



Pendekatan bertahap untuk memodernisasi aplikasi di Cloud AWS

AWS Panduan Preskriptif



AWS Panduan Preskriptif: Pendekatan bertahap untuk memodernisasi aplikasi di 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
Proses modernisasi	3
Langkah 1. Evaluasi aplikasi Anda	3
Menggunakan buku pedoman diagnostik modernisasi	4
Mengidentifikasi metrik	6
Langkah 2. Mulai dari yang kecil dan bangun momentum	7
Memvalidasi driver prioritas	7
Menyelesaikan detail	8
Membangun layanan platform dasar dan memodernisasi aplikasi	8
Langkah 3. Buat peta jalan modernisasi yang dapat diskalakan	9
Faktor kesiapan modernisasi	11
Kode	11
Bangun dan uji	11
Rilis	12
Jalankan	12
Optimalkan	13
Kesiapan	13
Langkah selanjutnya	14
Sumber daya	15
Panduan terkait	15
AWS sumber daya	15
Riwayat dokumen	16
.....	xvii

Pendekatan bertahap untuk memodernisasi aplikasi di Cloud AWS

Vijay Thumma dan Ashish Ameta, Amazon Web Services (AWS)

Mei 2023 ([riwayat dokumen](#))

Modernisasi membutuhkan pendekatan multi-dimensi untuk mengadopsi dan mengonsumsi teknologi baru, untuk memberikan portofolio, aplikasi, dan nilai infrastruktur lebih cepat, dan untuk memposisikan organisasi untuk penskalaan dengan harga yang optimal. Ini melibatkan mengoptimalkan, memelihara aplikasi, dan beroperasi dalam model modern tanpa gangguan, dan mengharuskan Anda menyederhanakan operasi bisnis, arsitektur, dan praktik rekayasa secara keseluruhan.

Modernisasi bukan hanya tentang aplikasi; itu membutuhkan infrastruktur modern yang menyediakan kerangka kerja operasi yang aman dan fleksibel. Aplikasi dan infrastruktur tidak dapat dipisahkan dalam hal kualitas, ketersediaan, dan kelincahan proses bisnis. Modernisasi aplikasi tanpa memperhitungkan infrastruktur menyebabkan biaya keseluruhan yang tinggi dan berdampak negatif pada kinerja dan kualitas. Aplikasi modern dibangun dengan kombinasi pola arsitektur baru, model operasional, dan proses pengiriman perangkat lunak. Mereka meningkatkan dan menurunkan dari nol hingga jutaan pengguna, mengelola terabyte (jika bukan petabyte) data, tersedia secara global, dan merespons dalam milidetik. Saat memodernisasi portofolio beban kerja yang Anda kelola di Amazon Web Services (AWS) Cloud, Anda memplatform ulang, memfaktorkan ulang, atau mengganti beban kerja ini dengan menggunakan container, teknologi tanpa server, penyimpanan data yang dibuat khusus, dan otomatisasi perangkat lunak, untuk mendapatkan manfaat kelincahan dan pengoptimalan biaya total (TCO) sepenuhnya yang ditawarkan oleh AWS.

Panduan ini untuk pemilik aplikasi, pemilik bisnis, arsitek, prospek teknis, dan manajer proyek. Ini membahas bagaimana mengembangkan kemampuan dasar untuk aplikasi yang dipilih dalam fase penilaian modernisasi, dan cara untuk mempercepat upaya modernisasi dengan menggunakan pendekatan bertahap.

