yang lebih baru)	Y 	Y 	Lihat Menggunakan pengambilan data perubahan dalam dokumentasi Amazon RDS.
Ubah pelacakan		Y 	Y 	Y
Indeks Columnstore	 (Edisi Perusahaan: 2014 dan yang lebih baru)	Y  (Edisi Perusahaan: 2019)	Y  (Edisi Perusahaan: 2014 dan yang lebih baru)	Y




Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
Layanan Kualitas Data		T 	Y 	Y Jika fitur ini sangat penting untuk beban kerja Anda, pertimbangkan untuk memilih Amazon RDS Custom atau Amazon EC2.
Database Mail		Y 	Y 	Y Lihat posting blog Menggunakan Database Mail di Amazon RDS for SQL Server . Kami mendorong Anda untuk menggunakan Amazon Simple Email Service (Amazon SES) untuk mengirim email keluar yang berasal AWS dari sumber daya, untuk memastikan tingkat pengiriman yang tinggi.










Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
Database Engine Tuning Advisor		Y 	Y 	Y
Pemberitahuan acara DB		Y 	Y  (melacak dan mengelola acara DB secara manual)	T Lihat Menggunakan an notifikasi peristiwa Amazon RDS dalam dokumentasi Amazon RDS.
Pemberitahuan acara DDL		T 	Y 	Y Jika fitur ini sangat penting untuk beban kerja Anda, pertimbangkan untuk memilih Amazon RDS Custom atau Amazon EC2.
Daya tahan transaksi tertunda (lazy commit)	 (SQL Server 2014 dan yang lebih baru)	Y  (SQL Server 2019)	Y  (SQL Server 2014 dan yang lebih baru)	Y








Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
Kueri terdistribusi	 (target SQL Server)	Y  (target SQL Server)	Y  (target SQL Server)	Lihat posting blog Menerapkan server tertaut dengan Amazon RDS for SQL Server.
Acara yang diperpanjang		Y 	Y 	
Prosedur tersimpan yang diperluas, termasuk xp_cmdshell		T 	Y 	Jika fitur ini sangat penting untuk beban kerja Anda, pertimbangkan untuk memilih Amazon RDS Custom atau Amazon EC2.
Tabel file		T 	Y 	Jika fitur ini sangat penting untuk beban kerja Anda, pertimbangkan untuk memilih Amazon RDS Custom atau Amazon EC2.













Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
FILESTREAM		T 	Y 	FILESTREAM tidak kompatibel dengan Amazon RDS. Namun, Anda dapat mengonfigurasi database dalam memori.
Pencarian teks lengkap	 (kecuali pencarian semantik)	Y 	Y 	Y
Database dalam memori	 (SQL Server 2014 dan yang lebih baru)	Y  (SQL Server 2019)	Y  (SQL Server 2014 dan yang lebih baru)	Y

Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
Server tertaut (target SQL Server dan Oracle)		Y 	Y 	Lihat Menerapkan server tertaut dengan Amazon RDS for SQL Server posting blog dan Support untuk server tertaut dengan Oracle OLEDB di Amazon RDS for SQL Server dalam dokumentasi Amazon RDS .



















Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
Layanan Machine Learning (dengan skrip R)		Y 	Y 	<p>Machine Learning Services harus diinstal secara terpisah pada mesin Windows atau Linux. Ini didukung pada Instans Cluster Always On Failover (FCI) hanya di SQL Server 2019 dan yang lebih baru.</p> <p>Meskipun R tidak didukung di Amazon RDS, Anda dapat menggunakannya di AWS (lihat posting blog Memulai dengan R on AWS).</p>

Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
Rencana pemeliharaan		T 	Y 	Y Amazon RDS menyediakan serangkaian fitur terpisah untuk memfasilitasi pencadangan dan pemulihan database. Untuk cadangan, Anda dapat mengonfigurasi pencadangan otomatis.
Layanan Data Master		T 	Y 	Y Jika fitur ini sangat penting untuk beban kerja Anda, pertimbangkan untuk memilih Amazon RDS Custom atau Amazon EC2.
Microsoft Distributed Transaction Coordinator (MSDTC)		Y 	Y 	Y Lihat posting blog Mengaktifkan dukungan transaksi terdistribusi untuk instans Amazon RDS for SQL Server yang bergabung dengan domain.

Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
OPENROWSET	 Y	 Y	 Y	
Sebagian berisi database	 (SQL Server 2014 dan yang lebih baru)	 (SQL Server 2019)	 (SQL Server 2014 dan yang lebih baru)	
Pengumpul Data Performa	 T	 Y	 Y	Di Amazon RDS, Anda dapat menggunakan Amazon CloudWatch, AWS CloudTrail, dan Performance Insights untuk memantau kinerja SQL Server Anda (lihat Ikhtisar pemantauan Amazon RDS dalam dokumentasi Amazon RDS).

Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
Manajemen Berbasis Kebijakan		T 	Y 	Jika fitur ini sangat penting untuk beban kerja Anda, pertimbangkan untuk memilih Amazon RDS Custom atau Amazon EC2.
PolyBase		T 	Y 	Jika fitur ini sangat penting untuk beban kerja Anda, pertimbangkan untuk memilih Amazon RDS Custom atau Amazon EC2.
Parameter yang telah dikonfigurasi sebelumnya		Y 	T 	
Resource Governor		T 	Y 	Jika fitur ini sangat penting untuk beban kerja Anda, pertimbangkan untuk memilih Amazon RDS Custom atau Amazon EC2.






Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
CLR Aman	 (SQL Server 2014 dan 2016)	Y 	Y 	Y
Urutan	 (SQL Server 2014 dan yang lebih baru)	Y  (SQL Server 2019)	Y  (SQL Server 2014 dan yang lebih baru)	Y
Pemicu tingkat server		T 	Y 	Jika fitur ini sangat penting untuk beban kerja Anda, pertimbangkan untuk memilih Amazon RDS Custom atau Amazon EC2.
Pialang Layanan	 (kecuali titik akhir)	Y 	Y 	Y
Fitur spasial dan lokasi		Y 	Y 	Y





Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
Agen SQL Server		Y 	Y 	Y
SQL Server Analysis Services (SSAS)	 (SQL Server 2016 dan yang lebih baru)	Y 	Y 	Lihat Support untuk SSAS di Amazon RDS for SQL Server dalam dokumentasi Amazon RDS.
SQL Server Integration Services (SSIS)	 (SQL Server 2016 dan yang lebih baru)	Y 	Y 	Lihat Support untuk SSIS di Amazon RDS for SQL Server dalam dokumentasi Amazon RDS.
Studio Manajemen SQL Server (SSMS)		Y 	Y 	Y
Asisten Migrasi SQL Server (SSMA)		Y 	Y 	Y
SQL Server Profiler	 (jejak sisi server dan sisi klien)	Y 	Y 	Y


Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
Layanan Pelaporan SQL Server (SSRS)	 (SQL Server 2016 dan yang lebih baru)	Y 	Y 	Lihat Support untuk SSRS di Amazon RDS for SQL Server dalam dokumentasi Amazon RDS.
sqlcmd		Y 	Y 	
Database Peregangan		T 	Y 	Jika fitur ini sangat penting untuk beban kerja Anda, pertimbangkan untuk memilih Amazon RDS Custom atau Amazon EC2.
Pernyataan THROW	 (SQL Server 2014 dan yang lebih baru)	Y  (SQL Server 2019)	Y  (SQL Server 2014 dan yang lebih baru)	

Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
Titik akhir transaksi-SQL		T 	Y 	Y Semua operasi yang digunakan tidak CREATE ENDPOINT tersedia di Amazon RDS. Kami menyarankan Anda menginstall SQL Server pada instans EC2 untuk operasi ini.
Dukungan UTF-16	 (SQL Server 2014 dan yang lebih baru)	Y 	Y  (SQL Server 2014 dan yang lebih baru)	Y
Layanan Data WCF		T 	Y 	Y Jika fitur ini sangat penting untuk beban kerja Anda, pertimbangkan untuk memilih Amazon RDS Custom atau Amazon EC2.




HA/DR

Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
Selalu Pada grup ketersediaan		Y  (sinkron dan asinkron)	Y 	Jika Anda memerlukan grup ketersediaan Selalu Aktif yang dikelola sendiri, sebaiknya gunakan AWS Launch Wizard untuk menyederhanakan penerapan SQL Server HA pada instans EC2. Lihat AWS Launch Wizard SQL Server dalam AWS dokumentasi.
Selalu Pada Instans Cluster Failover (FCI)		T 	Y 	Anda dapat menggunakan AWS Launch Wizard untuk menyederhanakan penyebaran SQL Server FCI Anda di Amazon EC2. Lihat AWS Launch Wizard








Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
				SQL Server dalam AWS dokumentasi.
Mencadangkan ke Amazon S3		Y 	Y 	Y Amazon RDS mendukung pencadangan dan pemulihan asli untuk database SQL Server dengan menggunakan file cadangan lengkap (file.bak) dan Amazon S3 sebagai repositori. Lihat Mengimpor dan Mengekspor database SQL Server dalam dokumentasi Amazon RDS.
Perintah BACKUP		T 	Y 	Y Lihat Bagaimana cara melakukan backup asli instans Amazon RDS DB yang menjalankan SQL Server? di Pusat AWS Pengetahuan.



Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
Pencerminan basis data	 (Multi-AZ)	Y 	Y 	Y
Replikasi database	 (berlangganan push terbatas)	T 	Y 	Y Jika Anda ingin mereplikasi satu tabel di Amazon RDS, Anda juga dapat menggunakan AWS DMS atau mengatur replika baca.
Grup ketersediaan terdistribusi		T 	Y 	Y Jika fitur ini sangat penting untuk beban kerja Anda, pertimbangkan untuk memilih Amazon RDS Custom atau Amazon EC2.
Pengiriman log		T 	Y 	Y Untuk tujuan pemulihan bencana, Anda dapat menggunakan replika baca atau AWS DMS .







Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
Pencadangan otomatis terkelola		Y 	Y  (memerlukan konfigurasi dan pengelolaan rencana pemeliharaan, atau menggunakan solusi pihak ketiga)	Lihat Bekerja dengan cadangan di dokumentasi Amazon RDS .
Multi-AZ dengan failover otomatis		Y  (dengan konfigurasi manual grup ketersediaan Selalu Aktif)	Y  (Hanya Edisi Perusahaan, dengan konfigurasi manual grup ketersediaan Selalu Aktif)	Lihat Penerapan multi-AZ untuk Amazon RDS for SQL Server dalam dokumentasi Amazon RDS .
Replika baca	 (SQL Server 2016 dan yang lebih baru)	Y  (dengan konfigurasi manual grup ketersediaan Selalu Aktif)	Y  (dengan konfigurasi manual grup ketersediaan Selalu Aktif)	Y

Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
Perintah RESTORE		Y 	Y 	Lihat Pusat AWS Penge-tahuan .







