



Opsi, alat, dan praktik terbaik untuk memigrasikan beban kerja Microsoft ke AWS

# AWS Panduan Preskriptif



# AWS Panduan Preskriptif: Opsi, alat, dan praktik terbaik untuk memigrasikan beban kerja Microsoft ke AWS

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Merek dagang dan tampilan dagang Amazon tidak boleh digunakan sehubungan dengan produk atau layanan apa pun yang bukan milik Amazon, dengan cara apa pun yang dapat menyebabkan kebingungan di antara 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
Gambaran Umum .....	1
Audiens yang dituju .....	2
Hasil bisnis yang ditargetkan .....	2
Mengapa memilih beban kerja AWS untuk Microsoft? .....	3
Praktik terbaik dasar .....	5
Jalur ke awan .....	7
Strategi migrasi .....	7
Transformasi utama .....	7
Memilih strategi migrasi .....	8
Proses migrasi Windows .....	10
Menilai .....	10
Memobilisasi .....	11
Migrasi dan modernisasi .....	11
Penemuan lingkungan Windows .....	13
Menilai .....	13
Arsitektur perusahaan .....	13
Standardisasi dan manajemen konfigurasi .....	13
Data yang bagus .....	14
Otomatisasi .....	14
Perencanaan terperinci .....	14
Memobilisasi .....	15
Tantangan migrasi dalam skala besar .....	15
Dependensi yang sensitif terhadap latensi .....	15
Layanan bersama TI .....	16
Pembaruan konfigurasi .....	16
Pengujian fungsional aplikasi .....	16
Alat untuk penemuan ketergantungan aplikasi .....	17
Memigrasi beban kerja Microsoft .....	18
Migrasi Direktori Aktif .....	18
Menilai .....	18
Memobilisasi .....	20
Migrasi .....	24
Migrasi Windows Server .....	27

---

Menilai .....	27
Memobilisasi .....	27
Migrasi .....	28
Migrasi server file .....	29
Menilai .....	29
Memobilisasi .....	31
Migrasi .....	32
Migrasi SQL Server .....	33
Menilai .....	33
Memobilisasi .....	34
Migrasi .....	35
Migrasi aplikasi.NET .....	39
Menilai .....	39
Memobilisasi .....	40
Migrasi .....	41
Platform Ulang .....	42
Sumber daya tambahan .....	46
Migrasi cluster failover Windows .....	46
Menilai .....	47
Memobilisasi .....	49
Migrasi .....	50
Memantau beban kerja Microsoft .....	51
Menilai .....	51
Memobilisasi .....	51
Migrasi .....	53
Alat migrasi, program, dan pelatihan .....	54
Alat .....	54
Alat penilaian .....	54
Alat Migrasi .....	57
Alat Mitra Migrasi .....	59
Alat manajemen .....	60
Program .....	61
Pelatihan .....	63
Lisensi Microsoft di AWS .....	65
Menilai .....	65
Lisensi termasuk pilihan .....	66

---

---

Opsi BYOL .....	67
Host Khusus Amazon EC2 .....	71
VMware Cloud on AWS .....	73
Memobilisasi .....	73
AWS License Manager .....	73
Pertimbangan perizinan .....	74
Migrasi .....	74
Mitra AWS .....	76
Manfaat melibatkan Partner Kompetensi AWS .....	76
Membangun rencana .....	76
Optimalkan biaya .....	76
Hemat waktu .....	77
Tingkatkan keamanan .....	78
Langkah selanjutnya .....	79
Sumber daya .....	80
Riwayat dokumen .....	81
Glosarium .....	83
# .....	83
A .....	84
B .....	87
C .....	89
D .....	92
E .....	96
F .....	98
G .....	99
H .....	100
I .....	101
L .....	104
M .....	105
O .....	109
P .....	112
Q .....	115
R .....	115
D .....	118
T .....	122
U .....	123

---

---

V .....	124
W .....	124
Z .....	125
.....	CXXvi

# Opsi, alat, dan praktik terbaik untuk memigrasikan beban kerja Microsoft ke AWS

Jerroll Harewood, Christine Megit, Pembantu Dror, Daniel Maldonado, Phil Ekins, Mani Pachnanda, Siddharth Mehta, Rich Benoit, Rob Higareda, Saleha Haider, Siavash Irani, dan Yogi Barot, Amazon Web Services (AWS)

April 2024 ([riwayat dokumen](#))

## Gambaran Umum

Organizations telah bermigrasi dan menjalankan beban kerja Microsoft mereka di AWS selama lebih dari satu dekade—lebih lama daripada penyedia cloud lainnya. Berdasarkan pengetahuan dan keahlian yang diperoleh AWS dari upaya migrasi dan modernisasi selama bertahun-tahun, panduan ini dirancang untuk merampingkan migrasi beban kerja Microsoft Anda ke AWS Cloud. Anda dapat menggunakan panduan ini untuk merencanakan dan mengimplementasikan semua fase migrasi Windows Anda. Panduan ini berlaku untuk berbagai kasus penggunaan migrasi, termasuk yang berikut:

- Anda memulai migrasi Windows sebagai bagian dari transformasi digital dan perjalanan modernisasi di organisasi Anda.
- Sewa di pusat data tempat Anda menjalankan beban kerja Microsoft hampir kedaluwarsa.
- Anda memiliki berbagai aplikasi Windows dengan persyaratan ketersediaan yang bervariasi, tetapi Anda tidak memiliki sumber daya untuk menyebarkan beban kerja Anda di seluruh lokasi yang didistribusikan secara geografis.

Dalam panduan ini, Anda mempelajari berbagai alat AWS yang dapat membantu merampingkan perjalanan migrasi Anda, seperti AWS Migration Hub, AWS Application Migration Service, dan banyak lagi. Untuk menyelaraskan dengan praktik terbaik AWS, panduan ini mengikuti [proses migrasi AWS tiga fase](#): menilai, memobilisasi, dan memigrasikan serta memodernisasi. Proses ini didasarkan pada kerangka kerja migrasi yang telah teruji waktu yang dapat membantu Anda menyusun dan merampingkan migrasi Windows Anda. Pada fase penilaian, Anda mengevaluasi kesiapan Anda untuk beroperasi di cloud. Pada fase mobilisasi, Anda menyusun rencana migrasi dan menutup kesenjangan kesiapan yang diidentifikasi dalam fase penilaian. Kemudian, Anda mulai memigrasikan beban kerja Anda dalam fase migrasi dan modernisasi dengan menggunakan kombinasi alat

otomatisasi dan templat untuk memigrasikan beban kerja Anda secara sistematis dan memenuhi persyaratan bisnis Anda.

## Audiens yang dituju

Panduan ini ditujukan untuk arsitek TI, prospek migrasi, prospek teknis, tim Partner AWS, dan peran lain yang bertanggung jawab atas hal-hal berikut:

- Memigrasi beban kerja Microsoft dari pusat data ke AWS Cloud
- Mengelola lingkungan Windows di AWS Cloud

## Hasil bisnis yang ditargetkan

Panduan ini dapat membantu Anda dan organisasi Anda mencapai tujuan berikut:

1. Pelajari tentang strategi, program, dan layanan yang tersedia untuk memigrasikan beban kerja Microsoft ke AWS.
2. Memahami jalur migrasi AWS untuk beban kerja Microsoft tertentu, seperti Active Directory, Windows File Server, SQL Server, dan beban kerja.NET.
3. Jalankan beban kerja Microsoft Anda di AWS sambil memenuhi persyaratan keamanan, ketersediaan, dan keandalan Anda.
4. Biasakan diri Anda dengan praktik terbaik lisensi untuk menjalankan beban kerja Microsoft di AWS.



# Mengapa memilih beban kerja AWS untuk Microsoft?

AWS telah membantu pelanggan memigrasi dan memodernisasi beban kerja Microsoft mereka selama lebih dari 14 tahun dan memiliki portofolio layanan, program, dan keahlian terluas untuk mempercepat transformasi aplikasi utama yang mendukung bisnis. Jika Anda menggunakan AWS untuk memigrasi dan memodernisasi, Anda dapat menantikan manfaat berikut:

- **Buka kunci inovasi**— Berpindah dari arsitektur monolitik tradisional ke arsitektur layanan mikro berbasis cloud dapat memberi Anda kebebasan untuk beradaptasi dan bereksperimen dengan cepat sehingga organisasi Anda dapat membuka inovasi lebih cepat. AWS memiliki serangkaian teknologi kontainer terluas, termasuk Amazon Elastic Container Service (Amazon ECS), Amazon Elastic Kubernetes Service (Amazon EKS), dan AWS Fargate. Selain itu, AWS memiliki penawaran tanpa server (AWS Lambda) yang paling matang, dukungan .NET yang terintegrasi secara mendalam, DevOps utilitas untuk mengotomatiskan siklus pengembangan, beberapa integrasi sumber terbuka, dan database yang dibuat khusus seperti Amazon Aurora untuk mendukung arsitektur modern.
- **Mengurangi biaya**— Anda dapat menghindari pembayaran untuk lisensi Windows atau SQL Server yang mahal dengan pindah ke solusi database open-source. Misalnya, Aurora menyediakan fungsionalitas yang sama dengan database komersial dengan biaya sepersepuluh. Jika Anda pindah ke DevOps dan menggunakan kontainer dan solusi tanpa server, Anda dapat mengurangi total biaya kepemilikan (TCO) dan memaksimalkan konsumsi komputasi.
- **Tingkatkan keamanan**— AWS menawarkan 230 layanan keamanan, kepatuhan, dan tata kelola serta fitur utama—lima kali lebih banyak layanan daripada penyedia cloud terbesar berikutnya. Anda dapat menggunakan [Layanan Direktori AWS](#), juga dikenal sebagai AWS Managed Microsoft AD, untuk meningkatkan keamanan cloud Anda dan menghilangkan kebutuhan untuk menyinkronkan atau mereplikasi data dari Active Directory yang ada selama migrasi. Anda juga dapat menggunakan [Layanan Identitas AWS](#) untuk mengelola identitas dan izin dalam skala besar, sambil memberikan opsi fleksibel untuk di mana dan bagaimana Anda mengelola informasi karyawan, mitra, dan pelanggan Anda.
- **Kembangkan keterampilan dengan para ahli tepercaya**— AWS memiliki pengalaman tak tertandingi yang membantu jutaan organisasi mencapai tujuan migrasi mereka lebih cepat melalui alat dan layanan unik. Yang [Program Akselerasi Migrasi AWS \(MAP\) untuk Windows](#) memberikan praktik, alat, dan insentif terbaik untuk mengurangi kompleksitas dan biaya migrasi ke cloud dengan dukungan dari AWS Partners dan AWS Professional Services. Yang [Program Migrasi Akhir Dukungan untuk Windows Server](#) dapat membantu Anda memigrasikan aplikasi Windows Server

lama ke versi Windows Server terbaru yang didukung di AWS. 90 persen perusahaan Fortune 100 dan mayoritas perusahaan Fortune 500 menggunakan solusi dan layanan AWS Partner.

- Tingkatkan harga dan kinerja daya pemrosesan Anda— AWS adalah pemimpin dalam inovasi pemrosesan, menawarkan instans berbasis Graviton2 yang harganya 20 persen lebih murah per jam dibandingkan instans berbasis Intel x86, dengan kinerja hingga 40 persen lebih baik. Aurora juga membawa lima kali throughput MySQL standar dan tiga kali throughput PostgreSQL standar. Kinerja ini setara dengan database komersial, dengan biaya sepersepuluh.
- Manfaatkan opsi lisensi yang fleksibel— AWS menawarkan opsi terbanyak di cloud untuk menggunakan lisensi perangkat lunak Microsoft baru dan yang sudah ada di AWS. Jika Anda membeli instans Amazon Elastic Compute Cloud (Amazon EC2) atau Amazon Relational Database Service (Amazon RDS) yang disertakan dengan lisensi, Anda mendapatkan lisensi SQL Server baru yang sepenuhnya sesuai dari AWS. Anda dapat membawa lisensi yang ada ke AWS dengan [Host Khusus Amazon EC2](#), [Instans Khusus Amazon EC2](#), atau instans EC2 dengan sewa default dengan menggunakan [Mobilitas Lisensi Microsoft melalui Jaminan Perangkat Lunak](#). AWS License Manager memudahkan Anda melacak penggunaan lisensi perangkat lunak dan mengurangi risiko ketidakpatuhan.

Untuk informasi lebih lanjut, lihat [Windows di AWS](#) dalam dokumentasi AWS.

## Praktik terbaik dasar

Membangun fondasi yang dapat diskalakan dan aman untuk migrasi AWS Anda dapat memungkinkan Anda mengelola dan menjalankan lingkungan Windows dengan mudah di AWS. Sebelum memigrasikan beban kerja Microsoft ke AWS, sebaiknya Anda mempertimbangkan praktik terbaik dasar berikut:

- Optimalkan pengeluaran Anda untuk lisensi Microsoft — Lisensi adalah faktor penting dalam migrasi cloud Anda karena hal itu memengaruhi semua keputusan lain di masa depan. Kami menyarankan Anda memahami opsi lisensi sedini mungkin. Untuk informasi selengkapnya tentang lisensi, lihat bagian [Lisensi Microsoft di AWS](#) dari panduan ini.
- Merampingkan arsitektur cloud Anda — [AWS Well-Architected](#) Framework membantu Anda menjalankan beban kerja dengan andal di cloud. Anda menerima panduan dan strategi untuk membantu Anda mengikuti kerangka kerja, menghindari masalah serius, dan skala untuk memenuhi kebutuhan organisasi Anda. Panduan ini juga mencakup penagihan, kontrol akses, dan kontrol keamanan.
- Membangun jaringan easy-to-manage cloud yang terintegrasi — [AWS Transit Gateway](#) dapat membantu Anda mengelola jaringan dengan lebih mudah dan mencegah jaringan yang tumpang tindih—misalnya, perencanaan rentang Perutean Antar-Domain Tanpa Kelas (CIDR) — dibuat dengan lingkungan cloud lokal atau lingkungan cloud lainnya. Dengan begitu, Anda dapat merutekan lalu lintas ke setiap jaringan sesuai kebutuhan. Anda harus menentukan bagaimana akun merutekan satu sama lain dan ke lingkungan lokal dan internet. Ini memungkinkan Anda untuk mengatur kontrol yang tepat untuk melindungi lalu lintas jaringan Anda. Misalnya, Anda harus memutuskan untuk menjadikan akun AWS sebagai perpanjangan dari pusat data lokal yang ada dan menggunakan pertahanan perimeternya, seperti firewall, sistem deteksi intrusi (IDS), dan sistem pencegahan intrusi (IPS), atau menyiapkan akun jaringan AWS yang mencakup pertahanan perimeter ini untuk melindungi sumber daya AWS Anda.
- Prioritaskan keamanan cloud — Kami merekomendasikan untuk berpindah dari satu akun ke lingkungan multi-akun sambil mengikuti praktik terbaik keamanan dalam menerapkan izin hak istimewa paling sedikit. Kami juga menyarankan agar Anda memiliki pemahaman menyeluruh tentang [model tanggung jawab bersama AWS](#) dan merencanakan bagaimana Anda dapat [mengamankan lingkungan](#) sambil mempertahankan kelincahan organisasi Anda. Untuk meningkatkan dan menjaga keamanan, Anda dapat menggunakan Amazon API Gateway, AWS WAF, Application Load Balancers, Amazon, AWS CloudWatch, GuardDuty Amazon CloudTrail,

---

dan layanan lainnya. Untuk mempelajari lebih lanjut tentang strategi multi-akun, lihat [Transisi ke beberapa akun AWS dalam dokumentasi AWS Prescriptive Guidance](#).

- **Kelola layanan TI bersama di cloud** — Untuk mengelola beban kerja di cloud secara efisien, sangat penting untuk mengidentifikasi semua layanan bersama yang digunakan oleh beban kerja Anda dan merencanakan bagaimana mereka akan disediakan di cloud. Misalnya, ini termasuk Active Directory, file server, database SQL, Domain Name System (DNS), jaringan pribadi virtual (VPN), Simple Mail Transfer Protocol (SMTP), cadangan, dan layanan pemantauan. Setelah Anda mengambil inventaris, Anda dapat memutuskan antara memperluas layanan yang ada ke cloud, menyiapkan instance layanan yang sama sekali baru, atau menggunakan layanan cloud terkelola alternatif. Bagian selanjutnya dari panduan ini akan membahas pertimbangan ini secara lebih rinci.

# Jalur ke awan

Bagian ini menjelaskan pendekatan tingkat tinggi untuk menerapkan praktik terbaik untuk memigrasi aplikasi Windows Anda ke AWS. Rincian strategi dan langkah-langkah migrasi ini dijelaskan di bagian selanjutnya dari panduan ini.

## Strategi migrasi

Strategi migrasi adalah pendekatan yang digunakan untuk memigrasi beban kerja ke AWS Cloud. Ada tujuh strategi migrasi untuk memindahkan aplikasi ke cloud. Strategi ini dikenal sebagai 7 Rs dan dibangun di atas [7 Rs](#) yang diidentifikasi Gartner pada tahun 2019.

- Rehost (angkat dan geser)- Pindahkan aplikasi ke cloud tanpa membuat perubahan apa pun untuk memanfaatkan kemampuan cloud.
- Pindah (pengangkatan dan pergeseran tingkat hypervisor)- Pindahkan infrastruktur ke cloud tanpa membeli perangkat keras baru, menulis ulang aplikasi, atau memodifikasi operasi yang ada.
- Replatform (angkat dan bentuk kembali)- Pindahkan aplikasi ke cloud dan perkenalkan beberapa tingkat pengoptimalan untuk memanfaatkan kemampuan cloud.
- Pembelian kembali (drop dan toko)- Beralih ke produk yang berbeda, biasanya dengan beralih dari lisensi tradisional ke model perangkat lunak sebagai layanan (SaaS).
- Refaktor/Re-arsitek- Pindahkan aplikasi dan modifikasi arsitekturnya dengan memanfaatkan sepenuhnya fitur cloud-native untuk meningkatkan kelincahan, kinerja, dan skalabilitas.
- Mempertahankan (meninjau 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 justifikasi bisnis untuk memigrasinya.
- Pensiun- Menonaktifkan atau menghapus aplikasi yang tidak lagi diperlukan di lingkungan sumber Anda.

## Transformasi utama

Transformasi utama berikut terjadi ketika Anda memodernisasi aplikasi Windows lama dan database:

- Rehost- Langkah pertama adalah memindahkan infrastruktur lokal Anda ke infrastruktur cloud. Strategi ini sering disebut sebagai “lift and shift” atau rehosting. Rehosting berarti memigrasi

aplikasi dan database yang ada ke instance server cloud. Tidak perlu perubahan kode dan Anda bertanggung jawab untuk mengelola konfigurasi instans, gambar perangkat lunak, dan sumber daya lainnya.

- **Replatform-** Setelah Anda bermigrasi ke lingkungan cloud, transformasi berikutnya adalah sekitar replatforming aplikasi dan database ke lingkungan yang lebih otomatis dan dikelola. Dari perspektif aplikasi, itu berarti berpindah dari mesin virtual (VM) ke kontainer. Aplikasi kontainer dapat membantu Anda mengembangkan, memelihara, dan menerapkan aplikasi lebih cepat dan meningkatkan portabilitas. AWS memiliki alat, seperti [AWS App2Kontainer](#), untuk membantu mengotomatiskan proses pengemasan aplikasi lama. Di sisi database, beralih dari model layanan mandiri ke layanan database terkelola, seperti Amazon RDS untuk SQL Server, menghilangkan kebutuhan untuk penyediaan, patching, dan pencadangan. Ini pada akhirnya membebaskan sumber daya untuk aktivitas yang dapat menambah nilai lebih bagi organisasi Anda.
- **Refaktor/Re-arsitek-** Area transformasi ketiga adalah beralih dari lisensi perangkat lunak komersial ke opsi sumber terbuka. Banyak vendor perangkat lunak komersial tradisional telah membangun bisnis mereka di sekitar perjanjian lisensi perangkat lunak yang ditujukan untuk mengunci pelanggan dan menggunakan persyaratan lisensi hukuman untuk memaksa peningkatan dan migrasi. Seringkali, biaya lisensi perangkat lunak komersial biasanya menambah 20-50 persen biaya di atas opsi sumber terbuka yang setara. Kami merekomendasikan refactoring aplikasi dan database Anda untuk memanfaatkan opsi sumber terbuka sehingga Anda dapat mengurangi biaya, meningkatkan kinerja, dan mendapatkan akses ke inovasi terbaru.

Anda dapat menyelesaikan bidang transformasi utama ini secara progresif secara bertahap atau sekaligus tergantung pada aplikasi Anda dan kesiapan keseluruhan untuk memodernisasi.

## Memilih strategi migrasi

Strategi migrasi yang dipilih bergantung pada tujuan bisnis dan TI organisasi Anda. Beberapa pendorong bisnis yang paling umum adalah mengurangi biaya, mengurangi risiko, meningkatkan efisiensi, mengatasi kesenjangan keterampilan, dan mempercepat inovasi. Kami menyarankan Anda untuk mengevaluasi driver mana yang penting bagi Anda, lalu memilih strategi migrasi berdasarkan driver Anda dengan menggunakan panduan berikut. Juga, ingatlah bahwa ketiga pendekatan adalah jalan yang mungkin dalam perjalanan modernisasi cloud Anda, tergantung pada prioritas Anda selama setiap fase perjalanan.

### Kapan harus rehost

Rehosting (atau lift and shift) biasanya lebih cepat dan lebih mudah karena Anda tidak perlu membuat kode atau perubahan arsitektur dalam aplikasi. Rehosting juga meminimalkan risiko dan gangguan pada bisnis. Tim operasi dapat terus menjalankan bisnis seperti biasa karena aplikasi tidak berubah. Hal ini terutama berlaku untuk migrasi pada skala di mana bahkan perubahan kecil menjadi signifikan karena banyaknya beban kerja yang terlibat. Namun, penting untuk mempertimbangkan bahwa rehosting tidak mengambil keuntungan penuh dari manfaat cloud. Misalnya, jika Anda memigrasi aplikasi dengan masalah platform yang ada, masalah tersebut akan tetap ada setelah migrasi. Akhirnya, perlu dipertimbangkan bahwa total biaya kepemilikan (TCO) dan laba atas investasi (ROI) untuk rehosting lebih rendah dibandingkan dengan pendekatan migrasi lainnya.

### Kapan replatform/re-arsitek

Replatforming umumnya lebih hemat biaya daripada rehosting. Anda dapat menggunakan replatforming untuk meningkatkan otomatisasi dan memungkinkan aplikasi Anda menggunakan kemampuan cloud dengan lebih baik seperti penskalaan otomatis, pemantauan, dan melakukan pencadangan. Replatforming mengurangi overhead operasional untuk tim operasi cloud dan meminimalkan risiko dari masalah platform yang sudah ada sebelumnya. Namun, replatforming membutuhkan waktu lebih lama daripada migrasi rehosting. Juga, replatforming membutuhkan keterampilan tambahan untuk mengkonfigurasi otomatisasi yang melakukan perubahan kode pada aplikasi dan untuk mengoperasikan platform baru.

### Kapan harus refactor

Refactor umumnya merupakan pendekatan migrasi yang paling hemat biaya. Refactoring adalah pendekatan cloud-native yang memungkinkan aplikasi untuk cepat beradaptasi dengan persyaratan baru dengan memisahkan komponen aplikasi untuk meningkatkan ketahanan aplikasi. Namun, refactoring membutuhkan keterampilan pengkodean dan otomatisasi yang lebih canggih. Refactoring juga membutuhkan waktu lebih lama untuk diimplementasikan karena melibatkan pembangunan kembali aplikasi.

# Proses migrasi Windows

Migrasi lingkungan Windows yang ada ke AWS memerlukan perencanaan dan implementasi yang cermat. Prosesnya melibatkan identifikasi penggunaan sumber daya Anda saat ini, menilai potensi penghematan biaya migrasi ke AWS, menentukan kebutuhan keamanan Anda, dan membangun arsitektur cloud yang terdefinisi dengan baik yang memenuhi semua persyaratan organisasi Anda. Anda dapat menggunakan AWS untuk memigrasikan infrastruktur server Windows Anda saat ini dengan cepat dan mudah, mengurangi biaya operasional sekaligus memaksimalkan efisiensi sistem. AWS juga menawarkan berbagai alat dan layanan canggih untuk membantu Anda mempertahankan kendali atas seluruh proses dan memastikan bahwa lingkungan Windows Anda di cloud dikonfigurasi secara optimal untuk kinerja maksimum.

Bagian ini memberikan gambaran umum tentang proses migrasi tiga fase yang dikembangkan AWS untuk membantu organisasi dalam keberhasilan migrasi beberapa aplikasi ke AWS Cloud: menilai, memobilisasi, dan memigrasikan serta memodernisasi.

## Menilai

Fase penilaian membantu Anda memahami keadaan kesiapan organisasi Anda untuk pindah ke cloud. Anda dapat menggunakan alat AWS untuk membantu Anda dalam fase penilaian dengan menilai sumber daya komputasi lokal Anda dan membuat proyeksi biaya untuk menjalankan aplikasi di AWS. Kami menyarankan Anda mempertimbangkan alat-alat berikut:

- Gunakan [penilaian kesiapan migrasi](#) untuk memahami di mana Anda berada dalam perjalanan cloud.
- Gunakan [AWS Optimization and Licensing Assessment \(AWS OLA\)](#) untuk menilai dan mengoptimalkan lingkungan lokal dan cloud saat ini, berdasarkan pemanfaatan sumber daya aktual, lisensi pihak ketiga, dan dependensi aplikasi.
- Gunakan [Migration Evaluator](#) untuk membantu Anda membuat kasus bisnis berbasis data untuk migrasi ke AWS.
- Gunakan [Cloud Economics Center](#) untuk membangun kasus bisnis untuk migrasi Anda dengan menentukan tujuan Anda, seperti peningkatan keandalan, pengoptimalan biaya, dan skalabilitas.
- Gunakan [AWS Migration Hub](#) untuk mengumpulkan data inventaris server dan aplikasi untuk penilaian, perencanaan, dan pelacakan migrasi Anda.



- Gunakan [PowerShell modul Migration Validator Toolkit untuk menemukan](#) beban kerja Microsoft Anda dan memigrasikannya ke AWS.