Panduan ini merupakan bagian dari seri konten yang mencakup pendekatan modernisasi aplikasi yang direkomendasikan oleh AWS. Serial ini juga mencakup:

- [Strategi untuk memodernisasi aplikasi di Cloud AWS](#)
- [Mengevaluasi kesiapan modernisasi untuk aplikasi di Cloud AWS](#)

- [Mengurai monolit menjadi layanan mikro](#)
- [Mengintegrasikan layanan mikro dengan menggunakan layanan tanpa server AWS](#)

Hasil bisnis yang ditargetkan

Anda harus mengharapkan hasil berikut dari pendekatan bertahap untuk modernisasi aplikasi:

- Kapasitas dan kemampuan organisasi untuk berinovasi lebih cepat, dengan menggunakan build-and-prove pendekatan dan arsitektur cloud-native seperti microservices.
- Manajemen perubahan dan model operasional yang membangun kesiapan organisasi melalui pelatihan dan peningkatan alat.
- Pendekatan tim, yang membantu memberikan hasil awal hanya dalam 12 minggu, memberikan pembelajaran berdasarkan pengalaman, dan memungkinkan kesuksesan pelanggan yang mandiri dan langgeng.
- Arsitektur aplikasi yang dapat dikomposisi berdasarkan layanan mikro, komponen yang dapat digunakan kembali APIs, dan kontainerisasi.
- Peta jalan modernisasi yang dapat diskalakan untuk aplikasi strategis tertentu, yang mencakup panduan preskriptif untuk bekerja dalam suatu model. split-and-lead Dalam model ini, kemampuan dan layanan modernisasi diskalakan di beberapa tim teknik yang berfokus pada hasil bisnis. Karena produk baru yang layak minimum (MVPs) didefinisikan, anggota tim awal berpisah untuk membuat tim produk baru.

Proses modernisasi

Tujuan dari pendekatan bertahap untuk modernisasi adalah untuk memberikan nilai tambahan dengan menggunakan dimensi modernisasi, untuk menerapkannya pada subset inti, aplikasi pembeda bisnis, dan untuk mempercepat adopsi teknologi modern. Pendekatan bertahap terdiri dari tiga langkah:

1. Evaluasi kematangan aplikasi dengan menggunakan buku pedoman diagnostik modernisasi. Ambil pendekatan komprehensif untuk adopsi, dan kembangkan proses yang selaras dengan hasil bisnis yang diinginkan.
2. Mulailah dari yang kecil dan bersiaplah untuk memberikan MVP untuk mendapatkan momentum dengan mendorong hasil bisnis lebih awal dan bertahap. Fase ini biasanya memakan waktu 12 hingga 16 minggu.
3. Buat peta jalan modernisasi yang dapat diskalakan untuk bekerja dalam model split-and-seed.

Bagian berikut membahas langkah-langkah ini secara lebih rinci.

Topik

- [Langkah 1. Evaluasi aplikasi Anda](#)
- [Langkah 2. Mulai dari yang kecil dan bangun momentum](#)
- [Langkah 3. Buat peta jalan modernisasi yang dapat diskalakan](#)

Langkah 1. Evaluasi aplikasi Anda

Tujuan dari fase ini adalah untuk:

- Pahami lanskap aplikasi Anda secara menyeluruh dan persiapkan aplikasi Anda untuk platform data modern, sehingga Anda dapat mempercepat waktu untuk menilai tanpa memengaruhi bisnis Anda, dan kemudian memodernisasi, mengoptimalkan, dan menskalakan.
- Buat profil lanskap aplikasi Anda untuk mengidentifikasi manfaat, risiko, dan biaya yang terkait dengan perubahan.
- Menyediakan serangkaian layanan end-to-end: mulai dari strategi dan perencanaan; melalui penyebaran, migrasi, dan modernisasi aplikasi; hingga dukungan berkelanjutan.
- Bangun kebijakan, rekomendasi, dan kontrol yang menyediakan praktik dan alat yang dapat digunakan kembali untuk memberikan nilai bisnis yang berkelanjutan.

Pada tahap evaluasi, pemilik aplikasi dan arsitek menggunakan buku pedoman diagnostik modernisasi untuk memvalidasi tujuan dan prioritas modernisasi mereka.

Menggunakan buku pedoman diagnostik modernisasi

Buku pedoman diagnostik modernisasi menyediakan proses untuk menentukan nilai perpindahan dari keadaan saat ini ke keadaan future untuk perusahaan. Ini termasuk perubahan teknologi yang melibatkan modernisasi.