Scalability

Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
Pemantaua-n dan metrik instans dan database bawaan		Y 	T  (ekspor metrik Anda sendiri ke CloudWatc-h atau gunakan solusi pihak ketiga)	Lihat posting blog Pantau database SQL Server Anda dengan menggunakan metrik khusus dengan Amazon CloudWatch dan AWS Systems Manager .
Ukuran penyimpanan yang dapat dikonfigurasi		Y 	Y 	Y
Jumlah maksimum database per instance	Tergantung pada ukuran instans dan konfigurasi Multi-AZ	SQL Server maksimum (5000)	 ada batasan	Lihat Spesifika-si kapasitas maksimum untuk SQL Server dalam dokumentasi




Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
				Microsoft SQL Server.
Ukuran penyimpanan maksimum instans DB	16 TiB	16 TiB	 ada batasan	T Amazon RDS juga mendukung database tempdb pada disk lokal dengan menggunakan penyimpanan instans Non-Volatile Memory Express (NVMe). Lihat Dukungan penyimpanan instans untuk database tempdb di Amazon RDS for SQL Server dalam dokumentasi Amazon RDS.
Ukuran penyimpanan minimum dari instans DB	20 GiB (Edisi Perusahaan, Standar, Web, dan Ekspres)	20 GiB (Edisi Perusahaan, Standar, Web, dan Ekspres)	 ada batasan	T







Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
Pengoptimal Kueri Baru	 (SQL Server 2014 dan yang lebih baru)	Y  (SQL Server 2014 dan yang lebih baru)	Y  (SQL Server 2014 dan yang lebih baru)	Y
Replika baca	 (SQL Server 2016 dan yang lebih baru)	Y  (dengan konfigurasi manual grup ketersediaan Selalu Aktif)	Y  (dengan konfigurasi manual grup ketersediaan Selalu Aktif)	Y

Security







Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
Penambalan perangkat lunak otomatis		Y 	T 	T
Penyimpanan terenkripsi menggunakan AWS KMS	 (semua edisi SQL Server kecuali Express)	Y 	Y 	Lihat posting blog Mengamankan data di Amazon RDS menggunakan an AWS KMS enkripsi.

Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
Peran server yang fleksibel	 (SQL Server 2014 dan yang lebih baru)	Y  (SQL Server 2019)	Y  (SQL Server 2014 dan yang lebih baru)	Y
Otentikasi SQL		Y 	Y 	Y
Audit SQL Server		Y 	Y 	Y
SSL (enkripsi dalam perjalanan)		Y 	Y 	Y Lihat Menggunakan SSL dengan instans Microsoft SQL Server DB dalam dokumentasi Amazon RDS.




Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
peran sysadmin		T 	Y 	<p>Untuk peran tingkat server yang tidak didukung, lihat Keamanan Microsoft SQL Server dalam dokumentasi Amazon RDS.</p> <p>Saat Anda membuat instans RDS DB baru, pengguna master default yang Anda gunakan mendapatkan hak istimewa tertentu untuk instans DB tersebut (lihat Hak istimewa Akun dalam dokumentasi Amazon RDS).</p>

Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
TDE (enkripsi saat istirahat)	 (Edisi Perusahaan: 2014-2019; Edisi Standar: 2019)	Y  (SQL Server 2019 Edisi Perusahaan, Standar, Web, dan Pengembangan)	Y  (Edisi Perusahaan: 2014-2019; Edisi Standar: 2019)	Y Lihat informasi tentang dukungan TDE di dokumentasi Kustom Amazon RDS dan Amazon RDS.
Otentikasi Windows		Y 	Y 	Y

Other features

Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
Kemampuan untuk menginstal agen pihak ketiga		T 	Y 	Y
Kemampuan untuk mengganti nama database yang ada	 (hanya Single-AZ)	Y  (tidak tersedia untuk database dalam grup ketersediaan atau diaktifkan)	Y  (tidak tersedia untuk database dalam grup ketersediaan atau diaktifkan)	Y Untuk penerapan multi-AZ di Amazon RDS, lihat Mengganti nama database Microsoft SQL Server dalam penerapan

Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
		untuk pencerminan)	untuk pencerminan)	Multi-AZ dalam dokumentasi Amazon RDS.
Kontrol atas instans DB dan sistem operasi		T 	Y 	Y Jika fitur ini sangat penting untuk beban kerja Anda, pertimbangkan untuk memilih Amazon RDS Custom atau Amazon EC2.
Zona waktu yang diatur khusus		Y 	Y 	Y
Replay Terdistribusi		T 	Y 	Y Layanan klien SQL Server Distributed Replay memerlukan izin sysadmin , itulah sebabnya layanan ini tidak didukung di Amazon RDS.

