



Cetak biru untuk migrasi yang sukses dari Oracle Exadata ke AWS

AWS Bimbingan Preskriptif



AWS Bimbingan Preskriptif: Cetak biru untuk migrasi yang sukses dari Oracle Exadata ke AWS

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

Pengantar	1
Tren basis data utama	4
Tren basis data di pasar perusahaan	4
Basis data yang dibangun dengan tujuan versus konvergen	5
Strategi migrasi basis data	8
Dependensi migrasi database sebelum migrasi	8
Jalur migrasi basis data	9
Pertimbangan migrasi	12
Migrasi online	12
Migrasi offline	12
Pertimbangan tambahan	13
Fase penemuan	14
Karakteristik beban kerja	15
Rasio baca/tulis	17
Beban kerja non-relasional	18
Dependensi mesin database	18
Edisi dan versi database	19
Konsolidasi database	20
Penggunaan fitur Exadata	21
Pemindaian Cerdas	22
Indeks penyimpanan	25
Cache Flash Cerdas	26
Kompresi Kolumnar Hibrida	31
Manajemen Sumber Daya I/O	33
Memori Persisten (PMEM)	34
Ringkasan fitur dan alternatif Exadata AWS	35
Alat untuk fase penemuan	37
AWR	38
CellCLI	40
Kontrol Awan OEM	44
Tampilan basis data	44
AWS SCT	47
Persyaratan sumber daya untuk platform target	47
Persyaratan CPU dan memori	47

Persyaratan I/O	49
Pengujian kinerja pada platform target	51
Persyaratan aplikasi SLA	51
Kebijakan manajemen dan retensi siklus hidup data	53
Faktor lainnya	53
Diagram alur keputusan	54
Melakukan migrasi	56
Exadata ke AWS alat migrasi	57
AWS DMS migrasi	58
Migrasi Oracle GoldenGate	60
Migrasi Pompa Data Oracle	62
Migrasi Oracle RMAN	64
Migrasi Oracle Data Guard	67
AWS pola migrasi sampel	68
Pertimbangan fitur khusus Exadata	70
Pertimbangan migrasi database homogen	72
Enkripsi	72
Pembuatan Partisi Data	73
Kompresi data	73
Strategi ILM	74
Integrasi OEM	75
CloudWatch Integrasi Amazon	75
Statistik pengoptimal basis data	76
Pengaturan AWR	76
Pertimbangan Oracle RAC	77
Praktik terbaik tambahan untuk migrasi homogen	78
Rekomendasi replatforming	80
Pertimbangan jenis volume Amazon EBS	80
Praktik terbaik Amazon RDS for Oracle	81
Rekomendasi rehosting	83
Pertimbangan jenis instans Amazon EC2	83
Pertimbangan jenis volume Amazon EBS	83
Pertimbangan Oracle ASM	84
Praktik terbaik Oracle di Amazon EC2	85
Rekomendasi refactoring	88
Kegiatan pasca-migrasi	89

Pemantauan berkelanjutan	89
Rencana pemantauan	89
Acuan dasar performa	90
Pedoman kinerja utama	90
Alat-alat pemantauan	91
Amazon CloudWatch	91
Pemantauan yang Ditingkatkan	93
Wawasan Performa	94
Oracle Enterprise Manager	96
Optimalisasi biaya berkelanjutan	97
Ukuran instans Anda yang tepat	97
Pertimbangkan untuk pindah ke Oracle Database SE2	98
Gunakan instans DB yang dicadangkan	99
Gunakan prosesor AWS Graviton	99
Optimalkan kueri SQL Anda	99
Pemantauan otomatis	100
CloudWatch Alarm Amazon dan deteksi anomali	100
Amazon DevOps Guru untuk Amazon RDS	100
Audit otomatis	101
Audit Amazon RDS dasar	101
Aliran aktivitas basis data	102
Ringkasan	103
Sumber daya	104
Alat dan layanan	104
Program	105
Studi kasus	105
AWSKonten Panduan Preskriptif	106
Kontributor	107
Riwayat dokumen	108
Glosarium	109
#	109
A	110
B	113
C	115
D	118
E	122

F	124
G	126
H	127
I	128
L	131
M	132
O	136
P	139
Q	142
R	142
D	145
T	149
U	151
V	151
W	152
Z	153
.....	cliv

Cetak biru untuk migrasi yang sukses dari Oracle Exadata ke AWS

Amazon Web Services ([kontributor](#))

Juli 2024 ([sejarah dokumen](#))

Database sedang mengalami transformasi besar sebagai akibat dari ledakan data dan pergeseran ke layanan cloud. Pasar sistem manajemen basis data (DBMS) telah menambahkan 40 miliar USD ke pendapatan 2017 sebesar 38,6 miliar USD — dua kali lipat dalam lima tahun — dan kisah pasar DBMS terbesar terus menjadi dampak pergeseran pendapatan ke cloud. Menurut Gartner Research, “Pasar DBMS tumbuh sebesar 14.4% pada tahun 2022, mencapai \$91 miliar. Cloud DBPaaS menangkap hampir semua keuntungan, dengan belanja cloud (55,2%) melebihi lokal (44,8%) . Perusahaan dapat menggunakan layanan cloud untuk membebaskan tim TI mereka dari tugas-tugas database yang memakan waktu seperti penyediaan server, patching, dan backup. Sebagai contoh, [layanan database yang dikelola AWS sepenuhnya](#) menyediakan pemantauan berkelanjutan, penyimpanan penyembuhan diri, dan penskalaan otomatis untuk membantu perusahaan fokus pada pengembangan aplikasi.

Ketika perusahaan berusaha memaksimalkan manfaat pindah ke cloud sebagai bagian dari transformasi digital mereka, mereka fokus pada memodernisasi infrastruktur data mereka. Untuk memenuhi tujuan modernisasi data, perusahaan berupaya mencapai kemampuan berikut:

- Pengurangan biaya kepemilikan total (TCO) — Perlambatan di pasar global, meningkatnya inflasi, ketakutan akan resesi global, dan kondisi pasar lainnya memaksa perusahaan untuk memprioritaskan efisiensi biaya.
- Kecepatan dan kelincuhan — Dalam lingkungan komputasi awan, sumber daya TI baru mudah digunakan, yang berarti bahwa perusahaan mengurangi waktu untuk membuat sumber daya tersebut tersedia bagi pengembang dari minggu menjadi hanya beberapa menit. Hal ini menghasilkan peningkatan dramatis dalam kelincuhan bagi organisasi, karena biaya dan waktu untuk eksperimen dan pengembangan secara signifikan lebih rendah.
- Skala global, keamanan, dan ketersediaan tinggi — Perusahaan melayani pelanggan di seluruh dunia, dan oleh karena itu mereka sering mencari cara yang lebih baik untuk mendukung pelanggan mereka di wilayah geografis yang berbeda dan memberikan pengawasan data penuh dengan berbagai tingkat keamanan, termasuk isolasi jaringan dan end-to-end enkripsi.

Ketersediaan, keandalan, dan keamanan yang tinggi adalah kunci untuk beban kerja perusahaan yang penting bagi bisnis.

- Kinerja dalam skala — Perusahaan mencari elastisitas: untuk memulai dari yang kecil dan skala basis data relasional atau non-relasional mereka saat aplikasi mereka tumbuh. Mereka ingin memenuhi kebutuhan penyimpanan dan komputasi mereka dengan lebih mudah, dan lebih disukai tanpa downtime.

Sebagai bagian dari pergeseran ke layanan cloud, perusahaan sering mencari untuk membebaskan diri dari arsitektur perangkat lunak monolitik dan menggunakan layanan mikro untuk mengurangi kompleksitas aplikasi dan meningkatkan inovasi dan kelincahan. Namun, beberapa perusahaan masih menggunakan database monolitik untuk melayani beberapa layanan mikro. Misalnya, layanan mikro yang memiliki persyaratan data yang berbeda, laju pertumbuhan, dan database (relasional atau non-relasional) mungkin dipaksa untuk menggunakan mesin database monolitik yang sama. Ini berarti bahwa pengembang sering diminta untuk menormalkan model data agar sesuai dengan model relasional daripada menggunakan model data yang mendukung persyaratan mereka. Oleh karena itu, menggunakan mesin database yang sama dapat berdampak negatif pada fleksibilitas dan kelincahan pengembang.

Contoh pendekatan monolitik adalah arsitektur yang menggunakan Oracle Database di Oracle Exadata dan yang melayani banyak beban kerja, beberapa aplikasi, dan berpotensi beberapa layanan mikro. Oracle Exadata adalah sistem rekayasa yang terdiri dari komponen perangkat keras dan perangkat lunak. Ini dirancang untuk secara eksklusif menjalankan beban kerja Oracle Database dengan kinerja tinggi.

Namun, menjalankan beban kerja Anda dengan mesin database tunggal dapat memperkenalkan tantangan kelincahan bisnis. Banyak perusahaan menyadari bahwa setiap beban kerja mungkin memerlukan mesin database yang berbeda untuk kebutuhannya. Selain itu, database monolitik mungkin memperkenalkan tantangan total biaya kepemilikan (TCO) bagi banyak perusahaan karena ketergantungan mereka pada Oracle untuk penyebaran dan pemeliharaan perangkat keras, dalam kasus database Oracle yang berjalan di Exadata lokal. Database monolitik juga menciptakan tantangan penguncian karena mereka menggunakan fitur eksklusif yang menghambat kemampuan mereka untuk memindahkan beban kerja dan aplikasi Oracle ke platform non-Exadata atau ke database lain.

Untuk alasan ini, beberapa perusahaan mempertimbangkan untuk bermigrasi dari Exadata ke database yang dikelola AWS sepenuhnya dan dibuat khusus. AWS menawarkan [banyak jenis database relasional dan dibangun khusus](#) untuk mendukung beragam model data, termasuk

relasional, nilai kunci, dokumen, dalam memori, grafik, deret waktu, dan database kolom lebar. AWS konsultan telah membantu pelanggan seperti [California Healthcare Eligibility, Enrollment, and Retention System \(CalHeers\)](#), [Australia Finance Group \(AFG\)](#), dan [EDF UK](#) untuk memigrasikan beban kerja Exadata mereka ke AWS.

Karena perusahaan mempertimbangkan migrasi beban kerja dari Oracle Exadata ke AWS, mereka perlu memiliki strategi migrasi yang efektif yang selaras dengan aplikasi dan kebutuhan bisnis mereka dan panduan yang jelas untuk memastikan migrasi yang lancar. Cetak biru untuk Oracle Exadata yang sukses untuk AWS migrasi adalah pendekatan multi-langkah dan sistematis yang mencakup penemuan pra-migrasi dan penilaian kinerja, migrasi data, dan rutinitas pasca-migrasi untuk kinerja dan biaya yang optimal.

Tujuan dari panduan ini adalah untuk berbagi wawasan, praktik terbaik, dan tips tentang cara merencanakan, melakukan, dan mempertahankan migrasi yang sukses dari Oracle Exadata ke AWS. Ini dimaksudkan untuk membantu khalayak teknis, termasuk DBAs, arsitek TI, DevOps insinyur, CTOs, dan lainnya dalam perjalanan migrasi mereka dari Oracle Exadata ke AWS.

Dalam panduan ini:

- [Tren basis data utama](#)
- [Strategi migrasi basis data](#)
- [Pertimbangan migrasi](#)
- [Fase penemuan](#)
- [Melakukan migrasi](#)
- [Kegiatan pasca-migrasi](#)
- [Ringkasan](#)
- [Sumber](#)

* [Pangsa Pasar: Sistem Manajemen Basis Data, Seluruh Dunia, 2022](#) (Gartner Research, 17 Mei 2023)

Tren basis data utama

Bagian ini membahas tren basis data utama pada saat publikasi. Informasi ini membantu memperjelas motivasi yang mendorong beban kerja database ke cloud. Bagian ini mencakup topik-topik berikut:

- [Tren basis data di pasar perusahaan](#)
- [Perbedaan antara database yang dibangun khusus dan database konvergen](#)

Tren basis data di pasar perusahaan

Pasar basis data saat ini sedang mengalami perubahan signifikan. Volume data tumbuh secara eksponensial. Jumlah total data yang diambil, disalin, dan dikonsumsi secara global per tahun meningkat. Pelanggan harus mendapatkan nilai lebih dari data mereka. Perusahaan cloud seperti AWS menawarkan berbagai teknologi database yang dibuat khusus untuk kebutuhan database. Layanan ini menawarkan kelincahan, inovasi, overhead pemeliharaan yang lebih sedikit, dan lebih banyak kontrol, dan lebih hemat biaya. Strategi data modern dapat mendukung kasus penggunaan saat ini dan masa depan, termasuk langkah-langkah untuk membangun solusi end-to-end data untuk menyimpan, mengakses, menganalisis, memvisualisasikan, dan memprediksi hasil masa depan. Untuk informasi selengkapnya tentang layanan dan solusi data AWS, lihat situs web [AWS untuk Data](#).

Database relasional komersial menjadi mainstream lebih dari 40 tahun yang lalu. Saat itu, kapasitas perangkat keras terbatas dan mahal. Biaya penyimpanan sangat tinggi, dan data dinormalisasi untuk menghindari penyimpanan duplikat. Sekarang, sebagian besar penyimpanan lebih murah daripada komputasi dan memori. Persyaratan juga telah berubah, dan Anda mungkin memerlukan kinerja mikrodetik pada kumpulan data berbeda yang menyertakan data terstruktur dan tidak terstruktur. Selama bertahun-tahun, pelanggan dibatasi untuk menggunakan satu set kecil platform database. Aplikasi Commercial off-the-shelf (COTS) seperti Oracle E-Business Suite, SiebelCRM, dan Peoplesoft hanya dapat berjalan di Oracle. Perusahaan mengembangkan aplikasi in-house dengan menggunakan fitur eksklusif seperti SQL PL/atau Pro* C, dan aplikasi khusus ini memenuhi permintaan bisnis. Namun, seiring waktu, fitur eksklusif menjadi kompleks dan lebih sulit untuk dipertahankan. Kendala anggaran TI memaksa banyak perusahaan untuk memikirkan kembali pendekatan mereka untuk memenuhi permintaan bisnis dan fokus pada mengoptimalkan struktur biaya mereka dengan bermigrasi ke opsi yang lebih murah, di mana biaya migrasi mereka ditentukan oleh tingkat penyesuaian yang diperlukan.

Sebagai alternatif untuk produk basis data komersial, AWS telah memperkenalkan portofolio database open source yang dikelola sepenuhnya, relasional, serta mesin database non-relasional yang dibangun khusus untuk optimalisasi beban kerja kasus penggunaan tertentu. Keuntungan utama dari database open source adalah biayanya yang lebih rendah. Anggaran TI tidak terbebani oleh pembayaran kontrak, karena mereka tidak lagi harus membayar biaya lisensi yang terkait dengan perangkat lunak komersial. Dengan penghematan ini, departemen TI memiliki fleksibilitas yang sangat besar, sehingga mereka dapat bereksperimen dan gesit. Misalnya, banyak pelanggan memodernisasi beban kerja Oracle mereka dengan pindah ke PostgreSQL. SQL SQLFungsionalitas PostgreSQL telah meningkat secara signifikan selama 10 tahun terakhir dan sekarang mencakup banyak fitur database perusahaan untuk mendukung beban kerja yang besar dan kritis.

Cara database beroperasi juga mengalami perubahan. Selama 30 tahun terakhir, pelanggan telah mengoperasikan pusat data mereka sendiri di tempat: mereka membeli dan mengelola infrastruktur, memelihara perangkat keras, jaringan berlisensi dan basis data komersial, dan mempekerjakan profesional TI untuk menjalankan pusat data. Administrator database (DBAs) dikonfigurasi dan dioperasikan terutama database relasional. Tugas operasional mereka termasuk instalasi perangkat keras dan perangkat lunak, memilah masalah lisensi, konfigurasi, penambalan, dan cadangan basis data. DBAs juga mengelola penyetelan kinerja, konfigurasi untuk ketersediaan tinggi, keamanan, dan masalah kepatuhan. Mengelola database termasuk tugas-tugas berulang yang membosankan dan memakan waktu dan mahal. Pelanggan menghabiskan waktu mengelola infrastruktur alih-alih berfokus pada kompetensi bisnis inti. Untuk alasan ini, perusahaan berinvestasi dalam otomatisasi DBA dan tugas operasional jika memungkinkan untuk memanfaatkan DBA sumber daya dengan lebih baik, sehingga mereka dapat menghabiskan lebih banyak waktu untuk inovasi. Untuk informasi selengkapnya, lihat IDC laporan [Amazon Relational Database Service Memberikan Kinerja Database yang Ditingkatkan dengan Biaya Total Lebih Rendah](#).

Basis data yang dibangun dengan tujuan versus konvergen

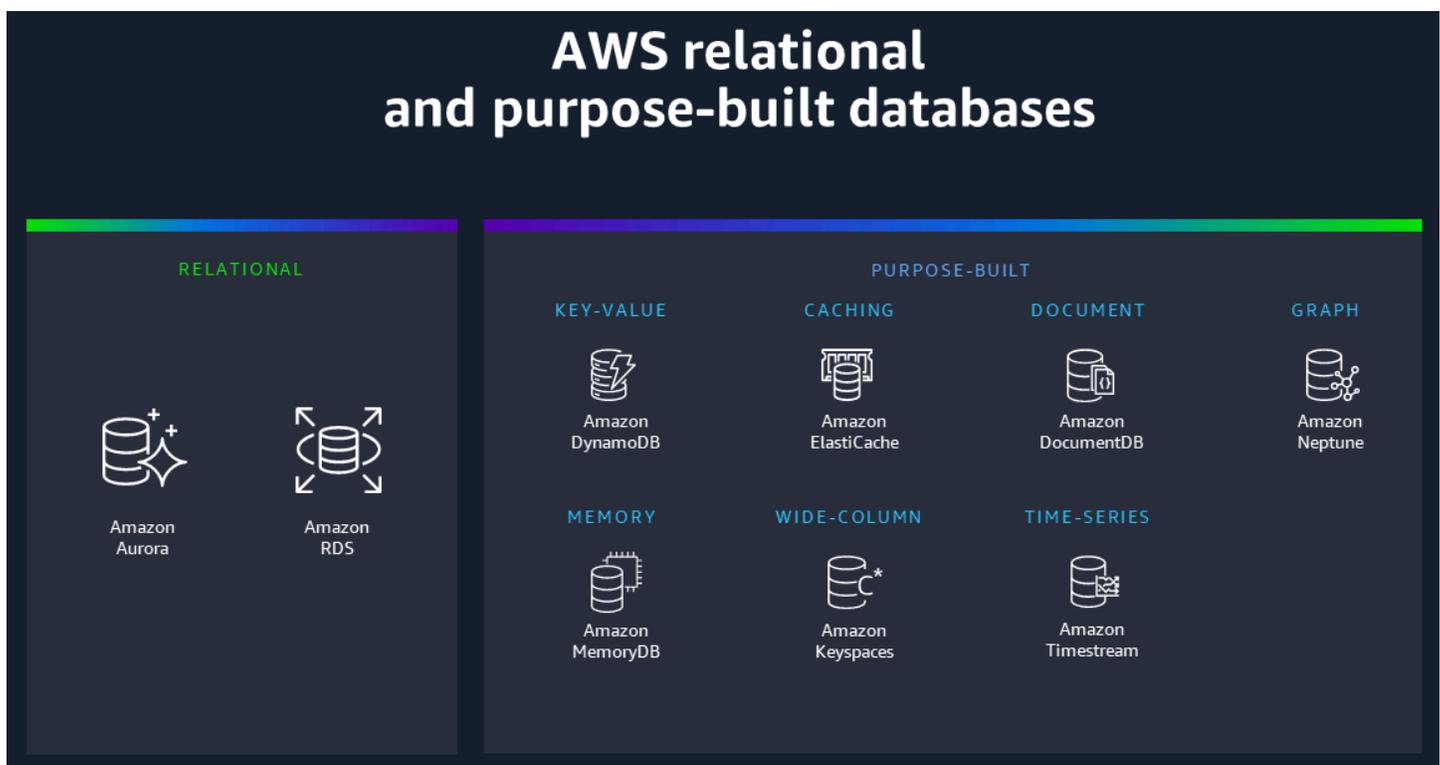
Oracle Exadata awalnya dirilis pada tahun 2008. Ini dirancang untuk mengatasi kemacetan umum dengan database besar: memindahkan volume besar data dari penyimpanan disk ke server database. Mengatasi masalah ini bisa sangat bermanfaat untuk aplikasi gudang data di mana pemindaian kumpulan data besar adalah hal biasa. Exadata meningkatkan pipa antara tingkat penyimpanan dan database dengan menggunakan InfiniBand, dan mengurangi jumlah data yang akan ditransfer dari disk ke tingkat database dengan menggunakan fitur perangkat lunak seperti Exadata Smart Scan. Dalam beberapa kasus, Exadata memperkenalkan peningkatan kinerja, tetapi ini datang dengan biaya peningkatan total biaya kepemilikan (TCO) dan mengurangi kelincahan, untuk alasan yang disebutkan di bagian sebelumnya.

Ada dua pendekatan untuk hosting aplikasi database:

- Menggunakan database khusus yang dibuat khusus untuk beban kerja atau kasus penggunaan tertentu
- Menggunakan database konvergen yang mendukung beban kerja database yang berbeda dalam database yang sama

Setelah pelanggan bermigrasi ke cloud, mereka sering ingin [memodernisasi arsitektur aplikasi mereka dengan menggunakan microservices, container, dan arsitektur](#) tanpa server. Aplikasi modern ini memiliki fungsionalitas, kinerja, dan tuntutan skalabilitas yang unik, yang memerlukan tipe database khusus untuk mendukung setiap beban kerja.

AWS menawarkan database relasional berkinerja tinggi dengan biaya yang jauh lebih rendah dibandingkan dengan database komersial kelas perusahaan, dan delapan basis data yang dibuat khusus. Setiap database yang dibuat khusus dirancang secara unik untuk memberikan kinerja optimal untuk kasus penggunaan tertentu, sehingga perusahaan tidak perlu berkompromi, seperti yang sering terjadi ketika menggunakan pendekatan database konvergen. Diagram berikut menggambarkan penawaran AWS database.



Jenis basis data	Kasus penggunaan	Layanan AWS
Relasional	Aplikasi tradisional, perencanaan sumber daya perusahaan, manajemen hubungan pelanggan	Amazon Aurora, AmazonRDS, Amazon Redshift
Nilai kunci	Aplikasi web lalu lintas tinggi, sistem e-niaga, aplikasi game	Amazon DynamoDB
Dalam memori	Caching, manajemen sesi, papan pemimpin game, aplikasi geospasial	Amazon ElastiCache, Amazon MemoryDB
Dokumen	Manajemen konten, katalog, profil pengguna	Amazon DocumentDB (dengan kompatibilitas MongoDB)
Kolom lebar	Aplikasi industri skala tinggi untuk pemeliharaan peralatan, manajemen armada, dan optimalisasi rute	Amazon Keyspaces (untuk Apache Cassandra)
Grafik	Deteksi penipuan, jejaring sosial, mesin rekomendasi	Amazon Neptune
Deret waktu	Aplikasi Internet of Things (IoT), DevOps, telemetri industri	Amazon Timestream

Strategi migrasi basis data

Bagian ini membahas strategi untuk memigrasikan beban kerja Exadata ke file. AWS Cloud Merencanakan strategi migrasi database yang komprehensif adalah kunci keberhasilan migrasi Exadata. Bagian ini mencakup topik-topik berikut:

- [Dependensi migrasi database sebelum migrasi](#)
- [Jalur migrasi basis data](#)

Dependensi migrasi database sebelum migrasi

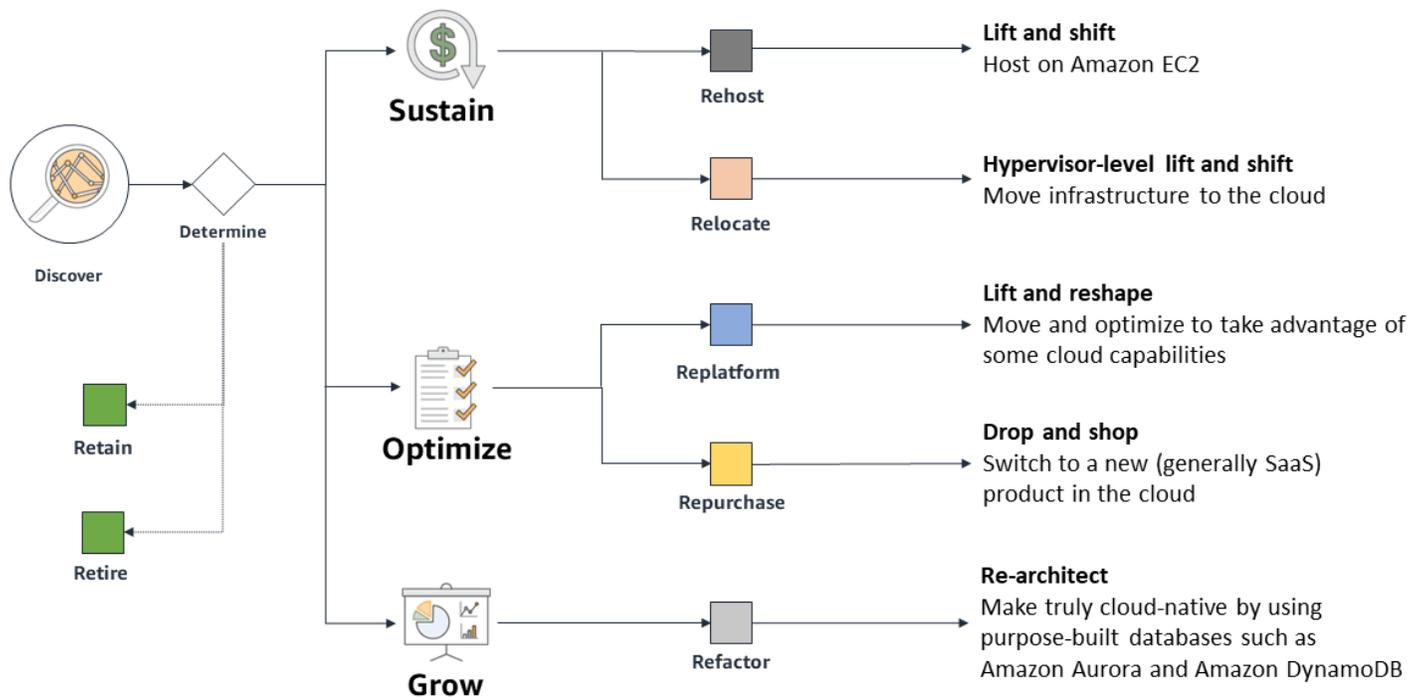
Merumuskan strategi migrasi membutuhkan pemahaman tentang dependensi kunci dan pengoperasian beban kerja di masa depan. AWS Sebelum Anda memilih pendekatan migrasi, kami sarankan Anda mengumpulkan dan menganalisis informasi berikut:

- Memahami sumber sistem Exadata.
 - Versi, edisi, dan ukuran alat perangkat keras Exadata
 - Opsi database dan versi, alat, dan utilitas yang tersedia
 - Ukuran dan jumlah database yang akan dimigrasikan
 - Posisi lisensi Oracle
- Memahami dependensi aplikasi dan database.
 - Aplikasi mana yang menggunakan database? Apakah database bagian dari aplikasi terintegrasi di mana beberapa database terhubung?
 - Apakah ada dependensi lokal untuk memindahkan database?
- Memahami persyaratan bisnis di sekitar jendela migrasi.
 - Berapa banyak waktu yang tersedia untuk migrasi?
 - Apa konektivitas jaringan antara server sumber dan AWS?
 - Apa prospek bisnis jangka panjang untuk database dan aplikasi?
 - Apakah migrasi dan peralihan akan AWS diselesaikan dalam satu langkah atau urutan langkah dari waktu ke waktu?
- Memahami tingkat modernisasi database yang mungkin, mengingat persyaratan aplikasi.
 - Apakah beban kerja harus tetap di Oracle?
 - Bisakah database sumber dimodernisasi? Jika demikian, ke level apa?

- Layanan AWS database mana yang dapat menampung beban kerja Oracle?
- Memahami persyaratan bisnis dan kinerja setelah beban kerja Exadata dimigrasikan ke. AWS

Jalur migrasi basis data

Jalur migrasi dan pilihan dikenal sebagai 7 Rs dan diilustrasikan dalam diagram berikut.



Jalur ini adalah:

- Rehost (angkat dan geser) — Pindahkan aplikasi ke cloud tanpa membuat perubahan apa pun. Misalnya, memigrasikan database Oracle lokal Anda ke Oracle di instans Amazon Elastic Compute Cloud (Amazon EC2) di instans Amazon Elastic Compute Cloud (Amazon EC2). AWS Cloud
- Relokasi (hypervisor-level lift and shift) — Pindahkan infrastruktur ke cloud tanpa membeli perangkat keras baru, menulis ulang aplikasi, atau memodifikasi operasi yang ada. Anda memigrasikan server dari platform lokal ke layanan cloud untuk platform yang sama. Misalnya, memigrasikan aplikasi Microsoft Hyper-V ke. AWS
- Replatform (angkat dan bentuk ulang) — Pindahkan aplikasi ke cloud dan perkenalkan beberapa tingkat pengoptimalan untuk memanfaatkan kemampuan cloud. Misalnya, memigrasikan database Oracle lokal ke Amazon RDS for Oracle di. AWS Cloud

- Pembelian kembali (drop and shop) — Ubah ke produk yang berbeda, biasanya dengan berpindah dari aplikasi tradisional ke produk perangkat lunak sebagai layanan (SaaS), dan memigrasikan data dari aplikasi lokal Anda ke produk baru. Misalnya, memigrasikan data pelanggan dari sistem manajemen hubungan pelanggan (CRM) lokal ke Salesforce.com.
- Refactor (re-architect) - Memindahkan aplikasi dan memodifikasi arsitekturnya dengan memanfaatkan sepenuhnya fitur cloud-native untuk meningkatkan kelincahan, kinerja, dan skalabilitas. Misalnya, bermigrasi menggunakan salah satu [strategi migrasi](#) Panduan AWS Preskriptif untuk database relasional. Strategi refactoring juga dapat mencakup penulisan ulang aplikasi untuk menggunakan database yang dibuat khusus yang menawarkan beban kerja yang berbeda. AWS Atau, pilih untuk memodernisasi aplikasi monolitik dengan memecahnya menjadi layanan mikro yang lebih kecil.
- Pertahankan (kunjungi kembali) - Simpan aplikasi di lingkungan sumber. Ini mungkin termasuk aplikasi yang memerlukan refactoring besar, di mana Anda mungkin ingin menunda pekerjaan sampai nanti. Atau Anda mungkin memiliki 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.

Biasanya, dengan tumpukan Exadata, rehost dan replatform adalah jalur migrasi utama. Pendekatan rehosting digunakan ketika beban kerja Exadata kompleks atau menggunakan aplikasi komersial off-the-shelf (COTS). Refactoring terlalu memakan waktu dan sumber daya intensif untuk diterapkan dalam satu langkah jika tujuannya adalah modernisasi database (misalnya, mengganti database Oracle Exadata dengan Amazon Aurora PostgreSQL Compatible Edition). Anda dapat mempertimbangkan pendekatan dua langkah sebagai gantinya: Pertama, rehost database Oracle di Amazon EC2 atau replatform database di Amazon RDS for Oracle. Anda kemudian dapat memfaktorkan ulang database ke Aurora PostgreSQL yang kompatibel. Pendekatan ini membantu mengurangi biaya, sumber daya, dan risiko selama fase pertama dan berfokus pada optimalisasi dan modernisasi di fase kedua.

Ada empat penawaran AWS database yang mendukung migrasi rehost atau replatform:

- Amazon Relational Database Service (Amazon RDS) dan Amazon Aurora adalah layanan yang dikelola sepenuhnya yang memudahkan pengaturan, pengoperasian, dan skala database di cloud. [Saat ini, mereka mendukung delapan mesin database: Amazon Aurora dengan kompatibilitas MySQL, Amazon Aurora dengan kompatibilitas PostgreSQL, dan Amazon RDS untuk Db2, MySQL, MariaDB, PostgreSQL, Oracle, dan SQL Server.](#)

- Amazon EC2 mendukung database Oracle yang dikelola sendiri. Ini memberikan kontrol penuh atas infrastruktur dan pengaturan lingkungan database. Menjalankan database Anda di Amazon EC2 sangat mirip dengan menjalankan database Anda di server khusus. Anda memiliki kontrol penuh atas basis data dan akses tingkat sistem operasi dengan pilihan alat untuk mengelola sistem operasi, perangkat lunak database, tambalan, replikasi data, pencadangan, dan pemulihan. Opsi migrasi ini memerlukan pengaturan, konfigurasi, pengelolaan, dan penyyetelan semua komponen seperti yang Anda lakukan di tempat. Ini mencakup konfigurasi instans EC2, volume penyimpanan, skalabilitas, jaringan, dan keamanan.
- Amazon RDS Custom for Oracle mendukung kustomisasi sistem operasi dan lingkungan database yang mendasarinya. Ini memberi Anda lebih banyak kontrol daripada Amazon RDS, tetapi juga lebih banyak tanggung jawab untuk tugas-tugas seperti penambalan sistem operasi. Anda juga perlu memastikan bahwa penyesuaian Anda tidak mengganggu AWS otomatisasi, yang merupakan bagian inti dari model tanggung jawab bersama kami dengan Amazon RDS Custom.

Pelanggan sering memigrasikan beban kerja mereka ke Amazon RDS atau Amazon EC2 (untuk database Oracle yang dikelola sendiri). Untuk [Amazon RDS](#), AWS mengelola sistem operasi dan memberikan izin terbatas pada lapisan database. Saat Anda membuat database Amazon RDS, AWS berikan endpoint database yang dapat digunakan untuk terhubung ke instance database. Amazon RDS Custom memberi Anda akses penuh ke database yang mendasarinya, sistem operasi, dan semua sumber daya. Beberapa aktivitas database dibagi antara Anda dan AWS otomatisasi. Jika Anda meng-rehost database Oracle Anda pada instans EC2, Anda mengelola database, sistem operasi, dan sumber daya seperti yang Anda lakukan ketika Anda menjalankan database Oracle Anda di tempat. Oleh karena itu, jika Anda memiliki beban kerja yang tidak dapat dipindahkan ke Amazon RDS, pertimbangkan untuk memigrasikan database Oracle Anda ke Amazon RDS Custom atau Amazon EC2. Untuk panduan tambahan, lihat [Memilih layanan AWS database](#) di Pusat Sumber Daya AWS Memulai. Bagian selanjutnya dari panduan ini membahas opsi ini secara lebih rinci.

Pertimbangan migrasi

Ada banyak alat dan teknik untuk memigrasikan beban kerja Exadata Anda. AWS Ini terbagi dalam dua kategori utama: migrasi fisik dan migrasi logis. Migrasi fisik mengacu pada mengangkat blok basis data demi blok dari satu server ke server lainnya. Migrasi logis melibatkan ekstraksi data dari satu database dan memuatnya ke database lain.

Anda juga dapat memilih metode migrasi online atau offline berdasarkan apakah beban kerja Anda dapat mentolerir waktu henti minimal (nol atau mendekati nol) atau lebih lama.

