Panduan penilaian portofolio aplikasi untuk migrasi AWS Cloud

AWS Bimbingan Preskriptif

AWS Bimbingan Preskriptif: Panduan penilaian portofolio aplikasi untuk migrasi AWS Cloud

Table of Contents

Panduan penilaian portofolio aplikasi untuk migrasi AWS Cloud

Goncalves Jerman, Mark Berner, dan Zach Hansen, Amazon Web Services ()AWS

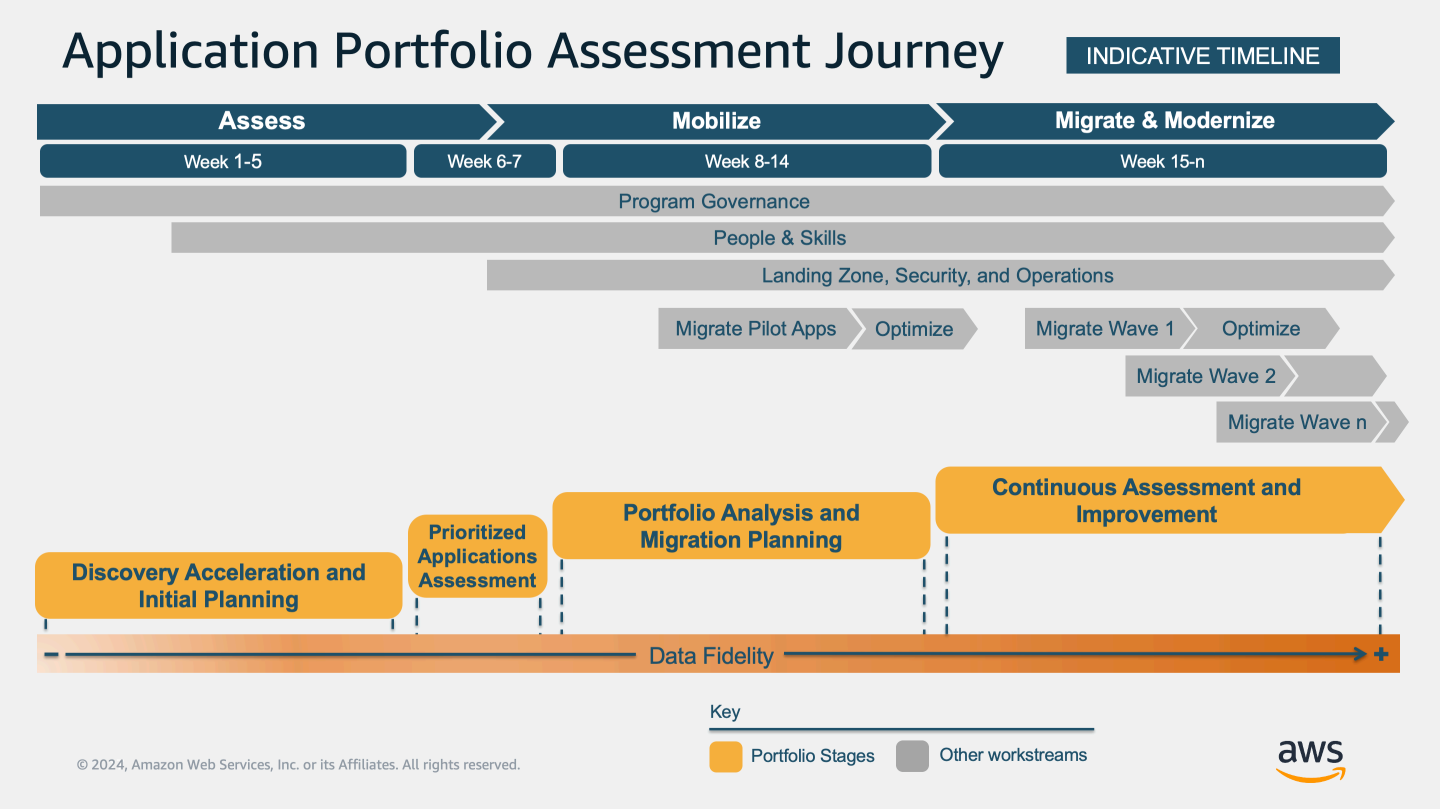
Mei 2024 (riwayat dokumen)

Dokumen Panduan Preskriptif Amazon Web Services (AWS) ini menyelami penerapan strategi penilaian portofolio aplikasi. Anda dapat menggunakan panduan ini untuk membantu Anda memulai dan maju melalui penilaian portofolio aplikasi dan infrastruktur terkait Anda. Penilaian meliputi penemuan, analisis, dan perencanaan. Infrastruktur meliputi komputasi, penyimpanan, dan jaringan.

Gambaran Umum

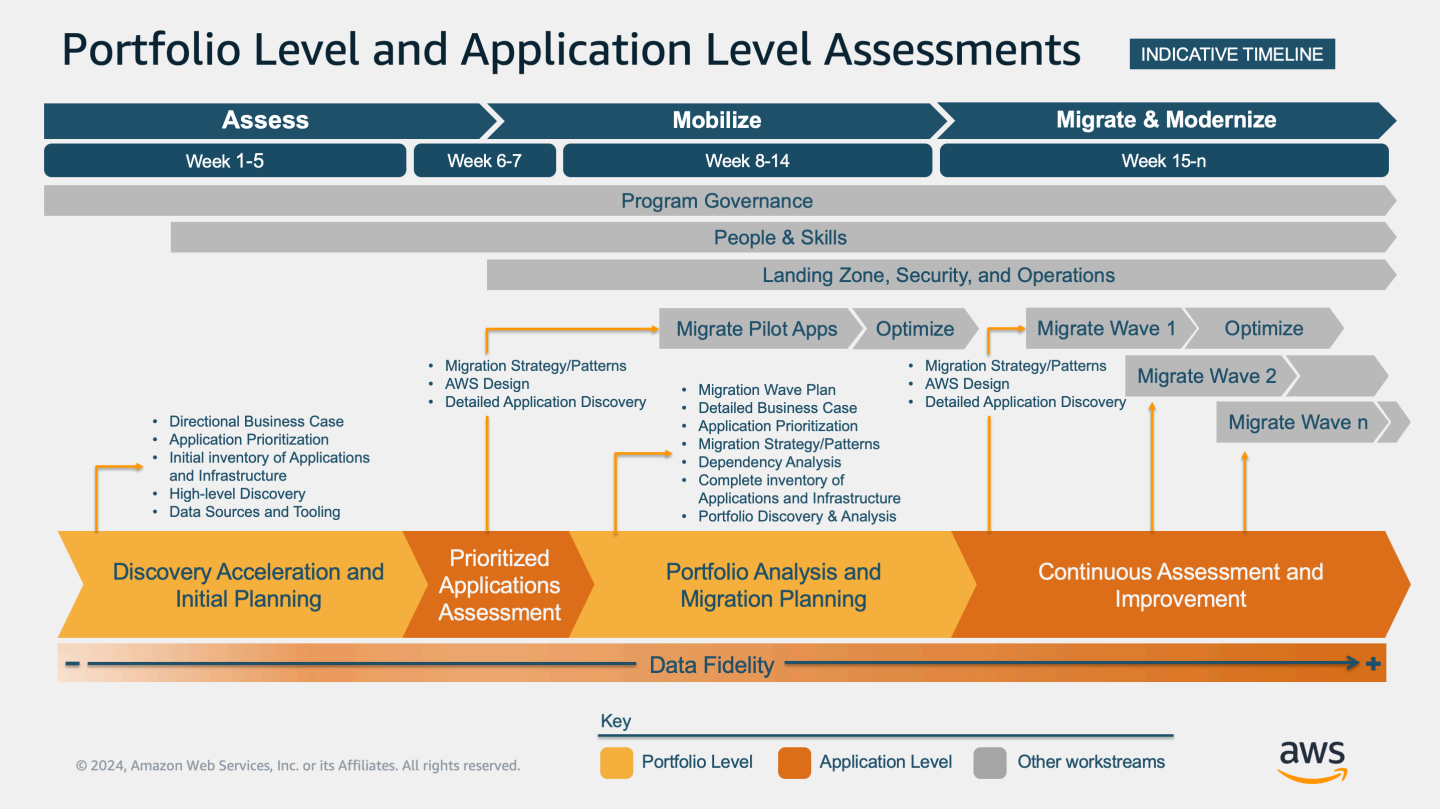
Program migrasi cloud yang berjalan lama memerlukan koordinasi beberapa workstream seperti tata kelola program, landing zone (lingkungan target operatif dengan kontrol keamanan), migrasi, dan portofolio aplikasi. Nama-nama alur kerja ini dapat bervariasi tergantung pada cara Anda memilih untuk mengatur program migrasi Anda. Sebagai alur kerja, penilaian portofolio aplikasi merupakan aktivitas dasar di seluruh siklus hidup program ini. Pemahaman tentang portofolio yang diperoleh melalui penilaian memberikan masukan kunci ke alur kerja lain yang bergantung pada data dan analisis yang dihasilkan dari penilaian portofolio aplikasi berkelanjutan.

Diagram berikut menunjukkan bagaimana tahapan penilaian portofolio sesuai dengan AWS fase migrasi dan alur kerja lainnya. Penemuan portofolio dan tahap perencanaan awal dimulai pada fase penilaian, biasanya selama lima minggu pertama. Penilaian aplikasi yang diprioritaskan, pada minggu keenam dan ketujuh, mencakup fase penilaian dan mobilisasi. Analisis portofolio dan tahap perencanaan migrasi terjadi pada minggu ke 8-14, dalam fase mobilisasi. Tahap penilaian dan peningkatan berkelanjutan terjadi pada fase migrasi dan modernisasi, dari minggu ke 15 hingga akhir program migrasi. Garis waktu ini bersifat indikatif. Durasi sebenarnya dari tahapan akan tergantung pada organisasi program secara keseluruhan. Tahap penilaian portofolio juga berlaku di luar kerangka kerja ini, dan mereka dapat dimasukkan ke dalam struktur program migrasi apa pun.



- Percepatan penemuan dan perencanaan awal berfokus pada pemahaman portofolio saat ini. Ini termasuk membuat kasus bisnis terarah, menetapkan model rasionalisasi dasar untuk migrasi, dan mengidentifikasi kandidat migrasi awal.
- Penilaian aplikasi yang diprioritaskan memberikan lebih cepat time-to-value melalui penilaian terperinci, desain awal arsitektur status target, dan identifikasi aplikasi yang dapat dipindahkan dalam jangka pendek. Memindahkan aplikasi dengan cepat memberi tim pengalaman migrasi dan membangun fondasi cloud, seperti landing zone awal dan komponen infrastruktur lainnya.
- Analisis portofolio dan perencanaan migrasi berfokus pada pembangunan yang lengkap dan up-to-date tampilan portofolio aplikasi. Pandangan ini dibangun dengan memperkaya dataset portofolio secara berulang, menutup kesenjangan data, mengembangkan kasus bisnis, dan membuat rencana gelombang migrasi dengan kepercayaan tinggi.
- Penilaian dan peningkatan berkelanjutan mendukung migrasi dalam skala besar dengan menghasilkan aplikasi terperinci dan penilaian teknologi untuk setiap gelombang migrasi sebagai aktivitas berkelanjutan. Tahap ini mencakup iterasi rencana gelombang migrasi dan melakukan analisis lebih lanjut dari beban kerja yang dimigrasi untuk optimasi dan modernisasi.

Diagram berikut menunjukkan kegiatan utama untuk setiap tahap penilaian dan bagaimana mereka berputar antara penilaian tingkat portofolio dan penilaian tingkat aplikasi. Penilaian tingkat portofolio berfokus pada penemuan tingkat tinggi dan analisis keseluruhan portofolio. Misalnya, sumber data portofolio, inventaris aplikasi dan infrastruktur, prioritas, dan kasus bisnis terarah. Penilaian tingkat aplikasi berfokus pada penemuan terperinci dari satu atau lebih aplikasi. Misalnya, penemuan aplikasi terperinci, AWS desain target, dan strategi migrasi di tingkat arsitektur dan teknologi aplikasi. Penilaian tingkat portofolio dan tingkat aplikasi mewakili luas dan kedalaman informasi yang diperlukan.



Akselerasi penemuan dan perencanaan awal

Tahap pertama penilaian portofolio ini berfokus pada langkah-langkah awal memperoleh dan menganalisis data di tingkat portofolio. Tujuan utamanya adalah untuk mengidentifikasi driver bisnis dan mengumpulkan data umum dari aplikasi dan infrastruktur untuk mendapatkan pandangan awal portofolio. Data ini mencakup atribut teknis dan bisnis tingkat tinggi seperti nama aplikasi, lingkungan, versi produk, kekritisan, nilai kinerja, dan lainnya, seperti yang dijelaskan di bagian persyaratan data. Menyelesaikan tahap ini adalah kunci untuk memahami ruang lingkup proyek, mengidentifikasi kandidat migrasi awal, dan menginformasikan kasus bisnis.

Hasil utama dari tahap ini

- Penggerak bisnis, hasil, tujuan, dan prinsip panduan teknis yang terdokumentasi.
- Inventarisasi awal aplikasi dan infrastruktur, dan kesenjangan data yang diidentifikasi. Ini adalah pandangan awal dari portofolio yang akan diulang dan disempurnakan dalam tahap selanjutnya.
- Kasus bisnis terarah dan perkiraan biaya untuk bermigrasi.
- Daftar kandidat migrasi awal (misalnya, tiga-lima aplikasi).
- Ditentukan langkah selanjutnya

Memahami persyaratan data penilaian awal

Pengumpulan data dapat memakan banyak waktu dan dengan mudah menjadi pemblokir ketika tidak ada kejelasan tentang data apa yang dibutuhkan dan kapan dibutuhkan. Kuncinya adalah memahami keseimbangan antara apa yang terlalu sedikit dan apa yang terlalu banyak data untuk hasil tahap ini. Untuk fokus pada data dan tingkat kesetiaan yang diperlukan untuk tahap awal penilaian portofolio ini, mengadopsi pendekatan berulang untuk pengumpulan data.

Sumber data dan persyaratan data

Langkah pertama adalah mengidentifikasi sumber data Anda. Mulailah dengan mengidentifikasi pemangku kepentingan utama dalam organisasi Anda yang dapat memenuhi persyaratan data. Ini biasanya anggota manajemen layanan, operasi, perencanaan kapasitas, pemantauan, dan tim dukungan, dan pemilik aplikasi. Buat sesi kerja dengan anggota kelompok-kelompok ini. Komunikasikan persyaratan data dan dapatkan daftar alat dan dokumentasi yang ada yang dapat menyediakan data.

Untuk memandu percakapan ini, gunakan serangkaian pertanyaan berikut:

- Seberapa akurat dan mutakhir infrastruktur dan inventaris aplikasi saat ini? Misalnya, untuk database manajemen konfigurasi perusahaan (CMDB), apakah kita sudah tahu di mana celahnya?
- Apakah kami memiliki alat dan proses aktif yang membuat CMDB (atau yang setara) diperbarui? Jika demikian, seberapa sering diperbarui? Apa tanggal penyegaran terbaru?
- Apakah inventaris saat ini, seperti CMDB, berisi application-to-infrastructure pemetaan? Apakah setiap aset infrastruktur terkait dengan aplikasi? Apakah setiap aplikasi dipetakan ke infrastruktur?
- Apakah inventaris berisi katalog lisensi dan perjanjian lisensi untuk setiap produk?
- Apakah inventaris berisi data ketergantungan? Perhatikan keberadaan data komunikasi seperti server ke server, aplikasi ke aplikasi, aplikasi atau server ke database.
- Alat apa lagi yang dapat menyediakan informasi aplikasi dan infrastruktur yang tersedia di lingkungan? Perhatikan keberadaan kinerja, pemantauan, dan alat manajemen yang dapat digunakan sebagai sumber data.
- Apa saja lokasi yang berbeda, seperti pusat data, hosting aplikasi dan infrastruktur kita?

Setelah pertanyaan-pertanyaan ini dijawab, daftarkan sumber data Anda yang teridentifikasi. Kemudian tetapkan tingkat kesetiaan, atau tingkat kepercayaan, untuk masing-masing. Data yang divalidasi baru-baru ini (dalam 30 hari) dari sumber program aktif, seperti alat, memiliki tingkat kesetiaan tertinggi. Data statis dianggap memiliki kesetiaan yang lebih rendah dan kurang dipercaya. Contoh data statis adalah dokumen, buku kerja, CMDB yang diperbarui secara manual, atau kumpulan data lain yang tidak dipelihara secara terprogram, atau yang tanggal penyegaran terakhirnya lebih dari 60 hari.

Tingkat kesetiaan data dalam tabel berikut disediakan sebagai contoh. Kami menyarankan Anda menilai persyaratan organisasi Anda dalam hal toleransi maksimum terhadap asumsi dan risiko terkait untuk menentukan tingkat kesetiaan yang sesuai. Dalam tabel, pengetahuan kelembagaan mengacu pada informasi apa pun tentang aplikasi dan infrastruktur yang tidak didokumentasikan.

Sumber data	Tingkat kesetiaan	Cakupan portofolio	Komentar
Pengetahuan kelembagaan	Rendah - Hingga 25% dari data akurat, 75% nilai atau data yang diasumsikan lebih tua dari 150 hari.	Rendah	Langka, fokus pada aplikasi kritis

Sumber data	Tingkat kesetiaan	Cakupan portofolio	Komentar
Basis pengetahuan	Sedang rendah - 35-40% dari data akurat, 65-60% nilai atau data yang diasumsikan berusia 120-150 hari.	Sedang	Tingkat detail yang dipelihara secara manual dan tidak konsisten
CMDB	Sedang - ~ 50% dari data akurat, ~ 50% nilai atau data yang diasumsikan berusia 90-120 hari.	Sedang	Berisi data dari sumber campuran, beberapa kesenjang an data
Ekspor vCenter VMware	Menengah-tinggi - 75-80% dari data akurat, 25-20% nilai asumsi atau data berusia 60-90 hari.	Tinggi	Mencakup 90% dari perkebunan virtual
Pemantauan kinerja aplikasi	Tinggi - Sebagian besar data akurat, ~ 5% nilai atau data yang diasumsikan berusia 0-60 hari.	Rendah	Terbatas untuk sistem produksi kritis (mencakup 15% dari portofolio aplikasi)

Tabel berikut menentukan atribut data yang diperlukan dan opsional untuk setiap kelas aset (aplikasi, infrastruktur, jaringan, dan migrasi), aktivitas spesifik (inventaris atau kasus bisnis), dan kesetiaan data yang direkomendasikan untuk tahap penilaian ini. Tabel menggunakan singkatan berikut:

- R, untuk yang dibutuhkan
- (D), untuk kasus bisnis terarah, diperlukan untuk perbandingan biaya kepemilikan total (TCO) dan kasus bisnis terarah
- (F), untuk kasus bisnis terarah penuh, diperlukan untuk perbandingan TCO dan kasus bisnis terarah yang mencakup biaya migrasi dan modernisasi
- O, untuk opsional

- N/A, karena tidak berlaku

Aplikasi

Nama atribut	Deskripsi	Inventarisasi dan prioritas	Kasus bisnis	Tingkat kesetiaan yang disarankan (minimum)
Pengenal unik	Misalnya, ID aplikasi. Biasanya tersedia pada CMDB yang ada atau inventaris internal dan sistem kontrol lainnya. Pertimbangkan untuk membuat ID unik setiap kali ini tidak ditentukan dalam organisasi Anda.	R	R (D)	Tinggi
Nama aplikasi	Nama yang dengannya aplikasi ini diketahui organisasi Anda. Sertakan vendor komersial off-the-shelf (COTS) dan nama produk jika berlaku.	R	R (D)	Tinggi medium

Nama atribut	Deskripsi	Inventarisasi dan prioritas	Kasus bisnis	Tingkat kesetiaan yang disarankan (minimum)
Apakah COTS?	Ya atau Tidak. Apakah ini aplikasi komersial atau pengembangan internal	R	R (D)	Tinggi medium
Produk dan versi COTS	Nama dan versi produk perangkat lunak komersial	R	R (D)	Sedang
Deskripsi	Fungsi dan konteks aplikasi utama	R	O	Sedang
Kekritisan	Misalnya, aplikasi strategis atau menghasilkan pendapatan, atau mendukung fungsi kritis	R	O	Tinggi medium
Tipe	Misalnya, database, manajemen hubungan pelanggan (CRM), aplikasi web, multimedia, layanan bersama TI	R	O	Sedang

Nama atribut	Deskripsi	Inventarisasi dan prioritas	Kasus bisnis	Tingkat kesetiaan yang disarankan (minimum)
Environment	Misalnya, produksi, pra-produksi, pengembangan, pengujian, kotak pasir	R	R (D)	Tinggi medium
Kepatuhan dan peraturan	Kerangka kerja yang berlaku untuk beban kerja (misalnya , HIPAA, SOX, PCI-DSS, ISO, SOC, FedRAMP) dan persyaratan peraturan	R	R (D)	Tinggi medium
Dependensi	Dependensi hulu dan hilir ke aplikasi atau layanan internal dan eksternal . Dependens i non-teknis seperti elemen operasional (misalnya, siklus pemeliharaan)	O	O	Rendah medium

Nama atribut	Deskripsi	Inventarisasi dan prioritas	Kasus bisnis	Tingkat kesetiaan yang disarankan (minimum)
Pemetaan infrastruktur	Pemetaan ke aset fisik dan/atau virtual yang membentuk aplikasi	O	O	Sedang
Lisensi	Jenis lisensi perangkat lunak komoditas (misalnya, Microsoft SQL Server Enterprise)	O	R	Tinggi medium
Biaya	Biaya untuk lisensi perangkat lunak, operasi perangkat lunak, dan pemeliharaan	N/A	O	Sedang

Infrastruktur

Nama Atribut	Deskripsi	Inventarisasi dan prioritas	Kasus bisnis	Tingkat kesetiaan yang disarankan (minimum)
Pengenal unik	Misalnya, ID server. Biasanya tersedia pada CMDB yang ada	R	R	Tinggi

	atau inventaris internal dan sistem kontrol lainnya. Pertimbangkan untuk membuat ID unik setiap kali ini tidak ditentukan dalam organisasi Anda.			
Nama jaringan	Nama aset dalam jaringan (misalnya, nama host)	R	O	Tinggi medium
Nama DNS (nama domain yang sepenuhnya memenuhi syarat, atau FQDN)	Nama DNS	O	O	Sedang
Alamat IP dan netmask	Alamat IP internal dan/atau publik	R	O	Tinggi medium
Jenis aset	Server fisik atau virtual, hypervisor, wadah, perangkat, instance database, dll.	R	R	Tinggi medium

Nama produk	Vendor komersial dan nama produk (misalnya, VMware ESXi, IBM Power Systems, Exadata)	R	R	Sedang
Sistem operasi	Misalnya, REHL 8, Windows Server 2019, AIX 6.1	R	R	Tinggi medium
Konfigurasi	CPU yang dialokasikan, jumlah inti, utas per inti, total memori, penyimpanan, kartu jaringan	R	R	Tinggi medium
Pemanfaatan	CPU, memori, dan puncak penyimpanan dan rata-rata . Database instance throughput.	R	O	Tinggi medium
Lisensi	Jenis lisensi komoditas (misalnya, Standar RHEL)	R	R	Sedang

Apakah infrastruktur bersama?	Ya atau Tidak untuk menunjukkan layanan infrastruktur yang menyediakan layanan bersama seperti penyedia otentikasi, sistem pemantauan, layanan cadangan, dan layanan serupa	R	R (D)	Sedang
Pemetaan aplikasi	Aplikasi atau komponen aplikasi yang berjalan di infrastruktur ini	O	O	Sedang
Biaya	Biaya terisi penuh untuk server bare-metal, termasuk perangkat keras, pemeliharaan, operasi, penyimpanan (SAN, NAS, Object), lisensi sistem operasi, pangsa rackspace, dan overhead pusat data	N/A	O	Tinggi medium

Jaringan

Nama Atribut	Deskripsi	Inventarisasi dan prioritas	Kasus bisnis	Tingkat kesetiaan yang disarankan (minimum)
Ukuran pipa (MB/s), redundansi (Y/N)	Spesifikasi tautan WAN saat ini (mis., 1000 MB/s redundan)	O	R	Sedang
Pemanfaatan tautan	Puncak dan pemanfaatan rata-rata, transfer data keluar (GB/bulan)	O	R	Sedang
Latensi (ms)	Latensi saat ini antara lokasi yang terhubung.	O	O	Sedang
Biaya	Biaya saat ini per bulan	N/A	O	Sedang

Migrasi

Nama Atribut	Deskripsi	Inventarisasi dan prioritas	Kasus bisnis	Tingkat kesetiaan yang disarankan (minimum)
Rehost	Upaya pelanggan dan mitra untuk setiap beban kerja (orang-hari), tarif biaya pelanggan	N/A	R (F)	Tinggi medium

	dan Mitra per hari, biaya alat, jumlah beban kerja			
Platform Ulang	Upaya pelanggan dan mitra untuk setiap beban kerja (orang-hari), tarif biaya pelanggan dan mitra per hari, jumlah beban kerja	N/A	R (F)	Tinggi medium
Refactor	Upaya pelanggan dan mitra untuk setiap beban kerja (orang-hari), tarif biaya pelanggan dan mitra per hari, jumlah beban kerja	N/A	O	Tinggi medium
Pensiun	Jumlah server, biaya dekomisi rata-rata	N/A	O	Tinggi medium
Zona pendaratan	Gunakan kembali yang ada (Y/N), daftar AWS Wilayah yang dibutuhkan, biaya	N/A	R (F)	Tinggi medium

Orang dan perubahan	Jumlah staf yang akan dilatih dalam operasi dan pengembangan cloud, biaya pelatihan per orang, biaya waktu pelatihan per orang	N/A	R (F)	Tinggi medium
Durasi	Durasi migrasi beban kerja dalam lingkup (bulan)	O	R (F)	Tinggi medium
Biaya paralel	Kerangka waktu dan tingkat biaya apa adanya dapat dihapus selama migrasi	N/A	O	Tinggi medium
	Kerangka waktu dan tingkat di mana AWS produk dan layanan, dan biaya infrastruktur lainnya, diperkenalkan selama migrasi	N/A	O	Tinggi medium

Mengevaluasi kebutuhan alat penemuan

Apakah organisasi Anda membutuhkan alat penemuan? Penilaian portofolio membutuhkan kepercayaan tinggi, up-to-date data tentang aplikasi dan infrastruktur. Tahap awal penilaian portofolio dapat menggunakan asumsi untuk mengisi kesenjangan data.