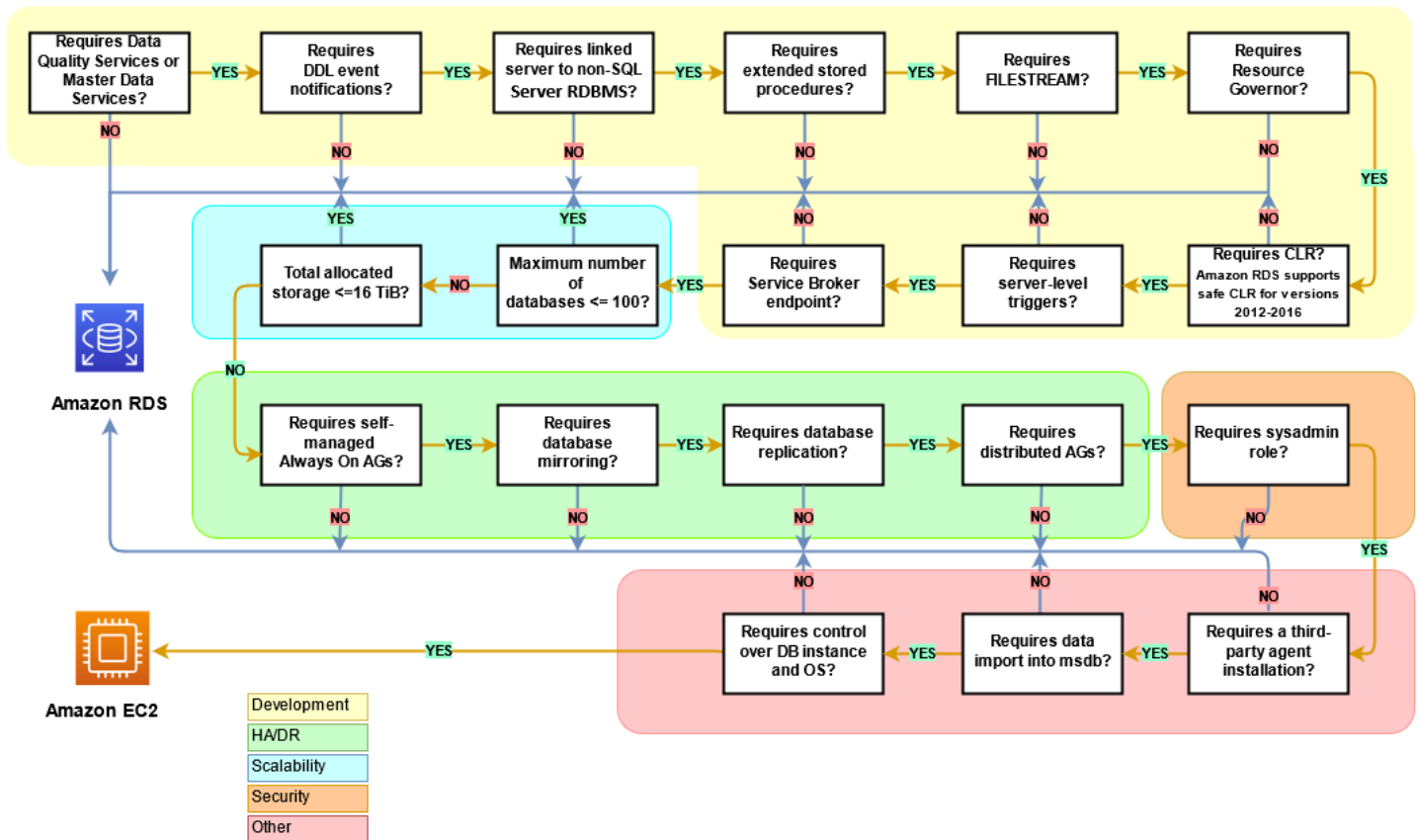
Fitur pengembangan	Amazon RDS	Amazon RDS Custom	Amazon EC2	Catatan
Impor data ke dalam database msdb		T 	Y 	Y Jika fitur ini sangat penting untuk beban kerja Anda, pertimbangkan untuk memilih Amazon RDS Custom atau Amazon EC2.
Metode instalasi	N/A	N/A	Amazon Machine Image (AMI) atau instalasi manual	
Edisi SQL Server	Perusahaan, Standar, Web, Ekspres	Perusahaan, Standar, Pengembang	Perusahaan, Standar, Web, Pengembang, Ekspres	
Versi SQL Server	2014, 2016, 2017, 2019, 2022	2019, 2022	2014, 2016, 2017, 2019, 2022	

Untuk informasi rinci tentang fitur-fitur ini, lihat berikut ini:

- [Produk Microsoft di AWS](#)
- [Arsitektur Referensi Direktori Aktif: Menerapkan Layanan Domain Direktori Aktif pada AWS](#)
- [Remote Desktop Gateway AWS aktif \(Mulai AWS Cepat\)](#)
- [Mengamankan Platform Microsoft AWS](#)
- [SQL Server dengan replikasi Selalu Aktif AWS \(Mulai AWS Cepat\)](#)
- [AWS Directory Service](#)

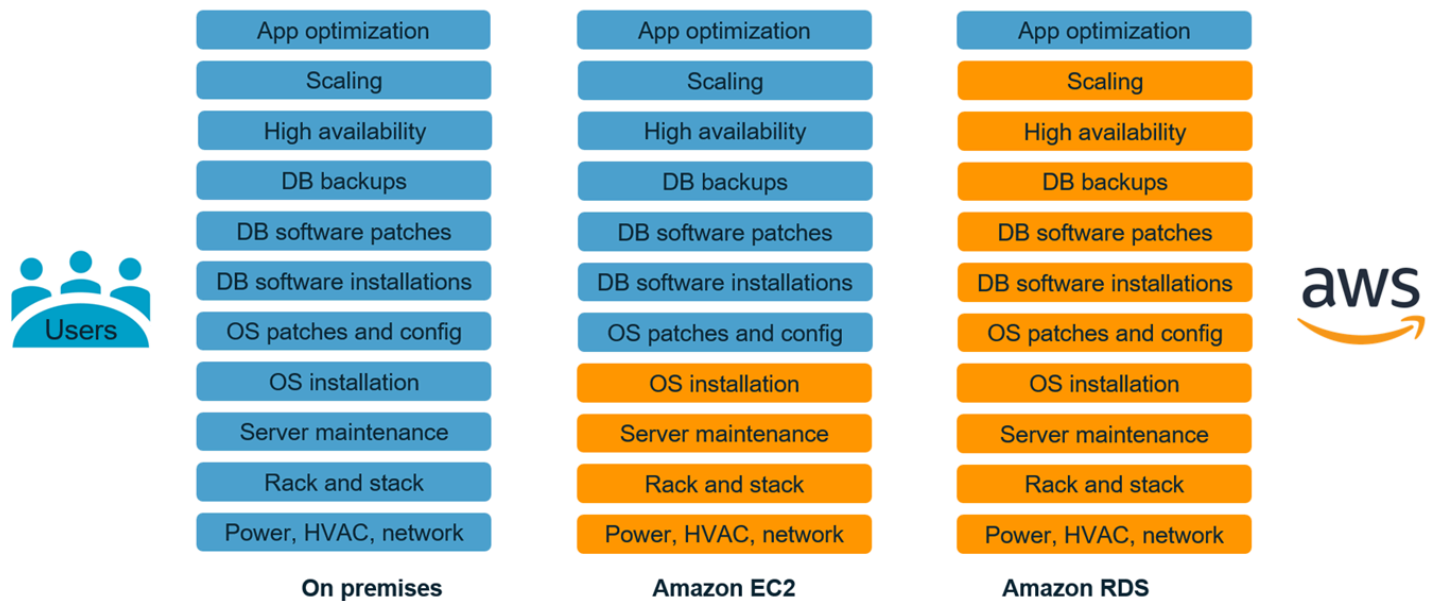
- [AWSEC2-SQLServerDBRestore](#)(Runbook Automation AWS Systems Manager yang mengembalikan backup database SQL Server yang disimpan di Amazon S3 ke SQL Server 2017 yang berjalan pada instans EC2 Linux)