## Memobilisasi

Selama fase mobilisasi, Anda mengembangkan rencana migrasi dan mengulangi rencana bisnis Anda dan mengatasi kesenjangan dalam kesiapan Anda yang terungkap dalam fase penilaian. Sangat penting untuk fokus pada membangun lingkungan dasar Anda, mendorong kesiapan operasional, dan mengembangkan keterampilan cloud. Migrasi portofolio aplikasi yang besar bisa menjadi tugas yang kompleks. Untuk memudahkan proses ini, AWS menyediakan berbagai alat dan layanan untuk membantu Anda memigrasikan serangkaian beban kerja pilot ke cloud dengan cepat, aman, dan hemat biaya. Mengumpulkan data pada portofolio aplikasi Anda dan merasionalisasi aplikasi menggunakan satu atau lebih dari tujuh strategi migrasi umum—rehost, relokasi, replatform, pembelian kembali, refactor/arsitek ulang, pertahankan, dan pensiun—dapat memberikan dasar yang lebih baik untuk pengambilan keputusan. AWS menawarkan serangkaian layanan yang dapat Anda gunakan untuk memigrasikan aplikasi dan beban kerja berbasis Windows ke cloud, termasuk yang berikut ini:

- [AWS Application Discovery Service](#)
- [Layanan Migrasi Aplikasi AWS](#)
- [AWS Database Migration Service](#)
- [Mitra Kompetensi Migrasi AWS](#)
- [Manajemen dan Tata Kelola di AWS](#)
- [AWS Control Tower](#)

## Migrasi dan modernisasi

Pada fase migrasi dan modernisasi, Anda harus hati-hati mendesain, memigrasi, dan memvalidasi setiap aplikasi yang berada dalam cakupan migrasi. Layanan Migrasi Aplikasi memudahkan migrasi server dalam jumlah besar dari infrastruktur fisik, virtual, atau cloud ke AWS. Dengan Layanan Migrasi Aplikasi, Anda dapat menggunakan proses otomatis yang sama untuk berbagai aplikasi dan dengan cepat mengangkat dan memindahkannya dari lingkungan yang ada ke cloud.

Solusi [Cloud Migration Factory on AWS](#) dirancang untuk mengoordinasikan dan mengotomatiskan proses manual untuk migrasi skala besar yang melibatkan sejumlah besar server. Solusi ini

membantu Anda meningkatkan kinerja dan mencegah jendela cutover yang panjang dengan menyediakan platform orkestrasi untuk memigrasikan beban kerja ke AWS dalam skala besar. [AWS Professional Services](#), [AWS Partners](#), dan perusahaan lain telah menggunakan solusi ini untuk membantu pelanggan memigrasikan ribuan server ke AWS Cloud.

Setelah migrasi selesai, Anda dapat menggunakan [AWS Migration Hub Refactor Spaces](#) untuk mengurangi pekerjaan yang tidak terdiferensiasi saat memfaktorkan ulang aplikasi Anda untuk AWS. Refactor Spaces menyediakan easy-to-use ruang kerja yang memungkinkan pengembang untuk secara bertahap memfaktorkan ulang aplikasi yang ada ke dalam arsitektur modern dengan overhead atau gangguan minimal. Anda dapat menggunakan Refactor Spaces untuk memanfaatkan berbagai layanan AWS dengan cepat yang dioptimalkan untuk aplikasi Anda.

Tim Anda ahli dalam membangun dan menjalankan beban kerja Microsoft di tempat. Pengalaman itu dapat ditingkatkan di cloud. Bermigrasi ke AWS dapat memberikan pengalaman yang lebih efisien dan andal untuk dunia Windows yang Anda andalkan. Dengan AWS, Anda akan mendapatkan akses ke berbagai layanan cloud yang dirancang agar lebih mudah dan lebih cepat untuk memigrasikan beban kerja Microsoft yang ada. Anda bisa mendapatkan keuntungan dari kapasitas yang lebih terukur, opsi penyimpanan yang ditingkatkan, dan keamanan yang ditingkatkan.

# Penemuan lingkungan Windows

Dengan teknologi yang tersedia saat ini, seperti Layanan Migrasi Aplikasi, memindahkan Windows Server, Linux, dan sistem operasi berbasis x86 lainnya serta beban kerjanya ke AWS cukup mudah. Namun, membuat beban kerja tersebut berfungsi dengan baik dan melakukannya dalam skala besar, menghadirkan serangkaian tantangan yang berbeda. Bagian ini dimaksudkan untuk mengidentifikasi pertimbangan migrasi yang memungkinkan Anda memigrasikan beban kerja Microsoft dengan cepat, aman, dan lancar.

## Menilai

Meskipun Anda dapat “brute force” migrasi yang lebih kecil (seperti yang melibatkan 100 server) dengan perencanaan dan otomatisasi minimal, Anda tidak dapat memindahkan 500 atau lebih server dengan menggunakan metodologi ini. Pertimbangan berikut adalah kontributor utama untuk migrasi skala besar yang sukses, dan Anda dapat menggunakan [Penilaian Kesiapan Migrasi \(MRA\)](#) untuk mengidentifikasi bidang pertimbangan yang ingin Anda fokuskan.

## Arsitektur perusahaan

Semakin banyak hutang teknologi yang ada di lingkungan semakin sulit untuk bermigrasi. Organisasi yang memiliki program arsitektur perusahaan yang sehat berusaha untuk membatasi lingkungan mereka ke versi perangkat lunak dan sistem saat ini dan terbaru (sering disebut versi N dan N -1 dari rilis utama). Ini tidak hanya mengurangi jumlah skenario yang harus Anda perhitungkan, tetapi juga memanfaatkan kemajuan rilis yang lebih baru. Misalnya, Windows Server 2012, Windows Server 2008, dan versi Windows Server yang lebih lama semakin jauh lebih sulit untuk diotomatisasi di lingkungan Windows Server daripada versi yang lebih saat ini. Lisensi juga lebih sulit untuk versi yang lebih lama dan tidak didukung.

## Standardisasi dan manajemen konfigurasi

Standardisasi lingkungan adalah faktor lain yang perlu dipertimbangkan. Organisasi yang memiliki lingkungan yang dibangun dengan tangan dan dipelihara dianggap lebih seperti hewan peliharaan. Setiap sistem unik dan ada kombinasi konfigurasi yang jauh lebih mungkin daripada jika mereka dibangun menggunakan gambar standar, infrastruktur sebagai kode (IAC), atau integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) pipa.

Misalnya, ini adalah praktik terbaik untuk membangun kembali server web biasa menggunakan IAC atau CI/CD saat bermigrasi, sebagai lawan memigrasi server individual secara manual. Ini juga merupakan praktik terbaik untuk menyimpan semua data persisten dalam datastore seperti database, berbagi file, atau repositori. Jika sistem tidak dibangun kembali menggunakan IAC atau CI/CD, mereka setidaknya harus menggunakan alat manajemen konfigurasi (seperti Puppet, Chef, atau Ansible) untuk menstandarisasi server yang mereka miliki.

## Data yang bagus

Data yang baik juga merupakan faktor kunci untuk migrasi yang sukses. Data yang akurat mengenai server saat ini dan metadata mereka sangat penting untuk otomatisasi dan perencanaan. Kurangnya data yang baik meningkatkan kesulitan saat merencanakan migrasi. Contoh data yang baik termasuk inventaris server yang akurat, aplikasi di server, perangkat lunak pada server dengan versi, jumlah CPU, jumlah memori, dan jumlah disk. Kami menyarankan Anda untuk menangkap data apa pun yang diperlukan oleh perencana gelombang untuk perencanaan atau data apa pun yang Anda rencanakan untuk digunakan sebagai bagian dari mengotomatiskan proses migrasi.

## Otomatisasi

Otomatisasi sangat penting untuk migrasi dalam skala besar. Contoh otomatisasi termasuk menginstal agen, memperbarui versi perangkat lunak utilitas yang diperlukan untuk otomatisasi seperti .NET atau PowerShell, memuat atau memperbarui perangkat lunak untuk AWS seperti Agen AWS Systems Manager (Agen SSM), AmazonCloudWatchagen, atau perangkat lunak pencadangan atau manajemen lain yang diperlukan untuk berjalan di AWS.

## Perencanaan terperinci

Mengembangkan dan mengelola rencana terperinci juga penting untuk migrasi dalam skala besar. Anda harus memiliki rencana yang terdefinisi dengan baik untuk memigrasi 50 server seminggu selama berminggu-minggu. Rencana yang efektif meliputi:

- Gunakan perencanaan gelombang untuk mengatur server menjadi gelombang sesuai dengan dependensi dan prioritas Anda.
- Gunakan perencanaan mingguan (mengarah ke cutover) untuk berkomunikasi dengan tim aplikasi dan mengidentifikasi jaringan, DNS, firewall, dan detail lainnya yang harus ditangani selama cutover.
- Gunakan rinci, hour-to-hour merencanakan (sekitar cutover aktual) untuk menggambarkan jendela pemeliharaan cutover.

- Gunakan kriteria go/no-go untuk menjelaskan dalam keadaan apa aplikasi akan dianggap dipotong ke AWS atau harus gagal kembali ke lokasi sumber.
- Gunakan kegiatan pembersihan sebagai tindak lanjut kegiatan yang harus diselesaikan. Kegiatan ini dapat terjadi di luar jendela pemeliharaan cutover atau setelah selesainya [hipercare](#). Aktivitas pembersihan termasuk memverifikasi cadangan dan berbagai agen, menghapus agen Layanan Migrasi Aplikasi dari server, atau menghapus server sumber dan sumber daya terkait.

## Memobilisasi

Selama fase mobilisasi, penting untuk menemukan sebanyak mungkin kompleksitas dan variasi organisasi Anda sehingga dapat dipertanggungjawabkan selama perencanaan migrasi. Idealnya, Anda dapat menghindari berurusan dengan kompleksitas dan variasi seperti itu selama jendela pemeliharaan langsung dan mencegah kegagalan apa pun.

## Tantangan migrasi dalam skala besar

Kegagalan migrasi terjadi ketika aplikasi atau aplikasi dipotong ke lingkungan baru mereka dan kinerja atau persyaratan fungsional tidak dapat dipenuhi dalam jendela pemeliharaan migrasi. Ini memaksa aplikasi atau aplikasi gagal kembali ke lokasi semula. Selain itu, semua aplikasi lain yang bergantung pada aplikasi atau aplikasi itu juga perlu gagal kembali. Migrasi yang gagal cenderung berdampak tidak hanya pada gelombang saat ini tetapi gelombang masa depan karena aplikasi harus dijadwal ulang.

## Dependensi yang sensitif terhadap latensi

Alasan utama migrasi gagal adalah dependensi yang sensitif terhadap latensi. Gagal mengidentifikasi dependensi yang sensitif terhadap latensi dapat menimbulkan masalah kinerja yang mengakibatkan waktu respons atau waktu transaksi yang tidak dapat diterima. Misalnya, biasanya aplikasi memindahkan database dan server aplikasinya ke cloud pada saat yang sama karena mereka sering berkomunikasi satu sama lain dan membutuhkan waktu respons sub-milidetik yang mereka miliki ketika keduanya berada di pusat data yang sama. Memindahkan hanya database ke cloud kemungkinan akan memperkenalkan latensi beberapa detik ke dalam transaksi tersebut, sehingga menghasilkan dampak kinerja yang signifikan terhadap aplikasi. Ini juga berlaku untuk aplikasi yang sangat bergantung satu sama lain dan harus berada di pusat data yang sama untuk melakukan secara memadai.

Oleh karena itu, memahami dan menangani dependensi aplikasi sangat penting saat merencanakan migrasi. Aplikasi dan layanan yang bergantung satu sama lain harus diidentifikasi sehingga dapat dimigrasi bersama.

## Layanan bersama TI

Setelah beban kerja berada di cloud, dibutuhkan berbagai layanan untuk berfungsi dan dipelihara dengan baik dan aman. Ini termasuk zona pendaratan, perimeter jaringan dan keamanan, otentikasi, penambalan, pemindai keamanan, alat manajemen layanan TI, cadangan, host benteng, dan sumber daya lainnya. Tanpa layanan ini, beban kerja mungkin tidak beroperasi dengan baik dan akan dipaksa untuk gagal kembali ke lokasi semula.

## Pembaruan konfigurasi

Dalam kebanyakan kasus, Anda harus membuat beberapa perubahan konfigurasi agar beban kerja berfungsi dengan baik setelah beban kerja dipindahkan ke cloud. Perubahan konfigurasi ini sering dikaitkan dengan dependensi beban kerja berikut:

- Aturan firewall
- Izinkan daftar
- Catatan DNS
- String koneksi

Jika Anda tidak membuat pembaruan konfigurasi yang tepat, maka beban kerja, penggunaannya, dan sistem dependennya mungkin gagal berkomunikasi satu sama lain. Menyelesaikan masalah ini dalam jendela pemadaman mungkin terjadi, tetapi perubahan saat ini dapat memakan waktu atau memerlukan catatan perubahan yang tidak dapat dipenuhi tepat waktu.

## Pengujian fungsional aplikasi

Tantangan lain untuk migrasi dalam skala besar adalah perlunya pengujian fungsional aplikasi. Ini sangat penting karena banyak organisasi mengandalkan tim aplikasi untuk mengidentifikasi dependensi yang sensitif terhadap latensi, layanan bersama TI, atau pembaruan konfigurasi yang diperlukan. Idealnya, tim aplikasi menyediakan rencana pengujian tertulis atau otomatis yang dapat mereka jalankan selama jendela pemeliharaan langsung untuk memvalidasi bahwa aplikasi mereka berfungsi penuh dengan kinerja yang dapat diterima. Untuk menjaga jendela pemeliharaan cutover seminimal mungkin, pengujian harus dapat diselesaikan dalam waktu 30 menit.

## Alat untuk penemuan ketergantungan aplikasi

Menentukan dependensi antar aplikasi sangat penting untuk migrasi yang berhasil—baik untuk mendeteksi dependensi yang sensitif terhadap latensi dan item konfigurasi konektivitas. Ada beberapa alat yang tersedia di pasar untuk menemukan dependensi, seperti [Layanan Penemuan Aplikasi](#) (agen dan alat tanpa agen) dan [Cloudfunder](#) (alat berbasis agen).

Saat Anda memilih alat untuk penemuan dependensi aplikasi, pertimbangkan hal berikut:

- **Durasi**- Kami menyarankan Anda menjalankan alat penemuan cukup lama untuk menangkap peristiwa khusus aplikasi seperti puncak yang diketahui, akhir bulan, dan acara lainnya. Minimum yang disarankan adalah 30 hari.
- **Aktif (berbasis agen)**- Alat penemuan ketergantungan aktif sering disematkan di kernel sistem operasi dan menangkap semua transaksi. Namun, ini biasanya metode yang paling mahal dan memakan waktu.
- **Pasif (tanpa agen)**- Alat penemuan dependensi pasif jauh lebih murah dan lebih cepat untuk diterapkan tetapi berisiko kehilangan beberapa koneksi yang digunakan lebih rendah.
- **Pengetahuan kelembagaan**- Meskipun alat penemuan aplikasi memberikan informasi yang lebih rinci dan akurat, sebagian besar organisasi mengandalkan tim aplikasi mereka dan pengetahuan kelembagaan mereka untuk menemukan dependensi aplikasi. Tim aplikasi sering memiliki pengetahuan tentang dependensi yang sensitif terhadap latensi, tetapi tidak jarang mereka melewatkan beberapa detail seperti pengaturan konfigurasi konektivitas, aturan firewall, atau mengizinkan persyaratan daftar dari mitra. Anda dapat menggunakan pengetahuan kelembagaan untuk meningkatkan penemuan ketergantungan aplikasi Anda, tetapi kami menyarankan Anda juga mempertimbangkan dan mengurangi risiko yang terlibat. Misalnya, ada risiko kehilangan item konfigurasi konektivitas atau dependensi yang sensitif terhadap latensi jika Anda hanya bergantung pada pengetahuan tim aplikasi Anda. Hal ini dapat mengakibatkan pemadaman atau migrasi gagal. Untuk mengurangi risiko ini, kami sarankan Anda melakukan pengujian fungsional aplikasi terperinci.

# Memigrasi beban kerja Microsoft

Bagian ini mencakup panduan preskriptif untuk beban kerja Microsoft tertentu. Semua pendekatan khusus beban kerja berikut mematuhi kerangka kerja menilai, memobilisasi, dan memigrasi serta memodernisasi.

## Migrasi Direktori Aktif

Active Directory adalah identitas khas dan solusi manajemen akses untuk banyak lingkungan perusahaan. Penggabungan DNS, pengguna, dan manajemen mesin menjadikan Active Directory pilihan ideal untuk beban kerja Microsoft dan Linux untuk otentikasi pengguna terpusat. Saat merencanakan perjalanan ke cloud atau AWS, Anda dihadapkan pada pilihan untuk memperluas Active Directory ke AWS atau menggunakan layanan terkelola untuk membongkar pengelolaan infrastruktur layanan direktori. Kami menyarankan Anda memahami risiko dan manfaat dari setiap opsi saat memutuskan pendekatan yang tepat untuk organisasi Anda.

Strategi yang tepat untuk migrasi Direktori Aktif adalah strategi yang sesuai dengan kebutuhan organisasi Anda dan memungkinkan Anda memanfaatkan AWS Cloud. Ini melibatkan pertimbangan tidak hanya layanan direktori itu sendiri tetapi bagaimana mereka berinteraksi dengan layanan AWS lainnya. Selain itu, Anda harus mempertimbangkan tujuan jangka panjang untuk tim yang mengelola Active Directory.

Selain migrasi Direktori Aktif, Anda harus memutuskan struktur akun untuk lokasi Active Directory, topologi jaringan akun AWS Anda, dan integrasi DNS apa dan layanan AWS potensial lainnya yang akan Anda gunakan yang memerlukan Active Directory. Untuk informasi tentang mendesain topologi akun Anda dan pertimbangan strategi migrasi lainnya, lihat bagian [Praktik terbaik dasar](#) dari panduan ini.

## Menilai

Untuk mengimplementasikan migrasi yang berhasil, penting untuk menilai infrastruktur yang ada dan memahami fitur utama yang diperlukan untuk lingkungan Anda. Kami menyarankan Anda meninjau area berikut sebelum memilih cara bermigrasi:

- Tinjau desain infrastruktur AWS yang ada — Ikuti panduan di bagian [penemuan lingkungan Windows](#) dari panduan ini dan gunakan metode penilaian untuk membantu meninjau infrastruktur Direktori Aktif yang ada jika Anda belum mengetahui jejak dan persyaratan infrastrukturnya. Kami



menyarankan Anda menggunakan ukuran yang ditentukan dari Microsoft untuk infrastruktur Direktori Aktif di AWS. Jika Anda memperluas infrastruktur Direktori Aktif ke AWS, Anda mungkin hanya memerlukan sebagian jejak otentikasi Direktori Aktif di AWS. Untuk alasan ini, hindari melebih-lebihkan lingkungan Anda kecuali Anda benar-benar memindahkan jejak Active Directory Anda ke AWS. Untuk informasi selengkapnya, lihat [Perencanaan kapasitas untuk Layanan Domain Direktori Aktif](#) di dokumentasi Microsoft.

- Tinjau desain Active Directory lokal yang ada — Tinjau pemanfaatan Direktori Aktif lokal (dikelola sendiri) saat ini. Jika Anda memperluas lingkungan Direktori Aktif ke AWS, sebaiknya jalankan Active Directory pada beberapa pengontrol domain di AWS bahkan sebagai ekstensi ke lingkungan lokal Anda. Ini mematuhi [AWS Well-Architected](#) Framework untuk merancang potensi kegagalan dengan menerapkan instance di beberapa Availability Zone.
- Identifikasi dependensi dalam aplikasi dan jaringan — Sebelum memilih strategi migrasi apa yang terbaik, Anda harus sepenuhnya memahami semua fitur Active Directory yang diperlukan organisasi Anda untuk fungsionalitas. Ini berarti bahwa ketika memilih antara layanan terkelola atau hosting mandiri, penting untuk memahami opsi untuk masing-masing. Pertimbangkan item berikut saat memutuskan migrasi mana yang tepat untuk Anda:
  - Persyaratan untuk akses — Persyaratan untuk akses untuk mengontrol Active Directory akan menetapkan jalur migrasi yang tepat untuk Anda. Jika Anda memerlukan akses penuh ke pengontrol domain Direktori Aktif untuk menginstal semua jenis agen untuk peraturan kepatuhan, AWS Managed Microsoft AD mungkin bukan solusi yang tepat untuk Anda. Sebagai gantinya, selidiki ekstensi Direktori Aktif dari pengontrol domain Anda ke Amazon EC2 dalam akun AWS Anda.
  - Garis waktu migrasi — Jika Anda memiliki garis waktu perpanjangan untuk migrasi yang tidak memiliki tanggal penyelesaian yang jelas, verifikasi bahwa Anda memiliki kemungkinan untuk administrasi instance di cloud dan di lingkungan lokal. Otentikasi adalah komponen kunci untuk beban kerja Microsoft untuk menghindari masalah administrasi. Sebaiknya Anda merencanakan pemindahan Active Directory di awal migrasi.
- Strategi Backup — Jika Anda menggunakan cadangan Windows yang ada untuk menangkap status sistem pengontrol domain Active Directory, Anda dapat terus menggunakan strategi pencadangan yang ada di AWS. Selain itu, AWS menawarkan opsi teknologi untuk membantu Anda mencadangkan instans Anda. Misalnya, [AWS Data Lifecycle Manager](#), [AWS Backup](#), dan [AWS Elastic Disaster Recovery](#) adalah teknologi yang didukung untuk mencadangkan pengontrol domain Active Directory. Untuk menghindari masalah, sebaiknya jangan mengandalkan pemulihan Active Directory. Praktik terbaik yang disarankan adalah membangun arsitektur yang tangguh, tetapi sangat penting untuk memiliki metode cadangan jika pemulihan diperlukan.

- Kebutuhan pemulihan bencana (DR) — Jika Anda memigrasikan Active Directory ke AWS, Anda harus merancang ketahanan jika terjadi bencana. Jika Anda memindahkan direktori aktif yang ada ke AWS, Anda dapat menggunakan Wilayah AWS sekunder dan menghubungkan kedua Wilayah dengan menggunakan Transit Gateway untuk memungkinkan terjadinya replikasi. Ini biasanya metode yang disukai. Ada beberapa organisasi yang memiliki berbagai persyaratan untuk menguji failover di lingkungan yang terisolasi, di mana Anda memutuskan konektivitas antara situs primer dan sekunder selama sehari-hari untuk menguji keandalan. Jika ini adalah persyaratan dalam organisasi Anda, mungkin perlu waktu untuk membersihkan masalah otak terpisah dari Active Directory. Anda mungkin dapat menggunakan [AWS Elastic Disaster Recovery](#) sebagai implementasi aktif/pasif di mana Anda meninggalkan situs DR Anda sebagai lingkungan failover dan harus secara rutin menguji strategi DR Anda secara terpisah. Merencanakan persyaratan tujuan waktu pemulihan (RTO) dan tujuan titik pemulihan (RPO) organisasi Anda merupakan faktor penting saat menilai migrasi Anda ke AWS. Pastikan Anda memiliki persyaratan yang ditentukan bersama dengan rencana pengujian dan failover untuk memvalidasi implementasi.

## Memobilisasi

Strategi yang tepat untuk memenuhi kebutuhan organisasi dan operasional Anda merupakan elemen penting dalam memigrasikan atau memperluas Active Directory ke AWS. Memilih bagaimana Anda akan berintegrasi dengan layanan AWS sangat penting untuk mengadopsi AWS. Pastikan untuk memilih ekstensi metode Active Directory atau AWS Managed Microsoft AD yang memenuhi persyaratan bisnis Anda. Ada beberapa fitur dalam layanan seperti Amazon RDS yang bergantung pada penggunaan AWS Managed Microsoft AD. Pastikan Anda mengevaluasi batasan layanan AWS untuk menentukan apakah ada batasan kompatibilitas untuk Active Directory di Amazon EC2 dan AWS Managed Microsoft AD. Kami menyarankan Anda mempertimbangkan poin integrasi berikut sebagai bagian dari proses perencanaan Anda.

Pertimbangkan alasan berikut untuk menggunakan Active Directory di AWS:

- Aktifkan aplikasi AWS untuk bekerja dengan Active Directory
- Menggunakan Active Directory untuk masuk ke AWS Management Console

### Aktifkan aplikasi AWS untuk bekerja dengan Active Directory

[Anda dapat mengaktifkan beberapa aplikasi dan layanan AWS seperti AWS Client VPN, AWS Management Console, AWS IAM Identity Center \(penerus AWS Single Sign-On\), AmazonChime,](#)

[Amazon Connect](#), [Amazon FSx for Windows QuickSight File Server](#), [Amazon](#), [Amazon RDS for SQL Server \(hanya berlaku untuk Direktori Directory Service WorkDocs\)](#), [Amazon](#), [Amazon](#), dan [Amazon WorkMail untuk menggunakan AWS Managed WorkSpaces](#) Direktori Microsoft AD. Saat Anda mengaktifkan aplikasi atau layanan AWS di direktori Anda, pengguna Anda dapat mengakses aplikasi atau layanan dengan kredensi Direktori Aktif mereka. Anda dapat menggunakan alat administrasi Direktori Aktif yang sudah dikenal untuk menerapkan objek kebijakan grup Active Directory (GPO) untuk mengelola instans Amazon EC2 untuk Windows atau Linux secara terpusat dengan menggabungkan instans Anda ke direktori [AWS](#) Managed Microsoft AD Anda.

Pengguna Anda dapat masuk ke instans Anda dengan kredensi Direktori Aktif mereka. Ini menghilangkan kebutuhan untuk menggunakan kredensial instans individu atau mendistribusikan file kunci pribadi (PEM). Ini memudahkan Anda untuk langsung memberikan atau mencabut akses ke pengguna dengan menggunakan alat administrasi pengguna Active Directory yang sudah Anda gunakan.

## Menggunakan Active Directory untuk masuk ke AWS Management Console

AWS Managed Microsoft AD memungkinkan Anda memberi anggota direktori akses ke AWS Management Console. Secara default, anggota direktori Anda tidak memiliki akses ke sumber daya AWS apa pun. Anda menetapkan peran AWS Identity and Access Management (IAM) kepada anggota direktori Anda untuk memberi mereka akses ke berbagai layanan dan sumber daya AWS. IAM role menentukan layanan, sumber daya, dan tingkat akses yang dimiliki anggota direktori Anda.

Misalnya, Anda dapat mengaktifkan pengguna untuk masuk ke AWS Management Console dengan [kredensial Direktori Aktif](#) mereka. Untuk melakukannya, Anda mengaktifkan AWS Management Console sebagai aplikasi di direktori, lalu tetapkan pengguna dan grup Active Directory Anda ke peran IAM. Saat pengguna Anda masuk ke AWS Management Console, mereka mengambil peran IAM untuk mengelola sumber daya AWS. Hal ini memudahkan Anda untuk memberi pengguna akses ke AWS Management Console tanpa perlu mengonfigurasi dan mengelola infrastruktur SAMP terpisah. Untuk informasi selengkapnya, lihat [Bagaimana sinkronisasi Direktori Aktif AWS IAM Identity Center meningkatkan pengalaman aplikasi AWS](#) di AWS Security Blog. Anda dapat memberikan akses ke akun pengguna di direktori Anda atau di Active Directory lokal Anda. Hal ini memungkinkan pengguna untuk masuk ke AWS Management Console atau melalui AWS Command Line Interface (AWS CLI) Command Line Interface (AWS CLI) dengan menggunakan kredensial dan izin yang ada untuk mengelola sumber daya AWS dengan menetapkan peran IAM langsung ke akun pengguna yang ada.

Sebelum Anda dapat memberikan akses konsol ke anggota direktori Anda, direktori Anda harus memiliki URL akses. Untuk informasi selengkapnya tentang cara melihat detail direktori dan mendapatkan URL akses Anda, lihat [Melihat informasi direktori](#) di Panduan Administrasi Layanan AWS Directory Service. Untuk informasi selengkapnya tentang cara membuat URL akses, lihat [Membuat URL akses](#) di Panduan Administrasi Layanan AWS Directory Service. Untuk informasi selengkapnya tentang cara membuat dan menetapkan peran IAM ke anggota direktori Anda, lihat [Memberi pengguna dan grup akses ke sumber daya AWS](#) di Panduan Administrasi Layanan Direktori AWS.

