Panduan keamanan dan operasi Autonomous Driving Data Framework (ADDF)

AWS Bimbingan Preskriptif

AWS Bimbingan Preskriptif: Panduan keamanan dan operasi Autonomous Driving Data Framework (ADDF)

Table of Contents

Panduan keamanan dan operasi Autonomous Driving Data Framework (ADDF)

Andreas Falkenberg, Junjie Tang, Torsten Reitemeyer, dan Srinivas Reddy Cheruku, Layanan Web Amazon (AWS)

November 2022(sejarah dokumen)

Autonomous Driving Data Framework (ADDF) adalah proyek sumber terbuka yang dirancang untuk menyediakan artefak kode modular yang dapat digunakan kembali untuk tim otomotif yang ingin menerapkan tugas umum untuk sistem bantuan pengemudi tingkat lanjut (ADAS), seperti mengonfigurasi penyimpanan data terpusat, jalur pemrosesan data, mekanisme visualisasi, antarmuka pencarian, beban kerja simulasi, antarmuka analitik, dan dasbor bawaan. Dengan menggunakan ADDF, Anda dapat berbagi, memodifikasi, atau membuat modul yang dapat disesuaikan sepenuhnya yang mengurangi jumlah upaya yang diperlukan untuk membuat dan menerapkan solusi ini.

Panduan ini dimaksudkan untuk membantu Anda memahami praktik terbaik untuk menerapkan dan mengoperasikan ADDF secara aman diAWS Cloud. Ini membahas topik-topik berikut:

- Arsitektur dan terminologi— Tinjau arsitektur umum, alur kerja, dan istilah penting.
- Model tanggung jawab bersamaMemahami peran Anda dan peranAWSdalam mengamankan penerapan ADDF dan sumber daya cloud Anda.
- Proses peninjauan keamanan— Karena ADDF adalah proyek sumber terbuka, tinjau caranyaAWSdan kontributor menyelesaikan tinjauan keamanan.
- Fitur keamanan bawaan— Tinjau bagaimana praktik dan fitur terbaik keamanan dibangun ke dalam proyek sumber terbuka ADDF dan kerangka penerapannya.
- Pengaturan dan operasi yang aman— Pelajari cara menerapkan dan mengoperasikan ADDF diAWS Cloud.

Audiens yang dituju

Panduan ini ditujukan untuk Operasi Pengembangan (DevOps) tim, insinyur infrastruktur, administrator, staf keamanan TI, dan tim respons insiden yang ditugaskan untuk menilai,

menyebarkan, menyesuaikan, dan mengoperasikan ADDF. Anda dapat menerapkan rekomendasi dalam panduan ini untukproof-of-conceptatau lingkungan produksi.

Panduan ini mengasumsikan Anda tidak memiliki pengetahuan sebelumnya tentang ADDF. Namun, kami sarankan Anda membacaADDF readme(GitHub) sebelum melanjutkan.

Hasil bisnis yang ditargetkan

Panduan ini dirancang untuk membantu Anda mengatur dan mengoperasikan ADDF dengan lebih percaya diri dan aman di lingkungan pengembangan dan produksi.

Arsitektur dan terminologi ADDF

Sebelum Anda dapat memahami topik keamanan dan operasional dalam panduan ini, penting untuk memiliki pemahaman tingkat tinggi tentang terminologi, komponen, dan arsitektur Autonomous Driving Data Framework (ADDF). Bagian ini terdiri dari topik-topik berikut:

- Terminologi ADDF
- Arsitektur ADDF

Terminologi ADDF

Terminologi kunci untuk ADDF adalah sebagai berikut:

- Modul ADDFModul adalah infrastruktur sebagai kode (IAc) yang mengimplementasikan tugas umum dalam sistem bantuan pengemudi tingkat lanjut (ADAS). Tugas umum termasuk mengonfigurasi penyimpanan data terpusat, jaringan pemrosesan data, mekanisme visualisasi, antarmuka pencarian, beban kerja simulasi, antarmuka analitik, dan dasbor bawaan. Anda dapat membuat modul berdasarkan kebutuhan Anda, atau Anda dapat menggunakan kembali atau menyesuaikan modul yang ada.

  Anda dapat menggunakanAWS Cloud Development Kit (AWS CDK)untuk menentukan modul ADDF, atau Anda dapat menggunakan kerangka kerja IAc umum, seperti Hashicorp Terraform atauAWS CloudFormation, untuk mengimplementasikan modul ADDF. Modul memiliki seperangkat parameter input. Parameter input dapat bergantung pada nilai output dari modul lain. Modul ADDF adalah unit penyebaran terkecil untuk target ADDFAkun AWS.
- File manifes penerapan ADDF— File ini mendefinisikan orkestrasi modul ADDF mandiri. Orkestrasimengacu pada urutan penyebaran modul. Dalam file manifes penerapan ADDF, Anda dapat menggunakanGrup ADDFuntuk mengelompokkan modul terkait bersama-sama. Dalam file ini, Anda juga mendefinisikan toolchain ADDFAkun AWS, target ADDFAkun AWS, dan targetWilayah AWS.
- Kerangka penerapan ADDF- Kerangka kerja ini menyebarkan modul ADDF ke dalam target ADDFAkun AWSberdasarkan orkestrasi yang ditentukan dalam file manifes penerapan ADDF. Kerangka kerja penerapan ADDF diimplementasikan dengan menggunakan yang berikutAWSproyek sumber terbuka:

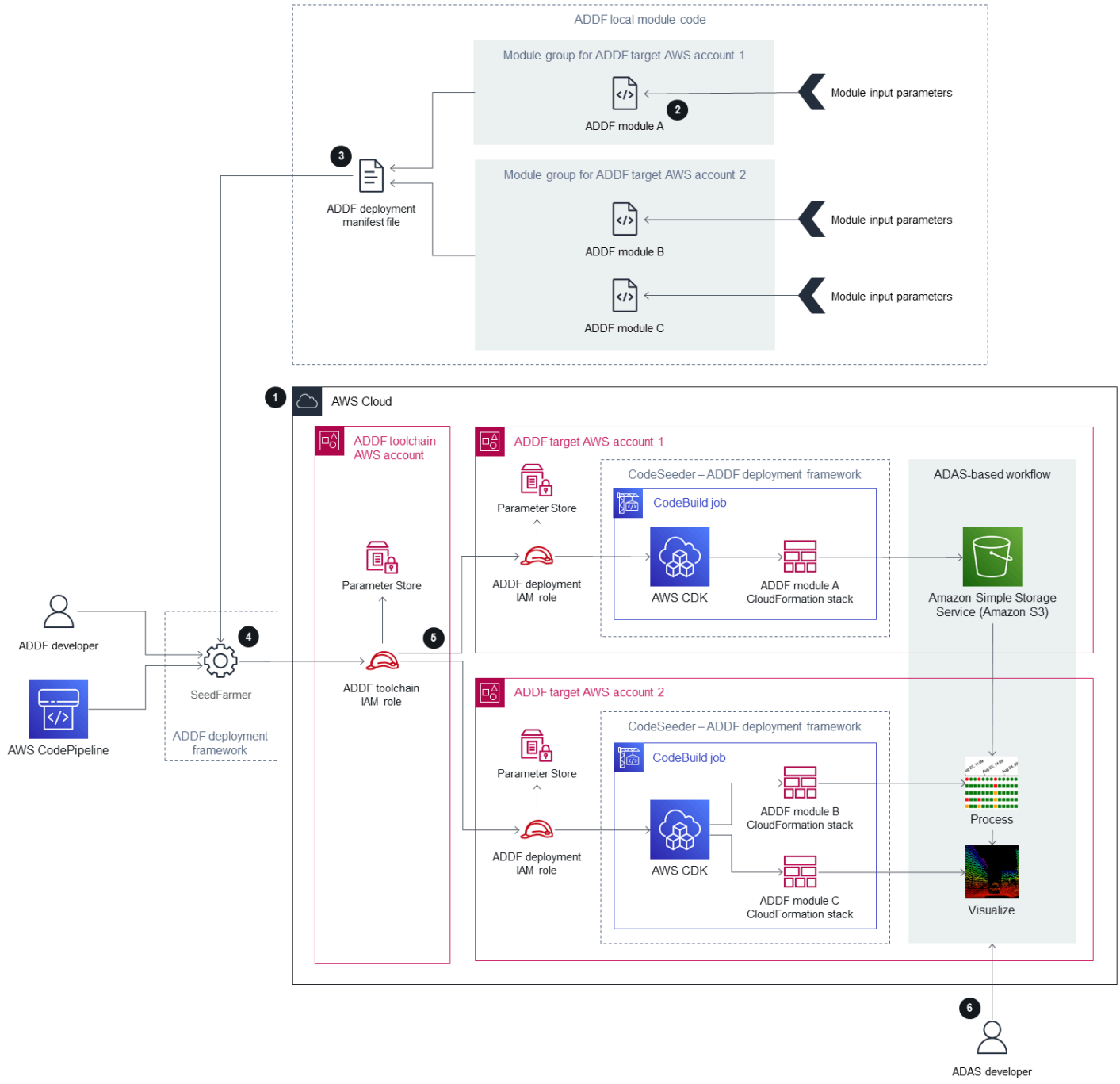
  - SeedFarmer(GitHub) —SeedFarmeradalah alat CLI yang digunakan untuk penerapan ADDF. Ini mengelola setiap status modul, menyiapkan dan mengemas kode modul, membuat kebijakan hak istimewa paling sedikit untuk peran penerapan ADDF, dan memberikan instruksi semantik yangCodeSeederdigunakan untuk penyebaran. Anda dapat berinteraksi langsung denganSeedFarmeruntuk menjalankan penerapan ADDF, atau Anda dapat mengintegrasikannya ke dalam pipeline integrasi berkelanjutan dan penerapan berkelanjutan (CI/CD).
  - CodeSeeder(GitHub) —CodeSeedermenyebarkan infrastruktur arbitrer sebagai paket kode melaluiAWS CodeBuildpekerjaan. SeedFarmersecara otomatis mengatur dan menjalankanCodeSeeder. HanyaSeedFarmerBerinteraksi langsung denganCodeSeeder.