Migrasi online

Metode ini digunakan ketika aplikasi membutuhkan waktu henti mendekati nol hingga minimal. Biasanya, database besar dan kritis menggunakan metode ini. Dalam migrasi online, database sumber dimigrasikan dalam beberapa langkah ke AWS. Pada langkah awal, data dalam database sumber disalin ke database target saat database sumber masih berjalan. Pada langkah selanjutnya, semua perubahan dari database sumber disebarkan ke database target. Ketika database sumber dan target disinkronkan, mereka siap untuk dipotong. Selama cutover, aplikasi mengalihkan koneksinya ke database target aktif. AWS

Migrasi online dari database Oracle Anda ke Amazon RDS for Oracle biasanya melibatkan Oracle Data Pump untuk langkah awal (full load). Transaksi dalam pesawat kemudian ditangani dengan menggunakan alat replikasi logis seperti AWS Database Migration Service (AWS DMS) atau Oracle GoldenGate. Jika Anda menggunakan metode ini untuk bermigrasi ke Amazon EC2, Anda dapat menangani transaksi muatan penuh dan dalam penerbangan dengan menggunakan Oracle Data Guard atau Oracle Recovery Manager (RMAN). Anda juga dapat menggunakan alat logis seperti AWS DMS dan Oracle GoldenGate. [Bagian Melakukan migrasi](#) menjelaskan alat ini secara lebih rinci.

Migrasi offline

Anda dapat menggunakan metode migrasi offline jika aplikasi Anda mampu membayar downtime yang direncanakan. Biasanya, database kecil dan kurang kritis menggunakan metode ini. Dengan jenis migrasi ini, alat replikasi logis biasanya tidak diperlukan. Untuk migrasi offline ke Amazon RDS for Oracle, Anda dapat menggunakan Oracle Data Pump. Untuk migrasi offline ke Amazon EC2, Anda dapat menggunakan Oracle RMAN atau Data Pump. Bagian [Melakukan migrasi](#) membahas alat-alat ini secara lebih rinci.

Pertimbangan tambahan

Pertimbangan lain adalah apakah akan memindahkan semua data ke lingkungan baru atau mengarsipkan data sebelum migrasi berlangsung. Juga, konsolidasi skema mungkin diperlukan. Jika migrasi melibatkan beberapa terabyte, menggunakan perangkat fisik untuk menyalin data dan kemudian mengangkutnya lebih cepat daripada menyalin data di seluruh jaringan. Bagian selanjutnya dari panduan ini memperluas teknik-teknik ini.

Fase penemuan

Exadata adalah sistem rekayasa yang dioptimalkan untuk menjalankan berbagai jenis beban kerja database Oracle dan banyak digunakan sebagai platform konsolidasi untuk database Oracle. Beban kerja ini termasuk pemrosesan transaksi online (OLTP) dan beban kerja pemrosesan analitik online (OLAP), aplikasi intensif transaksi tinggi, kritis bisnis, dan beban kerja non-kritis yang tidak memerlukan kemampuan sistem rekayasa seperti Exadata. Salah satu fase kunci dalam migrasi beban kerja Exadata yang sukses adalah fase penemuan. Pada fase ini, Anda menganalisis platform Exadata sumber untuk menilai detail penting seperti bagaimana aplikasi dan unit bisnis menggunakan sistem Exadata mereka untuk memenuhi persyaratan kinerja dan ketersediaan serta manfaat dari fitur khusus Exadata. Informasi yang Anda kumpulkan selama fase penemuan sangat penting untuk memahami persyaratan beban kerja Anda dan memilih platform yang tepat AWS untuk memenuhi persyaratan kinerja dan ketersediaan aplikasi Anda.

Bagian ini membahas bagaimana Anda dapat menilai platform Exadata sumber Anda untuk mengumpulkan informasi penting seperti karakteristik beban kerja, dependensi fitur Exadata, dan pertimbangan lainnya. Bagian ini juga mencakup cara memilih platform yang tepat untuk menampung beban kerja AWS dan cara mengukur instance target dengan tepat dengan menggunakan informasi yang Anda kumpulkan.

Untuk kuesioner yang dapat Anda gunakan sebagai titik awal untuk mengumpulkan informasi untuk fase penemuan proyek migrasi Anda, lihat [lampiran](#) dalam panduan panduan AWS preskriptif Migrasi database Oracle ke. AWS Cloud

Di bagian ini:

- [Karakteristik beban kerja](#)
- [Dependensi mesin database](#)
- [Edisi dan versi database](#)
- [Konsolidasi database](#)
- [Penggunaan fitur Exadata](#)
- [Alat untuk fase penemuan](#)
- [Persyaratan sumber daya untuk platform target](#)
- [Pengujian kinerja pada platform target](#)
- [Persyaratan aplikasi SLA](#)
- [Kebijakan manajemen dan retensi siklus hidup data](#)

- [Faktor lainnya](#)
- [Diagram alur keputusan](#)

Karakteristik beban kerja

Secara historis, platform komputasi basis data khusus dirancang untuk beban kerja tertentu, seperti pemrosesan transaksi online (OLTP) atau pemrosesan analitik online (OLAP), dan pola desain spesifik tersebut membuatnya menjadi pilihan yang buruk untuk beban kerja lainnya. Misalnya, database Oracle yang menghosting sistem pendukung keputusan biasanya menggunakan ukuran blok yang lebih besar untuk mendukung membaca lebih banyak data dari cache dengan operasi I/O yang lebih sedikit. Di sisi lain, beban kerja OLTP mendapat manfaat dari ukuran blok yang lebih kecil untuk mendukung akses acak ke baris kecil dan untuk mengurangi pertentangan blok. Exadata efektif dalam menjalankan semua jenis beban kerja database Oracle atau kombinasi beban kerja apa pun karena fitur seperti memori persisten (PMEM) dan Exadata Smart Flash Cache untuk meningkatkan kinerja transaksi OLTP, dan Hybrid Columnar Compression (HCC) dan Smart Scan untuk mendukung kueri analitis. Namun, memigrasikan beban kerja Exadata memberi Anda kesempatan yang baik untuk mempertimbangkan menggunakan mesin database yang dibuat khusus untuk beban kerja alih-alih menggunakan tipe atau instance database yang ada. [AWS database yang dibuat](#) khusus memudahkan untuk memilih jenis layanan tertentu untuk beban kerja tertentu pada model berbasis konsumsi daripada mencoba memaksa beberapa beban kerja ke platform yang sama. Seperti dibahas [sebelumnya](#), AWS menawarkan lebih dari 15 mesin yang dibuat khusus untuk mendukung beragam model data, termasuk relasional, nilai kunci, dokumen, dalam memori, grafik, deret waktu, dan database kolom lebar.

Secara tradisional, database yang dioptimalkan untuk sistem pendukung keputusan mengikuti pola desain dan karakteristik beban kerja tertentu seperti berikut:

- Ukuran blok database yang lebih besar (16K atau 32K)
- Skema bintang dengan tabel fakta dan dimensi dan `star_transformation_enabled` parameter diatur ke TRUE
- Fitur kompresi seperti HCC, Kompresi Lanjutan, atau Kompresi Dasar
- Fitur OLAP
- Kehadiran tampilan terwujud dalam database dengan `query_rewrite_enabled` set ke TRUE
- Pemrosesan paralel besar-besaran
- Jejak I/O yang berat

Di sisi lain, database yang dioptimalkan untuk OLTP memiliki ukuran blok database yang lebih kecil (8K atau lebih kecil), pembacaan blok tunggal, konkurensi berat, rasio hit cache buffer tinggi, dan eksekusi transaksi serial. Di Exadata, biasanya untuk melihat anti-pola di mana database yang dirancang untuk beban kerja OLTP banyak digunakan untuk kueri analitis, atau sebaliknya. Sangat tidak mungkin database Oracle digunakan untuk beban kerja OLTP murni, karena ini adalah praktik umum untuk menjalankan kueri pelaporan pada database transaksional untuk kenyamanan.

Berbagai statistik sistem yang tersedia dalam tampilan kinerja dinamis Oracle, laporan Automatic Workload Repository (AWR), dan laporan Statspack dapat mengungkapkan seberapa mirip beban kerja database dengan sistem OLTP atau OLAP. Statistik `Physical read total multi block requests` menunjukkan jumlah total permintaan baca yang dibaca dalam dua atau lebih blok database per permintaan. Perbedaan antara `Physical read total IO requests` dan `Physical read total multi block requests` menunjukkan jumlah total permintaan baca blok tunggal yang dikeluarkan oleh database. Sejumlah besar permintaan multi-blok biasanya menunjukkan sistem OLAP, dan sejumlah besar permintaan baca blok tunggal menunjukkan sistem OLTP. Selanjutnya, statistik berikut dalam laporan AWR juga dapat mengungkapkan apakah beban kerja yang berjalan pada database Oracle terutama merupakan beban kerja OLTP atau OLAP:

- `user commits`—Mencerminkan jumlah komitmen yang dikeluarkan pada batas transaksi. Biasanya, sistem OLTP memiliki jumlah transaksi kecil yang tinggi, yang menghasilkan sejumlah besar komitmen pengguna. Di sisi lain, sistem OLAP menjalankan sejumlah kecil transaksi berat.
- `Buffer hit`—Menunjukkan seberapa sering blok yang diminta ditemukan di cache buffer tanpa memerlukan akses disk. Sistem OLTP biasanya memiliki rasio hit buffer di atas 99 persen, sedangkan rasio hit buffer untuk sistem OLAP biasanya rendah.

Tabel berikut merangkum perbedaan umum dalam karakteristik beban kerja antara sistem OLTP dan OLAP.

Karakteristik	OLTP	OLAP
Ukuran blok	<= 8K	> 8K
Tingkat komit	Tinggi	Rendah
Rasio hit cache buffer	> 99%	< 99%
Acara tunggu I/O yang menonjol	File DB dibaca berurutan, sinkronisasi file log	File DB tersebar dibaca, jalur langsung dibaca

Karakteristik	OLTP	OLAP
Ukuran permintaan I/O rata-rata (throughput I/O/IOPS)	< 120K	> 400K
Skema bintang	Tidak ada	Mungkin ada
<code>star_transformation_enabled</code> parameter	SALAH	BETUL
Paralelisme	Derajat rendah atau dinonaktifkan	Diaktifkan dengan tingkat tinggi

Jika database Anda terutama mendukung beban kerja OLAP, solusi gudang data yang dibuat khusus seperti [Amazon Redshift](#) mungkin lebih cocok saat Anda memigrasikan beban kerja Anda ke AWS. Anda kemudian dapat membangun solusi analitis AWS dengan menggunakan layanan seperti [Amazon S3](#), [Amazon Athena](#), dan [Amazon QuickSight](#). Untuk beban kerja OLTP, Amazon RDS hadir dengan pilihan enam mesin relasional, termasuk [Amazon RDS for Oracle](#), jika Anda memiliki ketergantungan pada database Oracle. Jika tidak, Anda dapat memilih mesin open source seperti [Amazon RDS for PostgreSQL](#) atau [Aurora PostgreSQL](#) yang kompatibel. [Amazon DynamoDB](#) juga dapat menampung sistem transaksional yang sangat skalabel yang tidak memerlukan model relasional dan dapat dilayani oleh toko nilai kunci.

Rasio baca/tulis

Faktor penting lainnya adalah rasio baca/tulis dari beban kerja yang dihosting di database yang ingin Anda migrasikan. Sebagian besar sistem OLTP juga digunakan untuk tujuan pelaporan, dan ad-hoc, kueri intensif sumber daya dijalankan terhadap database transaksional kritis. Hal ini sering menyebabkan masalah kinerja dalam komponen aplikasi kritis. Kueri pelaporan yang kurang kritis dan intensif sumber daya dapat dialihkan ke salinan instance produksi untuk menghindari dampak kinerja apa pun terhadap aplikasi produksi kritis. `physical_writes` Statistik AWR mencerminkan jumlah total blok data yang ditulis ke disk, dan `physical_reads` statistik menentukan jumlah total blok data yang dibaca dari disk. Dengan menggunakan statistik ini, Anda dapat menentukan persentase baca beban kerja sebagai berikut:

$$\text{Read percentage} = \frac{\text{physical reads}}{(\text{physical reads} + \text{physical writes})} * 100$$

Bergantung pada bagaimana transaksi mengeluarkan operasi baca pada database, Anda dapat menerapkan solusi [replika baca](#) atau solusi caching yang berada di luar database — misalnya, [Amazon ElastiCache](#) — dalam arsitektur target. Ini membantu mengurangi sumber daya yang dibutuhkan instance database utama untuk melayani beban kerja baca. [Amazon Aurora](#), yang merupakan mesin database relasional cloud-native yang merupakan bagian dari keluarga Amazon RDS, menyediakan [opsi penskalaan otomatis yang mendukung beban kerja hanya-baca yang sangat skalabel](#) dengan hingga 15 instans baca. Anda juga dapat menggunakan [database global Aurora](#) untuk menjangkau beberapa Wilayah AWS dengan operasi baca lokal yang cepat dan latensi rendah di setiap Wilayah.

Beban kerja non-relasional

Oracle Database versi 12.c mendukung penyimpanan data JSON secara native dengan fitur database relasional. Pada 21c, Oracle Database memperkenalkan tipe data JSON. Selain itu, fitur Simple Oracle Document Access (SODA) memungkinkan Anda membuat, menyimpan, dan mengambil koleksi dokumen dengan menggunakan NoSQL. APIs Anda juga dapat menggunakan Oracle Graph Server untuk beban kerja grafik. [Namun, Anda dapat menjalankan beban kerja non-relasional tersebut dengan paling efisien saat menggunakan AWS database yang dibuat khusus seperti Amazon DynamoDB, Amazon DocumentDB, atau Amazon Neptune.](#) Layanan ini secara khusus dioptimalkan untuk pola akses NoSQL dan kasus penggunaan khusus.

Dependensi mesin database

[Banyak pelanggan mempertimbangkan untuk memigrasikan beban kerja mereka dari Oracle Database ke Amazon Aurora PostgreSQL yang kompatibel.](#) AWS Layanan ini menyediakan database cloud-native, relasional, hemat biaya dengan fitur kelas enterprise, peningkatan kinerja, dan keamanan tanpa biaya lisensi. [Migrasi heterogen dari Oracle Database ke PostgreSQL menjadi lebih mudah dengan \(\) dan \(\).](#) [AWS Database Migration Service](#) [AWS DMS](#) [AWS Schema Conversion Tool](#) [AWS SCT](#) AWS SCT membuat migrasi database heterogen dapat diprediksi. Ini secara otomatis mengubah sebagian besar objek skema dan kode ke platform target, dan juga memprediksi upaya yang diperlukan untuk mengonversi objek secara manual ketika konversi otomatis bukan merupakan pilihan.

Migrasi heterogen mungkin tidak layak di semua skenario migrasi. Misalnya, beban kerja yang melibatkan aplikasi paket Oracle seperti Oracle E-Business Suite (Oracle EBS) tidak dapat dengan mudah dimigrasikan ke PostgreSQL atau mesin database lainnya. Demikian pula, memodernisasi aplikasi yang bergantung pada fitur spesifik database Oracle, seperti Java Virtual Machine (JVM)

atau Advanced Compression, mungkin memerlukan lebih banyak waktu, tenaga, dan sumber daya. Selama fase penemuan, Anda harus menganalisis dependensi apa pun yang mungkin dimiliki aplikasi Anda di Oracle Database dan fitur-fiturnya. Pertimbangkan kelayakan memigrasikan beban kerja Anda ke mesin open source berdasarkan faktor-faktor seperti kompleksitas migrasi, upaya yang diperlukan, manfaat biaya, dan keahlian.

Edisi dan versi database

Jika beban kerja Exadata Anda dapat di-host di Oracle Database AWS, ada beberapa opsi yang dapat dipilih, termasuk Amazon RDS for Oracle, Amazon [RDS Custom for Oracle, instans yang dikelola sendiri di EC2 Amazon, dan opsi penyebaran Oracle Real Application Cluster \(RAC\)](#) aktif. AWS Anda harus mengevaluasi dependensi apa pun yang mungkin dimiliki aplikasi Anda pada edisi atau versi Oracle Database tertentu. Jika aplikasi Anda bergantung pada versi lama Oracle Database, Anda mungkin menghadapi tantangan saat mencoba menerapkan versi tersebut di Amazon RDS for Oracle, yang memberlakukan siklus hidup dukungan Oracle. Di sisi lain, Amazon RDS Custom for Oracle menggunakan kebijakan Bring Your Own Media (BYOM) dan Bring Your Own License (BYOL) yang saat ini memungkinkan Anda untuk menyebarkan versi lama Oracle Database seperti 12.1, 12.2, dan 18c.

Anda dapat mempertimbangkan untuk bermigrasi dari Oracle Database Enterprise Edition (EE) ke Standard Edition 2 (SE2) untuk mengurangi biaya lisensi. Memahami dependensi fitur dan perencanaan lanjutan strategi mitigasi adalah kunci keberhasilan migrasi dari Oracle Database EE ke SE2 Amazon RDS for [Oracle menyediakan dua](#) opsi lisensi: Lisensi Termasuk (LI) dan BYOL. Jika Anda menggunakan opsi LI untuk Oracle Database SE2, Anda tidak perlu membeli lisensi Oracle Database Anda secara terpisah. Anda dapat menjalankan Oracle Database SE2 dengan lisensi LI AWS tanpa kontrak dukungan dengan Oracle dan tidak ada biaya dukungan tahunan. Harga LI sudah termasuk perangkat lunak, sumber daya perangkat keras yang mendasarinya, dan kemampuan manajemen Amazon RDS. Dengan menggunakan Instans Sesuai Permintaan untuk model LI, Anda dapat membayar instans DB per jam tanpa komitmen jangka panjang.

AWS SCT dapat menganalisis penggunaan fitur khusus Oracle Database EE saat ini dari beban kerja Anda. Bagian Evaluasi Lisensi dan Dukungan Cloud pada AWS SCT laporan memberikan informasi terperinci tentang fitur Oracle yang digunakan untuk memastikan bahwa Anda dapat membuat keputusan berdasarkan informasi saat bermigrasi ke Amazon RDS for Oracle.

Jika beban kerja Anda menggunakan fitur Oracle Database EE dan opsi seperti Oracle Data Guard untuk ketersediaan tinggi atau Automatic Workload Repository (AWR), yang dilisensikan di bawah Oracle Diagnostics Pack, untuk mendiagnosis masalah kinerja, masih mungkin untuk pindah ke

Oracle Database pada. SE2 AWS Opsi Amazon RDS Multi-AZ menyediakan ketersediaan tinggi dan membantu mencegah kehilangan data. Fitur ini menggunakan replikasi penyimpanan tanpa bergantung pada Oracle Data Guard dan tersedia untuk Oracle Database EE dan. SE2 [Demikian pula, Anda dapat memenuhi persyaratan pemantauan kinerja Anda tanpa Oracle Diagnostics Pack, dengan menggunakan Oracle Statspack dengan kemampuan AWS pemantauan seperti Performance Insights, metrik Amazon CloudWatch , dan Enhanced Monitoring.](#)

Posting blog [Rethink Oracle Standard Edition Two di Amazon RDS for Oracle membahas berbagai taktik untuk](#) mengurangi kesenjangan fitur di Amazon RDS for Oracle saat Anda menggunakan Oracle Database. SE2 Kami juga menyarankan Anda meninjau publikasi AWS Prescriptive Guidance [Evaluating downgrade database Oracle ke Standard Edition 2 AWS dan Replatform Oracle Database Enterprise Edition ke Standard Edition 2 di Amazon RDS for Oracle.](#)

Konsolidasi database

Exadata dianggap sebagai platform yang nyaman untuk mengkonsolidasikan database Oracle ketika tujuan utamanya adalah untuk mengurangi biaya lingkungan database dengan pemanfaatan sumber daya infrastruktur yang lebih tinggi. Konsolidasi database di Exadata membantu membenarkan tingginya biaya Exadata ketika satu beban kerja database tidak dapat sepenuhnya memanfaatkan semua sumber daya dan kemampuan sistem Exadata. Konsolidasi juga membantu meningkatkan efisiensi operasional dan standarisasi.

Strategi umum pada platform Exadata meliputi konsolidasi:

- Beberapa database yang merupakan bagian dari satu Real Application Cluster (RAC) dalam sistem Exadata
- Beberapa database yang digunakan di bawah yang berbeda RACs atau kombinasi RAC dan database instans tunggal
- Beberapa database pluggable (PDBs) dalam database kontainer
- Beberapa skema dalam satu database

Strategi konsolidasi ini sering menyulitkan untuk memenuhi berbagai tingkat keamanan, skalabilitas, kinerja, dan persyaratan SLA yang terkait dengan beban kerja yang dikonsolidasikan di Exadata.

Pada AWS, Anda dapat menskalakan sumber daya dengan mudah dan mengadopsi model penerapan yang hemat biaya tanpa mengkonsolidasikan database Anda. Namun, Anda mungkin masih ingin mengkonsolidasikan database dan skema Anda di AWS lingkungan target karena

berbagai alasan, termasuk saling ketergantungan yang kompleks antara skema atau tautan basis data latensi rendah antara beberapa database.

Pertimbangan untuk mengkonsolidasikan database Oracle Anda pada: AWS

- Anda dapat menerapkan strategi konsolidasi skema dengan model penyebaran Oracle apa pun. AWS
- Amazon RDS for Oracle dan Amazon RDS Custom for Oracle mendukung arsitektur multitenant dengan beberapa database pluggable di dalam database container.

Penggunaan fitur Exadata

Bagian ini membahas fitur Exadata utama yang harus Anda pertimbangkan saat memigrasikan beban kerja Exadata. Fitur-fitur ini termasuk Smart Scan, Hybrid Columnar Compression (HCC), indeks penyimpanan, dan Persistent Memory (PMEM). Bagian ini juga membahas cara-cara untuk menilai ketergantungan fitur Exadata dari beban kerja Anda, cara mengukur tingkat penggunaan fitur Exadata, dan strategi untuk memenuhi persyaratan aplikasi Anda di platform target tanpa menggunakan fitur khusus Exadata.

Note

Oracle meningkatkan Exadata dengan memperkenalkan fitur perangkat keras dan perangkat lunak baru secara teratur. Ini di luar cakupan panduan ini untuk mencakup semua fitur tersebut.

Di bagian ini:

- [Pemindaian Cerdas](#)
- [Indeks penyimpanan](#)
- [Cache Flash Cerdas](#)
- [Kompresi Kolumnar Hibrida](#)
- [Manajemen Sumber Daya I/O](#)
- [Memori Persisten \(PMEM\)](#)
- [Ringkasan fitur dan alternatif Exadata AWS](#)

Pemindaian Cerdas

Exadata menggunakan subsistem penyimpanan yang sadar basis data untuk membongkar pemrosesan dari server database dengan memindahkan beberapa pemrosesan SQL ke server sel penyimpanan. Exadata Smart Scan dapat mengurangi volume data yang dikembalikan ke server database melalui filtrasi offloaded dan proyeksi kolom. Fitur ini memecahkan dua tantangan utama dalam menangani kumpulan data besar: pemindahan data besar dan tidak perlu dari lapisan penyimpanan ke server database, dan waktu dan sumber daya yang dihabiskan untuk memfilter data yang diperlukan. Smart Scan adalah kemampuan penting dari Cell Offload Processing, yang juga mencakup inisialisasi datafile, dekompresi HCC, dan fungsionalitas lainnya.

Aliran data dari Smart Scan tidak dapat di-buffer di kumpulan buffer system global area (SGA). Smart Scan membutuhkan pembacaan jalur langsung, yang di-buffer di area global program (PGA). Pernyataan SQL harus memenuhi beberapa persyaratan untuk bekerja dengan Smart Scan:

- Segmen yang ditanyakan oleh pernyataan SQL harus disimpan dalam sistem Exadata di mana atribut pengaturan `cell.smart_scan_capable` grup disk ASM diatur ke `TRUE`
- Pemindaian tabel penuh atau operasi pemindaian penuh indeks cepat harus terjadi.
- Segmen yang terlibat dalam pernyataan SQL harus cukup besar untuk menjalani [operasi baca jalur langsung](#).

Untuk menilai efisiensi Smart Scan dalam sistem Exadata, Anda harus mempertimbangkan statistik basis data utama berikut:

- `physical read total bytes`— Jumlah total byte I/O untuk operasi baca yang dikeluarkan oleh database, terlepas dari apakah operasi diturunkan ke server penyimpanan. Ini menunjukkan total operasi baca, dalam byte, yang dikeluarkan oleh server database ke sel penyimpanan Exadata. Nilai ini mencerminkan kapasitas I/O baca yang harus dipenuhi oleh platform target di AWS saat Anda memigrasikan beban kerja ke AWS tanpa menyetelnya.
- `cell physical IO bytes eligible for predicate offload`— Jumlah operasi baca, dalam byte, yang dimasukkan ke Smart Scan dan memenuhi syarat untuk pembongkaran predikat.
- `cell physical IO interconnect bytes`— Jumlah byte I/O yang dipertukarkan melalui interkoneksi antara server database dan sel penyimpanan. Ini mencakup semua jenis lalu lintas I/O antara database dan node penyimpanan, termasuk byte yang dikembalikan oleh Smart Scan, byte yang dikembalikan oleh kueri yang tidak memenuhi syarat untuk Smart Scan, dan operasi tulis.

- `cell physical IO interconnect bytes returned by smart scan`— Byte I/O dikembalikan oleh sel untuk operasi Smart Scan. Ini adalah output dari Smart Scan.
- `cell physical IO bytes eligible for predicate offload`— Anda dapat membandingkan nilai ini dengan total byte baca fisik untuk memahami berapa banyak operasi baca total yang tunduk pada Smart Scan. Rasio `cell physical IO bytes eligible for predicate offload` (input untuk Smart Scan) ke `cell physical IO interconnect bytes returned by smart scan` (output Smart Scan) menunjukkan efisiensi Smart Scan. Untuk sistem Exadata yang mencakup sebagian besar operasi baca, rasio `cell physical IO interconnect bytes returned by smart scan` to `cell physical IO interconnect bytes` dapat menunjukkan ketergantungan pada Smart Scan. Namun, ini mungkin tidak selalu terjadi, karena `cell physical IO interconnect bytes` juga mencakup dua kali lipat jumlah operasi tulis (dengan pencerminan ASM) antara server komputasi dan penyimpanan.

[Anda bisa mendapatkan statistik I/O database ini dan metrik khusus Exadata dari laporan AWR atau dengan langsung menanyakan tampilan V\\$ yang mendasarinya seperti,, dan.](#) `V$SYSSTAT V$ACTIVE_SESSION_HISTORY V$SQL`

Dalam contoh berikut dari laporan AWR yang dikumpulkan dari sistem Exadata, database meminta throughput baca 5,7 Gbps, 5,4 Gbps di antaranya memenuhi syarat untuk Smart Scan. Output Smart Scan berkontribusi pada 55 MBps dari 395 MBps total lalu lintas interkoneksi antara database dan node komputasi. Statistik ini menunjuk ke sistem Exadata yang memiliki ketergantungan tinggi pada Smart Scan.

Statistic	Total	per Second
<code>physical read total bytes</code>	41,486,341,567,488	5,758,375,137.90
<code>cell physical IO bytes eligible for predicate offload</code>	39,217,360,822,272	5,443,436,754.68
<code>cell physical IO interconnect bytes</code>	2,846,913,082,080	395,156,370.37
<code>cell physical IO interconnect bytes returned by smart scan</code>	400,725,918,720	55,621,456.14

Anda dapat menilai efisiensi dan dependensi Smart Scan di tingkat SQL dengan menggunakan kolom tampilan berikut. `V$SQL`

- `IO_CELL_OFFLOAD_ELIGIBLE_BYTES`— Jumlah byte I/O yang dapat disaring oleh sistem penyimpanan Exadata.
- `IO_INTERCONNECT_BYTES`— Jumlah byte I/O yang dipertukarkan antara database Oracle dan sistem penyimpanan.
- `PHYSICAL_READ_BYTES`— Jumlah byte yang dibaca dari disk oleh SQL yang dipantau.

Output kueri berikut menunjukkan manfaat Smart Scan untuk kueri SQL yang memiliki ID SQL. xn2fg7abff2d

```
select
  ROUND(physical_read_bytes/1048576) phyrd_mb
  , ROUND(io_cell_offload_eligible_bytes/1048576) elig_mb
  , ROUND(io_interconnect_bytes/1048576) ret_mb
  , (1-(io_interconnect_bytes/NULLIF(physical_read_bytes,0)))*100 "SAVING%"
from v$sql
where sql_id = 'xn2fg7abff2d' and child_number = 1;
PHYRD_MB      ELIG_MB      RET_MB      SAVING%
-----      -
10815          10815      3328        69.2%
```

Untuk menguji pengaruh Smart Scan pada beban kerja, Anda dapat menonaktifkan fitur dengan mengatur `cell_offload_processing` parameter ke `FALSE` tingkat sistem, sesi, atau kueri. Misalnya, untuk menonaktifkan pemrosesan pembongkaran sel Exadata Storage Server untuk pernyataan SQL, Anda dapat menggunakan:

```
select /*+ OPT_PARAM('cell_offload_processing' 'false') */ max(ORDER_DATE) from SALES;
```

Untuk menonaktifkan pemrosesan pembongkaran sel Exadata Storage Server untuk sesi database, Anda dapat mengatur parameter inisialisasi database Oracle berikut:

```
alter session set CELL_OFFLOAD_PROCESSING=FALSE;
```

Untuk menonaktifkan pemrosesan pembongkaran sel Exadata Storage Server untuk seluruh database Exadata, Anda dapat mengatur:

```
alter system set CELL_OFFLOAD_PROCESSING=FALSE;
```

Migrasi ke AWS

Saat Anda awalnya memigrasikan beban kerja ke Exadata, beberapa perubahan desain diimplementasikan sebagai praktik umum untuk mendukung Smart Scan, termasuk menjatuhkan indeks skema untuk mendukung pemindaian tabel penuh. Saat Anda memigrasikan beban kerja tersebut ke platform non-Exadata, Anda perlu membalikkan perubahan desain tersebut.

Saat memigrasikan beban kerja Exadata ke AWS, pertimbangkan tindakan penyetelan ini untuk mengoptimalkan performa kueri yang menggunakan Smart Scan:

- Gunakan instance yang dioptimalkan untuk memori dan konfigurasi SGA yang lebih besar untuk meningkatkan rasio hit buffer.
- Identifikasi kueri yang berjalan dengan rencana eksekusi suboptimal dan sesuaikan untuk mengurangi jejak I/O mereka.
- Sesuaikan parameter pengoptimal seperti `db_file_multiblock_read_count` dan `optimizer_index_cost_adj` untuk menghindari pemindaian tabel penuh.
- Pilih opsi kompresi yang sesuai.
- Buat indeks skema tambahan sesuai kebutuhan.

Indeks penyimpanan

Indeks penyimpanan adalah struktur berbasis memori yang mengurangi jumlah I/O fisik yang dilakukan dalam sel penyimpanan Exadata. Indeks penyimpanan melacak nilai kolom minimum dan maksimum, dan informasi ini digunakan untuk menghindari operasi I/O yang tidak perlu. Indeks penyimpanan memungkinkan Exadata mempercepat operasi I/O dengan menghilangkan akses ke wilayah penyimpanan yang tidak berisi data yang dicari kueri.

Statistik database berikut membantu menilai manfaat indeks penyimpanan dalam sistem:

- `cell physical IO bytes saved by storage index`— Menunjukkan berapa banyak byte I/O yang dihilangkan dengan penerapan indeks penyimpanan pada tingkat sel penyimpanan.
- `cell IO uncompressed bytes`— Mencerminkan volume data untuk pembongkaran predikat setelah penyaringan indeks penyimpanan dan dekompresi apa pun.

Untuk informasi lebih lanjut tentang ini, lihat [dokumentasi Oracle](#). Dalam contoh berikut dari laporan AWR yang dikumpulkan dari sistem Exadata, operasi baca 5,4 Gbps memenuhi syarat Smart Scan. 4,6 Gbps dari operasi I/O tersebut diproses oleh sel sebelum pembongkaran predikat, dan 55 MBps dikembalikan ke node komputasi dengan penghematan 820 I/O oleh indeks penyimpanan. MBps Dalam contoh ini, ketergantungan pada indeks penyimpanan tidak terlalu tinggi.

Statistic	Total	per Second
cell physical IO bytes eligible for predicate offload	39,217,360,822,272	5,443,436,754.68
cell physical IO interconnect bytes returned by smart scan	400,725,918,720	55,621,456.14
cell physical IO bytes saved by storage index	5,913,287,524,352	820,775,330.00
cell IO uncompressed bytes	33,217,076,600,832	4,610,586,117.33

Migrasi ke AWS

Jika Anda bermigrasi ke platform yang tidak menyediakan indeks penyimpanan, dalam banyak kasus, Anda dapat membuat indeks skema untuk menghindari pemindaian tabel lengkap dan mengurangi jumlah blok yang diakses oleh kueri. Untuk menguji pengaruh indeks penyimpanan pada kinerja beban kerja Anda, setel `kcfis_storageidx_disabled` parameter ke `TRUE` tingkat sistem, sesi, atau kueri.

Misalnya, gunakan pernyataan SQL berikut untuk menonaktifkan indeks penyimpanan di tingkat sesi:

```
alter session set "_KCFIS_STORAGEIDX_DISABLED"=TRUE;
```

Cache Flash Cerdas

Fitur Exadata Smart Flash Cache menyimpan objek database dalam memori flash untuk meningkatkan kecepatan mengakses objek database. Smart Flash Cache dapat menentukan jenis segmen data dan operasi mana yang perlu di-cache. Ini mengenali berbagai jenis permintaan I/O sehingga akses data yang tidak dapat diulang (seperti I/O cadangan RMAN) tidak membersihkan blok database dari cache. Anda dapat memindahkan hot table dan indeks ke Smart Flash Cache dengan `ALTER` perintah. Saat Anda menggunakan fitur Write Back Flash Cache, Smart Flash juga dapat menyimpan operasi penulisan blok database cache.

Perangkat lunak server penyimpanan Exadata juga menyediakan Smart Flash Logging untuk mempercepat operasi penulisan ulang log dan mengurangi waktu layanan untuk acara sinkronisasi file log. Fitur ini melakukan operasi redo write secara bersamaan ke memori flash dan cache pengontrol disk, dan menyelesaikan operasi tulis ketika yang pertama dari keduanya selesai.

Dua statistik berikut memberikan wawasan cepat tentang kinerja Exadata Smart Flash Cache. Ini tersedia dalam tampilan kinerja dinamis seperti `V$SYSSTAT` dan di bagian Statistik Aktivitas Global atau Statistik Aktivitas Instans pada laporan AWR.

- `Cell Flash Cache read hits`— Mencatat jumlah permintaan baca yang menemukan kecocokan di Smart Flash Cache.
- `Physical read requests optimized`— mencatat jumlah permintaan baca yang dioptimalkan baik oleh Smart Flash Cache atau melalui indeks penyimpanan.

Metrik Exadata yang dikumpulkan dari sel penyimpanan juga berguna untuk memahami bagaimana beban kerja menggunakan Smart Flash Cache. Perintah [CellCLI](#) berikut mencantumkan metrik berbeda yang tersedia untuk memantau penggunaan Smart Flash Cache.

```
CellCLI> LIST METRICDEFINITION ATTRIBUTES NAME,DESCRIPTION WHERE OBJECTTYPE =
FLASHCACHE
FC_BYKEEP_DIRTY          "Number of megabytes unflushed for keep objects
on FlashCache"
FC_BYKEEP_OLTP           "Number of megabytes for OLTP keep objects in
flash cache"
FC_BYKEEP_OVERWR        "Number of megabytes pushed out of the FlashCache
because of space limit
for keep objects"
FC_BYKEEP_OVERWR_SEC     "Number of megabytes per second pushed out of the
FlashCache because of
space limit for keep objects"
...
```

Migrasi ke AWS

Smart Flash Cache tidak ada di AWS. Ada beberapa opsi untuk mengurangi tantangan ini dan menghindari penurunan kinerja saat memigrasikan beban kerja Exadata ke AWS, termasuk ini, yang dibahas di bagian berikut:

- Menggunakan instance memori yang diperluas
- Menggunakan instance dengan penyimpanan instance NVMe berbasis
- Menggunakan opsi AWS penyimpanan untuk latensi rendah dan throughput tinggi

Namun, opsi ini tidak dapat mereproduksi perilaku Smart Flash Cache, jadi Anda perlu menilai kinerja beban kerja Anda untuk memastikan bahwa itu terus memenuhi kinerja SLAs Anda.