Namun, seiring kemajuan yang dicapai, data dengan ketelitian tinggi memungkinkan pembuatan rencana migrasi yang berhasil dan estimasi infrastruktur target yang benar untuk mengurangi biaya dan memaksimalkan manfaat. Ini juga mengurangi risiko dengan mengaktifkan implementasi yang mempertimbangkan dependensi dan menghindari jebakan migrasi. Kasus penggunaan utama untuk alat penemuan dalam program migrasi cloud adalah untuk mengurangi risiko dan meningkatkan tingkat kepercayaan pada data melalui hal-hal berikut:

- Pengumpulan data otomatis atau terprogram, menghasilkan data yang divalidasi dan sangat tepercaya
- Akselerasi tingkat di mana data diperoleh, meningkatkan kecepatan proyek dan mengurangi biaya
- Peningkatan tingkat kelengkapan data, termasuk data komunikasi dan dependensi yang biasanya tidak tersedia di CMDB
- Memperoleh wawasan seperti identifikasi aplikasi otomatis, analisis TCO, laju operasional yang diproyeksikan, dan rekomendasi pengoptimalan
- Perencanaan gelombang migrasi kepercayaan tinggi

Ketika ada ketidakpastian tentang apakah sistem ada di lokasi tertentu, sebagian besar alat penemuan dapat memindai subnet jaringan dan menemukan sistem yang merespons permintaan ping atau Simple Network Management Protocol (SNMP). Perhatikan bahwa tidak semua konfigurasi jaringan atau sistem akan memungkinkan lalu lintas ping atau SNMP. Diskusikan opsi ini dengan jaringan dan tim teknis Anda.

Tahap lebih lanjut dari penilaian portofolio aplikasi dan migrasi sangat bergantung pada informasi pemetaan ketergantungan yang akurat. Pemetaan ketergantungan memberikan pemahaman tentang infrastruktur dan konfigurasi yang akan diperlukan AWS (seperti grup keamanan, jenis instance, penempatan akun, dan perutean jaringan). Ini juga membantu pengelompokan aplikasi yang harus bergerak pada saat yang sama (seperti aplikasi yang harus berkomunikasi melalui jaringan latensi rendah). Selain itu, pemetaan ketergantungan memberikan informasi untuk mengembangkan kasus bisnis.

Saat memutuskan alat penemuan, penting untuk mempertimbangkan semua tahapan proses penilaian dan untuk mengantisipasi persyaratan data. Kesenjangan data berpotensi menjadi pemblokir, jadi penting untuk mengantisipasinya dengan menganalisis persyaratan data dan sumber data masa depan. Pengalaman di lapangan menentukan bahwa sebagian besar proyek migrasi yang macet memiliki kumpulan data terbatas di mana aplikasi dalam ruang lingkup, infrastruktur terkait, dan dependensinya tidak diidentifikasi dengan jelas. Kurangnya identifikasi ini dapat menyebabkan

metrik, keputusan, dan penundaan yang salah. Memperoleh up-to-date data adalah langkah pertama menuju proyek migrasi yang sukses.

Bagaimana cara memilih alat penemuan?

Beberapa alat penemuan di pasar menyediakan fitur dan kemampuan yang berbeda. Pertimbangkan kebutuhan Anda. Dan tentukan opsi yang paling tepat untuk organisasi Anda. Faktor paling umum saat memutuskan alat penemuan untuk migrasi adalah sebagai berikut:

Keamanan

- Apa metode otentikasi untuk mengakses repositori data alat atau mesin analitik?
- Siapa yang dapat mengakses data, dan apa kontrol keamanan untuk mengakses alat?
- Bagaimana alat mengumpulkan data? Apakah perlu kredensyal khusus?
- Kredensi dan tingkat akses apa yang dibutuhkan alat untuk mengakses sistem saya dan mendapatkan data?
- Bagaimana data ditransfer antar komponen alat?
- Apakah alat ini mendukung enkripsi data saat istirahat dan dalam perjalanan?
- Apakah data terpusat dalam satu komponen di dalam atau di luar lingkungan saya?
- Apa persyaratan jaringan dan firewall?

Pastikan bahwa tim keamanan terlibat dalam percakapan awal tentang alat penemuan.

Kedaulatan data

- Dimana data disimpan dan diproses?
- Apakah alat tersebut menggunakan model perangkat lunak sebagai layanan (SaaS)?
- Apakah itu memiliki kemungkinan untuk menyimpan semua data dalam batas-batas lingkungan saya?
- Dapatkah data disaring sebelum meninggalkan batas-batas organisasi saya?

Pertimbangkan kebutuhan organisasi Anda dalam hal persyaratan residensi data.

Arsitektur

- Infrastruktur apa yang dibutuhkan dan komponen apa yang berbeda?

- Apakah lebih dari satu arsitektur tersedia?
- Apakah alat ini mendukung pemasangan komponen di zona keamanan terkunci udara?

Kinerja

- Apa dampak pengumpulan data pada sistem saya?

Kompatibilitas dan ruang lingkup

- Apakah alat ini mendukung semua atau sebagian besar produk dan versi saya? Tinjau dokumentasi alat untuk memverifikasi platform yang didukung terhadap informasi terkini tentang ruang lingkup Anda.
- Apakah sebagian besar sistem operasi saya didukung untuk pengumpulan data? Jika Anda tidak tahu versi sistem operasi Anda, cobalah untuk mempersempit daftar alat penemuan untuk mereka yang memiliki jangkauan yang lebih luas dari sistem yang didukung.

Metode pengumpulan

- Apakah alat ini perlu menginstal agen pada setiap sistem yang ditargetkan?
- Apakah itu mendukung penerapan tanpa agen?
- Apakah agen dan tanpa agen menyediakan fitur yang sama?
- Apa proses pengumpulannya?

Fitur

- Apa saja fitur yang tersedia?
- Dapatkah menghitung total biaya kepemilikan (TCO) dan estimasi tingkat AWS operasional Cloud?
- Apakah itu mendukung perencanaan migrasi?
- Apakah itu mengukur kinerja?
- Bisakah itu merekomendasikan AWS infrastruktur target?
- Apakah itu melakukan pemetaan ketergantungan?
- Tingkat pemetaan ketergantungan apa yang disediakannya?
- Apakah itu menyediakan akses API? (misalnya, dapatkah diakses secara terprogram untuk mendapatkan data?)

Pertimbangkan alat dengan fungsi pemetaan ketergantungan aplikasi dan infrastruktur yang kuat dan yang dapat menyimpulkan aplikasi dari pola komunikasi.

Biaya

- Apa model perizinannya?
- Berapa biaya lisensi?
- Apakah harga untuk setiap server? Apakah itu harga berjenjang?
- Apakah ada opsi dengan fitur terbatas yang dapat dilisensikan sesuai permintaan?

Alat penemuan biasanya digunakan di seluruh siklus hidup proyek migrasi. Jika anggaran Anda terbatas, pertimbangkan setidaknya 6 bulan. Namun, tidak adanya alat penemuan biasanya mengarah pada upaya manual dan biaya internal yang lebih tinggi.

Model Support

- Tingkat dukungan apa yang disediakan secara default?
- Apakah ada rencana dukungan yang tersedia?
- Berapa waktu respons insiden?

Layanan profesional

- Apakah vendor menawarkan layanan profesional untuk menganalisis hasil penemuan?
- Bisakah mereka mencakup elemen-elemen panduan ini?
- Apakah ada diskon atau bundel untuk layanan perkakas +?

Fitur yang direkomendasikan untuk alat penemuan

Untuk menghindari penyediaan dan penggabungan data dari beberapa alat dari waktu ke waktu, alat penemuan harus mencakup fitur minimum berikut:

- Perangkat lunak — Alat penemuan harus dapat mengidentifikasi proses yang berjalan dan perangkat lunak yang diinstal.
- Pemetaan ketergantungan — Ini harus dapat mengumpulkan informasi koneksi jaringan dan membangun peta ketergantungan masuk dan keluar dari server dan aplikasi yang sedang berjalan. Selain itu, alat penemuan harus dapat menyimpulkan aplikasi dari kelompok infrastruktur berdasarkan pola komunikasi.

- Penemuan profil dan konfigurasi - Ini harus dapat melaporkan profil infrastruktur seperti keluarga CPU (misalnya, x86, PowerPC), jumlah inti CPU, ukuran memori, jumlah disk dan ukuran, dan antarmuka jaringan.
- Penemuan penyimpanan jaringan — Ini harus dapat mendeteksi dan memprofilkan berbagi jaringan dari penyimpanan terlampir jaringan (NAS).
- Kinerja — Ini harus dapat melaporkan puncak dan rata-rata pemanfaatan CPU, memori, disk, dan jaringan.
- Analisis kesenjangan — Ini harus dapat memberikan wawasan tentang kuantitas dan kesetiaan data.
- Pemindaian jaringan — Ini harus dapat memindai subnet jaringan dan menemukan aset infrastruktur yang tidak diketahui.
- Pelaporan — Ini harus dapat memberikan status pengumpulan dan analisis.
- Akses API — Ini harus dapat menyediakan sarana terprogram untuk mengakses data yang dikumpulkan.

Fitur tambahan untuk dipertimbangkan

- Analisis TCO untuk memberikan perbandingan biaya antara biaya lokal saat ini dan biaya yang diproyeksikan AWS .
- Analisis lisensi dan rekomendasi pengoptimalan untuk Microsoft SQL Server dan sistem Oracle dalam skenario rehost dan replatform.
- Rekomendasi strategi migrasi (Dapatkah alat penemuan membuat rekomendasi tipe R migrasi default berdasarkan teknologi saat ini?)
- Ekspor inventaris (ke CSV atau format serupa)
- Rekomendasi ukuran kanan (misalnya, dapatkah itu memetakan AWS infrastruktur target yang direkomendasikan?)
- Visualisasi ketergantungan (misalnya, dapatkah pemetaan ketergantungan divisualisasikan dalam mode grafis?)
- Tampilan arsitektur (misalnya, dapatkah diagram arsitektur diproduksi secara otomatis?)
- Prioritas aplikasi (Dapatkah itu menetapkan bobot atau relevansi dengan atribut aplikasi dan infrastruktur untuk membuat kriteria prioritas untuk migrasi?)
- Perencanaan gelombang (misalnya, kelompok aplikasi yang direkomendasikan dan kemampuan untuk membuat rencana gelombang migrasi)

- Estimasi biaya migrasi (estimasi upaya migrasi)

Pertimbangan penyebaran

Setelah Anda memilih dan mendapatkan alat penemuan, pertimbangkan pertanyaan berikut untuk mendorong percakapan dengan tim yang bertanggung jawab untuk menerapkan alat di organisasi Anda:

- Apakah server atau aplikasi dioperasikan oleh pihak ketiga? Ini bisa mendikte tim untuk melibatkan dan proses yang harus diikuti.
- Apa proses tingkat tinggi untuk mendapatkan persetujuan untuk menyebarkan alat penemuan?
- Apa proses otentikasi utama untuk mengakses sistem seperti server, kontainer, penyimpanan, dan database? Apakah kredensyal server lokal atau terpusat? Bagaimana proses untuk mendapatkan kredensyal? Kredensil akan diperlukan untuk mengumpulkan data dari sistem Anda (misalnya, kontainer, server virtual atau fisik, hypervisor, dan database). Memperoleh kredensi untuk alat penemuan untuk terhubung ke setiap aset dapat menjadi tantangan, terutama ketika aset ini tidak terpusat.
- Apa garis besar zona keamanan jaringan? Apakah diagram jaringan tersedia?
- Bagaimana proses untuk meminta aturan firewall di pusat data?
- Apa perjanjian tingkat layanan dukungan (SLA) saat ini dalam kaitannya dengan operasi pusat data (instalasi alat penemuan, permintaan firewall)?

Penggerak bisnis dan prinsip panduan teknis

Pengemudi bisnis

Apakah organisasi Anda telah memutuskan untuk pindah ke cloud atau mendekati keputusan itu, mendefinisikan dan mendokumentasikan driver bisnis untuk migrasi cloud akan mengklarifikasi alasan migrasi. Setelah alasan didokumentasikan, Anda dapat menentukan apa yang akan dimigrasikan dan bagaimana hal itu akan dimigrasikan. Kegiatan ini penting. Kami menyarankan agar dilakukan sedini mungkin dalam proses untuk menginformasikan dan memandu langkah selanjutnya.

Identifikasi pemangku kepentingan yang harus menjadi bagian dari diskusi untuk mendokumentasikan driver. Biasanya CxOs, manajer senior, dan pemimpin teknologi kunci dalam organisasi, dan pelanggan Anda sendiri. Meskipun pelanggan Anda tidak mungkin menjadi bagian

dari diskusi ini, kami menyarankan agar satu atau lebih orang di organisasi Anda ditunjuk mewakili pandangan dan tujuan pelanggan Anda.

Penggerak bisnis harus dikaitkan dengan metrik yang dapat diukur sepanjang perjalanan migrasi untuk memvalidasi apakah hasil telah tercapai. Sasaran strategis dan laporan tahunan perusahaan dapat bertindak sebagai titik awal.

Fokuskan percakapan di mana perusahaan ingin berada, berdasarkan metrik yang ada dan yang diproyeksikan, sebagai hasil dari pindah ke cloud. Pertimbangkan tujuan dan hasil bisnis. Juga, pertimbangkan seperti apa kesuksesan saat adopsi cloud meningkat.

Selanjutnya, tetapkan tingkat kepentingan untuk setiap pengemudi. Apa prioritasnya? Apa manfaat yang diharapkan? Bagaimana manfaat mendukung tujuan dan hasil bisnis? Dalam konteks penilaian portofolio aplikasi, jawabannya akan membantu memprioritaskan beban kerja untuk migrasi dan menetapkan prinsip panduan teknis. Namun, driver bisnis akan menentukan dan berdampak pada program migrasi secara keseluruhan.

Prinsip panduan teknis

Prinsip panduan teknis menginformasikan pemilihan strategi migrasi pada tahap selanjutnya dari penilaian portofolio. Pada tahap saat ini, fokusnya adalah mengidentifikasi mereka.

Prinsip-prinsip panduan dapat ditetapkan sebagai keputusan terkait teknologi umum dan terkait pendekatan yang berasal dari tujuan dan hasil bisnis.

Misalnya, perusahaan memiliki tujuan utama untuk mengurangi biaya, dan hasil yang diinginkan adalah menutup pusat data lokal pada tanggal tertentu dalam 6-12 bulan. Prinsip panduan yang dihasilkan adalah mengangkat dan memindahkan semua aplikasi ke cloud dengan menggunakan rehost atau relokasi strategi migrasi bila memungkinkan. Dalam hal ini, lift-and-shift pendekatan mempercepat hasil migrasi jangka pendek. Setelah aplikasi pindah dari pusat data lokal, perusahaan dapat fokus pada driver bisnis utama untuk mengoptimalkan atau memodernisasi beban kerja yang dimigrasi.

Untuk menetapkan prinsip-prinsip panduan teknis, mulailah dengan menganalisis driver bisnis. Identifikasi daftar teknologi dan teknik yang akan mencapai tujuan dan hasil bisnis. Selanjutnya, perbaiki daftar dan tetapkan urutan relevansi berdasarkan kesesuaian atau preferensi untuk mencapai hasil yang diinginkan.

Mendokumentasikan dan mengkomunikasikan prinsip-prinsip panduan dengan orang-orang yang terlibat dalam perencanaan dan pelaksanaan migrasi. Sorot kekhawatiran dan potensi konflik antara prinsip dan implementasi aktual.

Tabel berikut memberikan contoh driver bisnis dan prinsip-prinsip panduan teknis.

Pengemudi bisnis	Hasil	Metrik-metrik	Prinsip panduan teknis
Mempercepat inovasi.	Peningkatan daya saing, peningkatan kelincahan bisnis	Jumlah penyebaran per hari atau bulan, fitur baru yang dirilis per kuartal, skor kepuasan pelanggan, jumlah eksperimen	Memfaktorkan ulang aplikasi yang membedakan dengan menggunakan layanan mikro dan model DevOps operasi untuk meningkatkan kelincahan dan kecepatan ke pasar fitur baru.
Mengurangi biaya operasional dan infrastruktur.	Penawaran dan permintaan sesuai, basis biaya elastis (bayar untuk apa yang Anda gunakan)	Variasi pengeluaran dari waktu ke waktu	1. Rehost aplikasi dengan ukuran tepat infrastruktur. 2. Pensiun aplikasi yang memiliki pemanfaatan rendah atau tidak ada.
Meningkatkan ketahanan operasional.	Uptime yang ditingkatkan, mengurangi waktu rata-rata untuk pemulihan	SLA, jumlah insiden	1. Replatform aplikasi ke versi sistem operasi terbaru dan didukung terbaik. 2. Menerapkan arsitektur ketersedi

Pengemudi bisnis	Hasil	Metrik-metrik	Prinsip panduan teknis
			aan tinggi untuk aplikasi penting.
Keluar dari pusat data.	Penutupan pusat data pada tanggal dalam 6-12 bulan	Kecepatan migrasi server	Rehost aplikasi dengan menggunakan Cloud Migration Factory Solution.
Tetap di tempat, tetapi tingkatkan kelincahan dan ketahanan.	Peningkatan daya saing dan uptime sambil tetap berada di tempat	Jumlah penyebaran per hari atau bulan, rilis fitur baru per kuartal, SLA, jumlah insiden	1. Modernisasi sistem dengan memperluas fungsionalitasnya ke cloud. 2. Menilai rehosting atau replatforming ke Outposts. AWS

Memulai pengumpulan data

Pengumpulan data adalah proses pengumpulan metadata dari aplikasi dan infrastruktur. Prosesnya berulang di semua tahap penilaian. Di setiap tahap, kuantitas dan kesetiaan data akan meningkat. Pada tahap ini, fokusnya adalah mengumpulkan data umum yang dapat membantu membangun inventaris awal. Inventaris akan digunakan untuk membuat kasus bisnis terarah dan identifikasi kandidat migrasi awal.

Setelah sumber data saat ini diidentifikasi, kami sarankan untuk mengumpulkan informasi dari sebanyak mungkin sistem. Untuk informasi selengkapnya, lihat persyaratan data untuk tahap ini.

Pendekatan ini memiliki manfaat membantu memperbarui tampilan portofolio saat ini dan pengetahuan organisasi tentang aplikasi dan layanan mereka. Ini juga membantu menentukan apa yang ditargetkan untuk dipindahkan. Pendekatan yang disarankan adalah meninjau data yang ada, seperti output database manajemen konfigurasi (CMDB) dan sistem manajemen layanan teknologi informasi (ITSM). Kemudian buat daftar aset yang ditargetkan untuk pengumpulan data. Jika organisasi Anda memiliki kejelasan lengkap tentang apa yang ada di ruang lingkup dan di luar cakupan migrasi, Anda dapat membatasi pengumpulan data ke sistem yang berada dalam cakupan.

Saat membangun portofolio Anda, pertimbangkan aplikasi dan lingkungannya atau siklus hidup rilis perangkat lunak. Misalnya, alih-alih mengidentifikasi aplikasi manajemen hubungan pelanggan (CRM) dan menentukan bahwa ia memiliki lingkungan pengujian, pengembang, dan prod, daftarkan tiga aplikasi (misalnya, CRM-test, CRM-dev, CRM-prod). Atau, gunakan nama CRM tetapi tetapkan ID unik untuk setiap lingkungan dan sajikan sebagai catatan terpisah di repositori data Anda. Ini akan membantu merencanakan dan melacak migrasi lingkungan ini secara individual. Misalnya, Anda mungkin ingin memigrasikan lingkungan non-produksi terlebih dahulu. Dengan mencantumkan contoh aplikasi Anda sesuai dengan lingkungan, Anda dapat dengan jelas mengelola dan mengatur transisi mereka.

Selama pengumpulan data, mungkin ada ketidakpastian tentang aplikasi atau server mana yang berada di pusat data atau lokasi sumber tertentu. Dalam kasus ini, mendapatkan daftar bare-metal dan hypervisor dari alat manajemen yang ada sangat membantu. Misalnya, Anda dapat terhubung ke hypervisor untuk mendapatkan daftar mesin virtual yang akan ditargetkan untuk pengumpulan data.

Perhatikan bahwa output awal, saat menggabungkan sumber data yang ada, bisa jadi tidak lengkap. Kuncinya adalah melakukan analisis kesenjangan dalam hal persyaratan data untuk tahap ini dan apa yang dapat diperoleh dari sumber yang ada. Penting untuk membedakan persentase kelengkapan dengan tingkat kesetiaan data. Tingkat kelengkapan yang lebih tinggi dari sumber kesetiaan rendah akan berisi beberapa asumsi yang dapat mengarah pada analisis yang salah. Meskipun tahap penilaian ini tidak memerlukan kesetiaan data maksimum, kami merekomendasikan bahwa sumber data setidaknya memiliki kesetiaan sedang hingga menengah-tinggi. Bandingkan angka-angka ini dengan toleransi organisasi Anda terhadap risiko, termasuk penggunaan asumsi untuk mengisi kesenjangan data.

Analisis kesenjangan membantu Anda memahami kuantitas dan kualitas data yang Anda kerjakan. Analisis ini juga membantu Anda menetapkan tingkat asumsi yang harus dibuat untuk membuat kasus bisnis terarah dan memprioritaskan aplikasi untuk migrasi. Alat penemuan dapat membantu mengisi celah dan mengumpulkan data dengan kesetiaan tinggi. Untuk meningkatkan tingkat kepercayaan pada data dan mempercepat hasil migrasi, sebaiknya gunakan alat penemuan sedini mungkin. Tindakan awal juga penting karena proses pengadaan, keamanan, dan implementasi internal untuk alat baru dapat memerlukan beberapa minggu atau bulan untuk menyelesaikannya.

Kami merekomendasikan untuk membuat rencana komunikasi atau irama dan mekanisme kontrol perubahan ruang lingkup pada tahap ini. Ini membantu Anda untuk terus memberi tahu para pemangku kepentingan sehingga mereka dapat merencanakan ke depan dan mengurangi risiko. Elemen kunci untuk komunikasi yang jelas adalah mendefinisikan satu sumber kebenaran untuk portofolio aplikasi dan infrastruktur terkait. Hindari menyimpan beberapa sistem catatan dan daftar

aplikasi dan infrastruktur. Simpan data di satu tempat (misalnya, database, alat, atau spreadsheet) yang mendukung pembuatan versi dan kolaborasi online, dan tetapkan pemiliknya.

Strategi prioritas dan migrasi

Elemen kunci dari perencanaan migrasi adalah menetapkan kriteria prioritas. Inti dari latihan ini adalah untuk memahami urutan aplikasi yang akan dimigrasikan. Strateginya adalah mengambil pendekatan berulang dan progresif untuk mengembangkan model prioritas.

Memprioritaskan aplikasi

Tahap penilaian ini berfokus pada penetapan kriteria awal untuk memprioritaskan beban kerja berisiko rendah dan kompleksitas rendah. Beban kerja ini adalah kandidat yang baik untuk aplikasi percontohan. Menggunakan beban kerja berisiko rendah dan kompleksitas rendah dalam migrasi awal mengurangi risiko dan memberi tim kesempatan untuk mendapatkan pengalaman. Kriteria ini akan dikembangkan dalam tahap penilaian lebih lanjut untuk menyelaraskan prioritas dengan penggerak bisnis saat membuat rencana gelombang migrasi.

Kriteria awal harus memprioritaskan aplikasi dengan sejumlah kecil dependensi, berjalan di infrastruktur yang didukung cloud, dan dari lingkungan non-produksi. Contohnya adalah aplikasi dengan 0-3 dependensi siap direhost apa adanya di lingkungan pengembangan atau pengujian. Kriteria ini berlaku untuk mendefinisikan aplikasi percontohan dan berpotensi gelombang migrasi pertama dan kedua, tergantung pada tingkat kematangan adopsi cloud dan tingkat kepercayaan.

Memutuskan kriteria awal apa yang akan digunakan

Pilih 2—10 titik data yang akan digunakan untuk memprioritaskan beban kerja pertama Anda. Poin data ini berasal dari inventaris aplikasi dan infrastruktur awal Anda (lihat bagian pengumpulan data).

Selanjutnya, tentukan skor, atau bobot, untuk setiap nilai yang mungkin dari setiap titik data. Misalnya, jika atribut lingkungan dipilih, dan nilai yang mungkin adalah produksi, pengembangan, dan pengujian, setiap nilai diberi skor, angka yang lebih besar mewakili prioritas yang lebih tinggi. Meskipun opsional, kami merekomendasikan untuk menetapkan faktor pengganda untuk kepentingan atau relevansi dengan setiap titik data. Langkah opsional ini memberikan pembeda tingkat yang lebih tinggi untuk menekankan apa yang lebih penting, yang membantu menjaga kriteria tetap selaras saat Anda mengulangi penetapan skor ke nilai.