  Kerangka penerapan ADDF dirancang untuk mendukung penerapan dalam arsitektur akun tunggal dan multi-akun. Berdasarkan persyaratan organisasi Anda, Anda memutuskan apakah arsitektur satu akun atau multi-akun diperlukan.

- Rantai alat ADDFAkun AWS— Akun ini mengatur dan mengelola penyebaran modul ke target ADDFAkun AWS, berdasarkan definisi dalam file manifes penerapan ADDF. Penerapan ADDF hanya dapat memiliki satu toolchain ADDFAkun AWS. Dalam arsitektur akun tunggal, rantai alat ADDFAkun AWSjuga merupakan target ADDFAkun AWS. Akun ini berisiAWS Identity and Access Management(IAM) peran, disebutPeran IAM rantai alat ADDF, yang diasumsikan olehSeedFarmerselama proses penerapan ADDF. Dalam panduan ini, kami merujuk ke toolchain ADDFAkun AWSsebagaiakun toolchain.
- Target ADDFAkun AWS— Ini adalah akun target tempat Anda menggunakan modul ADDF. Anda dapat memiliki satu atau lebih akun target. Akun-akun ini berisi sumber daya dan logika aplikasi yang dijelaskan dalam file manifes penerapan ADDF dan modul yang dipetakan. Dalam arsitektur akun tunggal, target ADDFAkun AWSjuga merupakan rantai alat ADDFAkun AWS. Setiap akun target ADDF berisi peran IAM, yang disebutPeran IAM penerapan ADDF, yang diasumsikan olehCodeSeederselama proses penyebaran. Dalam panduan ini, kami mengacu pada target ADDFAkun AWSsebagaiakun target.
- Contoh ADDF— Saat Anda menerapkan ADDF dan modul Anda di cloud, seperti yang didefinisikan dalam file manifes penerapan ADDF Anda, ini menjadiContoh ADDF. Instans ADDF dapat memiliki arsitektur akun tunggal atau multi-akun, dan Anda dapat menerapkan beberapa instans ADDF. Untuk informasi selengkapnya tentang memilih jumlah instans dan mendesain arsitektur akun untuk kasus penggunaan Anda, lihatMendefinisikan arsitektur ADDF Anda.

Arsitektur ADDF

Diagram berikut menunjukkan arsitektur tingkat tinggi untuk instance ADDF diAWS Cloud. Ini menunjukkan arsitektur multi-akun, termasuk akun toolchain khusus dan dua akun target. Panduan ini membahas proses end-to-end menggunakan ADDF untuk menyebarkan sumber daya ke akun target.



1. Buat dan bootstrap ADDFAkun AWS.

   Agar berfungsi dengan baik, setiap akun harus di-bootstrap ke ADDF danAWS CDK. Jika ini merupakan penerapan ADDF baru atau Anda menambahkan akun target baru, lakukan hal berikut:

   a. mengebutAWS CDKdi akun toolchain dan setiap akun target. Untuk instruksi, lihatBootstrapping(AWS CDKdokumentasi). ADDF menggunakanAWS CDKuntuk menyebarkan infrastrukturnya.

   b. Bootstrap ADDF di akun toolchain dan setiap akun target. Untuk instruksi, lihatmengebutAkun AWS(s)diPanduan Penerapan ADDF. Ini mengatur semua peran IAM khusus ADDF yang diperlukan olehSeedFarmerdanCodeSeeder.

   > **Note**
   >
   > Anda perlu melakukan langkah ini hanya jika Anda awalnya menerapkan ADDF atau menambahkan akun target baru. Langkah ini bukan bagian dari penerapan ADDF yang berulang ke instance ADDF yang sudah ada.

2. Buat atau sesuaikan modul ADDF.

   Buat atau sesuaikan modul ADDF berdasarkan masalah spesifik yang Anda coba selesaikan. Modul Anda harus mewakili tugas atau kelompok tugas yang terisolasi. Tentukan parameter input untuk modul sesuai kebutuhan, dan gunakan nilai output modul sebagai parameter input untuk modul lain.

3. Tentukan orkestrasi modul dalam file manifes penerapan ADDF.

   Dalam file manifes ADDF, atur modul ke dalam grup dan tentukan urutan penerapan dan dependensi di antara keduanya. Dalam file ini, Anda juga menentukan akun toolchain tunggal dan akun target (termasukWilayah AWS) untuk setiap grup ADDF dan modulnya.

4. Evaluasi file manifes penerapan ADDF dan buat cakupan penerapan.

   Pengembang ADDF atau pipa CI/CD, sepertiAWS CodePipeline, memulai evaluasi file manifes penerapan ADDF dengan memanggil alat CLI,SeedFarmer. Untuk memulai evaluasi:

   - SeedFarmermenggunakan file manifes penerapan ADDF sebagai parameter input untuk evaluasi.
   - Untuk mengasumsikan peran IAM toolchain ADDF,SeedFarmermengharapkan peran IAM yang sama dan valid atau kredensional pengguna yang ditentukan selama proses bootstrap ADDF, pada langkah 1.

JikaSeedFarmertidak memiliki kredensional yang benar untuk mengasumsikan peran IAM toolchain ADDF atau tidak dapat mengakses file manifes penerapan ADDF, evaluasi tidak dimulai.

JikaSeedFarmerdapat memulai evaluasi, ini mengasumsikan peran IAM toolchain ADDF di akun toolchain. Dari sana,SeedFarmerdapat mengakses akun target apa pun, dengan mengasumsikan peran IAM penerapan ADDF di akun itu. SeedFarmerkemudian mencoba membaca metadata ADDF apa pun di akun toolchain dan akun target. Salah satu hal berikut terjadi:

- Jika tidak ada metadata ADDF untuk dibaca, itu menunjukkan bahwa ini adalah contoh ADDF baru. SeedFarmermenentukan bahwa cakupan penerapan adalah seluruh file manifes penerapan ADDF dan isinya.
- Jika metadata ADDF ada,SeedFarmermembandingkan file manifes penerapan ADDF dan isinya dengan hash MD5 dari artefak yang digunakan yang ada di akun target. Jika perubahan yang dapat diterapkan terdeteksi, proses ini berlanjut. Jika tidak ada perubahan deployable yang terdeteksi, prosesnya selesai.

5. Terapkan modul ADDF dalam ruang lingkup ke akun target.

   CodeSeedersekarang memiliki daftar penerapan yang diurutkan untuk dijalankan, sesuai dengan file manifes penerapan ADDF dan hasil evaluasi dari langkah sebelumnya. Berdasarkan daftar yang diurutkan itu,CodeSeedermengasumsikan peran IAM penerapan ADDF di setiap akun target terkait. Kemudian berjalanCodeSeederdalam sebuahAWS CodeBuildpekerjaan untuk membuat atau memperbarui penerapan IAC individu, sepertiAWS CDKaplikasi, untuk modul ADDF. Secara default, ADDF menggunakanAWS CDKsebagai kerangka kerja IAC-nya, tetapi kerangka kerja IAc umum lainnya juga didukung. Setelah proses selesai untuk setiap akun target, Anda memiliki alur kerja berbasis Adas end-to-end yang sepenuhnya diterapkan, lintas akun, dan end-to-end, seperti yang Anda tentukan dalam file manifes penerapan ADDF.

   Jika Anda menggunakan arsitektur akun tunggal, akun toolchain dan akun target adalah akun yang sama, dan satu akun memiliki semua fungsi yang dijelaskan.

6. Gunakan infrastruktur yang digunakan ADDF.

   Pengembang ADAS dapat menggunakan alur kerja berbasis ADAS yang diterapkan, seperti yang ditentukan oleh kasus penggunaan Anda.

