Membentuk ulang COTS dan aplikasi internal selama migrasi ke Cloud AWS

AWS Panduan Preskriptif

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

AWS Panduan Preskriptif: Membentuk ulang COTS dan aplikasi internal selama migrasi ke Cloud AWS

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

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

Table of Contents

Pengantar 1
Hasil bisnis yang ditargetkan 2
Memilih lingkungan replatforming 4
Membentuk ulang komponen aplikasi yang berjalan pada tidak didukung OSs 5
Mengganti server aplikasi OSs atau yang tidak didukung 6
Upgrade OS untuk aplikasi COTS 6
Upgrade OS untuk aplikasi in-house 6
Membentuk ulang pustaka aplikasi dan perangkat lunak dependen 7
Membentuk ulang basis data backend 8
Membentuk ulang basis data backend untuk aplikasi COTS 8
Membentuk ulang basis data backend untuk aplikasi internal 9
Membentuk ulang berbagi file 10
Memperbarui komponen logging dan pemantauan 11
Menguji dan memvalidasi aplikasi Anda 12
Mengotomatiskan penambalan OS yang sedang berlangsung 13
Menggunakan alat otomatisasi dan infrastruktur sebagai kode (IAc) 13
Pertanyaan yang Sering Diajukan 14
Kapan saya harus memilih untuk memplatform ulang daripada rehosting? 14
Dapatkah saya memfaktorkan ulang aplikasi COTS saya ke database sumber terbuka? 14
Apa AWS alat yang dapat saya gunakan untuk meng-host kembali server saya dengan cepat ke AWS Awan? 14
Apa AWS alat yang dapat saya gunakan untuk mem-replatforming aplikasi saya? 14
Sumber daya 16
Referensi 16
Alat 16
Riwayat dokumen 17
..... xviii

Membentuk ulang COTS dan aplikasi internal selama migrasi ke AWS Cloud

Anbu Selvan, Amazon Web Services ()AWS

Maret 2021 (riwayat dokumen)

Panduan ini menjelaskan tujuh area yang harus Anda fokuskan saat Anda memplatform ulang aplikasi komersial off-the-shelf (COTS) dan in-house di Amazon Web Services () Cloud.AWS Panduan ini juga menyediakan strategi, alat, dan AWS layanan untuk membantu Anda memplatform ulang komponen aplikasi. Aplikasi COTS adalah aplikasi pihak ketiga yang sudah jadi dan dapat dibeli di pasar komersial (misalnya, AWS Marketplace). In-house Aplikasi dikembangkan dan digunakan secara internal oleh organisasi Anda.

Setelah Anda memutuskan untuk memigrasikan COTS atau aplikasi internal ke AWS Cloud, Anda harus mengevaluasi mana dari tujuh strategi migrasi umum (7 Rs) yang akan digunakan. Strategi-strategi ini adalah refactor, replatform, repurchase, rehost, relokasi, pertahankan, dan pensiun. Kami menyarankan Anda memplatform ulang aplikasi yang menggunakan komponen atau database yang mencapai, atau hampir mencapai, tanggal akhir dukungan (EOS) mereka. EOS adalah ketika vendor menarik dukungan teknis untuk suatu produk. Jika Anda memilih untuk memplatform ulang aplikasi di AWS Cloud, Anda bisa mendapatkan keuntungan dari kemampuan berikut:

- Otomatiskan upgrade sistem operasi (OS) di tempat dengan. AWS Systems Manager
- Gunakan volume penyimpanan snapshot untuk membuat Amazon Machine Images (AMI) dengan cepat dari instans Amazon Elastic Compute Cloud (Amazon EC2).
- Buat subnet pribadi untuk mengisolasi beban kerja yang berjalan pada sistem operasi (OS) yang sudah ketinggalan zaman.
- Gunakan jaringan berkecepatan tinggi untuk mereplikasi lingkungan produksi dengan cepat untuk menguji replatforming.
- Siapkan tumpukan aplikasi terpisah dengan cepat dengan instans EC2 sesuai permintaan, tanpa menggunakan perangkat keras lokal tambahan.