Pertimbangkan opsi migrasi berikut untuk Active Directory:

- Perluas Direktori Aktif
- Migrasi ke AWS Managed Microsoft AD
- Gunakan kepercayaan untuk menghubungkan Active Directory dengan AWS Managed Microsoft AD
- Integrasikan Active Directory DNS dengan Amazon Route 53

## Perluas Direktori Aktif

Jika Anda sudah memiliki infrastruktur Direktori Aktif dan ingin menggunakannya saat memigrasikan beban kerja Active Directory-aware ke AWS Cloud, AWS Managed Microsoft AD dapat membantu. Anda dapat menggunakan [trust](#) untuk menghubungkan AWS Managed Microsoft AD ke Active Directory yang ada. Ini berarti pengguna Anda dapat mengakses aplikasi Active Directory-aware dan AWS dengan kredensi Active Directory lokal mereka, tanpa perlu Anda menyinkronkan pengguna, grup, atau kata sandi. Misalnya, pengguna Anda dapat masuk ke AWS Management Console dan WorkSpaces dengan menggunakan nama pengguna dan kata sandi Active Directory yang ada. Selain itu, saat Anda menggunakan aplikasi Active Directory-aware seperti SharePoint AWS Managed Microsoft AD, pengguna Windows yang masuk dapat mengakses aplikasi ini tanpa perlu memasukkan kredensialnya lagi.

Selain menggunakan kepercayaan, Anda dapat memperluas Active Directory dengan menerapkan Active Directory untuk berjalan pada instans EC2 di AWS. Anda dapat melakukannya sendiri atau [bekerja dengan AWS](#) untuk membantu Anda dalam prosesnya. Kami menyarankan Anda menerapkan setidaknya dua pengontrol domain di Availability Zone yang berbeda saat memperluas Active Directory ke AWS. Anda mungkin perlu menerapkan lebih dari dua pengontrol domain berdasarkan jumlah pengguna dan komputer yang Anda miliki di AWS, tetapi jumlah minimum yang kami rekomendasikan adalah dua untuk alasan ketahanan. Anda juga dapat memigrasikan domain

Active Directory lokal ke AWS agar bebas dari beban operasional infrastruktur Active Directory Anda dengan menggunakan Active [Directory Migration Toolkit \(ADMT\)](#) dan [Server Ekspor Kata Sandi \(PES\)](#) untuk melakukan migrasi. Anda juga dapat menggunakan [Active Directory Launch Wizard](#) untuk menyebarkan Active Directory di AWS.

## Migrasi ke AWS Managed Microsoft AD

Anda dapat menerapkan dua mekanisme untuk menggunakan Active Directory di AWS. Salah satu metode adalah dengan mengadopsi AWS Managed Microsoft AD untuk memigrasikan objek Active Directory Anda ke AWS. Ini termasuk pengguna, komputer, kebijakan grup, dan banyak lagi. Mekanisme kedua adalah pendekatan manual di mana Anda mengekspor semua pengguna dan objek, dan kemudian secara manual mengimpor pengguna dan objek dengan menggunakan [Alat Migrasi Direktori Aktif](#).

Ada alasan tambahan untuk pindah ke AWS Managed Microsoft Active Directory:

- AWS Managed Microsoft AD adalah domain Microsoft Active Directory aktual yang memungkinkan Anda menjalankan beban kerja tradisional yang sadar Active Directory seperti Microsoft [Remote Desktop Licensing Manager](#), [Microsoft SharePoint](#), dan [Microsoft SQL Server Always On di AWS Cloud](#).
- AWS Managed Microsoft AD membantu Anda menyederhanakan dan meningkatkan keamanan aplikasi.NET yang terintegrasi dengan Direktori Aktif dengan menggunakan Akun Layanan Terkelola grup (GMSAs) dan delegasi terbatas Kerberos (KCD). Untuk informasi selengkapnya, lihat [Menyederhanakan Migrasi dan Meningkatkan Keamanan Direktori Aktif—Aplikasi.NET Terintegrasi dengan Menggunakan AWS Microsoft AD dalam dokumentasi AWS](#).

Anda dapat membagikan AWS Managed Microsoft AD di beberapa akun AWS. Ini memungkinkan Anda mengelola layanan AWS, seperti [Amazon EC2](#), tanpa perlu mengoperasikan direktori untuk setiap akun dan setiap Amazon Virtual Private Cloud (Amazon VPC). Anda dapat menggunakan direktori Anda dari akun AWS apa pun dan dari [VPC Amazon](#) mana pun dalam Wilayah AWS. Kemampuan ini membuatnya lebih mudah dan lebih hemat biaya untuk mengelola beban kerja sadar direktori dengan satu direktori di seluruh akun dan VPC. Misalnya, Anda sekarang dapat dengan mudah mengelola [beban kerja Microsoft](#) yang diterapkan dalam instans EC2 di beberapa akun dan VPC Amazon dengan menggunakan satu direktori AWS Managed Microsoft AD. Saat membagikan direktori AWS Managed Microsoft AD dengan akun AWS lain, Anda dapat menggunakan konsol Amazon EC2 atau [AWS Systems Manager](#) untuk menggabungkan instans Anda dengan mulus dari VPC Amazon apa pun dalam akun dan Wilayah AWS.

Anda dapat dengan cepat menerapkan beban kerja sadar direktori pada instans EC2 dengan menghilangkan kebutuhan untuk menggabungkan instans Anda secara manual ke domain atau menerapkan direktori di setiap akun dan Amazon VPC. Untuk informasi selengkapnya, lihat [Membagikan direktori Anda](#) di Panduan Administrasi Layanan AWS Directory. Perlu diingat bahwa ada biaya untuk berbagi lingkungan AWS Managed Microsoft AD. Anda dapat berkomunikasi dengan lingkungan AWS Managed Microsoft AD dari jaringan atau akun lain dengan menggunakan rekan Amazon VPC atau rekan Transit Gateway, sehingga berbagi mungkin tidak diperlukan. Jika Anda bermaksud menggunakan direktori dengan layanan berikut, maka Anda harus berbagi domain: Amazon Aurora MySQL, Amazon Aurora PostgreSQL, Amazon FSX, Amazon RDS untuk MariaDB, Amazon RDS for MariaDB, Amazon RDS untuk MySQL Oracle, Amazon RDS untuk PostgreSQL, dan Amazon RDS for SQL Server.

## Gunakan kepercayaan dengan AWS Managed Microsoft AD

Untuk memberi pengguna akses direktori yang ada ke sumber daya AWS, Anda dapat menggunakan kepercayaan dengan implementasi AWS Managed Microsoft AD Anda. Anda juga dapat membuat kepercayaan antara lingkungan AWS Managed Microsoft AD. Untuk informasi selengkapnya, lihat [Semua yang ingin Anda ketahui tentang trust dengan postingan AWS Managed Microsoft AD](#) di AWS Security Blog.

## Integrasikan Active Directory DNS dengan Amazon Route 53

Saat Anda bermigrasi ke AWS, Anda dapat mengintegrasikan DNS ke lingkungan Anda dengan menggunakan resolver Route 53 untuk mengizinkan akses ke server Anda (dengan menggunakan nama DNS mereka). Kami menyarankan Anda menggunakan titik akhir resolver Route 53 untuk mencapai hal ini daripada memodifikasi set opsi DHCP. Ini adalah pendekatan yang lebih terpusat untuk mengelola konfigurasi DNS Anda daripada memodifikasi set opsi DHCP. Selain itu, Anda dapat memanfaatkan berbagai aturan resolver. Untuk informasi selengkapnya, lihat posting [Mengintegrasikan resolusi DNS Layanan Direktori Anda dengan Amazon Route 53 Resolvers](#) di Blog Networking & Content Delivery dan [Mengatur resolusi DNS untuk jaringan hybrid di lingkungan AWS multi-akun dalam dokumentasi AWS Prescriptive Guidance AWS](#).

## Migrasi

Saat memulai migrasi ke AWS, sebaiknya pertimbangkan opsi konfigurasi dan perkakas untuk membantu Anda bermigrasi. Penting juga untuk mempertimbangkan aspek keamanan dan operasional jangka panjang dari lingkungan Anda.

Pertimbangkan opsi berikut:



- Keamanan cloud-native
- Alat untuk memigrasikan Active Directory ke AWS

## Keamanan cloud-native

- Konfigurasi grup keamanan untuk pengontrol Direktori Aktif — Jika Anda menggunakan AWS Managed Microsoft AD, pengontrol domain dilengkapi dengan konfigurasi keamanan VPC untuk akses terbatas ke pengontrol domain. Mungkin perlu bagi Anda untuk memodifikasi aturan grup keamanan untuk mengizinkan akses untuk beberapa kasus penggunaan potensial. Untuk informasi selengkapnya tentang konfigurasi grup keamanan, lihat [Meningkatkan konfigurasi keamanan jaringan AWS Managed Microsoft AD Anda](#) di Panduan Administrasi Layanan AWS Directory Service. Kami menyarankan agar Anda tidak mengizinkan pengguna untuk memodifikasi grup ini atau menggunakannya untuk layanan AWS lainnya. Mengizinkan pengguna lain untuk menggunakan ini dapat menyebabkan gangguan layanan ke lingkungan Active Directory Anda jika pengguna memodifikasinya untuk memblokir komunikasi yang diperlukan.
- Integrasikan dengan CloudWatch Log Amazon untuk log peristiwa Direktori Aktif — Jika Anda menjalankan AWS Managed Microsoft AD atau menggunakan Direktori Aktif yang dikelola sendiri, Anda dapat memanfaatkan CloudWatch Log Amazon untuk memusatkan pencatatan Direktori Aktif Anda. Anda dapat menggunakan CloudWatch log untuk menyalin otentikasi, keamanan, dan log lainnya. CloudWatch Ini memberi Anda cara mudah untuk mencari log di satu tempat, dan ini dapat membantu memenuhi beberapa persyaratan kepatuhan. Kami merekomendasikan integrasi dengan CloudWatch Log karena dapat membantu Anda merespons insiden masa depan dengan lebih baik di lingkungan Anda. Untuk informasi selengkapnya, lihat [Mengaktifkan CloudWatch Log Amazon untuk AWS Managed Active Directory](#) di AWS Directory Service Administration Guide dan [Amazon CloudWatch Logs untuk Windows Event Logs](#) di AWS Knowledge Center.

## Alat untuk memigrasikan Active Directory ke AWS

Kami menyarankan Anda menggunakan Alat Migrasi Direktori Aktif (ADMT) dan Server Ekspor Kata Sandi (PES) untuk melakukan migrasi Anda. Ini memungkinkan Anda untuk dengan mudah memindahkan pengguna dan komputer dari satu domain ke domain lainnya. Perhatikan pertimbangan berikut jika Anda menggunakan PES atau bermigrasi dari satu domain Direktori Aktif terkelola ke domain lain:

- Alat Migrasi Direktori Aktif (ADMT) untuk pengguna, grup, dan komputer — Anda dapat menggunakan [ADMT](#) untuk memigrasikan pengguna dari Active Directory yang dikelola sendiri

ke AWS Managed Microsoft AD. Pertimbangan penting adalah timeline migrasi dan pentingnya Security Identifier (SID) History. Riwayat SID tidak ditransfer selama migrasi. Jika mendukung Riwayat SID adalah kebutuhan penting, pertimbangkan untuk menggunakan Direktori Aktif yang dikelola sendiri di Amazon EC2 alih-alih ADMT sehingga Anda dapat mempertahankan Riwayat SID.

- Server Ekspor Kata Sandi (PES) — PES dapat digunakan untuk memigrasikan kata sandi ke tetapi tidak keluar dari AWS Managed Microsoft AD. Untuk informasi tentang cara memigrasi pengguna dan kata sandi dari direktori Anda, lihat [Cara memigrasikan domain lokal Anda ke AWS Managed Microsoft AD menggunakan ADMT di AWS Security Blog](#) dan [Server Ekspor Kata Sandi versi 3.1 \(x64\) dari dokumentasi Microsoft](#).
- LDIF - LDAP Data Interchange Format (LDIF) adalah format file yang digunakan untuk memperluas skema direktori AWS Managed Microsoft AD. File LDIF berisi informasi yang diperlukan untuk menambahkan objek dan atribut baru ke direktori. File harus memenuhi standar LDAP untuk sintaks dan harus berisi definisi objek yang valid untuk setiap objek yang ditambahkan file. Setelah Anda membuat file LDIF, Anda harus mengunggah file ke direktori untuk memperluas skema. Untuk informasi selengkapnya tentang penggunaan file LDIF untuk memperluas skema direktori AWS Managed Microsoft AD, lihat [Memperluas skema AWS Managed AD di Panduan Administrasi Layanan AWS Directory](#).
- CSVDE — Dalam beberapa kasus, Anda mungkin perlu mengekspor dan mengimpor pengguna ke direktori tanpa membuat kepercayaan dan menggunakan ADMT. Meskipun tidak ideal, Anda dapat menggunakan [Csvde](#) (alat baris perintah) untuk memigrasikan pengguna Active Directory dari satu domain ke domain lainnya. Untuk menggunakan Csvde, Anda harus membuat file CSV yang berisi informasi pengguna, seperti nama pengguna, kata sandi, dan keanggotaan grup. Kemudian, Anda dapat menggunakan perintah csvde untuk mengimpor pengguna ke domain baru. Anda juga dapat menggunakan perintah ini untuk mengekspor pengguna yang ada dari domain sumber. Ini mungkin berguna jika Anda bermigrasi dari sumber direktori lain, seperti Layanan Domain SAMBA ke Microsoft Active Directory. Untuk informasi selengkapnya, lihat [Cara Memigrasi Pengguna Microsoft Active Directory Anda ke Simple AD atau AWS Managed Microsoft AD](#) di AWS Security Blog.

## Sumber daya tambahan

- [Semua yang ingin Anda ketahui tentang kepercayaan dengan AWS Managed Microsoft AD \(AWS Security Blog\)](#)



- [Cara memigrasikan domain lokal Anda ke AWS Managed Microsoft AD menggunakan ADMT](#) (AWS Security Blog)
- [Direktori Aktif pada Hari Perendaman AWS](#) (AWS Workshop Studio)

## Migrasi Windows Server

Bagian ini berfokus pada berbagai opsi yang tersedia untuk memigrasikan Windows Server ke AWS.

### Menilai

Pertama, identifikasi aplikasi dan beban kerja yang perlu dimigrasikan ke AWS. Anda dapat menggunakan [AWS Application Discovery Service](#) untuk membuat peta infrastruktur lokal dan dependensi antar aplikasi. Ini membantu Anda mengidentifikasi server, aplikasi, dan layanan yang perlu Anda migrasi ke AWS.

Anda dapat menggunakan [AWS Migration Hub](#) untuk membuat inventaris aplikasi Anda dan mengevaluasi kompatibilitasnya dengan AWS. Migration Hub menyediakan tampilan terpusat dari portofolio aplikasi Anda dan membantu Anda merencanakan, melacak, dan mengelola proyek migrasi Anda. Anda juga dapat menggunakan alat penilaian pihak ketiga yang mendukung AWS, seperti Cloudamize atau Evolve.

### Memobilisasi

Ini bisa menjadi tantangan yang signifikan untuk menemukan jalur yang tepat untuk rehosting (lift dan shift) infrastruktur skala besar. Meskipun ada banyak [praktik terbaik](#) yang membantu, pilihan alat tergantung pada beberapa faktor, seperti jenis beban kerja, waktu henti yang terjangkau, dan persyaratan sistem operasi. Kami menyarankan Anda menggunakan [AWS Application Migration Service](#) untuk melakukan rehost.

#### Layanan Migrasi Aplikasi AWS

Anda dapat menggunakan Layanan Migrasi Aplikasi untuk mengangkat dan menggeser server fisik, virtual, atau cloud dengan cepat tanpa masalah kompatibilitas, dampak kinerja, atau jendela cutover yang panjang. Layanan Migrasi Aplikasi terus mereplikasi server sumber Anda ke akun AWS Anda. Kemudian, ketika Anda siap untuk bermigrasi, Layanan Migrasi Aplikasi secara otomatis mengonversi dan meluncurkan server Anda di AWS dengan waktu henti minimal. Untuk informasi selengkapnya, lihat [Apa itu Layanan Migrasi Aplikasi AWS?](#) Panduan Pengguna Layanan Migrasi Aplikasi.

## Orkestrator Hub Migrasi AWS

[AWS Migration Hub Orchestrator](#) menyederhanakan dan mengotomatiskan migrasi server dan aplikasi perusahaan ke AWS dengan menggunakan Layanan Migrasi Aplikasi. Ini menyediakan satu lokasi untuk menjalankan dan melacak migrasi Anda. Anda dapat menggunakan Migration Hub Orchestrator untuk memigrasikan aplikasi NetWeaver berbasis SAP — seperti S/4HANA, BW/4HANA, SAP ECC di HANA, dan lainnya—ke AWS dan rehost aplikasi kustom yang didukung ke Amazon EC2. Migration Hub Orchestrator menawarkan template untuk membuat alur kerja migrasi yang dapat disesuaikan agar sesuai dengan persyaratan migrasi unik Anda. Selain itu, Migration Hub Orchestrator mengotomatiskan langkah-langkah dalam alur kerja yang Anda pilih dan menampilkan status migrasi.

## VM Import/Export

[AWS VM Import/Export](#) memungkinkan Anda mengimpor gambar VM dari lingkungan virtualisasi yang ada ke Amazon EC2, lalu mengekspornya kembali. Hal ini memungkinkan Anda untuk bermigrasi aplikasi dan beban kerja ke Amazon EC2, menyalin katalog citra VM Anda ke Amazon EC2, atau membuat repositori citra VM untuk cadangan dan pemulihan bencana. Untuk informasi selengkapnya, lihat [Apa itu Impor/Ekspor VM?](#) di Panduan Pengguna Amazon EC2.

Setelah menilai beban kerja untuk migrasi, buat rencana migrasi yang menguraikan strategi migrasi, garis waktu, dan biaya yang terlibat dalam proses migrasi. Anda dapat menggunakan [AWS Pricing/TCO Tools](#) untuk memperkirakan penghematan biaya menjalankan aplikasi Anda di AWS. Anda juga dapat menggunakan [Application Discovery Service](#) untuk mengidentifikasi layanan AWS yang tepat untuk meng-host beban kerja yang dimigrasi.

## Migrasi

Migrasi beban kerja Windows ke AWS melibatkan beberapa fase, termasuk perencanaan migrasi, penilaian kesiapan, dan fase implementasi migrasi. Fase migrasi adalah fase terakhir, yang melibatkan migrasi beban kerja Windows ke AWS. Berikut adalah beberapa langkah yang perlu dipertimbangkan selama fase migrasi:

- Siapkan lingkungan AWS — Sebelum memulai proses migrasi, Anda harus menyiapkan lingkungan AWS dengan membuat Amazon Machine Image (AMI) dan menyiapkan VPC tempat Anda memigrasikan beban kerja.
- Pilih alat migrasi — Ada berbagai metode migrasi untuk dipilih, termasuk Migration Hub, Application Migration Service, dan VM Impor/Export. Pilih metode yang paling sesuai dengan kebutuhan Anda.

- Konfigurasi migrasi — Konfigurasi migrasi dengan memilih server sumber dan menentukan jenis instans target, penyimpanan, dan pengaturan jaringan.
- Lakukan migrasi — Setelah konfigurasi selesai, lakukan migrasi. Prosesnya melibatkan mereplikasi data, menguji beban kerja yang dimigrasi, dan melakukan pemotongan akhir untuk beralih ke beban kerja yang dimigrasi. Alat migrasi yang Anda pilih di atas akan memandu Anda melalui langkah-langkah ini.
- Validasi migrasi — Setelah migrasi selesai, validasi bahwa beban kerja yang dimigrasi berfungsi seperti yang diharapkan. Lakukan pengujian dan pastikan bahwa persyaratan keamanan dan kepatuhan terpenuhi.
- Optimalkan beban kerja yang dimigrasi — Optimalkan beban kerja yang dimigrasi dengan mengubah ukuran instance, mengonfigurasi auto-scaling, dan menerapkan strategi penghematan biaya seperti Instans Cadangan atau Instans Spot.
- Pantau dan kelola beban kerja yang dimigrasi — Pantau dan kelola beban kerja yang dimigrasi secara terus menerus untuk memastikan kinerja dan keamanan yang optimal. Anda dapat menggunakan [Amazon CloudWatch](#) untuk pemantauan.

## Migrasi server file

Penyimpanan adalah komponen penting untuk setiap beban kerja yang Anda jalankan. AWS memiliki sejumlah opsi untuk menyimpan file di cloud, termasuk penyimpanan blok, file, dan objek. Untuk beban kerja Microsoft, opsi yang paling umum adalah opsi penyimpanan blok dan file. Bagian ini menyediakan strategi untuk membantu Anda memigrasikan penyimpanan untuk beban kerja Microsoft ke AWS Cloud dan memandu Anda melalui migrasi server file Anda.

## Menilai

Ada tiga jenis penyimpanan utama: objek, blok, dan penyimpanan file. AWS menawarkan portofolio layanan penyimpanan yang luas yang dapat dikategorikan di bawah masing-masing layanan ini. Migrasi yang berhasil bergantung pada pemahaman kebutuhan Anda saat ini dan kemudian [membandingkannya](#) dengan berbagai layanan penyimpanan AWS untuk mengukur apa yang paling cocok untuk Anda. Memilih teknologi yang tepat untuk beban kerja Anda adalah kunci kesuksesan jangka panjang. Kami menyarankan Anda menghindari mencoba mencocokkan dengan apa yang Anda gunakan saat ini untuk penyimpanan. Sebagai gantinya, kami menyarankan Anda untuk melihat apa saja semua opsi yang tersedia dan memilih opsi yang paling masuk akal untuk mengoptimalkan biaya dan kinerja beban kerja Microsoft Anda. Misalnya, pertimbangkan server file lokal besar yang memerlukan penyimpanan blok lokal. Di AWS, pilihan optimal adalah

memindahkannya ke [Amazon FSx](#) untuk mendapatkan kinerja yang sama dengan yang Anda miliki untuk server file Anda, sambil menghapus beban berat yang tidak terdiferensiasi dalam mengelola server file dan penyimpanan backend.

TCO adalah item utama untuk dievaluasi saat Anda menilai opsi penyimpanan mana yang paling cocok untuk Anda. Ingatlah bahwa menggunakan layanan terkelola AWS untuk membantu mengurangi biaya pengoperasian dapat membantu Anda memilih solusi penyimpanan keseluruhan yang tepat di AWS. Untuk meminta penilaian penyimpanan, hubungi kami di [migration-evaluator@amazon.com](mailto:migration-evaluator@amazon.com). Spesialis penyimpanan akan membantu Anda menilai beban kerja Anda, memetakan beban kerja Anda ke layanan penyimpanan AWS yang paling tepat, dan memberi Anda perkiraan biaya terarah. Penilaian penyimpanan memiliki tiga fase:

- Anda memulai proses penemuan dengan menginstal kolektor tanpa agen atau menerima output dari alat yang ada yang diatur dalam file datar.
- Anda membiarkan proses penemuan berjalan selama 7-60 hari.
- Kolektor penyimpanan menganalisis data dari alat penemuan, dan kemudian mengusulkan solusi penyimpanan target dan memberikan perkiraan biaya terarah untuk solusi tersebut.

Jika biayanya sedikit lebih tinggi untuk opsi penyimpanan, pertimbangkan apakah opsi penyimpanan itu mengurangi biaya keseluruhan dalam jangka panjang dan cari tahu apa yang harus dilakukan tim Anda untuk menjaga keamanan dan keandalan penyimpanan Anda. Ini bisa menjadi solusi jangka panjang yang tepat untuk beban kerja Anda.

Ketika Anda menilai solusi yang tepat, penting untuk melihat kinerja dan biaya. Anda dapat menggunakan alat seperti [Windows Performance Monitor](#) untuk mengidentifikasi IOPS, throughput, dan kebutuhan kinerja lain dari beban kerja Anda dan kemudian menerapkan pengujian yang sama pada solusi AWS yang Anda pilih untuk beban kerja Anda. Selain itu, Anda dapat menggunakan CloudWatch agen untuk [melihat metrik Monitor Kinerja di server Windows](#) dan menganalisis metrik beban kerja Anda sebelum memasukkan beban kerja tersebut ke dalam produksi.

## Identifikasi layanan penyimpanan AWS yang paling sesuai dengan kebutuhan Anda

Pilihan layanan penyimpanan biasanya tergantung pada kasus penggunaan Anda, kebutuhan aplikasi, keakraban, profil kinerja, dan kemampuan manajemen data Anda. Pertimbangkan hal berikut:

- Amazon Simple Storage Service (Amazon S3) — [Amazon S3](#) adalah penyimpanan objek yang dibuat untuk menyimpan dan mengambil sejumlah data dari mana saja. Amazon S3 menawarkan

berbagai kelas penyimpanan yang dapat Anda pilih berdasarkan akses data, ketahanan, dan persyaratan biaya beban kerja Anda. Anda dapat menerapkan akses berbasis file ke Amazon S3 dengan menggunakan AWS Storage [Gateway](#). Ini memungkinkan Anda untuk mengambil keuntungan dari penyimpanan biaya rendah Amazon S3, sementara tidak harus sepenuhnya menulis ulang aplikasi yang menggunakan Server Message Block (SMB).

- Amazon Elastic Block Store (Amazon EBS) — [Amazon EBS](#) menyediakan volume penyimpanan tingkat blok untuk digunakan dengan instans EC2. Volume EBS berfungsi seperti perangkat blok mentah yang tidak terformat. Anda dapat memasang volume ini sebagai perangkat di instans Anda. Volume Amazon EBS yang terpasang pada sebuah instans dipaparkan sebagai volume penyimpanan yang tetap ada secara independen terlepas dari masa pakai instans.
- Amazon FSx — Amazon FSx menawarkan empat sistem file yang berbeda: NetApp ONTAP, OpenZFS, Windows File Server, dan Lustre. Untuk panduan tentang memilih sistem yang tepat, lihat [Memilih Sistem File Amazon FSx dalam dokumentasi](#) Amazon FSx. Amazon FSx menawarkan solusi penyimpanan file terkelola dalam berbagai jenis sistem file untuk memungkinkan Anda memigrasikan beban kerja Microsoft ke AWS dan menghapus beberapa overhead operasional dari staf TI Anda. Hal ini memungkinkan TI untuk fokus pada driver bisnis penting lainnya.
- AWS Snow Family — Jika Anda memiliki skala data petabyte untuk dipindahkan ke AWS, pertimbangkan untuk menggunakan solusi penyimpanan dari [AWS Snow](#) Family. Meskipun penyimpanan Anda tidak akan bergantung pada perangkat AWS Snow Family untuk masa pakai data jangka panjang, penyimpanan dapat membantu Anda menyemai kumpulan data besar ke AWS offline dengan menggunakan perangkat AWS Snowcone, AWS Snowball, atau AWS Snowmobile. Untuk informasi selengkapnya, lihat [database SQL besar yang bermigrasi dengan mulus menggunakan postingan AWS Snowball dan AWS](#) di DataSync AWS Storage Blog.