   Alur kerja ini menjelaskan arsitektur satu contoh lingkungan multi-akun ADDF. Bergantung pada model pengembangan, penerapan, dan operasi Anda, sebaiknya Anda menjalankan beberapa instans ADDF di lingkungan multi-tahap. Pengaturan tipikal mungkin menyertakan instance

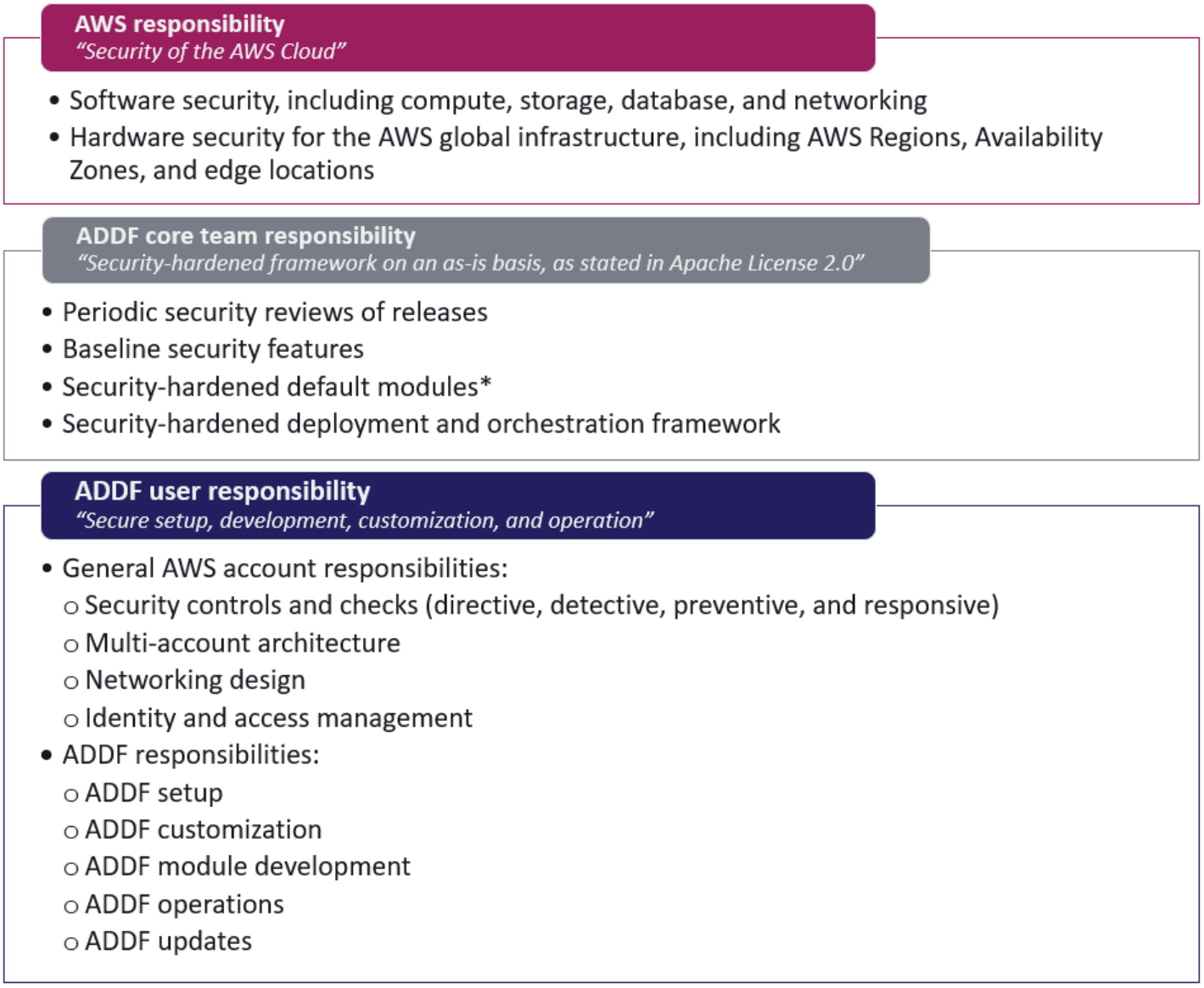
ADDF khusus dengan dedicatedAkun AWSuntuk setiap tahap penyebaran, seperti cabang untuk pengembangan, pengujian, dan produksi. Anda juga dapat menjalankan beberapa instans ADDF di lingkungan satu akun atau multi-akun yang sama di lingkungan yang samaWilayah AWS, dengan asumsi bahwa Anda membuat namespace sumber daya unik untuk setiap instance ADDF. Untuk informasi selengkapnya, lihat Mendefinisikan arsitektur ADDF Anda.

Model tanggung jawab bersama ADDF

Themodel tanggung jawab bersamayang berlaku untukLayanan AWSjuga berlaku untuk Autonomous Driving Data Framework (ADDF). Entitas berikut berbagi tanggung jawab untuk mengamankan ADDF sebagaimana diatur dalam diagram berikut:

- AWS— Penyedia infrastruktur cloud menawarkanLayanan AWS.
- Tim inti ADDF— Tim inti ADDF adalah entitas yang menerbitkan rilis ADDF diRepositori ADDF(GitHub).
- Pengguna ADDF— Pengguna ADDF termasuk, tetapi tidak terbatas pada:
  - Pengembang ADDF— Siapa pun yang mengubah, menyesuaikan, atau membuat kode modul ADDF baru.
  - Operator ADDF— Siapa pun yang mengatur dan mengoperasikan instance ADDF.
  - Pengembang ADAS— Pengguna akhir atau konsumen sumber daya yang digunakan oleh ADDF. Misalnya, pengembang ADAS dapat menanyakan frontend visualisasi yang dibuat sebagai bagian dari penerapan ADDF.

Diagram berikut merangkum tanggung jawab bersama antaraAWS, tim inti ADDF, dan pengguna ADDF.



* Excluding any modules in the ADDF **/modules/demo-only/** folder. Those modules exist only for proof-of-concept purposes and didn't receive security hardening.

AWStanggung jawab

AWSbertanggung jawab untuk melindungi infrastruktur yang menjalankan semua layanan yang ditawarkan diAWS Cloud, sebagaimana didefinisikan dalamAWSmodel tanggung jawab bersama. Infrastruktur ini terdiri dari perangkat keras, perangkat lunak, jaringan, dan fasilitas yang berjalanAWS Cloudlayanan.

Tanggung jawab tim inti ADDF

Tim inti ADDF menyediakan kerangka kerja yang aman dengan sendirinya, berdasarkan upaya terbaik, menurutLisensi Apache 2.0(GitHub). Tim inti ADDF bertanggung jawab untuk hal-hal berikut:

- Tinjauan keamanan berkala dari rilis
- Fitur keamanan dasar
- Modul default yang diperkeras keamanan (Ini tidak termasuk modul apa pun di`modules/demo-only/`folder. Modul-modul tersebut hanya untukproof-of-concepttujuan dan tidak menerima pengerasan keamanan.)
- Kerangka penerapan dan orkestrasi yang diperkeras keamanan

Tanggung jawab keamanan ini hanya mencakup kerangka kerja, sebagaimana diatur dalamGitHubrepositori, tanpa modifikasi atau kustomisasi. Ini mencakup semua modul ADDF, kecuali modul ADDF di`modules/demo-only/`folder. Modul ADDF di folder ini tidak diperkeras keamanan dan tidak boleh digunakan di lingkungan produksi atau di lingkungan apa pun dengan data sensitif atau terlindungi. Modul-modul ini disertakan untuk menampilkan kemampuan sistem, dan Anda dapat menggunakannya sebagai dasar untuk membuat modul yang disesuaikan dan diperkuat keamanan Anda sendiri.

> **Note**
>
> ADDF sebagai kerangka kerja disampaikan atas dasar apa adanya. Itu tidak datang dengan tanggung jawab dan garansi apa pun, sebagaimana dinyatakan dalamLisensi Apache 2.0(GitHub). Anda harus melakukan penilaian keamanan Anda sendiri terhadap ADDF dan memverifikasi bahwa ADDF sesuai dengan persyaratan keamanan spesifik organisasi Anda.

Tanggung jawab pengguna ADDF

ADDF dan modul-modulnya aman hanya jika ADDF diatur, disesuaikan, dan dioperasikan dengan cara yang aman. Pengguna ADDF bertanggung jawab penuh atas keamanan berikut ini:

- UmumAkun AWStanggung jawab:
  - Kontrol dan pemeriksaan keamanan (direktif, detektif, preventif, dan responsif)
  - Arsitektur multi-akun

  - Desain jaringan
  - Manajemen identitas dan akses
- Tanggung jawab khusus ADDF:
  - Pengaturan ADDF
  - Kustomisasi ADDF
  - Pengembangan modul ADDF
  - Operasi ADDF
  - Pembaruan ADDF

UmumAkun AWStanggung jawab

Sebelum Anda menyebarkan sumber daya terkait ADDF keAkun AWS, AndaAkun AWSharus dikonfigurasi sesuai dengan praktik terbaik diAWSKerangka Kerja yang Dirancang dengan Baik. Ini termasuk kontrol keamanan direktif, detektif, preventif, dan responsif. Anda harus memiliki proses mitigasi terperinci, jika terjadi pelanggaran atau insiden keamanan. Kebijakan organisasi Anda harus mencakup persyaratan untuk mengelola identitas dan akses dan jaringan secara terpusat. Umumnya, persyaratan dan layanan ini ditangani oleh tim zona pendaratan khusus.

Tanggung jawab khusus ADDF

Pengaturan ADDF yang aman

Tanggung jawab pengguna ADDF dimulai dengan pengaturan ADDF yang aman sesuai dengan dokumentasi ADDF. Kami sangat menyarankan agar Anda mengikuti instruksi diPanduan Penerapan ADDF(GitHub). Untuk informasi selengkapnya tentang pengaturan ADDF dengan aman, lihatMendefinisikan arsitektur ADDF AndadanPengaturan awal.

Kustomisasi ADDF yang aman

Dalam hal kustomisasi fungsionalitas inti ADDF, sepertiCodeSeeder,SeedFarmer, dan modul inti ADDF, pengguna ADDF memikul tanggung jawab penuh atas perubahan tersebut. Untuk informasi selengkapnya, lihat Menyesuaikan kode kerangka kerja penerapan ADDF.