Diagram berikut membantu memvisualisasikan informasi dalam tabel sebelumnya, untuk membantu dalam proses pengambilan keputusan Anda.



Tanggung Jawab Bersama

Diagram berikut menunjukkan pembagian tanggung jawab antara AWS dan pengguna dalam pengelolaan fitur SQL Server dan operasi.



Dengan AWS layanan, Anda tidak perlu khawatir tentang tugas-tugas administrasi seperti penyediaan server, patching, setup, konfigurasi, backup, atau pemulihan. AWS terus memantau cluster Anda untuk menjaga beban kerja Anda tetap aktif dan berjalan dengan penyimpanan penyembuhan diri dan penskalaan otomatis. Anda fokus pada tugas pengembangan aplikasi bernilai tinggi seperti desain skema, konstruksi kueri, dan pengoptimalan, sambil AWS menangani tugas operasional atas nama Anda.

Anda tidak perlu menyediakan infrastruktur yang berlebihan atau kurang penyediaan untuk mengakomodasi pertumbuhan aplikasi, lonjakan intermiten, dan persyaratan kinerja, atau menimbulkan biaya modal tetap, termasuk lisensi dan dukungan perangkat lunak, penyegaran perangkat keras, dan sumber daya untuk memelihara perangkat keras. AWS mengelola ini, sehingga Anda dapat menghabiskan waktu berinovasi dan membangun aplikasi baru, bukan mengelola infrastruktur.

Untuk informasi selengkapnya, lihat [Model Tanggung Jawab Bersama](#) di AWS situs web.

Pola migrasi database SQL Server

Gunakan link berikut untuk melihat pola panduan preskriptif untuk memigrasi database SQL Server ke AWS:

- [Rehost pola \(dari SQL Server ke Amazon EC2\)](#)
- [Pola replatform \(dari SQL Server ke Amazon RDS for SQL Server\)](#)
- [Pola re-arsitek \(dari SQL Server ke open-source dan AWS Database cloud-native\)](#)

Jika Anda mencari pola yang mencakup penggunaan alat tertentu, ketik nama alat di kotak pencarian atau pilih dari filter. Misalnya, Anda dapat menggunakan [kueri ini](#) untuk melihat semua pola migrasi SQL Server yang menggunakan AWS DMS.

Mitra

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

Berdasarkan persyaratan dan preferensi Anda, Anda dapat menggunakan mitra untuk menangani migrasi lengkap atau untuk membantu hanya beberapa aspek migrasi. Selain itu, Anda dapat menggunakan alat dan solusi yang disediakan oleh AWS Mitra Jaringan Mitra (APN) untuk membantu migrasi. Untuk katalog lengkap alat dan solusi migrasi, lihat [Alat dan solusi AWS mitra](#).

Sumber daya tambahan

Unggahan blog

- [Pemulihan bencana lintas Wilayah Amazon RDS for SQL Server](#)
- [Migrasi Database — Apa yang Perlu Anda Ketahui Sebelum Memulai?](#)
- [Menerapkan grup ketersediaan Selalu Aktifkan antara instans Amazon EC2 Windows dan Amazon Linux 2](#)
- [Cara merancang solusi Microsoft SQL Server hibrida menggunakan grup ketersediaan terdistribusi](#)
- [Cara bermigrasikan ke Amazon RDS for SQL Server menggunakan replikasi transaksional](#)
- [Memperkenalkan Replikasi yang sedang berlangsung dari Amazon RDS for SQL Server ke AWS Database Migration Service](#)
- [Pelajari alasannya AWS adalah cloud terbaik untuk menjalankan beban kerja Microsoft Windows Server dan SQL Server](#)
- [Migrasikan SQL Server on premise Anda ke Amazon EC2 Linux](#)
- [Migrasi Database SQL Server ke mesin Database MySQL kompatibel](#)
- [Migrasikan beban kerja SQL Server on premise Anda ke Amazon EC2 Linux](#)
- [Menyederhanakan deployment ketersediaan tinggi Microsoft SQL Server Anda menggunakan FSx for Windows File Server](#)
- [Simpan cadangan SQL Server di Amazon S3 menggunakan Storage Gateway](#)