Instans memori yang diperluas

Amazon EC2 menawarkan banyak instans memori tinggi, termasuk [instans dengan memori 12 TiB dan 24 TiB](#). Instans ini mendukung Oracle SGAs yang sangat besar yang dapat mengurangi dampak Smart Flash Cache yang hilang dengan meningkatkan rasio hit buffer.

Instance dengan penyimpanan instance NVMe berbasis

Toko instance menyediakan penyimpanan tingkat blok sementara untuk instance tersebut. Penyimpanan ini terletak pada disk yang secara fisik terpasang pada komputer host. Penyimpanan

instans memungkinkan beban kerja mencapai latensi rendah dan throughput yang lebih tinggi dengan menyimpan data pada NVMe disk berbasis. Data dalam penyimpanan instance hanya bertahan selama masa pakai instance, jadi penyimpanan instance ideal untuk ruang tabel dan cache sementara. Penyimpanan instans dapat mendukung jutaan IOPS dan throughput lebih dari 10 Gbps pada latensi mikrodetik tergantung pada jenis instans dan ukuran I/O. Untuk informasi selengkapnya tentang IOPS baca/tulis penyimpanan instans dan dukungan throughput untuk kelas instans yang berbeda, lihat instans [tujuan umum](#), [komputasi yang dioptimalkan](#), [dioptimalkan memori](#), dan [penyimpanan yang dioptimalkan dalam dokumentasi](#) Amazon. EC2

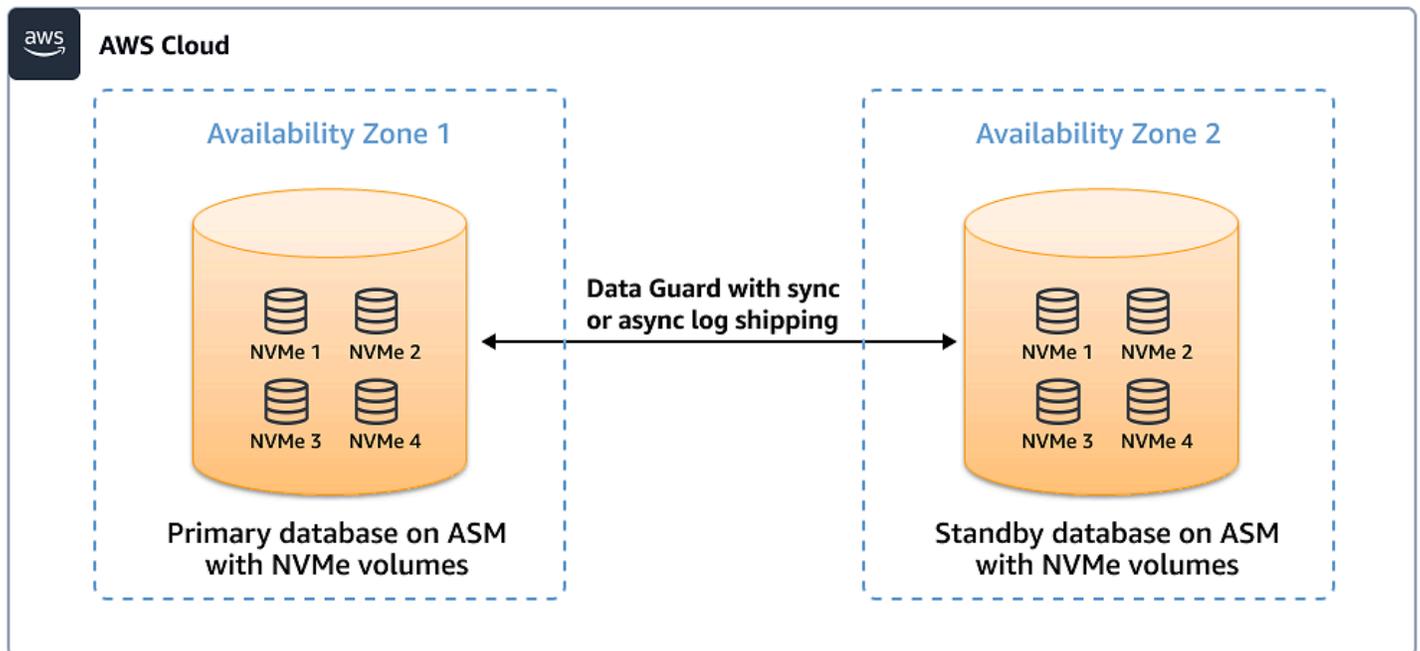
Di Exadata, Database Flash Cache memungkinkan pengguna untuk menentukan tingkat cache buffer kedua pada volume penyimpanan instance dengan latensi I/O rata-rata 100 mikrodetik untuk meningkatkan kinerja beban kerja baca. Anda dapat mengaktifkan cache ini dengan mengatur dua parameter inisialisasi database:

- `db_flash_cache_file = /<device_name>`
- `db_flash_cache_size = <size>G`

Anda juga dapat merancang arsitektur berkinerja tinggi untuk database Oracle yang di-host di Amazon EC2 dengan menempatkan file database di penyimpanan instance, dan menggunakan redundansi yang disediakan oleh Oracle Automatic Storage Management (ASM) dan Data Guard untuk perlindungan dan pemulihan data jika data hilang di penyimpanan instans. Pola arsitektur ini ideal untuk aplikasi yang memerlukan throughput I/O ekstrim pada latensi rendah dan mampu membeli RTO yang lebih tinggi untuk memulihkan sistem dalam skenario kegagalan tertentu. Bagian berikut secara singkat membahas dua arsitektur yang menyertakan file database yang dihosting di toko instance NVMe berbasis.

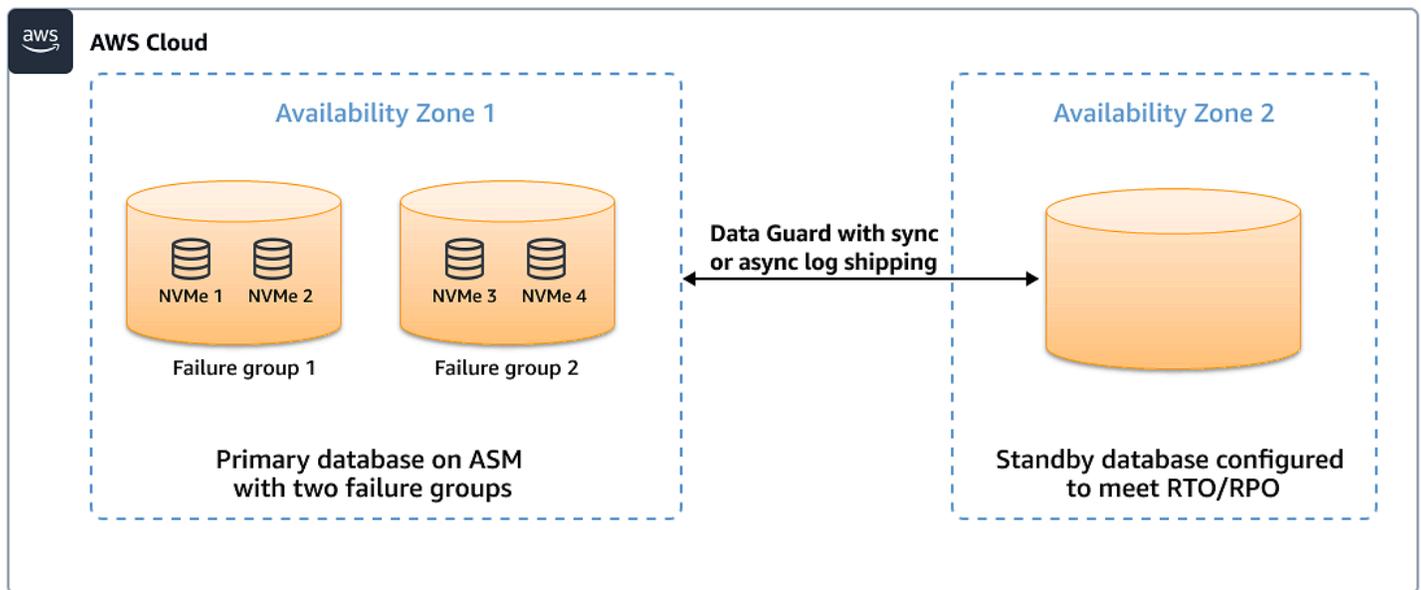
Arsitektur 1. Database di-host di penyimpanan instans pada instans primer dan siaga dengan Data Guard untuk perlindungan data

Dalam arsitektur ini, database di-host pada grup disk Oracle ASM untuk mendistribusikan I/O di beberapa volume penyimpanan instance untuk I/O throughput tinggi dan latensi rendah. Siaga Data Guard ditempatkan di Availability Zone yang sama atau di Availability Zone lain untuk perlindungan dari kehilangan data di penyimpanan instance. Konfigurasi grup disk tergantung pada RPO dan komit latensi. Jika penyimpanan instance hilang pada instance utama karena alasan apa pun, database dapat gagal ke siaga dengan nol atau kehilangan data minimum. Anda dapat mengonfigurasi proses pengamat Data Guard untuk mengotomatiskan failover. Operasi baca dan tulis mendapat manfaat dari throughput tinggi dan latensi rendah yang ditawarkan oleh toko instans.



Arsitektur 2. Database di-host pada grup disk ASM dengan dua grup kegagalan yang menggabungkan volume EBS dan penyimpanan instans

Dalam arsitektur ini, semua operasi baca dilakukan dari penyimpanan instance lokal dengan menggunakan `ASM_PREFERRED_READ_FAILURE_GROUP` parameter. Operasi penulisan berlaku untuk volume penyimpanan instans dan volume Amazon Elastic Block Store (Amazon EBS). Namun, bandwidth Amazon EBS didedikasikan untuk menulis operasi karena operasi baca diturunkan ke volume penyimpanan instans. Jika terjadi kehilangan data di penyimpanan instans, Anda dapat memulihkan data dari grup kegagalan ASM berdasarkan volume EBS atau dari database siaga. Untuk informasi lebih lanjut, lihat white paper Oracle [Mirroring and Failure Groups with ASM](#). Anda dapat menerapkan siaga Data Guard di Availability Zone yang berbeda untuk perlindungan tambahan.



Amazon RDS for [Oracle mendukung Database Smart Flash Cache dan](#) ruang meja sementara di toko instans. Beban kerja database Oracle dapat menggunakan fitur ini untuk mencapai latensi yang lebih rendah untuk operasi baca, throughput yang lebih tinggi, dan pemanfaatan bandwidth Amazon EBS yang efisien untuk operasi I/O database lainnya. Fitur ini saat ini didukung pada kelas instance db.m5d, db.r5d, db.x2idn, dan db.x2iedn. Untuk informasi terbaru, lihat [Kelas instans yang didukung untuk penyimpanan instans RDS untuk Oracle dalam dokumentasi](#) Amazon RDS.

Opsi penyimpanan AWS untuk beban kerja yang menuntut latensi rendah dan throughput tinggi

Jenis volume EBS yang saat ini didukung Amazon RDS for Oracle, [gp2, gp3, dan io1](#), didasarkan pada [solid-state drive](#) (). SSDs Saat Anda menerapkan jenis volume ini dengan [kelas instans yang dioptimalkan Amazon EBS](#) yang sesuai, mereka biasanya dapat memenuhi waktu layanan IOPs, dan persyaratan throughput Anda.

Untuk penerapan database Oracle yang dikelola sendiri di Amazon EC2, volume Amazon [EBS io2 dan io2 Block Express EBS](#) memberikan pilihan tambahan untuk beban kerja yang membutuhkan latensi lebih rendah dan throughput yang lebih tinggi.

Beban kerja yang membutuhkan throughput lebih tinggi atau latensi mikrodetik dapat menggunakan volume penyimpanan yang tidak didasarkan pada Amazon EBS saat menerapkan sebagai database Oracle yang dikelola sendiri di Amazon. EC2 Misalnya, [Amazon FSx untuk OpenZFS](#) dapat memberikan lebih dari 1 juta IOPS dengan throughput 20 Gbps atau lebih tinggi dengan latensi beberapa ratus mikrodetik. [Amazon FSx untuk NetApp ONTAP](#) dapat mengirimkan ratusan ribu IOPS dengan latensi kurang dari satu milidetik.

Kompresi Kolumnar Hibrida

Oracle Hybrid Columnar Compression (HCC) di Exadata memungkinkan rasio kompresi tertinggi di antara opsi kompresi yang tersedia untuk database Oracle. Ini menggunakan database dan kemampuan penyimpanan Exadata untuk mencapai rasio kompresi tinggi yang mengarah pada pengurangan biaya penyimpanan dan kinerja yang lebih baik untuk beban kerja tertentu karena berkurangnya I/O. Ada dua opsi HCC: Kompresi Gudang dan Kompresi Arsip. Kompresi Gudang mengurangi biaya penyimpanan dan memberikan kinerja yang lebih baik saat Anda menggunakan kueri Smart Scan untuk mendekompresi unit kompresi HCC dalam sel penyimpanan. Archive Compression adalah solusi manajemen siklus hidup informasi (ILM) yang memberikan rasio kompresi yang lebih tinggi dengan biaya overhead kinerja dan dimaksudkan untuk data yang jarang diakses.

Anda dapat menggunakan kueri berikut untuk mengidentifikasi tabel yang mengaktifkan kompresi:

```
select table_name, compression, compress_for from dba_tables where compression = 'ENABLED';
```

Untuk tabel berkemampuan HCC, `compress_for` kolom menunjukkan salah satu nilai berikut tergantung pada konfigurasi:

```
QUERY LOW, QUERY HIGH, ARCHIVE LOW, ARCHIVE HIGH
```

Selain itu, Anda dapat menggunakan `DBMS_COMPRESSION.GET_COMPRESSION_TYPE` fungsi untuk memahami konfigurasi HCC segmen, dan `dbms_compression.get_compression_ratio` prosedur untuk menganalisis rasio kompresi segmen yang diaktifkan untuk menggunakan HCC.

Dalam contoh berikut, `TEST_HCC` adalah tabel yang berukuran sekitar 30 MB. Ini diaktifkan HCC melalui penggunaan opsi `ARCHIVE HIGH` Output `dbms_compression.get_compression_ratio` menunjukkan bahwa tabel mendapat rasio kompresi 19,4.

Tanpa HCC, tabel ini akan tumbuh menjadi sekitar 580 MB.

```
SET SERVEROUTPUT ON

DECLARE
l_blkcnt_cmp PLS_INTEGER;
l_blkcnt_uncmp PLS_INTEGER;
l_row_cmp PLS_INTEGER;
l_row_uncmp PLS_INTEGER;
```

```

l_cmp_ratio NUMBER;
l_comptype_str VARCHAR2(32767);

BEGIN
DBMS_COMPRESSION.get_compression_ratio (
    scratchtbsname => 'USERS',
    ownname => upper('TEST_USER'),
    objname => upper('TEST_HCC'),
    subobjname => NULL,
    comptype => DBMS_COMPRESSION.COMP_ARCHIVE_HIGH,
    blkcnt_cmp => l_blkcnt_cmp,
    blkcnt_uncmp => l_blkcnt_uncmp,
    row_cmp => l_row_cmp,
    row_uncmp => l_row_uncmp,
    cmp_ratio => l_cmp_ratio,
    comptype_str => l_comptype_str,
    subset_numrows => DBMS_COMPRESSION.comp_ratio_allrows,
    objtype SQL> => DBMS_COMPRESSION.objtype_table
);

DBMS_OUTPUT.put_line('Number of blocks used (compressed) : ' || l_blkcnt_cmp);
DBMS_OUTPUT.put_line('Number of blocks used (uncompressed) : ' || l_blkcnt_uncmp);
DBMS_OUTPUT.put_line('Number of rows in a block (compressed) : ' || l_row_cmp);
DBMS_OUTPUT.put_line('Number of rows in a block (uncompressed) : ' || l_row_uncmp);
DBMS_OUTPUT.put_line('Compression ratio : ' || l_cmp_ratio);
DBMS_OUTPUT.put_line('Compression type : ' || l_comptype_str);
END;
/
Compression Advisor self-check validation successful. select count(*) on both
Uncompressed and EHCC Compressed format = 3851900 rows
Number of blocks used (compressed) : 3816
Number of blocks used (uncompressed) : 74263
Number of rows in a block (compressed) : 1009
Number of rows in a block (uncompressed) : 51
Compression ratio : 19.4
Compression type : "Compress Archive High"
PL/SQL procedure successfully completed.

```

Migrasi ke AWS

Karena HCC adalah teknologi kompresi yang bergantung pada perangkat keras, segmen yang diaktifkan untuk HCC harus tidak dikompresi selama migrasi ke platform target aktif. AWS Merupakan praktik umum untuk menyimpan data yang diarsipkan bersama dengan data yang lebih jarang

diakses di Exadata karena rasio kompresi tinggi yang ditawarkan oleh fitur Exadata HCC. [Untuk mengatasi tantangan mengelola kumpulan data yang lebih besar AWS tanpa HCC, pertimbangkan untuk memindahkan bagian yang tidak aktif dari kumpulan data Anda dari basis data utama Anda dan menyimpannya dalam solusi penyimpanan murah dan efisien lainnya seperti Amazon S3 Intelligent-Tiering.](#) Ini mungkin memerlukan perubahan dalam logika aplikasi atau alur kerja tergantung pada bagaimana aplikasi Anda mengakses data yang tidak aktif. Untuk informasi tambahan, lihat [bagian Manajemen siklus hidup data dari panduan](#) ini.

Untuk beban kerja yang memiliki ketergantungan pada Oracle Database, segmen yang mendukung HCC juga dapat dikonversi untuk menggunakan fitur kompresi dasar atau lanjutan yang ditawarkan oleh Oracle Database. Kompresi dasar dan lanjutan hanya didukung di Oracle Database EE. Kompresi lanjutan membutuhkan lisensi tambahan. Amazon EC2 dan Amazon RDS mendukung kedua opsi kompresi ini.

Manajemen Sumber Daya I/O

I/O Resource Management (IORM) adalah fitur Exadata yang mengelola bagaimana beberapa beban kerja dan database berbagi sumber daya I/O dari sistem Exadata. IORM melengkapi Oracle Database Resource Manager (DBRM) untuk menyediakan isolasi yang diperlukan untuk beban kerja yang berbeda dalam lingkungan yang terkonsolidasi. Setiap kali permintaan I/O mulai memenuhi kapasitas I/O server sel penyimpanan, IORM menjadwalkan dan memprioritaskan permintaan I/O yang masuk berdasarkan paket sumber daya yang Anda konfigurasi.

Anda dapat mengumpulkan metrik IORM dari sel penyimpanan Exadata dengan menggunakan skrip `metric_iorm.pl` seperti yang dibahas di My Oracle Support (MOS) Note 337265.1, [Alat untuk Mengumpulkan Metrik Manajer Sumber Daya I/O: metric_iorm.pl](#) (memerlukan akun Oracle). Metrik ini dapat berguna untuk mengatur beban kerja yang berjalan di lingkungan terkonsolidasi di Exadata saat Anda memigrasikan beban kerja ke platform target di AWS.

Migrasi ke AWS

Dalam AWS Cloud, kami menyarankan Anda meng-host beban kerja yang berbeda pada instance terpisah. Pendekatan ini memberikan lebih banyak fleksibilitas dalam memelihara database sesuai dengan sumber daya, kinerja, dan persyaratan SLA dari aplikasi individu alih-alih mengkonsolidasikannya menjadi satu contoh. Praktik berikut dapat berguna saat Anda memigrasikan beban kerja tersebut ke: AWS

- Identifikasi saling ketergantungan antar database dan klasifikasikan beban kerja yang harus dimigrasikan ke instance yang sama pada platform target. Database ini mungkin memiliki referensi lintas skema yang tidak dapat diselesaikan atau konektivitas tautan basis data latensi rendah.
- Berdasarkan statistik yang Anda kumpulkan dengan menggunakan `metric_iorm.pl` skrip, identifikasi database dan beban kerja yang memulai dan mendapat manfaat dari IORM. Gunakan informasi ini untuk menentukan database yang dapat dikonsolidasikan atau dimigrasikan ke instance independen. Pilih jenis penyimpanan dan kelas instance yang sesuai untuk menghindari saturasi I/O.
- Jika platform targetnya adalah Oracle Database, pertimbangkan untuk menggunakan [Oracle Database Resource Manager \(DBRM\)](#) untuk memprioritaskan atau membatasi sumber daya seperti CPU, PGA, dan paralelisme untuk beberapa beban kerja yang dikonsolidasikan dalam contoh yang sama dengan beberapa database atau skema pluggable.
- Pertimbangkan untuk menerapkan solusi caching seperti [Amazon ElastiCache dan Amazon RDS for Oracle read replicas untuk menyajikan beban kerja hanya-baca](#). Solusi ini mengurangi jejak I/O pada instance utama.
- Untuk beban kerja yang tidak memiliki ketergantungan pada Oracle Database, [Amazon Aurora](#) menyediakan arsitektur terdistribusi dan dipisahkan yang menyediakan throughput I/O tinggi. [Anda dapat memenuhi tuntutan beban kerja yang berat dan intensif I/O dengan merancang cluster Aurora dengan jumlah instance pembaca yang sesuai dan dengan menggunakan fitur seperti database global Amazon Aurora](#).

Memori Persisten (PMEM)

Oracle Exadata X8M dan rilis yang lebih baru menggunakan Persistent Memory (PMEM) untuk mencapai tingkat I/O yang lebih tinggi serta akses penyimpanan latensi rendah. Exadata mampu mencapai latensi penyimpanan kurang dari 19 mikrodetik dengan PMEM dikombinasikan dengan Remote Direct Memory Access over Converged Ethernet (RoCE) untuk memotong lapisan kode. Cache PMEM bekerja bersama dengan Exadata Smart Flash Cache untuk menyediakan tiga tingkatan lapisan penyimpanan: PMEM bertindak sebagai tingkat penyimpanan panas, Smart Flash Cache sebagai tingkat penyimpanan hangat, dan disk dalam sel penyimpanan sebagai tingkat penyimpanan dingin untuk memberikan IOPS yang lebih tinggi dan peningkatan kinerja untuk operasi komit.

Manfaat kinerja PMEM dapat dilihat dari statistik AWR sebagai waktu layanan yang rendah, dalam mikrodetik, untuk peristiwa tunggu baca seperti pembacaan fisik blok tunggal sel, dan redo log write wait event seperti sinkronisasi file log dan file log parallel write. Anda juga dapat memantau klik cache

PMEM dengan menggunakan statistik tambahan seperti klik baca cache pmem sel dan penulisan cache pmem sel, yang tersedia dalam tampilan kinerja dinamis seperti V\$SYSSTAT dan dalam laporan AWR.

Migrasi ke AWS

EC2 instans di AWS saat ini tidak menawarkan fitur PMEM. Namun, EC2 instance dengan kemampuan memori yang besar dapat mendukung Oracle yang sangat besar SGAs yang dapat menyimpan objek Oracle Database. Untuk beban kerja yang membutuhkan waktu layanan baca dan tulis dalam mikrodetik, [Amazon FSx untuk OpenZFS](#) dapat memberikan lebih dari 1 juta IOPS dengan 20 Gbps atau throughput yang lebih baik dengan latensi beberapa ratus mikrodetik.

Ringkasan fitur dan alternatif Exadata AWS

Tabel berikut merangkum taktik dan pendekatan umum untuk menangani fitur Exadata yang hilang saat Anda memigrasikan beban kerja Exadata ke AWS. Untuk diskusi mendetail tentang setiap fitur Exadata dan alternatif AWS, lihat bagian sebelumnya.

Fitur Exadata	Taktik untuk mengatasi kesenjangan fitur	Strategi migrasi yang berlaku
Pemindaian Cerdas	Gunakan instance yang dioptimalkan memori.	Rehost, replatform, refactor
	Optimalkan konfigurasi SGA/PGA.	Rehost, replatform
	Sesuaikan parameter pengoptimal seperti optimizer <code>_index_cost_adj</code> .	Rehost, replatform
	Buat indeks skema tambahan.	Rehost, replatform, refactor
	Optimalkan SQL untuk mengurangi jejak I/O.	Rehost, replatform, refactor
Indeks penyimpanan	Buat indeks skema yang sesuai.	Rehost, replatform, refactor

Fitur Exadata	Taktik untuk mengatasi kesenjangan fitur	Strategi migrasi yang berlaku
<u>Cache Flash Cerdas</u>	Gunakan instance yang dioptimalkan memori.	Rehost, replatform, refactor
	Optimalkan SGA.	Rehost, replatform
	Konfigurasi fitur Database Flash Cache di Amazon EC2 atau Amazon RDS for Oracle dengan penyimpanan SSD lokal.	Rehost, replatform
	Gunakan solusi caching eksternal seperti Amazon ElastiCache.	Rehost, replatform, refactor
	Pertimbangkan untuk membangun arsitektur berkinerja tinggi untuk Oracle di Amazon EC2 dengan menggunakan instance dengan disk. NVMe	Rehost
	Pertimbangkan volume io2 Block Express EBS dan FSx layanan Amazon sebagai lapisan penyimpanan.	Rehost
<u>Kompresi Kolumnar Hibrida (HCC)</u>	Migrasikan arsip dan data yang jarang diakses ke solusi penyimpanan lain.	Rehost, replatform, refactor
	Gunakan kompresi lanjutan atau kompresi dasar.	Rehost, replatform

Fitur Exadata	Taktik untuk mengatasi kesenjangan fitur	Strategi migrasi yang berlaku
Manajemen Sumber Daya I/O (IORM)	Gunakan instance dan tipe penyimpanan yang sesuai untuk menghindari saturasi I/O	Rehost, replatform, refactor
	Gunakan Oracle Database Resource Manager.	Rehost, replatform
	Gunakan solusi caching eksternal seperti Amazon ElastiCache.	Rehost, replatform, refactor
	Gunakan Amazon Aurora, yang menawarkan skalabilitas I/O tinggi.	Refactor
Memori Persisten (PMEM)	Gunakan EC2 instance dengan memori tinggi.	Rehost, replatform, refactor
	Pertimbangkan volume io2 Block Express EBS dan FSx layanan Amazon sebagai lapisan penyimpanan untuk latensi rendah.	Rehost

Alat untuk fase penemuan

Bagian ini membahas AWS dan alat Oracle yang tersedia untuk fase penemuan dan tujuan masing-masing. Anda dapat menggunakan satu atau beberapa alat dari daftar ini berdasarkan persyaratan, keterampilan, dan [lisensi](#) yang diperlukan untuk alat seperti Oracle Automatic Workload Repository (AWR).

Tujuan	Alat
Tentukan fitur Exadata yang sedang Anda gunakan	Oracle Automatic Workload Repository (AWR) , Oracle Enterprise Manager (OEM) , tampilan

Tujuan	Alat
Tentukan fitur Enterprise Edition yang sedang Anda gunakan	kamus, Antarmuka Baris Perintah Kontrol Sel (CellCLI)
Menganalisis statistik database dan menunggu acara	Tampilan kamus, AWS Schema Conversion Tool (AWS SCT)
Perkirakan sumber daya dan ukuran yang tepat	AWR, OEM, tampilan kamus
	AWR, OEM, tampilan kamus, CellCLI

AWR

Oracle Automatic Workload Repository (AWR) disertakan dalam Oracle Database Enterprise Edition (EE). Secara otomatis mengumpulkan, memproses, dan memelihara statistik kinerja untuk database. Anda dapat mengakses statistik ini melalui laporan AWR, tampilan database, atau Oracle Enterprise Manager (OEM). Saat Anda mengkonsolidasikan beberapa beban kerja ke dalam satu database dengan menggunakan layanan [Oracle yang berbeda, AWR mengumpulkan statistik tingkat layanan](#) yang berguna untuk mengukur beban kerja yang dikonsolidasikan menjadi instance yang berdiri sendiri. AWS

AWR dilisensikan di bawah Oracle Diagnostics Pack (lihat informasi [lisensi](#)). Statspack, alternatif untuk AWR, adalah alat gratis untuk menganalisis statistik dan metrik kinerja. Namun, Statspack tidak memberikan tingkat metrik dan statistik yang sama terkait dengan komponen Exadata seperti AWR.

Anda dapat menghasilkan laporan AWR di tingkat instans atau secara global untuk semua instance database Real Application Cluster (RAC) atau untuk ID SQL tertentu. Untuk informasi selengkapnya, lihat panduan [tuning kinerja Oracle Database](#).

Anda dapat menggunakan AWR untuk menganalisis beban kerja Exadata Anda, fitur Exadata spesifik yang digunakan oleh beban kerja Anda, manfaat dari fitur khusus Exadata, statistik database yang berbeda dan peristiwa menunggu, dan sumber daya yang diperlukan untuk menghosting beban kerja di AWS. Statistik dan metrik kaya yang dikumpulkan oleh AWR ini mencakup beberapa lapisan sistem Exadata, termasuk server database, sel penyimpanan, jaringan interkoneksi, RAC, dan grup disk ASM. Tabel berikut merangkum metrik dan statistik AWR utama yang akan difokuskan selama

migrasi Exadata. Mencakup semua statistik dan metrik yang relevan untuk fase penemuan berada di luar cakupan panduan ini.

Metrik	Menunjukkan	Relevansi
Commit pengguna	Komit yang dikeluarkan pada batas transaksi	Sifat beban kerja
Rasio hit cache buffer	Seberapa sering blok yang diminta ditemukan di cache buffer tanpa memerlukan akses disk	Sifat beban kerja
Permintaan multi-blok baca fisik	Jumlah total permintaan baca yang dibaca dalam dua atau lebih blok database per permintaan	Sifat beban kerja, karakteristik I/O
Permintaan I/O total baca fisik	Jumlah total permintaan baca	Sifat beban kerja, karakteristik I/O
Byte I/O fisik sel memenuhi syarat untuk pembongkaran predikat	Jumlah byte pada disk yang memenuhi syarat untuk pembongkaran predikat	Ketergantungan fitur Exadata Smart Scan
Byte interkoneksi I/O fisik sel	Jumlah byte I/O yang dipertukarkan melalui interkoneksi antara host database dan sel	Ketergantungan fitur Exadata Smart Scan
Byte interkoneksi I/O fisik sel dikembalikan oleh Smart Scan	Jumlah byte I/O yang dikembalikan oleh sel untuk operasi Smart Scan	Ketergantungan fitur Exadata Smart Scan
Byte I/O fisik sel disimpan oleh indeks penyimpanan	Berapa banyak byte I/O dihilangkan dengan penerapan indeks penyimpanan pada tingkat sel penyimpanan.	Ketergantungan fitur Exadata Storage Index

Metrik	Menunjukkan	Relevansi
Permintaan baca yang dioptimalkan secara fisik	Jumlah permintaan baca yang dioptimalkan baik oleh Exadata Smart Flash Cache atau melalui indeks penyimpanan	Indeks penyimpanan Exadata dan ketergantungan fitur Smart Flash Cache
Cell Flash Cache baca klik	Jumlah permintaan baca yang menemukan kecocokan di Exadata Smart Flash Cache	Ketergantungan fitur Exadata Smart Flash Cache

CellCLI

Cell Control Command-Line Interface (CellCLI) adalah alat administrasi dan pemantauan baris perintah untuk sel penyimpanan Exadata yang telah dikonfigurasi sebelumnya di server sel penyimpanan Exadata. Utilitas ini mengekstrak informasi langsung dari perangkat keras atau perangkat lunak server penyimpanan.

Untuk daftar lengkap metrik yang tersedia untuk CellCLI, lihat dokumentasi [Oracle](#) Exadata. Untuk melihat daftar semua metrik yang tersedia dan definisinya, jalankan perintah berikut saat terhubung ke CellCLI dari salah satu server penyimpanan.

```
CellCLI>LIST metricDefinition WHERE objectType=cell;
```

Untuk menganalisis metrik yang berbeda, sambungkan langsung ke server penyimpanan dan gunakan CellCLI `list metriccurrent` atau `list metrichistory` perintah untuk membacanya.

```
CellCLI> list metriccurrent
```

```

      CD_BY_FC_DIRTY                                CD_00_celladm-01
0.000 MB
...
...
      SIO_IO_WR_RQ_FC_SEC                            SMARTIO
      0.000 IO/sec
      SIO_IO_WR_RQ_HD                                SMARTIO
      3,660,097 IO requests
```

SIO_IO_WR_RQ_HD_SEC
0.000 IO/sec

SMARTIO

Anda harus menjalankan CellCli pada node sel individu untuk mengumpulkan metrik untuk node itu. Anda juga dapat menjalankan perintah CellCLI `dccli` untuk mengumpulkan metrik untuk sekelompok node sel.

```
./dccli -g mycells "cellcli -e list metriccurrent GD_IO_BY_R_LG \
attributes alertstate, metricvalue";
```

Exadata membongkar banyak tugas intensif sumber daya ke server sel penyimpanan. Oleh karena itu, penting untuk memahami bagaimana berbagai sumber daya digunakan pada sel penyimpanan untuk mengukur instance komputasi dengan benar di lingkungan target. Tabel berikut menunjukkan beberapa metrik Exadata kunci dari server sel penyimpanan yang dapat membantu Anda memahami bagaimana sumber daya digunakan dalam sel penyimpanan.

Metrik	Deskripsi
CL_CPU	Pemanfaatan CPU sel
CL_MEMUT	Persentase total memori fisik yang digunakan
N_HCA_MB_RCV_SEC	Jumlah megabita yang diterima oleh InfiniBand antarmuka per detik
N_HCA_MB_TRANS_SEC	Jumlah megabita yang ditransmisikan oleh InfiniBand antarmuka per detik
N_MB_RECEIVED_SEC	Tingkat (jumlah megabita) yang diterima per detik dari host tertentu
N_MB_SENT_SEC	Tingkat (jumlah megabita) yang dikirim per detik dari host tertentu
FL_RQ_TM_W_RQ	Latensi permintaan tulis redo log rata-rata
FL_IO_TM_W_RQ	Latensi tulis redo log rata-rata, yang hanya mencakup latensi I/O tulis

Metrik	Deskripsi
FC_IO_RQ_W_SKIP_SEC	Jumlah permintaan I/O tulis per detik yang melewati Flash Cache
FC_IO_RQ_R_SKIP_SEC	Jumlah permintaan I/O baca per detik yang melewati Flash Cache
SIO_IO_EL_OF_SEC	Jumlah megabita per detik yang memenuhi syarat untuk dibongkar oleh smart I/O
SIO_IO_OF_RE_SEC	Jumlah megabita interkoneksi per detik yang dikembalikan oleh smart I/O
SIO_IO_RD_FC_SEC	Jumlah megabyte per detik yang dibaca dari Flash Cache oleh smart I/O
SIO_IO_RD_HD_SEC	Jumlah megabyte per detik yang dibaca dari hard disk oleh smart I/O
SIO_IO_WR_FC_SEC	Jumlah megabyte per detik operasi penulisan populasi Flash Cache oleh smart I/O
SIO_IO_SI_SV_SEC	Jumlah megabyte per detik yang disimpan oleh indeks penyimpanan

Perintah CellCLI berikut berjalan terhadap simpul sel Exadata untuk menunjukkan statistik yang terkait dengan fitur Exadata utama.

```
CellCLI> list metristory where collectionTime > '2022-06-13T15:42:00+01:00' and
collectionTime < '2022-06-13T15:43:00+01:00' and name like 'SIO_.*SEC.*'
```

```

          SIO_IO_EL_OF_SEC          SMARTIO          1,223 MB/sec
2022-06-13T15:42:03+01:00

          SIO_IO_OF_RE_SEC          SMARTIO          34.688 MB/sec
2022-06-13T15:42:03+01:00

          SIO_IO_PA_TH_SEC          SMARTIO          0.000 MB/sec
2022-06-13T15:42:03+01:00
```

SIO_IO_RD_FC_HD_SEC 2022-06-13T15:42:03+01:00	SMARTIO	0.174 MB/sec
SIO_IO_RD_FC_SEC 2022-06-13T15:42:03+01:00	SMARTIO	843 MB/sec
SIO_IO_RD_HD_SEC 2022-06-13T15:42:03+01:00	SMARTIO	0.101 MB/sec
SIO_IO_RD_RQ_FC_HD_SEC 2022-06-13T15:42:03+01:00	SMARTIO	0.183 IO/sec
SIO_IO_RD_RQ_FC_SEC 2022-06-13T15:42:03+01:00	SMARTIO	850 IO/sec
SIO_IO_RD_RQ_HD_SEC 2022-06-13T15:42:03+01:00	SMARTIO	0.000 IO/sec
SIO_IO_RV_OF_SEC 2022-06-13T15:42:03+01:00	SMARTIO	3.392 MB/sec
SIO_IO_SI_SV_SEC 2022-06-13T15:42:03+01:00	SMARTIO	362 MB/sec
SIO_IO_WR_FC_SEC 2022-06-13T15:42:03+01:00	SMARTIO	0.008 MB/sec
SIO_IO_WR_HD_SEC 2022-06-13T15:42:03+01:00	SMARTIO	0.000 MB/sec
SIO_IO_WR_RQ_FC_SEC 2022-06-13T15:42:03+01:00	SMARTIO	0.017 IO/sec
SIO_IO_WR_RQ_HD_SEC 2022-06-13T15:42:03+01:00	SMARTIO	0.000 IO/sec

Dalam contoh statistik ini, SIO_IO_SI_SV_SEC menunjukkan bahwa 362 MBps I/O disimpan oleh indeks penyimpanan, SIO_IO_RD_RQ_FC_SEC menunjukkan bahwa 850 I/O per detik dilayani oleh Flash Cache, dan SIO_IO_OF_RE_SEC menunjukkan bahwa 34 I/O dikembalikan oleh MBps Smart Scan.