Anda menggunakan buku pedoman diagnostik untuk menentukan prioritas aplikasi atau rangkaian aplikasi Anda untuk modernisasi cloud, dan untuk mengidentifikasi komponen yang perlu ditangani selama modernisasi.

Dimensi diagnostik

Buku pedoman diagnostik modernisasi membantu Anda memahami dimensi berikut dari keadaan saat ini dan target (pasca-migrasi) aplikasi atau sekelompok aplikasi:

- Pengelompokan aplikasi — Apakah ada alasan untuk mengelompokkan aplikasi (misalnya, berdasarkan teknologi atau model operasi) untuk modernisasi?
- Sequencing — Apakah ada urutan di mana aplikasi harus dimodernisasi, berdasarkan dependensi?
- Teknologi — Apa saja kategori teknologi (misalnya, middleware, database, messaging)?
- Dependensi — Apakah aplikasi memiliki dependensi kunci pada sistem lain atau middleware?
- Lingkungan — Berapa banyak lingkungan pengembangan, pengujian, dan produksi yang digunakan?
- Penyimpanan — Apa persyaratan penyimpanan (misalnya, jumlah salinan data uji)?
- Model operasi — Dapatkah semua komponen aplikasi mengadopsi integrasi berkelanjutan dan pipa pengiriman berkelanjutan (CI/CD)?
 - Jika demikian, tanggung jawab infrastruktur apa yang harus didistribusikan ke tim aplikasi dan kepada siapa?
 - Jika tidak, tanggung jawab infrastruktur apa (misalnya, penambalan) yang harus tetap ada pada tim operasi?
- Model pengiriman:
 - Berdasarkan aplikasi atau kelompok aplikasi, haruskah Anda memplatform ulang, memfaktorkan ulang, menulis ulang, atau mengganti?

- Bagian modernisasi mana yang harus menggunakan layanan cloud-native?
- Skill set — Keahlian apa yang dibutuhkan? Contoh:
 - Latar belakang aplikasi cloud untuk membangun aplikasi dengan arsitektur modular dengan menggunakan teknologi kontainer dan tanpa server dari bawah ke atas.
 - DevOps keahlian untuk mengembangkan solusi di bidang CI/CD proses, infrastruktur sebagai kode, dan otomatisasi atau observabilitas aplikasi dengan menggunakan sumber terbuka dan AWS alat dan layanan.
- Pendekatan modernisasi — Mempertimbangkan keadaan aplikasi saat ini, pilihan teknologi cloud, utang teknis saat ini CI/CD, pemantauan, keterampilan, dan model operasi, apa pekerjaan migrasi teknis yang perlu dilakukan?
- Waktu modernisasi — Apa pertimbangan waktu portofolio bisnis atau pertimbangan kerja terencana lainnya yang dapat memengaruhi waktu modernisasi?
- Unit dan total biaya infrastruktur — Berapa biaya tahunan untuk mempertahankan beban kerja Anda di tempat vs. pada AWS, berdasarkan analisis ekonomi?

Mengevaluasi aplikasi terhadap dimensi ini membantu Anda tetap berlabuh dalam bisnis, teknologi, dan ekonomi saat Anda mendorong modernisasi Anda ke cloud.

Blok bangunan

Ketika Anda memodernisasi aplikasi, Anda dapat mengklasifikasikan pengamatan Anda menjadi tiga blok bangunan: kelincahan bisnis, kelincahan organisasi, dan efektivitas teknik.

- Kelincahan bisnis — Praktik yang menyangkut efektivitas dalam bisnis untuk menerjemahkan kebutuhan bisnis ke dalam persyaratan. Seberapa responsif organisasi pengiriman terhadap permintaan bisnis, dan seberapa besar kontrol yang dimiliki bisnis dalam melepaskan fungsionalitas ke lingkungan produksi.
- Kelincahan organisasi — Praktik yang mendefinisikan proses pengiriman. Contohnya termasuk metodologi dan DevOps upacara tangkas serta penugasan peran dan kejelasan, dan kolaborasi, komunikasi, dan pemberdayaan secara keseluruhan di seluruh organisasi.
- Efektivitas teknik — Praktik pengembangan yang terkait dengan jaminan kualitas, pengujian CI/CD, manajemen konfigurasi, desain aplikasi, dan manajemen kode sumber.