AWS dokumentasi

- [Amazon Aurora](#)
- [Amazon EC2](#)
- [Amazon RDS](#)
- [Amazon RDS Kustom](#)
- [AWS DMS](#)
- [AWS SCT](#)
- [Lisensi SQL Server](#)

Ucapan Terima Kasih

Penulis mengakui spesialis berikut untuk berkontribusi pada panduan ini:

- Marcelo Fernandes, Konsultan Senior AWS Migrasi - [Memilih antara](#) bagian Amazon EC2 dan Amazon RDS
- Tarun Chawla, Konsultan Migrasi Database - [Memilih antara bagian Amazon EC2 dan Amazon RDS](#)
- Alex Zuo, Manajer Produk Teknis Senior, SQL Server di Amazon EC2 - bagian [Orkestrator AWS Migration Hub](#)

Lampiran: Kuesioner migrasi basis data SQL Server

Gunakan kuesioner di bagian ini sebagai titik awal untuk mengumpulkan informasi untuk tahap penilaian dan perencanaan proyek migrasi Anda. Anda dapat mengunduh kuesioner ini dalam format Microsoft Excel dan menggunakannya untuk merekam informasi Anda.

[Mengu](#)[kuesioner](#)

Informasi umum

1. Apa nama instans SQL Server?
2. Apa versi SQL Server Anda?
3. Apa edisi database SQL Server Anda: Standar, Pengembang, atau Enterprise?
4. Apa jenis database (OLTP, DW, pelaporan, batch processing)?
5. Berapa banyak database yang Anda miliki pada contoh SQL Server?
6. Berapa ukuran database Anda?
7. Apa pemeriksaan database?
8. Apa zona waktu database?
9. Berapa rata-rata dan maksimum transaksi I/O per detik (TPS)?
10. Apa IOPS (rata-rata dan maksimum) untuk database ini untuk operasi baca/tulis?
11. Berapa banyak log transaksi yang Anda hasilkan per jam (dengan ukuran rata-rata dan maksimum)?
12. Apakah database memiliki server terkait menunjuk ke database lain?
13. Apa persyaratan SLA untuk database Anda?
14. Apa persyaratan RTO dan RPO untuk database Anda?
15. Berapa banyak downtime database yang dapat Anda izinkan untuk tujuan migrasi?
16. Apakah Anda memiliki persyaratan kepatuhan, peraturan, atau audit?
17. Alat apa yang Anda gunakan untuk memantau database SQL Server Anda?

infrastruktur

1. Apa hostname 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 database pada mesin virtualisasi atau server fisik?
6. Apakah Anda menggunakan penyimpanan lokal?
7. Apakah Anda menggunakan jenis penyimpanan terlampir jaringan (NAS) atau jaringan area penyimpanan (SAN)?
8. Apakah Anda memiliki klaster atau contoh tunggal?

Backup database

1. Bagaimana Anda membuat cadangan database Anda? Seberapa sering?
2. Berapa periode retensi Anda untuk log transaksi dan backup?
3. Di mana Anda menyimpan cadangan Anda?

Fitur database

1. Apakah Anda menggunakan penyetelan otomatis untuk contoh SQL Server Anda?
2. Apakah Anda menggunakan operasi yang diindeks paralel?
3. Apakah Anda menggunakan fitur paralelisme tabel dipartisi?
4. Apakah Anda menggunakan tabel dan indeks partisi?

Keamanan database

1. Apakah Anda menggunakan masking data dinamis?
2. Apakah Anda menggunakan fitur keamanan seperti Transparan Database Encryption (TDE)?
3. Apakah Anda menggunakan audit server atau database?
4. Apakah Anda menggunakan kompresi lanjutan?

Ketersediaan basis data dan pemulihan bencana

1. Apa kebutuhan ketersediaan tinggi Anda?
2. Apakah Anda menggunakan replikasi transaksional?
3. Apakah Anda menggunakan replikasi transaksional peer-to-peer?
4. Apa jenis solusi ketersediaan tinggi (misalnya, failover clustering, Always On availability group, database mirroring) yang Anda gunakan untuk lingkungan SQL Server Anda?
5. Di mana wilayah database utama dan siaga Anda?
6. Apa yang Anda gunakan sebagai solusi pemulihan bencana (misalnya, pengiriman log, grup ketersediaan Always On, lingkungan virtualisasi berbasis SAN)?
7. Apakah Anda menggunakan alias Sistem Nama Domain (DNS)?

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
Bagian yang diperbarui	Menambahkan informasi tentang Amazon EBS Multi-Lampirkan dengan reservasi persisten dan opsi lain untuk menerapkan FCI AWS ke bagian Instans Cluster Gagal Selalu Berada.	April 1, 2024
Informasi yang diperbarui	Memperbarui matriks keputusan untuk SQL Server 2022.	Maret 18, 2024
Informasi yang diperbarui	Memperbarui matriks keputusan untuk mencerminkan bahwa Amazon RDS Custom sekarang mendukung TDE.	16 November 2023
Bagian yang ditambahkan	Menambahkan informasi tentang Orkestrator AWS Migration Hub .	29 Juni 2023
Bagian yang dihapus	Menghapus informasi tentang CloudEndure Migrasi, yang sedang dihentikan. AWS Application Migration Service adalah layanan migrasi utama yang direkomendasikan untuk lift-and-shift migrasi ke AWS Cloud	September 23, 2022