Dalam contoh lain, output `dcli` perintah berikut menunjukkan pemanfaatan CPU yang sangat rendah di semua node sel dalam sistem Exadata. Ini berpotensi menunjukkan beban kerja yang tidak mendapat manfaat signifikan dari fitur lapisan penyimpanan Exadata.

```
dcli -g
../cell_group cellcli -e \
list metriccurrent where name='CL_CPUT';
cm01cel01: CL_CPUT cm01cel01 0.2 %
cm01cel02: CL_CPUT cm01cel02 0.2 %
cm01cel03: CL_CPUT cm01cel03 0.7 %
```

Kontrol Awan OEM

Oracle Enterprise Manager (OEM) Cloud Control menyediakan kemampuan terpusat, komprehensif, end-to-end pemantauan, manajemen, administrasi, dan dukungan untuk semua sistem Oracle utama. Cara terbaik untuk memantau dan mengelola Exadata adalah dengan menggunakan OEM, karena terintegrasi erat dengan semua komponen perangkat lunak dan perangkat keras Exadata.

Anda dapat mengakses banyak metrik yang telah dibahas sejauh ini dengan menggunakan dasbor OEM. Beberapa dasbor utama yang membantu dalam fase penemuan migrasi Exadata adalah:

- Pemanfaatan sumber daya pada server database
- Statistik penyimpanan dan I/O dari sel penyimpanan
- InfiniBand beralih statistik
- Statistik grup disk ASM
- Kinerja database menggunakan AWR, Automatic Database Diagnostic Monitor (ADDM), dan Active Session History (ASH)
- Alat penasihat seperti SGA Advisory dan SQL Tuning Advisor

Namun, beberapa dasbor dilisensikan di bawah paket yang berbeda seperti Oracle Diagnostics Pack atau Oracle Tuning Pack. Untuk detailnya, lihat informasi [perizinan Oracle](#).

Tampilan basis data

Anda dapat menanyakan tampilan database (tampilan kamus dan tampilan kinerja dinamis) dalam database Oracle untuk mengambil statistik berguna yang terkait dengan fitur Exadata untuk database atau instance Anda. Tabel berikut menunjukkan beberapa pandangan utama yang menampilkan statistik kritis yang berguna untuk fase penemuan.

Lihat	Deskripsi
DBA_TABLES	Mengidentifikasi tabel yang menggunakan fitur HCC
DBA_HIST_SYSSTAT	Menampilkan statistik historis terkait Exadata
DBA_FEATURE_USAGE_STATISTICS	Menampilkan informasi tentang penggunaan fitur database
DBA_HIST_SQLSTAT	Menampilkan informasi historis tentang statistik SQL
DBA_HIST_ASM_DISKGROUP_STAT	Menampilkan statistik kinerja untuk grup disk ASM
DBA_HIST_CELL_DISK_SUMMARY	Menampilkan informasi historis tentang kinerja disk pada sel
DBA_HIST_ACTIVE_SESS_HISTORY	Menampilkan riwayat sesi aktif
DBA_HIST_DB_CACHE_ADVICE	Memberikan prediksi jumlah operasi baca fisik untuk ukuran cache
DBA_ADVISOR_FINDINGS	Menampilkan temuan berbagai tugas penasehat seperti SQL Tuning Advisor

Contoh berikut menunjukkan statistik yang diambil dari tampilan database yang berguna untuk fase penemuan.

Kueri ini menunjukkan satu tabel dalam database yang diaktifkan untuk HCC dengan mode QUERY HIGH kompresi:

```
select table_name, compression, compress_for from dba_tables where compression =
'ENABLED';
TABLE_NAME COMPRESS COMPRESS_FOR
-----
ORDER_ITEMS ENABLED QUERY HIGH
```

Kueri ini menampilkan penggunaan fitur database, yang membantu menentukan ketergantungan fitur pada Oracle Database Enterprise Edition:

```
select
  name          c1,
  detected_usages c2,
  first_usage_date c3,
  currently_used  c4
from dba_feature_usage_statistics
where first_usage_date is not null;
```

feature	times used	first used	used now
Protection Mode - Maximum Performance	24	18-AUG-20	TRUE
Recovery Area	24	18-AUG-20	TRUE
Server Parameter File	24	18-AUG-20	TRUE
Shared Server	4	18-AUG-20	FALSE
Streams (system)	24	18-AUG-20	TRUE
Virtual Private Database (VPD)	24	18-AUG-20	TRUE
Automatic Segment Space Management (system)	24	18-AUG-20	TRUE
Automatic Segment Space Management (user)	24	18-AUG-20	TRUE
Automatic SQL Execution Memory	24	18-AUG-20	TRUE
Automatic Undo Management	24	18-AUG-20	TRUE
Character Set	24	18-AUG-20	TRUE
Dynamic SGA	1	18-AUG-20	FALSE
Locally Managed Tablespaces (system)	24	18-AUG-20	TRUE
Locally Managed Tablespaces (user)	24	18-AUG-20	TRUE
Multiple Block Sizes	7	25-DEC-20	TRUE
Partitioning (system)	24	18-AUG-20	TRUE

Kueri ini menunjukkan total byte baca fisik, byte yang memenuhi syarat untuk pembongkaran sel, dan byte yang dikembalikan dari sel penyimpanan untuk pernyataan SQL untuk snapshot AWR tertentu:

```
select
  ROUND(physical_read_bytes_delta/EXECUTIONS_DELTA)/1024/1024 phyrd_mb
  , ROUND(IO_OFFLOAD_ELIG_BYTES_TOTAL/EXECUTIONS_DELTA)/1024/1024 elig_mb
  , ROUND(io_interconnect_bytes_delta/EXECUTIONS_DELTA)/1024/1024 ret_mb
from dba_hist_sqlstat
where sql_id = 'zg2fg7abfx2y' and snap_id between 12049 and 12050;
```

PHYRD_MB	ELIG_MB	RET_MB	SAVING%
10815	10815	3328	69.2%

AWS SCT

The [AWS Schema Conversion Tool \(AWS SCT\)](#) membuat migrasi database heterogen dapat diprediksi. Ini secara otomatis mengubah skema database sumber dan sebagian besar objek kode database, termasuk tampilan, prosedur tersimpan, dan fungsi, ke format yang kompatibel dengan database target. Objek apa pun yang tidak dapat dikonversi secara otomatis ditandai dengan jelas sehingga Anda dapat mengonversinya secara manual untuk menyelesaikan migrasi. AWS SCT dapat memprediksi upaya yang diperlukan untuk migrasi heterogen ketika tindakan manual diperlukan untuk mengonversi objek database. Alat ini juga dapat menunjukkan ketergantungan pada fitur Oracle Database Enterprise Edition (EE). Anda dapat menggunakan analisis ini untuk memutuskan apakah akan mempertimbangkan untuk bermigrasi dari EE ke SE2. Untuk informasi selengkapnya, lihat bagian [Edisi dan versi Database](#) sebelumnya dalam panduan ini. Untuk informasi tentang penggunaan AWS SCT untuk migrasi heterogen, lihat bagian [Melakukan migrasi](#) nanti dalam panduan ini.

Persyaratan sumber daya untuk platform target

Kami menyarankan Anda mengukur lingkungan basis data target AWS berdasarkan pemanfaatan sumber Exadata, bukan konfigurasi. Banyak pelanggan membeli sistem Exadata dengan kapasitas tambahan untuk mengakomodasi pertumbuhan yang diantisipasi selama tiga hingga lima tahun ke depan. Biasanya, ketika beban kerja Exadata dimigrasikan ke AWS, lebih sedikit sumber daya yang digunakan dibandingkan dengan konfigurasi sistem Exadata sumber, jadi menyesatkan untuk menggunakan konfigurasi asli tersebut untuk memprediksi sumber daya AWS.

Untuk memperkirakan sumber daya yang diperlukan dalam instance target, Anda dapat menggunakan alat yang dibahas di [bagian sebelumnya](#), seperti AWR, tampilan basis data, OEM, dan CellCLI. Pada AWS, Anda dapat meningkatkan atau menurunkan sumber daya dengan lebih mudah dibandingkan dengan platform Exadata sumber. Bagian berikut membahas praktik terbaik untuk memperkirakan sumber daya seperti CPU, memori, dan IOPS untuk platform target. Selain itu, tim akun AWS dan spesialis database yang memiliki pengalaman luas dalam membantu pelanggan dengan migrasi Exadata mereka dapat membantu Anda mengukur lingkungan target Anda. AWS memiliki alat internal yang dapat digunakan oleh tim akun AWS untuk memperkirakan sumber daya yang diperlukan dan mengukur lingkungan target Anda dengan tepat di AWS.

Persyaratan CPU dan memori

Saat memigrasikan beban kerja Exadata ke opsi penyebaran database Oracle, seperti Amazon RDS for Oracle AWS, Anda tidak boleh mendasarkan sumber daya lapisan komputasi (CPU dan

memori) hanya pada statistik pemanfaatan dari server database Exadata. Beban kerja juga mendapat manfaat dari fitur khusus Exadata seperti Smart Scan dan indeks penyimpanan, yang menurunkan pemrosesan ke sel penyimpanan dan menggunakan sumber daya server penyimpanan. Oleh karena itu, Anda harus menyediakan lapisan komputasi dalam instance target dengan CPU dan sumber daya memori tambahan berdasarkan penggunaan fitur khusus Exadata dan tingkat pengoptimalan beban kerja yang dimungkinkan selama migrasi.

Sulit untuk memperkirakan secara akurat sumber daya CPU tambahan yang diperlukan. Gunakan hasil penemuan sebagai titik awal untuk mengukur lingkungan target. Untuk perhitungan kasar, pertimbangkan untuk menyertakan satu vCPU tambahan untuk setiap 500 MBps beban kerja Smart Scan ke total v yang CPUs diperlukan untuk lapisan komputasi.

Pendekatan lain adalah dengan mempertimbangkan pemanfaatan CPU pada server penyimpanan. Anda dapat menambahkan sekitar 20 persen dari total yang digunakan CPUs pada server penyimpanan ke total v CPUs yang diperlukan untuk lapisan komputasi sebagai titik awal. Anda dapat menyesuaikan persentase ini berdasarkan penggunaan fitur Exadata, sebagaimana ditentukan oleh alat seperti AWR dan CellCLI. Misalnya, untuk penggunaan rendah, Anda dapat menambahkan 10 persen untuk penggunaan rendah, 20 persen untuk penggunaan sedang, dan 40 persen untuk penggunaan tinggi. Jika Anda menggunakan total 20 utas CPU di semua server penyimpanan dan penggunaan fitur Exadata diamati sebagai media, Anda dapat mempertimbangkan 4 v tambahan CPUs untuk mengimbangi fitur Exadata yang hilang di lingkungan target.

Setelah perkiraan awal ini, Anda juga harus melakukan pengujian kinerja pada lingkungan target untuk menentukan apakah Anda perlu menskalakan sumber daya yang dialokasikan. Pengujian kinerja juga dapat mengungkapkan peluang optimasi beban kerja lebih lanjut yang dapat mengurangi sumber daya yang dibutuhkan.

Anda mungkin harus meningkatkan alokasi memori ke instance Oracle untuk meningkatkan rasio hit cache dan mengurangi jejak I/O. Platform Exadata sumber Anda mungkin tidak memiliki memori yang cukup untuk alokasi SGA yang besar, terutama dalam skenario konsolidasi. Ini mungkin tidak menyebabkan masalah kinerja di Exadata, karena operasi I/O umumnya cepat. Kami menyarankan Anda memulai dengan instance yang mendukung alokasi memori berikut:

```
Target memory required = larger of (8 GB per vCPUs required, two times the SGA+PGA allocation in the source)
```

```
SGA+PGA allocation = ~80% of available memory on the instance
```

Selama pengujian kinerja, Anda dapat menggunakan fitur Oracle seperti buffer pool advisory, SGA target advisory, dan PGA memory advisory untuk menyempurnakan alokasi SGA dan PGA untuk memenuhi persyaratan beban kerja Anda.

Instans Amazon EC2 atau Amazon RDS harus memiliki CPU, memori, dan sumber daya I/O yang memadai untuk menangani beban kerja database yang diantisipasi. Kami menyarankan Anda menggunakan kelas instance generasi saat ini untuk meng-host beban kerja Anda. AWS Jenis instance generasi saat ini, seperti instance yang dibangun di atas [Sistem Nitro](#), mendukung perangkat keras mesin virtual (V)HVMs. Untuk memanfaatkan jaringan yang ditingkatkan dan peningkatan keamanan, Anda dapat menggunakan Amazon Machine Images (AMIs) untuk HVMs. Amazon RDS for Oracle saat ini mendukung hingga 128 GBs vCPU dan 3.904 memori. Lihat [kelas instans DB](#) dalam dokumentasi Amazon RDS untuk informasi tentang spesifikasi perangkat keras kelas instans yang tersedia untuk Amazon RDS for Oracle. Lihat jenis [instans EC2 Amazon untuk daftar instance](#) dengan detail sumber EC2 daya.

Persyaratan I/O

Menggunakan laporan AWR untuk memperkirakan kebutuhan sumber daya membutuhkan keakraban dengan pola beban kerja untuk timing beban puncak, off-peak, dan rata-rata. Untuk memperkirakan persyaratan IOPS untuk beban kerja Anda berdasarkan laporan AWR yang dikumpulkan selama periode puncak, ikuti langkah-langkah berikut:

1. Tinjau laporan AWR untuk mengidentifikasi permintaan I/O baca fisik, permintaan I/O tulis fisik, total byte baca fisik, dan total byte penulisan fisik.

Metrik ini memperhitungkan manfaat fitur khusus Exadata seperti indeks penyimpanan, sehingga menunjukkan IOPS aktual dan nilai throughput yang dapat Anda gunakan untuk mengukur lapisan I/O penyimpanan lingkungan AWS target Anda.

2. Di bagian profil I/O laporan AWR, tinjau permintaan baca fisik yang dioptimalkan dan nilai yang dioptimalkan permintaan tulis fisik untuk menentukan apakah Smart Scan dan fitur Exadata lainnya yang terkait dengan I/O — seperti I/O yang disimpan oleh indeks penyimpanan, cache kolumnar, atau Smart Flash Cache — digunakan. Jika Anda melihat permintaan yang dioptimalkan di profil AWR I/O, tinjau statistik sistem untuk mendapatkan detail metrik Smart Scan dan indeks penyimpanan seperti byte I/O fisik sel yang memenuhi syarat untuk pembongkaran predikat, byte interkoneksi I/O fisik sel yang dikembalikan oleh Smart Scan, dan byte I/O fisik sel yang disimpan oleh indeks penyimpanan.

Meskipun metrik ini tidak langsung digunakan untuk mengukur lingkungan target, metrik ini berguna untuk memahami berapa banyak I/O disimpan oleh fitur khusus Exadata dan teknik penyetelan untuk mengoptimalkan beban kerja.

```
Total IOPS required for the workload = physical read IO requests + physical write IO requests
```

```
Total throughput = physical read bytes + physical write bytes
```

Statistik AWR permintaan I/O baca fisik, permintaan I/O tulis fisik, byte baca fisik, dan byte tulis fisik mencerminkan aktivitas I/O beban kerja, tidak termasuk I/O yang disumbangkan oleh komponen non-aplikasi seperti cadangan RMAN dan utilitas lainnya seperti expdp atau sqlldr. Dalam kasus tersebut, Anda dapat mempertimbangkan permintaan I/O total pembacaan fisik statistik AWR, permintaan I/O total penulisan fisik, total byte baca fisik, dan total byte penulisan fisik untuk memperkirakan dan persyaratan throughput. IOPs

Database yang berjalan di Exadata biasanya menghasilkan ratusan ribu IOPS dan throughput yang sangat tinggi (lebih dari 50 Gbps) karena faktor-faktor yang dibahas di bagian sebelumnya. Namun, dalam banyak kasus, teknik penyetelan dan pengoptimalan beban kerja mengurangi jejak I/O dari beban kerja secara drastis.

Jika persyaratan I/O sangat tinggi, waspadai keterbatasan Amazon EC2 dan Amazon RDS. Untuk throughput volume Amazon EBS yang tinggi, pertimbangkan untuk menggunakan kelas EC2 instans Amazon seperti x2iedn, x2idn, dan r5b, yang mendukung hingga 260.000 IOPS dengan throughput 10.000. MBps Lihat [instans Amazon EBS yang dioptimalkan](#) dalam EC2 dokumentasi Amazon untuk meninjau IOPS maksimum dan throughput yang didukung oleh berbagai instans. Untuk Amazon RDS for Oracle, kelas instans rb5 mendukung hingga 256.000 IOPS dengan throughput 4.000. MBps Lihat [kelas instans DB](#) untuk meninjau instans Amazon EBS yang dioptimalkan yang tersedia untuk Amazon RDS for Oracle.

Anda juga harus memahami bagaimana IOPS dan throughput diukur dalam kasus volume EBS berbeda yang tersedia untuk lingkungan target. Dalam beberapa kasus, Amazon EBS membagi atau menggabungkan operasi I/O untuk memaksimalkan throughput. Untuk mempelajari lebih lanjut, lihat [karakteristik dan pemantauan I/O](#) dalam EC2 dokumentasi Amazon dan [Bagaimana cara mengoptimalkan kinerja volume IOPS yang Disediakan Amazon EBS saya?](#) di pusat AWS pengetahuan.

Pengujian kinerja pada platform target

Anda dapat memilih instans target dan opsi penyimpanan yang sesuai AWS berdasarkan informasi sumber daya yang Anda kumpulkan selama fase penemuan.

Setelah instans target disediakan, sebaiknya Anda melakukan pengujian beban untuk memastikan bahwa instans dan konfigurasi yang disediakan memenuhi persyaratan kinerja aplikasi Anda. Anda harus melakukan pengujian beban ini dengan menggunakan beban kerja aplikasi nyata Anda untuk jumlah pengguna dan konkurensi yang diantisipasi alih-alih menggunakan alat pengujian beban generik seperti Swingbench. Jika target Anda adalah Amazon RDS for Oracle, Amazon RDS Custom for Oracle, EC2 atau Amazon, [Anda dapat menggunakan Oracle](#) Real Application Testing, yang merupakan fitur berlisensi terpisah, untuk menangkap beban kerja produksi dari database Exadata sumber dan memutarkannya kembali pada instance target untuk menilai kinerja. Untuk informasi lebih lanjut tentang menggunakan Pengujian Aplikasi Nyata AWS, lihat posting AWS blog [Menggunakan fitur Pengujian Aplikasi Nyata Oracle dengan Amazon RDS for Oracle dan Gunakan fitur Pengujian Aplikasi Nyata Oracle](#) dengan Amazon. EC2

Jika Anda merencanakan migrasi heterogen, di mana beban kerja dimigrasikan dari Oracle Database ke database open source seperti PostgreSQL, lebih sulit untuk memperkirakan sumber daya karena mereka tidak sebanding di mesin yang berbeda. Sebagai praktik umum, kami menyarankan Anda memulai dengan instance yang dapat mendukung sumber daya CPU, memori, dan I/O yang setara dengan sumber daya yang digunakan di Exadata, dan kemudian mengukur instans target dengan tepat berdasarkan hasil pengujian beban dengan menggunakan opsi penskalaan. AWS

Persyaratan aplikasi SLA

Selama fase penemuan, penting untuk menentukan persyaratan SLA aplikasi Anda yang di-host di Exadata, termasuk tujuan waktu pemulihan (RTO) dan tujuan titik pemulihan (RPO). Anda harus memahami persyaratan ini dari sudut pandang bisnis atau pengguna alih-alih menyalin arsitektur Anda saat ini seperti ke platform target. Misalnya, penyebaran Anda saat ini mungkin menggunakan fitur Oracle Real Application Cluster (RAC), yang terintegrasi dengan Exadata. Namun, jika aplikasi Anda tidak benar-benar membutuhkan fitur ini, mungkin layak untuk menerapkan solusi hemat biaya tanpa menggunakan RAC. AWS

Tabel berikut mencantumkan RTO dan RPO yang dapat Anda capai dengan model penerapan yang berbeda. AWS Informasi ini didasarkan pada ketersediaan tinggi dan opsi pemulihan bencana (HA/DR) dalam satu. Wilayah AWS Anda dapat memperluas kemampuan DR dengan menggunakan

model penyebaran Multi-wilayah, seperti menambahkan replika baca lintas wilayah di Amazon RDS for Oracle, atau menggunakan database global di Amazon Aurora.

Jenis penyebaran	RTO (dalam hitungan detik)	RPO (dalam hitungan detik)	Komentar
Amazon RDS for Oracle dengan Multi-AZ	~ 120	0	RTO dapat bervariasi tergantung pada faktor-faktor seperti waktu yang diperlukan untuk pemulihan misalnya.
Amazon RDS Custom for Oracle dengan solusi HA yang dikelola sendiri menggunakan Data Guard dan Fast Start Failover (FSFO)	~ 120	0	Membangun solusi HA yang tepat adalah tanggung jawab Anda. Sebagai praktik terbaik, terapkan instance siaga di Availability Zone yang berbeda dari instance utama.
Instans yang dikelola sendiri di Amazon EC2 dengan menggunakan Data Guard dan FSFO	~ 120	0	Membangun solusi HA yang tepat adalah tanggung jawab Anda. Sebagai praktik terbaik, terapkan instance siaga di Availability Zone yang berbeda dari instance utama.
yang kompatibel dengan Aurora PostgreSQL	< 30	0	Jika Anda menggunakan instance pembaca, failover dapat selesai dalam beberapa detik.

Jenis penyebaran	RTO (dalam hitungan detik)	RPO (dalam hitungan detik)	Komentar
Amazon RDS untuk PostgreSQL dengan Multi-AZ	~ 120	0	
RAC aktif AWS dengan Oracle Active Data Guard	0	0	Jenis penerapan ini menggunakan salah satu opsi penerapan RAC AWS dengan replikasi Data Guard ke Availability Zone lainnya.

Seperti halnya model penerapan, memilih strategi migrasi dan rollback yang tepat serta alat migrasi sangat penting untuk memenuhi persyaratan SLA bisnis Anda. Topik ini dibahas secara rinci di [bagian Melakukan migrasi](#) dari panduan ini.

Kebijakan manajemen dan retensi siklus hidup data

Organizations biasanya menyimpan data untuk waktu yang lama untuk memenuhi persyaratan kepatuhan mereka. Merupakan praktik umum untuk melihat seluruh kumpulan data untuk aplikasi, termasuk data aktif, data yang jarang diakses, dan data yang diarsipkan, disimpan dalam satu database yang dihosting di Exadata menggunakan fitur seperti HCC. Mungkin tidak efisien untuk mengikuti praktik yang sama saat Anda memigrasikan beban kerja Exadata ke AWS. AWS menyediakan beberapa solusi penyimpanan, seperti [Amazon S3 Intelligent-Tiering](#) dan [Amazon S3 Glacier](#) untuk menyimpan, menanyakan, dan mengambil data yang jarang diakses dan diarsipkan secara efisien alih-alih menyimpannya dalam database transaksional. Untuk informasi selengkapnya tentang pendekatan berbeda untuk menangani data yang diarsipkan dan jarang diakses selama migrasi ke AWS, lihat bagian [Melakukan migrasi pada](#) panduan ini.

Faktor lainnya

Memahami alat dan produk yang tersedia untuk digunakan berdasarkan perjanjian lisensi Oracle saat ini akan sangat membantu dalam memilih strategi migrasi yang tepat. AWS Misalnya, jika

Anda memiliki lisensi dan keterampilan untuk menggunakan Oracle GoldenGate, itu bisa menjadi alternatif untuk digunakan AWS DMS sebagai alat migrasi. Untuk informasi selengkapnya, lihat [bagian Melakukan migrasi](#) pada panduan ini.

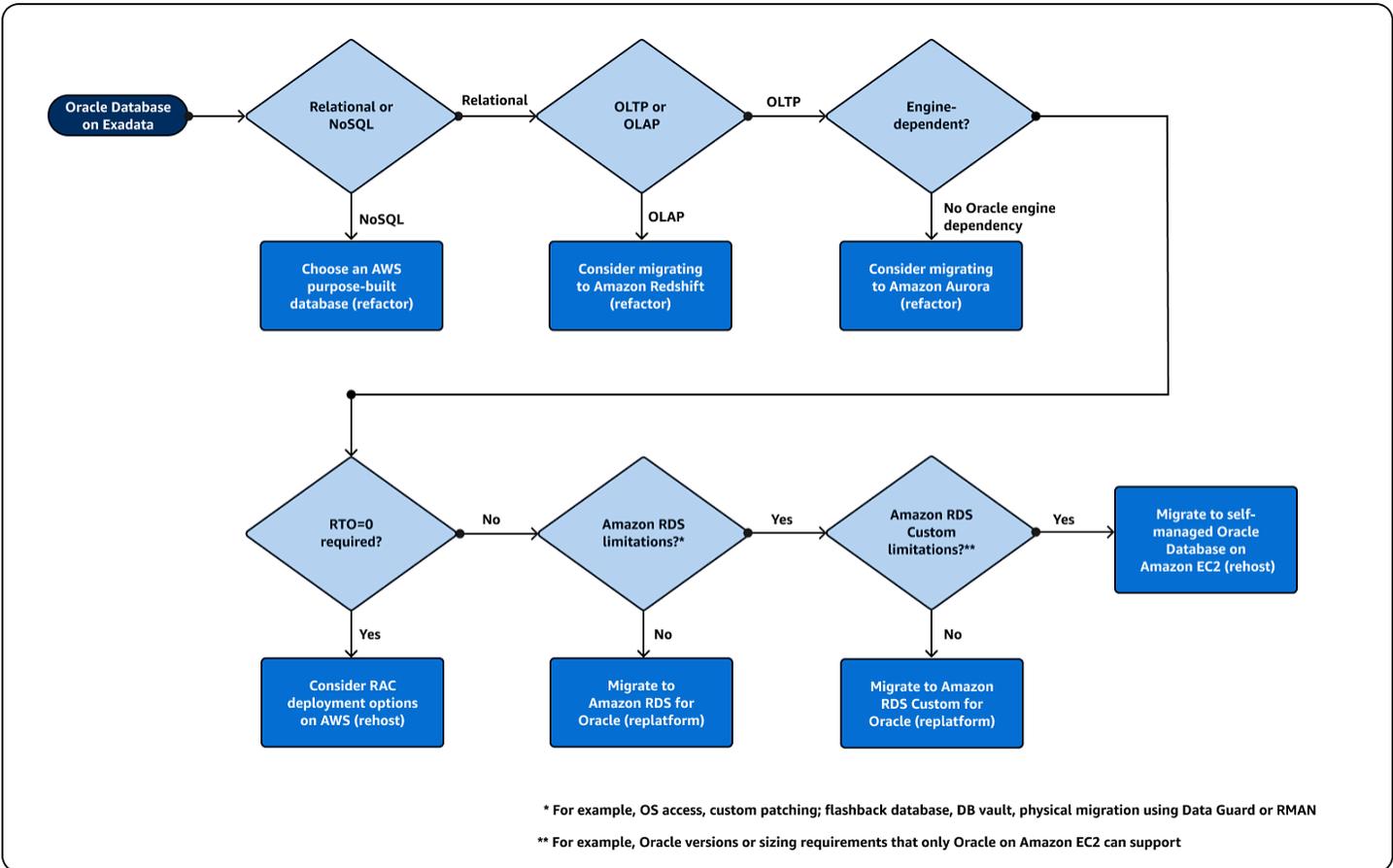
Selain itu, kami menyarankan Anda mengumpulkan rincian semua antarmuka masuk dan keluar ke database Anda di Exadata. Ini mencakup semua komponen aplikasi yang terhubung ke database, konektivitas interdatabase menggunakan tautan basis data, dan konektivitas database asing. Anda juga harus menyertakan antarmuka ini dalam pengujian fungsional dan beban Anda pada instance target, jika antarmuka yang ada memerlukan perubahan untuk bekerja dalam arsitektur target. Misalnya, Amazon RDS for Oracle tidak mendukung [konektivitas database asing menggunakan produk Oracle Database Gateway](#), jadi Anda mungkin perlu merancang ulang antarmuka untuk [menggunakan solusi lain seperti atau memigrasikan database AWS Gluetersebut ke Amazon RDS Custom for Oracle](#) atau database Oracle yang dikelola sendiri di Amazon. EC2

Kami juga menyarankan Anda mempertimbangkan fitur database yang digunakan beban kerja Anda untuk memilih lingkungan target yang tepat. AWS Jika beban kerja Anda memiliki ketergantungan pada fitur add-on Oracle Database, itu harus dimigrasikan ke AWS layanan yang mendukung fitur tersebut. Misalnya, jika beban kerja Anda memiliki ketergantungan pada Oracle Database Vault, Anda harus menghostingnya di Amazon RDS Custom for Oracle atau sebagai database yang dikelola sendiri di Amazon. EC2

Untuk informasi tentang fitur yang didukung, lihat dokumentasi [Amazon RDS for Oracle dan Amazon RDS Custom for Oracle](#).

Diagram alur keputusan

Diagram berikut menyediakan diagram alur keputusan sederhana untuk memigrasikan aplikasi yang berjalan di Exadata ke AWS. Diagram alur menawarkan panduan arah tentang opsi basis data target, tetapi tidak ada dua proyek migrasi yang sama. Kami menyarankan Anda melakukan penilaian penemuan atau bukti konsep (POC) sebelum membuat keputusan apa pun.



Melakukan migrasi

Database Oracle Anda AWS harus dimigrasikan, dikonfigurasi, dan disetel dengan benar untuk menangani beban kerja yang diantisipasi. Sebelum menyetel, pengujian AWS kinerja pertama mungkin tidak memenuhi kinerja lingkungan Exadata lokal. Namun, dengan ukuran dan penyetelan yang tepat, Anda dapat memenuhi atau melampaui kinerja Exadata asli dalam banyak kasus. AWS Untuk mencapai tujuan kinerja Anda, kami sarankan Anda mendapatkan bantuan Layanan AWS Profesional atau AWS Mitra berpengalaman untuk migrasi Anda.

Bagian ini membahas pendekatan migrasi, termasuk replatforming, rehosting, dan refactoring. Ini memberikan gambaran umum tentang layanan dan alat migrasi, dan mencakup praktik terbaik untuk menyiapkan database Oracle yang dioptimalkan dan berkinerja. AWS

Amazon RDS for Oracle dan Amazon EC2 mendukung Oracle Database Enterprise Edition (EE) dan Oracle Database Standard Edition 2 (SE2). Oracle Database EE menyediakan kinerja, ketersediaan, skalabilitas, dan keamanan untuk mengembangkan aplikasi seperti aplikasi pemrosesan transaksi online volume tinggi (OLTP), gudang data intensif kueri, dan aplikasi internet yang menuntut. Ini menyediakan semua komponen Oracle Database dan peningkatan tambahan dengan pembelian opsi dan paket. Sistem Exadata lokal menggunakan Oracle Database EE.

Note

Meskipun AWS mendukung kedua edisi Oracle, bagian ini mengasumsikan bahwa Oracle Database EE akan digunakan untuk database Oracle di AWS

Untuk informasi tambahan tentang strategi migrasi dan arsitektur, lihat Panduan AWS Preskriptif Panduan [Migrasi database Oracle](#) ke AWS Cloud

Di bagian ini:

- [Exadata ke AWS alat migrasi](#)
- [AWS pola migrasi sampel](#)
- [Pertimbangan fitur khusus Exadata](#)
- [Pertimbangan migrasi database homogen](#)
- [Rekomendasi replatforming](#)
- [Rekomendasi rehosting](#)

- [Rekomendasi refactoring](#)

Exadata ke AWS alat migrasi

Ada lebih dari 15 Exadata untuk pendekatan AWS migrasi. Tabel berikut menunjukkan alat yang paling umum digunakan. Tabel tidak termasuk ekspor/impor konvensional Oracle, Oracle SQL* Loader, Salinan Database Pengembang Oracle SQL, Oracle SQL* Wisaya Ekspor/Impor Pengembang, Oracle Transportable Tablespaces, tautan database Oracle menggunakan Create Table as Select (CTAS), tabel eksternal Oracle, atau solusi ekstrak, transformasi, dan muat (ETL).

Pendekatan migrasi	Mendukung strategi migrasi	Fisik atau logis	Mendukung pengambilan data perubahan (CDC)	Membutuhkan jaringan untuk AWS
AWS DMS	Semua	Logis	Ya	Ya
Oracle GoldenGate	Semua	Logis	Ya	Ya
Pompa Data Oracle	Rehost, replatform	Logis	Tidak	Tidak
Manajer Pemulihan Oracle (RMAN)	Rehost	Fisik	Tidak	Jika Anda menggunakan RMAN DUPLICATE atau Oracle Secure Backup ke Amazon S3
Penjaga Data Oracle	Rehost	Fisik	Ya	Ya

Oracle Data Guard dan Oracle Recovery Manager (RMAN) adalah opsi yang sangat baik untuk memigrasikan database Exadata ke Amazon EC2. Namun, Amazon RDS for Oracle tidak mendukung salah satu alat ini.

Anda dapat menerapkan Oracle Data Guard dengan menggunakan metode siaga logis atau siaga fisik. Database siaga logis menerapkan pernyataan bahasa manipulasi data (DHTML) pada database siaga untuk menjaga data tetap disinkronkan. Database siaga logis biasanya digunakan untuk membongkar pelaporan dari database utama. Semua referensi Oracle Data Guard di bagian ini berlaku langsung ke siaga fisik. Database siaga fisik cocok dengan database utama persis di tingkat blok.