Pengembangan modul ADDF yang aman

Pengguna ADDF bertanggung jawab penuh atas modul kustom apa pun yang digunakan menggunakan ADDF. Selain itu, pengguna ADDF bertanggung jawab atas perubahan kode apa pun

pada modul yang disediakan ADDF. Untuk informasi selengkapnya, lihat Menulis modul khusus di ADDF.

Pembaruan dan operasi ADDF yang aman

Seiring berkembangnya kerangka kerja, ADDF menerima pembaruan fitur dan keamanan. Ini adalah tanggung jawab pengguna ADDF untuk secara teratur memeriksa pembaruan yang dipublikasikan keGitHubrepositori dan untuk mengoperasikan ADDF dengan aman dalam jangka panjang. Untuk informasi lebih lanjut, lihat Penerapan ADDF yang berulang, Audit keamanan berulang, Pembaruan ADDF, dan Penonaktifan .

Proses peninjauan keamanan ADDF

Autonomous Driving Data Framework (ADDF) dibangun dengan mempertimbangkan keamanan. Sebelum dirilis ke publik,AWSmelakukan tinjauan keamanan internal awal terhadap ADDF dan menyelesaikan masalah keamanan yang teridentifikasi. KeduanyaAWSdan komunitas open-source berkontribusi pada tinjauan keamanan kerangka kerja yang sedang berlangsung.

Ulasan keamanan reguler olehAWS

ADDF diterbitkan di bawah awslabsGitHubOrganisasi yang dimiliki olehAWS.AWSmelakukan tinjauan keamanan otomatis dan manual reguler dari kode dalam organisasi ini, untuk memverifikasi keamanan dengan upaya terbaik. MenurutAWSkebijakan,AWStidak mengungkapkan informasi tentang frekuensi tinjauan keamanan, pendekatan, atau alat yang digunakan. Selanjutnya,AWStidak mempublikasikan laporan audit internal apa pun tentang ADDF. Namun, setiap temuan keamanan yang diidentifikasi diperbaiki dan dipublikasikan melalui permintaan tarik, dengan urgensi tinggi.

> **Note**
>
> ADDF sebagai kerangka kerja disampaikan atas dasar 'APA ADANYA', TANPA JAMINAN ATAU KETENTUAN APA PUN, baik tersurat maupun tersirat, termasuk namun tidak terbatas pada, jaminan atau ketentuan kepemilikan, non-pelanggaran, merchant, atau kesesuaian untuk tujuan tertentu, sebagaimana dinyatakan dalamLisensi Apache 2.0(GitHub). Anda harus melakukan penilaian keamanan Anda sendiri terhadap ADDF dan memverifikasi apakah itu sesuai dengan persyaratan keamanan spesifik organisasi Anda dan, sebagaimana ditetapkan dalam Apache License 2.0, Anda bertanggung jawab penuh untuk menentukan kesesuaian penggunaan atau pendistribusian ulang ADDF dan menanggung risiko apa pun yang terkait dengan latihan atau izin Anda berdasarkan lisensi tersebut.

Tinjauan dan kontribusi keamanan sumber terbuka

ADDF adalah proyek sumber terbuka yang menyambut kontribusi. Kami mengundang semua pengguna untuk melakukan tinjauan keamanan mereka sendiri terhadap kerangka kerja dan berkontribusi dengan melaporkan temuan terkait keamanan. Jika Anda menemukan masalah dalam kode, silakan ikuti pedoman diPemberitahuan masalah keamanan(Dokumentasi ADDF).

Fitur keamanan bawaan ADDF

Autonomous Driving Data Framework (ADDF) memiliki berbagai fitur keamanan bawaan. Secara default, fitur-fitur ini dirancang untuk membantu Anda menyiapkan kerangka kerja yang aman dan membantu organisasi Anda memenuhi persyaratan keamanan perusahaan umum.

Berikut ini adalah fitur keamanan bawaan:

- Hak istimewa paling sedikit untuk kode modul ADDF
- Infrastruktur sebagai kode
- Pemeriksaan keamanan otomatis untuk IAc
- Kebijakan hak istimewa paling tidak khusus untukAWS CDKperan penyebaran
- Kebijakan hak istimewa paling rendah untuk file deployspec modul
- Enkripsi data
- Penyimpanan kredensi menggunakan Secrets Manager
- Ulasan keamananSeedFarmerdanCodeSeeder
- Dukungan batas izin untukAWS CodeBuildperan untukCodeSeeder
- AWSarsitektur multi-akun
- Izin hak istimewa paling sedikit untuk penerapan multi-akun

Hak istimewa paling sedikit untuk kode modul ADDF

Keistimewaan paling sedikitadalah praktik terbaik keamanan untuk memberikan izin minimum yang diperlukan untuk melakukan tugas. Untuk informasi lebih lanjut, lihatTerapkan izin hak istimewa paling sedikit. Modul yang disediakan ADDF secara ketat mengikuti prinsip hak istimewa paling sedikit dalam kode dan sumber daya yang digunakan, sebagai berikut:

- SemuaAWS Identity and Access Management(IAM) kebijakan yang dihasilkan untuk modul ADDF memiliki izin minimum yang diperlukan untuk kasus penggunaan.
- Layanan AWSdikonfigurasi dan digunakan sesuai dengan prinsip hak istimewa paling sedikit. Modul yang disediakan ADDF hanya menggunakan layanan dan fitur layanan yang diperlukan untuk kasus penggunaan tertentu.

Infrastruktur sebagai kode

ADDF, sebagai kerangka kerja, dirancang untuk menyebarkan modul ADDF sebagai infrastruktur sebagai kode (IAc). IAC menghilangkan proses penerapan manual dan membantu mencegah kesalahan dan kesalahan konfigurasi, yang dapat dihasilkan dari proses manual.

ADDF dirancang untuk mengatur dan menyebarkan modul dengan menggunakan kerangka kerja IAc umum. Ini termasuk, tetapi tidak terbatas pada:

- AWS Cloud Development Kit (AWS CDK)
- AWS CloudFormation
- Hashicorp Terraform

Anda dapat menggunakan kerangka kerja IAc yang berbeda untuk menulis modul yang berbeda, dan kemudian Anda menggunakan ADDF untuk menerapkannya.

Kerangka kerja IAC default yang digunakan oleh modul ADDF adalahAWS CDK.AWS CDKadalah abstraksi berorientasi objek tingkat tinggi yang dapat Anda gunakan untuk mendefinisikanAWSsumber daya imperatif.AWS CDKsudah memberlakukan praktik terbaik keamanan secara default untuk berbagai layanan dan skenario. Dengan menggunakanAWS CDK, risiko kesalahan konfigurasi keamanan berkurang.

Pemeriksaan keamanan otomatis untuk IAc

Sumber terbukacdk-agutilitas (GitHub) diintegrasikan ke dalam ADDF. Utilitas ini secara otomatis memeriksa modul ADDF yang didasarkan padaAWS CDKuntuk kepatuhan terhadap praktik terbaik umum dan keamanan. Utilitas cdk-nag menggunakan aturan dan paket aturan untuk mendeteksi dan melaporkan kode yang melanggar praktik terbaik. Untuk informasi selengkapnya tentang aturan dan daftar lengkap, lihataturan cdk-nag(GitHub).

Kebijakan hak istimewa paling tidak khusus untukAWS CDKperan penyebaran

ADDF memanfaatkan secara ekstensifAWS CDKv2. Hal ini diperlukan bahwa Anda bootstrap semua ADDFAkun AWSkepadaAWS CDK. Untuk informasi lebih lanjut, lihatBootstrapping(AWS CDKdokumentasi).

Secara default,AWS CDKmenugaskan permisifAdministratorAccess AWSkebijakan yang dikelola untukAWS CDKperan penerapan yang dibuat di akun bootstrapped. Nama lengkap dari peran ini adalah`cdk-[CDK_QUALIFIER]-cfn-exec-role-[AWS_ACCOUNT_ID]-[REGION]`.AWS CDKmenggunakan peran ini untuk menyebarkan sumber daya ke dalam bootstrapAkun AWSsebagai bagian dariAWS CDKproses penyebaran.

Tergantung pada persyaratan keamanan organisasi Anda,`AdministratorAccess`Kebijakan mungkin terlalu permisif. Sebagai bagian dariAWS CDKproses bootstrap, Anda dapat menyesuaikan kebijakan dan izin sesuai dengan kebutuhan Anda. Anda dapat mengubah kebijakan dapat dengan melakukan bootstrap ulang akun dengan kebijakan yang baru ditentukan dengan menggunakan`--cloudformation-execution-policies`parameter. Untuk informasi lebih lanjut, lihatMenyesuaikan bootstrap(AWS CDKdokumentasi).

> **Note**
>
> Meskipun fitur keamanan ini tidak spesifik untuk ADDF, fitur ini tercantum di bagian ini karena dapat meningkatkan keamanan keseluruhan penerapan ADDF Anda.

Kebijakan hak istimewa paling rendah untuk file deployspec modul

Setiap modul berisi file spesifikasi penyebaran yang disebutdeployspec.yaml. File ini mendefinisikan instruksi penyebaran untuk modul. CodeSeedermenggunakannya untuk menyebarkan modul yang ditentukan di akun target dengan menggunakanAWS CodeBuild. CodeSeedermenetapkan peran layanan default keCodeBuilduntuk menyebarkan sumber daya, seperti yang diinstruksikan dalam file spesifikasi penerapan. Peran layanan ini dirancang sesuai dengan prinsip hak istimewa paling rendah. Ini mencakup semua izin yang diperlukan untuk menyebarkanAWS CDKaplikasi, karena semua modul yang disediakan ADDF dibuat sebagaiAWS CDKaplikasi.