Kami menyarankan Anda melakukan pengujian dengan menggunakan alat pengujian stres/beban sebelum memindahkan data produksi, setelah Anda mengidentifikasi layanan penyimpanan untuk beban kerja Anda. Misalnya, jika Anda memindahkan database SQL Anda di Amazon FSx for Windows File Server, Anda dapat menggunakan [Microsoft SQL](#) Server Distributed Replay. Demikian pula, Anda dapat menggunakan [DISKSPD](#) untuk IOPS umum dan throughput.

## Memobilisasi

Setelah Anda mengidentifikasi layanan penyimpanan, langkah selanjutnya adalah memilih alat untuk transfer data. Beberapa alat tersedia, termasuk solusi lama seperti [Robocopy](#) dan alat yang lebih modern seperti [AWS DataSync](#). DataSync mencakup sejumlah kontrol yang tidak tersedia di alat seperti Robocopy, seperti transfer terjadwal dan kontrol pelambatan jaringan yang lebih mudah

untuk membantu memigrasikan data Anda tanpa memengaruhi lalu lintas jaringan Anda secara keseluruhan. Untuk informasi selengkapnya tentang migrasi yang berhasil diselesaikan dengan Sinkronisasi Data, lihat [testimonial pelanggan di](#) pelanggan AWS. DataSync

Jika Anda merasa lebih nyaman dengan Robocopy, Anda dapat menggunakannya untuk memigrasikan data Anda ke AWS. Kami menyarankan Anda meninjau panduan ini tentang cara mengoptimalkan [kinerja transfer file](#). Panduan ini dapat membantu Anda menghindari masalah selama migrasi. Jika Anda menggunakan Robocopy dengan sistem file yang mengaktifkan deduplikasi, lihat Deduplikasi [data di Panduan Pengguna Amazon FSx Windows dan Pemecahan Masalah Kerusakan Deduplikasi Data](#) dalam dokumentasi Microsoft untuk menghindari masalah dengan kerusakan data.

[AWS Storage Gateway](#) dapat memigrasikan data ke AWS dengan tiga cara: file, volume, dan kaset virtual. Anda dapat menginstal Storage Gateway pada hypervisor VMware atau Hyper-V yang berjalan di lokasi, instans Amazon EC2 di Amazon VPC, atau perangkat perangkat keras khusus.

Storage Gateway dapat membantu Anda menjembatani kesenjangan dari lokal ke AWS dan membantu mengurangi biaya. Anda dapat menggunakan Storage Gateway untuk mengimplementasikan migrasi secara bertahap dan menggunakannya untuk mengganti perangkat dan kaset cadangan lokal dengan pustaka pita virtual (VTL). Anda juga dapat menggunakan Storage Gateway sebagai solusi penyimpanan arsip untuk mulai memigrasikan hanya file lokal yang tidak terpakai ke AWS sebagai fase pertama migrasi Anda. Ada sejumlah opsi untuk menggunakan Storage Gateway untuk meng-host beban kerja Microsoft Anda di AWS.

## Migrasi

DataSync dan Robocopy keduanya dilengkapi untuk melestarikan daftar kontrol akses jaringan (ACL, juga dikenal sebagai Windows ACL). Sebelum memulai migrasi, sebaiknya Anda mengambil salinan cadangan ACL dengan menggunakan [icacls](#) dan meninjau sumber daya berikut:

- [Memigrasi berbagi file lokal ke Amazon fsX untuk ONTAP NetApp \(AWS Storage Blog\)](#)
- [Memigrasi penyimpanan file yang ada ke Amazon FSx \(Panduan Pengguna Amazon FSx Windows\)](#)
- [Mentransfer file dari lokasi lokal ke AWS dan kembali tanpa meninggalkan VPC Anda menggunakan AWS DataSync \(AWS Storage Blog\)](#)
- [Migrasikan kumpulan data kecil dari lokasi ke Amazon S3 menggunakan AWS SFTP \(AWS Prescriptive Guidance\)](#)

# Migrasi SQL Server

Dalam perjalanan Anda ke cloud, Anda memiliki beberapa opsi untuk memigrasikan lingkungan SQL Server Anda ke AWS. [Migrasi yang berhasil didasarkan pada pembuatan inventaris terperinci dari beban kerja SQL Server Anda dan dependensinya, mengidentifikasi skema otentikasi Anda, menangkap persyaratan ketersediaan tinggi dan pemulihan bencana \(HADR\) Anda, menilai target kinerja Anda, dan mengevaluasi opsi lisensi Anda.](#) Inventaris ini membantu Anda menentukan platform basis data target dan menentukan opsi migrasi Anda.

Anda memiliki banyak opsi untuk dipertimbangkan saat memigrasikan beban kerja SQL Server Anda ke AWS, masing-masing menghasilkan harga/kinerja yang dioptimalkan, pengalaman pengguna yang lebih intuitif, dan TCO yang lebih rendah. Anda dapat memilih untuk menggunakan SQL Server berikut ini: [Amazon EC2](#), [Amazon RDS for SQL Server](#), atau [Amazon RDS Custom untuk SQL Server](#).

## Menilai

Untuk mengimplementasikan migrasi yang sukses, penting untuk mengevaluasi infrastruktur yang ada dan memahami fitur utama yang diperlukan untuk lingkungan Anda. Kami menyarankan Anda meninjau area utama berikut sebelum memilih paket migrasi:

- Tinjau infrastruktur yang ada — Tinjau infrastruktur SQL Server Anda yang ada dengan menggunakan data yang dikumpulkan dalam fase penemuan migrasi Anda (lihat [penemuan lingkungan Windows](#)). Kami menyarankan Anda menggunakan ukuran yang ditentukan Microsoft untuk infrastruktur SQL Server di AWS. Memahami pemanfaatan instans SQL Server lokal Anda saat ini—termasuk memori, CPU, IOPS, dan throughput—sangat penting untuk mengukur instans SQL Server Anda dengan benar di AWS.
- Tinjau lisensi yang ada — Anda dapat memanfaatkan AWS [Optimizing and Licensing Assessment \(AWS OLA\) yang saling melengkapi untuk membangun strategi migrasi dan lisensi](#) di AWS. AWS OLA memberi Anda laporan yang memodelkan opsi penerapan Anda menggunakan hak lisensi yang ada. Hasil ini dapat membantu Anda menjelajahi penghematan biaya yang tersedia di seluruh opsi lisensi AWS yang fleksibel.
- Tinjau arsitektur SQL Server yang ada — Jika Anda menggunakan kluster failover SQL Server dengan penyimpanan bersama atau arsitektur grup SQL Server Always On Availability, maka memahami persyaratan arsitektur ketersediaan tinggi saat ini akan membantu Anda menentukan opsi [penerapan SQL](#) Server di AWS.



- Kembangkan strategi pencadangan — Anda dapat menggunakan cadangan asli di SQL Server untuk mencadangkan database Anda ke cloud. Ada berbagai opsi untuk mencadangkan database ke Amazon EBS, Amazon FSx untuk Windows File Server, Amazon FSx NETAPP ONTAP, dan Amazon S3 menggunakan Storage Gateway. Selain itu, Anda dapat mencadangkan instance SQL Server Anda dengan menggunakan pendekatan snapshot. Untuk informasi selengkapnya tentang pencadangan SQL Server, lihat [opsi Backup dan restore untuk SQL Server di Amazon EC2 dalam dokumentasi AWS Prescriptive Guidance](#).
- Memahami kebutuhan pemulihan bencana (DR) — Jika Anda memindahkan beban kerja SQL Server yang ada ke AWS, maka Anda dapat menggunakan Wilayah sekunder dan menghubungkan kedua Wilayah dengan menggunakan Transit Gateway (yang memungkinkan terjadinya replikasi). Anda dapat menggunakan arsitektur grup ketersediaan terdistribusi SQL Server dalam edisi SQL Enterprise untuk menyiapkan DR, atau Anda dapat menggunakan pengiriman log berdasarkan persyaratan RTO dan RPO Anda. Selain itu, Anda dapat menggunakan AWS Elastic Disaster Recovery (AWS DRS) sebagai implementasi aktif/pasif di mana Anda meninggalkan DR sebagai lingkungan failover. Untuk informasi selengkapnya, lihat [Arsitek pemulihan bencana untuk SQL Server di AWS: Bagian 1](#) posting di AWS Database Blog.

## Memobilisasi

Ada [tiga opsi migrasi utama](#) yang kami sarankan Anda pertimbangkan untuk beban kerja SQL Server Anda:

- Rehosting (angkat dan geser) — Ini melibatkan migrasi database SQL Server lokal Anda ke SQL Server pada instans EC2 di AWS Cloud. Pendekatan ini berguna jika migrasi yang lebih cepat ke AWS adalah prioritas Anda.
- Pembuatan ulang (angkat dan bentuk ulang) — Ini melibatkan migrasi database SQL Server lokal Anda ke [Amazon RDS](#) for SQL Server di AWS Cloud. Replatforming paling cocok untuk saat Anda ingin terus menggunakan SQL Server tetapi ingin membongkar tugas angkat berat yang tidak terdiferensiasi, seperti instalasi, konfigurasi, penambalan, peningkatan, dan pengaturan ketersediaan tinggi. Untuk perbandingan fitur SQL Server di Amazon EC2, Amazon RDS, dan Amazon RDS Custom, lihat Memilih [antara Amazon EC2 dan Amazon RDS dalam dokumentasi AWS Prescriptive Guidance](#).
- Refactoring (re-architect) — Ini biasanya melibatkan perubahan aplikasi dan modernisasi dengan menggunakan database open-source atau database yang dibangun untuk cloud. [Dalam skenario ini, Anda memodernisasi database SQL Server lokal untuk menggunakan Amazon RDS for MySQL, Amazon RDS for PostgreSQL, atau Amazon Aurora](#). Dengan pindah ke database sumber



terbuka, Anda dapat mengurangi biaya lisensi dan mencegah periode penguncian vendor yang tidak perlu dan audit lisensi.

## Migrasi

Saat Anda memigrasikan beban kerja SQL Server ke AWS, pertimbangkan item berikut tentang konfigurasi dan perkakas.

### Rehosting

Rehosting adalah [homogen](#). Pilih pendekatan ini ketika Anda ingin memigrasikan database SQL Server apa adanya tanpa mengubah perangkat lunak atau konfigurasi database. Misalnya, dalam migrasi warisan skala besar, Anda mungkin ingin bergerak cepat untuk memenuhi tujuan bisnis Anda dan memilih untuk meng-host kembali sebagian besar aplikasi Anda.

### Migrasi SQL Server menggunakan Amazon EC2

Jika Anda bermigrasi ke Amazon EC2, Anda dapat membawa lisensi SQL Server yang ada. Ini dikenal sebagai model Bring Your Own License (BYOL). Atau, Anda dapat membeli instans yang disertakan lisensi dari AWS. Untuk informasi selengkapnya, lihat [Pengoptimalan biaya dengan SQL BYOL menggunakan instans Windows yang disertakan lisensi di Host Khusus Amazon EC2 posting](#) di AWS Cloud Operations & Migrations Blog. Opsi BYOL memungkinkan Anda mengurangi biaya dengan menggunakan lisensi SQL Server yang ada. [AWS License Manager](#) membantu mengontrol alokasi lisensi yang tersedia saat membuat instance VM dengan SQL Server di Amazon EC2. License Manager membantu memastikan kepatuhan terhadap aturan perizinan yang Anda tentukan.

Anda dapat meng-host ulang SQL Server ke instans EC2 penyewaan bersama (default) dengan menggunakan BYOL hanya jika Anda memiliki Microsoft Software Assurance (SA). Jika Anda tidak memiliki SA pada lisensi SQL Anda, Anda dapat melakukan rehost ke Host [Khusus Amazon EC2](#), selama lisensi dibeli sebelum 1 Oktober 2019, atau ditambahkan sebagai true-up di bawah Pendaftaran Perusahaan aktif yang berlaku sebelum 1 Oktober 2019.

Ada cara untuk memigrasikan database SQL Server ke instans Amazon EC2 dengan menggunakan fitur SQL Server seperti pencadangan dan pemulihan, pengiriman log, dan grup ketersediaan Selalu Aktif. Opsi ini sesuai jika Anda memigrasikan satu database atau kumpulan database ke instance SQL Server baru yang berjalan di Amazon EC2. Opsi ini adalah database-native dan tergantung pada versi dan edisi SQL Server tertentu. Selain migrasi database, Anda juga dapat diminta untuk

melakukan langkah-langkah untuk memigrasikan objek seperti login, pekerjaan, email database, dan server tertaut.

Pendekatan berikut tersedia untuk rehosting database SQL Server Anda di AWS:

- Rehosting server dengan menggunakan Layanan [Migrasi Aplikasi](#) atau [AWS Database Migration Service \(AWS DMS\)](#)
- [Pencadangan dan pemulihan SQL Server](#)
- [Replikasi transaksional SQL Server](#)
- [Memperluas grup ketersediaan Anda ke cloud](#)
- [AWS DMS](#)
- [Pengiriman log](#)

Anda juga dapat menggunakan [AWS Launch Wizard for SQL Server](#) untuk memandu Anda melalui ukuran, konfigurasi, dan penerapan Microsoft SQL Server di Amazon EC2. Ini mendukung instance tunggal SQL Server dan penerapan HA di Amazon EC2. Untuk mempelajari selengkapnya, lihat [AWS Launch Wizard for SQL Server](#).

## Migrasi SQL Server menggunakan Layanan Migrasi Aplikasi

Layanan Migrasi Aplikasi adalah pilihan yang baik jika Anda ingin mengangkat dan memindahkan satu atau lebih mesin skala besar dari lingkungan lokal ke AWS tanpa mengubah versi SQL Server, sistem operasi, atau kode dalam database dengan waktu henti mendekati nol atau minimal. Anda dapat menggunakan [AWS Application Migration Service](#) untuk mengangkat dan menggeser server fisik, virtual, atau cloud dengan cepat tanpa masalah kompatibilitas, dampak kinerja, atau jendela cutover yang panjang. Untuk panduan tentang memigrasikan database SQL Server dari lingkungan lokal ke instans Amazon EC2 menggunakan Layanan Migrasi Aplikasi, lihat Memigrasi [database Microsoft SQL Server ke AWS Cloud dalam](#) dokumentasi AWS Prescriptive Guidance. Anda juga dapat merujuk ke [praktik terbaik](#) saat menggunakan Layanan Migrasi Aplikasi untuk memigrasikan beban kerja database Microsoft SQL Server ke AWS.

## SQL Server di Linux

Mesin database SQL Server pada dasarnya berjalan dengan cara yang sama pada Windows Server dan Linux. Namun, ada beberapa perubahan pada tugas-tugas tertentu saat menggunakan Linux. [Launch Wizard](#) dapat membantu Anda menyesuaikan diri dengan perubahan ini dan mengonfigurasi

solusi yang sangat tersedia. Jika Anda memiliki keahlian administrasi Linux internal, rehosting ke Amazon EC2 Linux adalah pilihan yang baik untuk menghemat biaya lisensi Windows Server. Pertimbangkan untuk menggunakan [asisten replatforming Windows ke Linux untuk alat Microsoft SQL Server Databases](#) untuk mengotomatiskan proses ini. Untuk informasi selengkapnya, lihat [Memigrasikan database Microsoft SQL Server lokal ke Microsoft SQL Server di Amazon EC2 yang menjalankan Linux dalam dokumentasi AWS Prescriptive Guidance](#).

## Pembuatan ulang

Replatforming adalah pendekatan [homogen](#) yang paling cocok untuk mengurangi waktu yang Anda habiskan untuk mengelola instance database dengan menggunakan penawaran database yang dikelola sepenuhnya. Database yang dikelola sepenuhnya di Amazon RDS for SQL Server membatasi Anda untuk mengakses sistem operasi, volume sistem, atau pemasangan driver kustom yang mendasarinya. Untuk informasi selengkapnya, lihat [Amazon RDS for Microsoft SQL Server di Panduan Pengguna Amazon RDS](#). Jika kemampuan database yang dikelola sepenuhnya diperlukan untuk kasus penggunaan Anda atau jika Anda ingin menggunakan lisensi server SQL yang ada, pertimbangkan replatforming ke [Amazon RDS Custom for SQL Server](#).

Opsi Bring Your Own Media (BYOM) tersedia untuk Amazon RDS Custom untuk SQL Server. BYOM memungkinkan Anda untuk menggunakan media instalasi dan lisensi Anda sendiri, tetapi lisensi harus mematuhi persyaratan Mobilitas Lisensi Microsoft. Anda dapat memplatform ulang SQL Server ke Amazon RDS untuk SQL Server atau ke Amazon RDS Custom untuk SQL Server. Pilihannya tergantung pada apakah Anda memerlukan akses ke sistem operasi yang mendasarinya, memerlukan kustomisasi basis data, atau ingin memanfaatkan lisensi SQL Server yang ada dengan menggunakan BYOM.

Metode berikut tersedia untuk memigrasi SQL Server ke Amazon RDS for SQL Server:

- [Log pengiriman menggunakan PowerShell](#) atau [Log pengiriman menggunakan TSQL](#)
- [Pencadangan dan pemulihan SQL Server](#)
- [Replikasi transaksional](#)
- [AWS DMS](#)

Untuk memplatform ulang database SQL Server agar berjalan di Amazon RDS for SQL Server, pertimbangkan untuk menggunakan pendekatan yang disediakan di Amazon RDS for [SQL Server resource](#). Untuk informasi tentang cara memigrasi akhir beban kerja dukungan, lihat [Migrasi akhir dukungan basis data Microsoft SQL Server ke Amazon RDS for SQL Server yang diposting dengan](#)

[percaya diri di Blog Database AWS](#). Untuk informasi tentang database lokal, lihat [Memigrasi database lokal ke Amazon RDS Custom for SQL Server](#) di Panduan Pengguna Amazon RDS.

## Refactoring

[Refactoring bersifat heterogen](#). Pilih pendekatan ini ketika Anda siap untuk merestrukturisasi, menulis ulang, dan merancang ulang database dan aplikasi Anda untuk memanfaatkan penawaran open-source dan database. built-for-the-cloud [Jika Anda terbuka untuk refactoring database dan aplikasi masing-masing, Anda dapat memodernisasi beban kerja SQL Server Anda ke Amazon RDS untuk MySQL, Amazon RDS untuk PostgreSQL, Amazon Aurora MySQL Compatible Edition, atau Amazon Aurora PostgreSQL Edition yang kompatibel dengan Amazon Aurora](#). Anda dapat melakukan refactor tergantung pada banyak jadwal modernisasi dan persyaratan kinerja.

Amazon RDS for MySQL dan Amazon RDS for PostgreSQL adalah penawaran database yang dikelola sepenuhnya untuk database sumber terbuka masing-masing. Amazon Aurora adalah sistem manajemen basis data relasional (RDBMS) yang dibangun untuk cloud dengan kompatibilitas MySQL dan PostgreSQL penuh. Aurora memiliki sistem penyimpanan yang toleran terhadap kesalahan dan memberi Anda kinerja dan ketersediaan basis data kelas komersial dengan biaya sepersepuluh.

Anda juga dapat menggunakan [Amazon Aurora Tanpa Server](#) untuk menjalankan database di AWS tanpa mengelola kapasitas database. Amazon Aurora Serverless v2 menskalakan secara instan ke ratusan ribu transaksi dalam sepersekian detik. Anda hanya membayar untuk kapasitas yang dikonsumsi aplikasi Anda, dan Anda dapat menghemat hingga 90 persen biaya database dibandingkan dengan biaya kapasitas penyediaan untuk beban puncak.

Untuk memfaktorkan ulang database SQL Server Anda ke salah satu penawaran ini, pertimbangkan untuk menggunakan AWS [Schema Conversion Tool \(AWS SCT\) dengan AWS DMS](#). Untuk informasi selengkapnya, lihat [AWS SCT](#) di database Migrasi Microsoft SQL Server ke panduan AWS Cloud.

Jika tujuan Anda adalah mempercepat migrasi aplikasi dan database Anda ke AWS, pertimbangkan untuk menggunakan [Babelfish for Aurora PostgreSQL](#). Babelfish memungkinkan aplikasi yang awalnya ditulis untuk SQL Server untuk bekerja dengan Aurora dengan perubahan kode minimal. Akibatnya, upaya yang diperlukan untuk memodifikasi dan pindah ke Babelfish untuk aplikasi PostgreSQL Aurora yang dikembangkan untuk SQL Server 2019 atau yang lebih lama berkurang, yang mengarah ke refactoring yang lebih cepat, berisiko lebih rendah, dan lebih hemat biaya.

Pertimbangkan sumber daya berikut untuk bermigrasi dengan Babelfish:

- [Bermigrasi dari SQL Server ke Amazon Aurora menggunakan Babelfish](#) (AWS Database Blog)

- [Mempersiapkan migrasi Babelfish dengan laporan penilaian AWS SCT](#) (AWS Database Blog)
- [Migrasi dari SQL Server ke Aurora PostgreSQL menggunakan SSIS dan Babelfish](#) (AWS Database Blog)
- [Menggunakan Babelfish sebagai target untuk AWS Database Migration Service](#) (Panduan Pengguna AWS Database Migration Service)

## Sumber daya tambahan

- [Memigrasi SQL Server ke panduan preskriptif AWS](#) (AWS Prescriptive Guidance)
- [Strategi Migrasi dan Modernisasi untuk SQL Server Anda di AWS](#) (AWS Blog)

## Migrasi aplikasi.NET

Migrasi aplikasi.NET Anda ke AWS memungkinkan Anda membuat beban kerja yang sangat tersedia dengan kemampuan penskalaan elastis, mengurangi overhead operasi, dan meningkatkan kelincahan bisnis Anda dengan berfokus pada nilai pembeda Anda. Bagian ini berfokus pada berbagai opsi untuk menghosting aplikasi.NET Anda di AWS. Anda dapat memilih antara menggunakan VM, solusi terkelola seperti [AWS Elastic Beanstalk](#), mengontainerisasi kode Anda, atau memfaktorkan ulang kode Anda ke arsitektur berbasis layanan mikro atau tanpa server.

## Menilai

Memilih jalur migrasi untuk beban kerja.NET Anda bergantung pada faktor kunci berikut:

- Temukan versi.NET yang digunakan — Ada dua implementasi.NET berbeda yang didukung oleh Microsoft: .NET Framework (1.0-4.8) dan .NET (.NET Core 1.0-3.1 dan .NET 5 dan yang lebih baru). Keduanya berbagi banyak komponen yang sama dan dapat menjalankan kode aplikasi yang ditulis menggunakan bahasa pemrograman.NET yang berbeda (seperti C #, F #, dan VB.NET). Memilih strategi migrasi dan layanan hosting tergantung pada runtime yang digunakan sejak .NET Framework berjalan di Windows sedangkan .NET yang lebih baru adalah multi-platform. Untuk .NET Framework, Anda dapat meng-host pada OS Windows atau refactor kode Anda untuk menggunakan .NET yang lebih baru. .NET yang lebih baru juga dapat di-host pada layanan berbasis OS Linux. Saat memodernisasi beban kerja berbasis.NET Framework, Anda dapat menggunakan [Porting Assistant untuk.NET](#) atau [AWS Toolkit for .NET Refactoring untuk memindai](#) kode Anda dan menghasilkan laporan penilaian kompatibilitas. Dengan menemukan apakah ada API.NET Framework yang tidak kompatibel yang direferensikan oleh proyek Anda,

Anda dapat merencanakan kompleksitas proyek migrasi dan memutuskan apakah dan kapan akan memfaktorkan ulang kode Anda untuk menggunakan runtime yang lebih baru.

- Tinjau penerapan Anda saat ini — Periksa apakah beban kerja yang saat ini dimigrasi memiliki pipeline CI/CD yang sudah ada yang dapat diperbarui untuk menerapkan beban kerja yang sama ke cloud. Menggunakan pipeline build dan deploy yang ada dapat mengurangi waktu yang diperlukan untuk menerapkan aplikasi Anda ke cloud dengan mengotomatiskan langkah-langkah yang diperlukan untuk membangun, mengonfigurasi, dan menerapkan beban kerja Anda.
- Tinjau peta jalan Anda — Tergantung pada keadaan proyek saat ini, Anda mungkin sudah berencana untuk merancang ulang atau mendesain ulang aplikasi Anda. Setiap modernisasi yang dilakukan harus mempertimbangkan peta jalan produk. Misalnya, memutuskan untuk memasukkan kode yang ada atau memfaktorkan ulang arsitektur monolitik ke dalam layanan mikro idealnya merupakan bagian dari peta jalan produk dan selaras dengan upaya pengembangan lainnya.

## Memobilisasi

Ada tiga jalur migrasi berbeda yang perlu dipertimbangkan saat memigrasikan beban kerja.NET Anda ke AWS. Anda dapat memilih di antara berbagai opsi tergantung pada kompleksitas basis kode yang ada, waktu yang dialokasikan untuk migrasi, dan ukuran tim yang dialokasikan untuk mendukung upaya migrasi. Saat mempertimbangkan modernisasi sebagai bagian dari migrasi Anda, itu adalah praktik terbaik untuk diselaraskan dengan peta jalan produk.

- Rehost (angkat & geser) — Anda dapat memilih pendekatan ini jika prioritas Anda adalah migrasi yang lebih cepat ke AWS dengan sedikit atau tanpa perubahan. Anda dapat meng-host ulang situs web berbasis ASP.NET ke Internet Information Services (IIS) yang berjalan di instans Amazon EC2. [Anda dapat meng-host ulang aplikasi berbasis desktop Anda \(seperti Windows Presentation Foundation, Formulir Web, dan .NET MAUI\) ke salah satu platform komputasi pengguna akhir seperti Amazon 2.0 atau Amazon. AppStream WorkSpaces](#)
- Replatform — Replatforming paling cocok untuk saat Anda ingin meng-host aplikasi Anda menggunakan layanan terkelola tanpa membuat perubahan kode tetapi ingin mengurangi overhead operasional Anda dengan membongkar beban berat yang tidak berdiferensiasi seperti instalasi, patching, upgrade, dan manajemen instans. Strategi ini juga cocok untuk tim yang ingin pindah ke beban kerja berbasis kontainer. [Anda dapat memplatform ulang aplikasi yang ada ke Elastic Beanstalk, atau menggunakan container Docker yang dihosting di Amazon ECS, Amazon EKS, atau AWS App Runner.](#)

- Refactor — Pilih pendekatan ini jika Anda dapat menginvestasikan waktu dan upaya untuk membuat perubahan kode dan arsitektur yang mengurangi overhead operasional dan mencapai penskalaan yang lebih baik, ketersediaan tinggi, dan pemulihan bencana dengan menggunakan layanan cloud-native AWS. Refactoring melibatkan modernisasi basis kode Anda dengan mem-porting aplikasi.NET framework yang ada ke .NET (sebelumnya .NET Core) atau memodernisasi basis kode yang ada agar berjalan lebih baik di cloud. Anda dapat menggunakan [AWS SDK for .NET untuk](#) memanggil banyak layanan cloud AWS dari dalam kode .NET Anda. Alat seperti [Porting Assistant untuk.NET](#) dan [AWS Microservice Extractor untuk.NET dapat digunakan untuk](#) mem-port basis kode Anda dari .NET Framework ke .NET dan memecah aplikasi monolitik Anda menjadi layanan mikro. Dengan memfaktorkan ulang beban kerja.NET yang ada untuk berjalan di [Lambda](#), Anda dapat menggunakan komputasi tanpa server untuk menghindari penyediaan dan pengelolaan infrastruktur.