Berdasarkan strategi untuk memprioritaskan aplikasi sederhana dan berisiko rendah untuk beberapa gelombang migrasi pertama, tabel berikut menunjukkan contoh pemilihan atribut dan penetapan nilainya.

Atribut (titik data)	Nilai yang mungkin	Skor (0-99)	Pentingnya atau relevansi faktor pengalikan
Environment	Uji	60	Tinggi (1x)
	Pengembangan	40	
	Produksi	20	
Kekritisan bisnis	Rendah	60	Tinggi (1x)
	Sedang	40	
	Tinggi	20	
Kerangka peraturan atau kepatuhan	Tidak ada	60	Tinggi (1x)
	FedRAMP	10	
Dukungan sistem operasi	Cloud siap	60	Sedang tinggi (0.8x)
	Tidak didukung di cloud	10	
Jumlah instans komputasi	1-3	60	Sedang tinggi (0.8x)
	4-10	40	
	11 atau lebih	20	
Strategi migrasi	Rehost	70	Sedang (0.6x)
	Platform Ulang	30	
	Refactor, atau arsitek ulang	10	

Pastikan Anda memilih atribut yang dapat bertindak sebagai pembeda utama antar aplikasi. Jika tidak, kriteria akan menghasilkan banyak beban kerja yang berbagi prioritas yang sama. Setelah Anda menerapkan model, kami sarankan untuk melihat bagian atas dan bawah peringkat yang

dihasilkan untuk melihat apakah Anda setuju. Jika Anda umumnya tidak setuju, Anda dapat meninjau kembali kriteria yang Anda gunakan untuk menilai beban kerja.

Setelah Anda mendapatkan peringkat, lihat distribusi skor di seluruh portofolio. Skor itu sendiri tidak masalah. Ini adalah perbedaan antara skor yang penting. Misalnya, Anda mungkin menemukan bahwa skor total teratas adalah 8.000 dan skor terbawah adalah 800. Pertimbangkan untuk memplot skor yang dihasilkan sebagai histogram, sehingga Anda dapat memverifikasi bahwa Anda memiliki distribusi yang baik. Distribusi yang ideal terlihat seperti kurva lonceng standar, dengan beberapa beban kerja prioritas sangat tinggi dan beberapa beban kerja dengan prioritas sangat rendah. Sebagian besar aplikasi akan berada di suatu tempat di tengah.

Aspek kunci lain dari prioritas awal adalah memasukkan tim internal atau unit bisnis yang menunjukkan minat untuk menjadi pengadopsi awal cloud. Ini bisa menjadi tuas yang cukup besar dalam memperoleh dukungan bisnis untuk memigrasikan aplikasi tertentu, terutama di masa-masa awal. Jika ini terjadi di organisasi Anda, sertakan atribut unit bisnis di tabel sebelumnya. Tetapkan skor tinggi untuk unit-unit bisnis yang bersedia untuk maju dengan aplikasi mereka. Menggunakan atribut unit bisnis akan membantu membawa aplikasi tersebut ke bagian atas daftar.

Setelah Anda setuju dengan peringkat yang dihasilkan, pilih 5-10 aplikasi teratas. Ini akan menjadi kandidat migrasi aplikasi awal Anda. Perbaiki daftar sehingga Anda mengonfirmasi 3-5 aplikasi. Ini membantu Anda mengambil pendekatan yang ditargetkan saat melakukan penilaian aplikasi terperinci. Untuk informasi selengkapnya, lihat Penilaian aplikasi yang diprioritaskan.

Menentukan tipe R untuk migrasi

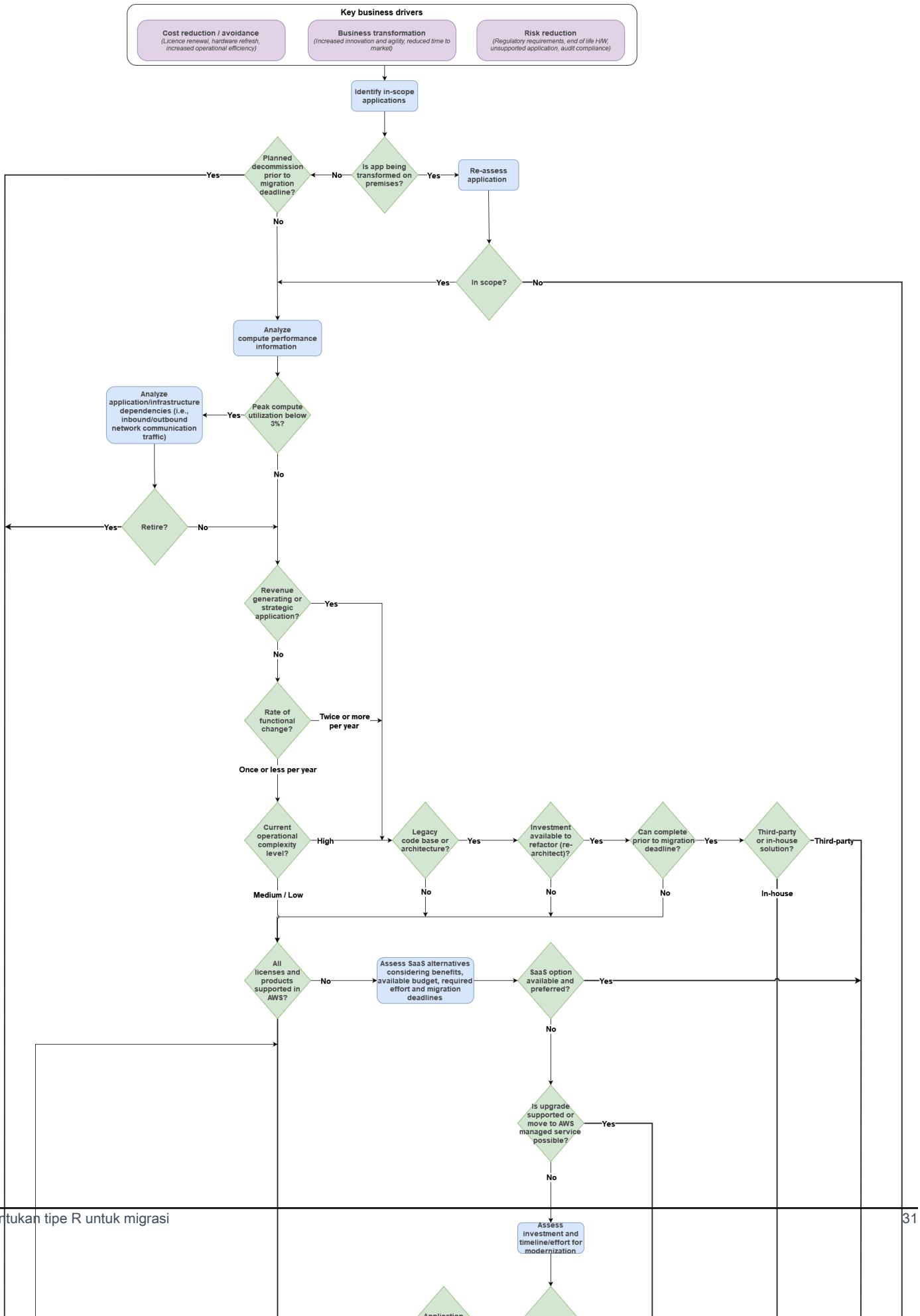
Memutuskan strategi migrasi untuk setiap aplikasi dan infrastruktur terkait akan berimplikasi pada kecepatan migrasi, biaya, dan tingkat manfaat. Ini adalah kunci untuk menentukan strategi berdasarkan kombinasi faktor yang seimbang, termasuk pendorong bisnis, prinsip panduan teknis, kriteria prioritas, dan strategi bisnis.

Terkadang faktor-faktor ini menciptakan pandangan yang saling bertentangan. Misalnya, pendorong utama migrasi mungkin inovasi dan kelincahan. Pada saat yang sama, Anda mungkin perlu mengurangi biaya dengan cepat. Memodernisasi semua aplikasi dalam lingkup akan mengurangi biaya dalam jangka panjang, tetapi akan membutuhkan investasi yang lebih besar di muka. Dalam hal ini, salah satu pendekatannya adalah memigrasikan aplikasi dengan menggunakan strategi yang membutuhkan lebih sedikit usaha, seperti rehost atau replatform. Itu dapat memberikan efisiensi cepat dan pengurangan biaya dalam jangka pendek. Kemudian investasikan kembali tabungan untuk memodernisasi aplikasi pada tahap selanjutnya, dan mencapai pengurangan biaya lebih lanjut.

Namun, dimulai dengan rehost lengkap dari semua aplikasi menunda manfaat modernisasi yang lebih besar. Kuncinya adalah menemukan keseimbangan antara strategi migrasi sehingga aplikasi strategis bisnis diprioritaskan untuk modernisasi sementara aplikasi lain dapat di-host ulang atau direplatform terlebih dahulu kemudian dimodernisasi.

Bagaimana cara menentukan strategi migrasi untuk aplikasi Anda?

Pada tahap penilaian ini, fokusnya adalah untuk menggabungkan model awal untuk memandu pemilihan strategi migrasi. Untuk memvalidasi strategi migrasi untuk aplikasi awal, gunakan model bersama dengan driver bisnis dan kriteria prioritas. Logika default dari pohon keputusan akan membantu Anda menentukan perlakuan awal untuk ruang lingkup. Di pohon, pendekatan yang paling kompleks, seperti refactor, atau arsitek ulang, disediakan untuk beban kerja strategis Anda.


Key business drivers
Cost reduction / avoidance
(Licence renewal, hardware refresh, increased operational efficiency)
Business transformation
(Increased innovation and agility, reduced time to market)
Risk reduction
(Regulatory requirements, end of life H/W, unsupported application, audit compliance)
Identify in-scope applications
Is app being transformed on premises?
Yes
Re-assess application
No
Planned decommission prior to migration deadline?
Yes
No
In scope?
Yes
No
Analyze compute performance information
Peak compute utilization below 3%?
Yes
Analyze application/infrastructure dependencies (i.e., inbound/outbound network communication traffic)
No
Retire?
Yes
No
Revenue generating or strategic application?
Yes
No
Rate of functional change?
Twice or more per year
Once or less per year
Current operational complexity level?
High
Medium / Low
Legacy code base or architecture?
Yes
No
Investment available to refactor (re-architect)?
Yes
No
Can complete prior to migration deadline?
Yes
No
Third-party or in-house solution?
Third-party
In-house
All licenses and products supported in AWS?
No
Assess SaaS alternatives considering benefits, available budget, required effort and migration deadlines
SaaS option available and preferred?
Yes
No
Is upgrade supported or move to AWS managed service possible?
Yes
No
Assess investment and timeline/effort for modernization

Versi draw.io yang dapat disesuaikan dari diagram ini tersedia di bagian Lampiran.

Langkah pertama untuk model awal adalah memperbarui driver bisnis di bagian atas pohon dengan yang ditentukan oleh organisasi Anda. Selanjutnya, terapkan pohon ke komponen aplikasi daripada aplikasi secara keseluruhan. Misalnya, dalam kasus aplikasi tiga tingkat yang memiliki tiga komponen (front-end, lapisan aplikasi, dan database), setiap komponen harus transit pohon secara independen dan diberi strategi dan pola tertentu. Ini karena dalam beberapa kasus Anda mungkin ingin meng-host ulang atau memplatform ulang tingkat tertentu dan refactor (arsitek ulang) tingkatan lain.

Penugasan komponen independen akan mengarahkan Anda untuk menentukan strategi migrasi untuk infrastruktur terkait. Strategi infrastruktur mungkin strategi yang sama dengan komponen aplikasi yang didukungnya, atau mungkin berbeda. Misalnya, komponen aplikasi yang akan direplatform menjadi mesin virtual baru dengan sistem operasi yang lebih baru akan mengikuti strategi replatform sementara mesin virtual saat ini yang menghostingnya akan dihentikan. Strategi migrasi untuk infrastruktur dihitung berdasarkan strategi yang dipilih untuk komponen aplikasi.

Sebelum menggunakan pohon keputusan untuk membuat strategi migrasi, uji logika dengan beberapa aplikasi dan lihat apakah Anda umumnya setuju dengan hasilnya. Pohon keputusan 6 Rs adalah panduan yang tidak menggantikan analisis yang diperlukan untuk menentukan kebenarannya. Logika pohon mungkin tidak berlaku untuk kasus tertentu. Perlakukan kasus-kasus tersebut sebagai pengecualian dan lanjutkan untuk mengganti keputusan yang didorong oleh pohon dengan mendokumentasikan alasan untuk penggantian daripada mengubah logika pohon. Ini mencegah beberapa versi pohon keputusan, yang bisa menjadi sulit untuk dikelola. Panduan umum adalah bahwa pohon harus berlaku untuk setidaknya 70-80 persen dari beban kerja. Selebihnya, akan ada pengecualian. Setiap penyesuaian pada logika pohon, pada tahap penilaian ini, harus difokuskan pada pembentukan model awal. Iterasi dan penyempurnaan lebih lanjut akan terjadi pada tahap selanjutnya, seperti analisis portofolio dan perencanaan migrasi.

Lampiran

attachment.zip

Membuat kasus bisnis terarah

Pemangku kepentingan dari seluruh bisnis harus memahami dan membeli ke dalam kasus bisnis untuk transformasi setiap langkah di sepanjang jalan.

Pada tahap awal, penting untuk dengan cepat menunjukkan nilai potensial yang cukup dari program migrasi, sehingga Anda dapat mengamankan sumber daya yang dibutuhkan untuk merencanakan

dan membuat program. Kasus bisnis terarah dirancang untuk memberikan kepercayaan yang masuk akal dalam mencapai nilai bisnis yang menarik dengan data terbatas yang dapat dikumpulkan lebih awal.

Setelah program ditetapkan, kasus bisnis dikembangkan lebih lanjut. Kasus terperinci memberikan akurasi yang lebih besar, gambaran yang lebih lengkap tentang nilai program, dan wawasan tentang prioritas perencanaan. Ini mendefinisikan dan mengukur hasil bisnis yang direncanakan yang dibeli organisasi, dan menetapkan dasar di mana kantor tata kelola program Anda kemudian dapat mengarahkan program dan mengukur pencapaiannya.

Memperbaiki ruang lingkup kasus bisnis terarah

Kasus bisnis terarah biasanya dirakit dengan cepat, dalam 2-4 minggu. Ini perlu menghasilkan kepercayaan diri yang cukup sehingga Anda dapat mengamankan sumber daya untuk membentuk tim inti, melibatkan AWS Mitra jika diperlukan, dan seminimal mungkin, menyelesaikan penilaian aplikasi yang diprioritaskan dan analisis portofolio dan tahap perencanaan migrasi.

Biasanya, kasus bisnis terarah yang mendukung migrasi portofolio dibuat sebagai salah satu dari berikut ini:

- Perbandingan biaya kepemilikan total (TCO) sederhana antara lanskap infrastruktur apa adanya dan arsitektur layanan pasca-migrasi AWS . Perbandingan menunjukkan perbedaan laju lari yang diharapkan untuk volume beban kerja tertentu.
- Kasus bisnis yang menunjukkan nilai sekarang bersih (NPV), laba atas investasi (ROI), periode pengembalian modal, tingkat pengembalian internal yang dimodifikasi (MIRR) dan analisis arus kas 3-5 tahun untuk bermigrasi ke AWS inklusif biaya migrasi vs tetap apa adanya.

Lingkup kasus bisnis terarah biasanya terbatas pada salah satu dari berikut ini:

- Perbandingan biaya teknologi infrastruktur
- Perbandingan teknologi infrastruktur dan biaya operasi

Secara umum, semakin besar portofolio, semakin kurang berkembang kasusnya. Ini karena asumsi yang lebih luas dapat dibuat tanpa mempengaruhi hasilnya secara signifikan. Untuk portofolio yang lebih kecil, setiap perubahan akan memiliki dampak yang lebih besar, sehingga diperlukan lebih banyak detail.

Mulailah dengan membangun perbandingan biaya infrastruktur dasar. Kemudian putuskan apakah perbandingannya cukup menarik sebelum Anda melanjutkan. Biasanya, portofolio lebih dari 400 server akan menunjukkan kasus bisnis yang positif pada pengurangan biaya infrastruktur saja dalam waktu 3 tahun beroperasi AWS, atau 250 server dalam 5 tahun, meskipun ini dapat bervariasi. Untuk portofolio yang lebih kecil, detail lebih lanjut mungkin diperlukan.

Sebaliknya, jarang berguna untuk memeriksa komponen nilai bisnis lainnya pada tahap ini, seperti nilai yang berasal dari peningkatan ketahanan atau kelincahan bisnis, kecuali total ruang lingkup migrasi kurang dari sekitar 5 beban kerja atau 50 server.

Driver nilai fokus

Perbandingan TCO teknologi infrastruktur membandingkan model biaya infrastruktur apa adanya dengan model dasar tagihan AWS layanan bahan yang dibutuhkan untuk menjalankan beban kerja Anda dengan kinerja dan ketersediaan yang setara. Banyak optimasi dapat dilakukan. Pada tahap ini, bagaimanapun, fokusnya adalah pada daftar berikut karena mereka lebih mudah untuk menilai dan biasanya menghasilkan sekitar 30 persen penghematan TCO, yang cukup untuk bergerak maju:

- Compute elastisitas — Server peta yang penggunaannya tidak 100 persen, seperti server pengembangan atau UAT yang menjalankan 8x5 (penggunaan 24 persen), 10x5 (30 persen), atau 10x6 (36 persen), dan server pemulihan bencana (DR) yang berjalan pada 2 persen, untuk layanan sesuai permintaan yang hanya ditagih saat digunakan.
- Pengadaan dengan rencana penghematan - Rencanakan untuk mendapatkan server produksi dan server lain dengan penggunaan tinggi (lebih dari 36 persen) dengan rencana penghematan yang sesuai untuk mengurangi biaya hingga 75 persen. Opsi termasuk komitmen 1 tahun dan 3 tahun, dengan tingkat pembayaran di muka yang berbeda untuk mendapatkan diskon yang lebih besar.
- Hapus zombie — Identifikasi server dengan pemanfaatan CPU kurang dari 2 persen yang dapat Anda konfirmasi tidak lagi diperlukan, dan hapus dari analisis biaya.
- Hitung ukuran kanan — Gunakan data deret waktu pemanfaatan CPU dan memori untuk menilai setiap server daya komputasi dan memori yang dibutuhkan. Kemudian pilih instans Amazon Elastic Compute Cloud (Amazon EC2) agar sesuai.
- Sistem manajemen basis data relasional (RDBMS) lisensi ukuran kanan - Menilai kembali kebutuhan lisensi RDBMS Anda setelah menghitung ukuran kanan pada server database Anda, bandingkan Bring Your Own License (BYOL) dan lisensi Pengadaan dari, dan jelajahi AWS potensi Amazon Relational Database Service (Amazon RDS) untuk meningkatkan penghematan.
- Penyimpanan — Ukuran yang tepat total volume penyimpanan yang dibutuhkan, dan mengidentifikasi kebutuhan operasi input/output per detik (IOPS) di seluruh portofolio. Tentukan

berapa banyak yang dapat dipindahkan ke penyimpanan objek dengan SLA dan biaya yang berbeda.

Kebutuhan data

Tabel dalam Memahami persyaratan data penilaian awal menunjukkan data yang diperlukan untuk membangun setiap bagian dari kasus bisnis terarah, dan apakah itu wajib atau opsional.

Untuk membangun kasus ini, Anda memerlukan subset infrastruktur dari data perencanaan awal ditambah data biaya. Menentukan cara mengidentifikasi infrastruktur yang akan disertakan tergantung pada tujuan bisnis Anda:

- Jika tujuan program adalah untuk memigrasi dan memodernisasi aplikasi tertentu, bangun portofolio infrastruktur berdasarkan apa yang dibutuhkan aplikasi, dengan mempertimbangkan infrastruktur yang dibagikan.
- Jika tujuan program adalah infrastruktur-sentris, seperti migrasi keluar dari pusat data yang sewa akan kedaluwarsa, pemetaan aplikasi tidak diperlukan untuk perbandingan TCO infrastruktur.

Data yang ditandai sebagai opsional (seperti CPU dan pemanfaatan puncak memori untuk server) biasanya dapat diganti dengan nilai benchmark standar. Anda dapat mendiskusikan hal ini dengan AWS Mitra atau Layanan AWS Profesional. Atau Anda dapat memperkirakan nilai dari titik data yang tersedia di bagian portofolio Anda (seperti data yang dikumpulkan oleh hypervisor). Semakin besar portofolio, semakin akurat ini.

Membangun infrastruktur perbandingan TCO

Alat sangat penting untuk membangun perbandingan TCO infrastruktur. AWS Layanan Profesional atau AWS Mitra dapat memberikan bantuan dengan semua jenis kasus terarah, terutama jika Anda berencana untuk melibatkan mereka untuk membantu dalam proses migrasi yang lebih luas.

Ada alat yang tersedia untuk melakukan hal berikut:

- Kumpulkan data inventaris.
- Kumpulkan data pemanfaatan.
- Menyediakan data benchmarking biaya infrastruktur yang akurat.
- Identifikasi dan hapus zombie.
- Buat penilaian ukuran yang tepat.

- Merekomendasikan opsi pembelian.
- Bandingkan opsi lisensi perangkat lunak.
- Menghasilkan analisis arus kas grafis sederhana.

Migration Evaluator dari AWS adalah salah satu pilihan. Ini menyediakan semua kemampuan ini sebagai layanan terkelola gratis. Anda dapat meminta Penilai Migrasi melalui manajer AWS akun atau Mitra Kompetensi AWS Migrasi atau dengan mengirimkan permintaan secara online. Migration Evaluator telah dirancang khusus sebagai solusi titik untuk menghasilkan perbandingan TCO teknologi infrastruktur dan cepat.

Keuntungan utama:

- Bebas biaya
- Penemuan tanpa agen atau konfigurasi manual data inventaris di mana penemuan berbasis alat dibatasi
- Dukungan khusus untuk membantu penyebaran, konfigurasi, pengumpulan data, dan membangun kasus dasar, atau kasus bisnis terarah
- Kenyamanan operasi SaaS, tetapi dapat menjalankan pengumpulan data sepenuhnya dalam jaringan pelanggan untuk mendukung scrubbing sebelum memuat ke mesin analitik
- Dukungan kuat untuk ukuran kanan lisensi Microsoft
- Kemampuan ekspor data lengkap

Keterbatasan utama:

- Menilai hanya server arsitektur x86 (Windows dan Linux)
- Opsi terbatas untuk mengonfigurasi atau mengkalibrasi data biaya apa adanya benchmark
- Tidak ada dukungan untuk pengoptimalan biaya operasi pemodelan
- Tidak ada dukungan untuk pemodelan biaya migrasi
- Tidak ada dukungan langsung untuk membangun kasus bisnis di luar perbandingan TCO

Jika Anda memutuskan untuk menggunakan alat penemuan komersial untuk penemuan portofolio dan kemampuan analisis seperti tumpukan aplikasi dan penemuan interdependensi, biasanya akan memberikan perbandingan TCO infrastruktur juga. Untuk panduan tentang penggunaan alat untuk penemuan dan penilaian portofolio, lihat Mengevaluasi kebutuhan alat penemuan.

Membangun optimalisasi biaya operasional

Peningkatan produktivitas operasi TI seringkali merupakan kontributor nilai yang signifikan untuk migrasi. Rata-rata, setelah migrasi ke AWS, produktivitas staf operasional TI meningkat sebesar 62 persen melalui migrasi, menurut whitepaper International Data Corporation (IDC) Fostering Business and Organizational Transformation to Generate Business Value with Amazon Web Services. Namun, ada dua tantangan dengan ukuran dan termasuk manfaat ini dalam kasus terarah.

Pertama, menilai berbagai keuntungan produktivitas membutuhkan pengumpulan data yang ekstensif dan lebih tepat untuk kasus bisnis yang terperinci. Tantangan ini dapat diatasi dengan berfokus pada beberapa elemen yang lebih mudah dinilai dan diukur dengan data benchmark sederhana tetapi masih menunjukkan keuntungan yang signifikan.

Kedua, berfokus pada produktivitas sebagai sumber pengurangan biaya dapat menimbulkan kekhawatiran dan negativitas di antara pemangku kepentingan pelanggan utama dan anggota program. Pastikan bahwa Anda memberikan kejelasan tentang bagaimana manfaat akan direalisasikan dan apa artinya bagi orang-orang yang terkena dampak. Masalah seperti itu dapat dihindari dengan mengklarifikasi bahwa ini hanya akan meningkatkan peran tim:

- Program migrasi mencakup jalur untuk mengembangkan dan memindahkan staf operasi internal ke peran baru, seperti bergabung dengan DevSecOps tim yang membangun infrastruktur sebagai otomatisasi kode dan otomatisasi pengujian yang akan mendorong pertumbuhan tim.
- Manfaat dapat direalisasikan dengan mengubah dan mengubah ukuran kontrak outsourcing operasi, sehingga staf internal dapat meningkatkan fokus mereka pada kegiatan yang bernilai lebih tinggi

Pendekatan membangun elemen kasus bisnis ini berdasarkan transformasi operasi apa yang ingin Anda pertimbangkan:

- Jika Anda memiliki tim operasi internal yang ada, tingkatkan keterampilan anggota tim, dan tunjukkan peningkatan produktivitas yang diharapkan.
- Atau, bermigrasi dari solusi operasi Anda saat ini ke AWS Managed Services (AMS) atau ke penawaran layanan terkelola alternatif dari AWS Partner.