Untuk mendapatkan manfaat dari ini dan kemampuan lain yang tersedia di AWS Cloud, kami sarankan Anda terlebih dahulu meng-host ulang aplikasi Anda dengan menggunakan AWS

Transform MGN. Anda kemudian dapat memutakhirkan aplikasi di AWS Cloud. Daftar berikut memberikan contoh kapan aplikasi harus direplatform:

- Support tidak lagi tersedia untuk OS aplikasi, runtime (misalnya, Apache Tomcat, JBoss, atau Oracle WebLogic Server), database, atau komponen runtime (misalnya, Java, Python, atau Perl).
- Aplikasi harus menjadi lebih tangguh dan secara otomatis pulih dari kegagalan (misalnya, bug perangkat lunak atau masalah infrastruktur).
- Fungsionalitas aplikasi baru diperlukan untuk segmen pelanggan baru atau untuk mendukung peningkatan beban.
- Aplikasi ini tidak stabil dan membutuhkan perbaikan untuk meningkatkan stabilitas operasional.

Sebelum Anda memulai perjalanan replatforming, Anda harus mencari alternatif untuk fungsionalitas aplikasi Anda; misalnya, mengevaluasi apakah Anda dapat menggantinya dengan solusi perangkat lunak sebagai layanan (SaaS) dari vendor perangkat lunak independen (ISV). Anda mungkin juga dapat membangun kembali fungsionalitas aplikasi dengan menggunakan AWS layanan seperti AWS Lambda, Amazon Cognito, AmazonMQ, Amazon Quick AWS Glue, atau Amazon Aurora.

Panduan ini untuk administrator TI, pemilik aplikasi, arsitek, prospek teknis, dan manajer proyek. Panduan ini menyediakan tujuh area berikut untuk difokuskan saat Anda memplatform ulang COTS dan aplikasi internal di Cloud: AWS

- Memilih lingkungan replatforming
- Membentuk ulang komponen aplikasi yang berjalan pada tidak didukung OSs
- Membentuk ulang basis data backend
- Membentuk ulang berbagi file
- Memperbarui komponen logging dan pemantauan
- Menguji dan memvalidasi aplikasi Anda
- Mengotomatiskan penambalan OS yang sedang berlangsung

Hasil bisnis yang ditargetkan

Anda harus mengharapkan empat hasil berikut setelah memplatform ulang COTS dan aplikasi internal di Cloud: AWS

- Mengurangi risiko keamanan dari aplikasi lama yang menjalankan perangkat lunak atau OS yang tidak didukung.
- Turunkan biaya kepemilikan aplikasi Anda secara keseluruhan dengan menghapus edisi database yang mahal dan tidak penting atau mengadopsi basis data sumber terbuka.
- Kurangi overhead operasional dengan menggunakan database AWS terkelola (misalnya, Amazon Relational Database Service (Amazon RDS) atau Aurora) untuk mencapai tingkat ketersediaan dan keandalan yang lebih tinggi untuk aplikasi Anda.
- Jadikan aplikasi lama lebih tangguh dengan mengadopsi fitur otomatisasi dan pemantauan cloud-native, seperti pemantauan Amazon CloudWatch atau penambalan OS Sistem. Manager-based

Memilih lingkungan replatforming

Sebaiknya Anda memplatform ulang aplikasi di AWS Cloud untuk menggunakan kemampuan seperti snapshot Amazon Elastic Block Store (Amazon EBS) atau mengkloning instans EC2 untuk membuat AMI. Kemampuan ini membantu proses peningkatan dan pengujian Anda.