AWS DMS migrasi

AWS Database Migration Service (AWS DMS) adalah solusi replikasi logis. Ini mendukung migrasi homogen seperti memigrasikan database lokal Oracle ke database Oracle, serta migrasi heterogen antara platform database yang berbeda AWS, seperti Oracle ke Microsoft SQL Server dan Oracle ke Amazon Aurora PostgreSQL Edisi yang kompatibel. AWS DMS mendukung berbagai [sumber](#) dan [target](#). AWS DMS [Target yang didukung termasuk Amazon Simple Storage Service \(Amazon S3\), AmazonDynamoDB, Amazon Redshift, Amazon Kinesis Data Streams, Amazon DocumentDB, dan Redis.](#)

Anda dapat menggunakannya AWS DMS untuk memigrasikan beban kerja Exadata ke Amazon RDS for Oracle atau ke database Oracle di Amazon EC2. AWS DMS menangani pemuatan awal serta perubahan data capture (CDC) pembaruan dari Exadata. Exadata beroperasi penuh selama proses migrasi. Jika Anda menggunakan CDC, database target tetap disinkronkan secara terus menerus dengan Exadata, sehingga cutover aplikasi Anda dapat terjadi pada waktu yang tepat.

Alat Oracle asli seperti Oracle RMAN, Oracle Data Guard, dan Oracle Data Pump lebih fleksibel dan dapat memuat data lebih cepat daripada AWS DMS. Jika Anda memigrasikan database Exadata besar (Multi-TIB), sebaiknya pilih utilitas Oracle asli ini, bukan untuk pemuatan data awal. AWS DMS

Oracle Data Pump mendukung beberapa proses pekerja yang dapat melakukan paralelisme antar-tabel dan antar-partisi untuk memuat dan membongkar tabel dalam beberapa aliran, paralel, atau jalur langsung. Semua pemrosesan impor dan ekspor di Pompa Data, termasuk membaca dan menulis file dump, ditangani oleh server dan tidak melibatkan klien. Format penyimpanan file dump Data Pump adalah format aliran internal dari API jalur langsung. Format ini sangat mirip dengan format yang disimpan dalam file data Oracle Database di dalam tablespaces. Oleh karena itu, Pompa Data tidak harus melakukan konversi sisi klien ke variabel pengikat INSERT pernyataan. Selain itu, Data Pump mendukung metode akses data, jalur langsung, dan tabel eksternal, yang lebih cepat daripada SQL konvensional. API jalur langsung memberikan kinerja aliran tunggal tercepat. Fitur tabel eksternal membuat penggunaan yang efisien dari query paralel dan kemampuan paralel DHTML dari Oracle Database. Jika migrasi Exadata ke Amazon RDS for Oracle memerlukan waktu

henti yang rendah, pendekatan migrasi Exadata yang umum adalah menggunakan Pompa Data untuk pemuatan awal dan kemudian menggunakan atau Oracle untuk CDC. AWS DMS GoldenGate

Ada batasan ketika Anda menggunakan Exadata sebagai sumber untuk. AWS DMS Untuk informasi lebih lanjut tentang ini, lihat [AWS DMS dokumentasi](#). Juga, konektivitas jaringan ke sumber (Exadata di tempat) dan target (database Oracle on AWS) diperlukan untuk. AWS DMS

Jika Anda menggunakan AWS DMS untuk pemuatan awal, pertimbangkan praktik terbaik berikut:

- Anda umumnya dapat meningkatkan kinerja dengan memilih contoh AWS DMS replikasi besar. Tabel besar membutuhkan waktu lebih lama untuk dimuat, dan transaksi pada tabel tersebut harus di-cache sampai tabel dimuat. Setelah tabel dimuat, transaksi ter-cache tersebut diterapkan dan tidak lagi disimpan dalam disk. Misalnya, jika beban memakan waktu lima jam dan menghasilkan 6 GiB transaksi setiap jam, pastikan bahwa 30 GiB ruang disk dialokasikan untuk transaksi cache. Saat pemuatan awal selesai, sebelum memulai CDC, Anda dapat memodifikasi instance AWS DMS replikasi untuk menggunakan instance yang lebih kecil.
- Untuk migrasi Exadata besar (Multi-TIB), kami sarankan Anda menggunakan AWS DMS Binary Reader alih-alih Oracle LogMiner (yang merupakan default). Pembaca Biner memiliki risiko dampak I/O atau CPU yang lebih rendah karena log ditambang secara langsung alih-alih memerlukan beberapa kueri basis data. Namun, Oracle LogMiner lebih baik ketika Anda memiliki volume perubahan yang tinggi dan Anda menggunakan Oracle ASM. Untuk menggunakan Binary Reader untuk mengakses log redo, tambahkan atribut koneksi tambahan berikut untuk titik akhir sumber:

```
useLogMinerReader=N;useBfile=Y
```

Untuk perbandingan selengkapnya, lihat [Menggunakan Oracle LogMiner atau AWS DMS Binary Reader untuk CDC](#) dalam dokumentasi. AWS DMS

- Nonaktifkan Amazon RDS for Oracle backup atau ubah NOARCHIVELOG mode pengarsipan jika Anda bermigrasi ke Oracle di Amazon EC2. Aktifkan pencadangan sebelum fase CDC atau setelah pemuatan data awal.
- Nonaktifkan semua database siaga pada. AWS Ini termasuk Amazon RDS for Oracle Multi-AZ dan replika baca. Ini juga termasuk Oracle Data Guard atau Oracle Active Data Guard standbys jika Anda bermigrasi ke Oracle di Amazon EC2.
- Jatuhkan indeks kunci primer, indeks sekunder, kendala integritas referensial, dan pemicu bahasa manipulasi data (DHTML) sebelum pemuatan awal pada database target. Aktifkan benda-benda ini sebelum memulai fase CDC.

- Untuk tabel besar, pertimbangkan untuk memecah satu tabel menjadi beberapa AWS DMS tugas dengan menggunakan pemfilteran baris, kunci, atau kunci partisi. Misalnya, jika database Anda memiliki ID kunci primer integer yang berkisar antara 1 hingga 8.000.000, buat delapan tugas dengan menggunakan pemfilteran baris untuk memigrasikan satu juta catatan untuk setiap tugas. AWS DMS Anda juga dapat menggunakan teknik ini dengan kolom tanggal.
- Bagilah AWS DMS migrasi menjadi beberapa AWS DMS tugas. Konsistensi transaksional dipertahankan dalam suatu tugas, sehingga tabel dalam tugas terpisah tidak boleh berpartisipasi dalam transaksi umum.
- Secara default, AWS DMS memuat delapan tabel sekaligus. Untuk peningkatan kinerja, Anda dapat meningkatkan nilai ini jika Anda menggunakan server replikasi besar.
- Secara default, AWS DMS proses berubah dalam mode transaksional, yang menjaga integritas transaksional. Mengubah ke opsi penerapan yang dioptimalkan batch dapat meningkatkan kinerja. Kami menyarankan Anda mematikan kendala ini selama pemuatan awal dan menyalakannya kembali untuk proses CDC.
- [Jika instance AWS DMS replikasi dan database Oracle AWS berada di cloud pribadi virtual \(VPC\) yang berbeda, kami sarankan Anda menggunakan VPC peering.](#)
- Aktifkan CloudWatch log [Amazon](#) saat Anda membuat atau memodifikasi tugas AWS DMS migrasi. Parameter ini tersedia di bagian Pengaturan Tugas saat Anda membuat AWS DMS tugas. Mengaktifkan parameter ini menangkap informasi seperti status tugas, persen lengkap, waktu berlalu, dan statistik tabel selama proses migrasi. Untuk informasi selengkapnya, lihat [Memantau tugas replikasi menggunakan Amazon CloudWatch](#) dalam AWS DMS dokumentasi.

Untuk praktik terbaik tambahan, lihat [Menggunakan database Oracle sebagai sumber AWS DMS](#) dan [Praktik terbaik AWS Database Migration Service](#) dalam AWS DMS dokumentasi.

Migrasi Oracle GoldenGate

Oracle GoldenGate adalah solusi replikasi logis. Anda dapat menggunakan alat ini untuk mereplikasi, memfilter, dan mengubah data dari satu database ke database lainnya. Anda dapat memindahkan transaksi yang berkomitmen di beberapa sistem heterogen dan mereplikasi data dari database Oracle ke database homogen lainnya dan basis data heterogen yang didukung. Oracle GoldenGate berbagi banyak karakteristik positif dan keterbatasan. AWS DMS

Kedua alat memberikan replikasi logis. Namun, AWS DMS adalah layanan terkelola yang tidak memerlukan instalasi dan konfigurasi, sedangkan Oracle GoldenGate harus diinstal dan dikonfigurasi. Anda dapat mengaturnya di tempat atau di AWS. Anda dapat menginstal Oracle

GoldenGate AWS [dengan menggunakan konfigurasi yang sangat tersedia](#) untuk memigrasikan data dari Exadata ke AWS. Jangan menginstal Oracle GoldenGate secara langsung di Exadata di tempat atau pada node database Oracle di Amazon EC2; node database harus didedikasikan untuk memproses beban kerja database.

Perbedaan utama lainnya antara AWS DMS dan Oracle GoldenGate adalah harga. AWS DMS biaya untuk penggunaan instance replikasi dan penyimpanan log. Semua transfer data ke dalam AWS DMS gratis, dan data yang ditransfer antara AWS DMS dan database di Amazon RDS dan Amazon EC2 instans di Availability Zone yang sama juga gratis. Oracle GoldenGate membutuhkan GoldenGate lisensi Oracle untuk setiap inti pada basis data sumber dan target. Anda dapat menggunakan Oracle GoldenGate untuk memigrasikan beban kerja Exadata ke Amazon RDS for Oracle atau Oracle di Amazon EC2, baik untuk pemuatan awal maupun untuk melakukan CDC dari Exadata. Proses ini memungkinkan Exadata beroperasi penuh selama proses migrasi.

Untuk memigrasikan database Exadata besar (Multi-TIB) ke Oracle di Amazon EC2, pertimbangkan untuk menggunakan Oracle RMAN, Oracle Data Guard, atau Oracle Data Pump alih-alih Oracle karena alasan berikut: GoldenGate

- Oracle GoldenGate membutuhkan konektivitas jaringan antara Exadata dan AWS
- Oracle GoldenGate tidak berkinerja sebaik alat migrasi Oracle lainnya untuk pemuatan data awal. Misalnya, untuk memigrasikan database Exadata besar ke Amazon RDS for Oracle, pertimbangkan untuk menggunakan Oracle Data Pump sebagai gantinya, karena lebih fleksibel dan dapat memuat data lebih cepat daripada Oracle GoldenGate

Jika migrasi Exadata ke Amazon RDS for Oracle memerlukan waktu henti yang rendah, pendekatan migrasi yang umum adalah menggunakan Oracle Data Pump untuk pemuatan awal dan Oracle atau untuk CDC. GoldenGate AWS DMS Keuntungan dari Oracle GoldenGate adalah dapat menangani beban awal serta CDC. CDC memungkinkan database target untuk tetap disinkronkan secara terus menerus dengan Exadata, sehingga Anda dapat beralih pada waktu yang tepat.

Ada batasan ketika Anda menggunakan Exadata sebagai sumber dengan Oracle GoldenGate. Untuk informasi tentang hal ini, lihat [Memahami Apa yang Didukung](#) dalam GoldenGate dokumentasi.

Jika Anda menggunakan Oracle GoldenGate untuk pemuatan awal, pertimbangkan praktik terbaik berikut:

- Gunakan Ekstrak dalam mode pengambilan terintegrasi untuk memanfaatkan integrasi dengan LogMiner server. Pengambilan terintegrasi memungkinkan ekstraksi mulus dari lebih banyak tipe

data dibandingkan dengan Ekstrak dalam mode klasik. Jenis data tambahan ini termasuk data terkompresi, termasuk Kompresi Dasar, pemrosesan transaksi online (OLTP), dan Exadata Hybrid Columnar Compression (HCC). Tidak ada konfigurasi tambahan yang diperlukan untuk Extract untuk membaca file log yang disimpan di Oracle ASM.

- Gunakan Replika Terpadu. Opsi ini menggunakan proses penerapan database. Ini mempertahankan integritas referensial dan secara otomatis menerapkan operasi DDL. Integrated Replicat juga menawarkan paralelisme otomatis, yang secara otomatis meningkat atau menurun berdasarkan beban kerja dan kinerja database saat ini.
- Atur BATCHSQL dalam file parameter Replicat. Secara default, Integrated Replicat mencoba menyusun ulang dan mengelompokkan pernyataan DHTML dari jenis yang sama terhadap objek yang sama dalam setiap transaksi. Menggunakan batch dapat mengurangi CPU dan menjalankan waktu pernyataan DHTML.
- Konfigurasi tabel GoldenGate detak jantung untuk memberikan tampilan lag end-to-end replikasi. Ini memungkinkan Anda untuk melihat latensi end-to-end replikasi dengan melihat tampilan GG_LAG database.
- Nonaktifkan Amazon RDS for Oracle backup atau ubah NOARCHIVELOG mode pengarsipan jika Anda menggunakan Oracle di Amazon EC2. Aktifkan pencadangan sebelum fase CDC atau setelah pemuatan data awal.
- Nonaktifkan semua database siaga di AWS. Ini termasuk Amazon RDS for Oracle Multi-AZ dan replika baca. Ini juga termasuk Oracle Data Guard atau Oracle Active Data Guard standbys jika Anda bermigrasi ke Oracle di Amazon EC2.
- Jatuhkan indeks kunci primer, indeks sekunder, kendala integritas referensial, dan pemicu bahasa manipulasi data (DHTML) sebelum pemuatan awal pada database target. Aktifkan benda-benda ini sebelum memulai fase CDC.
- [Jika instance GoldenGate replikasi Oracle dan database Oracle AWS berada di cloud pribadi virtual \(VPC\) yang berbeda, kami sarankan Anda menggunakan VPC peering.](#)

Migrasi Pompa Data Oracle

Anda dapat menggunakan Oracle Data Pump untuk memindahkan data dari satu database Oracle ke database lainnya. Data Pump memberikan berbagai manfaat, seperti mendukung rilis Oracle Database yang lebih lama (kembali ke versi 10.1) dan platform pendukung yang memiliki format, arsitektur database, dan versi yang berbeda. Anda dapat memilih untuk mengekspor database lengkap Anda atau hanya skema, ruang tabel, atau tabel tertentu.

Anda dapat mengontrol tingkat paralelisme, kompresi, dan enkripsi, dan menentukan objek dan jenis objek mana yang akan disertakan atau dikecualikan. Data Pump juga mendukung mode jaringan, di mana Anda dapat mentransfer data dengan menggunakan tautan database tanpa perlu penyimpanan perantara.

Data Pump API menyediakan cara yang cepat dan andal untuk memindahkan data dan metadata antara database Oracle. Utilitas Ekspor Pompa Data dan Impor Pompa Data didasarkan pada API Pompa Data. Instans Amazon RDS for Oracle tidak dapat diakses melalui protokol Secure Shell (SSH), jadi API Data Pump adalah satu-satunya cara untuk mengimpor data jika Anda menggunakan Data Pump untuk bermigrasi dari Exadata ke Amazon RDS for Oracle. Data Pump Command Line Interface (CLI) bukanlah opsi untuk bermigrasi ke Amazon RDS for Oracle.

Jika Anda menggunakan Pompa Data untuk pemuatan awal, pertimbangkan praktik terbaik berikut:

- Buat ruang tabel yang diperlukan sebelum Anda mengimpor data.
- Jika Anda ingin mengimpor data ke akun pengguna yang tidak ada, buat akun pengguna dan berikan izin dan peran yang diperlukan.
- Jika Anda bermigrasi ke Oracle di Amazon EC2, matikan cadangan Amazon RDS for Oracle atau ubah mode pengarsipan menjadi. NOARCHIVELOG Aktifkan cadangan sebelum Anda memulai fase CDC atau setelah pemuatan data awal.
- Matikan semua database siaga aktif. AWS Ini termasuk Amazon RDS for Oracle Multi-AZ dan replika baca. Ini juga termasuk Oracle Data Guard atau Oracle Active Data Guard standbys jika Anda bermigrasi ke Oracle di Amazon EC2.
- Jatuhkan indeks kunci primer, indeks sekunder, kendala integritas referensial, dan pemicu DHTML sebelum pemuatan awal pada database target. Aktifkan benda-benda ini sebelum Anda memulai fase CDC.
- Untuk mengimpor skema dan objek tertentu, lakukan impor dalam skema atau mode tabel.
- Batasi skema yang Anda impor ke skema yang dibutuhkan aplikasi Anda.
- Muat dan bongkar data secara paralel dengan menggunakan kompresi dan beberapa utas.
- File di Amazon S3 harus 5 TiB atau kurang. Gunakan PARALLEL opsi untuk membuat beberapa file dump Pompa Data untuk menghindari batasan ini.
- Jika Anda berencana untuk melakukan CDC setelah ekspor Pompa Data, gunakan Oracle system change number (SCN) dengan Data Pump.
- Jika Anda ingin memuat data ke Amazon RDS for Oracle, lakukan tugas-tugas ini:
 1. Buat kebijakan AWS Identity and Access Management (IAM) untuk mengizinkan Amazon RDS mengakses bucket S3.

2. Buat peran IAM dan lampirkan kebijakan.
3. Kaitkan peran IAM dengan instans Amazon RDS for Oracle.
4. Konfigurasi grup opsi Amazon RDS for Oracle untuk integrasi Amazon S3 dan tambahkan ke instans Amazon RDS for Oracle.

Untuk informasi tambahan, lihat [Integrasi Amazon S3](#) dalam dokumentasi Amazon RDS.

Migrasi Oracle RMAN

Oracle Recovery Manager (RMAN) adalah alat untuk membuat cadangan dan memulihkan database Oracle. Hal ini juga digunakan untuk memfasilitasi migrasi database di tempat dan antara database lokal dan cloud.

Oracle RMAN menyediakan pendekatan migrasi fisik. Untuk alasan ini, ini mendukung rehosting (migrasi ke Amazon EC2) tetapi tidak dapat digunakan untuk memplatform ulang Oracle Database Anda di Amazon RDS for Oracle. Toleransi downtime migrasi Anda harus cukup besar untuk mencadangkan dan memulihkan cadangan tambahan Oracle RMAN.

Bermigrasi ke Amazon S3

Untuk mencadangkan database Exadata Anda ke Amazon S3, Anda dapat menggunakan opsi berikut:

- Gunakan Modul Cloud [Oracle Secure Backup \(OSB\)](#) untuk mencadangkan database Exadata Anda langsung ke Amazon S3.
- Salin set cadangan Oracle RMAN ke Amazon S3 dari lokasi cadangan Exadata RMAN.
- Gunakan Peralatan Penyimpanan Oracle ZFS. Set cadangan Oracle RMAN yang disimpan di Oracle ZFS Storage Appliances dapat ditransfer langsung ke Amazon S3 dengan menggunakan [Oracle ZFS](#) Storage Appliance S3 Object API Service.
- Simpan backup Oracle RMAN langsung di Exadata Storage Server, Oracle Zero Loss Recovery Appliance, dan pustaka tape. Anda kemudian dapat mentransfer set cadangan RMAN pada salah satu platform penyimpanan ini ke Amazon S3.

Migrasi ke Amazon EC2

Anda juga dapat menggunakan RMAN untuk mencadangkan database Exadata Anda langsung ke Oracle Database di Amazon EC2 tanpa membuat set cadangan. Untuk melakukan ini, gunakan

DUPLICATE perintah Oracle RMAN untuk melakukan pencadangan dan pemulihan. Namun, Oracle RMAN DUPLICATE tidak direkomendasikan untuk migrasi Exadata besar (Multi-TIB).

Pengaturan RMAN biasanya dikonfigurasi berdasarkan faktor-faktor seperti ukuran cadangan, CPU Exadata, kompresi, dan paralelisme atau jumlah saluran RMAN. Menggunakan Oracle Service Bus (OSB) dan kompresi (rendah, sedang dan tinggi) dengan RMAN memerlukan lisensi Oracle Advanced Compression Option (ACO). OSB juga memerlukan lisensi Oracle yang didasarkan pada jumlah saluran RMAN yang ingin Anda gunakan dengan OSB.

Jika Anda ingin menggunakan RMAN untuk memigrasikan Exadata ke Oracle di Amazon EC2, pertimbangkan praktik terbaik berikut.

Note

Perintah yang disediakan di bagian ini harus dijalankan di Oracle pada instans Amazon EC2.

- Jika Anda ingin menggunakan nama grup disk Oracle ASM yang berbeda di Amazon EC2, jalankan perintah dengan proses set newname pemulihan RMAN:

```
set newname for datafile 1 to '+<disk_group>'; set newname for datafile 2 to '+<disk_group>';
```

- Jika redo log online akan berada di lokasi yang berbeda AWS, ganti nama file redo log:

```
alter database rename file '/<old_path>/redo01.log' to '+<disk_group>';  
alter database rename file '/<old_path>/redo02.log' to '+<disk_group>';
```

- Setelah Anda berhasil membuka database di AWS:

- Hapus grup redo log untuk redo thread dari instance lain:

```
alter database disable thread 2;  
alter database drop logfile group 4;  
alter database clear unarchived logfile group 4;
```

- Hapus ruang tabel undo dari instance lain:

```
drop tablespace UNDOTBS2 including contents and datafiles;
```

- Pastikan hanya ada satu TEMP tablespace. Hapus TEMP tablespace yang tidak perlu dan konfirmasikan bahwa TEMP tablespace yang ada cukup besar untuk menangani beban kerja database yang diantisipasi.

Pertimbangan HCC

Jika Anda menggunakan Hybrid Columnar Compression (HCC) di Exadata, semua tabel dengan HCC harus dikonversi ke Oracle ACO atau dinonaktifkan. AWS Jika tidak, pernyataan SQL akan gagal ketika Anda mengakses database Oracle Anda di Amazon EC2. Oracle ACO membutuhkan lisensi Oracle.

Biasanya, pengguna tidak dapat menghapus HCC dari database produksi Exadata lokal. Anda dapat menghapus HCC ketika Anda memigrasi database Anda ke AWS. Untuk menentukan apakah HCC diaktifkan pada tabel atau partisi setelah Anda memigrasikan database Anda ke AWS, jalankan pernyataan SQL berikut:

```
select TABLE_NAME, COMPRESSION, COMPRESS_FOR
from DBA_TABLES
where OWNER like 'SCHEMA_NAME';

select TABLE_NAME, PARTITION_NAME, COMPRESSION, COMPRESS_FOR
from DBA_TAB_PARTITIONS
where TABLE_OWNER = 'SCHEMA_NAME';
```

Jika nilai `compression` kolom diatur ke `ENABLED` dan `compress_for` kolom memiliki salah satu nilai berikut, HCC diaktifkan:

- QUERY LOW
- QUERY HIGH
- ARCHIVE LOW
- ARCHIVE HIGH
- QUERY LOW ROW LEVEL LOCKING
- QUERY HIGH ROW LEVEL LOCKING
- ARCHIVE LOW ROW LEVEL LOCKING
- ARCHIVE HIGH ROW LEVEL LOCKING
- NO ROW LEVEL LOCKING

Untuk mematikan HCC pada tabel atau partisi, jalankan pernyataan SQL berikut:

```
alter table table_name nocompress;  
alter table table_name modify partition partition_name nocompress;
```

Untuk mengaktifkan Oracle ACO AWS, ikuti instruksi dalam dokumentasi [Oracle](#).

Migrasi Oracle Data Guard

Oracle Data Guard memungkinkan Anda untuk membuat dan mengelola satu atau lebih database siaga untuk ketersediaan tinggi dan pemulihan bencana. Data Guard memelihara database siaga sebagai salinan dari basis data utama (biasanya produksi). Jika database produksi menghadapi masalah ketersediaan yang direncanakan atau tidak direncanakan, Data Guard dapat beralih peran untuk memastikan downtime minimal dan kontinuitas aplikasi.

Anda dapat menggunakan metode siaga logis dan siaga fisik untuk mengimplementasikan Data Guard. Dalam panduan ini, kami berasumsi bahwa Anda menggunakan database siaga fisik yang sama persis dengan database utama.

Data Guard mendukung migrasi dari Exadata ke Oracle Database di Amazon EC2 untuk membuat siaga fisik. Ini tidak dapat digunakan untuk bermigrasi ke Amazon RDS for Oracle, yang memerlukan pendekatan migrasi logis AWS DMS seperti, Oracle Data Pump, atau Oracle GoldenGate

Data Guard adalah pendekatan yang lebih sederhana dan lebih cepat untuk memigrasi seluruh database Exadata dibandingkan dengan mekanisme CDC seperti atau Oracle GoldenGate. Biasanya pendekatan yang disarankan jika Anda memiliki persyaratan downtime minimal (misalnya, Anda hanya punya waktu untuk peralihan).

Anda dapat mengonfigurasi Penjaga Data dengan transportasi sinkron atau asinkron. Secara umum, pelanggan Oracle memiliki kesuksesan yang lebih besar dengan transportasi sinkron ketika latensi jaringan pulang pergi kurang dari 5 ms. Untuk transportasi asinkron, Oracle merekomendasikan latensi jaringan pulang pergi yang kurang dari 30 ms.

Biasanya, siaga Data Guard sudah ada untuk database lokal Exadata produksi. Oracle di Amazon EC2 biasanya berfungsi sebagai database siaga tambahan untuk database lokal Exadata produksi. Kami menyarankan Anda membuat database siaga Data Guard AWS dengan menggunakan Oracle RMAN.

Ada banyak variabel yang mempengaruhi kinerja Data Guard. Kami menyarankan Anda melakukan pengujian sebelum menarik kesimpulan apa pun tentang dampak replikasi Data Guard pada beban kerja Anda.

Latensi (diukur melalui monitor ping) tidak signifikan untuk replikasi Data Guard, karena mekanisme yang digunakan berbeda. Utilitas Oracle `oratcptest` membantu menilai sumber daya jaringan. Anda dapat mengunduh `oratcptest` dalam format JAR dari My [Oracle Support \(MOS\) Note 2064368.1](#) ([memerlukan akun Oracle](#)). Catatan MOS juga memberikan informasi lebih lanjut tentang utilitas ini.

AWS pola migrasi sampel

Katakanlah Anda memiliki database Exadata 50 GiB yang harus di-replatform AWS (dimigrasikan ke Amazon RDS for Oracle). Pendekatan migrasi yang Anda gunakan akan bergantung pada faktor-faktor seperti toleransi downtime Anda, metode koneksi Anda, dan ukuran database Anda.

Tabel berikut memberikan contoh pendekatan migrasi yang paling efektif berdasarkan faktor-faktor kunci. Pendekatan migrasi yang paling sesuai dengan kebutuhan Anda tergantung pada kombinasi spesifik dari faktor-faktor ini.

Database sumber	Basis data target	Ukuran basis data	Toleransi downtime migrasi	Jaringan ke AWS	Pendekatan migrasi terbaik
Exadata 12c	Amazon RDS for Oracle 19c	1 TiB	48 jam	1 Gbps AWS Direct Connect	Gunakan Pompa Data Oracle.
Exadata 12c	Amazon RDS for Oracle 21c	5 TiB	2 jam	10 Gbps t AWS Direct Connect	Gunakan Oracle Data Pump untuk beban awal dan AWS DMS untuk CDC.
Exadata 19c	Oracle 19c di Amazon EC2	10 TiB	72 jam	10 Gbps AWS Direct Connect	Gunakan Oracle RMAN.

Database sumber	Basis data target	Ukuran basis data	Toleransi downtime migrasi	Jaringan ke AWS	Pendekatan migrasi terbaik
Exadata 19c	Oracle 19c di Amazon EC2	70 TiB	4 jam	1 Gbps AWS Direct Connect	Gunakan AWS Snowball untuk mentransfer cadangan RMAN, file log redo yang diarsipkan, dan mengontrol file ke. AWS Buat instance database siaga Oracle Data Guard di Amazon EC2 dari cadangan Exadata RMAN. Lakukan peralihan Data Guard setelah database siaga dikonfigurasi di Amazon EC2 dan sinkron.

Database sumber	Basis data target	Ukuran basis data	Toleransi downtime migrasi	Jaringan ke AWS	Pendekatan migrasi terbaik
Exadata 19c	Amazon RDS untuk PostgreSQL 13.4	10 TiB	2 jam	AWS Direct Connect 10 Gbps	Gunakan AWS Schema Conversion Tool (AWS SCT) untuk membuat skema PostgreSQL. Gunakan AWS DMS untuk beban penuh dan CDC.

Pertimbangan fitur khusus Exadata

Exadata memiliki perangkat lunak berpemilik yang berjalan pada sel penyimpanan untuk meningkatkan kinerja kueri, menurunkan latensi log ulang, mengompres data, dan meningkatkan operasi basis data lainnya. Banyak dari fitur ini tidak tersedia untuk database Oracle. AWS Kami menyarankan Anda mempertimbangkan untuk melakukan tugas-tugas yang dibahas nanti di bagian ini untuk mencapai kinerja yang setara dan fungsionalitas serupa.

Anda dapat menonaktifkan fungsionalitas Exadata pada sistem Exadata non-produksi untuk mendapatkan dasar bagaimana database akan bekerja tanpa fungsi ini. Anda dapat membandingkan baseline ini dengan tes kinerja pertama AWS untuk perbandingan yang realistis.

Petunjuk berikut menjelaskan cara menonaktifkan fungsionalitas Exadata pada sistem Exadata yang ada. Kami menyarankan Anda melakukan langkah-langkah ini di lingkungan non-produksi untuk menangkap dasar bagaimana database non-Exadata akan bekerja.

- Untuk menonaktifkan pemrosesan pembongkaran sel Server Penyimpanan Exadata: Mekanismenya bergantung pada ruang lingkup perubahan (tingkat pernyataan, tingkat sesi, atau tingkat basis data).

- Untuk pernyataan SQL, gunakan petunjuk SQL berikut:

```
select /*+ OPT_PARAM('cell_offload_processing' 'false') */ max(ORDER_DATE)
from SALES;
```

- Untuk sesi Oracle, atur parameter inialisasi database Oracle berikut:

```
alter session set CELL_OFFLOAD_PROCESSING=FALSE;
```

- Untuk seluruh database Exadata, atur parameter inialisasi database Oracle berikut:

```
alter system set CELL_OFFLOAD_PROCESSING=FALSE;
```

- Untuk menonaktifkan pengindeksan penyimpanan Exadata: Untuk mematikan pengindeksan penyimpanan Exadata untuk seluruh database Exadata, tetapkan parameter inialisasi database Oracle berikut:

```
alter system set KCFISSTORAGEIDX_DISABLED=TRUE scope=both;
```

- Untuk menonaktifkan pembongkaran dekripsi ke Exadata Storage Server: Secara default, dekripsi ruang tabel terenkripsi dan kolom terenkripsi diturunkan ke Exadata Storage Server. Untuk menonaktifkan decryption offload ke Exadata Storage Server, jalankan perintah berikut:

```
alter system set CELL_OFFLOAD_DECRYPTION=FALSE;
```

- Smart Flash Cache: Oracle tidak merekomendasikan mematikan Exadata Smart Flash Cache kecuali diarahkan oleh Oracle Support atau Oracle Development.

Dalam pengembangan produk tangkas, sprint adalah periode waktu tertentu di mana pekerjaan tertentu harus diselesaikan dan disiapkan untuk ditinjau. Setelah Anda memigrasikan database Exadata ke AWS dan menyelesaikan tiga atau empat sprint, tidak jarang IOPS dikurangi 30-70 persen. Selain itu, throughput penyimpanan dapat dikurangi hingga 90 persen dari nilai yang dilaporkan Exadata. Seperti disebutkan sebelumnya, Anda dapat menguji IOPS dan throughput pada sistem non-produksi Exadata yang merupakan salinan dari sistem produksi Exadata. Anda dapat menonaktifkan pemrosesan pembongkaran sel Exadata Storage Server, dekripsi Exadata Storage Server, dan indeks penyimpanan Exadata. Selain itu, Anda mungkin harus menyelesaikan hal berikut pada sistem non-produksi Exadata setelah memigrasikan Exadata ke: AWS

- Tambahkan indeks untuk meningkatkan kueri yang tidak diindeks. Jika indeks diubah menjadi tidak terlihat, Anda mungkin harus membuatnya terlihat dengan menggunakan `ALTER INDEX` pernyataan. Setiap indeks memerlukan pemeliharaan untuk menyisipkan, memperbarui, dan menghapus pernyataan.
- Tulis ulang kueri yang tidak dapat ditingkatkan dengan indeks.
- Tentukan apakah Anda dapat menjalankan beberapa pernyataan SQL lebih jarang.

Setelah beberapa sprint pengembangan, AWS pelanggan yang memindahkan sistem Exadata mereka ke Amazon EC2 AWS melaporkan hasil berikut, berdasarkan rata-rata di seluruh snapshot [Oracle Automatic Workload Repository \(AWR\)](#). Database Oracle pada AWS kinerja rata-rata 220 persen lebih baik daripada database lokal Exadata, meskipun IOPS puncak dan puncak throughput (MBps) lebih rendah. Juga, AWS database hanya memiliki 20 persen inti dibandingkan dengan Exadata di tempat.

Lingkungan	Puncak IOPS	Throughput puncak (MBps)
Exadata di tempat	201,470	62,617
Oracle di Amazon EC2	66,420	4,640

Pertimbangan migrasi database homogen

Bagian ini membahas praktik terbaik utama untuk migrasi homogen. Saat Anda memigrasikan database dari Exadata di tempat ke Amazon RDS for Oracle atau Oracle di Amazon EC2, pertimbangkan pedoman yang dibahas dalam subbagian berikut.

Enkripsi

Keamanan data adalah prioritas utama di AWS. AWS telah menerapkan langkah-langkah kontraktual, teknis, dan organisasi yang ketat untuk melindungi kerahasiaan, integritas, dan ketersediaan pelanggan. Untuk database, enkripsi sangat penting karena melindungi informasi pribadi dan data sensitif. Oracle di Amazon EC2 dan Amazon RDS for Oracle mendukung dua metode enkripsi untuk data saat istirahat:

- [AWS Key Management Service \(AWS KMS\)](#) untuk mengenkripsi volume Amazon EBS.

- [Oracle Advanced Security Option Transparent Data Encryption \(TDE\)](#) untuk mengenkripsi informasi sensitif yang disimpan dalam file data. Oracle TDE membutuhkan lisensi Oracle.

Kedua opsi mengenkripsi data pengguna di database Oracle dan di semua cadangan database. Enkripsi juga transparan terhadap pernyataan DML yang dikeluarkan dari aplikasi.

Untuk data dalam perjalanan, Oracle di Amazon EC2 dan Amazon RDS for Oracle mendukung Oracle Native Network Encryption (NNE). Untuk informasi selengkapnya tentang dukungan NNE, lihat dokumentasi [Amazon RDS](#).

Pembuatan Partisi Data

Dengan Oracle Partitioning, objek logis tunggal dalam database, seperti tabel atau indeks, dibagi menjadi objek database fisik yang lebih kecil, yang membantu meningkatkan pengelolaan, kinerja, dan ketersediaan. Oracle Partitioning membutuhkan lisensi Oracle.

Jika Anda memiliki beban kerja database yang besar, pertimbangkan untuk mempartisi tabel Anda. Pemangkas partisi memungkinkan pengoptimal Oracle Database untuk menganalisis FROM dan WHERE klausul dalam pernyataan SQL untuk menghilangkan partisi yang tidak dibutuhkan saat membangun daftar akses partisi. Oracle Database melakukan operasi hanya pada partisi yang relevan dengan pernyataan SQL, yang biasanya meningkatkan kinerja.

Partisi juga membantu ketersediaan. Jika partisi offline dan pernyataan SQL tidak memerlukan partisi offline untuk menyelesaikan operasi, pernyataan SQL akan berhasil. Namun, jika blok data hilang dalam tabel Oracle Database yang belum dipartisi, seluruh tabel tidak akan tersedia sampai operasi pemulihan selesai.