## Migrasi

Langkah-langkah migrasi beban kerja.NET Anda bergantung pada jalur migrasi yang Anda pilih selama tahap penilaian dan jenis aplikasi Anda.

### Rehost aplikasi.NET

Pilih jalur migrasi ini jika Anda ingin memigrasikan aplikasi Anda tanpa membuat perubahan kode apa pun tetapi ingin mendapatkan manfaat dari penskalaan otomatis, penyeimbangan beban, dan elastisitas di cloud. Untuk situs web berbasis Windows, rehosting biasanya berarti menjalankannya di Internet Information Services (IIS) di AWS. Untuk aplikasi berbasis desktop, Anda harus menginstal aplikasi dan memungkinkan pengguna untuk terhubung ke aplikasi dari luar.

### Layanan Informasi Internet di AWS

Internet Information Services (IIS) adalah server web Microsoft yang berjalan pada sistem operasi Windows dan digunakan untuk meng-host situs web dan layanan web. IIS dapat diinstal pada instans EC2 apa pun yang menjalankan Windows Server. Setelah IIS diaktifkan dan dikonfigurasi, Anda dapat menerapkan situs web dan layanan ASP.NET Anda dengan menggunakan mekanisme penerapan yang sama yang Anda gunakan untuk lingkungan lokal.

Jika Anda meng-host IIS pada instans Windows EC2, penting untuk mengikuti [AWS Well-Architected Framework](#) dengan menggunakan load balancing, grup Auto Scaling, dan penerapan Multi-AZ tergantung pada beban kerja dan kebutuhan HADR Anda. Sebaiknya gunakan [AWS Launch](#)



[Wizard](#) karena panduan ini memandu Anda melalui ukuran, konfigurasi, dan penerapan beban kerja Windows Server yang menjalankan sumber daya IIS di AWS. Launch Wizard menerapkan arsitektur yang sangat tersedia yang mencakup dua Availability Zone dengan komponen komputasi, jaringan, dan penyimpanan yang diperlukan untuk VPC yang baru dibuat atau yang sudah ada.

## Hosting aplikasi desktop di AWS

Banyak klien memiliki kebutuhan untuk mengakses aplikasi klien tebal berbasis Windows. Anda memiliki pilihan di antara tiga platform yang berbeda:

- [Amazon EC2](#) - Pilih opsi ini jika Anda ingin pengguna Anda terhubung ke lingkungan berbasis Server Windows dengan menggunakan Microsoft Remote Desktop. Dengan opsi ini Anda bertanggung jawab untuk menambal dan memelihara sistem operasi Anda. Anda juga harus membeli lisensi akses klien Layanan Desktop Jarak Jauh (RDS CAL) tambahan untuk pengguna Anda dan [Jaminan Perangkat Lunak \(SA\) aktif](#) Anda. Untuk informasi selengkapnya, lihat [Lisensi Microsoft di AWS](#) dalam dokumentasi AWS.
- [Amazon WorkSpaces](#) — Pilih opsi ini jika Anda memerlukan infrastruktur desktop virtual (VDI) yang dikelola sepenuhnya untuk pengguna Anda. Anda dapat menggunakan WorkSpaces untuk memberikan pengalaman Windows Desktop persisten kepada pengguna Anda. Anda juga dapat menyesuaikan WorkSpaces lingkungan dan menginstal aplikasi.NET dengan menggunakan gambar khusus, atau menggunakan [AWS System Manager](#) untuk mengirimkan aplikasi.NET Anda ke WorkSpaces lingkungan Anda. Pengguna dapat terhubung baik dengan menggunakan browser mereka atau [WorkSpacesklien Amazon](#).
- [Amazon AppStream 2.0](#) — Pilih opsi ini untuk menyediakan akses yang aman, andal, dan terukur ke aplikasi dan desktop non-persisten dari lokasi mana pun. Anda dapat menggunakan AppStream 2.0 untuk memungkinkan pengguna mengakses aplikasi.NET Anda dari web. [Jika Anda sudah memiliki RDS CAL dan SA aktif, maka Anda dapat menggunakan lisensi tersebut dengan AppStream 2.0 dengan menggunakan Mobilitas Lisensi.](#)

## Platform Ulang

Replatforming melibatkan perubahan lingkungan hosting Anda dengan sedikit atau tanpa perubahan kode. Pilih strategi ini untuk mengurangi biaya operasional Anda dan memanfaatkan kemampuan dan layanan cloud.



## AWS Elastic Beanstalk

Anda dapat menggunakan [Elastic Beanstalk](#) untuk memplatform ulang beban kerja.NET Framework Anda. Jika Anda mengemas aplikasi berbasis ASP.NET atau ASP.NET Core-based, maka Anda dapat dengan cepat menyebarkan dan mengelola aplikasi di AWS tanpa harus mempelajari infrastruktur yang menjalankan aplikasi tersebut. Ini mengurangi kompleksitas tanpa membatasi pilihan atau kontrol. Anda cukup mengunggah aplikasi Anda dan Elastic Beanstalk secara otomatis menangani detail penyediaan kapasitas, penyeimbangan beban, penskalaan, dan pemantauan kesehatan aplikasi.

Untuk mempelajari lebih lanjut, lihat sumber daya berikut:

- [Membuat dan menerapkan aplikasi.NET di Elastic Beanstalk](#) (Panduan Pengembang AWS Elastic Beanstalk)
- [Bekerja dengan.NET Core di Linux \(Panduan Pengembang AWS Elastic Beanstalk\)](#)
- [Dukungan Multi-Aplikasi dengan Domain Kustom untuk.NET dan AWS Elastic Beanstalk \(AWS Developer Tools Blog\)](#)

## Kontainerisasi aplikasi yang ada

Anda dapat menggunakan Amazon ECS atau Amazon EKS untuk meng-host aplikasi kontainer berbasis Docker Anda. AWS mengelola kedua layanan tersebut. Pilihan antara keduanya tergantung pada pengetahuan dan preferensi yang ada. Kedua opsi dapat menjalankan kontainer berbasis Linux atau wadah berbasis Windows.

Untuk mempelajari lebih lanjut, lihat sumber daya berikut:

- [Wadah Windows Amazon EC2](#) (Panduan Pengembang Amazon ECS)
- [Mengaktifkan dukungan Windows untuk kluster Amazon EKS Anda](#) (Panduan Pengguna Amazon EKS)
- [Menjalankan Windows Container dengan Amazon ECS di AWS Fargate](#) (AWS Blog)
- [Mempercepat waktu peluncuran kontainer Windows dengan pembuat Gambar EC2 dan strategi cache gambar](#) (AWS Blog)
- [Mulai cepat: CI/CD untuk Aplikasi.NET di AWS Fargate](#) (dokumentasi AWS)

Containerizing aplikasi berbasis .NET tergantung pada runtime .NET yang digunakan. Pertimbangkan hal berikut:

- .NET Framework berbasis aplikasi berjalan pada wadah Windows - Menambahkan dukungan Docker ke aplikasi yang ada dilakukan dengan membuat file Docker yang menguraikan bagaimana aplikasi perlu dikontainerisasi. Anda dapat menggunakan [AWS App2Container untuk dengan mudah mengkontainerisasi](#) dan memigrasikan aplikasi berbasis .NET Framework yang ada ke AWS. App2Container memindai server IIS Anda untuk menentukan file yang diperlukan dan mengekstrak aplikasi target untuk membuat gambar Docker. Anda juga dapat menggunakan App2Container untuk membuat artefak penerapan yang diperlukan untuk meng-host aplikasi Anda di AWS Cloud.
- .NET atau .NET Core — Selain menjalankan aplikasi web berbasis .NET yang lebih baru di Amazon ECS atau Amazon EKS, Anda juga dapat menggunakan [AWS App Runner](#). App Runner adalah solusi tanpa server yang dikelola sepenuhnya yang menjalankan kode atau gambar kontainer Anda dan mengelola penyeimbangan beban, penskalaan otomatis, pencatatan, sertifikat, dan jaringan.

## Refactor/Arsitek ulang kode yang ada

Pilih opsi ini jika Anda memiliki kebutuhan bisnis yang kuat untuk menambahkan fitur, skala, atau kinerja yang sulit dicapai di lingkungan aplikasi saat ini. Bergantung pada peta jalan aplikasi Anda, Anda dapat memilih untuk mengubah kode Anda untuk menggunakan kerangka kerja terbaru, layanan cloud-native, atau merancang ulang agar lebih baik dijalankan di cloud.

Opsi refactoring pertama yang tersedia adalah memigrasikan aplikasi .NET Framework Anda yang ada ke .NET. Pindah ke .NET memberi Anda manfaat berjalan di Linux, bukan Windows. Ini mengurangi total biaya lisensi Anda, memberi Anda kerangka kerja terbaru, dan menawarkan versi terbaru dari bahasa pemrograman .NET.

## AWS SDK for .NET

[AWS SDK for .NET](#) menyederhanakan penggunaan layanan AWS dengan menyediakan serangkaian pustaka yang konsisten dan akrab bagi pengembang .NET. AWS SDK menawarkan dukungan lintas platform dan didistribusikan menggunakan NuGet. Pengembang dapat menggunakan AWS SDK untuk dengan mudah memanggil layanan cloud dari kode .NET mereka, memenuhi persyaratan penyimpanan, antrian, otentikasi, dan konfigurasi aplikasi mereka.

## Modernisasi aplikasi.NET Framework

Anda dapat bermigrasi dari .NET Framework dengan menggunakan [Porting Assistant untuk .NET](#), yang memindai file kode Anda dan membuat laporan yang membantu merencanakan peta jalan migrasi portofolio aplikasi Anda. Porting Assistant juga dapat mengurangi overhead porting Anda dengan mengidentifikasi API dan paket .NET Core yang tidak kompatibel dan menemukan pengganti yang diketahui. [AWS Toolkit untuk .NET refactoring](#) adalah ekstensi Visual Studio yang mengurangi waktu dan upaya yang diperlukan bagi pengembang untuk memfaktorkan ulang aplikasi .NET lama menjadi alternatif berbasis cloud di AWS. Ini menilai kode sumber aplikasi untuk merekomendasikan jalur modernisasi yang mungkin seperti porting ke .NET Core, mengidentifikasi konfigurasi ketergantungan IIS dan Active Directory khusus Windows, melakukan modifikasi kode jika memungkinkan untuk mengaktifkan kompatibilitas Linux, dan membantu memvalidasi aplikasi refactored pada layanan AWS. Migrasi aplikasi .NET Framework ke .NET memungkinkan menjalankannya pada prosesor Graviton berbasis ARM64 untuk rasio harga terhadap kinerja yang lebih baik. Untuk informasi selengkapnya, lihat [.NET tentang Graviton on GitHub](#) dan [Graviton2 serta kontainer](#) dari Mengoptimalkan biaya dengan layanan berbasis AWS Graviton dalam dokumentasi AWS Workshop Studio.

## Monolith ke microservices

Banyak tim pengembangan ingin merancang ulang aplikasi monolitik mereka yang ada menjadi layanan mikro. Dengan beralih ke arsitektur berbasis layanan mikro, tim pengembangan Anda dapat meningkatkan kelincahan pengembangan, mengurangi biaya komputasi, menskalakan layanan secara individual, dan mengurangi waktu penerapannya. [AWS Microservice Extractor untuk .NET](#) menyederhanakan proses refactoring aplikasi monolitik lama menjadi arsitektur berbasis layanan mikro. Dengan mengidentifikasi komponen dan fungsionalitas pengelompokan, tim pengembangan dapat secara bertahap mengekstrak fungsionalitas dari aplikasi monolitik .NET Framework ke dalam layanan .NET.

## Refactor ke aplikasi tanpa server

[AWS Lambda](#) adalah layanan komputasi tanpa server dan berbasis peristiwa yang memungkinkan Anda menjalankan kode untuk hampir semua jenis aplikasi atau layanan backend tanpa menyediakan atau mengelola server. Anda dapat mengekstrak logika dari aplikasi yang ada untuk membuat alur kerja tanpa server berbasis peristiwa yang menskalakan secara otomatis saat diperlukan dengan menggunakan .NET dan Lambda. [Kasus penggunaan umum untuk Lambda](#) termasuk beban kerja berbasis peristiwa yang berjalan selama beberapa detik atau menit dengan berbagai kebutuhan penskalaan, seperti pemrosesan file, analitik, situs web, dan aplikasi

seluler. Untuk informasi selengkapnya, lihat [Membangun fungsi Lambda dengan C# di Panduan Pengembang](#) Lambda.

## Sumber daya tambahan

- [Amazon CodeCatalyst](#) ( CodeCatalyst dokumentasi Amazon)
- [AWS Toolkit untuk Azure \(dokumentasi DevOps AWS\)](#)
- [Menyiapkan pipeline CI/CD dengan mengintegrasikan Jenkins dengan AWS CodeBuild dan AWS CodeDeploy \(AWS Blog\)](#) DevOps
- [Tentang AWS Deploy Tool untuk .NET](#) (AWS GitHub)
- [.NET di AWS](#) (dokumentasi AWS)
- [aws/dotnet](#) () GitHub

## Migrasi cluster failover Windows

[Cluster failover Microsoft](#) adalah sekelompok server dengan sebagian besar penyimpanan bersama di antara mereka. Anda dapat menggunakan cluster failover untuk memfasilitasi ketersediaan tinggi untuk aplikasi dan layanan Anda. Anda juga dapat memigrasikan cluster failover Anda ke AWS Cloud untuk memanfaatkan keandalan, kinerja, dan TCO yang lebih rendah.

Cluster failover Windows bekerja secara berbeda di cloud daripada di lingkungan lokal. Penting untuk dicatat bahwa hanya cluster multi-subnet yang dapat digunakan di cloud. Tidak seperti di lingkungan lokal, alamat IP di cluster failover Windows ditetapkan ke Elastic Network Adapter (ENA) dan bukan di tingkat sistem operasi. Di lingkungan lokal, sistem operasi menangani penetapan alamat IP, tetapi penyedia cloud (AWS) menangani penetapan alamat IP di cloud. Karena pengelompokan failover adalah fitur tingkat sistem operasi, ia tidak dapat mengendalikan failover IP. Oleh karena itu, IP yang sama tidak dapat gagal di antara node. Untuk mengatasinya, Anda dapat menggunakan cluster multi-subnet di mana cluster gagal ke IP sekunder. IP sekunder ditugaskan ke ENA di subnet lain dan dapat online. Untuk informasi selengkapnya, lihat [Failover Clustering Networking Basics and Fundamentals](#) dalam dokumentasi Microsoft.

Migrasi cluster failover Windows ke AWS dapat menjadi proses yang kompleks, tetapi dengan perencanaan dan implementasi yang cermat, hal itu dapat dilakukan dengan gangguan minimal pada operasi bisnis Anda. Misalnya, setiap aplikasi dikonfigurasi secara berbeda pada cluster failover, jadi sangat penting untuk memahami kebutuhannya dan kemudian mencari tahu bagaimana mereka dapat dipenuhi di cloud sebelumnya. Prosesnya melibatkan langkah-langkah berikut:

- Memastikan bahwa semua node cluster menjalankan versi Windows yang sama dan semua pembaruan yang diperlukan
- Mengkonfigurasi kuorum cluster
- Memastikan bahwa semua aplikasi dan data dicadangkan dan dapat dipulihkan selama migrasi

## Menilai

Fase penilaian adalah langkah penting dalam proses migrasi cluster failover ke AWS. Selama fase ini, Anda mengumpulkan informasi tentang lingkungan Anda saat ini, menentukan kelayakan migrasi ke AWS, dan mengidentifikasi potensi tantangan atau risiko apa pun. Kami menyarankan Anda mengikuti langkah-langkah ini selama fase penilaian:

- Menilai kesiapan aplikasi Anda — Tentukan apakah aplikasi Anda dapat dimigrasikan ke AWS tanpa modifikasi atau apakah aplikasi tersebut perlu diperbarui atau ditulis ulang untuk memanfaatkan layanan cloud-native.
- Evaluasi persyaratan jaringan dan keamanan Anda — Tentukan persyaratan jaringan dan keamanan Anda, termasuk konfigurasi firewall, penyeimbang beban, dan VPN.
- Menilai persyaratan migrasi data Anda — Tentukan cara data Anda dimigrasi ke AWS, termasuk ukuran dan lokasi data Anda, waktu yang diperlukan untuk migrasi, dan biaya transfer data apa pun. Di lingkungan lokal, Anda mungkin menggunakan beragam teknologi penyimpanan seperti JBOD, NAS, dan SAN. Masing-masing dapat menyajikan data ke aplikasi Anda melalui metode akses yang berbeda, seperti SAN Fiber Channel, iSCSI, SAS, atau berbagi SMB/NFS.
- Identifikasi potensi risiko dan tantangan — Identifikasi potensi risiko atau tantangan yang dapat memengaruhi proses migrasi, seperti waktu henti, masalah kompatibilitas, atau kehilangan data.
- Perkirakan biaya — Perkirakan biaya migrasi ke AWS, termasuk biaya instans EC2, penyimpanan, transfer data, dan layanan AWS lainnya yang diperlukan.
- Buat rencana migrasi — Berdasarkan informasi yang dikumpulkan selama fase penilaian, buat rencana migrasi terperinci yang mencakup jadwal, sumber daya yang diperlukan, dan langkah-langkah yang terlibat dalam migrasi ke AWS.

## Evaluasi lingkungan Anda saat ini

Nilai lingkungan Anda saat ini, termasuk konfigurasi perangkat keras dan perangkat lunak, untuk menentukan apa yang perlu dimigrasikan ke AWS. Identifikasi dependensi antara aplikasi, server, dan database.

## Tentukan strategi migrasi Anda

Pertimbangkan opsi Anda untuk bermigrasi ke AWS, termasuk lift-and-shift pendekatan atau merancang ulang lingkungan Anda untuk memanfaatkan layanan cloud-native.

- Migrasi cluster failover tradisional — Jika Anda mengonfigurasi kluster dari awal di cloud, Anda dapat mengikuti langkah-langkah dari [Tutorial: Siapkan kluster Windows HPC di Amazon EC2 di Panduan Pengguna Amazon EC2](#) untuk Instans Windows, sambil melewati langkah-langkah khusus HPC. Atau, Anda dapat [membuat cluster grup ketersediaan SQL Server Always On](#) tanpa melalui langkah-langkah khusus SQL. Penyimpanan bersama adalah salah satu pertimbangan terpenting untuk migrasi cluster failover. Amazon EBS multi-attach tidak mendukung Reservasi Persisten SCSI-3, tetapi [Amazon FSx untuk Windows File Server dan FSx untuk NetApp ONTAP](#) keduanya berfungsi dengan baik sebagai opsi penyimpanan bersama. Salah satu kasus penggunaan yang paling umum adalah menggunakan Instans Cluster Always On Failover untuk cluster SQL Server dengan Amazon FSx for Windows File Server. Untuk informasi selengkapnya, lihat [Sederhanakan penerapan ketersediaan tinggi Microsoft SQL Server Anda menggunakan Amazon FSx for Windows File Server](#) postingan di AWS Storage Blog. Langkah selanjutnya adalah membawa node ke cloud. Hal ini dapat dicapai dengan menggunakan Layanan Migrasi Aplikasi. Untuk informasi selengkapnya, lihat [Memigrasi kluster Microsoft Windows Anda ke AWS menggunakan pos CloudEndure Migrasi](#) di Blog AWS Storage. Kemudian, Anda dapat mengonfigurasi peran yang dikelompokkan untuk aplikasi Anda untuk menyediakan ketersediaan tinggi.
- Bermigrasi tanpa downtime menggunakan stretch cluster — Stretch cluster bisa sangat cocok jika Anda memiliki aplikasi bisnis yang penting untuk bermigrasi ke cloud dan tidak mampu membayar downtime. Dengan [Microsoft stretch cluster](#), Situs A dan Situs B harus berkomunikasi satu sama lain melalui jaringan tetapi mereka dapat memiliki penyimpanan bersama masing-masing. Anda dapat menggunakan ini untuk keuntungan Anda dalam skenario migrasi. Misalnya, sumber Anda (baik di tempat maupun di cloud penyedia lain) dapat berupa Situs A, yang memiliki konektivitas jaringan dengan VPC Amazon tempat Anda menyebarkan situs B. Setelah Situs B aktif dan berjalan, Anda dapat memotong ke situs B. Mekanisme replikasi data sangat penting dalam pendekatan ini karena teknologi penyimpanan sumber Anda mungkin memiliki faktor pembatas dalam hal metode replikasi apa yang dapat bekerja.
- Memigrasi kluster failover yang diterapkan di VMware lokal ke VMware di cloud di AWS — VMware Cloud on AWS memiliki dukungan asli untuk Reservasi Persisten SCSI-3. Hal ini memungkinkan untuk meng-host cluster failover pada disk mesin virtual (VMDK) di VMware Cloud on AWS. Untuk

informasi selengkapnya, lihat [Memigrasi cluster FCI SQL Server dengan disk bersama ke VMware Cloud on AWS dalam dokumentasi VMware](#).

### Pemberitahuan

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

- Memigrasi SQL Server FCI menggunakan volume Multi-Lampirkan Amazon EBS — Anda dapat menggunakan reservasi Amazon EBS Multi-Attach dan NVMe untuk membuat SQL Server Failover Cluster Instances (FCI) dengan volume Amazon EBS sebagai penyimpanan bersama pada cluster failover Windows Server. `io2` Volume ini hanya dapat dilampirkan ke instance yang berada di Availability Zone yang sama. Menyebarkan cluster failover Windows Server dengan menggunakan `io2` volume Amazon EBS memerlukan driver Windows terbaru yang menerjemahkan perintah reservasi SCSI ke perintah reservasi NVMe. Untuk informasi selengkapnya tentang memigrasikan FCI SQL Server lokal Anda ke AWS dalam satu Availability Zone dengan menggunakan pendekatan ini, lihat [postingan blog AWS Cara menerapkan cluster failover SQL Server dengan Amazon EBS Multi-Attach di Windows Server](#).

Fase penilaian sangat penting untuk memastikan keberhasilan migrasi cluster failover Anda ke AWS. Jika Anda meluangkan waktu untuk mengumpulkan informasi dan mengidentifikasi potensi tantangan, Anda dapat mengembangkan rencana migrasi komprehensif yang meminimalkan waktu henti, mengurangi risiko, dan memastikan transisi yang mulus ke AWS.

## Memobilisasi

Selama migrasi cluster failover ke AWS, fase mobilisasi melibatkan persiapan cluster untuk migrasi ke AWS dan mengujinya untuk memastikan fungsinya dengan baik. Fase mobilisasi meliputi langkah-langkah berikut:

1. Siapkan lingkungan target — Pada langkah ini, Anda membuat sumber daya AWS yang diperlukan untuk meng-host cluster failover. Ini melibatkan pengaturan VPC Amazon, subnet, grup keamanan, dan sumber daya lain yang diperlukan.
2. Siapkan lingkungan sumber — Pada langkah ini, Anda menyiapkan kluster failover yang ada untuk migrasi. Ini dapat melibatkan membuat perubahan pada konfigurasi jaringan, mengkonfigurasi replikasi, atau menginstal perangkat lunak yang diperlukan.



3. Validasi cluster — Setelah lingkungan sumber dan target disiapkan, Anda dapat melakukan uji validasi untuk memastikan bahwa cluster berfungsi dengan baik. Ini melibatkan menjalankan serangkaian tes untuk memastikan bahwa cluster dapat gagal ke lingkungan target dengan sukses.
4. Buat tautan replikasi — Setelah uji validasi, Anda dapat membuat tautan replikasi antara sumber dan lingkungan target. Ini memastikan bahwa setiap perubahan yang dibuat pada lingkungan sumber direplikasi ke lingkungan target.
5. Pantau replikasi — Setelah tautan replikasi dibuat, pantau proses replikasi untuk memastikan bahwa semua perubahan direplikasi dengan benar.
6. Gagal di atas cluster — Setelah memverifikasi bahwa replikasi berfungsi dengan benar, lakukan failover akhir ke lingkungan target. Ini melibatkan menghentikan layanan cluster di lingkungan sumber dan memulainya di lingkungan target.
7. Uji failover — Setelah failover selesai, lakukan pengujian untuk memastikan bahwa aplikasi dan layanan yang berjalan di cluster berfungsi dengan baik di lingkungan baru

## Migrasi

Migrasi cluster failover Microsoft dapat menjadi proses kompleks yang memerlukan perencanaan dan implementasi yang cermat untuk memastikan hasil yang sukses. Sangat penting untuk menilai secara menyeluruh lingkungan yang ada, mengidentifikasi potensi masalah, dan mengembangkan rencana migrasi komprehensif yang mencakup pengujian dan validasi sebelum membuat perubahan apa pun pada lingkungan produksi. Selama fase migrasi, penting untuk memantau proses dengan cermat dan mengatasi masalah atau perilaku tak terduga dengan segera. Komunikasi dan kolaborasi antara semua pemangku kepentingan — termasuk tim TI, pengguna bisnis, dan vendor — sangat penting untuk proses migrasi yang lancar.

Selain itu, penting untuk mempertimbangkan dampak migrasi pada aplikasi atau layanan pihak ketiga yang berjalan di kluster failover. Identifikasi dependensi apa pun dan uji aplikasi tersebut secara menyeluruh untuk memastikan bahwa dependensi tersebut terus berfungsi seperti yang diharapkan setelah migrasi. Aspek kunci lain dari fase migrasi adalah membuat rencana rollback jika terjadi masalah atau kegagalan yang tidak terduga selama proses migrasi. Rencana ini idealnya mencakup langkah-langkah untuk mengembalikan migrasi dan memulihkan lingkungan semula, sambil meminimalkan dampak apa pun pada lingkungan produksi.

Akhirnya, setelah migrasi selesai dan cluster failover berhasil berjalan di lingkungan baru, penting untuk melakukan validasi dan pengujian pasca-migrasi untuk mengonfirmasi bahwa semuanya



berfungsi sebagaimana dimaksud. Ini termasuk memantau kinerja, memvalidasi kemampuan failover, dan memastikan bahwa semua aplikasi dan layanan berfungsi dengan baik.

## Memantau beban kerja Microsoft

Beban kerja Microsoft biasanya menggunakan SQL Server di backend untuk mengambil dan mempertahankan data. Seringkali dalam perjalanan ke cloud, keputusan rehost dibuat untuk solusi semacam itu dengan menggunakan lift-and-shift pendekatan sederhana. Ketika aplikasi tersebut di-host pada platform Windows di Amazon EC2, Anda dapat menggunakan alat berbasis Windows asli untuk memantau kesehatan aplikasi ini di tingkat server. Namun, mendapatkan pandangan holistik di berbagai komponen dan server yang digunakan sebagai bagian dari solusi merupakan tantangan, tetapi masalah ini dapat diatasi oleh [Amazon CloudWatch Application Insights](#).