Biasanya, Anda mulai dengan rehosting aplikasi ke AWS Cloud dan kemudian memulai proses replatforming. Namun, Anda harus berhati-hati jika memigrasikan aplikasi yang memiliki kerentanan atau menjalankan perangkat lunak dan OS yang tidak didukung. Aplikasi ini dapat mengekspos kerentanan keamanan yang berbahaya bagi migrasi dan operasi masa depan Anda. Sebagai gantinya, kami sarankan menggunakan MGN untuk mereplikasi komponen aplikasi ke subnet pribadi dengan jalan keluar terbatas. Pendekatan ini mengisolasi beban kerja Anda, dan Anda kemudian dapat memplatform ulang dan mengujinya dengan aman sebelum diterapkan ke lingkungan produksi.

Organisasi Anda mungkin telah membeli dukungan OS yang diperpanjang dari ISV, yang memperpanjang penambalan OS melampaui tanggal EOS resmi untuk jangka waktu tiga hingga lima tahun. Ini adalah tindakan sementara dan memberikan waktu tambahan untuk memfaktorkan ulang atau menyelaraskan jadwal upgrade atau migrasi produk Anda dengan jadwal rilis produk vendor aplikasi COTS. Namun, kami merekomendasikan penggunaan AWS alat dan keahlian dari Layanan AWS Profesional atau mitra kompetensi migrasi AWS untuk memplatform ulang beban kerja ini ke OS yang lebih baru. Ini membantu menghindari kontrak lisensi diperpanjang yang mahal dan berarti Anda dapat menyelesaikan migrasi dengan lebih cepat.

Membentuk ulang komponen aplikasi yang berjalan pada tidak didukung OSs

Pendekatan replatforming untuk komponen aplikasi yang berjalan pada unsupported OSs berbeda untuk setiap komponen aplikasi. Tabel berikut merangkum opsi replatforming yang tersedia untuk komponen aplikasi yang mencapai EOS.

Komponen aplikasi	Solusi untuk aplikasi COTS	Solusi untuk aplikasi in-house
Server aplikasi	Tingkatkan ke versi yang direkomendasikan oleh vendor aplikasi.	Identifikasi versi server aplikasi terbaru. Membangun dan memvalidasi dalam lingkungan pengembangan sebelum Anda meng-upgrade.
OS	Tingkatkan ke versi yang direkomendasikan oleh vendor aplikasi.	Identifikasi versi OS terbaru. Membangun dan memvalidasi dalam lingkungan pengembangan sebelum Anda meng-upgrade.
Pustaka runtime	Tingkatkan ke versi yang direkomendasikan oleh vendor aplikasi.	Tingkatkan dan validasi terhadap versi terbaru.
Komponen aplikasi lainnya	Minta binari aplikasi baru dari vendor aplikasi.	Bangun dengan versi OS, runtime, dan server aplikasi terbaru.

Bagian berikut memberikan informasi lebih lanjut tentang pendekatan replatforming untuk komponen aplikasi Anda.

Mengganti server aplikasi OSs atau yang tidak didukung

Jika Anda mengganti server aplikasi yang tidak didukung (misalnya, Apache Tomcat 6.0, Apache 2.2, atau IIS 7.x), versi server aplikasi baru Anda mungkin memerlukan upgrade OS yang mendasarinya. Sebagian besar yang tidak didukung OSs adalah Red Hat Enterprise Linux (RHEL) versi 5 dan 6, CentOS versi 5 dan 6, atau Windows 2008 R2. Anda harus menerapkan langkah-langkah berikut untuk aplikasi yang OSs menjalankannya:

1. Luncurkan instans EC2 dengan versi OS yang diperlukan.
2. Instal versi server aplikasi yang diperlukan.
3. Ada dua pendekatan terpisah untuk aplikasi in-house dan COTS:
   - Aplikasi internal — Menerapkan ulang aplikasi ke instans EC2.
   - Aplikasi COTS — Hubungi vendor aplikasi dan minta binari aplikasi yang disertifikasi untuk versi OS atau server aplikasi yang diperlukan.