Kompresi data

Untuk kompresi data, Oracle menawarkan HCC dan Advanced Compression. Kompresi Lanjutan meningkatkan kinerja dan mengurangi biaya penyimpanan dengan mengurangi jejak penyimpanan database untuk data relasional (tabel), data tidak terstruktur (file), indeks, data pengulangan Data Guard, data jaringan, cadangan RMAN, dan jenis data lainnya. Advanced Compression juga dapat meningkatkan kinerja komponen infrastruktur database, termasuk memori dan bandwidth jaringan.

Menurut [dokumentasi Oracle](#), Advanced Compression memiliki tingkat kompresi rata-rata minimal 2x. Oleh karena itu, 100 GiB data biasanya dapat berada di 50 GiB ruang penyimpanan. Saat memigrasikan database Oracle ke AWS, Anda dapat menggunakan Advanced Compression

diAmazon RDS for Oracle dan Oracle di Amazon EC2, dengan database OLTP dan data warehousing. Anda dapat mempertimbangkan untuk menggunakan Kompresi Lanjutan dengan database Oracle Anda AWS untuk meningkatkan kinerja dan menurunkan biaya penyimpanan Amazon EBS meskipun Anda tidak menggunakannya dengan Exadata. Advanced Compression membutuhkan lisensi Oracle.

Strategi ILM

Information Lifecycle Management (ILM) menyediakan proses, kebijakan, dan komponen yang membantu mengelola informasi dalam database berdasarkan frekuensi penggunaannya. Saat Anda bermigrasi dari Exadata ke Oracle aktif AWS, Anda harus menentukan apakah Anda dapat membersihkan data apa pun sebelum atau setelah memigrasinya ke AWS. Pada AWS, Anda dapat menerapkan aturan untuk memelihara data untuk jangka waktu tertentu saja. Anda dapat menerapkan Oracle Partitioning dan Oracle Advanced Compression untuk menyiapkan kebijakan siklus hidup data. Hal ini dapat meningkatkan kinerja sambil mempertahankan hanya data yang diperlukan untuk mendukung bisnis Anda.

Misalnya, katakanlah Anda memiliki tabel yang mengkonsumsi beberapa terabyte data yang tidak terkompresi. Saat ini Anda memiliki data 12 tahun, dan Anda harus menyimpan data selama 14 tahun. Sekitar 90 persen dari semua kueri mengakses data yang berusia kurang dari dua tahun. Anda biasanya membandingkan penggunaan data dari bulan ke bulan, kuartal ke kuartal, dan tahun ke tahun. Data tidak dapat diperbarui setelah 30 bulan, tetapi terkadang Anda harus mengakses data historis yang berusia hingga 12 tahun. Dalam hal ini, Anda dapat mempertimbangkan kebijakan ILM berikut:

- Menerapkan Kompresi Lanjutan. Manfaatkan Oracle Heat Map dan Automatic Data Optimization (ADO) dengan Advanced Compression.
- Mengatur partisi interval pada kolom tanggal.
- Gunakan fungsi yang menjatuhkan partisi yang lebih tua dari 14 tahun setiap bulan.
- Gunakan ruang tabel hanya-baca untuk menyimpan data yang berusia lebih dari 30 bulan. Tujuan utama dari read-only tablespaces adalah untuk menghilangkan kebutuhan untuk melakukan backup dan pemulihan sebagian besar, statis database (ketika Anda menggunakan Oracle RMAN dengan Oracle di Amazon EC2). Ruang tabel hanya-baca juga menyediakan cara untuk melindungi data historis sehingga pengguna tidak dapat memodifikasinya. Membuat tablespace read-only mencegah pembaruan pada semua tabel di tablespace, terlepas dari tingkat hak istimewa pembaruan pengguna.

Pengguna sering menyimpan data aktif, data yang jarang diakses, dan mengarsipkan data dalam satu database Oracle. [Selama migrasi database Oracle ke AWS, Anda dapat memigrasikan data yang jarang diakses, data audit historis, dan mengarsipkan data langsung ke Amazon S3 atau Amazon S3 Glacier.](#) Ini membantu Anda memenuhi kebutuhan tata kelola dan kepatuhan Anda untuk retensi data jangka panjang tanpa memengaruhi kinerja basis data. Seiring bertambahnya usia data dalam database relasional, data dapat diarsipkan ke Amazon S3 atau [Amazon S3 Glacier](#). Anda dapat dengan mudah menanyakan data yang diarsipkan dengan menggunakan [Amazon Athena atau Amazon S3 Glacier Select](#).

Integrasi OEM

Saat memigrasikan beban kerja Oracle ke AWS, Anda mungkin ingin menerapkan Kontrol Cloud Oracle Enterprise Manager (OEM). AWS OEM adalah platform manajemen Oracle yang menyediakan antarmuka tunggal untuk mengelola lingkungan Oracle.

Oracle di Amazon EC2 dan Amazon RDS for Oracle dapat menjadi target untuk lingkungan OEM. Oracle di Amazon EC2 mengikuti proses yang sama seperti Oracle di tempat untuk diintegrasikan dengan OEM. Untuk mengaktifkan OEM di Amazon RDS for Oracle:

1. Masuk ke AWS Management Console dan buka konsol Amazon RDS di <https://console.aws.amazon.com/rds/>.
2. Di panel navigasi, pilih Grup Opsi.
3. Tambahkan OEM_AGENT opsi ke grup opsi baru atau yang sudah ada.
4. Tambahkan informasi konfigurasi OEM, termasuk nama host server manajemen OEM, port, dan kata sandi pendaftaran agen OEM.

Amazon RDS for Oracle dan Oracle di Amazon EC2 juga dapat menjadi target untuk lingkungan OEM yang berjalan di tempat. Namun, ini mengharuskan semua port OEM dapat diakses melalui firewall Anda.

CloudWatch Integrasi Amazon

Amazon CloudWatch mengumpulkan data pemantauan dan operasional dalam bentuk log, metrik, dan peristiwa. Ini memvisualisasikan data dengan menggunakan dasbor otomatis yang menyediakan tampilan terpadu AWS sumber daya, aplikasi, dan layanan yang berjalan di AWS dan di tempat. Database Oracle yang di-host di Amazon EC2 dan Amazon RDS for Oracle dapat digunakan. CloudWatch

CloudWatch dan Amazon Simple Notification Service (Amazon SNS) terintegrasi sehingga Anda dapat mengumpulkan, melihat, dan menganalisis metrik untuk setiap notifikasi Amazon SNS yang aktif. Misalnya, Anda dapat mengatur alarm untuk mengirim pemberitahuan email atau SMS jika tindakan tertentu, seperti pesan kesalahan Oracle tertentu di log peringatan Oracle Database, terjadi.

Untuk menggunakan CloudWatch dan Amazon SNS dengan Oracle di Amazon EC2, Anda harus menginstal CloudWatch agen untuk mendorong log peringatan Oracle, log audit, log jejak, log OEM, dan log pendengar ke CloudWatch. Jika Anda menerapkan Amazon RDS for Oracle, Anda harus memodifikasi instance Oracle untuk mengaktifkan log ini untuk dikirim. CloudWatch Untuk informasi selengkapnya tentang CloudWatch integrasi, lihat [Memantau topik Amazon SNS menggunakan dokumentasi CloudWatch](#) Amazon SNS.

Amazon RDS for Oracle juga CloudWatch memiliki alarm bawaan untuk puluhan acara, termasuk pemanfaatan CPU, jumlah koneksi database, memori yang tersedia, ruang penyimpanan gratis, IOPS penyimpanan, throughput disk, dan lag replikasi.

Sebagian besar pengguna yang bermigrasi dari Exadata di tempat untuk AWS terus menggunakan OEM dan juga berintegrasi CloudWatch dengan database Oracle mereka di AWS.

Statistik pengoptimal basis data

Statistik pengoptimal Oracle Database memberikan informasi tentang database dan tabel, kolom, indeks, dan sistemnya. Pengoptimal menggunakan informasi ini untuk memperkirakan jumlah baris dan byte yang diambil dari tabel, partisi, atau indeks untuk kueri, untuk memperkirakan biaya akses, dan untuk memilih rencana eksekusi SQL yang memiliki biaya terendah.

Jika Anda memulihkan database lokal Exadata ke Amazon EC2 melalui Oracle RMAN, Oracle secara otomatis menyediakan statistik yang mencerminkan lingkungan Exadata. Segera setelah Anda memulihkan database Exadata ke Amazon EC2 atau pemuatan awal selesai di Amazon RDS for Oracle, praktik terbaik adalah mengumpulkan statistik sesegera mungkin. Hal ini dapat dicapai dengan menjalankan paket [Oracle DBMS_STATS](#).

Pengaturan AWR

Oracle Automatic Workload Repository (AWR) menyimpan statistik terkait kinerja untuk database Oracle. Secara default, Oracle Database menghasilkan snapshot sekali setiap jam, dan mempertahankan snapshot selama 8 hari. Anda dapat membuat atau menjatuhkan snapshot secara manual, dan memodifikasi pengaturan snapshot.

Untuk database Oracle produksi, Anda harus meningkatkan periode retensi AWR menjadi 60 atau 90 hari dan mengurangi interval AWR menjadi 15 atau 30 menit. Pengaturan ini mendukung month-over-month perbandingan dan memberikan lebih banyak perincian saat Anda melihat data AWR. Perubahan ini mengkonsumsi ruang database yang relatif kecil (diukur dalam gibibytes) dan memberikan manfaat sejarah tambahan. Untuk mengatur periode retensi AWR ke 60 hari dan interval AWR menjadi 15 menit, jalankan perintah berikut (nilai parameter dalam menit):

```
BEGIN
DBMS_WORKLOAD_REPOSITORY.modify_snapshot_settings
    (interval => 15,
     retention => 86400
    );
END;
/
```

Jika Anda memigrasikan database lokal Exadata ke Oracle di Amazon EC2 menggunakan Oracle RMAN atau Oracle Data Guard, Anda harus menghapus snapshot AWR yang diambil saat database berjalan di Exadata. Untuk melakukan ini, gunakan `DBMS_WORKLOAD_REPOSITORY.DROP_SNAPSHOT_RANGE` prosedur pada AWS.

Pertimbangan Oracle RAC

Secara default, Exadata menggunakan Oracle Real Application Clusters (RAC), yang memungkinkan Anda menjalankan database Oracle tunggal di beberapa server untuk memaksimalkan ketersediaan dan mengaktifkan skalabilitas horizontal. Oracle RAC menggunakan penyimpanan bersama. Penawaran Exadata terkecil mencakup dua node yang dikonfigurasi dengan menggunakan Oracle RAC.

Jika Anda memiliki persyaratan RPO nol dan persyaratan RTO dua menit atau kurang, Anda dapat menerapkan Amazon RDS for Oracle dengan Multi-AZ. Konfigurasi ini memberikan komitmen uptime bulanan 99,95%, yang setara dengan atau lebih baik daripada database cloud Oracle yang dikelola di industri, termasuk database Oracle terkelola yang menggunakan Oracle RAC.

Selain itu, Oracle di Amazon EC2 memungkinkan Anda menerapkan database yang sangat tersedia dengan menggunakan banyak komponen dalam [Oracle Maximum Availability Architecture \(MAA\)](#). Komponen-komponen ini termasuk, namun tidak terbatas pada, Active Data Guard, RMAN, Flashback Technologies, Edition-Based Redefinition, dan GoldenGate

Ada juga berbagai alternatif untuk mengimplementasikan Oracle RAC di AWS. Untuk mengetahui lebih lanjut tentang opsi RAC AWS, kami sarankan Anda menghubungi tim AWS akun Anda.

Praktik terbaik tambahan untuk migrasi homogen

Pengembang sering mengabaikan teknik penyetelan SQL dan praktik terbaik saat mereka menerapkan Exadata. Exadata menyembunyikan banyak masalah desain, sehingga pernyataan SQL dapat diterapkan ke dalam produksi tanpa menilai rencana eksekusi atau konsumsi sumber daya mereka, karena mereka selesai dalam waktu berlalu yang dapat diterima. Ikuti praktik tambahan ini saat Anda memigrasikan database lokal Exadata ke Oracle. AWS

- Terapkan Pembaruan Rilis Oracle (RU) terbaru atau Revisi Pembaruan Rilis (RUR).
- Pastikan parameter COMPATIBLE inisialisasi hanya berisi tiga level (misalnya, 19.0.0). Jika pemutakhiran dilakukan setelah Anda bermigrasi ke AWS, pastikan parameter ini diubah selama proses pemutakhiran.
- Pertimbangkan caching nomor urut untuk meminimalkan I/O. Nilai default adalah 20. Jika tidak ada caching nomor urut yang tidak mencukupi, pertengkaran dapat terjadi, yang akan muncul sebagai peningkatan waktu layanan untuk DHTML.
- Jika Anda menggunakan urutan, validasi nilai urutan terhadap database sumber (Exadata di tempat) untuk menghindari inkonsistensi urutan.
- Jika penyatuan koneksi tidak diterapkan pada tingkat aplikasi atau jumlah tingkatan aplikasi menghasilkan sejumlah besar koneksi database, pertimbangkan untuk menerapkan [Oracle Database Resident Connection Pooling](#) (DRCP). Fitur ini menangani memori dan menghitung sumber daya pada server database secara efisien.
- Pertimbangkan untuk menggunakan HugePages. Oracle merekomendasikan agar Anda menggunakan standar HugePages untuk Linux. [Mengaktifkan HugePages memungkinkan](#) sistem operasi untuk mendukung halaman memori yang lebih besar dari default (biasanya 4 KB). Menggunakan ukuran halaman yang sangat besar dapat meningkatkan kinerja sistem dengan mengurangi jumlah sumber daya sistem yang diperlukan untuk mengakses entri tabel halaman.
- Jika database Oracle AWS memiliki tautan database, konfirmasi bahwa parameter OPEN_LINKS dan OPEN_LINKS_PER_INSTANCE inisialisasi tidak disetel ke nilai default (4). Jika nilai ini terlalu rendah, pernyataan SQL yang memiliki tautan database mulai mengantri ketika nilai maksimum tercapai, yang berdampak negatif pada kinerja.
- Beban data awal mungkin tidak dapat ditransmisikan melalui jaringan. Misalnya, secara teoritis dibutuhkan setidaknya sembilan hari tanpa gangguan untuk mentransfer 100 TiB melalui tautan 1 Gbps. Pendekatan yang lebih baik adalah dengan menggunakan [AWS Snow Family](#) perangkat untuk memigrasikan database ke AWS.

- Hapus parameter tersembunyi khusus Exadata (lihat Oracle MOS Note 1274318.1). Parameter inisialisasi Exadata tersembunyi ini tidak boleh diaktifkan. AWS Mereka dapat menyebabkan ketidakstabilan, masalah kinerja, korupsi, dan crash.
- Cobalah untuk menyelesaikan semua objek yang tidak SYSTEM valid SYS dan tidak valid setelah Anda memigrasikan data ke Oracle. AWS
- Pertimbangkan caching statis, tabel yang sering diakses di Oracle System Global Area (SGA).
- Pilih instans yang dioptimalkan memori dengan konfigurasi Oracle SGA yang lebih besar untuk mengurangi tantangan I/O tambahan aktif. AWS Anda dapat menggunakan laporan Oracle SGA Advisory selama pengujian beban dalam instance target untuk menemukan konfigurasi Oracle SGA yang optimal.
- Buat indeks pada tabel yang menangani banyak pemindaian tabel penuh. V \$SEGMENT_STATISTICSTampilan mencantumkan segmen kandidat.
- Identifikasi kueri intensif sumber daya teratas dan optimalkan untuk rencana eksekusi yang lebih baik. Oracle SQL Tuning Advisor, yang dilisensikan di bawah Oracle Tuning Pack, dapat berguna untuk penyetelan SQL otomatis. Dalam beberapa kasus, Anda mungkin perlu menulis ulang kueri atau memecah kueri kompleks menjadi potongan-potongan yang lebih kecil.
- Pertimbangkan untuk menerapkan solusi caching seperti [Amazon ElastiCache dan Amazon RDS for Oracle read replicas, seperti Oracle Active Data Guard, untuk melayani beban kerja hanya-baca](#).
- Latih pengembang Anda dalam teknik pengoptimalan kueri, dan buat prosedur operasi standar untuk menilai kueri sebelum diterapkan ke produksi.
- Pastikan jumlah objek database sama dengan database lokal Exadata. AWS Validasi tabel, indeks, prosedur, pemicu, fungsi, paket, kendala, dan objek lainnya.
- Pertimbangkan modifikasi aplikasi jika memungkinkan. (Dalam beberapa kasus, aplikasi tidak dapat dimodifikasi seperti aplikasi ISV yang dikemas.) Hindari panggilan yang tidak perlu dan cobalah untuk mengurangi frekuensi panggilan yang diperlukan. Cobalah untuk meminimalkan volume data yang diambil oleh pernyataan SQL. Pastikan bahwa frekuensi komit sesuai dengan logika bisnis, tetapi tidak berlebihan. Cobalah untuk meningkatkan penggunaan caching tingkat aplikasi.
- Basis data harus berada di cloud pribadi pribadi virtual (VPC) aktif. AWS Batasi akses jaringan untuk lalu lintas masuk dan keluar ke model hak istimewa paling sedikit. Sumber grup keamanan harus merujuk ke grup keamanan di AWS akun, daftar awalan, atau kumpulan alamat IP tertentu (menggunakan format x.x.x.x/32). Sumber grup keamanan tidak boleh menggunakan CIDR dan grup keamanan tidak boleh diakses dari internet publik (0.0.0.0/0).

Rekomendasi replatforming

Sebagian besar pengguna memilih Amazon RDS for Oracle ketika mereka bermigrasi dari database lokal Exadata untuk memanfaatkan layanan database terkelola dan untuk meningkatkan kelincahan dan elastisitas. Amazon RDS for Oracle harus selalu menjadi pilihan pertama Anda untuk menjalankan AWS database Oracle karena fitur otomatisasi dan manajemennya.

Pertimbangan jenis volume Amazon EBS

Amazon RDS for Oracle menawarkan dua jenis volume EBS: General Purpose solid state drive (SSD) dan Provisioned IOPS SSD. Ukuran database Anda, persyaratan IOPS, dan estimasi throughput membantu Anda menentukan jenis volume EBS yang sesuai untuk digunakan.

Ketika aplikasi Anda tidak memerlukan kinerja penyimpanan yang tinggi, Anda dapat menggunakan penyimpanan General Purpose SSD (gp2). Performa dasar I/O untuk penyimpanan gp2 adalah 3 IOPS untuk setiap GiB, dengan minimal 100 IOPS. Ini berarti bahwa volume yang lebih besar memiliki kinerja yang lebih baik. Misalnya, kinerja dasar untuk satu volume 100 GiB adalah 300 IOPS. Performa dasar untuk satu volume 1.000 GiB adalah 3.000 IOPS. Performa dasar maksimum untuk satu volume gp2 (5334 GiB dan lebih) adalah 16.000 IOPS. Volume gp2 individual yang berukuran di bawah 1.000 GiB juga memiliki kemampuan untuk melonjak hingga 3.000 IOPS untuk jangka waktu yang lama.

Volume General Purpose SSD (gp3) mendukung maksimum 16.000 IOPS per volume EBS. Volume Amazon EBS gp3 dapat berkisar dari satu GiB hingga 16 TiB. Saat Anda menggunakan volume gp3, Anda dapat mencapai maksimum 64.000 IOPS untuk instans Amazon RDS for Oracle. Dengan menggunakan volume penyimpanan gp3, Anda dapat menyesuaikan kinerja penyimpanan secara independen dari kapasitas penyimpanan. Kinerja penyimpanan adalah kombinasi dari operasi I/O per detik (IOPS) dan seberapa cepat volume penyimpanan dapat melakukan operasi baca dan tulis (throughput penyimpanan). Pada volume penyimpanan gp3, Amazon RDS menyediakan kinerja penyimpanan dasar 3.000 IOPS dan 125 MiB/s.

Untuk setiap mesin Amazon RDS DB kecuali Amazon RDS for SQL Server, ketika ukuran penyimpanan untuk volume gp3 mencapai ambang tertentu, kinerja penyimpanan dasar meningkat menjadi 12.000 IOPS dan 500 MiB/s. Ini karena pembagian volume, penyimpanan menggunakan empat volume, bukan satu.

Volume SSD IOPS yang tersedia

Volume IOPS SSD (io1) yang disediakan dirancang untuk memenuhi kebutuhan beban kerja intensif I/O yang sensitif terhadap kinerja dan konsistensi penyimpanan. Volume Amazon EBS io1 menghasilkan latensi milidetik satu digit. Saat memilih volume Amazon EBS io1 untuk Amazon RDS for Oracle, Anda harus memberikan nilai penyimpanan yang dialokasikan dan nilai IOPS yang disediakan. Volume io1 dapat berkisar dalam ukuran dari 4 GiB hingga 16 TiB. IOPS maksimum per io1 volume adalah 64.000. Saat Anda menggunakan volume io1, Anda dapat mencapai maksimum 256.000 IOPS dan throughput maksimum 4 Gbps (membutuhkan 256 KB IOPS) untuk instans Amazon RDS for Oracle. Throughput penulisan maksimum untuk instans Amazon RDS for Oracle dengan Multi-AZ diaktifkan adalah 625 MBps.

io2 Block Express adalah opsi penyimpanan IOPS SSD Provisioned yang lebih baru. Volume io2 dapat berkisar dalam ukuran dari 4 GiB hingga 64 TiB. IOPS maksimum per io2 volume adalah 256.000. io2 Block Express juga menyediakan latensi rata-rata sub-milidetik dan karenanya mengungguli io1. Saat menggunakan penyimpanan SSD IOPS Tertentu, io2 adalah opsi yang disarankan untuk digunakan. Anda dapat meningkatkan volume io1 ke io2 Block Express volume tanpa downtime, dan secara signifikan meningkatkan kinerja dan keandalan aplikasi Anda tanpa meningkatkan biaya penyimpanan. Untuk informasi selengkapnya, lihat posting AWS blog [Amazon RDS sekarang mendukung volume io2 Block Express untuk beban kerja database yang sangat penting](#).

Praktik terbaik Amazon RDS for Oracle

Pertimbangkan praktik terbaik berikut saat Anda bermigrasi dari Exadata di tempat ke Amazon RDS for Oracle:

- Sebelum Anda memigrasikan data dari Exadata ke Amazon RDS for Oracle, tingkatkan ukuran redo log dari nilai default 128 MB. Jika tidak, pengalihan redo log mungkin terjadi terlalu sering dan menyebabkan penurunan kinerja.
- [Aktifkan Performance Insights](#) (yang memiliki periode retensi data 7 hari default) setelah pemuatan data awal.
- [Siapkan Multi-AZ](#) untuk database produksi setelah pemuatan data awal.
- [Integrasikan Amazon RDS for Oracle CloudWatch](#) dengan Amazon (minimal, gunakan log peringatan, pendengar, dan agen OEM) setelah pemuatan data awal.

- Instal Agen Oracle Enterprise Manager (OEM) di grup opsi Amazon RDS for Oracle terkait. Ini membutuhkan OEM fungsional yang sudah ada di AWS atau di tempat. Anda dapat mengatur OEM dalam [mode yang sangat tersedia AWS](#).
- [Menerapkan alarm Amazon RDS](#) untuk hal berikut untuk memberi tahu administrator sebelum kapasitas maksimum dilanggar:
 - Pemanfaatan CPU, tulis IOPS, baca IOPS, tulis throughput
 - Membaca throughput, memori yang dapat dibebaskan, penggunaan swap
- Amazon RDS mengunggah log transaksi untuk instans DB ke Amazon S3 setiap lima menit. Untuk melihat waktu restorable terbaru untuk instans DB, gunakan AWS CLI [describe-db-instances](#) perintah dan lihat nilai yang dikembalikan di LatestRestorableTime bidang untuk instans DB. Amazon RDS dapat mengunggah log transaksi lebih sering jika persyaratan point-in-time pemulihan Anda kurang dari lima menit. Untuk mengubah nilai default, ubah parameter ARCHIVE_LAG_TARGET inisialisasi di grup parameter Amazon RDS for Oracle terkait. Anda dapat mengatur nilai parameter ini ke 60, 120, 180, 240, atau 300 detik. Namun, ada pengorbanan jika Anda menetapkan nilai yang lebih rendah: Lebih banyak file log redo akan dihasilkan dan peralihan file log akan terjadi lebih sering.
- Menerapkan Oracle Unified Audit, yang merupakan kerangka audit yang direkomendasikan Oracle, dalam mode campuran. Secara default, audit terpadu tidak diaktifkan di Amazon RDS (). AUDIT_TRAIL=NONE Anda dapat mengaktifkannya dengan mengatur AUDIT_TRAIL=DB atau AUDIT_TRAIL=DB, EXTENDED. Untuk informasi selengkapnya, lihat posting AWS blog [Audit keamanan di Amazon RDS for Oracle](#): Bagian 1.
- Untuk melindungi dari ancaman internal, konfigurasi [aliran aktivitas database](#) jika berlaku. Fitur ini bekerja dengan audit terpadu Oracle dan menyediakan aliran real-time hampir semua pernyataan yang diaudit (SELECT, DML DDL DCL, TCL) yang berjalan di instans DB. Data audit dikumpulkan dari lokasi audit database terpadu, sedangkan penyimpanan dan pemrosesan aktivitas database dikelola di luar database di Amazon Kinesis Data Streams. Untuk informasi selengkapnya, lihat posting AWS blog [Audit keamanan di Amazon RDS for Oracle](#): Bagian 2.
- Jika Anda lebih suka audit standar, Anda dapat mengintegrasikan laporan audit dengan Amazon CloudWatch setelah pemuatan data awal. Saat Anda mengaktifkan audit standar dengan menyetel AUDIT_TRAIL parameter ke OS, atau XML XML, EXTENDED, Amazon RDS for Oracle menghasilkan catatan audit yang disimpan .AUD sebagai .XML atau file sistem operasi di instans Amazon RDS for Oracle. File audit ini biasanya disimpan di Amazon RDS for Oracle misalnya selama tujuh hari. Anda dapat mengonfigurasi Amazon RDS for Oracle untuk mempublikasikan CloudWatch file-file ini, di mana mereka dapat melakukan analisis real-time dari data log, menyimpan data dalam penyimpanan yang sangat tahan lama, dan mengelola data CloudWatch

dengan agen log. AWS menyimpan data log yang dipublikasikan ke CloudWatch log untuk jangka waktu yang tidak terbatas di AWS akun kecuali Anda menentukan periode retensi.

Rekomendasi rehosting

Saat Anda meng-rehost Oracle di Amazon EC2, Anda menginstal dan mengonfigurasi database Oracle dan melakukan semua operasi pemeliharaan, termasuk upgrade Oracle minor, upgrade Oracle utama, patching sistem operasi, konfigurasi sistem operasi, konfigurasi database, alokasi memori, alokasi penyimpanan, dan konfigurasi penyimpanan.

Pertimbangan jenis instans Amazon EC2

Instans EC2 harus memiliki CPU, memori, dan penyimpanan yang memadai untuk menangani beban kerja database yang diantisipasi. Kami menyarankan Anda menggunakan kelas instans EC2 generasi saat ini untuk database Oracle. Jenis instance ini, seperti instance yang dibangun di atas [Sistem Nitro](#), mendukung Hardware Virtual Machine (HVM). HVM Amazon Machine Images (AMI) diperlukan untuk memanfaatkan jaringan yang ditingkatkan, dan mereka juga menawarkan peningkatan keamanan.

Instans tervirtualisasi yang dibangun pada Sistem Nitro termasuk R5b, X2idn, dan X2IEdn. Untuk throughput volume Amazon EBS yang tinggi, pertimbangkan jenis instans Amazon EC2 R5b dan X2. Instans ini mendukung hingga 260.000 IOPS. Throughput maksimum untuk instans Amazon EC2 R5b adalah 7.500 MBps. Throughput maksimum untuk instans Amazon EC2 x2idN dan X2iEDN adalah 10.000 MBps. [Untuk informasi selengkapnya, tinjau instans Amazon EBS yang dioptimalkan dan IOPS maksimum dalam dokumentasi Amazon EC2.](#)

Pertimbangan jenis volume Amazon EBS

Volume Amazon EBS General Purpose (gp3) lebih murah daripada volume IOPS (io2) Amazon EBS Provisioned. Jika volume gp3 memenuhi persyaratan I/O dan throughput Anda, mereka harus menjadi solusi pilihan Anda. Volume gp3 tunggal tidak boleh melebihi 16.000 IOPS per volume. Anda juga harus mempertimbangkan jumlah maksimum volume EBS yang dapat ditetapkan ke instans EC2. Jumlah ini bervariasi berdasarkan jenis instans EC2; Namun, jumlah maksimum volume EBS untuk instans Sistem Nitro adalah 28. Biasanya, tidak lebih dari 24 volume EBS harus didedikasikan untuk database Oracle.

Jika persyaratan I/O disk Anda tinggi, pertimbangkan volume Amazon EBS [io2 Block Express](#). Ini dirancang untuk menawarkan throughput hingga 4.000 MBps per volume, 256.000 IOPS per volume,

kapasitas penyimpanan 64 TiB, latensi sub-milidetik, dan daya tahan 99,999%. Kami menyarankan Anda menggunakan volume Amazon EBS io2 Block Express dalam skenario berikut:

- Basis data yang dialokasikan ruang melebihi 384 TiB. Ini termasuk, tetapi tidak terbatas pada, file database, redo log, TEMP ruang, ruang, UNDO ruang Flashback Recovery Area, dan area pementasan data. Volume Amazon EBS io2 Block Express dapat mendukung hingga 1,536 piB dengan satu instans EC2.
- Anda memerlukan latensi penyimpanan dalam kisaran sub-milidetik.
- Anda memerlukan database yang dirancang untuk daya tahan 999%, dibandingkan dengan daya tahan 99,9% dengan volume Amazon EBS gp3.
- Anda memerlukan [array penyimpanan virtual](#) untuk mengirimkan 1 juta IOPS atau lebih ke satu instans EC2.
- Exadata Smart Flash Cache dan Exadata Smart Flash Logging sangat tinggi di sistem lokal Exadata Anda. Latensi I/O untuk Exadata Smart Flash Cache biasanya kurang dari 400 mikrodetik untuk operasi baca. Latensi I/O untuk Amazon EBS io2 Block Express biasanya berkisar antara 400 dan 600 mikrodetik.

Pertimbangan Oracle ASM

[Saat Anda menggunakan Oracle di Amazon EC2, Oracle AWS dan merekomendasikan agar Anda menerapkan redundansi eksternal Oracle Automatic Storage Management \(ASM\) untuk menghindari tingkat kegagalan Amazon EBS.](#) Namun, jika volume EBS menjadi tidak tersedia dalam mode redundansi eksternal ASM, grup disk ASM terkait masuk ke desmount paksa. Semua disk harus ditempatkan agar berhasil memasang grup disk ASM. Oleh karena itu, database menjadi tidak tersedia sampai semua volume EBS tersedia. Redundansi eksternal ASM secara efektif memberikan keandalan RAID level 0, sehingga kemungkinan dampak pada grup disk ASM meningkat dengan setiap volume EBS ditambahkan, dan tingkat kegagalan keseluruhan adalah kelipatan dari setiap tingkat kegagalan volume EBS individu.

Volume Amazon EBS direplikasi dalam AWS Availability Zone. Namun, volume EBS masih bisa mengalami kegagalan. Misalnya, volume gp3 memiliki tingkat kegagalan tahunan 0,1-0,2 persen, dan volume io2 memiliki tingkat kegagalan tahunan 0,001 persen. Anda dapat mengimplementasikan grup disk ASM dengan redundansi normal atau redundansi tinggi untuk mengurangi pemadaman yang disebabkan oleh kegagalan volume EBS tunggal. Namun, ini bukan praktik terbaik, karena volume EBS direplikasi dalam Availability Zone, dan volume EBS grup kegagalan ASM juga dapat berada pada host fisik yang sama dengan volume EBS grup utama ASM.

Pertimbangan ASM tambahan:

- Gunakan [Oracle ASM Filter Driver \(ASMFDD\) untuk mengimplementasikan ASM](#).
- Pastikan bahwa semua disk Oracle ASM dalam grup disk memiliki kinerja penyimpanan dan karakteristik ketersediaan yang serupa. Dalam konfigurasi penyimpanan yang memiliki drive kecepatan campuran, seperti memori flash dan hard disk drive (HDD), kinerja I/O dibatasi oleh drive kecepatan paling lambat.
- Pastikan disk Oracle ASM dalam grup disk memiliki kapasitas yang sama untuk menjaga keseimbangan.
- Oracle ASM mendistribusikan data secara acak ke dalam set disk ASM yang dipilih. Saat Anda mengonfigurasi penyimpanan sistem, pertimbangkan kapasitas awal sistem dan rencana untuk pertumbuhan di masa depan. Oracle ASM menyederhanakan tugas mengakomodasi pertumbuhan. Seperti disebutkan sebelumnya, instans Amazon EC2 Nitro System mendukung hingga 28 volume. Jika grup disk DATA ASM membutuhkan 96 TiB, empat volume 24 TiB Amazon EBS io2 Block Express akan menjadi pilihan yang lebih baik daripada enam belas volume 6 TiB Amazon EBS io2 Block Express.
- Siapkan setidaknya dua file kontrol di dua grup disk ASM.

Praktik terbaik Oracle di Amazon EC2

Setelah Anda memigrasikan data dari Exadata di tempat ke Oracle di Amazon EC2, dan sebelum Anda memberikan akses ke pengguna akhir, pertimbangkan praktik terbaik berikut:

- Aktifkan perlindungan [terminasi instans EC2](#). Ini mencegah instance EC2 dihentikan secara tidak sengaja dengan mengharuskan pengguna untuk menonaktifkan perlindungan sebelum menghentikan instance.
- Aktifkan [fitur pemulihan otomatis Amazon EC2](#), yang menyelesaikan masalah jika perangkat keras yang menghosting instans EC2 mengalami gangguan. Fitur ini memulihkan instance pada perangkat keras yang mendasari yang berbeda dan mengurangi kebutuhan akan intervensi manual.
- Amazon EC2 menawarkan instans yang memiliki memori hingga 24 TiB. Instans ini mendukung SGA Oracle yang sangat besar dan harus menjadi pilihan pertama Anda jika Anda menggunakan SGA Oracle Multi-TIB. Namun, banyak instans EC2 dan Amazon RDS for Oracle juga mendukung penyimpanan instans lokal. Jika Anda menggunakan Amazon EC2 atau [Amazon RDS untuk instans Oracle](#) dengan penyimpanan instans NVMe SSD, Anda dapat menggunakan penyimpanan sementara untuk memperluas buffer blok database Oracle SGA. Pendekatan ini memungkinkan

Anda untuk cache objek dengan menggunakan penyimpanan instance dan menyediakan latensi I/O rata-rata 100 mikrodetik untuk operasi baca. [Smart Flash Cache dan/atau Level 2 Flash Cache](#) hanya berfungsi pada instance yang menggunakan penyimpanan instans dan memerlukan sistem operasi Oracle Linux. Lingkungan OLTP dan data warehouse dapat mengambil manfaat dari teknologi ini. Atur parameter inisialisasi Oracle `DB_FLASH_CACHE_FILE` dan `DB_FLASH_CACHE_SIZE` gunakan Smart Flash Cache.