Mengidentifikasi metrik

Untuk mengetahui apakah Anda memberikan apa yang penting bagi pelanggan Anda, Anda harus menerapkan langkah-langkah yang mendorong peningkatan dan mempercepat pengiriman. Sasaran, pertanyaan, metrik (GQM) menyediakan kerangka kerja yang efektif untuk memastikan bahwa ukuran Anda memenuhi kriteria ini. Gunakan kerangka kerja ini untuk bekerja kembali dari tujuan Anda dengan mengikuti langkah-langkah berikut:

1. Identifikasi tujuan atau hasil yang Anda lakukan.
2. Turunkan pertanyaan yang harus dijawab untuk menentukan apakah tujuan tercapai.
3. Putuskan apa yang harus atau dapat diukur untuk menjawab pertanyaan secara memadai. Ada dua kategori ukuran:
 - Metrik produk, yang memastikan bahwa Anda memberikan apa yang penting bagi pelanggan Anda.
 - Metrik operasional, yang memastikan bahwa Anda meningkatkan siklus hidup pengiriman perangkat lunak Anda.

Metrik produk

Metrik produk berfokus pada hasil bisnis dan harus ditetapkan ketika laba atas investasi (ROI) untuk ruang lingkup pekerjaan baru ditentukan. Teknik yang berguna untuk menetapkan metrik produk adalah dengan menanyakan apa yang akan berubah dalam bisnis ketika ruang lingkup pekerjaan baru itu diterapkan. Sangat membantu untuk memformalkan pemikiran ini ke dalam bentuk tes yang berfokus pada apa yang benar ketika fitur modernisasi disampaikan.

Misalnya, jika Anda yakin bahwa memigrasikan transaksi dari sistem lama akan membuka peluang baru bagi klien onboard, apa peningkatannya? Berapa banyak kapasitas yang harus dibuat untuk bergabung dengan klien berikutnya? Bagaimana tes dibangun untuk memvalidasi hasil itu? Untuk skenario ini, metrik produk Anda mungkin termasuk yang berikut:

- Identifikasi uji nilai bisnis atau hipotesis (misalnya, membebaskan x persen dari kapasitas transaksi akan masuk y persen dari bisnis baru).
- Tetapkan baseline (misalnya, kapasitas transaksi x saat ini mendukung pelanggan y).
- Validasi hasilnya (misalnya, Anda telah meningkatkan kapasitas sebesar x persen, jadi dapatkah Anda sekarang bergabung dengan y persen bisnis baru?)

Metrik operasional

Untuk menentukan apakah Anda meningkatkan siklus hidup pengiriman perangkat lunak dan mempercepat modernisasi Anda, Anda harus mengetahui waktu tunggu dan waktu implementasi Anda untuk mengirimkan perangkat lunak. Artinya, seberapa cepat Anda dapat mengubah kebutuhan bisnis menjadi fungsionalitas dalam produksi?

Metrik operasional yang berguna meliputi:

- **Lead time** — Berapa lama waktu yang dibutuhkan untuk lingkup pekerjaan untuk beralih dari permintaan ke produksi?
- **Waktu siklus** — Berapa lama waktu yang dibutuhkan untuk mengimplementasikan ruang lingkup pekerjaan, dari awal hingga akhir?
- **Frekuensi penyebaran** — Seberapa sering Anda menerapkan perubahan pada produksi?
- **Waktu untuk memulihkan layanan** - Berapa lama waktu yang dibutuhkan untuk pulih dari kegagalan (diukur sebagai waktu rata-rata untuk memperbaiki atau MTTR)?
- **Perubahan tingkat kegagalan** — Berapa waktu rata-rata antara kegagalan (MTBF)?

Langkah 2. Mulai dari yang kecil dan bangun momentum

Tujuan dari langkah ini adalah untuk memberikan produk minimal layak awal (MVP) untuk mendapatkan momentum. Pendekatan ini memungkinkan Anda untuk mendorong hasil bisnis lebih awal dan bertahap.