Upgrade OS untuk aplikasi COTS

Sebagian besar vendor aplikasi COTS mendukung Windows 2016 atau RHEL 7. Jika aplikasi COTS lama Anda tidak mendukung Windows 2016, kami merekomendasikan peningkatan di tempat dari Windows 2008 R2 ke Windows 2012 R2 dengan menggunakan opsi pemutakhiran di tempat yang disediakan oleh Microsoft. Anda juga dapat menggunakan runbook AWS Systems Manager Otomasi untuk secara otomatis memutakhirkan Windows Server yang berjalan pada instans EC2. Kami menyarankan Anda menghubungi vendor aplikasi dan meminta mereka untuk mengesahkan perangkat lunak mereka untuk versi OS terbaru.

Upgrade OS untuk aplikasi in-house

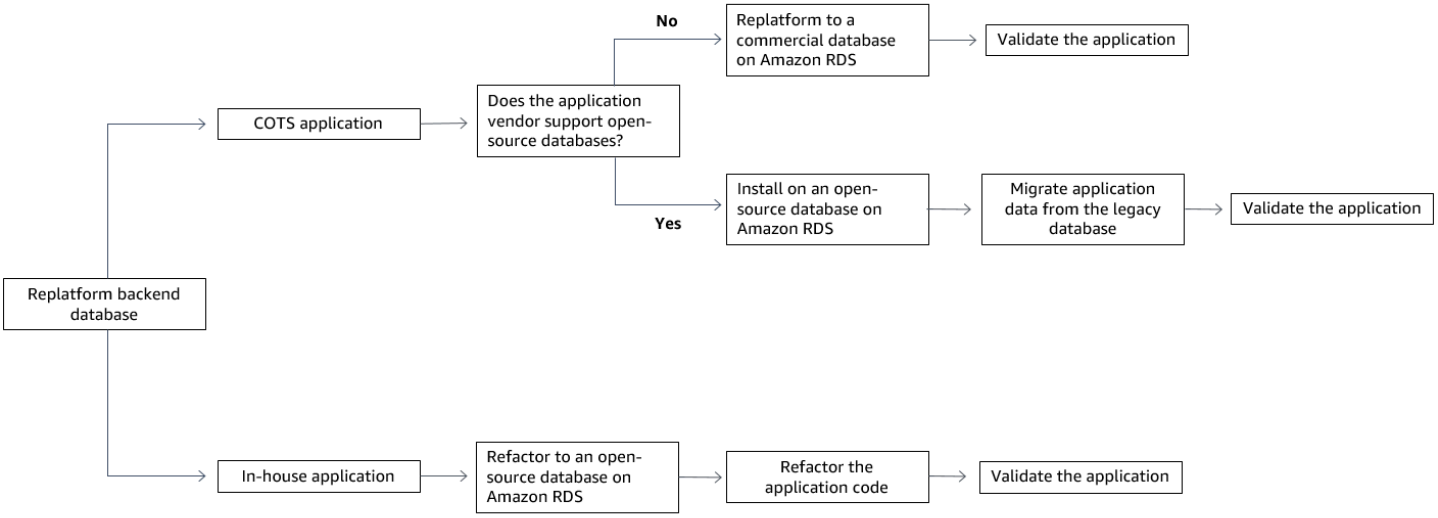
Kami menyarankan Anda mengkompilasi dan membangun kembali perangkat lunak aplikasi internal Anda dengan menggunakan versi runtime OS dan perangkat lunak terbaru (misalnya, Java, C ++, .NET, atau Python). Anda kemudian dapat mengkloning lingkungan aplikasi yang ada, menerapkan dan memvalidasi fungsionalitas secara manual, dan memperbarui lingkungan build Anda ke OS terbaru, komponen perangkat lunak runtime, dan pustaka sebelum memutakhirkan ke lingkungan produksi Anda.