Bagian yang diperbarui	Menambahkan lebih banyak informasi ke matriks keputusan .	3 Agustus 2022
Bagian yang ditambahkan dan diperbarui	Menambahkan informasi tentang migrasi database SQL Server ke Amazon RDS Custom dan menggunakan Babelfish untuk memigrasi database SQL Server ke Aurora PostgreSQL. L. Memperbarui matriks keputusan dengan informasi terbaru.	Juli 29, 2022
Teks yang dikoreksi	Mengoreksi informasi CLR dalam matriks keputusan .	21 Juni 2022
Bagian yang diperbarui	Memperbarui bagian CloudEndure Migrasi dengan informasi terbaru tentang ketersediaan produk.	10 Mei 2022
Bagian yang dihapus	Menghapus informasi tentang Amazon RDS untuk VMware, yang tidak lagi didukung.	19 April 2022
Bagian yang ditambahkan	Menambahkan informasi tentang AWS Launch Wizard untuk SQL Server .	15 Juli 2021
Ditambahkan matriks keputusan	Di bagian Memilih antara Amazon EC2 dan Amazon RDS , disediakan side-by-side perbandingan dukungan SQL Server.	28 Juni 2021

[Informasi pengiriman log yang diperbarui](#)

Di bagian [pengiriman Log](#), diklarifikasi bahwa pengiriman log di Amazon RDS for SQL Server memerlukan skrip khusus.

25 Maret 2021

[Publikasi awal](#)

—

9 Oktober 2020

AWSGlosarium 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 Cloud. AWS
- 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 Cloud. AWS
- Relokasi (hypervisor-level lift and shift) — Pindahkan infrastruktur ke cloud tanpa membeli perangkat keras baru, menulis ulang aplikasi, atau memodifikasi operasi yang ada. Skenario migrasi ini khusus untuk VMware Cloud onAWS, yang mendukung kompatibilitas mesin virtual (VM) dan portabilitas beban kerja antara lingkungan lokal Anda dan. AWS Anda dapat menggunakan teknologi VMware Cloud Foundation dari pusat data lokal saat memigrasikan infrastruktur ke VMware Cloud. AWS Contoh: Pindahkan hypervisor yang menghosting database Oracle Anda ke VMware Cloud on. 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

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.

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-ArchitectedAWS.

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](#).

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.

Cloud Center of Excellence (CCoE)

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

komputasi cloud

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

model operasi cloud

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

tahap adopsi cloud

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

- Proyek — Menjalankan beberapa proyek terkait cloud untuk bukti konsep dan tujuan pembelajaran
- Foundation — Melakukan investasi dasar untuk meningkatkan adopsi cloud Anda (misalnya, membuat landing zone, mendefinisikan CCoE, membuat model operasi)

- Migrasi — Migrasi aplikasi individual
- Re-invention — Mengoptimalkan produk dan layanan, dan berinovasi di cloud

Tahapan ini didefinisikan oleh Stephen Orban dalam posting blog The [Journey Toward Cloud-First & the Stages of Adoption](#) di blog AWS Cloud Enterprise Strategy. 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 AWS CodeCommit. 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 atau kelas penyimpanan yang berkinerja lebih rendah dan lebih murah dapat mengurangi biaya.

visi komputer

Bidang AI yang digunakan oleh mesin untuk mengidentifikasi orang, tempat, dan benda dalam gambar dengan akurasi pada atau di atas tingkat manusia. Sering dibangun dengan model pembelajaran mendalam, ini mengotomatiskan ekstraksi, analisis, klasifikasi, dan pemahaman informasi yang berguna dari satu gambar atau urutan gambar.

database manajemen konfigurasi (CMDB)

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

paket kesesuaian

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

integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD)

Proses mengotomatiskan sumber, membangun, menguji, pementasan, dan tahap produksi dari proses rilis perangkat lunak. CI/CD umumnya 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](#).

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.

minimalisasi data

Prinsip pengumpulan dan pemrosesan hanya data yang sangat diperlukan. Mempraktikkan minimalisasi data 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)

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

kembar digital

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

tabel dimensi

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

musibah

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

pemulihan bencana (DR)

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

DML~

Lihat [bahasa manipulasi database](#).

desain berbasis domain

Pendekatan untuk mengembangkan sistem perangkat lunak yang kompleks dengan menghubungkan komponennya ke domain yang berkembang, atau tujuan bisnis inti, yang dilayani 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

pengecik, 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](#).

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.

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).

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 pengguna akhir dapat mengakses. Dalam pipa CI/CD, 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](#).

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.

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.

G

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 disukai.

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, 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.

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.

|

IloT

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

migrasi inkremental

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

infrastruktur

Semua sumber daya dan aset yang terkandung dalam lingkungan aplikasi.

infrastruktur sebagai kode (IAC)

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

Internet of Things industri (IIoT)

Penggunaan sensor dan perangkat yang terhubung ke internet di sektor industri, seperti manufaktur, energi, otomotif, perawatan kesehatan, ilmu kehidupan, dan pertanian. Untuk

informasi selengkapnya, lihat [Membangun strategi transformasi digital Internet of Things \(IIoT\) industri](#).

inspeksi VPC

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

Internet of Things (IoT)

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

interpretabilitas

Karakteristik model pembelajaran mesin yang menggambarkan sejauh mana manusia dapat memahami bagaimana prediksi model bergantung pada inputnya. Untuk informasi selengkapnya, 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](#).

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

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

lingkungan yang lebih rendah

Lihat [lingkungan](#).

M

pembelajaran mesin (ML)

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

cabang utama

Lihat [cabang](#).

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.