Memvalidasi driver prioritas

Sebelum Anda memulai pekerjaan modernisasi dengan tim aplikasi, kami sarankan Anda memvalidasi driver prioritas yang Anda tentukan sebelumnya. Ikuti langkah-langkah ini:

1. **Kompilasi informasi yang Anda butuhkan dari buku pedoman diagnostik.**
 - Kumpulkan driver prioritas dan penilaian kelayakan dari daftar aplikasi prioritas.
 - Kumpulkan disposisi transisi dan status sasaran untuk aplikasi Anda.
 - Identifikasi pemilik aplikasi, arsitek, dan pemangku kepentingan dalam perencanaan modernisasi cloud.
 - Minta informasi tentang dependensi atau pengurutan suite aplikasi, jika diketahui.

- Tentukan bagaimana entri inventaris berhubungan dengan dependensi atau pengelompokan suite aplikasi. Aplikasi mungkin memiliki komponen individual yang digabungkan erat dengan, atau bergantung pada, komponen lain, dan Anda mungkin ingin memodernisasi komponen ini bersama-sama.
2. Jadwalkan pertemuan satu jam atau dua jam dengan orang-orang dari langkah 1 untuk memvalidasi driver prioritas.
 - Cobalah untuk mengelompokkan beberapa (hingga tiga atau empat) aplikasi oleh insinyur solusi atau arsitek, dan diskusikan dalam satu pertemuan, berdasarkan ketergantungan aplikasi atau informasi suite aplikasi.
 - Tentukan peran dan harapan untuk setiap anggota tim untuk pertemuan mendatang ini.
 3. Lakukan pertemuan.

Menyelesaikan detail

Setelah Anda mengikuti proses di bagian sebelumnya untuk memvalidasi driver prioritas, Anda dapat mengumpulkan detail untuk menentukan pendekatan dan waktu modernisasi.

Pada fase ini, tim inti bekerja berdampingan dengan tim aplikasi dalam sprint singkat dua hari untuk merancang status future untuk aplikasi mereka di Cloud. AWS Kegiatan meliputi definisi produk, penemuan produk, penulisan cerita, pemetaan aliran nilai, dan CI/CD proses perancangan. Berikut adalah beberapa ide:

- Model masing-masing komponen aplikasi (misalnya, konfigurasi jaringan, konfigurasi penyimpanan, database, server, dan bagaimana aplikasi digunakan di server).
- Dekonstruksi model itu ke dalam blok bangunan dan konfigurasi yang berbeda dengan menggunakan alat seperti wadah atau teknologi tanpa server.
- Pisahkan fungsionalitas aplikasi dari dependensi apa pun pada infrastruktur yang mendasarinya. Abstrak fungsi aplikasi menjadi komponen yang dapat Anda pindahkan tanpa mengubah kode sumber apa pun.
- Integrasikan DevOps dengan erat dengan menggunakan CI/CD alat dan mekanisme.

Membangun layanan platform dasar dan memodernisasi aplikasi

Dalam fase 12 minggu ini, tim inti didukung oleh tim full-stack untuk memberikan kasus penggunaan bisnis yang diprioritaskan. Pekerjaan ini dilakukan oleh beberapa tim dua pizza. Misalnya, tim

rekayasa platform dibentuk untuk mengembangkan layanan platform dasar, dan tim produk dibentuk untuk memberikan hasil bisnis baru:

- Tim rekayasa platform mengonfigurasi, mengintegrasikan, dan menyesuaikan AWS layanan yang mendukung fondasi cloud, alur kerja pengembang, dan kemampuan analitik data. Perusahaan yang lebih besar dan lebih kompleks mungkin memiliki banyak tim yang mendukung masing-masing kemampuan ini.
- Tim produk mengembangkan layanan dan pengalaman baru untuk hasil bisnis yang diprioritaskan pada tahap awal. Ketika tim produk mengembangkan layanan baru, mereka juga memodernisasi kemampuan bisnis inti.