Membentuk ulang pustaka aplikasi dan perangkat lunak dependen

Pendekatan untuk replatforming pustaka aplikasi dan perangkat lunak dependen mirip dengan pendekatan OSs tetapi Anda hanya meningkatkan pustaka. Anda kemudian menguji fungsionalitas aplikasi dan mereplikasi pustaka yang diperlukan di server pra-produksi dan produksi Anda. Biasanya, vendor aplikasi COTS menangani pembaruan yang diperlukan untuk komponen aplikasi melalui rilis perangkat lunak yang sedang berlangsung.

Membentuk ulang basis data backend

Pendekatan untuk replatforming database backend berbeda untuk COTS dan aplikasi in-house. Ini karena kode sumber biasanya hanya tersedia untuk aplikasi in-house. Ilustrasi berikut menunjukkan opsi replatforming yang tersedia untuk database backend aplikasi Anda.



Bagian berikut menjelaskan pendekatan replatforming untuk database backend milik COTS atau aplikasi internal.

Membentuk ulang basis data backend untuk aplikasi COTS

Kami menyarankan Anda menggunakan database Aurora jika aplikasi COTS Anda mendukung database sumber terbuka. Menggunakan database sumber terbuka membantu mengurangi biaya lisensi, dan Anda juga dapat menggunakan alat seperti AWS Schema Conversion Tool (AWS SCT) dan AWS Database Migration Service (AWS DMS) untuk mencapai cutover dengan waktu henti minimal selama migrasi.

Jika aplikasi COTS Anda tidak mendukung basis data sumber terbuka, sebaiknya replatforming ke database komersial di Amazon Relational Database Service (Amazon RDS) seperti Amazon RDS for Oracle atau Amazon RDS for Microsoft SQL Server. Anda harus mengevaluasi fitur database yang digunakan oleh aplikasi Anda dan memastikan bahwa fitur tersebut didukung di Amazon RDS sebelum memulai migrasi. Untuk informasi selengkapnya, lihat Membatasi instance database Microsoft SQL Server dalam dokumentasi Amazon RDS.

Anda juga dapat menggunakan lisensi database yang tersisa dan menjalankan database komersial yang dikelola sendiri pada instans EC2. Jika Anda memilih pendekatan ini, kami sarankan Anda memulai proses verifikasi lisensi dengan vendor database Anda. Setelah proses verifikasi lisensi selesai, Anda harus merancang solusi database yang dikelola sendiri di Amazon EC2 untuk tujuan waktu pemulihan (RTO) atau tujuan titik pemulihan (RPO) yang diperlukan aplikasi Anda.

Terakhir, kami merekomendasikan untuk membuat ulang aplikasi COTS yang sensitif terhadap keamanan dan berkinerja tinggi yang menggunakan database SQL Server ke SQL Server yang berjalan di instans Amazon EC2 Linux. Untuk informasi selengkapnya tentang hal ini, lihat Memigrasi beban kerja Windows SQL Server lokal Anda ke Amazon EC2 Linux.

Membentuk ulang basis data backend untuk aplikasi internal

Anda dapat mengurangi biaya lisensi database dan meningkatkan skalabilitas dengan memplatform database backend aplikasi internal Anda ke database AWS terkelola (misalnya, Amazon RDS for PostgreSQL, Amazon RDS for MySQL, Aurora, atau Amazon DynamoDB).

Database terkelola AWS membantu Anda mengurangi tugas administratif berulang untuk database Anda (misalnya, melakukan pencadangan atau menambal database dan OS). Jika Anda menggunakan Multi-AZ penerapan Amazon RDS, Anda juga dapat meningkatkan ketersediaan aplikasi dengan mencegah pemadaman dari kegagalan perangkat keras database. Multi-AZ database terus direplikasi ke Availability Zone yang berbeda dan aplikasi secara transparan gagal ke database yang direplikasi selama pemadaman.