Namun, jika Anda perlu menjalankan perintah panggung apa pun di luarAWS CDK, Anda perlu membuat kebijakan IAM khusus alih-alih menggunakan peran layanan default untukCodeBuild. Misalnya, jika Anda menggunakan kerangka kerja penyebaran IAC selainAWS CDK, seperti Terraform, Anda perlu membuat kebijakan IAM yang memberikan izin yang cukup agar kerangka kerja tertentu berfungsi. Skenario lain yang memerlukan kebijakan IAM khusus adalah ketika Anda menyertakan directAWS Command Line Interface(AWS CLI) panggilan ke orang lainLayanan AWSdi`install`,`pre_build`,`build`, atau`post_build`perintah panggung. Misalnya, Anda memerlukan kebijakan khusus jika modul menyertakan perintah Amazon Simple Storage Service

(Amazon S3) untuk mengunggah file ke bucket S3. Kebijakan IAM kustom memberikan kontrol halus untuk apa punAWSperintah di luarAWS CDKpenyebaran. Untuk contoh kebijakan IAM kustom, lihatModuleStack(SeedFarmerdokumentasi). Saat membuat kebijakan IAM khusus untuk modul ADDF Anda, pastikan Anda menerapkan izin hak istimewa paling sedikit.

Enkripsi data

ADDF menyimpan dan memproses data yang berpotensi sensitif. Untuk membantu melindungi data ini,SeedFarmer,CodeSeeder, dan modul yang disediakan ADDF mengenkripsi data saat istirahat dan dalam perjalanan untuk semua yang digunakanLayanan AWS(kecuali secara eksplisit dinyatakan sebaliknya untuk modul di`demo-only`folder).

Penyimpanan kredensi menggunakan Secrets Manager

ADDF menangani berbagai rahasia untuk layanan yang berbeda, seperti Docker Hub,JupyterHub, danPergeseran Merah Amazon. ADDF menggunakanAWS Secrets Manageruntuk menyimpan rahasia terkait ADDF. Ini membantu Anda menghapus data sensitif dari kode sumber.

Rahasia Manajer Rahasia disimpan hanya di akun target, sesuai kebutuhan agar akun itu berfungsi dengan baik. Secara default, akun toolchain tidak berisi rahasia apa pun.

Ulasan keamananSeedFarmerdanCodeSeeder

SeedFarmerdanCodeSeeder(GitHubrepositori) digunakan untuk menyebarkan ADDF dan modul ADDF-nya. Proyek-proyek open-source ini mengalami reguler yang samaAWSproses peninjauan keamanan internal sebagai ADDF, seperti yang dijelaskan dalamProses peninjauan keamanan ADDF.

Dukungan batas izin untukAWS CodeBuildperan untukCodeSeeder

IAMbatas izinadalah mekanisme keamanan umum yang mendefinisikan izin maksimum yang dapat diberikan oleh kebijakan berbasis identitas kepada entitas IAM. SeedFarmerdanCodeSeedermendukung lampiran batas izin IAM untuk setiap akun target. Batas izin membatasi izin maksimum dari setiap peran layanan yang digunakan oleh CodeBuild ketika CodeSeeder menyebarkan modul. Batas izin IAM harus dibuat di luar ADDF oleh tim

keamanan. Lampiran kebijakan batas izin IAM diterima sebagai atribut dalam file manifes penerapan ADDF,deployment.yaml. Untuk informasi lebih lanjut, lihatDukungan batas izin(SeedFarmerdokumentasi).

Alur kerjanya adalah sebagai berikut:

1. Tim keamanan Anda menentukan dan membuat batas izin IAM sesuai dengan persyaratan keamanan Anda. Batas izin IAM harus dibuat secara individual di setiap ADDFAkun AWS. Outputnya adalah daftar kebijakan batas izin Amazon Resource Name (ARN).
2. Tim keamanan membagikan daftar ARN kebijakan dengan tim pengembang ADDF Anda.
3. Tim pengembang ADDF mengintegrasikan daftar ARN kebijakan ke dalam file manifes. Untuk contoh integrasi ini, lihatsampel-permissionboundary.yaml(GitHub) danManifes penyebaran(SeedFarmerdokumentasi).
4. Setelah penerapan berhasil, batas izin dilampirkan ke semua peran layanan yangCodeBuilddigunakan untuk menyebarkan modul.
5. Tim keamanan memantau bahwa batas izin diterapkan sesuai kebutuhan.

AWSarsitektur multi-akun

Seperti yang didefinisikan dalam pilar keamananAWSWell-Architected Framework, dianggap praktik terbaik untuk memisahkan sumber daya dan beban kerja menjadi beberapaAkun AWS, berdasarkan kebutuhan organisasi Anda. Hal ini karena sebuahAkun AWSbertindak sebagai batas isolasi. Untuk informasi lebih lanjut, lihatAkun AWSmanajemen dan pemisahan. Implementasi konsep ini disebut aarsitektur multi-akun. Didesain dengan baikAWSarsitektur multi-akun menyediakan kategorisasi beban kerja dan mengurangi cakupan dampak jika terjadi pelanggaran keamanan, dibandingkan dengan arsitektur akun tunggal.

ADDF secara native mendukungAWSarsitektur multi-akun. Anda dapat mendistribusikan modul ADDF Anda di banyakAkun AWSsesuai kebutuhan untuk keamanan organisasi Anda danseparation-of-dutiespersyaratan. Anda dapat menerapkan ADDF menjadi satuAkun AWS, menggabungkan fungsi toolchain dan akun target. Atau, Anda dapat membuat akun target individual untuk modul ADDF atau grup modul.

Satu-satunya batasan yang perlu Anda pertimbangkan adalah bahwa modul ADDF mewakili unit penerapan terkecil untuk masing-masingAkun AWS.

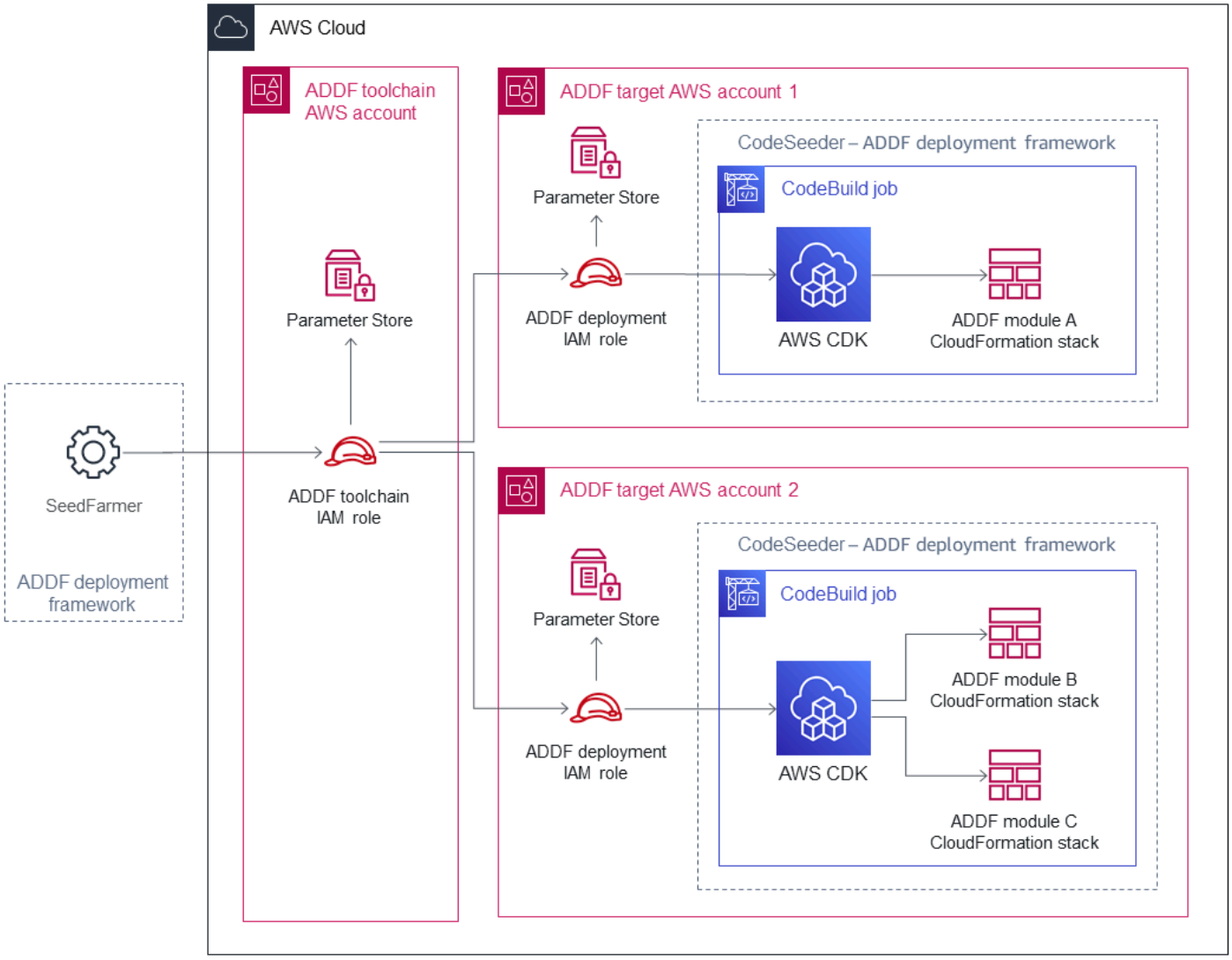
Untuk lingkungan produksi, Anda disarankan untuk menggunakan arsitektur multi-akun yang terdiri dari akun toolchain dan setidaknya satu akun target. Untuk informasi selengkapnya, lihat Arsitektur ADDF.