CloudWatch Application Insights adalah layanan pemantauan cloud-native yang dapat membantu Anda mengatur dan memantau sumber daya aplikasi untuk beban kerja AWS Anda. Pelanggan perusahaan menangani berbagai beban kerja dan memerlukan layanan pemantauan yang dapat menghubungkan data telemetri dari sumber yang berbeda. Jika Anda adalah pelanggan perusahaan, CloudWatch Application Insights dapat membantu Anda menghindari kerumitan dalam menyiapkan pemantauan dengan mengotomatiskan penemuan sumber daya dan membantu membuat aplikasi dari berbagai sumber daya.

## Menilai

Melacak kinerja aplikasi dan kesehatan backend sangat penting bagi sebagian besar organisasi. Anda perlu tahu kapan dan di mana sepanjang perjalanan ditemukan kelainan dan mengapa itu terjadi. Anda juga perlu memantau sistem Anda dan mengurangi biaya perawatan.

CloudWatch dapat membantu Anda dengan kebutuhan pemantauan Anda, dan Wawasan CloudWatch Aplikasi menggunakan CloudWatch metrik, alarm, dan peristiwa. Anda dapat menggunakannya CloudWatch untuk mengatur pemantauan dan pengelolaan metrik, telemetri, dan log untuk banyak sumber daya AWS. [Amazon CloudWatch ServiceLens](#) menyediakan kombinasi layanan untuk memberi Anda semua yang Anda butuhkan untuk memantau kesehatan aplikasi Anda.

## Memobilisasi

CloudWatch Application Insights menyediakan antarmuka pengguna klik rendah yang dapat Anda gunakan untuk mengatur metrik dan log telemetri optimal untuk aplikasi Anda dengan cepat dan

mudah. CloudWatch Application Insights menyesuaikan monitornya dengan beban kerja spesifik Anda sehingga Anda dapat terus menganalisis tanda-tanda masalah untuk aplikasi spesifik Anda. Ini juga memberikan konfigurasi otomatis dan analisis telemetri beban kerja yang direkomendasikan. Beberapa contoh termasuk .NET CLR, permintaan per detik untuk teknologi aplikasi/server web, mengidentifikasi masalah umum yang terkait dengan pengumpulan sampah.NET, dan pencadangan gagal SQL Server.

Saat Anda mencari solusi pemantauan, Anda biasanya harus memahami dan mengonfigurasi CPU, memori, dan persyaratan ambang batas lainnya. Namun, CloudWatch Application Insights secara otomatis mendeteksi sumber daya ini dan metrik yang relevan. Saat Anda menambahkan aplikasi ke CloudWatch Application Insights, aplikasi akan memindai sumber daya, dan merekomendasikan serta mengonfigurasi metrik dan log on CloudWatch untuk komponen aplikasi. Contoh komponen aplikasi termasuk database backend SQL Server dan Microsoft IIS/Web tier.

Berdasarkan grup sumber daya yang dipilih, CloudWatch Application Insights secara otomatis menyiapkan pemantauan untuk setiap komponen. Dalam kasus pemantauan aplikasi berbasis akun, semua sumber daya yang ditemukan di akun Anda ditambahkan secara otomatis. Anda juga bisa mendapatkan keuntungan dari kemampuan deteksi sumber daya dari CloudWatch Application Insights.

CloudWatch Application Insights menganalisis pola metrik menggunakan data historis untuk mendeteksi anomali, dan terus mendeteksi kesalahan dan pengecualian dari aplikasi, sistem operasi, dan log infrastruktur. Wawasan Aplikasi ini mengorelasikan observasi ini menggunakan sebuah kombinasi antara algoritma klasifikasi dan aturan bawaan. Kemudian, Wawasan Aplikasi akan secara otomatis membuat dashboard yang menampilkan observasi yang relevan dan informasi kepelikan masalah untuk membantu Anda menentukan prioritas tindakan. Untuk masalah umum di tumpukan aplikasi.NET dan SQL, seperti latensi aplikasi, pencadangan gagal SQL Server, kebocoran memori, permintaan HTTP yang besar dan tidak valid, dan operasi I/O yang dibatalkan, CloudWatch Application Insights memberikan wawasan tambahan yang mengarah ke kemungkinan akar penyebab dan langkah-langkah untuk resolusi.

Integrasi bawaan dengan [AWS Systems Manager OpsCenter](#) memungkinkan Anda menyelesaikan masalah dengan menjalankan dokumen AWS Systems Manager Automation yang relevan. CloudWatch Application Insights meneruskan tingkat keparahan untuk setiap masalah ke AWS Systems Manager OpsCenter, yang selanjutnya membantu Anda memprioritaskan dan menetapkan tugas dalam tim dukungan Anda.

## Migrasi

CloudWatch Application Insights adalah bagian dari ekosistem Windows di Amazon EC2. Menggunakan Wawasan CloudWatch Aplikasi untuk pemantauan adalah bagian penting dari penawaran ini. Setelah memulai migrasi beban kerja ke AWS, Anda dapat mengandalkan CloudWatch Application Insights untuk memantau beban kerja Microsoft Anda. Selain itu, CloudWatch Application Insights menyediakan dukungan di luar beban kerja Microsoft, termasuk dukungan untuk SAP, Java, Oracle, MySQL, PostgreSQL, dan sumber daya AWS lainnya (termasuk dukungan untuk aplikasi tanpa server). Untuk memulai dengan CloudWatch Application Insights, lihat [Menyiapkan](#) di Panduan CloudWatch Pengguna Amazon.

# Alat migrasi, program, dan pelatihan

Bagian ini menguraikan AWS dan alat mitra yang tersedia untuk membantu migrasi cloud Anda, peluang pelatihan yang tersedia untuk memberi tim Anda keterampilan yang mereka butuhkan untuk bermigrasi dan beroperasi di cloud, dan program migrasi utama yang tersedia untuk mempercepat perjalanan migrasi Anda dan mengurangi biaya migrasi.

## Alat

### Alat penilaian

#### Penilaian Pengoptimalan dan Lisensi AWS

Sebaiknya gunakan [AWS Optimization and Licensing Assessment \(AWS OLA\)](#) untuk membangun strategi migrasi dan lisensi Anda di AWS. Anda dapat menggunakan AWS OLA untuk mengevaluasi lingkungan Windows Anda. Evaluasi ini membantu Anda mengidentifikasi potensi penghematan pada biaya lisensi Anda dan menemukan cara untuk menjalankan sumber daya Anda secara lebih efisien.

AWS OLA adalah program bebas kewajiban untuk pelanggan baru dan yang sudah ada. Anda dapat menggunakan AWS OLA untuk menilai dan mengoptimalkan lingkungan lokal dan cloud Anda saat ini, berdasarkan pemanfaatan sumber daya aktual, lisensi pihak ketiga, dan dependensi aplikasi. Sebuah studi pihak ketiga pada tahun 2022 oleh [Enterprise Strategy Group dan Evolve Cloud Services](#) menghitung bahwa AWS OLA menghemat rata-rata 45 persen pada biaya lisensi Microsoft SQL Server dan 77 persen di Windows Server. Biaya lisensi sama dengan tiga kali biaya untuk benar-benar menjalankan beban kerja ini di AWS Cloud sehingga potensi penghematan dapat berdampak signifikan pada TCO Anda.

AWS OLA memberi Anda laporan yang memodelkan opsi penerapan Anda. Hasil ini dapat membantu Anda menjelajahi penghematan biaya yang tersedia di seluruh opsi lisensi fleksibel yang ditawarkan oleh AWS. Anda juga dapat menggunakan AWS OLA dalam kombinasi dengan [AWS Migration Acceleration Program for Windows](#) untuk mendapatkan dukungan dan sumber daya selama migrasi cloud Anda.

Anda dapat menggunakan AWS OLA sebelum, selama, atau bahkan setelah migrasi Anda. Pendekatan berbasis alat ini dapat membantu Anda menentukan persyaratan pemanfaatan aktual Anda. AWS OLA membuat rekomendasi untuk ukuran dan jenis instans EC2 dengan biaya terendah

untuk setiap beban kerja. Ini juga dapat membantu Anda menemukan perpaduan yang tepat antara Instans Sesuai Permintaan, Instans Spot, Host Khusus Amazon EC2, paket tabungan, dan opsi lain yang spesifik untuk lingkungan Anda. Selain itu, AWS OLA memberi Anda rencana migrasi, kasus bisnis terarah, dan peta jalan.

Penghematan lisensi adalah bagian penting dari TCO Anda, dan AWS OLA dapat membantu Anda mengurangi biaya lisensi dengan memberikan Bring Your Own License (BYOL) atau rekomendasi yang disertakan lisensi berdasarkan hak lisensi dan beban kerja Anda yang ada. AWS OLA mengoptimalkan lisensi Anda dengan mengonfigurasi instans agar memerlukan lebih sedikit lisensi sekaligus mempertahankan kinerja tinggi untuk aplikasi Anda. AWS OLA juga membantu Anda memahami perbedaan antara lisensi lokal dibandingkan dengan lisensi di cloud. Anda dapat menggunakan pengetahuan ini untuk menyesuaikan strategi lisensi Anda untuk lebih mengurangi biaya di masa depan.

Ruang lingkup AWS OLA mencakup kasus penggunaan berikut:

- Kasus bisnis terarah, rekomendasi yang menguraikan biaya instans EC2, dan konfigurasi berdasarkan pemanfaatan dan data lokal yang sebenarnya
- Pemodelan Host Khusus untuk lisensi tingkat Host
- Pengurangan CPU virtual (vCPU) untuk pengoptimalan dan konsolidasi instans SQL
- Estimasi TCO lokal berdasarkan rata-rata industri
- Pemodelan VMware Cloud on AWS

#### Pemberitahuan

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

- Rekomendasi berdasarkan posisi lisensi Microsoft Anda (mengenai mobilitas lisensi dan potensi pengurangan)
- Pemodelan dampak lisensi untuk Host Khusus T3
- Pemodelan SQL dan Oracle di Amazon RDS, optimasi edisi, dan analisis Oracle Real Application Clusters (RAC) dan Oracle Exadata
- Pemodelan aktif dan pasif untuk dampak lisensi ketersediaan tinggi SQL
- Penilaian modernisasi

AWS menggunakan [Penilai Migrasi](#) internal atau alat tepercaya dari vendor pihak ketiga (atau mitra migrasi AWS OLA yang memenuhi syarat) untuk melakukan penemuan berbasis luas atau mengunggah ekspor dengan aman jika Anda memiliki inventaris yang sudah ada. Alat yang digunakan tergantung pada kebutuhan dan persyaratan spesifik Anda. AWS menggunakan output alat penemuan dan menggabungkannya dengan rekomendasi ahli dari konsultan lisensi pihak ketiga untuk memberi Anda TCO yang dioptimalkan yang dapat Anda percayai.

Untuk informasi selengkapnya, lihat sumber daya berikut:

- [Penilaian Pengoptimalan dan Lisensi](#) AWS (dokumentasi AWS)
- [Optimalkan Beban Kerja Windows Anda untuk AWS - AWS Online Tech Talks](#) () YouTube
- [Jalankan Penilaian Pengoptimalan dan Lisensi](#) (dokumentasi AWS)

## Rekomendasi Strategi Migrasi Hub

[Rekomendasi Strategi Hub Migrasi](#) membantu Anda merencanakan inisiatif migrasi dan modernisasi dengan menawarkan rekomendasi strategi migrasi dan modernisasi untuk jalur transformasi yang layak untuk aplikasi Anda. Rekomendasi Strategi melakukan analisis inventaris server dan lingkungan runtime Anda. Itu juga dapat melakukan kode sumber dan analisis basis data. Rekomendasi Strategi menggabungkan analisis ini dengan tujuan bisnis Anda, dan preferensi transformasi aplikasi dan database yang disediakan untuk merekomendasikan hal-hal berikut:

- Strategi migrasi paling efektif untuk setiap aplikasi Anda
- Alat migrasi dan modernisasi atau program yang dapat Anda gunakan
- Ketidakcocokan aplikasi dan anti-pola untuk menyelesaikan opsi tertentu

Rekomendasi Strategi merekomendasikan strategi migrasi dan modernisasi untuk rehosting, replatforming, dan refactoring dengan tujuan, alat, dan program penyebaran terkait. Misalnya, Rekomendasi Strategi mungkin merekomendasikan opsi langsung, seperti rehosting di Amazon EC2 dengan menggunakan Layanan Migrasi Aplikasi. Rekomendasi yang lebih dioptimalkan mungkin termasuk replatforming ke container dengan menggunakan AWS App2Container atau refactoring ke teknologi open-source seperti .NET Core dan PostgreSQL.

Untuk menggunakan Rekomendasi Strategi, ikuti petunjuk dari [Memulai Rekomendasi Strategi](#) di Panduan Pengguna Rekomendasi Strategi Hub Migrasi.

## Modul Alat Validator Migrasi PowerShell

Kami menyarankan Anda menggunakan [PowerShell modul Migration Validator Toolkit](#) untuk menemukan beban kerja Microsoft Anda dan memigrasikannya ke AWS. Modul ini bekerja dengan melakukan beberapa pemeriksaan dan validasi untuk tugas umum yang terkait dengan beban kerja Microsoft apa pun. PowerShell Modul Migration Validator Toolkit dapat membantu organisasi Anda mengurangi waktu dan upaya yang terlibat dalam menemukan aplikasi dan layanan apa yang berjalan pada beban kerja Microsoft Anda. Modul ini juga dapat membantu Anda mengidentifikasi konfigurasi beban kerja Anda sehingga Anda dapat mengetahui apakah konfigurasi Anda didukung di AWS. Modul ini juga memberikan rekomendasi untuk langkah selanjutnya dan tindakan mitigasi, sehingga Anda dapat menghindari kesalahan konfigurasi sebelum, selama, atau setelah migrasi.

## Penilaian Kesiapan AWS Cloud

Kami menyarankan Anda menggunakan [AWS Cloud Readiness Assessment](#) untuk mengubah ide Anda untuk pindah ke cloud menjadi paket terperinci yang mengikuti praktik terbaik AWS Professional Services. Anda dapat menggunakan AWS Cloud Readiness Assessment untuk mengembangkan rencana yang efisien dan efektif untuk adopsi cloud dan migrasi cloud perusahaan, terlepas dari ukuran organisasi Anda. Laporan survei dan penilaian online 16 pertanyaan ini merinci kesiapan migrasi cloud Anda di enam perspektif, termasuk bisnis, orang, proses, platform, operasi, dan keamanan.

Setelah menyelesaikan penilaian, Anda dapat memberikan detail kontak untuk mengunduh penilaian migrasi cloud khusus yang memetakan kesiapan Anda dan apa yang dapat Anda lakukan untuk memperbaikinya. Laporan ringkasan Anda mencakup peta panas dan bagan radar dengan informasi penilaian terperinci dan sumber daya untuk membantu Anda meningkatkan skor kesiapan Anda. Kabin laporan take-away ini membantu Anda merencanakan dan berkomunikasi dengan pemangku kepentingan Anda. Untuk contoh laporan penilaian, lihat [Laporan Penilaian Kesiapan Adopsi AWS Cloud](#). Untuk mengikuti penilaian, buka [Penilaian Kesiapan Adopsi AWS Cloud](#).

## Alat Migrasi

### AWS Migration Hub

[AWS Migration Hub](#) menyediakan lokasi pusat untuk mengumpulkan data inventaris server dan aplikasi untuk penilaian, perencanaan, dan pelacakan migrasi ke AWS. Migration Hub juga dapat membantu Anda mempercepat modernisasi aplikasi setelah migrasi. Visualisasi jaringan Migration Hub memungkinkan Anda mempercepat perencanaan migrasi dengan mengidentifikasi server dan dependensinya dengan cepat, mengidentifikasi peran server, dan mengelompokkan server ke dalam aplikasi. Untuk menggunakan visualisasi jaringan, instal [AWS Application Discovery Agent \(Discovery Agent\)](#), lalu mulai pengumpulan data.

## Orkestrator Hub Migrasi AWS

[AWS Migration Hub Orchestrator](#) membantu mempercepat migrasi aplikasi Anda untuk mengurangi waktu dan upaya migrasi. Anda dapat menggunakan templat alur kerja yang telah ditentukan untuk membuat alur kerja migrasi dengan mudah, menyesuaikan alur kerja sesuai kebutuhan spesifik Anda, mengotomatiskan langkah migrasi, dan melacak kemajuan migrasi dari awal hingga akhir di satu tempat. Orchestrator mendukung hal-hal berikut:

- Migrasi aplikasi berbasis SAP NetWeaver dengan database SAP HANA
- Rehosting aplikasi apa pun ke Amazon EC2
- Rehosting database SQL Server ke Amazon EC2
- Pembuatan ulang database SQL Server ke Amazon RDS
- Mengimpor gambar VM dari Open Virtual Appliance (OVA) atau VMware Virtual Machine Disk (VMDK) ke AMI untuk Amazon EC2

## Dasbor AWS Migration Hub

[Dasbor Migration Hub](#) menampilkan status dan metrik terbaru untuk migrasi rehost dan replatform Anda. Anda dapat menggunakan dasbor untuk memahami kemajuan migrasi Anda dengan cepat dan mengidentifikasi serta memecahkan masalah apa pun. Migration Hub memungkinkan Anda melacak status migrasi ke Wilayah AWS mana pun yang didukung oleh alat migrasi Anda. Terlepas dari Wilayah mana Anda bermigrasi, status migrasi akan muncul di Hub Migrasi saat menggunakan alat terintegrasi.

## Layanan Migrasi Aplikasi AWS

[AWS Application Migration Service](#) meminimalkan proses manual yang intensif waktu dan rawan kesalahan dengan mengotomatiskan konversi server sumber Anda agar berjalan secara native di AWS. Ini juga menyederhanakan modernisasi aplikasi dengan opsi pengoptimalan bawaan dan kustom. Kasus penggunaan untuk Layanan Migrasi Aplikasi meliputi:

- Beban kerja lokal seperti SAP, Oracle, dan SQL Server yang berjalan di server fisik atau di VMware vSphere, Microsoft Hyper-V, dan infrastruktur lokal lainnya
- Beban kerja berbasis cloud berjalan dari cloud publik lainnya ke AWS

Anda dapat menggunakan Layanan Migrasi Aplikasi untuk mengakses lebih dari 200 layanan yang mengurangi biaya, meningkatkan ketersediaan, dan memfasilitasi inovasi. Selain itu, Anda dapat



menggunakannya untuk memindahkan beban kerja EC2 antara Wilayah AWS, Availability Zone, atau akun dengan lebih mudah untuk memenuhi kebutuhan bisnis, ketahanan, dan kepatuhan Anda.

Atau, sebagai strategi modernisasi, Anda dapat mengoptimalkan aplikasi Anda dengan menerapkan tindakan modernisasi khusus atau memilih tindakan bawaan seperti pemulihan bencana lintas wilayah, konversi CentOS, dan konversi langganan SUSE Linux.

## AWS Database Migration Service

[AWS Database Migration Service \(AWS DMS\)](#) adalah layanan migrasi dan replikasi terkelola yang membantu memindahkan beban kerja database dan analitik Anda ke AWS dengan cepat, aman, dan dengan waktu henti minimal dan nol kehilangan data. AWS DMS mendukung migrasi antara 20-plus database dan mesin analitik, termasuk SQL Server.

AWS DMS memungkinkan Anda menggunakan model database terkelola untuk bermigrasi dari database lama atau lokal ke layanan cloud terkelola melalui proses migrasi yang disederhanakan, yang memberi waktu kepada pengembang untuk berinovasi. Anda juga dapat menggunakan AWS DMS untuk membebaskan diri dari biaya lisensi, mempercepat pertumbuhan bisnis, dan menggunakan database yang dibuat khusus untuk berinovasi dan membangun lebih cepat untuk kasus penggunaan apa pun dalam skala sepersepuluh biaya.

Anda juga dapat menggunakan AWS DMS untuk melakukan hal berikut:

- Replikasi file cadangan
- Membuat redundansi database bisnis penting dan penyimpanan data untuk meminimalkan downtime dan kehilangan data
- Membangun data lake untuk melakukan pemrosesan real-time pada data perubahan dari penyimpanan data Anda
- Integrasikan data mart dengan membangun data lake
- Lakukan pemrosesan real-time pada data perubahan dari penyimpanan data Anda

## Alat Mitra Migrasi

### CloudBasix

[CloudBasix](#) membuat produk pengoptimalan beban kerja cloud-native dan integrasi data. Anda dapat menggunakan produk andalannya, [CLOUDBASIX untuk RDS SQL Server Read Replicas and Disaster Recovery \(DR\)](#), untuk mengaktifkan hal berikut:

- Replika baca In-Region
- Lintas Wilayah DR
- Pemulihan bencana antar-cloud Azure ke AWS
- Danau data dan rumah data berbasis AI
- Integrasi untuk Amazon Redshift dan Snowflake

## Alat manajemen

### Wawasan CloudWatch Aplikasi Amazon

[Amazon CloudWatch Application Insights](#) memfasilitasi pengamatan untuk aplikasi Anda dan sumber daya AWS yang mendasarinya. Ini membantu Anda mengatur monitor terbaik untuk sumber daya aplikasi Anda untuk terus menganalisis data untuk tanda-tanda masalah dengan aplikasi Anda. CloudWatch Application Insights, yang didukung oleh Amazon SageMaker dan teknologi AWS lainnya, menyediakan dasbor otomatis yang menunjukkan potensi masalah dengan aplikasi yang dipantau. Ini dapat membantu Anda dengan cepat mengisolasi masalah yang sedang berlangsung dengan aplikasi dan infrastruktur Anda.

Saat Anda menambahkan aplikasi ke CloudWatch Application Insights, aplikasi akan memindai sumber daya dalam aplikasi dan merekomendasikan serta mengonfigurasi metrik dan log on CloudWatch untuk komponen aplikasi. Contoh komponen aplikasi termasuk database backend SQL Server dan Microsoft IIS atau tingkatan web. CloudWatch Application Insights menganalisis pola metrik menggunakan data historis untuk mendeteksi anomali dan terus mendeteksi kesalahan dan pengecualian dari log aplikasi, sistem operasi, dan infrastruktur Anda. Wawasan Aplikasi ini mengorelasikan observasi ini menggunakan sebuah kombinasi antara algoritma klasifikasi dan aturan bawaan. Kemudian, CloudWatch Application Insights secara otomatis membuat dasbor yang menunjukkan pengamatan yang relevan dan informasi tingkat keparahan masalah untuk membantu Anda memprioritaskan tindakan Anda. Untuk masalah umum di tumpukan aplikasi.NET dan SQL — seperti latensi aplikasi, pencadangan gagal SQL Server, kebocoran memori, permintaan HTTP yang besar, dan operasi I/O yang dibatalkan — ini memberikan wawasan tambahan yang mengarah ke kemungkinan akar penyebab dan langkah-langkah untuk resolusi. Integrasi bawaan dengan [AWS Systems Manager OpsCenter](#) memungkinkan Anda menyelesaikan masalah dengan menjalankan dokumen Otomasi Systems Manager yang relevan.

### AWS License Manager

[AWS License Manager](#) memudahkan Anda mengelola lisensi perangkat lunak dari vendor, seperti Microsoft, SAP, Oracle, dan IBM, di AWS dan lingkungan lokal Anda. Anda dapat menggunakan License Manager untuk merampingkan manajemen lisensi dengan beralih di antara jenis lisensi dan mengotomatiskan penemuan, pelacakan, dan pelaporan lisensi yang ada. Anda juga dapat menyederhanakan pengalaman windows BYOL melalui pengelolaan kumpulan Host Khusus sebagai entitas tunggal dengan alokasi, rilis, dan pemulihan otomatis. Selain itu, Anda dapat menangani lisensi marketplace di seluruh akun dengan mengotomatiskan distribusi dan aktivasi hak perangkat lunak dan beban kerja di seluruh akun AWS untuk pengguna akhir.

## AWS Backup

[AWS Backup](#) adalah layanan berbasis kebijakan yang hemat biaya, dikelola sepenuhnya, yang menyederhanakan perlindungan data dalam skala besar. Anda dapat menggunakan AWS Backup untuk membuat backup cloud-native untuk penyimpanan data utama, seperti bucket, volume, database, dan sistem file di seluruh layanan AWS. AWS Backup memusatkan perlindungan data Anda dengan menyediakan manajemen perlindungan data untuk aplikasi Anda yang berjalan di lingkungan hybrid, seperti beban kerja VMware dan volume AWS Storage Gateway. Anda juga dapat mengelola kebijakan secara terpusat untuk mengonfigurasi, mengelola, dan mengatur aktivitas pencadangan di seluruh akun AWS, sumber daya, dan Wilayah organisasi Anda.

## Manajer Armada AWS Systems Manager

[Fleet Manager](#), kemampuan AWS Systems Manager, adalah pengalaman antarmuka pengguna (UI) terpadu yang membantu Anda mengelola node yang berjalan di AWS atau di lokasi dari jarak jauh. Dengan Fleet Manager, Anda dapat melihat status kesehatan dan kinerja seluruh armada server Anda dari satu konsol. Anda juga dapat mengumpulkan data dari masing-masing node untuk melakukan pemecahan masalah umum dan tugas manajemen dari konsol. Ini termasuk menghubungkan ke instance Windows dengan menggunakan Remote Desktop Protocol (RDP), melihat folder dan konten file, manajemen registri Windows, manajemen pengguna sistem operasi, dan banyak lagi. Anda dapat menggunakan Fleet Manager jika ingin memusatkan pengelolaan armada node atau cluster Amazon ECS Anda.

# Program

## AWS Migration Acceleration Program

[AWS Migration Acceleration Program \(MAP\)](#) adalah program migrasi cloud yang komprehensif dan terbukti berdasarkan pengalaman AWS yang memigrasikan ribuan pelanggan perusahaan

ke cloud. Migrasi perusahaan bisa rumit dan memakan waktu, tetapi MAP dapat membantu Anda mempercepat migrasi cloud dan perjalanan modernisasi dengan metodologi berbasis hasil.

MAP menyediakan alat yang mengurangi biaya dan mengotomatiskan serta mempercepat implementasi, pendekatan dan konten pelatihan yang disesuaikan, keahlian dari Mitra di AWS Partner Network, komunitas mitra global, dan investasi AWS. MAP juga menggunakan kerangka kerja tiga fase yang telah terbukti untuk membantu Anda mencapai tujuan migrasi. Melalui MAP, Anda dapat membangun fondasi AWS cloud yang kuat sekaligus mengurangi risiko, meningkatkan produktivitas, meningkatkan ketahanan operasional, dan mengimbangi biaya awal migrasi. Anda juga dapat memanfaatkan kinerja, keamanan, dan keandalan cloud.

### Akselerator Migrasi AWS Windows

[AWS Windows Migration Accelerator](#) membantu mengurangi biaya migrasi Anda dengan menggunakan AWS Promotional Credit saat Anda mempercepat migrasi server Windows menggunakan [Layanan Migrasi Aplikasi](#). Insentif AWS Windows Migration Accelerator dapat diterapkan di atas insentif penjualan dan program promosi lainnya yang disepakati. Jika Anda menggunakan Layanan Migrasi Aplikasi untuk memigrasikan setidaknya 40 server ke AWS dalam satu bulan, termasuk minimal 15 server Windows, Anda mungkin memenuhi syarat untuk menerima Kredit Promosi AWS senilai \$200 per server Windows, hingga 31 Desember 2023. Jika Anda memigrasikan lebih dari 80 server, termasuk setidaknya 25 server Windows, dalam satu bulan kalender, diskon akan meningkat menjadi \$250 AWS Kredit Promosi untuk setiap server Windows yang Anda migrasi ke AWS menggunakan Layanan Migrasi Aplikasi. Server yang dimigrasi harus dimigrasikan dari lokasi di luar AWS dan terus berjalan di AWS setidaknya selama empat minggu setelah migrasi.