Untuk transformasi pertama, untuk mendapatkan perkiraan keuangan konservatif dari peningkatan produktivitas yang dapat dimasukkan dalam kasus ini, kami merekomendasikan yang berikut:

1. Fokus pada produktivitas operasi manajemen server secara khusus. Ini cenderung menjadi proporsi yang signifikan dari upaya operasi, dapat lebih mudah dinilai, dan lebih mudah diverifikasi nanti.
2. Hitung kepegawaian yang dibutuhkan berdasarkan tolok ukur jumlah server yang dapat dikelola oleh setiap karyawan full-time equivalent (FTE). Di tempat, jumlah itu sekitar 150 severs. Pada AWS, itu sekitar 400 server.
3. Terapkan metrik ini ke jumlah server lokal dibandingkan dengan jumlah instans EC2.
4. Lipat gandakan waktu yang dihemat dengan tingkat biaya campuran untuk seluruh tim operasi.

Anda kemudian dapat memeriksa hasil Anda dengan salah satu pendekatan dengan memverifikasi hasilnya tidak jauh melebihi perolehan produktivitas rata-rata berdasarkan peran yang disediakan dalam tabel berikut (data yang bersumber dari whitepaper IDC Mendorong Bisnis dan Transformasi Organisasi untuk Menghasilkan Nilai Bisnis dengan Amazon Web Services.

Peran	Keuntungan Efisiensi
Manajemen infrastruktur TI	62%
Dukungan TI	59%
Manajemen aplikasi	43%
Pengelolaan basis data	19%
Pengembangan aplikasi	25%

Untuk transformasi kedua, Anda dapat menambahkan penghematan biaya operasional dengan langsung membandingkan total operasi saat ini dan biaya dukungan untuk portofolio dalam ruang lingkup dengan biaya layanan terkelola yang dipertimbangkan.

Untuk mendapatkan biaya layanan terkelola, berikan kepada manajer AWS akun atau AWS Managed Services Mitra Anda dengan AWS Bill of Material yang Anda usulkan, pilihan tingkat layanan Anda (Plus atau Premium), dan paket AMS Anda (AMS Accelerate atau AMS Advanced). Ini akan memberi Anda total biaya layanan terkelola untuk komponen AWS layanan dari solusi yang diubah. Demikian pula, Anda dapat memperoleh harga dari AWS Mitra yang menawarkan paket layanan terkelola sendiri berdasarkan parameternya sendiri.

Memperluas ke kasus bisnis terarah penuh

Secara umum, untuk merakit kasus bisnis terarah penuh, membangun perbandingan TCO, dengan atau tanpa elemen produktivitas TI, dan perkirakan semua biaya migrasi dan modernisasi. Kemudian buat arus kas yang mencakup pasangan t-migrate-and-modernize skenario migrate-and-modernize dan jangan.

Kasus yang paling mendasar adalah persiapan sepasang skenario, di mana t-migrate-and-modernize skenario jangan adalah situasi Anda saat ini dan migrate-and-modernize skenario memiliki karakteristik sebagai berikut:

- Tidak ada pertumbuhan atau penyusutan volume transaksional, komputasi, atau kapasitas jaringan
- Pertumbuhan volume rendah yang stabil dalam persyaratan penyimpanan
- uality-of-service Kemampuan Q (seperti ketersediaan, daya tahan, throughput, dan kinerja) yang sesuai dengan kemampuan sistem yang ada

Untuk semua kecuali portofolio yang sangat kecil, ini sesuai dengan tujuan membangun kasus terarah dengan baik. Ini menunjukkan nilai yang cukup cepat untuk mendapatkan mandat untuk bergerak maju.

Untuk portofolio yang lebih kecil, akan bermanfaat untuk menambahkan pasangan t-migrate-and-modernize skenario migrate-and-modernize dan jangan yang menunjukkan aspek lain dari peningkatan nilai migrasi cloud, seperti:

- Campuran persyaratan pertumbuhan kapasitas sedang dan tinggi di seluruh beban kerja di mana pertumbuhan itu diharapkan
- Dimasukkannya ketahanan yang ditingkatkan, seperti ketersediaan tinggi, DR, dan toleransi kesalahan
- Peningkatan kinerja global dengan komputasi tepi, jaringan pengiriman konten (CDN), replikasi database multi-wilayah.
- Kualitas layanan spesifik lainnya yang telah Anda jadikan prioritas bisnis untuk program ini

Untuk skenario ini, pastikan bahwa biaya dan implikasi arus kas dari peningkatan arsitektur infrastruktur non-cloud saat ini agar sesuai dengan spesifikasi baru diperkirakan secara akurat. Cara paling langsung untuk mendapatkan perkiraan ini dapat meminta kutipan dari integrator sistem, terutama jika mereka juga merupakan Mitra AWS Konsultasi dengan Kompetensi Migrasi, yang dapat mendukung Anda dengan skenario migrate-and-modernize dan tidak. t-migrate-and-modernize

Untuk setiap pasangan skenario, kumpulkan kasus yang terdiri dari hal-hal berikut:

- Biaya t-migrate-and-modernize skenario don '. Dalam kasus yang paling mendasar, ini termasuk:
  - Total biaya kepemilikan selama jangka waktu kasus bisnis untuk konfigurasi infrastruktur saat ini
  - Peningkatan berkala dalam komputasi, penyimpanan, dan konsumsi lalu lintas jaringan
- Biaya skenario migrate-and-modernize;, termasuk:
  - Menyiapkan program, yang mencakup penemuan terperinci, perencanaan migrasi, pengembangan kasus bisnis terperinci, pembentukan tim inti dan peningkatan keterampilan mereka, membangun landing zone jika belum ada, dan membangun manajemen keamanan dan integrasi operasi untuk beban kerja yang bermigrasi
  - Biaya migrasi dan modernisasi beban kerja
  - Biaya infrastruktur migrasi, termasuk koneksi jaringan, layanan migrasi data seperti AWS Snowball dan AWS DataSync, dan biaya AWS utilitas untuk arsitektur yang diperlukan selama proses migrasi itu sendiri (misalnya, untuk pengujian)
  - Peningkatan biaya AWS utilitas selama migrasi saat gelombang ditayangkan, dan menurunnya biaya infrastruktur yang ada karena digantikan oleh layanan AWS berbasis dan dinonaktifkan

Biaya penonaktifan dan penghapusan untuk setiap aset yang terdampar

Memperkirakan pengaturan program migrasi dan modernisasi

Untuk menyiapkan program untuk sukses, Anda mungkin perlu menjalankan serangkaian kegiatan dasar untuk membangun kemampuan dasar dan rencana terperinci jika ini belum dilakukan sebelumnya. Kegiatan dasar ini meliputi:

1. Melakukan penemuan portofolio terperinci, perencanaan migrasi, dan pengembangan kasus bisnis terperinci, seperti yang dijelaskan di bagian Analisis portofolio dan perencanaan migrasi, ditambah pendokumentasian biaya alat penemuan apa pun yang digunakan.
2. Membangun bisnis cloud dan tim inti teknis dan mengembangkan keterampilan internal melalui pelatihan dan perekrutan. Identifikasi anggota organisasi TI yang akan membutuhkan pelatihan, dan alokasikan anggaran pelatihan untuk setiap orang.
3. Menetapkan landing zone dan mengonfigurasinya untuk mendukung kemampuan tata kelola biaya, operasional, dan keamanan yang Anda perlukan.

AWS Mitra Konsultasi dapat membantu memberikan perkiraan untuk item 1 dan 3.

Memperkirakan biaya migrasi dan modernisasi

Untuk memenuhi tujuan kasus bisnis terarah dan menunjukkan potensi komersial yang cukup untuk melanjutkan ke fase berikutnya, pertahankan estimasi biaya migrasi dan modernisasi sedasar mungkin.

Untuk tujuan ini, kami menyarankan Anda menyiapkan kasus bisnis terarah dengan berfokus pada aplikasi yang termasuk dalam strategi migrasi berikut:

- Pensiun
- Mempertahankan
- Pindah
- Rehost
- Platform Ulang
- Pembelian kembali

Biasanya, sekitar 70 persen beban kerja dapat direhost, dipindahkan atau direplatform, dan 5 persen lainnya dapat dihentikan. Menilai aplikasi dengan strategi migrasi biasanya membahas inti dari kasus pengurangan biaya.

Memperkirakan biaya untuk refactoring, atau re-architecting, bisa jadi rumit. Hal ini tidak praktis untuk mencoba ini dalam jangka waktu yang diberikan untuk mempersiapkan kasus bisnis terarah. Seperti yang dibahas sebelumnya dalam Menentukan tipe R untuk migrasi, pertimbangkan untuk menggunakan strategi rehost, relokasi, atau replatform untuk fase pertama migrasi dan modernisasi Anda. Strategi R ini kemungkinan akan mempercepat pengembalian awal, mengurangi risiko implementasi, dan meningkatkan kasus bisnis dalam jangka pendek. Hal ini juga secara material lebih mudah bagi tim aplikasi Anda untuk memodernisasi aplikasi yang berjalan dalam AWS lingkungan daripada yang tidak. Perkiraan untuk refactoring (re-architecting) aplikasi spesifik paling baik ditambahkan ketika kasus bisnis terperinci disiapkan.

Memperkirakan upaya migrasi dengan strategi

Setiap migrasi berbeda. Sebelum melakukan anggaran atau rencana apa pun, perkiraan beban kerja benih untuk aktivitas migrasi dari tim yang akan bertanggung jawab atas proyek, apakah itu tim aplikasi internal Anda, Layanan AWS Profesional, atau organisasi Mitra. AWS

Untuk membantu membangun kasus terarah, tabel berikut memberikan rentang upaya indikatif untuk perawatan yang berbeda. Rentang ini mengasumsikan bahwa medium-to-large portofolio

sedang dimigrasikan dan tim migrasi dilatih dan berpengalaman. Untuk portofolio kecil, yang terbaik adalah meminta tim yang bertanggung jawab atas migrasi menyiapkan perkiraan bahkan untuk kasus terarah.

Strategi migrasi	Proses estimasi	Elemen	Jam orang	Jam orang
Mempertahankan	Jangan melakukan apa-apa, tanpa biaya, tanpa manfaat, dan tidak ada pengurangan hutang teknologi.	–	–	–
Pensiun	Perkiraan penonaktifan peralatan perangkat keras yang digunakan, jika ada.	–	–	–
Pindah	Perkiraan menyalin beban kerja dalam VMware menggunakan alat VMware. Ini termasuk menyalin data, pengujian asap untuk memverifikasi, dan penonaktifan perangkat keras apa pun. Upaya untuk merelokasi VM biasanya	–	–	–

Strategi migrasi	Proses estimasi	Elemen	Jam orang	Jam orang
	kurang dari pola rehost dengan kompleksitas rendah.			

Strategi migrasi	Proses estimasi	Elemen	Jam orang	Jam orang
Rehost	Perkirakan penyalinan beban kerja dan data dengan salinan gambar, pengujian asap, pengujian ketersediaan tinggi (HA) dan pemulihan bencana (DR) jika sesuai untuk server produksi, dan penonaktifan perangkat keras apa pun. Praktik terbaik adalah menggunak an alat seperti Layanan Migrasi AWS Aplikasi. Bagilah beban kerja menjadi kompleksitas rendah, sedang, dan tinggi, berdasarkan faktor-faktor seperti apakah database atau perangkat lunak infrastruktur lainnya berjalan, kompleksi	Upaya per aplikasi per server	Migrasi:	Tes HA/DR
		Rendah	10—14	3—5
		Sedang	16—24	4—6
		Tinggi	26—38	8—12

Strategi migrasi	Proses estimasi	Elemen	Jam orang	Jam orang
	tas basis data, apakah dikelompokkan, kompleksitas integrasi, dan volume data.			

Strategi migrasi	Proses estimasi	Elemen	Jam orang	Jam orang
Platform Ulang	Untuk migrasi replatform yang mencakup peningkatan ke sistem operasi atau versi RDBMS, ambil perkiraan untuk rehost, dan tambahkan waktu untuk menjalankan uji pembangunan kembali dan asap pada platform baru. Jika replatform termasuk mengubah teknologi platform, perkirakan waktu tambahan untuk penggunaan alat konversi, seperti dan, dan pengujian aplikasi yang lebih lengkap. AWS Schema Conversion ToolAWS Database Migration	Upaya per aplikasi per server	Versi naik	Perubahan teknologi
		Rendah	Tambahkan 1-3	Tambahkan 10—15
		Sedang	Tambahkan 2—5	Tambahkan 20—30
		Tinggi	Tambahkan 4-8	Tambahkan 40-60

Strategi migrasi	Proses estimasi	Elemen	Jam orang	Jam orang
	Service Contoh perubahan teknologi adalah bermigrasi dari database komersial berpemilik ke pengganti open source.			
Pembelian kembali	Perkirakan ekstraksi data, transformasi, dan pengungga han ke dalam penggantian layanan SaaS yang baru dibeli, dan penonaktifan perangkat keras apa pun.	–	–	–

Memperkirakan biaya infrastruktur migrasi

Sertakan perkiraan untuk infrastruktur yang akan Anda gunakan selama migrasi. Biasanya, perkiraan ini terdiri dari:

- Anggaran untuk konektivitas dan layanan pertukaran data untuk beban kerja dan migrasi data dari lingkungan saat ini ke AWS
- Anggaran untuk AWS layanan (terutama komputasi dan penyimpanan) yang diperlukan untuk menghosting beban kerja yang dimigrasi selama proses migrasi, pengujian, dan pemotongan
- Peningkatan biaya AWS utilitas karena setiap gelombang migrasi selesai
- Biaya penonaktifan infrastruktur yang ada yang tidak akan lagi menjalankan beban kerja yang dimigrasi

Untuk pertukaran data, periksa total volume data Anda dan nilai kelayakan menggunakan jaringan. Jika Anda telah menyediakan AWS Direct Connecttautan atau AWS VPNdari AWS satu titik di WAN Anda sebelumnya untuk penggunaan operasional setelah migrasi, Anda dapat menggunakan sumber daya tersebut hingga kuota layanannya.

Jika kapasitas jaringan Anda tidak mencukupi, peningkatan bandwidth internet jangka pendek dengan jaringan pribadi virtual (VPN) seringkali merupakan solusi yang sangat hemat biaya. Jika tidak, perangkat pertukaran AWS media seperti AWS Snowball dan AWS Snowcone menawarkan solusi di sebagian besar Wilayah. AWS Juga, untuk migrasi data volume sangat tinggi, pertimbangkan untuk memasukkan anggaran untuk AWS DataSync, yang meningkatkan keandalan dan dapat mempercepat transfer terlepas dari media yang digunakan.

Memodelkan peningkatan AWS layanan dan menuruni infrastruktur yang ada penting untuk elemen analisis arus kas dari kasus bisnis. Pada tahap ini, Anda tidak mungkin memiliki rencana gelombang untuk menentukan dengan tepat kapan biaya akan dikeluarkan. Sebaiknya lakukan hal berikut:

- Meningkatkan biaya untuk AWS pada tingkat konstan selama migrasi.
- Mengurangi biaya untuk infrastruktur yang ada yang Anda rencanakan untuk dinonaktifkan pada tingkat konstan selama durasi yang sama.

Memulai kenaikan AWS biaya 1-2 bulan sebelum infrastruktur yang ada turun. Ini menyediakan 1 bulan penggunaan AWS utilitas untuk melakukan migrasi untuk setiap gelombang. Ini termasuk waktu untuk pengujian, dan waktu tambahan untuk menyelesaikan pekerjaan penonaktifan yang diperlukan untuk menghentikan biaya pada infrastruktur yang diganti.

Memperkirakan biaya penonaktifan

Menonaktifkan peralatan yang tidak dapat dipindahkan, dan membuangnya dengan cara yang legal dan ramah lingkungan, dapat menimbulkan biaya kecil. Namun, untuk kasus bisnis terarah, biasanya satu-satunya jumlah material yang berpotensi adalah biaya penghapusan nilai buku yang tersisa dari aset yang diganti.

Untuk kasus bisnis terarah, kami sarankan Anda melakukan hal berikut:

- Tinjau daftar aset Anda.
- Identifikasi mereka yang akan dinonaktifkan.
- Untuk mengurangi penghapusan, periksa peluang untuk beralih perangkat sehingga perangkat yang lebih baru dalam daftar dapat digunakan untuk menggantikan aset yang lebih tua dan lebih terdepresiasi sepenuhnya.

- Buat penilaian nilai buku future dari aset yang akan dinonaktifkan pada saat itu.
- Sertakan ini sebagai biaya migrasi penonaktifan.

Merakit dan menyesuaikan kasus bisnis terarah penuh

Setelah Anda menyiapkan set lengkap biaya untuk setiap pasangan skenario, buatlah laporan arus kas diskon untuk masing-masing dan buat grafik. Kami merekomendasikan untuk membangun kasus bisnis terarah selama periode yang sama dengan siklus penyegaran perangkat keras. Ini biasanya 5 tahun untuk server, penyimpanan, dan perangkat jaringan. Saat Anda menggunakan periode yang sama dengan siklus penyegaran perangkat keras, biaya tepat satu penyegaran disertakan dalam biaya apa adanya untuk setiap skenario.

Kemudian hitung metrik keuangan utama yang Anda butuhkan untuk mendapatkan persetujuan untuk pindah ke fase berikutnya dari program. Kami biasanya menyertakan yang berikut:

- Net present value (NPV) untuk mengukur nilai absolut dari pengurangan biaya dan keuntungan produktivitas yang dinilai
- Periode pengembalian modal dalam beberapa bulan untuk memverifikasi bahwa pengembalian cukup cepat
- Perbandingan run-rate akhir untuk memverifikasi apakah proses tersebut mengeluarkan biaya yang cukup selama jangka waktu tersebut
- Pengembalian investasi (ROI) dan tingkat pengembalian investasi yang dimodifikasi (MIRR) untuk menilai kinerja keuangan relatif program di atas tuntutan modal lain yang mungkin diprioritaskan organisasi Anda.

Gunakan iterasi pertama kasus untuk menentukan apakah kinerja keuangan yang diharapkan berarti bahwa penyempurnaan harus dilakukan, seperti dalam contoh berikut:

- Jika pengembalian dana terlalu lambat, pertimbangkan opsi untuk mempercepat dan mengurangi biaya migrasi, seperti berikut ini:
  - Gunakan AWS Mitra atau Layanan AWS Profesional untuk memperluas sumber daya yang tersedia dan selanjutnya memparalelkan beban kerja migrasi dengan pola yang lebih mendasar.
  - Untuk beban kerja yang berjalan di VMware, bandingkan strategi relokasi dengan strategi rehost atau replatform, setidaknya untuk fase awal. Menggunakan strategi relokasi dapat mengurangi biaya migrasi dan meningkatkan kecepatan migrasi.

    - Jika secara teknis layak, dorong beban kerja yang membutuhkan strategi replatform atau refactor (re-architect) yang lebih kompleks ke fase future, di luar lingkup kasus bisnis awal.
- Jika ROI dan MIRR terlalu rendah, pertimbangkan hal berikut:
    - Apakah skenario yang Anda anggap terlalu konservatif? Apakah Anda memiliki skenario yang mencerminkan pertumbuhan kapasitas dan kebutuhan elastisitas yang paling mungkin? Apakah Anda memiliki skenario yang membandingkan biaya termasuk peningkatan kualitas layanan dalam tujuan Anda?
    - Dapatkah Anda menyempurnakan cakupan portofolio aplikasi yang akan dimigrasikan pada tahap pertama untuk fokus pada beban kerja yang akan menghasilkan pengembalian yang lebih kuat, seperti yang memiliki pemanfaatan arus yang lebih rendah atau kebutuhan pemulihan bencana (DR) yang mahal?
    - Dapatkah menyempurnakan ruang lingkup portofolio aplikasi untuk awalnya mengecualikan beban kerja tertentu yang mencapai kurang komersial? Misalnya, dapatkah Anda menunda beban kerja yang lisensi perangkat lunak pihak ketiga menjadi lebih mahal karena persyaratan yang berbeda untuk penyebaran di infrastruktur cloud publik?
- Jika perbandingan run-rate akhir tidak memenuhi target yang diharapkan, jelajahi hal berikut:
    - Pertama, konfirmasikan bahwa metrik lainnya memenuhi harapan. Kasus bisnis terarah terutama untuk menunjukkan bahwa ada peluang keuangan yang cukup untuk membenarkan memulai fase persiapan migrasi berikutnya.
    - Identifikasi daftar peluang untuk terus meningkatkan kinerja biaya AWS setelah fase awal migrasi.

Sertakan penilaian daftar peluang saat menyiapkan kasus bisnis terperinci. Selain itu, sertakan penilaian peluang dalam pemeliharaan kasus yang sedang berlangsung dan proses month-to-month pengoptimalan biaya setelah migrasi selesai.

Penilaian aplikasi yang diprioritaskan

Salah satu hasil utama dari tahap sebelumnya, penemuan portofolio dan perencanaan awal, adalah memprioritaskan subset aplikasi untuk penilaian terperinci. Bagian ini mengeksplorasi penilaian rinci aplikasi.

Melihat detail beberapa aplikasi sejak dini akan mendorong akselerasi. Proses penilaian dan desain arsitektur calon memunculkan pemblokir potensial dan mengklarifikasi tugas-tugas penting yang merupakan prekursor migrasi dengan cakupan yang lebih besar. Tugas-tugas ini termasuk mengumpulkan persyaratan untuk membangun AWS fondasi, seperti landing zone on AWS, atau untuk memperluas dan memvalidasi landing zone yang ada. Penilaian ini juga merupakan waktu untuk mempertimbangkan langkah-langkah dan strategi migrasi.

Hasil utama dari tahap ini adalah sebagai berikut:

- Daftar aplikasi yang diprioritaskan yang divalidasi
- Arsitektur keadaan saat ini yang didokumentasikan
- Arsitektur target awal yang terdokumentasi dan strategi migrasi untuk kandidat migrasi
- Pola dan perkakas migrasi yang teridentifikasi
- Persyaratan platform yang terdokumentasi (keamanan, AWS infrastruktur, dan operasi)
- Pertimbangan cutover terdokumentasi untuk perencanaan migrasi
- Perkiraan laju AWS lari

Memahami persyaratan data penilaian terperinci

Tabel berikut menjelaskan informasi yang diperlukan untuk mendapatkan tampilan portofolio lengkap dari aplikasi dalam migrasi dan infrastruktur terkait.