Anda dapat menggunakan AWS DMS dan AWS SCT mengonversi basis data komersial ke Aurora dan Amazon RDS. AWS SCT mengotomatiskan proses konversi skema database, dan AWS DMS memungkinkan replikasi data dari database lokal ke Amazon RDS. AWS DMS juga membantu mencapai penghentian waktu henti minimal saat Anda memigrasikan aplikasi lokal ke Cloud. AWS

Membentuk ulang berbagi file

Jika aplikasi internal Anda menggunakan berbagi file eksternal lokal, seperti Network File Systems (NFS) atau Server Message Block (SMB), sebaiknya ubah kode aplikasi untuk menggunakan Amazon Simple Storage Service (Amazon S3) dan Amazon. CloudFront Namun, jika akses berbasis blok diperlukan, Anda dapat menggunakan Amazon Elastic File System (Amazon EFS) atau Amazon FSx untuk Lustre. Amazon EFS dan Amazon FSx adalah AWS opsi untuk aplikasi yang memerlukan penyimpanan file bersama.

Untuk aplikasi COTS, sebaiknya migrasi data dari berbagi file lokal ke Amazon EFS di Linux atau FSx untuk Windows File Server. Anda harus mengekspos berbagi file berbasis blok ke aplikasi dengan cara yang sama seperti berbagi file lokal diekspos. Amazon EFS dan Amazon FSx menyediakan berbagi file berbasis NFS dan SMB dengan fungsionalitas yang sama dengan berbagi file lokal yang disediakan. Untuk menggunakan Amazon EFS atau Amazon FSx untuk berbagi file, hanya perubahan konfigurasi yang diperlukan untuk aplikasi COTS.

Memperbarui komponen logging dan pemantauan

Beberapa lingkungan lama menggunakan alat pencatatan dan pemantauan terpusat (misalnya, Splunk, SolarWinds, atau Zabbix) untuk pemantauan infrastruktur dan aplikasi. Tim pendukung aplikasi mungkin juga menggunakan protokol Secure Shell (SSH) atau protokol desktop jarak jauh (RDP) untuk memantau dan men-debug. Untuk menghindari proses manual dan berulang ini, Anda dapat menggunakannya CloudWatchuntuk mengotomatiskan pemantauan di Cloud. AWS

Sebaiknya gunakan CloudWatch metrik untuk memantau infrastruktur dan CloudWatch agen Anda untuk mengirim log aplikasi. CloudWatch Setelah log aplikasi diterima oleh CloudWatch, Anda dapat membuat filter CloudWatch metrik dan menggunakan CloudWatch alarm untuk memantau kesalahan aplikasi dan secara otomatis memberi tahu tim dukungan.

CloudWatch juga menyediakan alat untuk membangun dasbor operasional untuk tinjauan berkelanjutan operasi produksi untuk aplikasi Anda. Alat pemantauan terpusat pihak ketiga dapat diintegrasikan dengan CloudWatch dan AWS layanan lainnya, dan ini membantu memperluas praktik operasional Anda yang ada ke infrastruktur dan aplikasi di AWS Cloud. Namun, Anda mungkin harus melatih tim dukungan atau operasi Anda jika Anda memilih untuk mengoperasikan aplikasi di AWS Cloud. Untuk informasi selengkapnya tentang hal ini, lihat bagian Perspektif operasi: Kelola dan skala pada whitepaper AWS Cloud Adoption Framework (AWS CAF).