### AWS Migration Acceleration Program untuk Windows

[AWS Migration Acceleration Program \(MAP\) untuk Windows](#), perpanjangan dari program AWS MAP yang ada, dirancang untuk membantu organisasi mencapai tujuan migrasi mereka lebih cepat dengan layanan AWS, praktik terbaik, alat, dan insentif. AWS menggunakan pendekatan tiga langkah untuk membantu Anda mengurangi ketidakpastian, kompleksitas, dan biaya migrasi ke cloud. Selain itu, MAP dapat membantu Anda memodernisasi versi Windows Server dan SQL Server saat ini dan lama untuk mengurangi biaya dengan menggunakan solusi cloud seperti SQL Server yang berjalan di Linux, Aurora, layanan berbasis kontainer, dan Lambda. Solusi cloud-native atau open-source dapat membantu Anda membebaskan diri dari tingginya biaya lisensi komersial.

### AWS Infrastructure Event Management

[AWS Infrastructure Event Management \(IEM\)](#) menawarkan panduan arsitektur dan penskalaan serta dukungan operasional selama persiapan dan implementasi acara yang direncanakan, seperti liburan belanja, peluncuran produk, dan migrasi. Untuk acara ini, IEM membantu Anda menilai kesiapan operasional, mengidentifikasi dan mengurangi risiko, dan mengimplementasikan acara Anda dengan percaya diri dengan pakar AWS di sisi Anda. Program ini termasuk dalam paket Enterprise Support dan tersedia untuk pelanggan Business Support dengan biaya tambahan.

Pakar AWS memimpin keterlibatan yang sangat terfokus untuk memberi Anda panduan arsitektur dan operasional untuk acara yang Anda rencanakan menggunakan pendekatan preskriptif dan bertahap yang membantu Anda melakukan hal berikut:

- Memahami kriteria keberhasilan Anda dan hasil bisnis yang diinginkan
- Menilai kesiapan lingkungan AWS Anda, membantu mengidentifikasi dan mengurangi risiko, dan mendokumentasikan rencana Anda
- Selenggarakan acara Anda dengan percaya diri dengan pakar AWS di sisi Anda
- Analisis hasil pasca acara dan skala layanan ke tingkat operasi normal, sehingga Anda dapat fokus pada perencanaan acara berikutnya

## Pelatihan

Pelatihan mandiri, interaktif, dan kelas

AWS menawarkan pelatihan digital dan kelas untuk mendukung Anda dalam perjalanan migrasi Anda. Anda dapat mulai belajar dengan ratusan kursus pelatihan digital mandiri yang dibangun oleh para ahli di AWS. Kemudian, Anda dapat memperoleh keterampilan langsung dengan menyelesaikan pelatihan interaktif dengan [AWS Skill Builder](#). Dengan pelatihan kelas Anda dapat mengajukan pertanyaan, mengerjakan solusi secara langsung, dan mendapatkan umpan balik dari instruktur terakreditasi AWS dengan pengetahuan teknis yang mendalam. Untuk informasi selengkapnya, jelajahi penawaran [AWS Training and Certification](#).

Pelatihan AWS Partner

AWS Partners juga menawarkan pelatihan digital sebagai kursus mandiri yang mencakup berbagai topik mulai dari fundamental AWS Cloud hingga pembelajaran mesin di platform pembelajaran online teratas seperti edX dan Coursera. Untuk informasi selengkapnya, jelajahi penawaran [Pelatihan dan Sertifikasi AWS Partner](#). Anda dapat disertifikasi oleh peran dan solusi. Misalnya, peran termasuk

---

Praktisi Cloud, Arsitek Solusi, Pengembang, dan SysOps Administrator. Solusi termasuk Jaringan Lanjutan, Analisis Data, Database, Machine Learning, Keamanan, Penyimpanan, dan banyak lagi.

# Lisensi Microsoft di AWS

Bagian ini menjelaskan cara kerja lisensi Microsoft di AWS, menyediakan praktik dan strategi terbaik lisensi untuk penerapan beban kerja Microsoft di AWS, dan membantu Anda tetap mematuhi persyaratan lisensi Microsoft sambil mengoptimalkan biaya. Karena dampak lisensi pada biaya migrasi, opsi lisensi Microsoft dan Bring Your Own License (BYOL) sering memengaruhi opsi penerapan yang tersedia untuk pelanggan AWS. Itulah mengapa penting untuk memahami cara kerja perizinan sebelum Anda memulai proses migrasi.

## Menilai

Saat menilai beban kerja Microsoft Anda untuk migrasi ke AWS, penting untuk mempertimbangkan persyaratan lisensi. Untuk beban kerja Microsoft, kami menyarankan Anda memanfaatkan [AWS Optimization and Licensing Assessment \(AWS OLA\)](#) untuk menilai beban kerja lokal atau cloud serta membuat peta jalan berukuran tepat dan dioptimalkan untuk menjalankan beban kerja di AWS. AWS OLA tidak hanya akan membuat saran yang dioptimalkan untuk instans EC2 yang tepat untuk beban kerja Anda, tetapi juga akan melihat posisi lisensi Microsoft Anda. Hasilnya akan menjadi rekomendasi untuk jalur terbaik ke depan untuk menghemat biaya komputasi dan lisensi. AWS OLA tersedia untuk pelanggan baru dan yang sudah ada, dan sepenuhnya didanai dan bebas kewajiban. Untuk informasi selengkapnya, hubungi [tim AWS OLA](#).

Jika AWS OLA bukan pilihan untuk Anda saat ini, masih penting untuk memahami cara kerja lisensi Microsoft di AWS. Jika Anda mencari BYOL, sebaiknya Anda meminta salinan Pernyataan Lisensi Microsoft (MLS) terbaru dari kontak pembelian lisensi Microsoft Anda. Gunakan ini untuk meninjau lisensi apa yang Anda miliki dan tanggal pembelian dan jumlah SA jika berlaku. Untuk bantuan dengan MLS Anda, hubungi perwakilan AWS Anda. Perwakilan Anda dapat menghubungkan Anda dengan spesialis Microsoft.

Produk Microsoft yang berbeda memiliki persyaratan lisensi yang berbeda, jadi penting untuk memiliki gambaran yang jelas tentang produk Microsoft apa yang telah Anda gunakan. AWS memiliki opsi berbeda yang tersedia untuk memenuhi kebutuhan berbagai produk Microsoft, termasuk penyewaan bersama/default untuk Amazon EC2 untuk produk dengan Mobilitas Lisensi dan opsi khusus untuk produk tanpa Mobilitas Lisensi. AWS juga memiliki opsi yang disertakan lisensi, di mana biaya lisensi termasuk dalam biaya komputasi Amazon EC2. Anda bisa mendapatkan keuntungan dari model lisensi campuran saat bermigrasi ke AWS. Model lisensi campuran adalah tempat instans EC2 penyewaan bersama digunakan dengan semua atau beberapa opsi yang disertakan lisensi. Model lisensi campuran paling baik untuk beban kerja variabel dan ketika opsi EC2

khusus digunakan untuk beban kerja yang stabil dan dapat diprediksi — terutama ketika Windows Server Datacenter atau SQL Server Enterprise BYOL adalah sebuah opsi.

Untuk informasi selengkapnya tentang persyaratan lisensi Microsoft saat ini untuk produk yang dibeli melalui program Lisensi Volume Microsoft, lihat situs [Persyaratan Produk Microsoft](#).

## Lisensi termasuk pilihan

Lisensi yang disertakan mengacu pada instans Amazon EC2 yang mencakup biaya lisensi dalam biaya komputasi. [Untuk beban kerja server Microsoft, AWS saat ini menawarkan Windows Server \(Amazon EC2, Host Khusus Amazon EC2, Instans Khusus Amazon EC2, AWS Outposts\) dan edisi SQL Server Enterprise, Standar, dan Web \(Amazon EC2\)](#). Lisensi server ini ditawarkan per vCPU per detik dengan pay-as-you-go model sebagai manfaat dari lisensi termasuk instans EC2. Jika instans EC2 dijadwalkan untuk berhenti, atau naik atau turun berdasarkan permintaan, Anda hanya membayar lisensi untuk waktu instans berjalan. Dengan harga sesuai permintaan, tidak ada komitmen jangka panjang, yang ideal untuk rencana modernisasi masa depan.

Lisensi yang disertakan tersedia untuk versi saat ini dan lama dengan Amazon Machine Images (AMI) tersedia untuk semua versi yang didukung. End-of-support Versi E, seperti Windows Server 2008 atau SQL Server 2012, masih dapat dilisensikan dengan lisensi yang disertakan, tetapi Anda harus membawa media Anda sendiri.

Tidak ada biaya peningkatan perangkat lunak dengan opsi yang disertakan lisensi. Segera setelah versi baru produk dirilis oleh Microsoft, versi baru segera tersedia di konsol Amazon EC2 tanpa biaya tambahan di atas biaya yang disertakan lisensi saat ini. Yang terpenting, AWS bertanggung jawab atas kepatuhan lisensi untuk instans EC2 yang disertakan dengan lisensi. Ini dapat menghemat banyak waktu dan tenaga untuk Anda karena kepatuhan perizinan bisa rumit dan sulit.

Lisensi SQL Server termasuk opsi menawarkan lisensi berbasis inti tanpa lisensi akses klien (CAL) diperlukan. Jumlah pengguna yang tidak terbatas dapat mengakses lisensi termasuk Windows Server EC2 instans tanpa menghitung atau melisensikan CAL. Lisensi Windows Server termasuk instans EC2 juga mencakup dua koneksi Microsoft Remote Desktop untuk tujuan administratif saja. Jika Anda memerlukan koneksi Microsoft Remote Desktop tambahan, Anda dapat membeli CAL Pengguna Layanan Desktop Jarak Jauh dengan Jaminan Perangkat Lunak (SA) dari Microsoft dan membawanya ke AWS melalui manfaat Mobilitas Lisensi.

AWS juga menawarkan beberapa opsi yang disertakan lisensi berbasis pengguna. Edisi Visual Studio 2022 Enterprise and Professional ([Amazon EC2](#) dan [Lambda](#)) dan Office LTSC Professional Plus 2021 (Amazon [EC2](#)) dikenakan biaya per pengguna, per bulan. Ini termasuk koneksi Microsoft



Remote Desktop untuk setiap pengguna. [Amazon WorkSpaces](#) juga menawarkan Office Professional Plus 2016 atau 2019 sebagai add-on, dibebankan per pengguna, per bulan.

AWS menawarkan opsi yang disertakan lisensi berikut untuk beban kerja Microsoft:

Produk	Ketersediaan	Versi tersedia
Windows Server	EC2, Instans Khusus EC2, Host Khusus EC2, Outposts	Semua*
SQL Server Enterprise	EC2	Semua*
SQL Server Standard	EC2	Semua*
Web SQL**	EC2	Semua*
Visual Studio Perusahaan	EC2, Lambda	2022
Visual Studio Profesional	EC2, Lambda	2022
Kantor Profesional Plus	WorkSpaces	2019, 2016
Kantor Profesional Plus LTSC	EC2	2021

\*O ut-of-support dan versi yang didukung memerlukan media Anda sendiri.

\*\* SQL Server Web edition memiliki kasus penggunaan terbatas berdasarkan persyaratan lisensi Microsoft. SQL Server Web edition hanya dapat digunakan untuk mendukung halaman web publik dan internet yang dapat diakses, situs web, aplikasi web, dan layanan web. Ini mungkin tidak digunakan untuk mendukung line-of-business aplikasi (misalnya, manajemen hubungan pelanggan, manajemen sumber daya perusahaan, dan aplikasi serupa lainnya).

Opsi yang disertakan lisensi adalah yang terbaik untuk beban kerja variabel. Misalnya, ini adalah saat beban kerja tidak perlu dijalankan sebagian besar waktu atau ketika beban kerja sering perlu ditingkatkan dan turun.

## Opsi BYOL

Menggunakan model Bring Your Own License (BYOL) adalah cara yang bagus untuk memanfaatkan investasi Anda yang ada di perangkat lunak lokal sambil memanfaatkan efisiensi AWS Cloud.

BYOL memungkinkan Anda untuk memperpanjang siklus hidup versi dan pembelian perangkat lunak sebelumnya, dan menerapkan produk yang tidak ditawarkan oleh AWS sebagai lisensi yang disertakan. Setiap kali Anda membawa lisensi Anda sendiri, Anda juga harus membawa media Anda sendiri. Ini berarti Anda harus membuat AMI sendiri dengan media Anda sendiri, daripada menggunakan AMI yang disediakan Amazon. Alat [Impor/Ekspor VM](#) gratis untuk digunakan dan memungkinkan Anda membuat AMI Anda sendiri. Atau, Anda dapat menggunakan [Layanan Migrasi Aplikasi](#) untuk membuat media dan AMI Anda sendiri.

## Produk Microsoft dengan Lisensi Mobilitas melalui Jaminan Perangkat Lunak

Karena AWS adalah [Mitra Mobilitas Resmi](#), produk Microsoft apa pun dengan Mobilitas Lisensi yang dicakup oleh SA aktif dapat dibawa ke AWS di lingkungan penyewa bersama atau khusus. Produk yang memenuhi syarat untuk Lisensi Mobilitas melalui SA termasuk SQL SharePoint Server, Server, Exchange Server, Project Server, Skype for Business Server, BizTalk Server, Remote Desktop Services User CALS, dan System Center Server. Produk Microsoft yang memiliki Hak Mobilitas Lisensi tidak terpengaruh oleh [perubahan lisensi](#) 1 Oktober 2019 yang dibuat oleh Microsoft. Akibatnya, produk dengan Mobilitas Lisensi tidak memiliki tanggal pembelian atau batasan versi. Mereka memenuhi syarat untuk BYOL ke AWS selama lisensi memiliki SA aktif. Misalnya, lisensi SQL Server 2022 dengan SA aktif dapat dibawa ke instans EC2 penyewaan bersama (default) (tidak memerlukan Instans Khusus) selama SA dipertahankan.

Produk dengan Mobilitas Lisensi melalui SA dilisensikan di AWS dengan cara yang sama seperti dalam lingkungan lokal yang tervirtualisasi, dengan pengecualian Server Pusat Sistem. Lisensi Server Pusat Sistem memiliki penghitungan lisensi khusus yang diterapkan saat dibawa ke AWS Cloud. Untuk setiap 16 core edisi System Center Server Datacenter, Anda dapat mengelola hingga 10 instans EC2 (dari berbagai ukuran). Untuk setiap 16 core edisi System Center Server Standard, Anda dapat mengelola hingga dua instans EC2 (dari berbagai ukuran).

SQL Server adalah produk yang paling sering dibawa dengan Mobilitas Lisensi ke AWS. Lisensi inti SQL Server dengan SA aktif atau lisensi berlangganan (kecuali yang dibeli melalui Penyedia Solusi Cloud, atau program CSP) dilisensikan per vCPU pada instans EC2 penyewaan bersama (default), dengan persyaratan lisensi Microsoft minimum empat vCPU per instans EC2. Lisensi SQL Server/CAL dengan SA aktif dilisensikan dengan satu lisensi server per instans EC2. Plus, semua pengguna atau perangkat dengan akses harus memiliki CAL yang sesuai yang ditetapkan untuk mereka. SQL Server juga memiliki manfaat failover pasif dengan SA aktif dan langganan. Untuk setiap SQL Server yang aktif dan berlisensi di EC2, Anda memenuhi syarat untuk instance SQL Server pasif sekunder di EC2 tanpa harus melisensikan bagian SQL Server pada instance pasif. Untuk informasi

selengkapnya, lihat [panduan Lisensi Microsoft SQL Server 2022](#) (PDF yang dapat diunduh) di situs web Microsoft.

AWS adalah [Mitra Mobilitas Resmi](#) (PDF yang dapat diunduh). Jika Anda membawa produk Microsoft dengan [Mobilitas Lisensi](#) ke AWS, Anda harus mengisi dan mengirimkan Formulir Verifikasi Mobilitas Lisensi ke Microsoft. Formulir ini adalah dokumen Microsoft Word singkat yang menanyakan hal-hal berikut:

- Nama dan informasi kontak Anda
- Nomor perjanjian Microsoft
- Mitra cloud Anda
- Produk dibawa melalui Lisensi Mobilitas
- Jumlah lisensi yang Anda bawa

Anda harus mengirimkan formulir ke Microsoft secara langsung atau melalui reseller Microsoft Anda dalam waktu 10 hari setelah membawa produk ke AWS. Untuk mempelajari lebih lanjut tentang proses verifikasi, lihat [Mobilitas Lisensi melalui Jaminan Perangkat Lunak](#) di dokumentasi Microsoft. Formulir Verifikasi Mobilitas Lisensi memiliki bagian untuk memberikan informasi tentang Mitra Mobilitas Resmi. Anda dapat menggunakan [microsoft@amazon.com](mailto:microsoft@amazon.com) sebagai alamat email, Amazon Web Services sebagai nama Mitra, dan [aws.amazon.com](http://aws.amazon.com) sebagai situs web Mitra. Untuk panduan selengkapnya, lihat [Panduan Verifikasi Microsoft untuk Pelanggan](#) (PDF yang dapat diunduh) di dokumentasi Microsoft. Untuk mengunduh salinan Formulir Verifikasi Mobilitas [Lisensi](#), lihat [Sumber Daya dan Dokumen](#) Lisensi dalam dokumentasi Microsoft.

#### Catatan

Program Virtualisasi Fleksibel yang ditawarkan oleh Microsoft tidak tersedia di AWS karena AWS telah dinamai cloud Penyedia\* Terdaftar oleh Microsoft. Microsoft menyebut Alibaba, Amazon, dan Google Cloud sebagai [Penyedia Terdaftar](#) sebagai bagian dari perubahan [lisensi](#) 1 Oktober 2019. Mulai 1 Oktober 2019, lisensi lokal yang dibeli tanpa hak SA dan Mobilitas Lisensi tidak dapat digunakan layanan cloud host yang ditawarkan oleh Penyedia Terdaftar.

## Produk Microsoft tanpa Lisensi Mobilitas

Windows Server, Visual Studio, Microsoft Developer Network (MSDN), sistem operasi desktop Windows, Microsoft Office, dan aplikasi Microsoft 365 (sebelumnya Office 365) tidak memiliki hak Lisensi Mobilitas yang diberikan kepada mereka dalam Persyaratan Produk Microsoft, bahkan jika lisensi memiliki SA atau lisensi berlangganan aktif. Akibatnya, membawa lisensi untuk produk ini memerlukan infrastruktur khusus: Host Khusus EC2, Instans Khusus EC2, VMware Cloud on AWS, dan Host Khusus di Outposts. Anda juga harus mengikuti persyaratan khusus lainnya agar memenuhi syarat untuk BYOL to AWS. Persyaratan ini merupakan hasil dari perubahan yang dibuat Microsoft terhadap persyaratan lisensi untuk produk tanpa Lisensi Mobilitas saat diterapkan di cloud Penyedia Terdaftar, efektif 1 Oktober 2019. Untuk informasi selengkapnya, lihat [Persyaratan lisensi Microsoft yang diperbarui untuk layanan cloud yang dihosting khusus](#) di dokumentasi Microsoft.

Agar memenuhi syarat untuk BYOL ke AWS, lisensi untuk produk tanpa Lisensi Mobilitas harus memenuhi persyaratan berikut dari Microsoft:

- Lisensi harus dibeli sebagai hak penggunaan abadi (bukan berlangganan).
- Tanggal pembelian lisensi harus sebelum 1 Oktober 2019, atau lisensi harus dibeli dalam jangka waktu Perjanjian Microsoft Enterprise yang dimulai sebelum 1 Oktober 2019.
- Versi yang digunakan harus tersedia untuk umum sebelum 1 Oktober 2019.
- Produk harus digunakan pada infrastruktur khusus.

Lisensi berlangganan untuk produk tanpa Lisensi Mobilitas akan kehilangan BYOL setelah dibeli atau diperbarui pada atau setelah 1 Oktober 2019.

### Catatan

Produk tanpa Lisensi Mobilitas tidak memerlukan SA aktif untuk BYOL di AWS, selama lisensi memenuhi persyaratan di atas.

Karena perizinan bisa rumit, lihat [situs Amazon Web Services dan Microsoft FAQ](#) untuk menentukan apakah lisensi Anda memenuhi syarat untuk opsi BYOL to AWS. [Jika Anda tidak menemukan informasi yang Anda butuhkan di FAQ atau tidak yakin harus mulai dari mana dengan memigrasikan beban kerja Microsoft Anda ke AWS, hubungi Microsoft@Amazon.com.](#) AWS memiliki spesialis beban kerja dan lisensi Microsoft yang tersedia untuk membantu memastikan Anda memiliki semua informasi yang Anda butuhkan.

**i** Catatan

Windows Server BYOL memerlukan Host Khusus EC2, Host Khusus di Outpost, atau VMware Cloud on AWS karena Windows Server BYOL harus dilisensikan oleh inti fisik.

## BYOL untuk Perjanjian Lisensi Penyedia Layanan (SPLA)

Program Perjanjian Lisensi Penyedia Layanan (SPLA) tidak terpengaruh oleh [perubahan lisensi](#) 1 Oktober 2019 yang dibuat oleh Microsoft. Akibatnya, lisensi Windows Server baru bersih dapat dibawa melalui SPLA untuk pelanggan dengan lisensi SPLA mereka sendiri, tanpa tanggal pembelian atau batasan versi. Setiap produk inti atau berbasis prosesor yang dilisensikan melalui SPLA memerlukan Host Khusus EC2, di mana Lisensi Akses Pelanggan (SAL) berbasis pengguna dapat dibawa ke instans EC2 penyewaan bersama (default). Hal ini karena SAL berbasis pengguna di SPLA memenuhi syarat untuk penyedia pusat data (DCP) dalam Hak Penggunaan [Penyedia Layanan](#) (SPUR).

**i** Catatan

Microsoft telah [mengumumkan](#) bahwa mereka tidak akan lagi mengizinkan SPLA BYOL di AWS atau cloud Penyedia Terdaftar lainnya setelah 30 September 2025.

## Host Khusus Amazon EC2

Beberapa kemampuan utama [Host Khusus Amazon EC2](#) meliputi:

- Hypervisor EC2 Nitro dan Xen yang telah dikonfigurasi sebelumnya dengan visibilitas ke soket dan inti fisik
- Beberapa ukuran instans dalam keluarga yang sama didukung pada Host Khusus yang sama (Untuk kumpulan terbaru jenis instans yang didukung, lihat [Host Khusus](#) di Panduan Pengguna Amazon EC2 untuk Instans Windows.)
- Manajemen otomatis, auto-scaling, dan kontrol penempatan instans
- Kemampuan untuk berbagi host di beberapa akun AWS
- Terintegrasi dengan [AWS License Manager](#) untuk melacak penggunaan dan pengelolaan lisensi
- Kemampuan untuk mempertahankan afinitas instance ke host

- Pemulihan host otomatis
- Pemantauan berkelanjutan dengan AWS Config

Karena Windows Server BYOL memerlukan infrastruktur khusus dan jumlah inti fisik, Host Khusus EC2 adalah pilihan bagus yang dapat membantu Anda:

- Mencapai penghematan yang signifikan
- Memungkinkan Anda membawa aplikasi Microsoft apa pun ke AWS, terlepas dari SA atau Mobilitas Lisensi (tunduk pada persyaratan pembelian dan versi 1 Oktober 2019)
- Maksimalkan manfaat lisensi inti fisik dari edisi Windows Server Datacenter dan SQL Server Enterprise
- Bayar hanya per host, bukan per instans EC2 (Ini berarti bahwa ketika Anda menggunakan host khusus, Anda dapat menggunakan jumlah maksimum instans yang tersedia di host tanpa dikenakan biaya komputasi tambahan.)

Jika Anda membawa lisensi Windows Server yang memenuhi syarat BYOL ke Host Khusus EC2, Anda dapat melisensikan semua inti fisik (bukan VCPU) dari host. Misalnya, Host Khusus R5 EC2 memiliki 48 inti fisik. Membawa 48 core edisi Windows Server Datacenter ke Host Khusus R5 EC2 memungkinkan sebanyak mungkin instans EC2 untuk digunakan pada host secara teknis. Membawa 48 core Windows Server edisi Standar memungkinkan hingga dua instans EC2 dari berbagai ukuran pada host.

Anda dapat menumpuk lisensi edisi Standar Windows Server untuk memungkinkan instans EC2 tambahan pada host yang sama, di mana semua inti fisik host berlisensi untuk kedua kalinya memungkinkan dua instans EC2 tambahan (dan seterusnya). Lisensi SQL Server Enterprise dengan inti fisik juga mengharuskan semua inti fisik host dilisensikan. Ini memungkinkan Anda untuk menyebarkan jumlah instans EC2 untuk SQL Server pada host sama dengan jumlah inti fisik yang dilisensikan. Misalnya, R5 EC2 Dedicated Host berlisensi dengan 48 core SQL Server Enterprise memungkinkan Anda untuk menyebarkan hingga 48 instans EC2 yang menjalankan SQL Server pada host tersebut.

Jika Anda membawa lisensi Windows Server Datacenter dan SQL Server Enterprise yang memenuhi syarat BYOL dan melisensikan total inti fisik host, Anda dapat melihat penghematan biaya yang signifikan atas lisensi yang disertakan untuk jumlah dan ukuran instans EC2 yang sama. Ini mengasumsikan beban kerja sebagian besar dapat mengisi host dan berjalan sebagian besar waktu. Misalnya, Anda dapat menerapkan 12 instans EC2 R5.2xLarge pada instance penyewaan bersama

dengan lisensi termasuk Windows Server dan SQL Server Enterprise BYOL dengan total 96 core SQL Server Enterprise yang diperlukan untuk lisensi. Namun, jika Anda menerapkan Host Khusus R5 EC2 (yang dapat memuat 12 instans EC2 R5.2xLarge yang sama), Anda dapat membawa 48 core Windows Server Datacenter dan 48 core dari lisensi yang memenuhi syarat SQL Server Enterprise BYOL. Anda tidak hanya akan menghemat biaya termasuk lisensi Windows Server, tetapi Anda juga hanya perlu membawa setengah dari jumlah lisensi inti SQL Server Enterprise.

BYOL pada Host Khusus EC2 adalah yang terbaik untuk beban kerja yang stabil dan dapat diprediksi di mana Anda dapat mengisi host setidaknya 70 persen dan di mana beban kerja berjalan sebagian besar waktu. Untuk mempelajari selengkapnya tentang Lisensi Microsoft di AWS, lihat [Microsoft Licensing on AWS](#) on YouTube and [Amazon Web Services dan Pertanyaan yang Sering Diajukan](#) Microsoft dalam dokumentasi Microsoft.