Tabel menggunakan singkatan berikut:

- R, untuk yang dibutuhkan
- O, untuk opsional
- N/A, karena tidak berlaku

Aplikasi

Nama atribut	Deskripsi	Strategi Penemuan, Desain, dan Migrasi	Perkiraan Rate Run	Tingkat kesetiaan yang disarankan (minimum)
Pengidentifikasi unik	Misalnya, ID aplikasi. Biasanya tersedia pada CMDB yang ada atau inventaris internal dan sistem kontrol lainnya. Pertimbangkan untuk membuat ID unik setiap kali ini tidak ditentukan dalam organisasi Anda.	R	O	Tinggi
Nama aplikasi	Nama yang dengannya aplikasi ini diketahui organisasi Anda. Sertakan vendor komersial off-the-shelf (COTS) dan nama produk jika berlaku.	R	R	Tinggi
Apakah COTS?	Ya atau Tidak. Apakah ini aplikasi komersial atau	R	R	Tinggi

Nama atribut	Deskripsi	Strategi Penemuan, Desain, dan Migrasi	Perkiraan Rate Run	Tingkat kesetiaan yang disarankan (minimum)
	pengembangan internal			
Produk dan versi COTS	Nama dan versi produk perangkat lunak komersial	R	R	Tinggi
Deskripsi	Fungsi dan konteks aplikasi utama	R	O	Tinggi
Kekritisan	Misalnya, aplikasi strategis atau menghasilkan pendapatan, atau mendukung fungsi kritis	R	O	Tinggi
Tipe	Misalnya, database, manajemen hubungan pelanggan (CRM), aplikasi web, multimedia, layanan bersama TI	R	O	Tinggi

Nama atribut	Deskripsi	Strategi Penemuan, Desain, dan Migrasi	Perkiraan Rate Run	Tingkat kesetiaan yang disarankan (minimum)
Environment	Misalnya, produksi, pra-produksi, pengembangan, pengujian, kotak pasir	R	R	Tinggi
Kepatuhan dan peraturan	Kerangka kerja yang berlaku untuk beban kerja (misalnya , HIPAA, SOX, PCI-DSS, ISO, SOC, FedRAMP) dan persyaratan peraturan	R	O	Tinggi
Dependensi	Dependensi hulu dan hilir ke aplikasi atau layanan internal dan eksternal	R	N/A	Tinggi
Pemetaan infrastruktur	Pemetaan ke aset fisik dan/ atau virtual yang membentuk aplikasi	R	R	Tinggi

Nama atribut	Deskripsi	Strategi Penemuan, Desain, dan Migrasi	Perkiraan Rate Run	Tingkat kesetiaan yang disarankan (minimum)
Lisensi	Jenis lisensi perangkat lunak komoditas (misalnya, Microsoft SQL Server Enterprise)	R	R	Tinggi
Biaya	Biaya untuk lisensi perangkat lunak, operasi perangkat lunak, dan pemeliharaan	N/A	R	Tinggi medium
Unit bisnis	Misalnya, pemasaran, keuangan, penjualan	R	O	Tinggi
Detail pemilik	Informasi kontak untuk pemilik aplikasi	R	O	Tinggi
Jenis arsitektur	Misalnya, aplikasi web, 2-tier, 3-tier, microservices, service-oriented architecture (SOA)	R	R	Tinggi

Nama atribut	Deskripsi	Strategi Penemuan, Desain, dan Migrasi	Perkiraan Rate Run	Tingkat kesetiaan yang disarankan (minimum)
Tujuan titik pemulihan (RPO), tujuan waktu pemulihan (RTO), dan/ service-level agreement (SLA)	Atribut manajemen layanan saat ini	R	R	Tinggi
Aplikasi yang menghasilkan pendapatan atau aplikasi strategis bisnis?	Ya, jika aplikasi secara langsung atau tidak langsung mempengar uhi pendapata n perusahaan atau dianggap strategis oleh bisnis.	R	O	Tinggi medium
Jumlah pengguna (bersamaan)	Misalnya, pengguna internal, atau eksternal atau, pengguna/ pelanggan internal dan/atau eksternal	R	R	Tinggi medium
Lokasi pengguna	Asal sesi pengguna	R	R	Tinggi medium

Nama atribut	Deskripsi	Strategi Penemuan, Desain, dan Migrasi	Perkiraan Rate Run	Tingkat kesetiaan yang disarankan (minimum)
Risiko dan masalah	Risiko dan masalah yang diketahui	R	O	Tinggi medium
Pertimbangan migrasi	Informasi tambahan apa pun yang mungkin relevan untuk migrasi	R	R	Tinggi medium
Strategi migrasi	Misalnya, salah satu dari AWS 6 Rs untuk migrasi	R	R	Tinggi medium
Rincian basis data	Misalnya, partisi, enkripsi, replikasi, ekstensi, dukungan Secure Sockets Layer (SSL)	R	R	Tinggi
Tim Support	Misalnya, nama tim operasi aplikasi	R	O	Tinggi medium
Solusi pemantauan	Produk yang digunakan untuk memantau aplikasi ini	R	O	Tinggi medium

Nama atribut	Deskripsi	Strategi Penemuan, Desain, dan Migrasi	Perkiraan Rate Run	Tingkat kesetiaan yang disarankan (minimum)
Persyaratan Backup	Jadwal pencadangan yang diperlukan di AWS	R	R	Tinggi medium
Informasi DR	Misalnya, komponen pemulihan bencana untuk aplikasi ini	R	R	Tinggi medium
AWS Persyaratan target	Misalnya, komponen, penempatan akun, jaringan, keamanan	R	R	Tinggi

Infrastruktur

Nama Atribut	Deskripsi	Strategi Penemuan, Desain, dan Migrasi	Perkiraan Rate Run	Tingkat kesetiaan yang disarankan (minimum)
Pengidentifikasi unik	Misalnya, ID server. Biasanya tersedia pada CMDB yang ada atau inventaris internal dan sistem kontrol lainnya.	R	O	Tinggi

	Pertimbangkan untuk membuat ID unik setiap kali ini tidak ditentukan dalam organisasi Anda.			
Nama jaringan	Nama aset dalam jaringan (misalnya, nama host)	R	O	Tinggi
Nama DNS (nama domain yang sepenuhnya memenuhi syarat, atau FQDN)	Nama DNS	O	O	Tinggi medium
Alamat IP dan netmask	Alamat IP internal dan/atau publik	R	R	Tinggi
Jenis aset	Server fisik atau virtual, hypervisor, wadah, perangkat, instance database, dll.	R	R	Tinggi
Nama produk	Vendor komersial dan nama produk (misalnya, VMware ESXi, IBM Power Systems, Exadata)	R	R	Tinggi

Sistem operasi	Misalnya, REHL 8, Windows Server 2019, AIX 6.1	R	R	Tinggi
Konfigurasi	CPU yang dialokasikan, jumlah inti, utas per inti, total memori, penyimpanan, kartu jaringan	R	R	Tinggi
Pemanfaatan	CPU, memori, dan puncak penyimpanan dan rata-rata . Throughput instance database.	R	R	Tinggi
Lisensi	Jenis lisensi komoditas (misalnya, Standar RHEL)	R	R	Tinggi

Apakah infrastruktur bersama?	Ya atau Tidak untuk menunjukkan layanan infrastruktur yang menyediakan layanan bersama seperti penyedia otentikasi, sistem pemantauan, layanan cadangan, dan layanan serupa	R	O	Tinggi
Pemetaan aplikasi	Aplikasi atau komponen aplikasi yang berjalan di infrastruktur ini	R	O	Tinggi
Data komunikasi	Misalnya, server ke server pada tingkat proses	R	N/A	Tinggi medium
AWS Persyaratan target	Misalnya, jenis contoh, akun, subnet, grup keamanan, perutean	R	R	Tinggi
Strategi, pola, dan alat migrasi	Misalnya, salah satu dari 6 Rs untuk migrasi, pola teknis tertentu, perkakas migrasi	R	O	Tinggi

Risiko dan masalah	Risiko dan masalah yang diketahui	R	O	Tinggi medium

Penilaian aplikasi terperinci

Tujuan dari penilaian aplikasi terperinci adalah pemahaman lengkap tentang aplikasi yang ditargetkan dan infrastruktur terkait (komputasi, penyimpanan, dan jaringan). Data kesetiaan tinggi diperlukan untuk menghindari jebakan. Misalnya, adalah umum bagi organisasi untuk berasumsi bahwa mereka sepenuhnya memahami aplikasi. Ini wajar, dan itu benar dalam banyak kasus. Namun, untuk meminimalkan risiko terhadap bisnis, penting untuk memvalidasi pengetahuan kelembagaan dan dokumentasi statis dengan memperoleh data terprogram sebanyak mungkin. Ini akan menangani proses penemuan yang berat. Anda dapat fokus pada elemen data yang berasal dari sumber alternatif, seperti informasi spesifik bisnis, peta jalan strategis, dan lain-lain.

Kuncinya adalah menghindari perubahan menit terakhir selama dan setelah migrasi. Misalnya, saat bermigrasi, penting untuk menghindari perubahan berdasarkan dependensi tak dikenal yang mungkin memerlukan penyertaan server ke dalam gelombang migrasi yang sedang berlangsung. Tak lama setelah migrasi, penting untuk menghindari perubahan berdasarkan persyaratan platform terkait untuk mengizinkan lalu lintas atau menyebarkan layanan tambahan. Perubahan yang tidak direncanakan ini meningkatkan risiko masalah keamanan dan operasional. Kami sangat merekomendasikan penggunaan alat penemuan terprogram untuk memvalidasi pola lalu lintas dan dependensi saat melakukan penilaian aplikasi terperinci.

Di awal penilaian, Anda harus mengidentifikasi pemangku kepentingan aplikasi. Ini biasanya sebagai berikut:

- Unit bisnis memimpin
- Pemilik aplikasi
- Arsitek
- Operasi dan dukungan
- Tim pemberdayaan cloud
- Tim platform tertentu seperti komputasi, penyimpanan, dan jaringan

Ada dua pendekatan untuk penemuan terperinci. Penemuan top-down dimulai dengan aplikasi, atau bahkan dengan pengguna, dan berjalan sampai ke infrastruktur. Ini adalah pendekatan yang disarankan ketika identifikasi aplikasi jelas. Sebaliknya, penemuan bottom-up dimulai dengan infrastruktur dan berjalan sampai ke aplikasi atau layanan dan penggunanya. Pendekatan ini berguna ketika program migrasi didorong oleh tim infrastruktur dan saat application-to-infrastructure pemetaan tidak jelas. Secara umum, Anda cenderung menggunakan kombinasi keduanya.

Untuk menyelam jauh ke dalam aplikasi, diagram arsitektur yang ada adalah awal yang baik. Jika ini tidak tersedia, buat satu berdasarkan pengetahuan saat ini. Jangan meremehkan pentingnya tugas ini, bahkan untuk rehost sederhana atau relokasi strategi migrasi. Merencanakan diagram arsitektur membantu Anda mengidentifikasi inefisiensi yang dapat dengan cepat diatasi dengan perubahan kecil saat berada di cloud.

Bergantung pada apakah Anda melakukan pendekatan top-down atau bottom-up, diagram awal akan memplot komponen aplikasi dan layanan atau komponen infrastruktur seperti server dan penyeimbang beban. Setelah komponen dan antarmuka utama diidentifikasi, validasi dengan data terprogram dari alat penemuan dan alat pemantauan kinerja aplikasi. Alat harus mendukung analisis ketergantungan dan memberikan informasi komunikasi antar komponen. Setiap komponen yang membentuk aplikasi ini harus diidentifikasi. Selanjutnya, dokumentasikan dependensi ke aplikasi dan layanan lain, baik internal maupun eksternal.

Dengan tidak adanya perkakas untuk memvalidasi dependensi dan pemetaan, diperlukan pendekatan manual. Misalnya, Anda dapat masuk ke komponen infrastruktur dan menjalankan skrip untuk mengumpulkan informasi komunikasi seperti port terbuka dan koneksi yang sudah mapan. Demikian juga, Anda dapat mengidentifikasi proses yang berjalan dan perangkat lunak yang diinstal. Jangan meremehkan upaya yang diperlukan untuk penemuan manual. Perkakas terprogram dapat menangkap dan melaporkan sebagian besar dependensi dalam beberapa hari, kecuali yang terjadi pada interval yang lebih besar (biasanya persentase kecil). Penemuan manual dapat memakan waktu berminggu-minggu untuk mengumpulkan dan menggabungkan semua titik data, dan masih rentan terhadap kesalahan dan data yang hilang.

Lanjutkan untuk mendapatkan informasi yang ditentukan di bagian persyaratan data untuk setiap aplikasi yang diprioritaskan dan infrastruktur yang dipetakan. Selanjutnya, gunakan kuesioner berikut untuk memandu Anda melalui proses penilaian terperinci. Bertemu dengan pemangku kepentingan yang teridentifikasi untuk mendiskusikan jawaban atas pertanyaan-pertanyaan ini.

Umum

- Apa tingkat kekritisan aplikasi ini? Apakah itu menghasilkan pendapatan? Apakah ini aplikasi bisnis-strategis atau mendukung bisnis? Apakah ini layanan infrastruktur inti yang dimiliki oleh sistem lain?
- Apakah ada proyek transformasi yang sedang berlangsung untuk aplikasi ini?
- Apakah ini aplikasi yang dihadapi secara internal atau eksternal?

Arsitektur

- Apa jenis arsitektur saat ini (misalnya, SOA, layanan mikro, monolit)? Berapa banyak tingkatan yang dimiliki arsitektur? Apakah itu digabungkan dengan erat atau digabungkan secara longgar?
- Apa saja komponennya (misalnya, komputasi, database, penyimpanan jarak jauh, penyeimbang beban, layanan caching)?
- Apa itu API? Jelaskan ini, termasuk nama API, operasi, URL, port, dan protokol.
- Apa latensi maksimum yang ditoleransi antara komponen dan antara ini dan aplikasi atau layanan lainnya?

Operasi

- Di lokasi apa aplikasi ini beroperasi?
- Siapa yang mengoperasikan aplikasi dan infrastruktur? Apakah ini dioperasikan oleh tim internal atau AWS Mitra?
- Apa yang terjadi jika aplikasi ini turun? Siapa yang terpengaruh? Apa dampaknya?
- Di mana pengguna atau pelanggan berada? Bagaimana mereka mengakses aplikasi? Berapa jumlah pengguna bersamaan?
- Kapan penyegaran teknologi terakhir? Apakah penyegaran dijadwalkan di masa depan? Jika demikian, kapan?
- Apa risiko dan masalah yang diketahui untuk aplikasi ini? Bagaimana sejarah pemadaman dan insiden tingkat keparahan sedang dan tingkat keparahan tinggi?
- Apa siklus penggunaan (dalam jam kerja)? Apa zona waktu operasi?
- Apa periode pembekuan perubahan?
- Solusi apa yang digunakan untuk memantau aplikasi ini?

Kinerja

- Apa yang ditunjukkan oleh informasi kinerja yang dikumpulkan? Apakah penggunaan runcing atau konstan dan dapat diprediksi? Berapa kerangka waktu, interval, dan tanggal data kinerja yang tersedia?
- Apakah ada pekerjaan batch terjadwal yang merupakan bagian dari atau berinteraksi dengan aplikasi ini?

Siklus hidup perangkat lunak

- Berapa tingkat perubahan saat ini (mingguan, bulanan, triwulanan, atau tahunan)?
- Apa siklus hidup pengembangan (misalnya, pengujian, pengembangan, QA, UAT, pra-produksi, produksi)?
- Apa metode penyebaran untuk aplikasi dan infrastruktur?
- Apa itu perkakas penyebaran?
- Apakah aplikasi atau infrastruktur ini menggunakan continuous integration (CI) /continuous delivery (CD)? Apa tingkat otomatisasi? Apa tugas manualnya?
- Apa persyaratan lisensi untuk aplikasi dan infrastruktur?
- Apa itu Service Level Agreement (SLA)?
- Apa mekanisme pengujian saat ini? Apa tahapan tesnya?

Migrasi:

- Apa pertimbangan migrasi?

Pada titik ini, perhatikan pertimbangan apa pun saat memigrasikan aplikasi ini. Untuk penilaian yang lebih lengkap dan akurat, dapatkan jawaban atas pertanyaan ini dari berbagai pemangku kepentingan. Kemudian kontraskan pengetahuan dan pendapat mereka.

Ketahanan

- Apa metode pencadangan saat ini? Produk apa yang digunakan untuk cadangan? Apa jadwal pencadangan? Apa kebijakan retensi cadangan?
- Apa tujuan titik pemulihan (RPO) dan tujuan waktu pemulihan (RTO) saat ini?

- Apakah aplikasi ini memiliki rencana pemulihan bencana (DR)? Jika demikian, apa solusi DR?
- Kapan tes DR terakhir?

Keamanan dan kepatuhan

- Apa saja kerangka kepatuhan dan peraturan yang berlaku untuk aplikasi ini? Apa tanggal audit terakhir dan berikutnya?
- Apakah aplikasi ini meng-host data sensitif? Apa klasifikasi datanya?
- Apakah data dienkripsi saat transit atau diam, atau keduanya? Apa mekanisme enkripsi?
- Apakah aplikasi ini menggunakan sertifikat Secure Sockets Layer (SSL)? Apa otoritas penerbit?
- Apa metode otentikasi untuk pengguna, komponen, dan aplikasi dan layanan lainnya?

Basis Data

- Database apa yang digunakan aplikasi ini?
- Berapa jumlah tipikal koneksi bersamaan ke database? Berapa jumlah minimum dan jumlah maksimum koneksi?
- Apa metode koneksi (misalnya, JDBC, ODBC)?
- Apakah string koneksi didokumentasikan? Jika demikian, di mana?
- Apa skema database?
- Apakah database menggunakan tipe data khusus?

Dependensi

- Apa ketergantungan antar komponen? Perhatikan dependensi apa pun yang tidak dapat diselesaikan dan yang memerlukan migrasi komponen bersama-sama.
- Apakah komponen dibagi di seluruh lokasi? Apa konektivitas antara lokasi-lokasi ini (misalnya, WAN, VPN)?
- Apa dependensi aplikasi ini ke aplikasi atau layanan lain?
- Apa dependensi operasional? Misalnya, siklus pemeliharaan dan rilis seperti menambal jendela.

AWS desain aplikasi dan strategi migrasi

Merancang dan mendokumentasikan keadaan masa depan aplikasi Anda adalah faktor keberhasilan migrasi utama. Sebaiknya buat desain untuk semua jenis strategi migrasi, tidak peduli seberapa sederhana atau kompleksnya. Membuat desain akan memunculkan potensi pemblokir, dependensi, dan peluang untuk mengoptimalkan aplikasi bahkan dalam kasus di mana arsitektur tidak diharapkan berubah.

Kami juga merekomendasikan untuk mendekati status aplikasi di masa depan AWS dengan lensa strategi migrasi. Pada tahap ini, pastikan Anda menentukan seperti apa tampilan aplikasi AWS sebagai hasil dari migrasi ini. Desain yang dihasilkan akan berfungsi sebagai dasar untuk evolusi lebih lanjut setelah migrasi.

Daftar berikut berisi sumber daya untuk membantu proses desain:

- AWS Pusat Arsitektur menggabungkan alat dan panduan, seperti AWS Well-Architected Framework. Selain itu, ia menyediakan arsitektur referensi yang dapat Anda gunakan untuk aplikasi Anda.
- Amazon Builders' Library berisi beberapa sumber tentang bagaimana Amazon membangun dan mengoperasikan perangkat lunak.
- AWS Solutions Library menawarkan koleksi solusi berbasis cloud, diperiksa oleh AWS, untuk puluhan masalah teknis dan bisnis. Ini mencakup banyak koleksi arsitektur referensi.
- AWS Panduan Preskriptif menyediakan strategi, panduan, dan pola yang membantu proses desain dan praktik terbaik migrasi.
- AWS Dokumentasi berisi informasi tentang AWS layanan, termasuk Panduan Pengguna dan Referensi API.
- Pusat Sumber Daya Memulai menyediakan beberapa tutorial langsung dan penyelaman mendalam untuk mempelajari dasar-dasarnya sehingga Anda dapat mulai membangun. AWS

Tergantung di mana Anda berada dalam perjalanan cloud, AWS yayasan mungkin sudah ada. AWS Yayasan ini meliputi:

- AWS Daerah telah diidentifikasi.
- Akun telah dibuat atau dapat diperoleh sesuai permintaan.
- Jaringan umum telah diterapkan.
- AWS Layanan dasar telah digunakan di dalam akun.

Sebaliknya, Anda mungkin berada di awal proses, dan AWS fondasi belum didirikan. Kurangnya fondasi yang mapan dapat membatasi ruang lingkup desain aplikasi Anda atau memerlukan pekerjaan lebih lanjut untuk mendefinisikannya. Jika ini masalahnya, kami sarankan untuk mendefinisikan dan menerapkan desain dasar dari landing zone secara paralel dengan pekerjaan desain aplikasi. Desain aplikasi membantu mengidentifikasi persyaratan seperti struktur AWS akun, jaringan, virtual private cloud (VPC), rentang Classless Inter-Domain Routing (CIDR), layanan bersama, keamanan, dan operasi cloud.

AWS Control Towermenyediakan cara termudah untuk mengatur dan mengatur AWS lingkungan multi-akun yang aman, yang disebut landing zone. AWS Control Tower membuat landing zone Anda menggunakan AWS Organizations, yang menyediakan pengelolaan dan tata kelola akun yang berkelanjutan serta implementasi pengalaman berbasis praktik AWS terbaik bekerja dengan ribuan pelanggan saat mereka pindah ke cloud.

Aplikasi future state

Mulailah dengan menetapkan strategi migrasi awal untuk aplikasi ini. Pada titik ini, strategi dianggap awal karena dapat berubah sebagai bagian dari desain negara masa depan, yang dapat mengungkap potensi keterbatasan. Untuk memvalidasi asumsi awal, lihat pohon keputusan 6 Rs. Juga, dokumentasikan fase migrasi potensial. Misalnya, apakah aplikasi ini akan dimigrasikan dalam satu acara (semua komponen dimigrasikan secara bersamaan)? Atau apakah ini migrasi bertahap (beberapa komponen dimigrasikan nanti)?

Perhatikan bahwa strategi migrasi untuk aplikasi tertentu mungkin tidak unik. Ini karena beberapa tipe R dapat digunakan untuk memigrasikan komponen aplikasi. Misalnya, pendekatan awal bisa mengangkat dan menggeser aplikasi tanpa perubahan. Namun, komponen aplikasi mungkin berada di aset infrastruktur yang berbeda yang mungkin memerlukan perawatan yang berbeda. Misalnya, aplikasi terdiri dari tiga komponen, masing-masing berjalan di server terpisah, dan salah satu server menjalankan sistem operasi lama yang tidak didukung di cloud. Komponen itu akan memerlukan pendekatan replatform, sementara dua komponen lainnya, berjalan dalam versi server yang didukung, dapat dihosting ulang. Ini adalah kunci untuk menetapkan strategi migrasi ke setiap komponen aplikasi dan infrastruktur terkait yang sedang dimigrasikan.

Selanjutnya, dokumentasikan konteks dan masalah, dan tautkan artefak yang ada yang menentukan status saat ini:

- Mengapa aplikasi ini dimigrasikan?
- Apa saja perubahan yang diusulkan?

- Apa manfaatnya?
- Apakah ada risiko atau pemblokir utama?
- Apa kerugian saat ini?
- Apa yang ada di ruang lingkup dan di luar ruang lingkup?

Pengulangan

Sepanjang pekerjaan desain, pertimbangkan bagaimana solusi dan arsitektur untuk aplikasi ini dapat digunakan kembali untuk aplikasi lain. Bisakah solusi ini digeneralisasi?

Persyaratan

Dokumentasikan persyaratan fungsional dan nonfungsional untuk aplikasi ini, termasuk keamanan. Ini termasuk persyaratan status saat ini dan masa depan, tergantung pada strategi migrasi yang dipilih. Gunakan informasi yang dikumpulkan selama penilaian aplikasi terperinci untuk memandu proses ini.

Arsitektur calon

Jelaskan arsitektur future untuk aplikasi ini. Pertimbangkan untuk membuat templat diagram yang dapat digunakan kembali yang berisi blok bangunan untuk lingkungan sumber Anda (lokal) dan AWS lingkungan target (misalnya, AWS Wilayah target, akun, VPC, dan Zona Ketersediaan).

Buat tabel komponen yang sedang dimigrasikan dan komponen yang akan menjadi baru. Sertakan aplikasi dan layanan lain (baik di tempat atau di cloud) yang berinteraksi dengan aplikasi ini.

Tabel berikut mencantumkan contoh komponen. Ini tidak mewakili arsitektur referensi atau konfigurasi yang diperiksa.

Nama	Deskripsi	Detail
Aplikasi	Layanan eksternal (koneksi masuk)	Layanan menggunakan data dari API yang terbuka.
DNS	Resolusi nama (internal)	Amazon Route 53 digunakan sebagai bagian dari pengaturan akun dasar

Nama	Deskripsi	Detail
Penyeimbang Beban Aplikasi	Mendistribusikan lalu lintas di antara layanan backend	Menggantikan penyeimbang beban lokal. Migrasi Kolam A.
Keamanan aplikasi	Perlindungan DDoS	Diimplementasikan dengan menggunakan AWS Shield
Grup keamanan	Firewall virtual	Batasi akses ke instance aplikasi pada port 443 (inbound).
Server A	Front-end	Rehost, menggunakan Amazon Elastic Compute Cloud (Amazon EC2).
Peladen B	Front-end	Rehost menggunakan Amazon EC2.
Server C	Logika aplikasi	Rehost menggunakan Amazon EC2.
Peladen D	Logika aplikasi	Rehost menggunakan Amazon EC2.
Layanan Database Relasional Amazon (Amazon RDS) - Amazon Aurora	Basis Data	Menggantikan server E dan F
Pemantauan dan peringatan	Ubah kontrol	Amazon CloudWatch
Pencatatan audit	Ubah kontrol	AWS CloudTrail
Penambalan dan akses jarak jauh	Maintenance	AWS Systems Manager
Akses sumber daya	Kontrol akses yang aman	AWS Identity and Access Management (IAM)
Autentikasi	Akses pengguna	Amazon Cognito

Nama	Deskripsi	Detail
Sertifikat	SSL/TLS	AWS Certificate Manager
API 1	API Eksternal	Amazon API Gateway
Penyimpanan objek	Hosting gambar	Amazon Simple Storage Service (Amazon S3)
Kredensial	Manajemen dan hosting kredensional	AWS Secrets Manager
AWS Lambda fungsi	Pengambilan kredensi database dan kunci API	AWS Lambda
gateway internet	Akses internet outbound	Gerbang internet ke VPC
Subnet pribadi 1	Backend dan DB	Zona Ketersediaan 1 — VPC 1
Subnet pribadi 2	Backend dan DB	Zona Ketersediaan 2 — VPC 1
Subnet publik 1	Front-end	Zona Ketersediaan 1 — VPC 1
Subnet publik 2	Front-end	Zona Ketersediaan 2 — VPC 1
Layanan Backup	Database dan cadangan instans EC2	AWS Backup
DR	Ketahanan Amazon EC2	CloudEndure Pemulihan Bencana

Setelah komponen diidentifikasi, plot mereka dalam diagram menggunakan alat pilihan Anda. Bagikan desain awal dengan pemangku kepentingan aplikasi utama, termasuk pemilik aplikasi, arsitek perusahaan, dan tim platform dan migrasi. Pertimbangkan untuk mengajukan pertanyaan-pertanyaan berikut:

- Apakah tim umumnya setuju dengan desainnya?
- Bisakah tim operasi mendukungnya?
- Bisakah desain dikembangkan?

- Apakah ada pilihan lain?
- Apakah desain sesuai dengan standar arsitektur dan kebijakan keamanan?
- Apakah ada komponen yang hilang (misalnya, repositori kode, perkakas CI/CD, titik akhir VPC)?

Keputusan arsitektur

Sebagai bagian dari proses desain, Anda mungkin akan menemukan lebih banyak opsi untuk keseluruhan arsitektur atau bagian tertentu darinya. Dokumentasikan opsi ini di samping alasan untuk opsi yang disukai atau dipilih. Keputusan ini dapat didokumentasikan sebagai keputusan arsitektur.

Pastikan bahwa opsi utama terdaftar dan dijelaskan dengan cukup detail bagi pembaca baru untuk memahami opsi dan alasan di balik keputusan untuk menggunakan satu opsi di atas yang lain.

Lingkungan siklus hidup perangkat lunak

Dokumentasikan setiap perubahan pada lingkungan saat ini. Misalnya, lingkungan pengujian dan pengembangan akan dibuat ulang AWS dan tidak dimigrasikan.