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.

layanan mikro

Layanan kecil dan independen yang berkomunikasi melalui API yang terdefinisi dengan baik dan biasanya dimiliki oleh tim kecil yang mandiri. Misalnya, sistem asuransi mungkin mencakup layanan mikro yang memetakan kemampuan bisnis, seperti penjualan atau pemasaran, atau subdomain, seperti pembelian, klaim, atau analitik. Manfaat layanan mikro termasuk kelincahan, penskalaan yang fleksibel, penyebaran yang mudah, kode yang dapat digunakan kembali,

dan ketahanan. Untuk informasi selengkapnya, lihat [Mengintegrasikan layanan mikro dengan menggunakan layanan tanpa AWS server](#).

arsitektur microservices

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

Program Percepatan Migrasi (MAP)

AWS Program yang menyediakan dukungan konsultasi, pelatihan, dan layanan untuk membantu organisasi membangun fondasi operasional yang kuat untuk pindah ke cloud, dan untuk membantu mengimbangi biaya awal migrasi. MAP mencakup metodologi migrasi untuk mengeksekusi migrasi lama dengan cara metodis dan seperangkat alat untuk mengotomatisasi dan mempercepat skenario migrasi umum.

migrasi dalam skala

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

pabrik migrasi

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 Cloud. AWS 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 Cloud. AWS 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](#).

MPA

Lihat [Penilaian Portofolio Migrasi](#).

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 Cloud. AWS](#)

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 selengkapnya, lihat [Mengevaluasi kesiapan modernisasi untuk aplikasi](#) di Cloud. AWS

aplikasi monolitik (monolit)

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

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.

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-ArchitectedAWS.

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 diAWS 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](#).

keluar (jalan keluar) VPC

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

P

batas izin

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

Informasi Identifikasi Pribadi (PII)

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

PII

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

buku pedoman

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

kebijakan

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

ketekunan poliglott

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. Untuk informasi selengkapnya, lihat [Mengaktifkan persistensi data di layanan mikro](#).

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 optimasi 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

Sebuah entitas dalam 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 oleh Desain

Pendekatan dalam rekayasa sistem yang memperhitungkan privasi di seluruh proses rekayasa. 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 beberapa VPC. Untuk informasi selengkapnya, lihat [Bekerja dengan zona yang dihosting pribadi](#) di dokumentasi Route 53.

kontrol proaktif

[Kontrol keamanan](#) yang dirancang untuk mencegah penyebaran sumber daya yang tidak sesuai. Kontrol ini memindai sumber daya sebelum disediakan. Jika sumber daya tidak sesuai dengan kontrol, maka itu tidak disediakan. Untuk informasi selengkapnya, lihat [panduan referensi Kontrol](#) dalam AWS Control Tower dokumentasi dan lihat [Kontrol proaktif](#) dalam Menerapkan kontrol keamanan pada AWS.

lingkungan produksi

Lihat [lingkungan](#).

pseudonimisasi

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

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 \(RACI\)](#).

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 hilangnya 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](#).

Wilayah

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 [Mengelola Wilayah AWS](#) di Referensi Umum AWS.

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](#).

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](#).

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 AWS Management Console 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.

SCP

Lihat [kebijakan kontrol layanan](#).

Rahasia

Dalam AWS Secrets Manager, informasi rahasia atau terbatas, seperti kata sandi atau kredensi 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 [Rahasia](#) dalam dokumentasi Secrets Manager.

kontrol keamanan

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

pengerasan keamanan

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

sistem informasi keamanan dan manajemen acara (SIEM)

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

otomatisasi respons keamanan

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

enkripsi sisi server

Enkripsi data di tujuannya, oleh Layanan AWS yang menerimanya.

kebijakan kontrol layanan (SCP)

Kebijakan yang menyediakan kontrol terpusat atas izin untuk semua akun di organisasi. AWS Organizations SCP menentukan pagar pembatas atau menetapkan batasan pada tindakan yang dapat didelegasikan oleh administrator kepada pengguna atau peran. Anda dapat menggunakan SCP sebagai daftar izin atau daftar penolakan, untuk menentukan layanan atau tindakan mana yang diizinkan atau dilarang. Untuk informasi selengkapnya, lihat [Kebijakan kontrol layanan](#) dalam AWS Organizations dokumentasi.

titik akhir layanan

URL titik masuk untuk fileLayanan AWS. Anda dapat menggunakan endpoint untuk terhubung secara terprogram ke layanan target. Untuk informasi selengkapnya, lihat [Layanan AWStitik 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.

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.

T

tag

Pasangan nilai kunci yang bertindak sebagai metadata untuk mengatur sumber daya Anda. AWS Tanda dapat 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 VPC dan jaringan lokal Anda. Untuk informasi selengkapnya, lihat [Apa itu gateway transit](#) dalam AWS Transit Gateway dokumentasi.

alur kerja berbasis batang

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

akses tepercaya

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

penyetelan

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

tim dua pizza

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

U

waswas

Sebuah konsep yang mengacu pada informasi yang tidak tepat, tidak lengkap, atau tidak diketahui yang dapat merusak keandalan model ML prediktif. Ada dua jenis ketidakpastian: ketidakpastian epistemik disebabkan oleh data yang terbatas dan tidak lengkap, sedangkan ketidakpastian aleatorik disebabkan oleh kebisingan dan keacakan yang melekat dalam data. 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 VPC yang memungkinkan Anda merutekan lalu lintas dengan menggunakan alamat IP pribadi. Untuk informasi selengkapnya, lihat [Apa itu peering VPC](#) di dokumentasi VPC Amazon.

kerentanan

Kelemahan perangkat lunak atau perangkat keras yang membahayakan keamanan sistem.

W

cache hangat

Cache buffer yang berisi data saat ini 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 [Kerangka Kualifikasi Beban Kerja AWS](#).

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.

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.