- Gunakan Oracle Linux sebagai sistem operasi untuk contoh Anda. Jika Oracle Linux bukan pilihan, pertimbangkan Red Hat Enterprise Linux (RHEL). Instans EC2 yang didasarkan pada prosesor Graviton tidak mendukung database Oracle, karena Oracle belum merilis binari Oracle Database yang dikompilasi untuk prosesor ARM. Selain itu, Amazon Linux tidak didukung untuk database Oracle.
- Gunakan rilis terbaru dari perangkat lunak Oracle untuk menginstal Oracle Grid Infrastructure. Anda dapat menyebarkan rilis terbaru dari Oracle Grid Infrastructure dengan versi Oracle Database yang lebih lama. Misalnya, Oracle Grid Infrastructure 21c mendukung Oracle Database 19c.
- Jika Anda menggunakan Oracle RMAN atau Oracle Data Guard untuk bermigrasi dari rilis Oracle Database yang lebih lama di Exadata, pertimbangkan untuk meningkatkan rilis database ke versi terbaru setelah migrasi. Jika Anda menggunakan Oracle Data Pump, instal rilis Oracle Database terbaru AWS sebelum migrasi.
- Gunakan Oracle flash recovery area (FRA) untuk memulihkan database Anda dengan cepat tanpa menggunakan cadangan [RMAN](#). Jika memungkinkan, atur FRA minimal satu hari. Anda harus mengatur parameter inisialisasi `OracleDB_RECOVERY_FILE_DEST_SIZE`, `DB_RECOVERY_FILE_DEST`, dan `DB_FLASHBACK_RETENTION_TARGET` (mewakili jumlah waktu, dalam menit).
- Jika Anda memigrasikan beberapa beban kerja database ke dalam satu instans EC2, pertimbangkan untuk menerapkan [Oracle Database Resource Manager untuk mengelola alokasi sumber daya database](#).
- Menerapkan Oracle SPFILE bukan standalonePFILE. An SPFILE adalah file biner yang memungkinkan modifikasi dinamis tanpa memerlukan instance restart. Jangan tentukan PFILE kapan menggunakan STARTUP perintah jika SPFILE sedang digunakan.
- Aktifkan [Oracle Automatic Shared Memory Manager \(ASMM\)](#), yang menyederhanakan manajemen memori SGA. Oracle Database secara otomatis mendistribusikan memori di antara komponen SGA untuk memastikan pemanfaatan memori yang paling efektif.
- Anda mungkin mengalami Oracle db file parallel write wait event dengan database writer process (DBWR). Penantian ini menunjukkan waktu yang dihabiskan DBWR menunggu penyelesaian I/O. Untuk mengatasi masalah ini, konfirmasi bahwa I/O asinkron diaktifkan (parameter inisialisasi

OracleDISK_ASYNCH_IO), tingkatkan IOPS untuk volume EBS, dan verifikasi bahwa cache buffer database cukup besar untuk mencegah thrashing.

- Jalankan pemindaian secara berkala (minimal setiap dua minggu) terhadap instans EC2 dan verifikasi kepatuhan. Anda dapat menggunakan [Amazon Inspector untuk pemindaian](#) ini. Amazon Inspector adalah layanan penilaian keamanan otomatis yang membantu meningkatkan keamanan dan kepatuhan aplikasi yang digunakan. AWS Ini secara otomatis menilai aplikasi untuk eksposur, kerentanan, dan penyimpangan dari praktik terbaik. Setelah melakukan penilaian, itu menghasilkan daftar rinci temuan keamanan yang diprioritaskan berdasarkan tingkat keparahan. Anda dapat meninjau temuan ini secara langsung atau dalam laporan penilaian terperinci yang tersedia melalui konsol Amazon Inspector atau API.
- Siapkan CloudWatch alarm Amazon untuk [AWS CloudTrail](#). Misalnya, CloudWatch alarm harus diaktifkan ketika perubahan konfigurasi terjadi pada grup keamanan. Ini memberi tahu tim operasi ketika seseorang mencoba mendapatkan akses ke instans EC2.
- Jika organisasi Anda memerlukan sasaran titik pemulihan nol atau mendekati nol (RPO), gunakan Oracle Data Guard atau Oracle Active Data Guard dalam mode ketersediaan maksimum. Database siaga harus berada di Availability Zone yang berbeda dari database utama. Perlindungan maksimum dan mode ketersediaan maksimum menyediakan lingkungan failover otomatis yang dirancang tanpa kehilangan data. Mode kinerja maksimum menyediakan lingkungan failover otomatis yang dirancang untuk kehilangan tidak lebih dari jumlah data (dalam detik) yang ditentukan oleh properti `FastStartFailoverLagLimit` konfigurasi. Kami juga menyarankan Anda menerapkan Data Guard Broker dengan Oracle Data Guard atau Oracle Active Data Guard. Data Guard Broker mengotomatiskan tugas konfigurasi dan pemantauan untuk Data Guard. Active Data Guard membutuhkan lisensi Oracle.
- Pertimbangkan untuk menggunakan Oracle Active Data Guard pemulihan media blok otomatis. Jika blok data rusak ditemui ketika Anda mengakses database utama, blok secara otomatis diganti dengan salinan blok yang tidak rusak dari database siaga fisik. Namun, untuk menggunakan fitur ini, Active Data Guard harus berjalan dalam mode ketersediaan maksimum dan memiliki parameter inisialisasi Oracle `LOG_ARCHIVE_DEST_n` diatur ke mode SYNC redo transport. Mode kinerja maksimum tidak mendukung fitur ini.
- Jika organisasi Anda memerlukan pemulihan bencana lintas wilayah, pertimbangkan untuk menerapkan [Oracle Far Sync](#). Far Sync memerlukan lisensi Oracle Active Data Guard.
- Gunakan [Oracle Secure Backup \(OSB\)](#) untuk mencadangkan database Anda ke Amazon S3 dengan menggunakan Oracle RMAN. OSB membutuhkan lisensi Oracle. Harga OSB didasarkan pada jumlah saluran Oracle RMAN yang digunakan. Anda juga dapat menggunakan [AWS Storage Gateway](#) untuk mencadangkan database Anda ke Amazon S3 secara langsung. Anda dapat

menerapkan kebijakan siklus hidup ke pencadangan di Amazon S3 untuk memindahkan cadangan lama ke Amazon S3 Glacier untuk pengarsipan.

Rekomendasi refactoring

AWS menawarkan dua alat yang mengaktifkan migrasi heterogen dari Oracle ke Amazon RDS untuk PostgreSQL atau Amazon Aurora PostgreSQL Compatible Edition. Alat-alat ini adalah [AWS Schema Conversion Tool \(AWS SCT\)](#) dan [AWS Database Migration Service \(AWS DMS\)](#).

AWS SCT secara otomatis mengubah skema basis data sumber dan sebagian besar kode kustom ke format yang kompatibel dengan database target. Selama migrasi database dari Oracle ke PostgreSQL, mengotomatiskan konversi kode Oracle PL/SQL AWS SCT ke kode PL/PGSQL yang setara di PostgreSQL. Kode khusus yang dikonversi alat mencakup tampilan, prosedur tersimpan, dan fungsi. Ketika fragmen kode tidak dapat secara otomatis dikonversi ke bahasa target, AWS SCT mendokumentasikan semua lokasi yang memerlukan input manual dari pengembang aplikasi. AWS DMS menggunakan CDC untuk memigrasikan Oracle ke PostgreSQL atau MySQL.

Untuk memigrasikan database Oracle ke PostgreSQL atau MySQL, Anda biasanya perlu menyelesaikan tugas otomatis dan manual. AWS menawarkan buku pedoman migrasi yang memberikan step-by-step instruksi untuk strategi konversi kode dasar dan kompleks. Untuk informasi tentang refactoring database Oracle Anda, lihat pedoman berikut:

- [Buku Pedoman Migrasi Oracle ke Aurora PostgreSQL](#)
- [Playbook Migrasi MySQL Oracle ke Aurora](#)

Kegiatan pasca-migrasi

Setelah Anda memigrasikan beban kerja Exadata AWS, ada tugas tambahan dan praktik terbaik yang penting untuk mempertahankan lingkungan database yang andal, sangat tersedia, berkinerja, dan dioptimalkan biaya. Bagian ini menguraikan praktik dan tip terbaik pasca-migrasi utama.

Di bagian ini:

- [Pemantauan berkelanjutan](#)
- [Alat-alat pemantauan](#)
- [Optimalisasi biaya berkelanjutan](#)
- [Pemantauan otomatis](#)
- [Audit otomatis](#)

Pemantauan berkelanjutan

Pemantauan merupakan bagian penting dari menjaga keandalan, ketersediaan, dan kinerja database. AWS Agar lebih mudah men-debug kegagalan multi-titik, kami sarankan untuk mengumpulkan data pemantauan dari semua bagian lingkungan database Anda. AWS

Bagian ini mengeksplorasi AWS layanan dan alat yang menyediakan kemampuan diagnostik kinerja tingkat lanjut. Sebelum Anda menggunakan alat-alat tersebut, kami sarankan Anda menentukan rencana pemantauan yang jelas.

Rencana pemantauan

Kami menyarankan Anda menjawab pertanyaan-pertanyaan berikut sebelum Anda membuat rencana pemantauan Anda:

- Apa saja sasaran pemantauan Anda?
- Sumber daya apa yang akan Anda gunakan untuk pemantauan?
- Seberapa sering sumber daya ini akan dipantau?
- Apa sajakah alat pemantauan yang akan Anda gunakan?
- Siapakah yang akan melakukan tugas pemantauan?
- Siapa yang harus diberi tahu saat terjadi kesalahan?

Setelah Anda menentukan rencana pemantauan Anda, buat garis dasar untuk metrik utama, untuk mengukur apakah tujuan pemantauan Anda terpenuhi.

Acuan dasar performa

Ukur kinerja dalam kondisi beban yang berbeda pada berbagai waktu. Anda dapat memantau metrik-metrik seperti:

- Pemanfaatan CPU
- Throughput jaringan
- Koneksi klien
- I/O untuk operasi baca atau tulis
- Saldo kredit burst

Ketika kinerja berada di luar garis dasar yang ditetapkan, Anda mungkin harus membuat perubahan untuk mengoptimalkan ketersediaan database untuk beban kerja. Misalnya, perubahan ini mungkin termasuk mengubah kelas instans instans DB Anda atau mengubah jumlah instans DB dan membaca replika yang tersedia untuk klien.

Pedoman kinerja utama

Secara umum, nilai yang dapat diterima untuk metrik kinerja bergantung pada apa yang dilakukan aplikasi relatif terhadap baseline. Selidiki varians yang konsisten atau tren dari baseline. Metrik-metrik berikut sering menjadi sumber masalah performa:

- Konsumsi CPU atau RAM yang tinggi. Nilai tinggi untuk konsumsi CPU atau RAM mungkin sesuai, jika konsisten dengan tujuan aplikasi seperti throughput atau konkurensi, dan diharapkan.
- Konsumsi ruang disk. Selidiki konsumsi ruang disk jika ruang yang digunakan secara konsisten pada atau di atas 85 persen dari total ruang disk. Evaluasi apakah mungkin untuk menghapus data dari instance atau mengarsipkan data ke sistem yang berbeda untuk mengosongkan ruang.
- Lalu lintas jaringan. Untuk lalu lintas jaringan, bekerja dengan administrator sistem Anda untuk menentukan throughput yang diharapkan untuk jaringan domain dan koneksi internet. Kami menyarankan Anda menyelidiki lalu lintas jaringan jika throughput secara konsisten lebih rendah dari yang diharapkan.
- Koneksi basis data. Jika Anda menemukan sejumlah besar koneksi pengguna bersama dengan penurunan kinerja instans dan waktu respons, Anda dapat mempertimbangkan untuk membatasi

koneksi database. Jumlah optimal koneksi pengguna untuk instance DB bervariasi berdasarkan kelas instance dan kompleksitas operasi yang dilakukan.

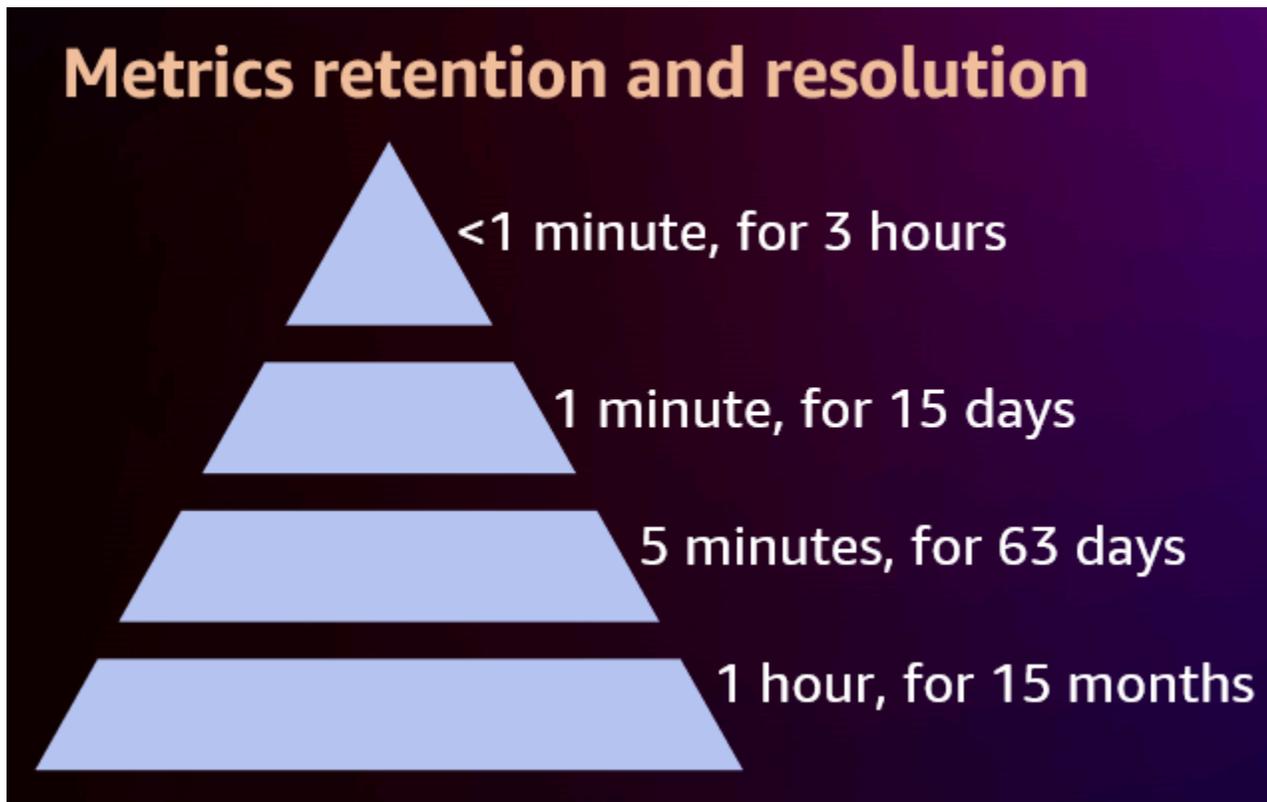
- **Metrik IOPS.** Saat Anda bermigrasi dari Oracle Exadata, pemantauan IOPS sangat penting. Oracle Exadata dikenal untuk memberikan throughput penyimpanan yang tinggi dan IOPS. Kami menyarankan Anda menentukan garis dasar untuk aktivitas I/O tipikal untuk memastikan konfigurasi terbaik aktif. AWS

Alat-alat pemantauan

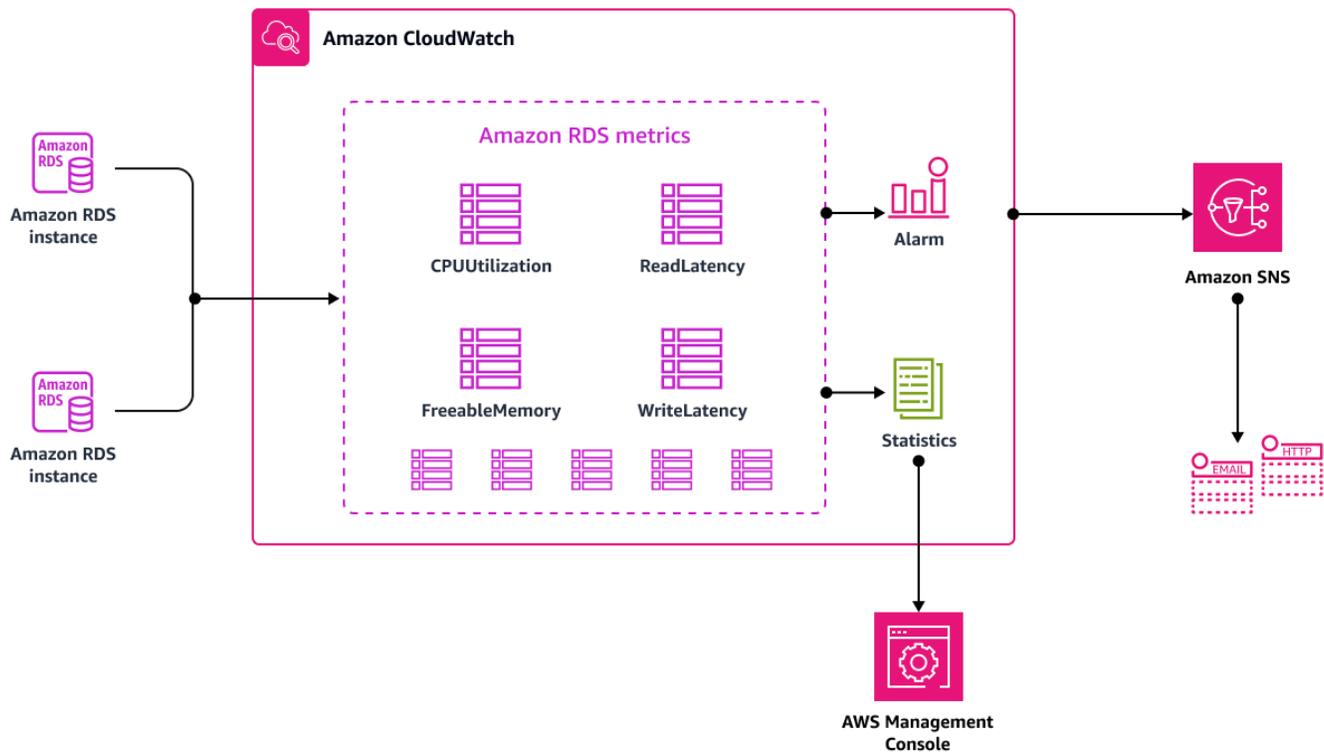
Bagian ini membahas alat pemantauan dari Amazon dan Oracle yang dapat Anda gunakan selama fase pasca-migrasi untuk mempertahankan lingkungan database yang andal, sangat tersedia, berkinerja, dan dioptimalkan biaya.

Amazon CloudWatch

[Amazon CloudWatch](#) adalah layanan pemantauan dan observabilitas yang memberikan pandangan terpadu tentang kesehatan operasional dan memberi Anda visibilitas lengkap ke AWS sumber daya, aplikasi, dan layanan yang berjalan di dalam AWS dan di tempat. Anda dapat menggunakannya CloudWatch untuk mendeteksi perilaku anomali di lingkungan Anda, menyetel alarm, memvisualisasikan log dan metrik secara berdampingan, mengambil tindakan otomatis, memecahkan masalah, dan menemukan wawasan agar aplikasi Anda berjalan lancar. Analogi terbaik untuk resolusi dan retensi CloudWatch metrik adalah struktur piramida yang diilustrasikan dalam diagram berikut. Tingkat atas mewakili frekuensi paling granular (hingga 1 detik) tetapi juga retensi metrik terendah. Saat pengguna mengeksplorasi lebih banyak data pemantauan historis, semakin sedikit titik datanya. Misalnya, untuk retensi maksimum (antara 63 hari dan 15 bulan), granularitas akan menjadi satu jam, seperti yang diilustrasikan di tingkat bawah piramida.



Seperti yang ditunjukkan diagram berikut, Anda dapat mengatur alarm untuk CloudWatch metrik. Misalnya, Anda dapat membuat alarm yang diaktifkan ketika penggunaan CPU untuk sebuah instance melebihi 70 persen.



Anda dapat mengonfigurasi Amazon Simple Notification Service (Amazon SNS) untuk mengirim email atau SMS setiap kali ambang batas dilewati. Anda juga dapat menggunakan Amazon SNS untuk memberi tahu protokol atau layanan tambahan seperti Amazon Simple Queue Service (Amazon SQS), atau HTTP/HTTPS. AWS Lambda Misalnya, Anda dapat membuat alarm yang diaktifkan jika total IOPS yang digunakan melebihi 90 persen dari maksimum yang dikonfigurasi untuk instance. Tindakan alarm mungkin merupakan fungsi Lambda yang meningkatkan jumlah IOPS yang disediakan (PIOPS) jika status alarm adalah Alarm. Untuk informasi tambahan, [lihat presentasi Menonaktifkan beban: Mendiagnosis & menyelesaikan masalah kinerja dengan Amazon RDS](#) (AWS re:invent 2023).

Pemantauan yang Ditingkatkan

Beberapa pengguna yang bermigrasi dari Oracle Exadata terbiasa memiliki visibilitas tingkat OS ke perangkat fisik yang dipetakan ke dalam grup disk ASM mereka, dan melihat metrik tingkat OS granular seperti halaman besar, aktivitas swap, dan detail daftar proses/utas. Amazon CloudWatch tidak memberikan tingkat visibilitas itu, tetapi Amazon RDS dan Amazon Aurora menawarkan Pemantauan yang Ditingkatkan, yang menyediakan pemantauan tingkat OS terperinci untuk database Anda. Enhanced Monitoring menyediakan retensi default 30 hari dan frekuensi pengambilan sampel satu menit, tetapi kedua pengaturan dapat dikonfigurasi.

[Untuk informasi selengkapnya, lihat bagian Metrik Monitoring OS dengan Enhanced Monitoring pada dokumentasi Amazon RDS dan Aurora.](#)

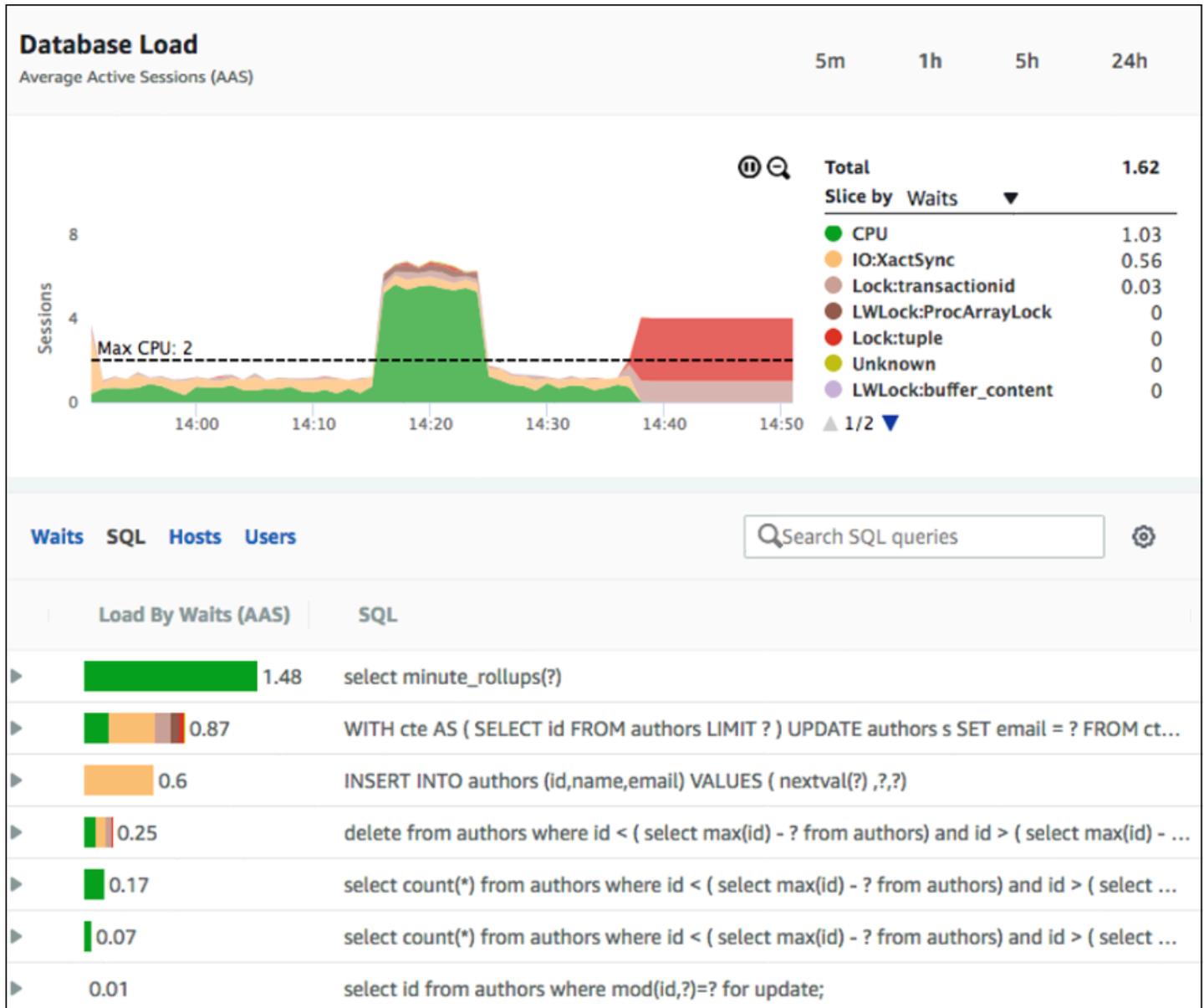
 Note

Enhanced Monitoring saat ini tidak mendukung database Oracle di Amazon. Untuk database ini, Anda dapat menggunakan solusi mitra pihak ketiga atau solusi asli seperti Oracle Enterprise Manager, seperti yang dibahas di bagian [selanjutnya](#).

Wawasan Performa

Amazon CloudWatch dan Amazon RDS Enhanced Monitoring adalah alat yang hebat untuk pemantauan tingkat instans dan tingkat OS. Namun, alat ini tidak menyediakan kemampuan diagnostik kinerja mendalam tingkat mesin basis data. Metrik mesin database membantu DBAs mengidentifikasi kemacetan database seperti kueri SQL intensif dan memvisualisasikan pemuatan basis data dengan jelas dari waktu ke waktu. Di Amazon RDS dan Amazon Aurora, dasbor Performance Insights menampilkan pemuatan database dengan menggunakan metrik bernama average active session (AAS).

Contoh berikut menunjukkan maksimum dua v CPUs dalam instans Amazon RDS yang dipantau. Namun, dua lonjakan utama melebihi jumlah v CPUs dan dapat menunjukkan kemacetan kinerja. Satu lonjakan mewakili beban CPU utama, ditampilkan dalam warna hijau, dan lonjakan lainnya mewakili hambatan pernyataan SQL utama, ditunjukkan dengan warna merah.



Performance Insights memberikan tingkat visibilitas tersebut dengan mengambil sampel setiap detik sesi database, mencari sesi aktif, dan mengabaikan sesi idle. Untuk setiap sesi aktif, Performance Insights mengumpulkan hal-hal berikut:

- Pernyataan SQL
- Tunggu acara seperti CPU, I/O, kunci, dan menunggu log komit
- Dimensi tambahan seperti host dan pengguna

Berdasarkan data ini, Anda dapat memvisualisasikan beban kerja database Anda dan memecahkan masalah kinerja dengan mudah. Anda juga dapat memfilter aktivitas berdasarkan berbagai dimensi

seperti host dan pengguna untuk analisis akar penyebab tambahan. Setiap mesin database memiliki serangkaian [dimensi yang didukung](#) sendiri.

Salah satu manfaat utama dari Performance Insights adalah tidak bergantung pada Oracle Diagnostics Pack, sehingga Anda dapat menggunakannya untuk memantau Oracle Database SE2 dan edisi non-Enterprise lainnya yang berjalan di Amazon RDS. [Untuk informasi selengkapnya, lihat bagian Performance Insights dari dokumentasi Amazon RDS dan Aurora.](#)

Note

Performance Insights saat ini tidak mendukung database Oracle di Amazon. EC2 Untuk database ini, Anda dapat menggunakan solusi mitra pihak ketiga atau solusi asli seperti Oracle Enterprise Manager, seperti yang dibahas di bagian berikutnya.

Oracle Enterprise Manager

Dalam beberapa kasus, pengguna Oracle Exadata mungkin lebih suka bekerja dengan Oracle Enterprise Manager (OEM). Amazon RDS mendukung OEM melalui opsi berikut:

Opsi	ID Opsi	Rilis OEM yang didukung	Rilis Oracle Database yang didukung
Database Ekspres OEM	OEM	OEM Database Express 12c	Oracle Database 19c (hanya non-CDB) dan Oracle Database 12c
Agen Manajemen OEM	OEM_AGENT	<ul style="list-style-type: none"> OEM Cloud Control untuk 13c OEM Cloud Control untuk 12c 	Oracle Database 19c (hanya non-CDB) dan Oracle Database 12c

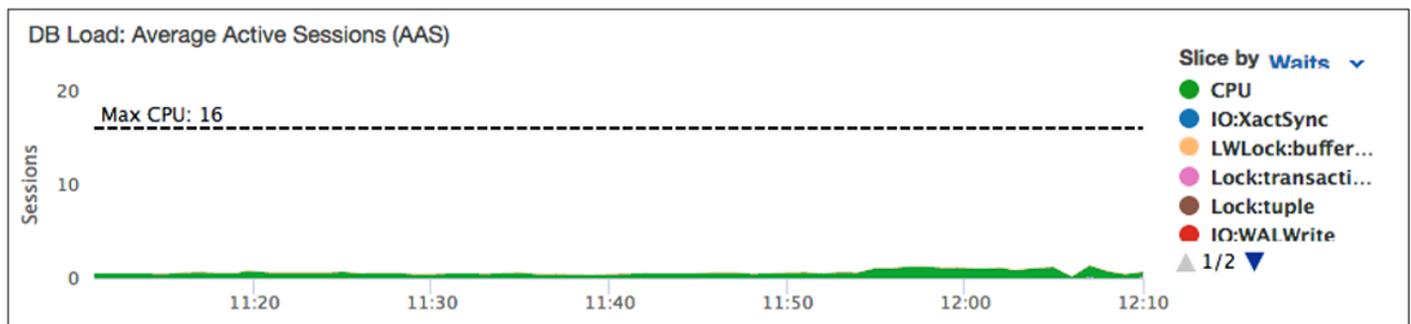
Optimalisasi biaya berkelanjutan

Ada berbagai praktik untuk mengoptimalkan biaya database. AWS Ini termasuk teknik seperti ukuran kanan instance, pindah ke Oracle Database, menggunakan instance cadangan SE2, menggunakan Amazon dengan prosesor Graviton2, dan mengoptimalkan pernyataan SQL.

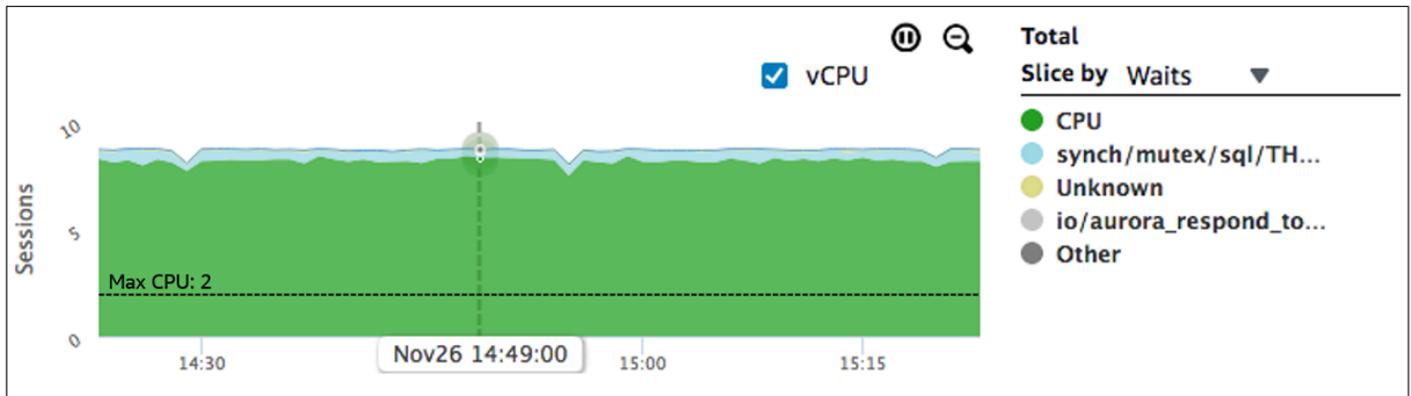
Ukuran instans Anda yang tepat

Ukuran kanan adalah proses memilih jenis instans dan penyimpanan yang memenuhi persyaratan kinerja dan kapasitas beban kerja spesifik Anda dengan biaya terendah. Ini adalah bagian penting dari mengoptimalkan AWS biaya.

Bagian sebelumnya mencakup Performance Insights, yang dapat Anda gunakan untuk diagnostik kinerja serta pengoptimalan ukuran dan biaya yang tepat. Misalnya, jika beban CPU secara signifikan lebih rendah daripada jumlah vCPUs, seperti yang ditunjukkan pada ilustrasi layar berikut, instance Anda terlalu besar dan Anda memiliki peluang penghematan biaya yang signifikan.



Di sisi lain, jika beban CPU secara signifikan lebih tinggi daripada jumlah vCPUs, instance Anda berukuran kecil, seperti yang ditunjukkan pada ilustrasi layar berikut. Dalam hal ini, Anda memiliki peluang optimasi kinerja yang memerlukan pengoptimalan pernyataan SQL untuk mengurangi rata-rata sesi aktif, atau pindah ke instance yang lebih besar yang dapat memenuhi persyaratan pemuatan.



Pertimbangkan untuk pindah ke Oracle Database SE2

Oracle Database Enterprise Edition (EE) telah menjadi standar bagi banyak organisasi. Namun, ketika Anda melakukan penilaian database yang mendalam, Anda mungkin menemukan bahwa aplikasi Anda mungkin tidak memerlukan semua fitur Oracle Database EE.

Oracle Database Standard Edition (SE) sekarang tersedia sebagai Oracle Database Standard Edition 2 (SE2) untuk Oracle 12c dan 19c. Oracle Database SE2 adalah sistem manajemen database relasional (RDBMS) yang mencakup fitur inti dari Oracle Database. Ini termasuk fitur yang dapat digunakan perusahaan untuk mendukung beban kerja kelas perusahaan. Mengingat fitur tambahan yang disediakan oleh Amazon RDS dan Amazon Aurora, yang tersedia untuk EE dan (seperti [Amazon RDS Multi-AZ SE2](#) dan [Amazon RDS untuk pencadangan otomatis lintas wilayah, enkripsi Amazon RDS](#) saat istirahat dan dalam perjalanan, dan aliran aktivitas basis data), Anda dapat mempertimbangkan untuk menggunakan untuk menghemat biaya. SE2

Dengan beralih ke SE2, Anda dapat mengoptimalkan penggunaan lisensi Oracle Database. Anda dapat menyediakan Oracle Database SE2 untuk digunakan dengan Amazon RDS dengan menggunakan opsi [Bring Your Own License \(BYOL\)](#) dan [Oracle License Included \(LI\)](#). Namun, sebelum Anda memutuskan perubahan besar seperti itu, sebaiknya Anda menilai fitur EE mana yang digunakan, fitur mana yang dapat diganti dengan menggunakan kemampuan Amazon RDS atau Aurora, dan fitur mana yang wajib dan tidak dapat diganti atau dihapus, yang mungkin mencegah Anda mengubah edisi basis data.

Untuk informasi selengkapnya, lihat [Mengevaluasi penurunan database Oracle ke Standard Edition 2 AWS](#) di situs web Prescriptive Guidance. AWS

Gunakan instans DB yang dicadangkan

Anda dapat menggunakan instans DB cadangan Amazon RDS untuk memesan instans DB untuk jangka waktu satu tahun atau tiga tahun, dan, pada gilirannya, menerima diskon yang signifikan dibandingkan dengan instans DB sesuai permintaan.