Izin hak istimewa paling sedikit untuk penerapan multi-akun

Jika Anda menggunakan arsitektur multi-akun,SeedFarmerperlu mengakses akun target untuk melakukan tiga tindakan berikut:

1. Tulis metadata modul ADDF ke akun toolchain dan akun target.
2. Baca metadata modul ADDF saat ini dari akun toolchain dan akun target.
3. MemulaiAWS CodeBuildpekerjaan di akun target, untuk tujuan menyebarkan atau memperbarui modul.

Gambar berikut menunjukkan hubungan lintas akun, termasuk operasi untuk mengasumsikan ADDF spesifikAWS Identity and Access Management(IAM) peran.



Tindakan lintas akun ini dicapai dengan menggunakan operasi peran asumsi yang terdefinisi dengan baik.

- Peran IAM toolchain ADDF diterapkan di akun toolchain. SeedFarmermengasumsikan peran ini. Peran ini memiliki izin untuk melakukan`iam:AssumeRole`tindakan dan dapat mengasumsikan peran IAM penerapan ADDF di setiap akun target. Selain itu, peran IAM toolchain ADDF dapat dijalankan secara lokalAWS Systems ManagerOperasi Parameter Store.
- Peran IAM penerapan ADDF diterapkan di setiap akun target. Peran ini hanya dapat diasumsikan dari akun toolchain dengan menggunakan peran IAM toolchain ADDF. Peran ini memiliki izin untuk menjalankan lokalAWS Systems ManagerOperasi Parameter Store dan memiliki izin untuk dijalankanAWS CodeBuildtindakan yang memulai dan mendeskripsikanCodeBuildpekerjaan melaluiCodeSeeder.

Peran IAM khusus ADDF ini dibuat sebagai bagian dari proses ADDF-Bootstrapping. Untuk informasi lebih lanjut, lihatmengebutAkun AWS(s)diPanduan Penerapan ADDF(GitHub).

Semua izin lintas akun diatur sesuai dengan prinsip hak istimewa paling sedikit. Jika satu akun target dikompromikan, ada dampak minimal atau tidak ada pada ADDF lainnyaAkun AWS.

Dalam kasus arsitektur akun tunggal untuk ADDF, hubungan peran tetap sama. Mereka hanya runtuh menjadi satuAkun AWS.

Pengaturan dan operasi aman ADDF

Autonomous Driving Data Framework (ADDF) harus diperlakukan sebagai perangkat lunak khusus yang membutuhkan pemeliharaan dan perawatan berkelanjutan oleh yang berdedikasiDevOpsdan tim keamanan di organisasi Anda. Bagian ini menjelaskan tugas-tugas umum terkait keamanan yang membantu Anda mengatur dan mengoperasikan ADDF sepanjang siklus hidupnya.

Bagian ini mencakup tugas-tugas berikut:

- Mendefinisikan arsitektur ADDF Anda
- Pengaturan awal
- Menyesuaikan kode kerangka kerja penerapan ADDF
- Menulis modul khusus di ADDF
- Penerapan ADDF yang berulang
- Audit keamanan berulang
- Pembaruan ADDF
- Penonaktifan

Mendefinisikan arsitektur ADDF Anda

Instans ADDF hanya seamanAkun AWSlingkungan tempat itu digunakan. IniAkun AWSlingkungan harus dirancang untuk memenuhi kebutuhan keamanan dan operasional kasus penggunaan spesifik Anda. Misalnya, tugas dan pertimbangan terkait keamanan dan operasi untuk menyiapkan instance ADDF diproof-of-concept(PoC) lingkungan berbeda dari lingkungan untuk menyiapkan ADDF di lingkungan produksi.

Menjalankan ADDF di lingkungan PoC

Jika Anda bermaksud menggunakan ADDF di lingkungan PoC, kami sarankan Anda membuat khususAkun AWSuntuk ADDF yang tidak mengandung beban kerja lainnya. Ini membantu menjaga akun Anda tetap aman saat Anda menjelajahi ADDF dan fitur-fiturnya. Berikut ini adalah manfaat dari pendekatan ini:

- Jika terjadi kesalahan konfigurasi ADDF yang parah, tidak ada beban kerja lain yang akan terpengaruh.

- Tidak ada risiko kesalahan konfigurasi beban kerja lainnya yang dapat mempengaruhi pengaturan ADDF.

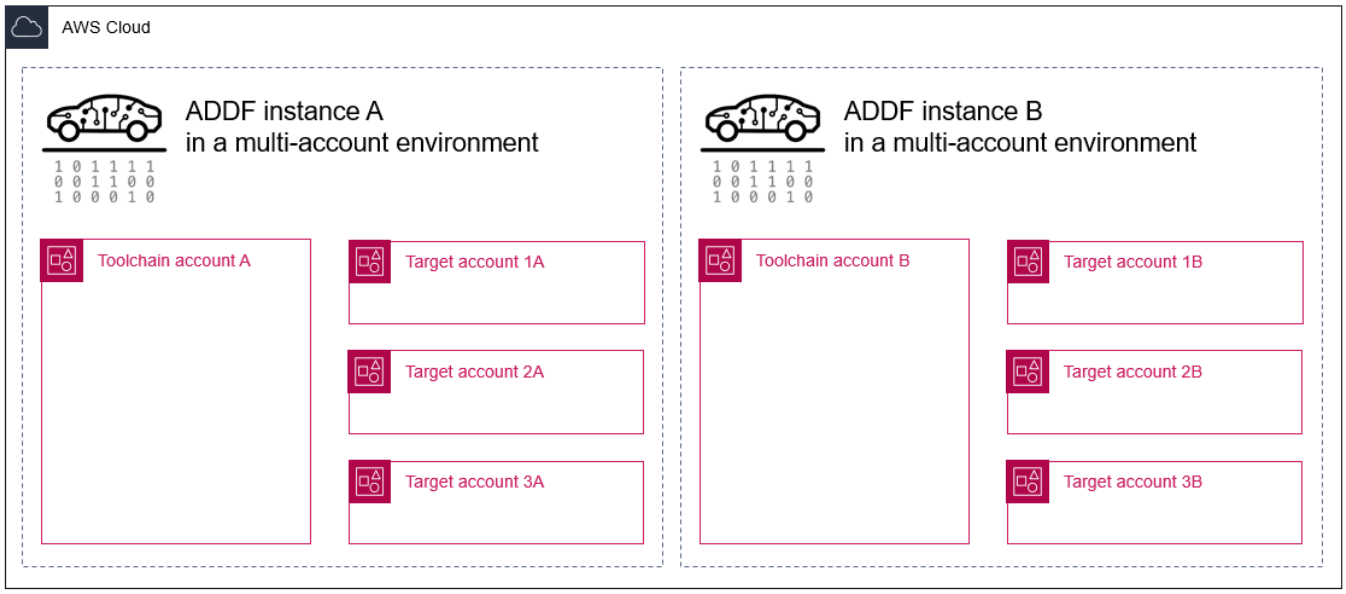
Bahkan untuk lingkungan PoC, kami tetap menyarankan Anda mengikuti sebanyak mungkin praktik terbaik yang tercantum diMenjalankan ADDF di lingkungan produksisebisa mungkin.