Tim rekayasa platform dan produk memberikan produk minimal yang layak (MVP) yang dapat Anda evaluasi. Setelah keberhasilan MVP awal, Anda dapat menskalakan program modernisasi Anda dengan menggunakan pendekatan split-and-seed, di mana aplikasi baru diidentifikasi dan anggota tim awal dibagi untuk membuat tim produk baru.

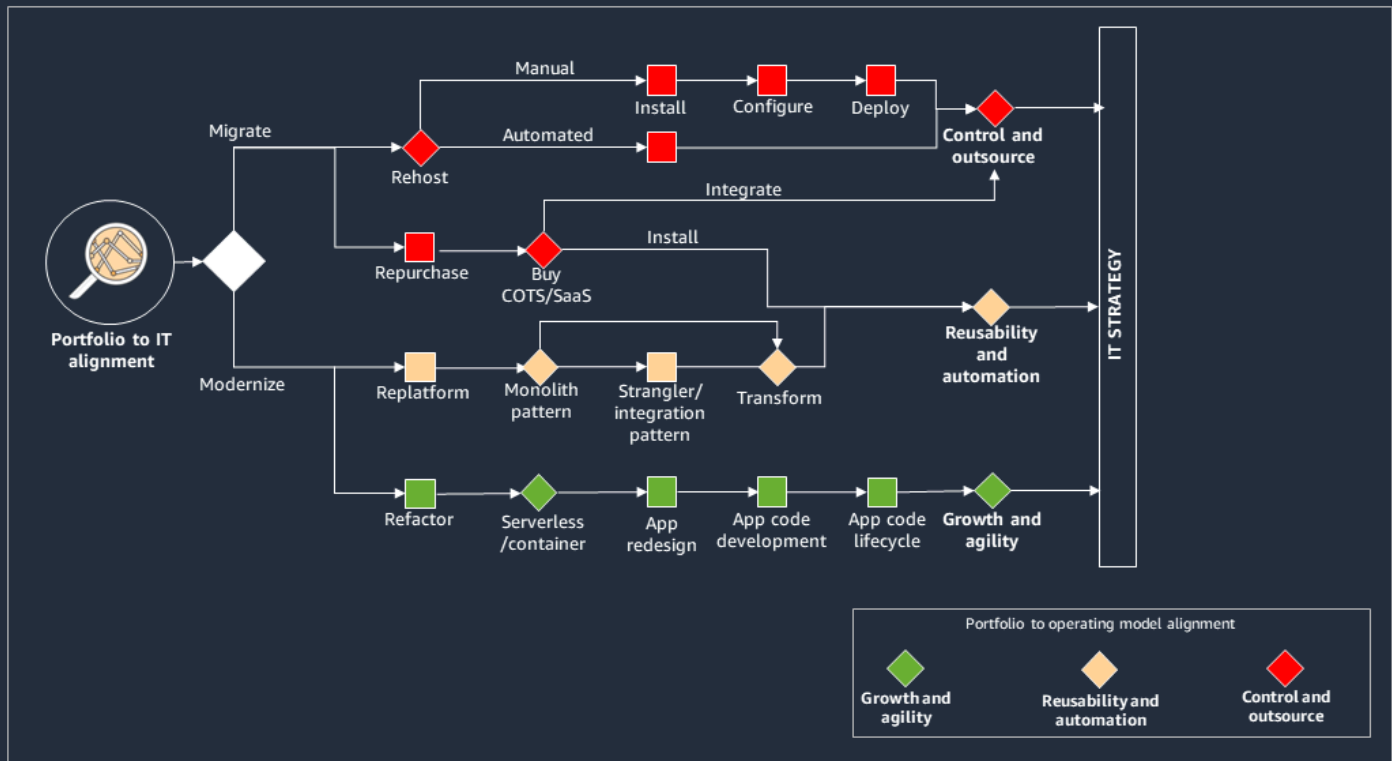
Langkah 3. Buat peta jalan modernisasi yang dapat diskalakan

Setelah rilis MVP awal dari hasil dan aplikasi yang diprioritaskan, kami menyarankan Anda mengembangkan peta jalan untuk meningkatkan dan mempercepat upaya modernisasi Anda, meningkatkan produktivitas pengembang, dan berinovasi dengan cepat. Tim inti membagi dan menyemai tim baru untuk meningkatkan kemampuan dan layanan organisasi Anda di beberapa tim teknik yang berfokus pada hasil bisnis. Dengan menggunakan pendekatan split-and-seed dari waktu ke waktu, organisasi Anda dapat mengambil lebih banyak pengembangan dan mempercepat kecepatan modernisasi.