Penandaan

Jelaskan penandaan wajib dan direkomendasikan untuk setiap komponen infrastruktur serta nilai penandaan untuk desain ini.

Strategi migrasi

Pada titik desain ini, asumsi awal tentang strategi migrasi harus divalidasi. Konfirmasikan bahwa ada konsensus tentang strategi R yang dipilih. Dokumentasikan strategi migrasi aplikasi secara keseluruhan dan strategi untuk komponen aplikasi individual. Seperti disebutkan sebelumnya, komponen aplikasi yang berbeda mungkin memerlukan tipe R yang berbeda untuk migrasi.

Selain itu, selaraskan strategi migrasi dengan pendorong dan hasil bisnis utama. Juga, jelaskan setiap pendekatan bertahap untuk migrasi, seperti pergerakan komponen dalam peristiwa migrasi yang berbeda.

Untuk informasi lebih lanjut tentang menentukan 6 Rs Anda, lihat rekomendasi AWS Migration Hub strategi.

Pola dan alat migrasi

Dengan strategi migrasi yang ditentukan untuk komponen aplikasi dan infrastruktur, Anda sekarang dapat menjelajahi pola teknis tertentu. Misalnya, strategi rehost dapat diimplementasikan dengan alat migrasi seperti Layanan Migrasi AWS Aplikasi. Jika Anda tidak perlu mereplikasi status atau data, Anda dapat mencapai hasil yang sama dengan menerapkan ulang aplikasi menggunakan Amazon Machine Image (AMI) dan pipeline penerapan aplikasi.

Demikian pula, untuk memplatform ulang atau memfaktorkan ulang (arsitek ulang) aplikasi, Anda dapat menggunakan alat seperti AWS App2Container, AWS Database Migration Service(AWS DMS), (),. AWS Schema Conversion ToolAWS SCTAWS DataSync Untuk containerizing, Anda dapat menggunakan Amazon Elastic Container Service (Amazon ECS), Amazon Elastic Kubernetes Service (Amazon EKS), atau. AWS Fargate Saat membeli kembali, Anda dapat menggunakan AMI untuk produk tertentu atau solusi Perangkat Lunak sebagai Layanan (SaaS) dari AWS Marketplace.

Evaluasi berbagai pola dan opsi yang tersedia untuk mencapai tujuan. Pertimbangkan pro dan kontra, dan kesiapan operasional migrasi. Untuk membantu analisis Anda, gunakan pertanyaan-pertanyaan berikut:

- Dapatkah tim migrasi mendukung pola-pola ini?
- Apa keseimbangan antara biaya dan manfaat?
- Bisakah aplikasi, layanan, atau komponen ini dipindahkan ke layanan terkelola?
- Apa upaya untuk menerapkan pola ini?
- Apakah ada peraturan atau kebijakan kepatuhan yang mencegah penggunaan pola tertentu?
- Bisakah pola ini digunakan kembali? Pola yang dapat digunakan kembali lebih disukai. Namun, terkadang sebuah pola hanya akan digunakan satu kali. Pertimbangkan keseimbangan antara upaya pola sekali pakai atas pola alternatif yang dapat digunakan kembali.

AWS Panduan Preskriptif berisi berbagai pola dan teknik migrasi.

Manajemen dan operasi layanan

Saat membuat desain aplikasi untuk migrasi ke AWS, pertimbangkan kesiapan operasional. Saat mengevaluasi persyaratan kesiapan dengan tim aplikasi dan infrastruktur Anda, pertimbangkan pertanyaan-pertanyaan berikut:

- Apakah mereka siap mengoperasikannya?

- Apakah prosedur respons insiden didefinisikan?
- Apa perjanjian tingkat layanan yang diharapkan (SLA)?
- Apakah pemisahan tugas diperlukan?
- Apakah tim yang berbeda siap untuk mengoordinasikan tindakan dukungan?
- Siapa yang bertanggung jawab untuk apa?

Pertimbangan cutover

Mempertimbangkan strategi dan pola migrasi, apa yang penting untuk diketahui saat aplikasi dimigrasikan? Perencanaan cutover adalah kegiatan pasca-desain. Namun, dokumentasikan pertimbangan apa pun untuk kegiatan dan persyaratan yang dapat diantisipasi. Misalnya, mendokumentasikan persyaratan untuk melakukan bukti konsep, jika berlaku, dan menguraikan persyaratan pengujian, audit, atau validasi.

Risiko, asumsi, masalah, dan ketergantungan

Dokumentasikan setiap risiko terbuka, asumsi, dan potensi masalah yang belum terselesaikan. Tetapkan kepemilikan yang jelas untuk item ini, dan lacak kemajuan sehingga desain dan strategi keseluruhan dapat disetujui untuk implementasi. Selain itu, dependensi kunci dokumen untuk mengimplementasikan desain ini.

Memperkirakan biaya operasional

Untuk memperkirakan biaya AWS arsitektur target Anda, gunakan Kalkulator AWS Harga. Tambahkan komponen infrastruktur Anda seperti yang ditentukan oleh desain Anda, dan dapatkan perkiraan biaya operasional. Faktor dalam lisensi perangkat lunak yang diperlukan untuk komponen aplikasi Anda dan yang belum termasuk dalam AWS layanan yang akan Anda gunakan.

Analisis portofolio dan perencanaan migrasi

Tahap penilaian ini berfokus pada penyelesaian penemuan dan analisis tingkat portofolio yang dimulai di bagian penemuan Portofolio dan perencanaan awal. Tujuannya adalah untuk mengulangi dan menetapkan dasar untuk portofolio awal aplikasi dan infrastruktur. Garis dasar ini mencakup identifikasi semua dependensi, mengulangi model rasionalisasi untuk migrasi, membuat kasus bisnis terperinci, dan menguraikan rencana gelombang migrasi. Akibatnya, kesetiaan data yang dibutuhkan lebih tinggi. Tahap ini akan membutuhkan investasi waktu. Untuk mempercepat hasil penilaian, sebaiknya gunakan sebanyak mungkin sumber data terprogram, seperti alat penemuan.

Hasil utama dari tahap ini meliputi:

- Aplikasi dengan kesetiaan tinggi dan inventaris infrastruktur
- Strategi migrasi tingkat tinggi untuk setiap aplikasi
- Rencana gelombang migrasi dengan kepercayaan tinggi
- Kasus bisnis terperinci

Memahami persyaratan data penilaian lengkap

Tabel berikut menjelaskan informasi yang diperlukan untuk mendapatkan tampilan portofolio lengkap dari aplikasi dalam migrasi dan infrastruktur terkait.

Tabel menggunakan singkatan berikut:

- R, untuk yang dibutuhkan
- O, untuk opsional
- N/A, karena tidak berlaku

Aplikasi

Nama atribut	Deskripsi	Inventarisasi dan prioritas	Kasus Bisnis Terperinci	Tingkat kesetiaan yang disarankan (minimum)
Pengidentifikasi unik	Misalnya, ID aplikasi. Biasanya tersedia pada CMDB yang ada atau inventaris internal dan sistem kontrol lainnya. Pertimbangkan untuk membuat ID unik setiap kali ini tidak ditentukan dalam organisasi Anda.	R	R	Tinggi
Nama aplikasi	Nama yang dengannya aplikasi ini diketahui organisasi Anda. Sertakan vendor komersial off-the-shelf (COTS) dan nama produk jika berlaku.	R	R	Tinggi
Apakah COTS?	Ya atau Tidak. Apakah ini aplikasi komersial atau	R	R	Tinggi

Nama atribut	Deskripsi	Inventarisasi dan prioritas	Kasus Bisnis Terperinci	Tingkat kesetiaan yang disarankan (minimum)
	pengembangan internal			
Produk dan versi COTS	Nama dan versi produk perangkat lunak komersial	R	R	Tinggi
Deskripsi	Fungsi dan konteks aplikasi utama	R	R	Tinggi
Kekritisan	Misalnya, aplikasi strategis atau menghasilkan pendapatan, atau mendukung fungsi kritis	R	R	Tinggi
Tipe	Misalnya, database, manajemen hubungan pelanggan (CRM), aplikasi web, multimedia, layanan bersama TI	R	R	Tinggi

Nama atribut	Deskripsi	Inventarisasi dan prioritas	Kasus Bisnis Terperinci	Tingkat kesetiaan yang disarankan (minimum)
Environment	Misalnya, produksi, pra-produksi, pengembangan, pengujian, kotak pasir	R	R	Tinggi
Kepatuhan dan peraturan	Kerangka kerja yang berlaku untuk beban kerja (misalnya , HIPAA, SOX, PCI-DSS, ISO, SOC, FedRAMP) dan persyaratan peraturan	R	R	Tinggi
Dependensi	Dependensi hulu dan hilir ke aplikasi atau layanan internal dan eksternal . Dependens i non-teknis seperti elemen operasional (misalnya, siklus pemeliharaan)	R	O	Tinggi

Nama atribut	Deskripsi	Inventarisasi dan prioritas	Kasus Bisnis Terperinci	Tingkat kesetiaan yang disarankan (minimum)
Pemetaan infrastruktur	Pemetaan ke aset fisik dan/atau virtual yang membentuk aplikasi	R	R	Tinggi
Lisensi	Jenis lisensi perangkat lunak komoditas (misalnya, Microsoft SQL Server Enterprise)	R	R	Tinggi medium
Biaya	Biaya untuk lisensi perangkat lunak, operasi perangkat lunak, dan pemeliharaan	N/A	R	Tinggi medium
Unit bisnis	Misalnya, pemasaran, keuangan, penjualan	R	R	Tinggi
Detail pemilik	Informasi kontak untuk pemilik aplikasi	R	R	Tinggi
Informasi DR	Komponen pemulihan bencana	R	R	Tinggi

Nama atribut	Deskripsi	Inventarisasi dan prioritas	Kasus Bisnis Terperinci	Tingkat kesetiaan yang disarankan (minimum)
Strategi migrasi	Misalnya, salah satu dari 6 Rs untuk migrasi ke AWS	R	R	Tinggi
Tiket Support	12-24 bulan data untuk membantu menilai produktivitas dan dampak keuangan dari pemadaman, perlambatan, pembatasan transaksi, dan pembengkakan jendela batch	O	R	Sedang

Infrastruktur

Nama Atribut	Deskripsi	Inventarisasi dan prioritas	Kasus bisnis	Tingkat kesetiaan yang disarankan (minimum)
Pengidentifikasi unik	Misalnya, ID server. Biasanya tersedia pada CMDB yang ada atau inventaris internal dan sistem	R	R	Tinggi

Nama Atribut	Deskripsi	Inventarisasi dan prioritas	Kasus bisnis	Tingkat kesetiaan yang disarankan (minimum)
	kontrol lainnya. Pertimbangkan untuk membuat ID unik setiap kali ini tidak ditentukan dalam organisasi Anda.			
Nama jaringan	Nama aset dalam jaringan (misalnya, nama host)	R	R	Tinggi
Nama DNS (nama domain yang sepenuhnya memenuhi syarat, atau FQDN)	Nama DNS	R	O	Tinggi
Alamat IP dan netmask	Alamat IP internal dan/atau publik	R	R	Tinggi
Jenis aset	Server fisik atau virtual, hypervisor, wadah, perangkat, instance database, dll.	R	R	Tinggi

Nama Atribut	Deskripsi	Inventarisasi dan prioritas	Kasus bisnis	Tingkat kesetiaan yang disarankan (minimum)
Nama produk	Vendor komersial dan nama produk (misalnya, VMware ESXi, IBM Power Systems, Exadata)	R	R	Tinggi
Sistem operasi	Misalnya, REHL 8, Windows Server 2019, AIX 6.1	R	R	Tinggi
Konfigurasi	CPU yang dialokasikan, jumlah inti, utas per inti, total memori, penyimpanan, kartu jaringan	R	R	Tinggi
Pemanfaatan	CPU, memori, dan puncak penyimpanan dan rata-rata . Database instance throughput.	R	R	Tinggi

Nama Atribut	Deskripsi	Inventarisasi dan prioritas	Kasus bisnis	Tingkat kesetiaan yang disarankan (minimum)
Lisensi	Jenis lisensi komoditas (misalnya, Standar RHEL)	R	R	Tinggi
Apakah infrastruktur bersama?	Ya atau Tidak untuk menunjukkan layanan infrastruktur yang menyediakan layanan bersama seperti penyedia otentikasi, sistem pemantauan, layanan cadangan, dan layanan serupa	R	R	Tinggi
Pemetaan aplikasi	Aplikasi atau komponen aplikasi yang berjalan di infrastruktur ini	R	R	Tinggi

Nama Atribut	Deskripsi	Inventarisasi dan prioritas	Kasus bisnis	Tingkat kesetiaan yang disarankan (minimum)
Biaya	Biaya terisi penuh untuk server bare-metal, termasuk perangkat keras, pemeliharaan, operasi, penyimpanan (SAN, NAS, Object), lisensi sistem operasi, pangsa rackspace, dan overhead pusat data	N/A	R	Tinggi medium
Perkiraan volume transfer data (masuk/keluar)	Misalnya, per aset infrastruktur lebih dari per hari selama periode 30 hari	O	R	Sedang

Jaringan

Nama Atribut	Deskripsi	Inventarisasi dan prioritas	Kasus bisnis	Tingkat kesetiaan yang disarankan (minimum)
Ukuran pipa (MB/s), redundansi (Y/N)	Spesifikasi tautan WAN saat ini (mis., 1000 MB/s redundan)	R	R	Tinggi medium
Pemanfaatan tautan	Puncak dan pemanfaat an rata-rata, transfer data keluar (GB/bulan)	R	R	Tinggi medium
Latensi (ms)	Latensi saat ini antara lokasi yang terhubung.	R	O	Tinggi
Biaya	Biaya saat ini per bulan	N/A	R	Tinggi medium

Migrasi

Nama Atribut	Deskripsi	Inventarisasi dan prioritas	Kasus bisnis	Tingkat kesetiaan yang disarankan (minimum)
Rehost	Upaya pelanggan dan mitra untuk setiap beban kerja (orang-ha ri), tarif biaya	N/A	R	Tinggi medium

Nama Atribut	Deskripsi	Inventarisasi dan prioritas	Kasus bisnis	Tingkat kesetiaan yang disarankan (minimum)
	pelanggan dan Mitra per hari, biaya alat, jumlah beban kerja			
Platform Ulang	Upaya pelanggan dan mitra untuk setiap beban kerja (orang-hari), tarif biaya pelanggan dan mitra per hari, jumlah beban kerja	N/A	R	Tinggi medium
Refactor	Upaya pelanggan dan mitra untuk setiap beban kerja (orang-hari), tarif biaya pelanggan dan mitra per hari, jumlah beban kerja	N/A	R	Tinggi medium
Pensiun	Jumlah server, biaya dekomisi rata-rata	N/A	R	Tinggi medium

Nama Atribut	Deskripsi	Inventarisasi dan prioritas	Kasus bisnis	Tingkat kesetiaan yang disarankan (minimum)
Zona pendaratan	Gunakan kembali yang ada (Y/N), daftar AWS Wilayah yang dibutuhkan, biaya	N/A	R	Tinggi medium
Orang dan perubahan	Jumlah staf yang akan dilatih dalam operasi dan pengembangan cloud, biaya pelatihan per orang, biaya waktu pelatihan per orang	N/A	R	Tinggi medium
Durasi	Durasi migrasi beban kerja dalam lingkup (bulan)	O	R	Tinggi medium
Biaya paralel	Kerangka waktu dan tingkat biaya apa adanya dapat dihapus selama migrasi	N/A	R	Tinggi medium

Nama Atribut	Deskripsi	Inventarisasi dan prioritas	Kasus bisnis	Tingkat kesetiaan yang disarankan (minimum)
	Kerangka waktu dan tingkat di mana AWS produk dan layanan, dan biaya infrastru ktur lainnya, diperkenalkan selama migrasi	N/A	R	Tinggi medium

Menetapkan baseline untuk portofolio aplikasi

Untuk membuat rencana gelombang migrasi dengan kepercayaan tinggi, Anda harus menetapkan dasar untuk portofolio aplikasi dan infrastruktur terkaitnya. Garis dasar portofolio memberikan pandangan komprehensif tentang ruang lingkup migrasi, termasuk dependensi teknis dan strategi migrasi. Garis dasar portofolio memberikan kejelasan tentang aplikasi mana yang berada dalam lingkup migrasi dan bahwa titik data yang diuraikan dalam bagian Memahami persyaratan data penilaian lengkap dikumpulkan. Demikian juga, semua infrastruktur terkait (komputasi, jaringan penyimpanan) dipahami dan dipetakan ke aplikasi.

Ketergantungan teknis dapat dijelaskan dalam empat kategori:

- pplication-to-infrastructureDependensi membangun hubungan antara perangkat lunak dan perangkat keras fisik atau virtual. Misalnya, ada ketergantungan antara aplikasi CRM dan mesin virtual tempat ia diinstal.
- Dependensi komponen aplikasi menggambarkan bagaimana komponen yang berjalan di aset infrastruktur yang berbeda berinteraksi. Contoh ketergantungan komponen aplikasi adalah front end web yang berjalan pada mesin virtual, dengan lapisan aplikasi yang berjalan pada mesin virtual yang berbeda, dan database yang berjalan pada cluster database.
- pplication-to-applicationDependensi berhubungan dengan interaksi antara aplikasi atau komponen aplikasi dengan aplikasi lain atau komponennya. Contoh application-to-application ketergantungan

adalah aplikasi pemrosesan pembayaran dan aplikasi manajemen saham. Aplikasi ini independen, tetapi mereka terus-menerus berinteraksi menggunakan operasi API yang ditentukan.

- Dependensi pplication-to-infrastructure layanan secara teknis adalah application-to-application dependensi, mengingat bahwa layanan infrastruktur itu sendiri merupakan aplikasi. Namun, kami sarankan untuk mengkategorikan ini secara terpisah. Alasan utamanya adalah bahwa layanan infrastruktur biasanya dibagi oleh banyak aplikasi, sehingga mereka memiliki jejak ketergantungan yang panjang. Mereka juga biasanya mengikuti strategi dan pola migrasi yang berbeda.Misalnya, penyeimbang beban dapat berisi kumpulan penyeimbang untuk beberapa aplikasi. Yang penting adalah ketergantungan ke kumpulan, yang kemungkinan akan dimigrasikan secara individual, di samping aplikasi dependen, sedangkan penyeimbang beban itu sendiri dipertahankan atau dihentikan. Selain itu, individualisasi dependensi application-to-infrastructure layanan membantu menghindari grup ketergantungan palsu. Kelompok ketergantungan palsu adalah ketika beberapa aplikasi bisnis dikelompokkan bersama, menyiratkan bahwa memiliki ketergantungan umum ke layanan infrastruktur harus dimigrasikan pada saat yang sama. Misalnya, layanan otentikasi, seperti Active Directory, cenderung dikaitkan dengan kelompok besar aplikasi. Kuncinya adalah mendekati aplikasi ini secara individual dan untuk mengatasi ketergantungan dengan mengaktifkan layanan, seperti AWS Directory Service untuk Microsoft Active Directory, di lingkungan cloud.

Saat Anda menetapkan baseline untuk portofolio, sebaiknya Anda mengonfirmasi strategi migrasi untuk setiap komponen aplikasi. Strategi migrasi akan menjadi salah satu dari 6 Rs untuk migrasi (lihat bagian Mengulangi strategi migrasi 6 Rs). Dalam baseline portofolio, salah satu dari 6 Rs harus dikaitkan dengan setiap aplikasi. Strategi 6 R juga harus dikaitkan dengan masing-masing komponen infrastruktur aplikasi.

Untuk menetapkan versi dasar portofolio, termasuk dependensi dan strategi migrasi, gunakan alat penemuan otomatis (lihat Mengevaluasi kebutuhan alat penemuan). Lengkapi data dengan informasi yang dikumpulkan dari pemangku kepentingan utama seperti pemilik aplikasi dan tim infrastruktur. Terus kumpulkan data sampai Anda mendapatkan inventaris portofolio lengkap yang sesuai dengan atribut dan tingkat kesetiaan yang diuraikan di bagian persyaratan data untuk tahap ini. Dataset yang dihasilkan akan berperan penting dalam mendorong migrasi.

Pertimbangkan bahwa, tergantung pada luas cakupan migrasi Anda dan alat yang tersedia, aktivitas ini dapat memakan waktu beberapa minggu untuk diselesaikan.

Mengulangi kriteria prioritas

Sebelum Anda membuat rencana gelombang migrasi, kami sarankan Anda mengulangi kriteria prioritas aplikasi untuk beralih dari pemilihan aplikasi pilot ke perencanaan gelombang jangka panjang.

Di bagian sebelumnya, kami memperkenalkan kriteria prioritas default yang akan memprioritaskan aplikasi sederhana yang siap cloud (lihat Memprioritaskan aplikasi). Ini karena pada tahap awal kami merekomendasikan memulai dengan aplikasi nonkritis untuk menyempurnakan proses migrasi dan menggabungkan pelajaran yang dipetik. Namun, pada tahap ini, dan untuk membuat rencana jangka panjang, urutan migrasi aplikasi harus diselaraskan dengan driver bisnis. Menerapkan kriteria baru akan menghasilkan peringkat aplikasi baru yang akan menjadi masukan utama untuk perencanaan gelombang.

Tinjau poin data yang tersedia dari portofolio aplikasi, dan pilih atribut yang akan menentukan prioritas aplikasi berdasarkan driver bisnis.

Pertama, validasi driver bisnis Anda (lihat Penggerak bisnis dan prinsip panduan teknis). Selanjutnya, berdasarkan driver bisnis Anda, pilih atribut yang akan membantu memprioritaskan aplikasi untuk migrasi.

Tabel berikut menunjukkan contoh kriteria prioritas yang selaras dengan pendorong bisnis untuk inovasi.

Atribut atau titik data	Nilai yang mungkin	Skor (0-99)	Pentingnya atau relevansi faktor pengalikan
Sistem operasi	AIX	80	Tinggi (1x)
	Solaris	80	
	HP-UX	80	
	Mainframe	70	
	Windows	50	
	Linux	20	

Atribut atau titik data	Nilai yang mungkin	Skor (0-99)	Pentingnya atau relevansi faktor pengalikan
Kekritisan bisnis	Tinggi	60	Tinggi (1x)
	Sedang	40	
	Rendah	20	
Arsitektur	Dipasangkan dengan erat	60	Tinggi (1x)
	Berpasangan secara longgar	20	
Model operasi	Tradisional - tidak ada CI/CD	60	Sedang tinggi (0.8x)
	CI/CD dasar	40	
	Penuh DevOps	20	
Jumlah instans komputasi	1-3	60	Sedang tinggi (0.8x)
	4-10	40	
	11 atau lebih	20	
Strategi migrasi	Refactor (arsitek ulang)	70	Sedang (0.6x)
	Platform Ulang	40	
	Pembelian kembali	30	
	Rehost	10	

Tabel berikut menunjukkan contoh kriteria prioritas yang selaras dengan pendorong bisnis untuk pengurangan biaya cepat.

Atribut atau titik data	Nilai yang mungkin	Skor (0-99)	Pentingnya atau relevansi faktor pengalikan
Produk basis data	Oracle	70	Tinggi (1x)
	Microsoft SQL	70	
	Lainnya	20	
Sistem operasi	Windows	70	Tinggi (1x)
	Linux	70	
	Lainnya	20	
Pemanfaatan CPU (rata-rata)	Lebih dari 36%	60	Tinggi (1x)
	Kurang dari 36%	40	
Jumlah instans komputasi	11 atau lebih	60	Sedang tinggi (0.8x)
	4-10	40	
	1-3	20	
Strategi Migrasi	Pensiun	80	Sedang (0.6x)
	Rehost	70	
	Platform Ulang	50	
	Refactor (arsitek ulang)	10	

Uji kriteria prioritas dan ulangi sampai Anda umumnya setuju dengan output. Dibutuhkan setidaknya tiga atau empat iterasi untuk mendapatkan versi dasar.

Mengulangi pemilihan strategi migrasi 6 Rs

Pada tahap ini, kami menyarankan Anda mengulangi dan mengembangkan pohon keputusan 6 Rs. Bagian Menentukan tipe R untuk migrasi memperkenalkan pohon keputusan default. Kami merekomendasikan untuk merevisi pohon, mempertimbangkan pembelajaran selama migrasi aplikasi percontohan awal, dan memastikan bahwa itu masih selaras dengan driver bisnis, kriteria prioritas, dan keadaan unik Anda. Validasi pohon keputusan dengan aplikasi sampel, dan verifikasi bahwa itu masih menghasilkan strategi yang diharapkan. Jika tidak, perbarui logika yang sesuai. Pohon yang dihasilkan akan menjadi kunci dalam menetapkan garis dasar untuk portofolio aplikasi dan dalam mengalokasikan strategi migrasi untuk setiap komponen aplikasi.

Seperti yang dijelaskan di bagian 6 Rs sebelumnya, 6 Rs juga berlaku untuk infrastruktur, dan sama pentingnya untuk menetapkannya sesuai. Sementara komponen aplikasi tertentu akan memiliki strategi migrasi, pada tingkat infrastruktur, setiap aset infrastruktur akan mengikuti strategi migrasi tertentu yang mungkin berbeda dari strategi yang ditetapkan untuk komponen aplikasi yang didukungnya.

Ingat bahwa pohon keputusan 6 Rs hanya berlaku untuk komponen aplikasi. Strategi migrasi untuk infrastruktur berasal dari strategi yang dipilih untuk aplikasi. Misalnya, untuk komponen aplikasi yang akan di-replatform, infrastruktur saat ini yang menjadi host dapat dihentikan.