Menjalankan ADDF di lingkungan produksi

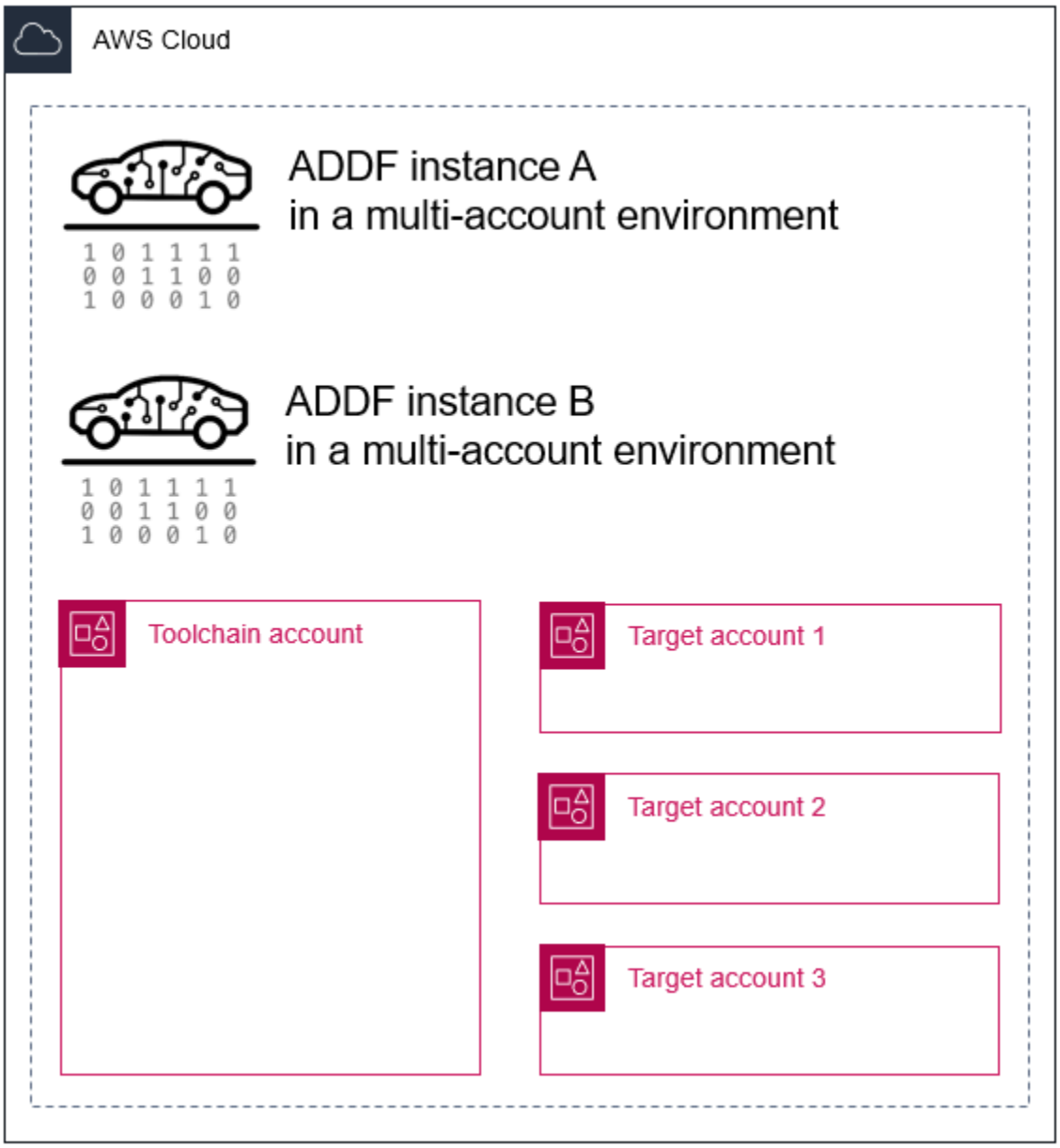
Jika Anda bermaksud menggunakan ADDF di lingkungan produksi perusahaan, kami sangat menyarankan Anda mempertimbangkan praktik terbaik keamanan organisasi Anda dan menerapkan ADDF sesuai dengan itu. Selain praktik terbaik keamanan organisasi Anda, kami menyarankan Anda menerapkan hal berikut:

- Buat ADDF jangka panjang dan berkomitmenDevOpstim— ADDF perlu diperlakukan sebagai perangkat lunak khusus. Ini membutuhkan pemeliharaan dan perawatan berkelanjutan oleh yang berdedikasiDevOpstim. Sebelum mulai menjalankan ADDF di lingkungan produksi, aDevOpstim dengan ukuran dan kemampuan yang memadai harus didefinisikan dengan komitmen sumber daya penuh, sampai end-of-lifedari penyebaran ADDF.
- Gunakan arsitektur multi-akun- Setiap instans ADDF harus digunakan di dedikasinya sendiriAWSlingkungan multi-akun, tanpa beban kerja lain yang tidak terkait. Seperti yang didefinisikan dalamAWSmanajemen akun dan pemisahan(AWSWell-Architected Framework), dianggap praktik terbaik untuk memisahkan sumber daya dan beban kerja menjadi beberapaAkun AWS, berdasarkan kebutuhan organisasi Anda. Hal ini karena sebuahAkun AWSbertindak sebagai batas isolasi. Didesain dengan baikAWSarsitektur multi-akun menyediakan kategorisasi beban kerja dan mengurangi cakupan dampak jika terjadi pelanggaran keamanan, dibandingkan dengan arsitektur akun tunggal. Menggunakan arsitektur multi-akun juga membantu akun Anda tetap berada di dalamnyaLayanan AWSkuota. Mendistribusikan modul ADDF Anda di sebanyak mungkinAkun AWSsesuai kebutuhan untuk memenuhi keamanan organisasi Anda danseparation-of-dutiespersyaratan.
- Menyebarkan beberapa instance ADDF— Siapkan sebanyak mungkin instance ADDF terpisah yang Anda butuhkan untuk mengembangkan, menguji, dan menerapkan modul ADDF dengan benar sesuai dengan proses pengembangan perangkat lunak organisasi Anda. Saat menyiapkan beberapa instans ADDF, Anda dapat menggunakan salah satu pendekatan berikut:
  - Beberapa instance ADDF berbedaAWSlingkungan multi-akun— Anda dapat menggunakan terpisahAkun AWSuntuk mengisolasi instance ADDF yang berbeda. Misalnya, jika organisasi Anda memiliki tahapan pengembangan, pengujian, dan produksi khusus khusus, Anda dapat

membuat instans ADDF terpisah dan akun khusus untuk setiap tahap. Ini memberikan banyak manfaat, seperti mengurangi risiko penyebaran kesalahan di seluruh tahapan, membantu Anda menerapkan proses persetujuan, dan membatasi akses pengguna hanya ke lingkungan tertentu. Gambar berikut menunjukkan dua instance ADDF yang diterapkan di lingkungan multi-akun yang terpisah.



- Beberapa instance ADDF dalam hal yang samaAWSlingkungan multi-akun— Anda dapat membuat beberapa instance ADDF yang berbagi hal yang samaAWSlingkungan multi-akun. Ini secara efektif menciptakan cabang yang terisolasi dalam hal yang samaAkun AWS. Misalnya, jika pengembang yang berbeda bekerja secara paralel, pengembang dapat membuat instance ADDF khusus dalam hal yang samaAkun AWS. Ini membantu pengembang bekerja di cabang yang terisolasi untuk tujuan pengembangan dan pengujian. Jika Anda menggunakan pendekatan ini, untuk setiap instance ADDF, sumber daya ADDF Anda harus memiliki nama sumber daya yang unik. Ini didukung dalam modul pra-suplai ADDF secara default. Anda dapat menggunakan pendekatan ini selama Anda tidak melebihiLayanan AWSkuota. Gambar berikut menunjukkan dua instance ADDF yang diterapkan di lingkungan multi-akun bersama.



- Beberapa instance ADDF dalam hal yang samaAWSlingkungan akun tunggalArsitektur ini sangat mirip dengan contoh sebelumnya. Perbedaannya adalah bahwa beberapa instance ADDF diterapkan dalam lingkungan akun tunggal alih-alih lingkungan multi-akun. Arsitektur ini dapat memuat kasus penggunaan ADDF yang sangat sederhana yang memiliki cakupan yang sangat terbatas dan banyak pengembang yang bekerja pada cabang yang berbeda pada saat yang bersamaan.

KarenaSeedFarmeradalah alat tunggal yang mengontrol penerapan untuk instance ADDF, Anda dapat membangun lingkungan dan arsitektur akun apa pun yang sesuai dengan strategi penerapan organisasi Anda dan proses CI/CD.

- SesuaikanAWS Cloud Development Kit (AWS CDK)proses bootstrap sesuai dengan persyaratan keamanan organisasi Anda— Secara default,AWS CDKmenugaskanAdministratorAccess AWSkebijakan terkelola selama proses bootstrap. Kebijakan ini memberikan hak administratif penuh. Jika kebijakan ini terlalu permisif untuk persyaratan keamanan organisasi Anda, Anda dapat menyesuaikan kebijakan mana yang diterapkan. Untuk informasi selengkapnya, lihat Kebijakan hak istimewa paling tidak khusus untukAWS CDKperan penyebaran.
- Patuhi praktik terbaik saat mengatur akses di IAM— Membangun yang terstrukturAWS Identity and Access Management(IAM) solusi akses yang memungkinkan pengguna Anda mengakses ADDFAkun AWS. Kerangka kerja ADDF dirancang untuk mematuhi prinsip hak istimewa paling sedikit. Pola akses IAM Anda juga harus mengikuti prinsip hak istimewa terkecil, harus sesuai dengan persyaratan organisasi Anda dan harus mematuhiPraktik terbaik keamanan di IAM(Dokumentasi IAM).

- Siapkan jaringan sesuai dengan praktik terbaik organisasi Anda— ADDF mencakup jaringan opsionalAWS CloudFormationstack yang menciptakan cloud pribadi virtual publik atau pribadi dasar (VPC). Bergantung pada konfigurasi organisasi Anda, VPC ini mungkin mengekspos sumber daya langsung ke internet. Kami menyarankan Anda mengikuti praktik terbaik jaringan organisasi Anda dan membuat modul jaringan yang diperkeras keamanan khusus.
- Menyebarkan tindakan pencegahan, deteksi, dan mitigasi keamanan diAkun AWSlevel— AWSmenawarkan berbagai layanan keamanan, seperti AmazonGuardDuty,AWS Security HubDetektif Amazon, danAWS Config. Aktifkan layanan tersebut di ADDF AndaAkun AWSdan mengintegrasikan proses pencegahan, deteksi, mitigasi, dan penanganan insiden keamanan organisasi Anda. Kami menyarankan Anda mengikutiPraktik Terbaik untuk Keamanan, Identitas, & Kepatuhan(AWSArchitecture Center) dan rekomendasi khusus layanan apa pun yang terkandung dalam dokumentasi untuk layanan tersebut. Untuk informasi lebih lanjut, lihatAWSDokumentasi Keamanan.