Menguji dan memvalidasi aplikasi Anda

Pengujian fungsional dan kinerja adalah bagian penting dari perjalanan replatforming aplikasi. Biasanya, aplikasi lama bergantung pada pengetahuan pemilik aplikasi untuk pengujian karena detail fungsional tidak didokumentasikan dengan benar atau sepenuhnya. Namun, kami menyarankan Anda merekam kasus penggunaan aplikasi dengan menggunakan pengujian perilaku dan otomatis. Pendekatan ini dengan cepat dan andal memvalidasi fungsionalitas aplikasi sebelum dan sesudah replatforming. Anda dapat menggunakan alat pengujian otomatis (misalnya, Selenium, Tricentis, atau Gatling) untuk membangun tes fungsional dan kinerja. Hasil dasar harus dihasilkan dengan menjalankan pengujian fungsional dan kinerja pada lingkungan aplikasi Anda saat ini. Hasil pengujian antara lingkungan aplikasi saat ini dan target dapat dibandingkan dan digunakan sebagai kriteria penerimaan.

Kami merekomendasikan penggunaan pengujian kenari untuk aplikasi yang menghadap pelanggan. Pengujian Canary secara berkala menguji alur kerja aplikasi penting di lingkungan produksi dan memberi tahu tim dukungan tentang kesalahan. Untuk informasi selengkapnya, lihat bagian penyebaran Canary dari AWS Well-Architected Framework.

Mengotomatiskan penambalan OS yang sedang berlangsung

Aplikasi lama di pusat data lokal sering mengandalkan proses operasional manual untuk penambalan OS yang sedang berlangsung dan pembaruan perangkat lunak. Selama perjalanan replatforming Anda, kami menyarankan Anda mengotomatiskan patching OS dengan menggunakan Systems Manager Patch Manager atau proses patching otomatis lainnya. Patch Manager menyediakan proses yang terpusat dan konsisten untuk mengumpulkan wawasan operasional dan mengimplementasikan tugas operasional rutin di AWS Cloud dan sumber daya lokal.

Kami merekomendasikan lingkungan pengembangan tambalan lebih awal dari jendela waktu penambalan yang digunakan untuk lingkungan produksi. Untuk informasi selengkapnya tentang hal ini, lihat runbook Patch Manager untuk mengotomatisasi patch OS. Anda juga harus menerapkan pengujian kenari untuk menguji fungsionalitas aplikasi kunci secara berkala di lingkungan pra-produksi atau produksi, dan memperingatkan tim dukungan jika pengujian gagal. Ini membantu menghindari pemadaman yang tidak direncanakan untuk aplikasi Anda.

Menggunakan alat otomatisasi dan infrastruktur sebagai kode (IAc)

Sebagai bagian dari perjalanan replatforming aplikasi Anda, Anda harus mengotomatiskan build platform dengan menggunakan alat manajemen konfigurasi seperti Chef, Puppet, atau Ansible. Alat-alat ini memungkinkan pembuatan tumpukan aplikasi yang dapat diulang dan memformalkan langkah-langkah untuk menghasilkan instance aplikasi, termasuk konfigurasi tumpukan.

Kami menyarankan Anda menyediakan infrastruktur Anda dengan menggunakan praktik terbaik IAc. Ada beberapa opsi yang tersedia untuk ini, termasuk AWS Cloud Development Kit (AWS CDK), CloudFormationdan Terraform. Chef, Ansible, dan Puppet juga memiliki kemampuan terbatas yang mungkin memberikan otomatisasi yang cukup untuk kasus penggunaan Anda.

Build berulang yang menggunakan IAc dan kode manajemen konfigurasi membantu Anda menguji infrastruktur tanpa overhead dan risiko membangun kembali sumber daya tersebut. Menambal dan memperbarui instance yang ada dapat menyebabkan status yang membuatnya sulit untuk mereproduksi dan mengidentifikasi masalah.

Jika aplikasi COTS tidak mendukung penginstalan otomatis, sebaiknya konsultasikan dengan AWS Partner Network (APN). Untuk informasi selengkapnya tentang hal ini, lihat bagian Perspektif Platform: Aplikasi dan infrastruktur pada AWS whitepaper CAF.