Pastikan bahwa strategi migrasi dialokasikan untuk setiap komponen aplikasi dan infrastruktur yang terkait. Informasi ini akan menjadi faktor kunci ketika memperkirakan upaya, kapasitas, dan keterampilan yang dibutuhkan, dan saat membuat rencana gelombang migrasi.

Untuk informasi lebih lanjut tentang menentukan 6 Rs Anda, lihat rekomendasi AWS Migration Hub strategi.

Perencanaan gelombang

Dalam perencanaan gelombang, kelompok ketergantungan adalah kumpulan aplikasi dan infrastruktur yang memiliki dependensi teknis dan nonteknis yang tidak dapat diselesaikan. Karena dependensi ini, aplikasi dan infrastruktur dalam grup dependensi harus dimigrasikan pada waktu yang sama atau pada tanggal tertentu. Misalnya, aplikasi yang berjalan pada mesin virtual dan database yang berjalan di mesin virtual terpisah, di mana ada persyaratan latensi rendah atau volume lalu lintas tinggi dan kueri kompleks, cenderung bermigrasi bersama daripada mengoperasikan satu komponen di cloud dan yang lainnya di tempat. Demikian juga, aplikasi

independen yang berinteraksi melalui API dengan persyaratan latensi rendah serupa juga akan dimigrasikan pada saat yang bersamaan.

Gelombang migrasi biasanya berlangsung 4-8 minggu, dan dapat berisi satu atau lebih peristiwa migrasi. Kelompok ketergantungan digabungkan menjadi gelombang sehingga gelombang dapat berisi satu atau lebih kelompok ketergantungan. Gelombang juga berisi aktivitas lain yang diperlukan untuk migrasi. Ini termasuk penyiapan AWS infrastruktur (seperti landing zone, keamanan, dan operasi), alat migrasi, dan aktivitas migrasi seperti replikasi data, perencanaan cut-over, pengujian, dan dukungan pasca-migrasi.

Untuk mengukur keberhasilan dan melacak kemajuan, gelombang harus diselaraskan dengan hasil dan pendorong bisnis. Ini juga akan mempengaruhi durasi gelombang dan kelompok ketergantungan yang terkandung dalam gelombang. Penyelesaian gelombang harus mencerminkan pencapaian yang terukur. Perencanaan gelombang juga dapat menggabungkan faktor-faktor lain, seperti prinsip panduan teknis. Misalnya, gelombang dapat didefinisikan oleh lingkungan (misalnya, pengembangan, pengujian, produksi) atau dengan strategi migrasi (misalnya, gelombang rehost, gelombang replatform).

Untuk membuat rencana gelombang migrasi yang efektif dan percaya diri tinggi, Anda harus mendapatkan tampilan lengkap dari portofolio aplikasi, infrastruktur terkait (komputasi, penyimpanan, jaringan), pemetaan ketergantungan, dan strategi migrasi.

Bagian tentang menetapkan garis dasar untuk portofolio aplikasi menjelaskan empat kategori dependensi teknis. Dependensi ini berkontribusi pada penciptaan gelombang migrasi dan definisi kelompok ketergantungan. Kelompok ketergantungan akan ditentukan oleh kekritisan ketergantungan. Selain itu, dependensi nonteknis harus dipertimbangkan. Misalnya, jadwal rilis aplikasi, jendela pemeliharaan, dan tanggal bisnis utama seperti pemrosesan akhir bulan akhir kuartal akan memengaruhi rencana gelombang.

Tentukan apakah ketergantungannya lunak atau keras. Ketergantungan lunak adalah hubungan antara dua atau lebih aset, atau dari aset ke kendala, yang tidak tergantung pada lokasi komponen. Misalnya, dua sistem yang beroperasi di jaringan lokal yang sama (atau infrastruktur yang sama) dapat dipisahkan dengan memindahkan salah satu sistem tersebut ke cloud sementara yang lain tetap di tempat. Contoh lain adalah sistem yang dapat dimigrasikan selama jendela pemeliharaan tanpa memengaruhi aktivitas pemeliharaan.

Ketergantungan keras adalah hubungan antara dua atau lebih aset, atau dari aset ke kendala, yang bergantung pada lokasi. Misalnya, dua sistem yang beroperasi di jaringan lokal yang sama, dan yang sangat bergantung pada latensi rendah untuk komunikasi antara server aplikasi dan server database,

memiliki ketergantungan keras. Memindahkan hanya satu dari sistem ini ke cloud akan menyebabkan masalah fungsionalitas atau kinerja yang tidak dapat diselesaikan. Demikian juga, alasan nonteknis, seperti ketersediaan sumber daya (misalnya, tim yang melakukan migrasi) atau kendala operasional, seperti jendela pemeliharaan di mana dua sistem hanya dapat dimigrasikan dalam jendela waktu tertentu, dapat membuat ketergantungan keras untuk aset ini.

Untuk membuat rencana gelombang migrasi, tentukan grup dependensi Anda dengan menganalisis dependensi, idealnya dari sumber data yang sangat tepercaya seperti alat penemuan khusus, dan gabungkan informasi ini dengan kriteria prioritas aplikasi dan keadaan operasional Anda. Aplikasi di bagian atas peringkat prioritas harus ditargetkan untuk gelombang migrasi awal Anda. Tentukan kapasitas gelombang (jumlah aplikasi yang dapat ditampung gelombang) berdasarkan ketersediaan sumber daya, toleransi risiko, kendala bisnis dan teknis, pengalaman, dan keterampilan. Pertimbangkan untuk bekerja dengan Layanan AWS Profesional atau Mitra Kompetensi AWS Migrasi, yang dapat menyediakan spesialis untuk membantu Anda selama proses berlangsung.

Kriteria prioritas adalah indikasi awal dari urutan di mana Anda akan memindahkan aplikasi Anda ke cloud. Namun, kelompok ketergantungan akan menjadi penentu aktual untuk aplikasi yang akan dipindahkan pada waktu tertentu. Ini karena aplikasi yang digolongkan sebagai prioritas tinggi dapat memiliki ketergantungan keras pada aplikasi yang berada di tengah atau di bawah peringkat.

Strategi migrasi juga akan mempengaruhi komposisi gelombang. Misalnya, aplikasi prioritas tinggi yang memerlukan strategi refactor yang mungkin memerlukan beberapa minggu atau bulan analisis, desain, pengujian, dan persiapan kemungkinan akan ditempatkan dalam gelombang berikutnya.

Membuat rencana gelombang

Prasyarat untuk memigrasikan gelombang aplikasi adalah data portofolio aplikasi dan penilaian aplikasi terperinci dari kelompok aplikasi yang akan dimigrasikan dalam gelombang. Penilaian terperinci harus mencakup daftar aplikasi dalam gelombang, detail infrastruktur terkait, desain target, dan strategi migrasi untuk setiap aplikasi.

Menetapkan kepemilikan dan tata kelola gelombang adalah kunci untuk mengelola dan melacak pekerjaan gelombang, ketergantungan program, manajemen perubahan, masalah, dan risiko. Pastikan bahwa kerangka kerja tata kelola ada untuk mengelola rencana.

Untuk menguraikan rencana gelombang, mulailah dengan konstruksi gelombang default. Apa yang terjadi dalam gelombang? Setelah input awal ditentukan, gelombang dapat dimulai. Biasanya, kegiatannya adalah:

1. Perbaiki rencana cutover. Kegiatan ini harus menguraikan runbook dan langkah-langkah yang harus diambil pada saat migrasi, termasuk koordinasi dengan tim internal dan eksternal lainnya.
2. Perbaiki rencana rollback. Apa yang harus dilakukan untuk memutar kembali aplikasi jika ada yang salah?
3. Siapkan infrastruktur target. Misalnya, Anda dapat membuat atau memperluas AWS landing zone (AWS akun, keamanan, jaringan, layanan infrastruktur, infrastruktur pendukung lainnya).
4. Uji infrastruktur target.
5. Mengoperasikan perkakas migrasi. Misalnya, instal agen replikasi dan mulai transfer data.
6. Lakukan rencana cutover dan runbook dry run. Kelompokkan semua anggota tim yang berpartisipasi, dan tinjau semua langkah sebelumnya.
7. Memantau replikasi data dan penyebaran infrastruktur.
8. Konfirmasikan kesiapan untuk pengoperasian infrastruktur dan aplikasi di AWS.
9. Konfirmasikan kesiapan keamanan.
10. Konfirmasikan kepatuhan dan persyaratan peraturan (misalnya, validasi beban kerja sebelum migrasi dan pasca-migrasi) jika berlaku.
11. Migrasikan aplikasi ke AWS dan lakukan pengujian pra go-live.
12. Berikan dukungan pasca-migrasi untuk jangka waktu tertentu, seperti 3 hari, saat tim operasi dan tim migrasi sepenuhnya tersedia untuk menyelesaikan masalah, dan menerapkan pengoptimalan.
13. Lakukan tinjauan pasca-migrasi. Dokumentasikan pelajaran yang dipetik, dan gabungkan ke dalam gelombang masa depan.
14. Lakukan penutupan gelombang dengan mengonfirmasi serah terima operasional dan perolehan metrik untuk pelaporan.

Berapa lama masing-masing kegiatan ini akan ditentukan oleh kompleksitas ruang lingkup, kapasitas gelombang, orang-orang yang terlibat, dan keadaan unik Anda. Jika memungkinkan, gelombang yang lebih kecil lebih disukai karena itu akan mengurangi dampak penundaan atau penghambat migrasi. Tentukan, dengan tim Anda, berapa durasi default gelombang.

Selanjutnya, lanjutkan untuk menganalisis tanggal untuk membuat struktur gelombang kosong tingkat tinggi awal (tanpa aplikasi yang ditetapkan). Pertimbangkan pertanyaan-pertanyaan berikut:

- Berapa total panjang program migrasi?
- Apa tenggat waktu?
- Apakah ada tanggal keluar pusat data tetap?

- Apakah ada tanggal akhir kontrak kolokasi?
- Apa siklus penyegaran aplikasi dan infrastruktur?
- Apa siklus pemeliharaan dan rilis aplikasi?
- Apakah ada tanggal ketika migrasi harus dihindari (misalnya, siklus rilis dan pemeliharaan, akhir tahun, hari libur, pemrosesan akhir bulan)?

Dengan pertimbangan ini, plot gelombang ke dalam rencana. Untuk mempercepat proses migrasi, kami merekomendasikan gelombang yang tumpang tindih jika memungkinkan. Kunci untuk gelombang yang tumpang tindih adalah mendefinisikan dan mempertimbangkan apa yang terjadi dalam gelombang. Biasanya, aktivitas penyebaran, validasi infrastruktur target, dan sinkronisasi data akan terjadi selama paruh pertama gelombang. Babak kedua akan fokus pada migrasi aktual, pengujian, dan serah terima operasional. Ini berarti bahwa tim yang berbeda terlibat dalam setiap setengah proses, dan Anda dapat memperoleh beberapa efisiensi. Misalnya, segera setelah tim yang terlibat dalam persiapan infrastruktur target menyelesaikan pekerjaan mereka, mereka dapat mulai mengerjakan persyaratan gelombang berikutnya. Secara umum, lebih disukai bahwa sebagian besar gelombang memiliki panjang dan struktur yang sama untuk memfasilitasi pendekatan migrasi seperti pabrik. Namun, selama proses perencanaan gelombang, ukuran gelombang tertentu dapat diperpanjang untuk memenuhi dependensi atau persyaratan operasional.

Selanjutnya, berdasarkan kelompok ketergantungan yang telah diidentifikasi, tentukan ukuran maksimum gelombang dalam hal jumlah kelompok ketergantungan yang dapat dikandungnya. Ukuran gelombang biasanya ditentukan oleh selera risiko (misalnya, berapa banyak perubahan paralel yang dapat ditoleransi), dan ketersediaan sumber daya (misalnya, berapa banyak perubahan paralel yang dapat dilakukan dengan sumber daya, keterampilan, dan anggaran yang tersedia). Namun, selama perencanaan awal, jangan dibatasi oleh persyaratan dan ketersediaan sumber daya. Gelombang yang mengandung lebih dari satu kelompok ketergantungan dapat didekomposisi menjadi gelombang yang lebih kecil dalam iterasi future.

Setelah kelompok dependensi untuk gelombang tertentu telah dikonfirmasi, tinjau persyaratan sumber daya untuk memigrasikan gelombang. Pertimbangkan untuk menyesuaikan ukuran gelombang (jumlah kelompok ketergantungan yang dikandungnya) berdasarkan kebutuhan sumber daya. Hal ini dapat menyebabkan gelombang yang lebih kecil atau lebih besar. Ulangi rencana gelombang sesuai kebutuhan sampai semua gelombang telah ditentukan.

Mengelola perubahan

Portofolio aplikasi dan infrastruktur terkait akan berubah selama siklus hidup program migrasi. Program migrasi yang berjalan lama hidup berdampingan dengan evolusi dan perubahan bisnis normal. Aplikasi terus berkembang saat menunggu untuk dimigrasi. Server ditambahkan atau dihapus, infrastruktur baru digunakan di tempat. Diharapkan bahwa ruang lingkup gelombang atau kelompok ketergantungan akan membutuhkan perubahan. Perubahan diperlukan terutama ketika, lebih dekat ke tanggal migrasi, ketergantungan yang sebelumnya tidak diketahui diidentifikasi, atau server baru disertakan dalam inventaris. Terkadang ini bisa terjadi selama migrasi itu sendiri.

Perubahan lingkup mempengaruhi kelompok ketergantungan dan gelombang. Untuk menangani perubahan dan meminimalkan dampak, penting untuk menetapkan mekanisme kontrol ruang lingkup. Mekanisme kontrol perubahan ruang lingkup membutuhkan definisi satu sumber kebenaran untuk ruang lingkup. Ini bisa menjadi alat untuk mengelola ruang lingkup, atau file.csv, spreadsheet, atau database, seperti yang didefinisikan oleh tata kelola program migrasi. Anda harus mengidentifikasi perubahan, menganalisis dampak, dan mengkomunikasikan perubahan kepada pemangku kepentingan yang relevan sehingga mereka dapat mengambil tindakan. Rencana gelombang akan diulang sebagai hasilnya.

Kasus bisnis terperinci

Pada tahap ini, kami merekomendasikan untuk memvalidasi dan memperluas ruang lingkup kasus bisnis untuk memberikan tingkat detail yang lebih besar untuk mendukung program transformasi. Kasus bisnis terarah awal yang dirakit dengan cepat dirancang untuk memberikan kepercayaan yang cukup untuk berinvestasi dalam langkah-langkah dasar dan tingkat perencanaan terperinci berikutnya.

Mengembangkan kasus bisnis terperinci mendukung proses perencanaan ini dengan cara berikut:

- Memberikan analisis keuangan yang menginformasikan keputusan tentang apa yang harus dimigrasikan dan dimodernisasi, opsi mana yang harus dipilih dan bagaimana melakukan fase dan memprioritaskan pekerjaan
- Memvalidasi, menyempurnakan, dan mengembangkan kasus keuangan terarah asli dengan memeriksa ulang secara rinci:
  - Potensi pengurangan biaya infrastruktur
  - Produktivitas TI internal dan efisiensi operasi outsourcing
  - Perkiraan untuk investasi yang diperlukan untuk pengaturan program, migrasi, dan modernisasi

- Mengidentifikasi, memperkirakan skala, dan menyiapkan proses untuk melacak driver nilai lebih lanjut yang dibawa migrasi

Dalam kasus bisnis terperinci, Anda menetapkan yang berikut:

- Dasar obyektif untuk mengamankan mandat dan investasi untuk melaksanakan setidaknya tahap pertama migrasi
- Harapan kinerja keuangan minimum dasar untuk program
- Kejelasan atas dasar keuangan di mana berbagai desain migrasi dan keputusan prioritas dibuat, sehingga ketika keadaan dan orang-orang berubah selama program, kepemimpinan baru dapat membuat pilihan berdasarkan informasi.
- Wawasan tentang area tambahan pengoptimalan biaya yang akan dieksplorasi setelah data penggunaan awal tersedia saat beban kerja dimigrasikan dan mulai beroperasi
- Perkiraan nilai yang dibawa transformasi cloud ke bisnis dari peningkatan ketahanan dan kelincahan
- KPI, metrik, dan asumsi terkait digunakan untuk memperkirakan pengembalian finansial dari peningkatan ketahanan dan kelincahan, yang kemudian membentuk dasar untuk mendorong realisasi manfaat utama keluar dari program

Tentukan skenario yang diperlukan untuk kasus ini

Ketika membangun kasus bisnis terperinci, biasanya perlu untuk mengembangkan beberapa skenario untuk mendukung berbagai tujuan yang digunakan untuk kasus bisnis.

Skenario perubahan minimum — Untuk menilai ekspektasi kinerja keuangan minimum, siapkan skenario yang mengasumsikan perubahan minimum yang diharapkan pada status quo. Skenario ini, sebagai skenario terburuk, adalah dukungan yang berguna saat mendapatkan mandat untuk berinvestasi dalam migrasi. Skenario ini memodelkan tingkat pertumbuhan kapasitas minimum yang diharapkan dan perubahan minimum untuk quality-of-service kebutuhan lain, seperti ketersediaan dan ketahanan. Perubahan terkecil menciptakan biaya terendah dan inefisiensi sumber daya paling sedikit untuk model operasi saat ini.

Skenario yang paling mungkin — Untuk menginformasikan strategi program dan keputusan prioritas, siapkan skenario yang mencerminkan apa yang diharapkan bisnis terjadi. Skenario ini harus mencakup kemungkinan pertumbuhan atau pengurangan pemanfaatan puncak dan biaya

peningkatan untuk memenuhi permintaan akan kualitas layanan tingkat tinggi (terutama ketersediaan dan ketahanan) dari bisnis.

Skenario spesifik lainnya — Di mana masih perlu untuk membuat asumsi yang dapat berdampak besar pada kasus bisnis, kembangkan skenario baik di mana asumsi itu benar dan di mana tidak. Namun, kami sarankan untuk menjaga jumlah skenario alternatif ini seminimal mungkin. Membuat lebih dari tiga hingga empat skenario secara total memperlambat kemajuan, dan menjadi mahal, membingungkan, dan sulit dipertahankan. Jika memungkinkan, lakukan eksperimen dan bekerja untuk menghilangkan asumsi yang lebih besar.

Memvalidasi dan menyempurnakan infrastruktur dan model biaya migrasi

Setelah Anda menyelesaikan analisis portofolio dan menyiapkan desain dan ukuran AWS layanan target, perbaiki perkiraan biaya operasional untuk model operasi saat ini (COM) dan model operasi future (FOM) AWS untuk setiap skenario. Biasanya perlu untuk menyempurnakan perkiraan untuk hal-hal berikut:

- Biaya infrastruktur COM dari server host hypervisor, server bare-metal, penyimpanan, perangkat jaringan, penyegaran perangkat keras alat keamanan, instalasi, dan pemeliharaan. Hitung ini dengan harga aktual dan tingkat diskon untuk kapasitas yang dibutuhkan untuk skenario tersebut.
- Biaya pusat data COM dan fasilitas kolokasi, termasuk ruang, pendinginan, daya, rak, catu daya tak terputus (UPS), pemasangan kabel, sistem keamanan fisik, ukuran untuk pertumbuhan dan ditentukan untuk memenuhi kapasitas, dan tingkat ketersediaan dan pemulihan bencana (DR) yang tinggi untuk skenario tersebut.
- Biaya layanan jaringan COM, termasuk biaya untuk tautan WAN, jaringan pengiriman konten, dan jaringan pribadi virtual (VPN), dihitung menggunakan harga kontrak untuk konektivitas, bandwidth, throughput, dan kebutuhan latensi untuk skenario tersebut.
- Biaya aplikasi dan perangkat lunak infrastruktur COM berdasarkan kontrak yang ada untuk memberikan pertumbuhan atau pengurangan penggunaan untuk skenario tersebut.
- Biaya AWS utilitas FOM, termasuk dukungan teknis dan layanan terkelola sesuai kebutuhan, berdasarkan arsitektur layanan yang disempurnakan, ukuran instans, model harga pilihan, penggunaan yang diharapkan, dan volatilitas penggunaan.
- Lisensi aplikasi FOM berdasarkan desain aplikasi akhir, konfigurasi infrastruktur yang menjalankan aplikasi, pertumbuhan dari waktu ke waktu, dan aturan transferabilitas lisensi.
- Perkiraan biaya migrasi dan modernisasi FOM, disempurnakan untuk mencerminkan rencana gelombang migrasi dasar untuk skenario tersebut, dan dirinci untuk menyediakan biaya untuk

setiap beban kerja, terutama bagi mereka yang akan direplatform, dibeli kembali, atau difaktorkan ulang.

- Biaya penonaktifan FOM, termasuk perkiraan penghapusan aset dan biaya penghentian awal kontrak, direvisi untuk mencerminkan waktu penonaktifan dalam rencana gelombang migrasi dasar, verifikasi aset apa yang dapat digunakan kembali dan aset apa yang dapat dialihkan untuk meminimalkan penghapusan, dan biaya pembuangan aset fisik dan media.
- Biaya proses paralel migrasi disempurnakan untuk mencerminkan waktu setiap pemotongan migrasi dan setiap penonaktifan layanan yang ada.

Menyempurnakan produktivitas TI dan operasi TI serta mendukung model nilai efisiensi

Seperti halnya kasus bisnis terarah, ada dua pendekatan utama untuk menyempurnakan dan mengembangkan model nilai seputar operasi dan dukungan TI. Pendekatan yang Anda pilih tergantung pada apakah COM dikelola secara internal atau dengan kontraktor atau layanan outsourcing:

Peningkatan produktivitas tim internal

Di mana operasi dan dukungan TI dikelola di rumah, fokus kasus bisnis adalah sebagai berikut:

- Mengidentifikasi dan mengukur keuntungan produktivitas dari migrasi dan otomatisasi operasional apa pun yang termasuk dalam ruang lingkup
- Memvalidasi bahwa waktu yang dibebaskan untuk tim internal dapat dengan mudah dan produktif diterapkan pada kegiatan lain yang biasanya bernilai lebih tinggi, memberikan peluang untuk kemajuan dan penghargaan yang lebih besar kepada tim dan nilai lebih bagi organisasi

Menilai berapa banyak waktu yang dihabiskan setiap anggota dalam setiap peran dalam tim untuk berbagai kegiatan rutin mereka, dan bimbingan tentang pengurangan beban kerja yang diharapkan untuk kegiatan yang berbeda.

Tabel berikut memberikan panduan awal untuk tingkat tipikal pengurangan beban kerja berdasarkan aktivitas untuk tugas-tugas yang menghabiskan sebagian besar operasi TI dan upaya dukungan di berbagai peran dalam tim. Tabel ini mencakup deskripsi tentang bagaimana produktivitas dicapai.

Catatan: Aktivitas yang tercantum biasanya dilakukan oleh anggota tim dalam beberapa peran yang berbeda, sehingga penghematan produktivitas untuk setiap tugas harus dinilai di seluruh

rangkaian peran lengkap dalam tim. Misalnya, dalam tim operasi TI yang diselenggarakan oleh menara infrastruktur (seperti komputasi, penyimpanan, dan jaringan), perencanaan belanja modal dan penganggaran mungkin umum untuk menara lead untuk setiap menara.

Kegiatan operasional dan dukungan	Tingkat tabungan	Penggerak produktivitas
Desain infrastruktur	Sedang	Desain disederhanakan, dengan parameter yang lebih sedikit untuk dipertimbangkan.
Perencanaan dan penganggaran belanja modal	Tinggi	Layanan elastis opex-centric menghapus hampir semua masalah penganggaran dan perencanaan.
Pembelian	Tinggi	Pengadaan sangat disederhanakan setelah AWS akun dibuat.
Perencanaan kapasitas	Menengah-sangat tinggi	Jaringan dan beban kerja manajemen kapasitas komputasi biasanya dihilangkan, dan untuk penyimpanan sangat disederhanakan
Penyetelan	Tinggi-sangat tinggi	Tuning tidak diperlukan untuk layanan terkelola dan hampir tidak diperlukan untuk layanan lain karena instance dapat diubah ukurannya kapan saja.
Mengelola kegagalan perangkat keras	Sangat tinggi	Semua aspek penanganan perangkat keras di cloud ditangani secara transparan oleh. AWS

Kegiatan operasional dan dukungan	Tingkat tabungan	Penggerak produktivitas
Memantau ketersediaan dan komunikasi server	Tinggi	Pemantauan dan komunikasi disederhanakan secara ekstensif dengan dukungan AWS alat dan otomatisasi.
Manajemen keamanan	Sedang	Beban kerja berkurang secara signifikan dengan kemampuan AWS keamanan dan dengan AWS memiliki tanggung jawab keamanan untuk perangkat keras, perangkat lunak, jaringan, dan fasilitas AWS Cloud.
Peningkatan jaringan dan penyimpanan, pemeliharaan, dan tambalan.	Sangat tinggi	Semua aspek pemeliharaan jaringan dan penyimpanan di cloud ditangani secara transparan oleh. AWS
Racking dan susun - logistik perangkat keras	Sangat tinggi	Semua aspek pengelolaan perangkat keras di cloud ditangani secara transparan oleh. AWS
Cadangan	Sedang	Backup disederhanakan secara ekstensif dengan AWS alat, sistem penyimpanan yang fleksibel, dan otomatisasi.