## VMware Cloud on AWS

Untuk mempelajari lebih lanjut tentang migrasi ke VMware Cloud di AWS, lihat gambaran umum dan [model pengoperasian VMware Cloud on AWS di AWS dalam dokumentasi AWS Prescriptive Guidance](#).

### Pemberitahuan

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

## Memobilisasi

### AWS License Manager

[Sebagai bagian dari fase mobilisasi untuk pertimbangan lisensi Microsoft, kami menyarankan Anda memasukkan lisensi yang Anda rencanakan untuk dialokasikan ke beban kerja Anda di AWS di AWS License Manager](#). License Manager adalah alat gratis yang memudahkan Anda mengelola lisensi perangkat lunak dari vendor seperti Microsoft, Oracle, IBM, dan SAP tidak hanya di AWS tetapi juga beban kerja di tempat atau di cloud lain.

Memasukkan lisensi Microsoft yang Anda bawa ke AWS ke License Manager akan membantu Anda:



- Dapatkan visibilitas dan kontrol yang lebih besar atas bagaimana lisensi perangkat lunak digunakan dan mencegah penyalahgunaan sebelum terjadi.
- Hemat uang dengan penggunaan maksimum lisensi, termasuk bagaimana Anda melacak dan mengelola lisensi.
- Mengurangi risiko ketidakpatuhan dengan menerapkan batas penggunaan lisensi, memblokir peluncuran baru, dan menggunakan kontrol lainnya.
- Tingkatkan produktivitas Anda dengan mengotomatiskan penempatan, rilis, dan pemulihan host menggunakan grup sumber daya host.

Untuk mempelajari selengkapnya tentang License Manager, lihat [Bekerja dengan AWS License Manager](#) di Panduan Pengguna AWS License Manager.

## Pertimbangan perizinan

Pertimbangkan untuk merencanakan migrasi Anda di sekitar lisensi yang saat ini ditetapkan ke beban kerja sebelum migrasi. Misalnya, jika Anda membawa beberapa host lokal ke AWS, pertimbangkan untuk bermigrasi berdasarkan host, bukan dengan mengelompokkan beban kerja yang ada di beberapa host berbeda. Ini karena saat Anda menonaktifkan host lokal, Anda membebaskan lisensi yang terkait dengan host tersebut untuk digunakan di AWS. Atau, Anda dapat menggunakan instance yang disertakan lisensi untuk Windows Server atau SQL Server selama migrasi dan beralih ke opsi BYOL setelah migrasi selesai. Namun, opsi ini memerlukan penggunaan media Anda sendiri dan AMI dari awal (bahkan untuk opsi yang disertakan lisensi). [Fitur konversi lisensi](#) yang tersedia dengan AWS License Manager hanya memungkinkan Anda untuk beralih ke BYOL dari lisensi yang disertakan jika instans EC2 awalnya dibuat dari media dan AMI Anda sendiri.

## Migrasi

Dalam 10 hari setelah menerapkan beban kerja Microsoft Anda di AWS, pastikan untuk mengirimkan [Formulir Verifikasi Mobilitas Lisensi](#) ke Microsoft untuk lisensi apa pun dengan Mobilitas Lisensi yang Anda bawa ke AWS. Anda dapat mengirimkan formulir ini beberapa kali, berdasarkan berbagai tahapan migrasi Anda. Formulir meminta yang berikut:

- Nama dan informasi kontak Anda
- Nomor perjanjian Microsoft
- Mitra cloud Anda



- Produk dibawa melalui Lisensi Mobilitas
- Jumlah lisensi yang Anda bawa

Untuk mempelajari lebih lanjut tentang proses verifikasi, lihat [Mobilitas Lisensi melalui Jaminan Perangkat Lunak](#) di dokumentasi Microsoft. Untuk panduan selengkapnya, lihat [Panduan Verifikasi Microsoft untuk Pelanggan](#) (PDF yang dapat diunduh) di dokumentasi Microsoft. Untuk mengunduh salinan Formulir Verifikasi Mobilitas [Lisensi](#), lihat [Sumber Daya dan Dokumen](#) Lisensi dalam dokumentasi Microsoft.

# Mitra AWS

## Manfaat melibatkan Partner Kompetensi AWS

Migrasi beban kerja Microsoft Anda secara efisien ke cloud memerlukan perencanaan yang matang dan implementasi yang efisien. Langkah-langkah kunci termasuk pelingkupan, menciptakan kasus bisnis migrasi cloud, mendapatkan penyelarasan sponsor eksekutif, menetapkan KPI manajemen keuangan cloud, membangun pusat keunggulan cloud, memvalidasi layanan migrasi, menerapkan alat otomatisasi untuk migrasi skala besar, dan memperluas strategi keamanan ke cloud.

Kami menyarankan Anda melibatkan [Partner Kompetensi AWS](#) yang divalidasi untuk memimpin organisasi Anda melalui perjalanan migrasi Anda. AWS Partners adalah pakar strategis dan pembangun berpengalaman yang membantu mengatasi langkah-langkah kunci yang disebutkan di atas dan tujuan bisnis Anda dengan memimpin Anda melalui semua fase perjalanan migrasi Anda. Komunitas AWS Partner memiliki lebih dari 100.000 mitra dari lebih dari 150 negara yang dapat mendukung Anda dalam perjalanan cloud Anda dan membantu Anda fokus pada inovasi, meningkatkan kelincahan, dan mengurangi biaya.

## Membangun rencana

AWS Partners dapat melakukan penilaian kesiapan, membuat rencana migrasi, dan menghadirkan alat migrasi untuk mempercepat perjalanan Anda ke cloud. Selain itu, mereka dapat membantu Anda menutup kesenjangan keterampilan, merekomendasikan strategi pengoptimalan biaya, dan membantu Anda memenuhi syarat untuk mendapatkan insentif migrasi eksklusif untuk mensubsidi biaya migrasi ke AWS.

## Optimalkan biaya

Dalam lanskap teknologi yang berkembang pesat saat ini, banyak organisasi menghadapi tantangan biaya yang signifikan dalam hal perjalanan transformasi digital mereka. Salah satu kekhawatiran umum adalah persepsi bahwa cloud terlalu mahal, sehingga sulit untuk melihat manfaat bisnis yang signifikan yang ditawarkannya. Selain itu, biaya memodernisasi tumpukan teknologi Anda dapat menimbulkan tantangan keuangan.

Bekerja dengan Mitra [Kompetensi Beban Kerja Microsoft AWS memastikan akses ke Mitra AWS](#) yang paling memenuhi syarat untuk menerapkan beban kerja Microsoft di AWS. Mitra ini telah

memvalidasi kemampuan teknis dan menunjukkan keberhasilan dalam membantu pelanggan bermigrasi, mengelola, atau menyebarkan beban kerja Microsoft ke AWS. Beban kerja yang didukung oleh mitra ini termasuk Windows Server, Microsoft SQL Server, Windows File Server SharePoint, dan aplikasi.NET.

AWS Partners menggunakan praktik terbaik AWS untuk membangun arsitektur yang aman, tersedia, andal, berkinerja, dan dioptimalkan biaya. Mitra juga membantu memanfaatkan sepenuhnya pendanaan yang disediakan oleh AWS untuk mengoptimalkan biaya dan memastikan waktu yang lebih cepat untuk menghargai dengan keahlian mereka. Terakhir, AWS Partners dapat memanfaatkan [AWS Migration Acceleration Program for Windows](#) untuk mengimbangi biaya migrasi Anda ke AWS.

## Hemat waktu

### Pemberitahuan

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

Banyak perusahaan banyak berinvestasi dalam infrastruktur lokal. Ada kemungkinan bahwa organisasi Anda telah melakukan investasi besar dalam perangkat lunak VMware untuk mengelola infrastruktur lokal Anda, dan ingin menggunakan alat lokal yang sama untuk mengelola infrastruktur Anda di AWS. Anda bahkan mungkin memiliki beban kerja dan infrastruktur khusus yang menantang untuk bermigrasi ke cloud tetapi memiliki ketergantungan pada beban kerja yang dimigrasi. Selain itu, Anda mungkin memiliki pola infrastruktur hybrid, di mana beberapa infrastruktur Anda berada di pusat data lokal tradisional dengan bagian lain yang digunakan di cloud.

Ketika waktu sangat penting, kami sarankan Anda melibatkan [Mitra Kompetensi Migrasi AWS](#) dengan rekam jejak yang terbukti dalam memberikan berbagai migrasi skala besar karena bakat terampil, proses yang disempurnakan, dan kemampuan teknologinya. Kategori beban kerja yang didukung meliputi Windows, SAP, Oracle, VMware on AWS, Database, Analytics, Storage, IoT, Machine learning, dan Software as a Service.

AWS Partners memahami bahwa pindah ke AWS tidak berarti all-or-nothing pindah dan menyingkirkan investasi Anda saat ini. Mereka mahir dalam mengoptimalkan dan merampingkan

infrastruktur, mengoptimalkan suku cadang mana yang paling baik disimpan di tempat dan bagian mana yang paling cocok untuk cloud. AWS memiliki penawaran luas solusi cloud hybrid, yang mencakup Amazon VPC, Direct Connect, dan Storage Gateway.

AWS Partners dapat memenuhi syarat pelanggan yang memenuhi syarat untuk [AWS Migration Acceleration Program \(MAP\)](#), program migrasi cloud yang komprehensif dan terbukti berdasarkan pengalaman AWS dalam memigrasikan ribuan pelanggan perusahaan ke cloud. MAP mendukung beban kerja khusus melalui perkakas, layanan, bimbingan, pelatihan, dan insentif tambahan yang komprehensif. Dukungan beban kerja khusus tersedia untuk mainframe, Windows, penyimpanan, VMware Cloud on AWS, SAP, database, dan Amazon Connect.

## Tingkatkan keamanan

Anda mungkin khawatir tentang privasi dan keamanan data Anda. Selain itu, Anda mungkin memerlukan jaminan bahwa praktik penanganan data mematuhi Undang-Undang Klarifikasi Penggunaan Data Luar Negeri (CLOUD) yang Sah dan Peraturan Perlindungan Data Umum (GDPR). Kami menyarankan Anda melibatkan [Partner Kompetensi AWS Security](#) yang dapat memberi Anda tim pakar keamanan untuk memberikan solusi yang berfokus pada keamanan untuk beban kerja dan kasus penggunaan spesifik Anda. Solusi AWS Partner memungkinkan otomatisasi dan kelincahan serta penskalaan dengan beban kerja Anda.

Pada saat penerbitan, AWS mendukung berbagai standar keamanan dan sertifikasi kepatuhan, seperti PCI-DSS, HIPAA/HITECH, FedRAMP, GDPR, FIPS 140-2, dan NIST 800-171. Kami membantu memenuhi persyaratan kepatuhan untuk sebagian besar badan pengatur di seluruh dunia.

Organisasi swasta dan sektor publik, di beberapa vertikal yang paling sensitif terhadap keamanan seperti perawatan kesehatan, perbankan, hukum, dan farmasi, telah mempercayai AWS untuk meningkatkan postur keamanan mereka. Baik Anda perusahaan kecil, menengah, atau besar, atau organisasi sektor publik, ada Partner AWS dengan keterampilan dan pengalaman yang tepat yang tersedia untuk membantu Anda memajukan bisnis Anda. Spesialis AWS Partner dapat membantu Anda menemukan dan terhubung dengan mitra cloud yang tepat sesuai dengan kebutuhan bisnis Anda. Untuk informasi selengkapnya, hubungi [spesialis AWS Partner](#). Untuk mempelajari bagaimana pelanggan di seluruh dunia mempercepat adopsi cloud mereka dan mendorong inovasi dengan AWS Partner Network (APN), lihat [Sukses Pelanggan dengan AWS Partners](#).

## Langkah selanjutnya

Kami menyarankan Anda mengambil langkah-langkah berikut ini:

1. Pelajari selengkapnya tentang skenario migrasi dan modernisasi tertentu. Untuk informasi lebih lanjut, lihat [Memigrasi database Microsoft SQL Server ke AWS Cloud](#), [Memodernisasi aplikasi Anda dengan memigrasi dari RDBMS ke Amazon DynamoDB](#), dan [Memilih pendekatan untuk memodernisasi aplikasi .NET](#).
2. Pelajari lebih lanjut tentang dampak organisasi dari migrasi besar. Migrasi besar tidak hanya transformasi teknologi tetapi juga menyertai perubahan peran, proses, dan prioritas organisasi Anda. Untuk informasi lebih lanjut, lihat [Strategi dan Praktik Terbaik untuk Migrasi Besar AWS](#).
3. Tinjau [Panduan Belajar Mandiri AWS for Microsoft Workloads](#).
4. Lengkapi [Memigrasi Beban Kerja Microsoft ke AWS Hands-on Workshop](#).

# Sumber daya

## Pedoman migrasi Microsoft ke AWS

- [Memigrasi Beban Kerja Microsoft ke AWS: Panduan Belajar Mandiri](#)
- [Memigrasi Beban Kerja Microsoft ke AWS: Lab Langsung](#)
- [Memigrasi database Microsoft SQL Server ke AWS Cloud](#)
- [Memodernisasi aplikasi Anda dengan memigrasi dari RDBMS ke Amazon DynamoDB](#)
- [Memilih pendekatan untuk memodernisasi aplikasi .NET](#)
- [Strategi dan Praktik Terbaik untuk Migrasi Besar AWS.](#)

## Pedoman umum

- [Windows di AWS](#)
- [Strategi dan praktik terbaik untuk migrasi besar AWS](#)
- [Selamat Datang di Dokumentasi AWS](#)

## Video

- [AWS RE:Invent 2020: Memigrasi beban kerja Microsoft ke AWS](#)
- [Rehost Beban Kerja Windows dengan Layanan Migrasi Aplikasi AWS - AWS Virtual Workshop](#)

## Postingan Blog AWS

- [Cara memigrasi beban kerja lokal dengan AWS Application Migration Service](#)
- [Mengapa Anda harus memigrasi beban kerja Windows Anda dengan AWS \(dan bagaimana kami dapat membantu\)](#)

## Riwayat dokumen

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

Perubahan	Deskripsi	Tanggal
<a href="#">Perbarui</a>	Menambahkan informasi tentang Amazon EBS Multi-Lampirkan ke bagian <a href="#">Migrating Windows failover cluster</a> .	April 1, 2024
<a href="#">Perbarui</a>	Menambahkan tautan ke modul Migration Validator Toolkit PowerShell . Petunjuk yang diklarifikasi untuk menggunakan Tutorial: Siapkan cluster Windows HPC di Amazon EC2 di halaman Migrating Windows failover cluster.	14 Desember 2023
<a href="#">Perbarui</a>	Memperbarui halaman cluster failover Migrasi Windows	8 Desember 2023
<a href="#">Perbarui</a>	Daftar terbaru jenis instans yang didukung untuk Host Khusus di bagian Host Khusus Amazon EC2 pada halaman lisensi Microsoft di AWS	16 November 2023
<a href="#">Perbarui</a>	Menambahkan daftar lengkap keluarga instans yang didukung ke bagian Host Khusus Amazon EC2 pada halaman lisensi Microsoft di AWS	31 Juli 2023

---

<a href="#">Perbarui</a>	Menambahkan panduan BYOM ke bagian Replatfor ming dari halaman server Migrating SQL	23 Juni 2023
<a href="#">Publikasi awal</a>	—	9 Juni 2023



# AWS Glosarium Panduan Preskriptif

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

## Nomor

### 7 Rs

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

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

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

## A

### ABAC

Lihat [kontrol akses berbasis atribut](#).

### layanan abstrak

Lihat [layanan terkelola](#).

### ASAM

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

### migrasi aktif-aktif

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

### migrasi aktif-pasif

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

### fungsi agregat

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

## AI

Lihat [kecerdasan buatan](#).

### AIOps

Lihat [operasi kecerdasan buatan](#).

## anonimisasi

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

## anti-pola

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

## kontrol aplikasi

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

## portofolio aplikasi

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

## kecerdasan buatan (AI)

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

## operasi kecerdasan buatan (AIOps)

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

## enkripsi asimetris

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

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

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

## kontrol akses berbasis atribut (ABAC)

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

## sumber data otoritatif

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

## Zona Ketersediaan

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

## AWS Kerangka Adopsi Cloud (AWS CAF)

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

## AWS Kerangka Kualifikasi Beban Kerja (AWS WQF)

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

## B

### bot buruk

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

### BCP

Lihat [perencanaan kontinuitas bisnis](#).

### grafik perilaku

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

### sistem big-endian

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

### klasifikasi biner

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

### filter mekar

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

### deployment biru/hijau

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

### bot

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

## botnet

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

## cabang

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

## akses break-glass

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

## strategi brownfield

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

## cache penyangga

Area memori tempat data yang paling sering diakses disimpan.

## kemampuan bisnis

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

## perencanaan kelangsungan bisnis (BCP)

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

## C

### KAFE

Lihat [Kerangka Adopsi AWS Cloud](#).

#### penyebaran kenari

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

### CCoE

Lihat [Cloud Center of Excellence](#).

### CDC

Lihat [mengubah pengambilan data](#).

#### ubah pengambilan data (CDC)

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

#### rekayasa kekacauan

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

### CI/CD

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

#### klasifikasi

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

#### Enkripsi sisi klien

Enkripsi data secara lokal, sebelum target Layanan AWS menerimanya.

## Cloud Center of Excellence (CCoE)

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

### komputasi cloud

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

### model operasi cloud

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

### tahap adopsi cloud

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

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

Tahapan ini didefinisikan oleh Stephen Orban dalam posting blog [The Journey Toward Cloud-First & the Stages of Adoption](#) di blog Strategi 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 AWS CodeCommit Setiap versi kode disebut cabang. Dalam struktur layanan mikro, setiap repositori



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

#### cache dingin

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

#### data dingin

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

#### visi komputer (CV)

Bidang [AI](#) yang menggunakan pembelajaran mesin untuk menganalisis dan mengekstrak informasi dari format visual seperti gambar dan video digital. Misalnya, AWS Panorama menawarkan perangkat yang menambahkan CV ke jaringan kamera lokal, dan Amazon SageMaker menyediakan algoritme pemrosesan gambar untuk CV.

#### konfigurasi drift

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

#### database manajemen konfigurasi (CMDB)

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

#### paket kesesuaian

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

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

Proses mengotomatiskan sumber, membangun, menguji, pementasan, dan tahap produksi dari proses rilis perangkat lunak. CI/CD umumnya digambarkan sebagai pipa. CI/CD dapat membantu

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

## 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 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 di tempat. Untuk informasi selengkapnya, lihat Kontrol [Detektif dalam Menerapkan kontrol](#) keamanan pada. AWS

## pemetaan aliran nilai pengembangan (DVSM)

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

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

## kembar digital

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

## tabel dimensi

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

## musibah

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

## pemulihan bencana (DR)

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

## DML~

Lihat [bahasa manipulasi database](#).

## desain berbasis domain

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

### komputasi tepi

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

### enkripsi

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

### kunci enkripsi

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

### endianness

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

### titik akhir

Lihat [titik akhir layanan](#).

## layanan endpoint

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

## 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.

## FGAC

Lihat kontrol [akses berbutir halus](#).

### kontrol akses berbutir halus (FGAC)

Penggunaan beberapa kondisi untuk mengizinkan atau menolak permintaan akses.

## migrasi flash-cut

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

## G

### pemblokiran geografis

Lihat [pembatasan geografis](#).

### pembatasan geografis (pemblokiran geografis)

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

## Alur kerja Gitflow

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

### strategi greenfield

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

### pagar pembatas

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

## H

### HA

Lihat [ketersediaan tinggi](#).

### migrasi database heterogen

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

### ketersediaan tinggi (HA)

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

## modernisasi sejarawan

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

## migrasi database homogen

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

## data panas

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

## perbaikan terbaru

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

## periode hypercare

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

|

## IAC

Lihat [infrastruktur sebagai kode](#).

## kebijakan berbasis identitas

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

|

## aplikasi idle

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

## IloT

Lihat [Internet of Things industri](#).

## infrastruktur yang tidak dapat diubah

Model yang menyebarkan infrastruktur baru untuk beban kerja produksi alih-alih memperbarui, menambal, atau memodifikasi infrastruktur yang ada. [Infrastruktur yang tidak dapat diubah secara inheren lebih konsisten, andal, dan dapat diprediksi daripada infrastruktur yang dapat berubah](#). Untuk informasi selengkapnya, lihat praktik terbaik [Deploy using immutable infrastructure](#) di AWS Well-Architected Framework.

## masuk (masuknya) VPC

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

## migrasi inkremental

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

## Industri 4.0

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

## infrastruktur

Semua sumber daya dan aset yang terkandung dalam lingkungan aplikasi.

## infrastruktur sebagai kode (IAC)

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

## Internet of Things industri (IIoT)

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

## inspeksi VPC

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

## Internet of Things (IoT)

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

## interpretabilitas

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

## IoT

Lihat [Internet of Things](#).

## Perpustakaan informasi TI (ITIL)

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

## Manajemen layanan TI (ITSM)

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

### ITIL

Lihat [perpustakaan informasi TI](#).

### ITSM

Lihat [manajemen layanan TI](#).

## L

### kontrol akses berbasis label (LBAC)

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

### landing zone

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

### migrasi besar

Migrasi 300 atau lebih server.

### LBAC

Lihat [kontrol akses berbasis label](#).

### hak istimewa paling sedikit

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

angkat dan geser

Lihat [7 Rs](#).

sistem endian kecil

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

lingkungan yang lebih rendah

Lihat [lingkungan](#).

## M

pembelajaran mesin (ML)

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

cabang utama

Lihat [cabang](#).

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

### arsitektur microservices

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



## Program Percepatan Migrasi (MAP)

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

### migrasi dalam skala

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

### pabrik migrasi

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

### metadata migrasi

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

### pola migrasi

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

## Penilaian Portofolio Migrasi (MPA)

Alat online yang menyediakan informasi untuk memvalidasi kasus bisnis untuk bermigrasi ke. 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 [Mengurai monolit](#) menjadi layanan mikro.

## MPA

Lihat [Penilaian Portofolio Migrasi](#).

## MQTT

Lihat [Transportasi Telemetri Antrian Pesan](#).

## klasifikasi multiclass

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

## infrastruktur yang bisa berubah

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

## O

### OAC

Lihat [kontrol akses asal](#).

### OAI

Lihat [identitas akses asal](#).

### OCM

Lihat [manajemen perubahan organisasi](#).

## migrasi offline

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

## OI

Lihat [integrasi operasi](#).

## OLA

Lihat [perjanjian tingkat operasional](#).

## migrasi online

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

## OPC-UA

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

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

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

## perjanjian tingkat operasional (OLA)

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

## Tinjauan Kesiapan Operasional (ORR)

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

## teknologi operasional (OT)

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

## integrasi operasi (OI)

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

## jejak organisasi

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

## manajemen perubahan organisasi (OCM)

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

## kontrol akses asal (OAC)

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

## identitas akses asal (OAI)

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

## ORR

Lihat [tinjauan kesiapan operasional](#).

## OT

Lihat [teknologi operasional](#).

## keluar (jalan keluar) VPC

Dalam arsitektur AWS multi-akun, VPC yang menangani koneksi jaringan yang dimulai dari dalam aplikasi. [Arsitektur Referensi AWS Keamanan](#) merekomendasikan pengaturan akun Jaringan

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

## P

### batas izin

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

### Informasi Identifikasi Pribadi (PII)

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

### PII

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

### buku pedoman

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

### PLC

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

### PLM

Lihat [manajemen siklus hidup produk](#).

### kebijakan

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

## persistensi poliglot

Secara independen memilih teknologi penyimpanan data microservice berdasarkan pola akses data dan persyaratan lainnya. Jika layanan mikro Anda memiliki teknologi penyimpanan data yang sama, mereka dapat menghadapi tantangan implementasi atau mengalami kinerja yang buruk. Layanan mikro lebih mudah diimplementasikan dan mencapai kinerja dan skalabilitas yang lebih baik jika mereka menggunakan penyimpanan data yang paling sesuai dengan kebutuhan mereka. 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 pengoptimalan kueri database yang menyaring data dalam kueri sebelum transfer. Ini mengurangi jumlah data yang harus diambil dan diproses dari database relasional, dan meningkatkan kinerja kueri.

## kontrol preventif

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

## principal

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

## Privasi oleh Desain

Pendekatan dalam rekayasa sistem yang memperhitungkan privasi di seluruh proses rekayasa.

## zona host pribadi

Container yang menyimpan informasi tentang bagaimana Anda ingin Amazon Route 53 merespons kueri DNS untuk domain dan subdomainnya dalam satu atau beberapa VPC. Untuk

informasi selengkapnya, lihat [Bekerja dengan zona yang dihosting pribadi](#) di dokumentasi Route 53.

## kontrol proaktif

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

## manajemen siklus hidup produk (PLM)

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

## lingkungan produksi

Lihat [lingkungan](#).

## pengontrol logika yang dapat diprogram (PLC)

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

## pseudonimisasi

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

## terbitkan/berlangganan (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 \(RACI\)](#).

### ransomware

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

### Matriks RASCI

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

### RCAC

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

### replika baca

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

### arsitek ulang

Lihat [7 Rs](#).

## tujuan titik pemulihan (RPO)

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

## tujuan waktu pemulihan (RTO)

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

## refactor

Lihat [7 Rs](#).

## Wilayah

Kumpulan AWS sumber daya di wilayah geografis. Masing-masing Wilayah AWS terisolasi dan independen dari yang lain untuk memberikan toleransi kesalahan, stabilitas, dan ketahanan.

Untuk informasi selengkapnya, lihat [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 prinsip mana yang diizinkan mengakses, tindakan yang didukung, dan kondisi lain yang harus dipenuhi.

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

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

## kontrol responsif

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

## melestarikan

Lihat [7 Rs](#).

## pensiun

Lihat [7 Rs](#).

## 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.

### PENIPUAN

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.

### kontrol keamanan

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

## pengerasan keamanan

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

## sistem informasi keamanan dan manajemen acara (SIEM)

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

## otomatisasi respons keamanan

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

## enkripsi sisi server

Enkripsi data di tujuannya, oleh Layanan AWS yang menerimanya.

## kebijakan kontrol layanan (SCP)

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

## titik akhir layanan

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

## perjanjian tingkat layanan (SLA)

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

## indikator tingkat layanan (SLI)

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

## tujuan tingkat layanan (SLO)

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

## model tanggung jawab bersama

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

## 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.

# T

## tag

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

## variabel target

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

## daftar tugas

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

## lingkungan uji

Lihat [lingkungan](#).

## pelatihan

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

## gerbang transit

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

## alur kerja berbasis batang

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



## akses tepercaya

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

## penyetelan

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

## tim dua pizza

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

# U

## waswas

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

## tugas yang tidak terdiferensiasi

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

## lingkungan atas

Lihat [lingkungan](#).

## V

### menyedot debu

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

### kendali versi

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

### Peering VPC

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

### kerentanan

Kelemahan perangkat lunak atau perangkat keras yang membahayakan keamanan sistem.

## W

### cache hangat

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

### data hangat

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

### fungsi jendela

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

### beban kerja

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

## aliran kerja

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

## CACING

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

## WQF

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

## tulis sekali, baca banyak (WORM)

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

## Z

### eksploitasi zero-day

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

### kerentanan zero-day

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

### aplikasi zombie

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

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