Anda dapat memilih di antara tiga opsi pembayaran saat membeli instans yang dipesan: Semua di muka, sebagian di muka, dan tidak ada di muka. Dengan opsi All Upfront, Anda membayar seluruh instans cadangan sebelum mulai menggunakannya. Opsi ini memberikan diskon terbesar dibandingkan dengan harga sesuai permintaan. Opsi Partial Upfront membutuhkan pembayaran di muka yang rendah dan tarif per jam diskon untuk instance selama jangka waktu. Opsi No Upfront memberikan tarif per jam diskon selama jangka waktu tanpa pembayaran di muka.

Jenis instans DB cadangan tersedia di Amazon RDS dan Aurora, untuk mesin database MySQL, MariaDB, PostgreSQL, Oracle, dan SQL Server.

Gunakan prosesor AWS Graviton

[Jika Anda bermigrasi dari Oracle Exadata ke salah satu database open source Amazon RDS dan Aurora, Anda bisa mendapatkan keuntungan dari kinerja biaya yang lebih baik dari prosesor Graviton2 dan Graviton3 untuk Amazon RDS. AWS](#)

Optimalkan kueri SQL Anda

Sebaiknya Anda memantau kinerja database dan mengidentifikasi pernyataan SQL teratas yang menggunakan sumber daya basis data yang signifikan—misalnya, dengan menggunakan Amazon RDS Performance Insights—secara teratur. Setelah Anda mengidentifikasi pernyataan SQL intensif sumber daya, terapkan praktik penyetelan SQL untuk meningkatkan kinerja database. Praktik penyetelan ini mencakup, namun tidak terbatas pada, operasi seperti pembuatan atau penghapusan indeks, penulisan ulang kueri SQL, pemodelan skema, dan fitur seperti tampilan terwujud.

Pengoptimalan SQL meningkatkan kinerja, yang menghasilkan waktu respons aplikasi yang lebih baik dan pengalaman pengguna yang lebih baik, dan menurunkan biaya database. Misalnya, kueri mungkin mengkonsumsi 60 persen dari beban database karena IOPS dan CPU yang tinggi terkait, yang mungkin memerlukan 200.000 IOPS (PIOPS) yang disediakan dan instance Amazon RDS () yang besar. `r5b.24x1` Dengan mengoptimalkan kueri — misalnya, dengan membuat indeks — Anda dapat mengukurnya dengan benar. Akibatnya, Anda mungkin dapat membayar lebih sedikit untuk instans Amazon RDS DB yang lebih kecil yang memiliki lebih sedikit PIOPS.

Pemantauan otomatis

Bagian ini membahas kemampuan otomatisasi utama untuk memantau beban kerja Exadata Anda di AWS.

CloudWatch Alarm Amazon dan deteksi anomali

Membuat alarm dan menjalankan tindakan alarm adalah praktik terbaik untuk pemantauan proaktif. Saat Anda mengatur alarm, pertanyaan umum adalah ambang batas untuk metrik yang ingin Anda pantau. Misalnya, Anda dapat membuat alarm yang diubah ke ALARM status ketika penggunaan CPU untuk sebuah instance melebihi ambang batas 70 persen.

Menentukan nilai ambang tidak selalu mudah, terutama karena banyak perusahaan memantau lusinan, terkadang ratusan, metrik di banyak instance database. Di sinilah deteksi CloudWatch anomali Amazon bisa berguna.

Saat Anda menggunakan deteksi anomali untuk metrik, CloudWatch terapkan algoritma statistik dan pembelajaran mesin (ML). Algoritma ini terus menganalisis metrik sistem dan aplikasi, menghasilkan berbagai nilai yang diharapkan yang mewakili perilaku metrik yang khas, dan anomali permukaan dengan intervensi pengguna minimal. Jenis-jenis alarm ini tidak memiliki ambang statis untuk menentukan status alarm. Alih-alih, jenis-jenis alarm tersebut membandingkan nilai metrik dengan nilai yang diharapkan berdasar model deteksi anomali. Anda dapat memilih apakah alarm merespons ketika nilai metrik berada di atas pita nilai yang diharapkan, di bawah pita, atau keduanya. [Untuk informasi selengkapnya tentang penggunaan deteksi anomali, lihat dokumentasi. CloudWatch](#)

Misalnya, Anda dapat menentukan alarm berdasarkan metrik RDS ReadIOPS untuk instans Amazon RDS for Oracle dengan menggunakan [CloudWatch wizard](#) dan memilih opsi deteksi anomali alih-alih opsi statis. Untuk petunjuk, lihat [CloudWatch dokumentasi Amazon](#).

Amazon DevOps Guru untuk Amazon RDS

Amazon DevOps Guru for Amazon RDS adalah kemampuan bertenaga ML yang membantu Anda mendeteksi, mendiagnosis, dan memperbaiki berbagai masalah terkait database dengan cepat. Ketika DevOps Guru for Amazon RDS secara otomatis mendeteksi masalah terkait database seperti pemanfaatan sumber daya yang berlebihan atau perilaku buruk kueri SQL, layanan segera memberi tahu Anda dan memberikan informasi diagnostik, detail tentang sejauh mana masalah, dan rekomendasi cerdas untuk membantu Anda menyelesaikan masalah dengan cepat.

Note

DevOpsGuru untuk Amazon RDS saat ini mendukung migrasi heterogen dari Oracle Exadata ke Amazon Aurora Edisi yang kompatibel dengan MySQL, Edisi yang kompatibel dengan Aurora PostgreSQL, dan Amazon RDS for PostgreSQL. Itu tidak mendukung database Oracle di Amazon, EC2 Amazon RDS, atau Aurora.

Misalnya, pertimbangkan toko buku online. Mari kita asumsikan bahwa situs web toko buku memiliki lonjakan konkurensi yang tinggi karena sejumlah besar pengguna ingin membeli buku setelah dipromosikan di TV. Setiap pembelian pelanggan mengurangi ketersediaan buku itu. Berikut adalah contoh pernyataan SQL yang berjalan di belakang layar setelah setiap pembelian:

```
update book_inventory
set available = available -1
where book_series =: series and book_title =: title;
```

Konkurensi tinggi dari banyak pernyataan DHTML mengakses baris yang sama pada saat yang sama dapat menghasilkan kunci tabel. Namun, Amazon CloudWatch tidak akan menampilkan lonjakan besar dalam beban CPU, karena kunci biasanya tidak mengkonsumsi sumber daya CPU yang signifikan. Dalam skenario ini, DevOps Guru dapat secara otomatis mengidentifikasi lonjakan yang tidak biasa dalam aktivitas database dengan melihat metrik sesi aktif rata-rata dan mendeteksi nilai yang menyimpang dari baseline tipikal.

Untuk informasi selengkapnya, lihat [Menganalisis anomali kinerja dengan Amazon DevOps Guru untuk Amazon RDS](#) di Amazon. RDSdocumentation

Audit otomatis

Menerapkan audit keamanan menjadi semakin penting karena persyaratan kepatuhan dan ancaman keamanan. Banyak pengguna lebih suka melanjutkan aktivitas audit yang mereka lakukan dengan Oracle di Exadata. AWS menyediakan dua opsi audit untuk database Anda: audit Amazon RDS dasar dan aliran aktivitas basis data.

Audit Amazon RDS dasar

Amazon RDS for Oracle menyediakan fitur audit berikut:

- Log dan **listener.log** file. Anda dapat mendorong file log penting ini secara otomatis ke Amazon CloudWatch untuk retensi dan analisis yang lebih lama.
- Audit standar. Anda dapat menggunakan fitur Oracle asli ini untuk mengaudit pernyataan SQL, hak istimewa, skema, objek, jaringan, dan aktivitas multi-tier. Oracle merekomendasikan penggunaan audit standar pada versi sebelum Oracle Database 12c rilis 1 (12.1). Audit standar bisa sulit dikelola karena beberapa jejak audit yang memiliki parameter berbeda untuk mengontrol perilaku audit dan kurangnya opsi audit terperinci.
- Audit terpadu. Oracle Database 12.1 dan versi yang lebih baru menawarkan audit terpadu. Fitur ini menyediakan data audit di satu lokasi dan dalam satu format. Amazon RDS for Oracle mendukung audit mode campuran, yang diaktifkan secara default untuk mendukung audit standar dan audit terpadu.

Aliran aktivitas basis data

Aliran aktivitas database menyediakan aliran data real-time dari semua aktivitas database. Fitur ini membantu perusahaan memantau, mengaudit, dan melindungi database dari akses yang tidak sah dan memenuhi persyaratan kepatuhan dan peraturan. Ini mengurangi pekerjaan yang diperlukan untuk memenuhi tujuan kepatuhan dan memfasilitasi migrasi ke layanan database terkelola di AWS. Aliran aktivitas database menyediakan data real-time yang terintegrasi ke dalam infrastruktur pemantauan dan peringatan yang ada, sehingga Anda dapat menggunakan proses, alat, dan laporan yang ada. Berikut adalah kasus penggunaan yang khas:

1. Berikan akses ke aplikasi Mitra untuk Amazon Kinesis Data AWS Key Management Service Streams AWS KMS dan () untuk memantau aktivitas database.
2. Hubungkan Amazon Kinesis Data Streams ke Amazon Data Firehose untuk menyimpan aktivitas ke Amazon S3 untuk retensi jangka panjang.
3. Connect AWS Lambda to untuk menganalisis atau memantau aktivitas database.

Note

Fitur aliran aktivitas database tersedia di Amazon RDS dan Amazon Aurora. Ini mendukung skenario migrasi database heterogen dan homogen.

Ringkasan

Untuk membangun aplikasi modern dan memaksimalkan kelincahan bisnis dan penghematan biaya, Anda memerlukan infrastruktur data yang dapat memenuhi kebutuhan unik aplikasi Anda dan layanan mikro-nya. Saat memodernisasi aplikasi, sebaiknya pertimbangkan faktor-faktor seperti persyaratan sumber daya, penggunaan fitur, serta kebutuhan pemantauan dan audit sebelum menentukan jalur migrasi target.

Panduan ini mencakup aspek-aspek kunci dari proyek Exadata ke AWS migrasi, termasuk penemuan pra-migrasi, melakukan migrasi, dan memelihara lingkungan database yang andal, sangat tersedia, hemat kinerja, dan dioptimalkan biaya setelah migrasi. Untuk memulai perjalanan modernisasi, hubungi [tim AWS akun](#) untuk menyiapkan sesi penemuan gratis.

Sumber daya

Bagian ini merangkum AWS alat, program, dan sumber daya lain yang dapat membantu migrasi Anda dari Oracle Exadata ke AWS

Alat dan layanan

- [AWS Database Migration Service\(AWS DMS\)](#) membantu Anda memigrasikan database Anda dengan cepat dan aman. AWS Database sumber Anda tetap beroperasi penuh selama migrasi, yang meminimalkan waktu henti untuk aplikasi yang bergantung pada database. AWS DMS mendukung basis data komersial dan sumber terbuka yang banyak digunakan, termasuk migrasi homogen, seperti Oracle Database lokal ke Oracle Database di cloud, dan migrasi heterogen antara platform database yang berbeda, seperti Oracle Database atau Microsoft SQL Server ke Amazon Aurora. Anda juga dapat menggunakan AWS DMS untuk terus mereplikasi data dengan latensi rendah dari sumber yang didukung ke target yang didukung. Misalnya, Anda dapat mereplikasi data dari berbagai sumber ke Amazon S3 untuk membuat solusi data lake yang sangat tersedia dan dapat diskalakan. Anda juga dapat mengkonsolidasikan database ke dalam gudang data skala petabyte dengan mengalirkan data ke Amazon Redshift. AWS DMS dapat sangat berguna jika Anda memerlukan waktu henti minimum selama migrasi, yang biasanya melibatkan solusi pengambilan data perubahan (CDC). AWS DMS memiliki keunggulan dibandingkan solusi CDC lainnya seperti Oracle GoldenGate, karena merupakan layanan asli AWS. Ini juga hemat biaya: Biaya Anda terbatas pada instans EC2 yang mendasari yang menjalankan instance AWS DMS replikasi, dan mungkin biaya penyimpanan dan transfer data tambahan. Selain itu, karena AWS DMS merupakan layanan yang dikelola sepenuhnya, persyaratan sumber daya dan biaya operasional yang terkait dengannya minimal dibandingkan dengan sebagian besar solusi migrasi dan replikasi data lainnya.
- [AWS Schema Conversion Tool\(AWS SCT\)](#) menawarkan migrasi database heterogen yang dapat diprediksi. Ini secara otomatis mengubah skema database sumber dan sebagian besar objek kode database, termasuk tampilan, prosedur tersimpan, dan fungsi, ke format yang kompatibel dengan database target. Objek apa pun yang tidak dapat dikonversi secara otomatis ditandai untuk konversi manual. AWS SCT juga dapat memindai kode sumber aplikasi untuk pernyataan SQL tertanam dan mengonversinya sebagai bagian dari proyek konversi skema database. Selama proses ini, AWS SCT lakukan pengoptimalan kode cloud-native dengan mengonversi fungsi Oracle dan SQL Server lama menjadi setara AWS layanan mereka, yang membantu Anda memodernisasi aplikasi Anda.

Program

- [AWS Migration Acceleration Program \(MAP\)](#) adalah program migrasi cloud komprehensif yang didasarkan pada pengalaman kami dalam AWS memigrasikan ribuan pelanggan perusahaan ke perusahaan. AWS Cloud Migrasi perusahaan bisa rumit dan memakan waktu, tetapi MAP dapat membantu mempercepat migrasi cloud dan tugas modernisasi Anda dengan menggunakan metodologi berbasis hasil. MAP menyediakan alat yang mengurangi biaya dan mengotomatiskan serta mempercepat implementasi, pendekatan dan konten pelatihan yang disesuaikan, keahlian dari Layanan AWS Profesional, komunitas AWS Mitra global, dan AWS investasi. MAP juga menggunakan kerangka kerja tiga tahap yang terbukti (menilai, memobilisasi, dan bermigrasi dan memodernisasi) untuk membantu perusahaan mencapai tujuan migrasi.
- [AWS Optimization and Licensing Assessment \(AWSOLA\)](#) membantu Anda menghemat biaya lisensi pihak ketiga dan menjalankan sumber daya dengan lebih efisien. AWS OLA adalah program gratis bagi pelanggan baru dan yang sudah ada untuk menilai dan mengoptimalkan lingkungan lokal dan cloud saat ini, berdasarkan pemanfaatan sumber daya aktual, lisensi pihak ketiga, dan dependensi aplikasi. Gunakan AWS OLA untuk membangun strategi migrasi dan lisensi Anda. AWS Program ini menyediakan laporan yang memodelkan opsi penerapan dengan menggunakan hak lisensi yang ada. Hasil ini dapat membantu Anda menjelajahi penghematan biaya yang tersedia di seluruh opsi lisensi yang fleksibel.
- [Amazon Database Migration Accelerator \(DMA\)](#) menyatukan AWS DMS, AWS SCT, dan pakar AWS database untuk membantu pelanggan bermigrasi dari database komersial tradisional dan layanan analitik. Program ini menawarkan layanan konsultasi migrasi, seperti membuat strategi migrasi, solusi, dan rencana implementasi, atau membuka blokir migrasi yang terhenti atau tertunda yang sedang berlangsung. Amazon DMA telah mendukung ribuan pelanggan, termasuk [BMC Software](#) dan [Thomson Reuters](#), dengan memodernisasi database mereka ke Amazon Aurora, Amazon RDS for PostgreSQL atau MySQL, Amazon Redshift, Amazon DynamoDB, dan lainnya.

Studi kasus

- Posting AWS blog [EDF Menyelesaikan Migrasi Terobosan untuk Menjalankan Solusi Utilitas Oracle di Amazon RDS](#) menjelaskan bagaimana penyedia listrik EDF bermigrasi dari Oracle Exadata ke AWS. Ini memberikan contoh dunia nyata dari migrasi yang sukses yang menggunakan beberapa praktik dan alat terbaik yang tercakup dalam panduan ini.

AWSKonten Panduan Preskriptif

- [Memigrasi database Oracle ke AWS Cloud](#) menjelaskan opsi, alat, dan praktik terbaik untuk memigrasi database lokal Oracle Anda. AWS
- [Memigrasi database Oracle yang besar ke AWS lingkungan lintas platform](#) menjelaskan bagaimana Anda dapat mengurangi waktu henti migrasi untuk database Oracle yang lebih besar dari 100 TB dengan menggunakan, AWS Snowball dan AWS Direct Connect Amazon FSx dengan cadangan tambahan Oracle XTTS dan RMAN.
- [Mentransfer file dump Oracle Database dari lokasi untuk AWS menjelaskan cara memigrasi file dump Oracle Database dengan menggunakan tautan basis data AWS Amazon S3, Amazon EFS, dan Oracle.](#)
- [Memilih kemampuan DR untuk edisi standar Amazon RDS for Oracle dan SQL Server](#) membahas skenario pemulihan bencana (DR) aktif-aktif dan pasif aktif, serta manfaat dan batasan setiap opsi untuk edisi standar Amazon RDS for Oracle dan SQL Server.
- [Mengevaluasi penurunan database Oracle ke Standard Edition 2 pada AWS](#) memberikan panduan untuk menilai database Oracle Anda dan menentukan apakah mereka dapat diturunkan untuk mengurangi biaya lisensi Oracle.
- [Panduan prioritas untuk refactoring Microsoft SQL Server dan database Oracle membahas proses untuk mengidentifikasi database kandidat untuk pindah ke mesin open source seperti PostgreSQL dan MySQL AWS on. AWS](#)
- [Strategi migrasi untuk database relasional](#) berfokus pada strategi dan kerangka kerja untuk memigrasi database relasional lokal, seperti Oracle atau Microsoft SQL Server, ke. AWS
- Lihat juga: [Pola migrasi dan modernisasi untuk Oracle Database.](#)

Kontributor

Penulis dan rekan penulis berikut berkontribusi pada panduan ini:

- Pini Dibask, Arsitek Solusi Database Senior, AWS
- Tom Harper, Manajer Arsitek Solusi NoSQL, AWS
- Jobin Joseph, Senior Amazon RDS for Oracle Solutions Architect, AWS
- Marvin Vinson, Arsitek Solusi Basis Data Utama, AWS

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
Informasi terbaru tentang jenis volume Amazon EBS	Memperbarui bagian rekomendasi Replatforming dengan informasi tentang opsi penyimpanan io2 Block Express.	Juli 12, 2024
Informasi terbaru tentang konsolidasi basis data	Memperbarui bagian Konsolidasi database untuk mengklarifikasi bahwa Amazon RDS for Oracle sekarang mendukung arsitektur multitenant dengan beberapa database pluggable .	Februari 28, 2024
Publikasi awal	—	Januari 24, 2024

AWS Glosarium Panduan Preskriptif

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

Nomor

7 Rs

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

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

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

A

ABAC

Lihat [kontrol akses berbasis atribut](#).

layanan abstrak

Lihat [layanan terkelola](#).

ASAM

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

migrasi aktif-aktif

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

migrasi aktif-pasif

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

fungsi agregat

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

AI

Lihat [kecerdasan buatan](#).

AIOps

Lihat [operasi kecerdasan buatan](#).

anonimisasi

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

anti-pola

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

kontrol aplikasi

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

portofolio aplikasi

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

kecerdasan buatan (AI)

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

operasi kecerdasan buatan (AIOps)

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

enkripsi asimetris

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

atomisitas, konsistensi, isolasi, daya tahan (ACID)

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

kontrol akses berbasis atribut (ABAC)

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

sumber data otoritatif

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

Zona Ketersediaan

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

AWS Kerangka Adopsi Cloud (AWS CAF)

Kerangka pedoman dan praktik terbaik AWS untuk membantu organisasi mengembangkan rencana yang efisien dan efektif untuk bergerak dengan sukses ke cloud. AWS CAF mengatur panduan ke dalam enam area fokus yang disebut perspektif: bisnis, orang, tata kelola, platform, keamanan, dan operasi. Perspektif bisnis, orang, dan tata kelola fokus pada keterampilan dan proses bisnis; perspektif platform, keamanan, dan operasi fokus pada keterampilan dan proses teknis. Misalnya, perspektif masyarakat menargetkan pemangku kepentingan yang menangani sumber daya manusia (SDM), fungsi kepegawaian, dan manajemen orang. Untuk perspektif ini, AWS CAF memberikan panduan untuk pengembangan, pelatihan, dan komunikasi orang untuk membantu mempersiapkan organisasi untuk adopsi cloud yang sukses. Untuk informasi lebih lanjut, lihat [situs web AWS CAF dan whitepaper AWS CAF](#).

AWS Kerangka Kualifikasi Beban Kerja (AWS WQF)

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

B

bot buruk

[Bot](#) yang dimaksudkan untuk mengganggu atau membahayakan individu atau organisasi.

BCP

Lihat [perencanaan kontinuitas bisnis](#).

grafik perilaku

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

sistem big-endian

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

klasifikasi biner

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

filter mekar

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

deployment biru/hijau

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

bot

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

botnet

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

cabang

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

akses break-glass

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

strategi brownfield

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

cache penyangga

Area memori tempat data yang paling sering diakses disimpan.

kemampuan bisnis

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

perencanaan kelangsungan bisnis (BCP)

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

C

KAFE

Lihat [Kerangka Adopsi AWS Cloud](#).

penyebaran kenari

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

CCoE

Lihat [Cloud Center of Excellence](#).

CDC

Lihat [mengubah pengambilan data](#).

ubah pengambilan data (CDC)

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

rekayasa kecacauan

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

CI/CD

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

klasifikasi

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

Enkripsi sisi klien

Enkripsi data secara lokal, sebelum target Layanan AWS menerimanya.

Pusat Keunggulan Cloud (CCoE)

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

komputasi cloud

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

model operasi cloud

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

tahap adopsi cloud

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

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

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

CMDB

Lihat [database manajemen konfigurasi](#).

repositori kode

Lokasi di mana kode sumber dan aset lainnya, seperti dokumentasi, sampel, dan skrip, disimpan dan diperbarui melalui proses kontrol versi. Repositori cloud umum termasuk GitHub atau Bitbucket Cloud Setiap versi kode disebut cabang. Dalam struktur layanan mikro, setiap repositori

dikhususkan untuk satu bagian fungsionalitas. Pipa CI/CD tunggal dapat menggunakan beberapa repositori.

cache dingin

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

data dingin

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

visi komputer (CV)

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

konfigurasi drift

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

database manajemen konfigurasi (CMDB)

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

paket kesesuaian

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

integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD)

Proses mengotomatiskan sumber, membangun, menguji, pementasan, dan tahap produksi dari proses rilis perangkat lunak. CI/CD is commonly described as a pipeline. CI/CD dapat membantu

Anda mengotomatiskan proses, meningkatkan produktivitas, meningkatkan kualitas kode, dan memberikan lebih cepat. Untuk informasi lebih lanjut, lihat [Manfaat pengiriman berkelanjutan](#). CD juga dapat berarti penerapan berkelanjutan. Untuk informasi selengkapnya, lihat [Continuous Delivery vs Continuous Deployment](#).

CV

Lihat [visi komputer](#).

D

data saat istirahat

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

klasifikasi data

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

penyimpangan data

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

data dalam transit

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

jala data

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

minimalisasi data

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

perimeter data

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

prapemrosesan data

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

asal data

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

subjek data

Individu yang datanya dikumpulkan dan diproses.

gudang data

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

bahasa definisi database (DDL)

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

bahasa manipulasi basis data (DHTML)

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

DDL

Lihat [bahasa definisi database](#).

ansambel yang dalam

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

pembelajaran mendalam

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

defense-in-depth

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

administrator yang didelegasikan

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

deployment

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

lingkungan pengembangan

Lihat [lingkungan](#).

kontrol detektif

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

pemetaan aliran nilai pengembangan (DVSM)

Sebuah proses yang digunakan untuk mengidentifikasi dan memprioritaskan kendala yang mempengaruhi kecepatan dan kualitas dalam siklus hidup pengembangan perangkat lunak. DVSM memperluas proses pemetaan aliran nilai yang awalnya dirancang untuk praktik

manufaktur ramping. Ini berfokus pada langkah-langkah dan tim yang diperlukan untuk menciptakan dan memindahkan nilai melalui proses pengembangan perangkat lunak.

kembar digital

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

tabel dimensi

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

musibah

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

pemulihan bencana (DR)

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

DML~

Lihat [bahasa manipulasi basis data](#).

desain berbasis domain

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

DR

Lihat [pemulihan bencana](#).

deteksi drift

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

DVSM

Lihat [pemetaan aliran nilai pengembangan](#).

E

EDA

Lihat [analisis data eksplorasi](#).

EDI

Lihat [pertukaran data elektronik](#).

komputasi tepi

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

pertukaran data elektronik (EDI)

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

enkripsi

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

kunci enkripsi

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

endianness

Urutan byte disimpan dalam memori komputer. Sistem big-endian menyimpan byte paling signifikan terlebih dahulu. Sistem little-endian menyimpan byte paling tidak signifikan terlebih dahulu.

titik akhir

Lihat [titik akhir layanan](#).

layanan endpoint

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

perencanaan sumber daya perusahaan (ERP)

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

enkripsi amplop

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

lingkungan

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

- Development Environment — Sebuah contoh dari aplikasi yang berjalan yang hanya tersedia untuk tim inti yang bertanggung jawab untuk memelihara aplikasi. Lingkungan pengembangan digunakan untuk menguji perubahan sebelum mempromosikannya ke lingkungan atas. Jenis lingkungan ini kadang-kadang disebut sebagai lingkungan pengujian.
- lingkungan yang lebih rendah — Semua lingkungan pengembangan untuk aplikasi, seperti yang digunakan untuk build awal dan pengujian.
- lingkungan produksi — Sebuah contoh dari aplikasi yang berjalan yang 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](#).

ERP

Lihat [perencanaan sumber daya perusahaan](#).

analisis data eksplorasi (EDA)

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

F

tabel fakta

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

gagal cepat

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

batas isolasi kesalahan

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

cabang fitur

Lihat [cabang](#).

fitur

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

pentingnya fitur

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

transformasi fitur

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

beberapa tembakan mendorong

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

FGAC

Lihat kontrol [akses berbutir halus](#).

kontrol akses berbutir halus (FGAC)

Penggunaan beberapa kondisi untuk mengizinkan atau menolak permintaan akses.

migrasi flash-cut

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

FM

Lihat [model pondasi](#).

model pondasi (FM)

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

G

AI generatif

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

pemblokiran geografis

Lihat [pembatasan geografis](#).

pembatasan geografis (pemblokiran geografis)

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

Alur kerja Gitflow

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

gambar emas

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

strategi greenfield

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

pagar pembatas

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

H

HA

Lihat [ketersediaan tinggi](#).

migrasi database heterogen

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

ketersediaan tinggi (HA)

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

modernisasi sejarawan

Pendekatan yang digunakan untuk memodernisasi dan meningkatkan sistem teknologi operasional (OT) untuk melayani kebutuhan industri manufaktur dengan lebih baik. Sejarawan

adalah jenis database yang digunakan untuk mengumpulkan dan menyimpan data dari berbagai sumber di pabrik.

data penahanan

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

migrasi database homogen

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

data panas

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

perbaikan terbaru

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

periode hypercare

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

|

IAC

Lihat [infrastruktur sebagai kode](#).

kebijakan berbasis identitas

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

|

aplikasi idle

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

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 inbound, outbound, dan inspeksi VPCs untuk melindungi antarmuka dua arah antara aplikasi Anda dan internet yang lebih luas.

migrasi inkremental

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

Industri 4.0

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

infrastruktur

Semua sumber daya dan aset yang terkandung dalam lingkungan aplikasi.

infrastruktur sebagai kode (IAC)

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

Internet of Things industri (IIoT)

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

inspeksi VPC

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

Internet of Things (IoT)

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

interpretabilitas

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

IoT

Lihat [Internet of Things](#).

Perpustakaan informasi TI (ITIL)

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

Manajemen layanan TI (ITSM)

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

ITIL

Lihat [perpustakaan informasi TI](#).

ITSM

Lihat [manajemen layanan TI](#).

L

kontrol akses berbasis label (LBAC)

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

landing zone

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

model bahasa besar (LLM)

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

migrasi besar

Migrasi 300 atau lebih server.

LBAC

Lihat [kontrol akses berbasis label](#).

hak istimewa paling sedikit

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

angkat dan geser

Lihat [7 Rs](#).

sistem endian kecil

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

LLM

Lihat [model bahasa besar](#).

lingkungan yang lebih rendah

Lihat [lingkungan](#).

M

pembelajaran mesin (ML)

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

cabang utama

Lihat [cabang](#).

malware

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

layanan terkelola

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

sistem eksekusi manufaktur (MES)

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

PETA

Lihat [Program Percepatan Migrasi](#).

mekanisme

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

akun anggota

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

MES

Lihat [sistem eksekusi manufaktur](#).

Transportasi Telemetri Antrian Pesan (MQTT)

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

layanan mikro

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

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

arsitektur microservices

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

Program Percepatan Migrasi (MAP)

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

migrasi dalam skala

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

pabrik migrasi

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

metadata migrasi

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

pola migrasi

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

Penilaian Portofolio Migrasi (MPA)

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

Penilaian Kesiapan Migrasi (MRA)

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

strategi migrasi

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

ML

Lihat [pembelajaran mesin](#).

modernisasi

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

penilaian kesiapan modernisasi

Evaluasi yang membantu menentukan kesiapan modernisasi aplikasi organisasi; mengidentifikasi manfaat, risiko, dan dependensi; dan menentukan seberapa baik organisasi dapat mendukung keadaan masa depan aplikasi tersebut. Hasil penilaian adalah cetak biru arsitektur target, peta

jalan yang merinci fase pengembangan dan tonggak untuk proses modernisasi, dan rencana aksi untuk mengatasi kesenjangan yang diidentifikasi. Untuk informasi lebih lanjut, lihat [Mengevaluasi kesiapan modernisasi untuk](#) aplikasi di. AWS Cloud

aplikasi monolitik (monolit)

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

MPA

Lihat [Penilaian Portofolio Migrasi](#).

MQTT

Lihat [Transportasi Telemetri Antrian Pesan](#).

klasifikasi multiclass

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

infrastruktur yang bisa berubah

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

O

OAC

Lihat [kontrol akses asal](#).

OAI

Lihat [identitas akses asal](#).

OCM

Lihat [manajemen perubahan organisasi](#).

migrasi offline

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

OI

Lihat [integrasi operasi](#).

OLA

Lihat [perjanjian tingkat operasional](#).

migrasi online

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

OPC-UA

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

Komunikasi Proses Terbuka - Arsitektur Terpadu (OPC-UA)

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

perjanjian tingkat operasional (OLA)

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

Tinjauan Kesiapan Operasional (ORR)

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

teknologi operasional (OT)

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

integrasi operasi (OI)

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

jejak organisasi

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

manajemen perubahan organisasi (OCM)

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

kontrol akses asal (OAC)

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

identitas akses asal (OAI)

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

ORR

Lihat [tinjauan kesiapan operasional](#).

OT

Lihat [teknologi operasional](#).

keluar (jalan keluar) VPC

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

P

batas izin

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

Informasi Identifikasi Pribadi (PII)

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

PII

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

buku pedoman

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

PLC

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

PLM

Lihat [manajemen siklus hidup produk](#).

kebijakan

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

ketekunan poliglot

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

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

privasi berdasarkan desain

Pendekatan rekayasa sistem yang memperhitungkan privasi melalui seluruh proses pengembangan.

zona yang dihosting pribadi

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

kontrol proaktif

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

manajemen siklus hidup produk (PLM)

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

lingkungan produksi

Lihat [lingkungan](#).

pengontrol logika yang dapat diprogram (PLC)

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

rantai cepat

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

pseudonimisasi

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

publish/subscribe (pub/sub)

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

Q

rencana kueri

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

regresi rencana kueri

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

R

Matriks RACI

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

LAP

Lihat [Retrieval Augmented Generation](#).

ransomware

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

Matriks RASCI

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

RCAC

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

replika baca

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

arsitek ulang

Lihat [7 Rs](#).

tujuan titik pemulihan (RPO)

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

tujuan waktu pemulihan (RTO)

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

refactor

Lihat [7 Rs](#).

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 [Menentukan Wilayah AWS akun yang dapat digunakan](#).

regresi

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

rehost

Lihat [7 Rs](#).

melepaskan

Dalam proses penyebaran, tindakan mempromosikan perubahan pada lingkungan produksi.

memindahkan

Lihat [7 Rs](#).

memplatform ulang

Lihat [7 Rs](#).

pembelian kembali

Lihat [7 Rs](#).

ketahanan

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

kebijakan berbasis sumber daya

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

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

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

kontrol responsif

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

melestarikan

Lihat [7 Rs](#).

pensiun

Lihat [7 Rs](#).

Retrieval Augmented Generation (RAG)

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

rotasi

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

kontrol akses baris dan kolom (RCAC)

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

RPO

Lihat [tujuan titik pemulihan](#).

RTO

Lihat [tujuan waktu pemulihan](#).

buku runbook

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

D

SAML 2.0

Standar terbuka yang digunakan oleh banyak penyedia identitas (IdPs). Fitur ini memungkinkan sistem masuk tunggal gabungan (SSO), sehingga pengguna dapat masuk ke 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.

SCADA

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

SCP

Lihat [kebijakan kontrol layanan](#).

Rahasia

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

keamanan dengan desain

Pendekatan rekayasa sistem yang memperhitungkan keamanan melalui seluruh proses pengembangan.

kontrol keamanan

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

pengerasan keamanan

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

sistem informasi keamanan dan manajemen acara (SIEM)

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

otomatisasi respons keamanan

Tindakan yang telah ditentukan dan diprogram yang dirancang untuk secara otomatis merespons atau memulihkan peristiwa keamanan. Otomatisasi ini berfungsi sebagai kontrol keamanan

[detektif](#) atau [responsif](#) yang membantu Anda menerapkan praktik terbaik AWS keamanan. Contoh tindakan respons otomatis termasuk memodifikasi grup keamanan VPC, menambal instans EC2 Amazon, atau memutar kredensial.

enkripsi sisi server

Enkripsi data di tujuannya, oleh Layanan AWS yang menerimanya.

kebijakan kontrol layanan (SCP)

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

titik akhir layanan

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

perjanjian tingkat layanan (SLA)

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

indikator tingkat layanan (SLI)

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

tujuan tingkat layanan (SLO)

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

model tanggung jawab bersama

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

SIEM

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

titik kegagalan tunggal (SPOF)

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

SLA

Lihat [perjanjian tingkat layanan](#).

SLI

Lihat [indikator tingkat layanan](#).

SLO

Lihat [tujuan tingkat layanan](#).

split-and-seed model

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

SPOF

Lihat [satu titik kegagalan](#).

skema bintang

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

pola ara pencekik

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

subnet

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

kontrol pengawasan dan akuisisi data (SCADA)

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

enkripsi simetris

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

pengujian sintetis

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

sistem prompt

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

T

tag

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

alur kerja berbasis batang

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

akses tepercaya

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

penyetelan

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

tim dua pizza

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

U

waswas

Sebuah konsep yang mengacu pada informasi yang tidak tepat, tidak lengkap, atau tidak diketahui yang dapat merusak keandalan model ML prediktif. Ada dua jenis ketidakpastian: ketidakpastian epistemik disebabkan oleh data yang terbatas dan tidak lengkap, sedangkan ketidakpastian aleatorik disebabkan oleh kebisingan dan keacakan yang melekat dalam data. Untuk informasi lebih lanjut, lihat panduan [Mengukur ketidakpastian dalam sistem pembelajaran mendalam](#).

tugas yang tidak terdiferensiasi

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

lingkungan atas

Lihat [lingkungan](#).

V

menyedot debu

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

kendali versi

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

Peering VPC

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

kerentanan

Kelemahan perangkat lunak atau perangkat keras yang membahayakan keamanan sistem.

W

cache hangat

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

tulis sekali, baca banyak (WORM)

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

Z

eksploitasi zero-day

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

kerentanan zero-day

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

bisikan zero-shot

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

aplikasi zombie

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

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