Peta jalan modernisasi harus menguraikan pendekatan pragmatis dan berkelanjutan untuk modernisasi aplikasi dengan pola yang jelas seperti event-driven, strangler, desain berbasis domain, dekomposisi, opsi database modern, dan sebagainya.

Peta jalan harus mencakup matriks pohon keputusan, seperti yang ditunjukkan pada diagram berikut, untuk mengidentifikasi komponen aplikasi dan memindahkannya ke layanan terkelola (seperti layanan basis data) tanpa perubahan pada logika bisnis, atau untuk membuat perubahan tingkat kode untuk meningkatkan kinerja, skalabilitas, pengelolaan, keandalan, dan penggunaan sumber daya.

Migration and modernization decisions



Faktor kesiapan modernisasi

Perhatikan standar dan praktik terbaik berikut saat Anda memodernisasi aplikasi Anda.

Topik

- [Kode](#)
- [Bangun dan uji](#)
- [Rilis](#)
- [Jalankan](#)
- [Optimalkan](#)
- [Kesiapan](#)

Kode

- Berikan komentar kode yang mendokumentasikan fungsionalitas perangkat lunak Anda, dan gunakan untuk menghasilkan dokumentasi.
- Ikuti manajemen kode dan proses penerapan yang mendukung check-in kode yang sering dan keterlacakan ke permintaan fitur.
- Buat rangkaian pengujian yang mencakup pengujian unit, fungsional, kinerja, dan jalur kritis, dengan cakupan kode 100 persen.
- Dorong penggunaan kembali kode untuk memberikan fungsionalitas yang sama atau serupa di basis kode Anda.
- Kembangkan prototipe untuk memvalidasi fitur dengan pengguna sebelum berinvestasi dalam pengembangan kode penuh.

Bangun dan uji

- Mendefinisikan ulang kelengkapan fitur berdasarkan pengujian, untuk meningkatkan kualitas dan mencegah masalah berulang.
- Otomatiskan tes penerimaan.
- Pantau semua pengujian otomatis, dan buat proses untuk menangani kegagalan di tempat.

- Lacak kinerja di lingkungan produksi dan non-produksi, tentukan tujuan tingkat layanan (SLOs) berdasarkan pengujian lalu lintas dan beban yang realistis, dan berikan kemampuan untuk menskalakan untuk memenuhi persyaratan kinerja.
- Abstrak data sensitif dari file konfigurasi, dan menyediakan alat yang mengotomatiskan dan memantau konfigurasi.

Rilis

- Otomatiskan penerapan dengan dukungan untuk dependensi (misalnya, rilis database), pengujian regresi, dan pelacakan.
- Rilis kode ke lingkungan produksi secara bertahap, setelah setiap build berhasil.
- Kelola flag fitur (toggle) secara efektif: mendukung konfigurasi run-time, memantau penggunaan, memelihara flag sepanjang siklus pengembangan, dan menetapkan pemilik berdasarkan kategori.
- Berikan ketertelusuran di pipeline build Anda, untuk melacak pemicu, pemberitahuan kegagalan, dan penyelesaian yang berhasil.
- Jalankan proses penerapan otomatis dan tes untuk pembaruan kode “no sentuhan” dalam pengiriman berkelanjutan.
- Gunakan metodologi penyebaran biru/hijau zero-downtime yang sepenuhnya otomatis.
- Pastikan bahwa perubahan skema database Anda diimplementasikan secara konsisten di semua lingkungan pengembangan dan produksi.