ADDF tidak membahas topik ini karena detail implementasi dan konfigurasi sangat bergantung pada persyaratan dan proses yang spesifik untuk organisasi Anda. Sebaliknya, itu adalah tanggung jawab inti organisasi Anda untuk membahas topik-topik ini. Umumnya, tim yang mengelolaAWSzona pendaratanmembantu Anda merencanakan dan mengimplementasikan lingkungan ADDF Anda.

Pengaturan awal

Mengatur ADDF sesuai denganPanduan Penerapan ADDF(GitHub). Titik awal untuk penerapan apa pun adalah`/manifest`folder diautonomous-driving-data-frameworkRepositori Git Hub. The`/manifest/example-dev`folder berisi contoh penyebaran untuk tujuan demo. Gunakan contoh ini sebagai titik awal untuk merancang penerapan Anda sendiri. Di direktori itu, ada file manifes penerapan ADDF yang disebutdeployment.yaml. Ini berisi semua informasi untukSeedFarmeruntuk mengelola, menyebarkan, atau menghapus ADDF dan sumber dayanya diAWS Cloud. Anda dapat membuat grup modul ADDF dalam file khusus. Thecore-modules.yamladalah contoh dari kelompok modul inti, dan mencakup semua modul inti yang disediakan oleh ADDF. Untuk meringkas,deployment.yamlfile berisi semua referensi ke grup dan modul yang akan digunakan ke akun target mereka dan menentukan urutan penerapan.

Untuk konfigurasi yang aman dan sesuai, terutama di lingkungan yang bukan untuk bukti konsep, kami sarankan Anda meninjau kode sumber setiap modul yang ingin Anda terapkan. Menurut praktik terbaik pengerasan keamanan, Anda harus menggunakan hanya modul yang diperlukan untuk kasus penggunaan yang Anda maksudkan.

> **Note**
>
> Modul ADDF di`modules/demo-only/`folder tidak diperkeras keamanan dan tidak boleh digunakan di lingkungan produksi atau di lingkungan apa pun dengan data sensitif atau terlindungi. Modul-modul ini disertakan untuk menampilkan kemampuan sistem, dan Anda dapat menggunakannya sebagai dasar untuk membuat modul yang disesuaikan dan diperkuat keamanan Anda sendiri.

Menyesuaikan kode kerangka kerja penerapan ADDF

Kerangka penerapan ADDF dan logika orkestrasi dan penerapannya dapat sepenuhnya disesuaikan untuk memenuhi persyaratan apa pun. Namun, kami menyarankan Anda untuk tidak menyesuaikan atau meminimalkan perubahan Anda karena alasan berikut:

- Pertahankan kompatibilitas hulu— Kompatibilitas hulu memudahkan untuk memperbarui ADDF untuk fitur terbaru dan pembaruan keamanan. Mengubah kerangka kerja merusak kompatibilitas mundur asli denganSeedFarmer,CodeSeeder, dan modul inti ADDF apa pun.
- Konsekuensi keamanan— Mengubah kerangka kerja penerapan ADDF dapat menjadi tugas kompleks yang dapat memiliki konsekuensi keamanan yang tidak diinginkan. Dalam skenario terburuk, perubahan kerangka kerja dapat menciptakan kerentanan keamanan.

Jika memungkinkan, buat dan sesuaikan kode modul Anda sendiri alih-alih memodifikasi kerangka kerja penerapan ADDF dan kode modul inti ADDF.

> **Note**
>
> Jika Anda merasa bahwa bagian tertentu dari kerangka penerapan ADDF memerlukan peningkatan atau pengerasan keamanan lebih lanjut, harap kontribusikan perubahan Anda ke repositori ADDF melalui permintaan tarik. Untuk informasi selengkapnya, lihat Tinjauan dan kontribusi keamanan sumber terbuka.

Menulis modul khusus di ADDF

Membuat modul ADDF baru atau memperluas modul yang ada adalah konsep inti ADDF. Saat membuat atau menyesuaikan modul, kami sarankan Anda mengikuti umumAWSpraktik terbaik

keamanan dan praktik terbaik organisasi Anda untuk pengkodean yang aman. Selain itu, kami menyarankan Anda melakukan tinjauan keamanan teknis internal atau eksternal awal dan berkala, berdasarkan persyaratan keamanan organisasi Anda, untuk lebih mengurangi risiko masalah keamanan.

Penerapan ADDF yang berulang

Terapkan ADDF dan modulnya seperti yang dijelaskan dalamPanduan Penerapan ADDF(GitHub). Untuk mendukung penerapan ADDF berulang yang menambah, memperbarui, atau menghapus sumber daya di akun target Anda,SeedFarmermenggunakan hash MD5, yang disimpan di Penyimpanan Parameter toolchain dan target acccount Anda, untuk membandingkan infrastruktur yang saat ini digunakan dengan infrastruktur yang ditentukan dalam file manifes di basis kode lokal Anda.

Pendekatan ini mengikutiGitOpsparadigma, di mana repositori sumber Anda (basis kode lokal tempat Anda beroperasiSeedFarmer) adalah sumber kebenaran, dan infrastruktur yang dinyatakan secara eksplisit di dalamnya adalah hasil yang diinginkan dari penyebaran Anda. Untuk informasi lebih lanjut tentangGitOps, lihatApa ituGitOps(GitLabsitus web).

Audit keamanan berulang

Sama seperti perangkat lunak lain di organisasi Anda, integrasikan ADDF dan kode modul ADDF khusus Anda ke dalam manajemen risiko keamanan, tinjauan keamanan, dan siklus audit keamanan Anda.

Pembaruan ADDF

ADDF menerima pembaruan rutin sebagai bagian dari upaya pengembangannya yang sedang berlangsung. Ini termasuk pembaruan fitur, dan peningkatan dan perbaikan terkait keamanan. Kami menyarankan Anda secara teratur memeriksa rilis kerangka kerja baru dan menerapkan pembaruan tepat waktu. Untuk informasi lebih lanjut, lihatLangkah-langkah untuk memperbarui ADDF(Dokumentasi ADDF).

Penonaktifan

Jika ADDF tidak lagi diperlukan, hapus ADDF dan semua sumber daya terkaitnya dariAkun AWS. Setiap infrastruktur yang tidak dijaga dan tidak terpakai menimbulkan biaya yang tidak perlu dan

menimbulkan risiko keamanan potensial. Untuk informasi lebih lanjut, lihatLangkah-langkah untuk menghancurkan ADDF(Dokumentasi ADDF).

Langkah selanjutnya

Panduan ini meninjau praktik dan pertimbangan terbaik keamanan dan operasi saat menerapkan Autonomous Driving Data Framework (ADDF) diAWS Cloudlingkungan. Panduan ini mengulas model tanggung jawab bersama antara pengguna ADDF, tim inti ADDF, danAWSsehingga Anda memahami peran dan tanggung jawab Anda untuk menyiapkan dan mengoperasikan ADDF dengan aman. Ini juga mencakup rekomendasi untuk mengoperasikan ADDF dengan aman melalui siklus hidupnya, termasuk rekomendasi khusus lingkungan.

Kami menyarankan Anda membiasakan diri dengan sumber daya diSumber dayabagian. Ketika Anda siap, Anda dapat mengatur ADDF sesuai dengan instruksi diPanduan Penerapan ADDF(GitHub).

Saat Anda menyiapkan dan mengoperasikan ADDF, jika menurut Anda kerangka kerja penerapan memerlukan peningkatan atau pengerasan keamanan lebih lanjut, harap kontribusikan perubahan Anda ke repositori ADDF melalui permintaan tarik. Untuk informasi selengkapnya, lihat Tinjauan dan kontribusi keamanan sumber terbuka.

Sumber daya

Dokumentasi AWS

- Kembangkan dan terapkan alur kerja yang disesuaikan menggunakan ADDF padaAWS(AWSposting blog)
- AWSdokumentasi layanan keamanan
- Praktik terbaik keamanan di IAM
- AWSmanajemen akun dan pemisahan
- Bootstrapping untukAWS CDK
- AWSmodel tanggung jawab bersama
- AWSKerangka Kerja yang Dirancang dengan Baik

Sumber daya sumber terbuka

- Repositori ADDF(GitHub)
- Panduan Penerapan ADDF(GitHub)
- CodeSeederrepositori(GitHub)
- SeedFarmerrepositori(GitHub)

Pemberitahuan

Pelanggan bertanggung jawab untuk membuat penilaian independen mereka sendiri atas informasi dalam dokumen ini. Dokumen ini: (a) hanya untuk tujuan informasi, (b) mewakili saat iniAWSpenawaran dan praktik produk, yang dapat berubah tanpa pemberitahuan, dan (c) tidak membuat komitmen atau jaminan apa pun dariAWSdan afiliasinya, pemasok atau pemberi lisensinya.AWSproduk atau layanan disediakan “sebagaimana adanya” tanpa jaminan, representasi, atau kondisi apa pun, baik tersurat maupun tersirat.

Tanggung jawab dan kewajibanAWSUntuk pelanggannya dikendalikan olehAWSperjanjian, dan dokumen ini bukan bagian dari, juga tidak mengubah, perjanjian apa pun antaraAWSdan pelanggannya.

Riwayat dokumen

Tabel berikut menjelaskan perubahan signifikan pada panduan ini. Jika Anda ingin diberi tahu tentang pembaruan di masa mendatang, Anda dapat berlanggananUmpan RSS.

Perubahan	Deskripsi	Tanggal
Publikasi awal	—	November 15, 2022

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.