Kegiatan operasional dan dukungan	Tingkat tabungan	Penggerak produktivitas
Layanan terkelola (seperti Amazon S3, Amazon RDS, AWS Lambda dan) AWS Fargate	Sangat tinggi	Layanan terkelola berjalan di lingkungan yang dikelola sepenuhnya oleh AWS, sehingga tidak memerlukan pemeliharaan, penambalan, pemantauan, atau aktivitas manajemen penyediaan.
Pengaturan dan commissio ning perangkat dan layanan	Tinggi-sangat tinggi	Aktivitas untuk pengatura n perangkat keras untuk perkebunan yang dimigrasi ke AWS biasanya dikurangi , kecuali untuk perangkat konektivitas WAN untuk membangun VPN atau AWS Direct Connect koneksi ke AWS pusat data.
Perlindungan titik akhir dan perlindungan antivirus	Tinggi	Aplikasi dan pemeliharaan perlindungan endpoint dan layanan antivirus biasanya secara ekstensif otomatis sebagai bagian dari desain migrasi.
Ancaman, kerentanan, dan penilaian risiko	Tinggi	AWS memberikan dukungan untuk elemen-elemen ini, berfokus pada platform inti dan mekanisme yang AWS menyediakan untuk mengamankan arsitektur menyederhanakan penilaian.

Kegiatan operasional dan dukungan	Tingkat tabungan	Penggerak produktivitas
Manajemen proyek infrastru ktur pusat data	Tinggi	Manajemen proyek untuk pekerjaan instalasi untuk perluasan, penyegaran, atau penonaktifan layanan infrastru ktur. Sementara beberapa manajemen perangkat lunak dan layanan infrastruktur tetap ada, ini jauh lebih sederhana daripada infrastruktur lokal, dan aktivitas perangkat keras dihilangkan.
Manajemen fasilitas pusat data	Menengah-sangat tinggi	Pekerjaan manajemen fasilitas yang dapat dikaitkan dengan semua server, perangkat penyimpanan, peralatan keamanan, dan rak terkait dihapus untuk semua yang dimigrasikan. Namun, beberapa pekerjaan biasanya tetap untuk menyediakan fasilitas untuk perangkat jaringan tautan WAN dan untuk infrastruktur apa pun yang disimpan di tempat dalam arsitektur hibrida.

Kegiatan operasional dan dukungan	Tingkat tabungan	Penggerak produktivitas
Arsitektur aplikasi, pengembangan, manajemen, dan pengujian	Rendah	Penggunaan rantai alat pengembangan tangkas, dikombinasikan dengan otomatisasi instantiasi dan penghancuran tumpukan aplikasi untuk membangun lingkungan pengujian sesuai kebutuhan, mengurangi waktu tunggu pengembangan aplikasi dan menghilangkan banyak langkah pengujian manual.
Menginstal dan mengonfigurasi perangkat lunak aplikasi	Sedang	Instalasi dan konfigurasi tumpukan aplikasi lengkap siap diotomatisasi menggunakan layanan seperti AWS CloudFormation dan disederhanakan melalui penggunaan zona pendaratan, yang dapat dengan mudah dikonfigurasi dengan menggunakan. AWS Control Tower

Kegiatan operasional dan dukungan	Tingkat tabungan	Penggerak produktivitas
Dukungan TI	Sedang	Pengurangan dukungan L1 dan L2 dicapai dengan mengurangi masalah kapasitas dan kinerja melalui penggunaan kemampuan Katalog Layanan untuk penyediaan layanan mandiri, peningkatan penggunaan arsitektur ketersediaan tinggi berbiaya rendah (mengurangi pemadaman dan mengonfig urasi penskalaan otomatis dan komputasi tepi).
Administrasi basis data	Minimal-rendah	Kegiatan ini sebagian besar tetap tidak berubah. Mereka biasanya sumber daya pada tingkat yang AWS sama untuk infrastruktur lokal.
Pengambilan, analisis, dan desain persyaratan infrastru ktur dan keamanan	Minimal	
Dokumentasi	Minimal	
Aplikasi dan pemantauan kinerja	Minimal	
Dukungan teknis L3, menjawab pertanyaan, dan pemecahan masalah dan pemecahan masalah	Minimal	
Menginstal dan mengonfig urasi perangkat lunak aplikasi	Minimal	

Kegiatan operasional dan dukungan	Tingkat tabungan	Penggerak produktivitas
Dukungan aplikasi L3 (tidak termasuk penganggaran dan perencanaan kapasitas jangka panjang)	Minimal	

Tabel berikut menunjukkan penghematan yang diharapkan untuk setiap tingkat pengurangan beban kerja.

Tingkat	Diharapkan
Sangat tinggi	85% - 100%
Tinggi	60% - 90%
Sedang	30% - 70%
Rendah	10% - 35%
Minimal	0% - 10%

Metrik ini memberikan titik awal untuk menilai peningkatan produktivitas dan memasukkannya ke dalam kasus bisnis terperinci. Keuntungan produktivitas aktual bervariasi berdasarkan situasi tertentu. Ini dapat berguna untuk menghitung penghematan produktivitas di titik tengah dan ujung bawah rentang untuk memperkirakan skenario tipikal dan konservatif.

Seiring kemajuan program, penting untuk menangkap data aktual untuk waktu yang dihabiskan pada setiap aktivitas berdasarkan peran. Data tersebut membangun basis yang lebih baik untuk memperkirakan operasi dan mendukung biaya untuk proyek baru dan perluasan layanan.

Mengalihdayakan operasi TI dan mendukung pengurangan biaya

Di mana operasi dan dukungan TI terutama dialihdayakan atau dikelola dengan kontraktor, alokasi biaya untuk model operasi future (FOM) dapat disiapkan dengan meminta penawaran dari AWS Mitra yang menawarkan solusi layanan terkelola, termasuk Partner-led (AMS). AWS AWS Managed Services Anda juga dapat menghubungi manajer AWS akun Anda dan meminta harga untuk AMS

secara langsung, seperti yang dijelaskan dalam ayat tentang Membangun optimalisasi biaya operasional dalam bagian Membuat kasus bisnis terarah.

Untuk kasus bisnis terperinci, ganti angka tolok ukur apa pun dengan kutipan berdasarkan tagihan bahan layanan yang direvisi dan konsumsi AWS layanan yang diharapkan, paket AMS dan opsi apa pun yang diperlukan, dan tingkat layanan yang diperlukan. Biaya akan memiliki komponen implementasi satu kali dan run rate berbasis konsumsi.

Sertakan operasi TI yang tersisa, dukungan yang harus dipertahankan untuk layanan apa pun yang tidak akan dimigrasikan AWS, dan biaya satu kali jika ada penalti kontrak (misalnya, untuk penghentian dini).

Mengembangkan model nilai ketahanan

Pada AWS, Anda dapat membangun berbagai ketersediaan tinggi, pemulihan bencana, dan arsitektur toleran kesalahan. Penetapan harga berbasis konsumsi berarti bahwa layanan dibebankan hanya jika digunakan. Bersama-sama, kedua faktor ini memberikan kinerja biaya yang luar biasa untuk ketahanan.

Selain itu, AWS ccustomer telah menggunakan ini untuk meningkatkan ketahanan beban kerja mereka. Survei IDC 2018 memberikan contoh pelanggan yang berpartisipasi mencapai 73 persen lebih sedikit pemadaman per tahun, pengurangan 58 persen dalam mean time to recover (MTTR) dan penurunan 94 persen dalam produktivitas yang hilang. Survei yang sama menunjukkan bahwa manfaat finansial yang diperoleh melalui peningkatan ketahanan adalah 50 persen lebih besar daripada manfaat pengurangan biaya infrastruktur TI.

Selain itu, ketahanan lebih lanjut dicapai melalui modernisasi siklus hidup pengembangan perangkat lunak untuk aplikasi. Di mana pipa CI/CD dengan otomatisasi pengujian diperkenalkan untuk mendukung kelincahan bisnis yang lebih besar, cacat perangkat lunak ditangkap lebih awal dalam siklus pengembangan, sangat mengurangi biaya pemeliharaan perangkat lunak.

Untuk menilai dan memasukkan nilai ini dalam kasus bisnis, pertama-tama bekerja dengan pemilik bisnis aplikasi untuk membangun gambaran peluang manfaat total untuk setiap beban kerja yang akan dimigrasikan.Ini mungkin termasuk sebagai item berikut:

- Jumlah, durasi rata-rata, dan sifat interupsi dalam layanan:
  - Contoh gangguan layanan termasuk pemadaman, perlambatan kinerja, batch terencana dan pemeliharaan jendela overrunning, bug dalam fungsi utama, dan pembatasan akses selama periode puncak.

- Dampak pada pendapatan oleh gangguan layanan yang menghasilkan pendapatan, seperti sistem e-commerce:
  - Kemungkinan jumlah transaksi yang tidak dapat diselesaikan melalui gangguan layanan, berdasarkan waktu interupsi dan tingkat transaksi
  - Nilai rata-rata untuk setiap transaksi yang terkena dampak
- Biaya tambahan waktu insinyur pendukung untuk menyelesaikan cacat dalam sistem produksi dibandingkan dengan biaya untuk menemukannya lebih awal dalam proses pengembangan
- Dampak pada produktivitas pengguna internal dan biaya waktu yang hilang

Kemudian buat penilaian pengurangan waktu yang diharapkan dan lebih konservatif yang hilang karena gangguan layanan yang harus dihasilkan oleh peningkatan ketahanan. Misalnya, pertimbangkan untuk memasukkan item-item berikut:

- Mengurangi jumlah pemadaman dan MTTR menggunakan arsitektur ketersediaan tinggi dan peningkatan tujuan waktu pemulihan (RTO) dan tujuan titik pemulihan (RPO)
- Pengurangan perlambatan, penghapusan pelambatan kapasitas dan penghindaran dalam kelebihan pemrosesan batch, menggunakan kemampuan seperti penskalaan otomatis
- Mengurangi jumlah bug aplikasi yang hanya ditemukan dalam produksi, melalui implementasi pipa CI/CD dan pengujian regresi otomatis pada infrastruktur yang diputar dan diputar ke bawah untuk meminimalkan biaya

Satukan ini untuk portofolio aplikasi yang akan dimigrasikan dan dimodernisasi, dan hitung angka nilai bisnis yang diharapkan dan lebih konservatif untuk setiap tahun kasus. Manfaat harus meningkat sejalan dengan jadwal migrasi dan kemudian meningkatkan volume sesuai dengan ekspektasi pertumbuhan penggunaan dari aplikasi yang berkontribusi.

Mengembangkan model nilai kelincahan bisnis

Kelincahan bisnis adalah alasan utama AWS pelanggan bermigrasi. AWSSurvei AWS pelanggan IDC 2018 menunjukkan bahwa bagi mereka, manfaat kelincahan bisnis menyumbang 47 persen dari total manfaat yang diukur dan lebih dari lima kali manfaat yang diperoleh dari pengurangan biaya infrastruktur.

Memprediksi secara akurat semua manfaat kelincahan bisnis yang akan diperoleh dari transformasi apa pun adalah tantangan. Namun, dengan berfokus pada aplikasi yang mendukung sejumlah besar

pengguna atau merupakan sumber diferensiasi bisnis, Anda dapat memodelkan dan memasukkan bagian material dari manfaat ini ke dalam kasus bisnis rinci dasar.

Ketika migrasi berlangsung, secara bertahap menyempurnakan dan memperluas model nilai kelincahan bisnis karena lebih banyak manfaat menjadi dapat diukur. Ini membuat kasus bisnis tetap relevan, sehingga dapat digunakan sebagai alat pendukung keputusan utama untuk mengarahkan program.

Untuk membangun model nilai kelincahan bisnis, gunakan panduan berikut:

- Pilih beban kerja yang memiliki kesempatan untuk mendorong peningkatan kinerja bisnis terbesar, seperti:
  - Beban kerja yang menghasilkan pendapatan
  - Beban kerja operasi bisnis dengan ruang lingkup untuk mendorong peningkatan efisiensi dan menghilangkan biaya dari bisnis
  - Alat produktivitas bisnis yang mendukung basis pengguna yang besar
- Untuk beban kerja yang menghasilkan pendapatan dan efisiensi, lakukan hal berikut:
  - Buat penilaian yang realistis dan lebih konservatif terhadap pertumbuhan pendapatan atau efisiensi operasional yang diharapkan dapat didorong oleh peningkatan aplikasi besar dan kecil.
  - Perkirakan peningkatan jumlah rilis mayor dan minor per tahun yang AWS meningkatkan kecepatan pengembangan aplikasi dan mengurangi waktu penyebaran infrastruktur memungkinkan. Beberapa metrik dasar untuk ini disediakan dalam laporan IDC.
  - Hitung ekspektasi manfaat yang realistis dan lebih konservatif. Petakan mereka selama periode kasus bisnis, membuat tunjangan untuk meningkatkan efisiensi penuh beberapa saat setelah beban kerja masing-masing dimigrasikan.
- Untuk alat produktivitas bisnis, lakukan hal berikut:
  - Buat penilaian yang realistis dan lebih konservatif tentang penghematan waktu yang diharapkan dapat didorong oleh peningkatan aplikasi besar dan kecil.
  - Perkirakan biaya rata-rata waktu dan upaya orang di seluruh basis pengguna yang terkena dampak.
  - Gunakan angka untuk meningkatkan frekuensi rilis mayor dan minor, dan hitung manfaatnya selama jangka waktu kasus bisnis.

Karena peningkatan produktivitas pengembang dan pengurangan waktu untuk peluncuran tidak memerlukan sumber daya tambahan, tambahkan garis manfaat bersih untuk setiap beban kerja ke

dalam model arus kas kasus bisnis untuk dimasukkan dalam arus kas diskon, NPV, ROI, MIRR, dan perhitungan pengembalian.

Penilaian dan peningkatan berkelanjutan

Tahap penilaian ini berfokus pada dua aspek:

- Penilaian aplikasi terperinci yang sedang berlangsung, untuk setiap gelombang aplikasi
- Evolusi berkelanjutan dan peningkatan portofolio Anda

Aspek pertama, penilaian aplikasi terperinci yang sedang berlangsung, berfokus pada penemuan dan analisis terperinci, hingga tingkat arsitektur dan teknologi, untuk sepenuhnya memahami setiap aplikasi dalam gelombang tertentu,AWS desain yang diusulkan, dan strategi migrasi. Penilaian kesiapan migrasi ini merupakan prasyarat untuk memulai gelombang migrasi tertentu.

Aspek kedua, evolusi berkelanjutan dan peningkatan portofolio Anda, berfokus pada manajemen portofolio dan bagaimana Anda berencana untuk meningkatkan aplikasi dari waktu ke waktu, termasuk evolusi dan pelacakan kasus bisnis.

Hasil migrasi utama dari tahap ini meliputi:

- Cakupan migrasi tervalidasi untuk setiap gelombang
- Arsitektur target terdokumentasi dan strategi migrasi untuk aplikasi dalam gelombang migrasi tertentu
- Pola migrasi dan perkakas yang diidentifikasi dan divalidasi
- Persyaratan terdokumentasi (keamanan,AWS infrastruktur, dan operasi) dan pertimbangan pemotongan migrasi untuk setiap gelombang

Hasil optimasi utama dari tahap ini meliputi:

- Model rasionalisasi portofolio dan hasil bisnis
- Usulan perubahan arsitektur dan teknologi, dan manfaat yang diharapkan
- Persyaratan platform (keamanan,AWS infrastruktur, dan operasi)
- Rencana implementasi

Memahami persyaratan data penilaian berkelanjutan

Persyaratan data untuk penilaian berkelanjutan dan peningkatan portofolio aplikasi adalah kombinasi dari persyaratan data dari bagian sebelumnya. Untuk terus mengelola migrasi portofolio dan evolusinya, lihat bagian berikut untuk memahami persyaratan data:

- Untuk penilaian gelombang dan pengoptimalan aplikasi, gunakan persyaratan data dari bagian penilaian aplikasi yang diprioritaskan.
- Untuk manajemen portofolio berkelanjutan, gunakan persyaratan data dari bagian analisis Portofolio dan perencanaan migrasi.
- Untuk menentukan rencana gelombang, lihat bagian Wave planning.

Penilaian gelombang terperinci

Penilaian terperinci aplikasi, menjelang gelombang migrasi dan sebagai penggerak utama untuk migrasi, memiliki persyaratan dan rekomendasi yang sama dengan tahap penilaian aplikasi yang diprioritaskan. Tujuannya adalah untuk memahami secara rinci keadaan aplikasi saat ini dalam gelombang tertentu, dan untuk menghasilkan desain arsitektur negara dan strategi migrasi di future, termasuk aspek operasional, perkakas, dan pola migrasi tertentu.

Terapkan penilaian aplikasi yang diprioritaskan ke kelompok aplikasi dalam gelombang tertentu. Ulangi proses ini sebelum setiap gelombang dalam rencana migrasi Anda. Kuncinya adalah menjadwalkan cukup waktu di antara penilaian terperinci dan awal gelombang. Jumlah waktu yang dibutuhkan akan ditentukan oleh persyaratan tim platform dan migrasi yang menerapkan persyaratan gelombang dan melakukan migrasi. Bekerja dengan tim-tim tersebut untuk menjadwalkan penilaian gelombang terperinci dan gelombang. Kami merekomendasikan untuk menerapkan model seperti pabrik yang meniru jalur produksi.

Penilaian untuk optimasi dan modernisasi

Proses penilaian untuk optimasi beban kerja dan modernisasi yang sudah dimigrasikan ke dalamAWS mirip dengan penilaian beban kerja yang akan dimigrasi keAWS. Apa yang akan berubah, terutama, adalah sumber data untuk melakukan penilaian. DiAWS, ada beberapa out-of-the-box alat dan layanan yang dapat Anda gunakan untuk mendapatkan informasi lebih lanjut tentang aplikasi Anda berjalan diAWS.

Apa dan bagaimana mengoptimalkan dan memodernisasi aplikasi Anda akan didasarkan pada driver dan keadaan unik Anda. Optimalisasi berfokus pada penerapan perubahan pada arsitektur dan teknologi saat ini untuk mengurangi biaya, menyesuaikan persyaratan kinerja, dan menggabungkan pelajaran yang dipetik. Modernisasi berfokus pada membawa aplikasi Anda ke tingkat berikutnya, seperti mengadopsi model tanpa server dan arsitektur microservice.

Ikuti pedoman penilaian aplikasi yang diprioritaskan. Untuk lebih membantu upaya pengoptimalan dan modernisasi Anda, lihat sumber daya yang dimaksud:

- AWSoptimasi biaya memberikan informasi tentang optimasi TI dan menghemat biaya TI Anda.
- AWSCompute Optimizer merekomendasikanAWS sumber daya untuk beban kerja Anda guna mengurangi biaya dan meningkatkan kinerja dengan menggunakan machine learning untuk menganalisis metrik pemanfaatan historis.
- AWSlayanan dan alat pengoptimalan biaya dan kapasitas membantu mengelola sumber daya komputasi sehingga Anda dapat menghabiskan lebih banyak waktu untuk membangun dan mengurangi waktu mengelola biaya komputasi
- Amazon S3 Storage Lens memberikan visibilitas yang dimaksud dengan penggunaan dan aktivitas penyimpanan objek. Ini membuat rekomendasi yang dapat ditindaklanjuti untuk meningkatkan efisiensi biaya dan menerapkan praktik terbaik perlindungan data.
- Database Freedom memfasilitasi migrasi ke layananAWS database dan analisis.
- Amazon CodeGuru adalah alat pengembang yang memberikan rekomendasi cerdas untuk meningkatkan kualitas kode dan mengidentifikasi baris kode aplikasi yang paling mahal.
- AWSlayanan cloud hybrid memberikanAWS pengalaman yang konsisten di mana pun Anda membutuhkannya - dari cloud, hingga di tempat, dan di tepi.

Sumber daya tambahan

- Optimalisasi dan inovasi biaya: Pengantar modernisasi aplikasi (posting blog)
- Mengoptimalkan biaya aplikasi web tanpa server (posting blog)
- Windows padaAWS (blog)
- Aplikasi modern
- Modernisasi aplikasi (AWSRE: Invent 2020)
- AWSpanduan layanan mikro

Iterasi rencana gelombang

Ketika program migrasi bergerak maju dan lebih banyak gelombang dimigrasi, adalah kunci untuk mengembangkan rencana gelombang migrasi berdasarkan pelajaran yang dipelajari dan mengubah prioritas bisnis. Secara khusus, untuk program migrasi yang berjalan lama, penting untuk menilai kembali driver bisnis dan perubahan organisasi, dan untuk memastikan bahwa rencana gelombang migrasi masih berlaku.

Demikian pula, pelajaran yang didapat dari migrasi akan mempengaruhi komposisi rencana gelombang dan ruang lingkup setiap gelombang. Untuk menghindari kehilangan visibilitas terhadap apa yang terjadi, perbarui rencana gelombang. Rencana harus mencerminkan dan melacak apa yang sedang disampaikan, dan harus mengelola dan menilai perubahan lingkup migrasi.

Berkembang dan melacak kasus bisnis

Ketika migrasi berlangsung, terutama untuk program yang berjalan lama, tidak dapat dihindari bahwa tekanan bisnis akan menyebabkan prioritas migrasi dan modernisasi diperiksa ulang secara teratur.

Kami menyarankan Anda berdua mengembangkan kasus bisnis sebagai informasi baru tersedia, dan bahwa Anda melacak kinerja komersial aktual terhadap harapan didokumentasikan dalam kasus bisnis rinci. Rekomendasi ini mencakup hal hal hal hal hal hal hal hal hal hal ini hal

- Perubahan struktural baru dalam organisasi yang mempengaruhi prioritas bisnis dan memengaruhi strategi TI dan portofolio aplikasi dengannya
- Peningkatan kepentingan komersial dari satu bagian dari portofolio aplikasi atau perubahan yang ditargetkan migrasi dan modernisasi untuk mencapai
- Ketersediaan data pemanfaatan sumber daya aktual untuk aplikasi yang dimigrasi, termasuk ukuran penyulingan dan penghitungan dan konfirmasi kasus untuk modernisasi tambahan
- Ketersediaan data tentang upaya yang dikonsumsi dalam operasi TI dan kegiatan pendukung, dan analisis kemungkinan perbaikan operasional dan otomatisasi
- Ketersediaan data mengukur perubahan dalam pengembangan perangkat lunak dan waktu siklus pemeliharaan, cacat perangkat lunak berdasarkan tahap pengembangan dan informasi ketersediaan layanan, dan analisis akar penyebab untuk area yang terbuka untuk perbaikan lebih lanjut

Dengan melacak kinerja terhadap kasus bisnis, Anda dapat mengembangkan kasus ini untuk menyertakan perbaikan lebih lanjut yang dapat lebih mudah dinilai dan diukur setelah migrasi dimulai.

Organisasi tata kelola program jauh lebih siap untuk menanggapi perubahan tekanan bisnis dan mengarahkan transformasi ke arah yang mendorong nilai terbesar pada tingkat risiko yang dapat dikelola dan dapat diterima.

Hal ini sangat penting untuk manfaat produktivitas, ketahanan, dan kelincahan bisnis TI dalam kasus ini. Ini biasanya driver yang lebih besar dan lebih sulit untuk menilai sebelumnya. Dengan melacak kinerja driver ini, tim dapat menyelam lebih dalam dan menyelesaikan masalah yang menghambat realisasi manfaat. Atau kasus bisnis dapat disesuaikan untuk memprioritaskan inisiatif yang mencapai optimalisasi kinerja keuangan yang paling berkelanjutan.

Sumber daya

AWSreferensi

- Amazon Builders' Library
- Modernisasi aplikasi (AWSRE: Invent 2020)
- Strategi penilaian portofolio aplikasi
- AWSPusat Arsitektur
- AWSPengoptimal Compute Optimizer
- AWSlayanan dan alat pengoptimalan biaya dan kapasitas
- AWSoptimasi biaya
- Optimalisasi dan inovasi biaya: Pengantar modernisasi aplikasi (posting blog)
- AWSDokumentasi
- Memulai Pusat Sumber Daya
- AWSMarketplace
- AWS Managed ServicesMitra
- AWSpanduan layanan mikro
- AWSMitra Kompetensi Migrasi
- Aplikasi modern
- Mengoptimalkan biaya aplikasi web tanpa server (posting blog)
- AWSPanduan Preskriptif
- AWSLayanan Profesional
- AWSSolusi Perpustakaan
- Windows padaAWS (blog)

AWSjasa

- AWSApp2Container
- AWSLayanan Migrasi Aplikasi
- Amazon CodeGuru
- AWS Control Tower

- Kebebasan Database
- AWS Database Migration Service
- AWS DataSync
- AWS Direct Connect
- Amazon ECS
- Amazon
- AWS Fargate
- AWS Managed Services
- Evaluator Migrasi
- AWS Migration Hubrekomendasi strategi
- AWSZona Pendaratan
- AWSKalkulator Harga
- AWS Schema Conversion Tool
- Amazon S3 Storage Lens
- AWSSnowball
- AWSSnowcone
- AWS VPN

Sumber daya lainnya

- Membina Transformasi Bisnis dan Organisasi untuk Menghasilkan Nilai Bisnis dengan Amazon Web Services
- Survei IDC 2018

Riwayat dokumen

Tabel berikut menjelaskan perubahan signifikan pada strategi ini. Jika Anda ingin diberi tahu tentang pembaruan masa depan, Anda dapat berlangganan umpan RSS.

Perubahan	Deskripsi	Tanggal
Update	Mengganti nama bagian penemuan Portofolio dan perencanaan awal Akseleras i penemuan dan perencanaan awal; memperbarui diagram pohon keputusan.	Mei 20, 2024
—	Publikasi awal	12 November 2021

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 kekritisan 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.

I

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.

IIoT

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 kebijakanberbasis 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 kredensil.

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 kredensil pengguna, yang Anda simpan dalam bentuk terenkripsi. Ini terdiri dari nilai rahasia dan metadata-nya. 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 kredensil.

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.