Jalankan

- Buat runbook DevOps triase yang terintegrasi dengan sistem notifikasi Anda.
- Pastikan bahwa sistem pemantauan dan notifikasi Anda memenuhi tujuan tingkat layanan (SLOs) dan mendukung ambang batas, pemeriksaan kesehatan, respons HTTP non-standar, dan hasil yang tidak terduga.
- Menetapkan manajemen risiko yang efektif dan proses pemulihan bencana.
- Kembangkan strategi rotasi dan retensi log yang memenuhi persyaratan bisnis dan hukum Anda.
- Kembangkan dasbor yang melacak kinerja produk, mengukur keberhasilan fitur baru, dan menampilkan peringatan saat metrik tidak memenuhi harapan.

Optimalkan

- Tinjau dan tingkatkan proses secara teratur, berdasarkan ukuran kinerja dan kualitas.
- Menerapkan analisis akar penyebab dan proses pencegahan untuk mencegah masalah berulang.
- Berikan metrik berbasis data yang menangkap kesehatan produk, dan pastikan bahwa semua pemberitahuan dan tindakan didasarkan pada metrik ini.

Kesiapan

- Dedikasikan tim lintas fungsi (termasuk mitra bisnis, pengembang, penguji, dan arsitek) untuk upaya modernisasi Anda.

Langkah selanjutnya

Panduan ini memberikan pendekatan bertahap dan bertahap untuk memodernisasi aplikasi Anda. Kami menyarankan Anda mulai melihat alur kerja Anda dengan teknologi dan pengiriman untuk jangka pendek, menengah, dan panjang. Anda dapat berharap untuk berinvestasi lebih banyak dalam alat, kerangka kerja, dan praktik untuk mendorong praktik pengiriman dan teknik yang lebih efektif di seluruh tim. Tetapkan tujuan untuk proses modernisasi, pahami setiap aplikasi yang Anda rencanakan untuk dimodernisasi, dan pilih pendekatan modernisasi yang optimal untuk setiap aplikasi. Pantau dan optimalkan kemajuan Anda.

AWS akan terlibat dengan tim pengembangan bisnis organisasi Anda untuk mendorong modernisasi teknologi dan mengelola end-to-end kebutuhan Anda, termasuk arsitektur aplikasi, desain, dan pengembangan; pemberdayaan seluler; pengujian; dan solusi kolaboratif. Layanan ini menggabungkan proses yang telah terbukti, otomatisasi cerdas, penggunaan data dan pola, standar terbuka, dan orang yang tepat untuk membantu Anda memodernisasi aplikasi lama Anda. Tujuan utama ditargetkan untuk:

- Membantu mengembangkan platform yang menawarkan pendekatan berbasis alat yang sepenuhnya otomatis untuk memodernisasi teknologi warisan, dan memasangkannya dengan pengetahuan holistik tentang modernisasi.
- Bangun MVP dengan tim full stack, dengan kemampuan teknik modernisasi yang berfokus pada hasil dan skala pelanggan.

AWS Layanan dan AWS Mitra Profesional bekerja secara langsung dengan teknologi senior dan pemangku kepentingan pelanggan bisnis untuk mendorong modernisasi perusahaan, sambil secara berulang memberikan nilai pelanggan akhir.

Sumber daya

Panduan terkait

- [Strategi untuk memodernisasi aplikasi di Cloud AWS](#)
- [Mengevaluasi kesiapan modernisasi untuk aplikasi di Cloud AWS](#)
- [Memodernisasi operasi di Cloud AWS](#)
- [Mengurai monolit menjadi layanan mikro](#)
- [Mengintegrasikan layanan mikro dengan menggunakan layanan tanpa server AWS](#)
- [AWS Ubah Kerangka Kerja 6 Titik Akselerasi dan Toolkit Manajemen Perubahan Organisasi](#)

AWS sumber daya

- [AWS dokumentasi](#)
- [AWS referensi umum](#)
- [AWS glosarium](#)

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
Panduan yang diperluas	—	25 Mei 2023
Publikasi awal	—	18 Desember 2020

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