Membentuk ulang COTS dan aplikasi in-house FAQ

Bagian ini memberikan jawaban atas pertanyaan yang sering diajukan tentang replatforming COTS dan aplikasi internal.

Kapan saya harus memilih untuk memplatform ulang daripada rehosting?

Anda harus memplatform ulang komponen non-database jika vendor OS telah menghentikan pembaruan keamanan (misalnya, RHEL 5 dan Windows 2008). Untuk OS yang tidak didukung, Anda dapat meng-host ulang komponen ke subnet pribadi yang terisolasi dan memutakhirkannya sebelum Anda menerapkannya ke lingkungan produksi.

Untuk komponen database, Anda harus mengevaluasi apakah replatforming ke database komersial di Amazon RDS adalah pilihan yang tepat. Anda juga dapat menggunakan database yang dikelola sendiri yang berjalan di Amazon EC2 jika vendor database memungkinkan Anda untuk menggunakan lisensi Anda saat ini di AWS Cloud.

Dapatkah saya memfaktorkan ulang aplikasi COTS saya ke database sumber terbuka?

Dimungkinkan untuk memfaktorkan ulang aplikasi COTS ke database sumber terbuka jika vendor aplikasi Anda mendukung basis data sumber terbuka. Kami tidak menyarankan menggunakan database sumber terbuka untuk aplikasi COTS tanpa mendapatkan dukungan vendor.

Apa AWS alat yang dapat saya gunakan untuk meng-host kembali server saya dengan cepat ke AWS Awan?

Anda dapat menggunakan MGN sebagai layanan migrasi utama Anda untuk mengangkat dan memindahkan aplikasi Anda ke AWS.

Apa AWS alat yang dapat saya gunakan untuk mem-replatforming aplikasi saya?

Ada beberapa alat yang tersedia di AWS Cloud termasuk:

- Systems Manager upgrade otomatis — Upgrade Windows 2008 dan SQL Server 2008 di Cloud. AWS
- AWS SCT— Mengkonversi skema database dari komersial ke database open-source.
- AWS DMS— Replikasi data dari database lokal ke AWS Cloud, baik di database Amazon RDS atau database yang berjalan di Amazon EC2.

Sumber daya

Referensi

- AWS Cloud Adoption Framework dari AWS whitepaper migrasi
- Buat CloudWatch dasbor
- Batas untuk instans Microsoft SQL Server DB
- Memigrasikan database Microsoft SQL Server yang dikelola sendiri ke database yang dikelola sepenuhnya di Amazon RDS
- AWS lokakarya: Instans cluster failover Microsoft SQL Server di Amazon FSx
- Tingkatkan OS instans Amazon EC2 Windows Server

Alat

- AWS CodeBuilddan AWS CodePipelinembantu Anda membangun integrasi berkelanjutan dan pipeline pengembangan berkelanjutan (CI/CD) di AWS Cloud.
- AWS DataSync— DataSync membantu mereplikasi penyimpanan file lokal ke Cloud. AWS
- AWS DMS— AWS Database Migration Service (AWS DMS) membantu mereplikasi data dari database lokal ke database Amazon RDS atau database di Amazon EC2.
- AWS Systems Manager Pemutakhiran otomatis berbasis membantu meningkatkan Windows 2008 dan SQL Server 2008 di Cloud. AWS
- AWS SCT— AWS Schema Conversion Tool membantu mengonversi skema basis data dari basis data komersial ke sumber terbuka.
- MGN — AWS Transform MGN adalah solusi lift-and-shift (rehost) yang sangat otomatis yang menyederhanakan, mempercepat, dan mengurangi biaya migrasi aplikasi. AWS
- Asisten replatforming Windows to Linux untuk database Microsoft SQL Server adalah alat replatforming.

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
Nama layanan AWS yang diperbarui	Update CloudEndure Migrasi ke Layanan Migrasi Aplikasi	12 Mei 2022
—	Publikasi awal	12 Maret 2021

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