Laporan Resmi AWS

Integrasi Berkelanjutan dan Pengiriman Berkelanjutan untuk Jaringan 5G di AWS

Integrasi Berkelanjutan dan Pengiriman Berkelanjutan untuk Jaringan 5G di AWS: Laporan Resmi AWS

Table of Contents

Integrasi Berkelanjutan dan Pengiriman Berkelanjutan untuk Jaringan 5G di AWS

Tanggal publikasi: 8 Maret 2021 (Revisi dokumen)

Abstrak

Laporan resmi ini memperkenalkan integrasi berkelanjutan dan pengiriman berkelanjutan (CI/CD) untuk jaringan 5G, serta bagaimana alat dan layanan Amazon Web Services (AWS) dapat digunakan untuk mengotomatiskan deployment dan meng-upgrade fungsi jaringan 5G sepenuhnya. Laporan resmi ini memberikan penjelasan mendetail tentang berbagai tahapan CI/CD untuk fungsi jaringan 5G, termasuk penyiapan jaringan, deployment infrastruktur, deployment fungsi jaringan cloud native, dan pembaruan fungsi jaringan yang berkelanjutan. Laporan resmi ini juga memberikan detail tentang integrasi dengan alat sumber terbuka dan pihak ketiga untuk pengujian, pengamatan, dan orkestrasi.

Laporan resmi ini ditujukan untuk Penyedia Layanan Komunikasi (CSP) serta Vendor Perangkat Lunak Independen (ISV).

Pengantar

Secara historis, pengembangan, pengujian integrasi laboratorium dan lapangan, serta deployment produksi node jaringan baru atau fitur baru dalam jaringan seluler membutuhkan waktu berminggu-minggu atau bahkan berbulan-bulan untuk memastikan stabilitas layanan telekomunikasi yang sangat penting untuk misi dan bisnis. Siklus deployment yang panjang disebabkan oleh arsitektur monolitik node jaringan tradisional, lingkungan multi-vendor, dan banyak antarmuka titik ke titik di antara entitas jaringan di jaringan seluler 2G, 3G, dan 4G.

Seperti yang dijelaskan dalam laporan resmi Evolusi Jaringan 5G dengan AWS, jaringan seluler 5G, sebagaimana distandardisasi oleh 3GPP, sekarang mendukung arsitektur cloud native yang didukung oleh virtualisasi dan kontainerisasi. Lebih khusus lagi, jaringan 5G memperkenalkan dan mendukung paradigma baru arsitektur layanan mikro, stateless, dan berbasis layanan.

Arsitektur 5G ini berarti bahwa fungsi jaringan yang berbeda-beda dapat berfungsi sebagai layanan independen yang di-coupling secara fleksibel yang berkomunikasi satu sama lain melalui antarmuka dan API yang terdefinisi dengan baik. Yang paling penting, setiap fungsi jaringan dapat diperbarui secara independen. Pergeseran arsitektur dalam 5G ini memungkinkan CSP mencapai lebih banyak ketangkasan dan efisiensi operasional dengan mempermudah untuk meluncurkan pembaruan untuk fungsi jaringan secara lebih sering, sambil mempertahankan persyaratan dan standar pengujian dan keamanan melalui otomatisasi.

Integrasi dan deployment fitur baru untuk CSP umumnya dimulai ketika vendor fungsi jaringan merilis paket perangkat lunak fungsi jaringan baru, seperti image Docker dalam fungsi jaringan berbasis kontainer, atau file konfigurasi baru, seperti bagan Helm dalam kasus aplikasi Kubernetes . (Bagan Helm adalah kumpulan file yang menggambarkan set sumber daya Kubernetes terkait).

Gagasan dalam menggunakan paradigma CI/CD untuk deployment fungsi jaringan 5G makin populer, tetapi realisasi praktis dalam gagasan ini telah menjadi tantangan dalam industri telekomunikasi.

AWS telah memelopori pengembangan alat CI/CD baru untuk pengiriman perangkat lunak guna membantu spektrum industri yang luas mengembangkan dan meluncurkan perubahan perangkat lunak dengan cepat, sambil mempertahankan stabilitas dan keamanan sistem. Alat tersebut mencakup serangkaian layanan Pengembangan dan Operasi Perangkat Lunak (DevOps) seperti AWS CodeStar, CodeCommit, CodePipeline, CodeBuild, dan CodeDeploy.

AWS juga menyebarluaskan gagasan Infrastruktur sebagai Kode (IaC) menggunakan AWS Cloud Development Kit (AWS CDK), AWS CloudFormation, dan alat pihak ketiga berbasis API seperti

Terraform. Dengan menggunakan alat ini, AWS dapat menyimpan proses deployment fungsi jaringan dalam AWS sebagai kode sumber, dan memelihara kode sumber IaC ini di alur CI/CD untuk mewujudkan pengiriman berkelanjutan.

Laporan resmi ini menjelaskan proses mendetail untuk memanfaatkan alat IaC dan CI/CD AWS untuk deployment dan pembaruan fungsi jaringan 5G. Selain itu, laporan resmi ini membahas integrasi dengan alat pihak ketiga untuk pengujian, kemampuan pengamatan, dan orkestrasi.

Alat CI/CD AWS tidak terbatas pada fungsi jaringan 5G. Alat ini juga digunakan untuk mengotomatisasi deployment jaringan 4G, yang memungkinkan CSP men-deploy dan memperbarui fungsi jaringan 4G dengan cepat dan efisien. Sebagian besar fungsi jaringan 4G berbasis Virtual Network Function (VNF). Rangkaian alat CI/CD AWS seperti AWS CloudFormation dapat digunakan untuk mengotomatisasi deployment VNF 4G, sehingga menyediakan skala dan efisiensi waktu untuk deployment jaringan 4G.

Integrasi Berkelanjutan (CI) dan Pengiriman Berkelanjutan (DI)

Integrasi Berkelanjutan

Integrasi Berkelanjutan (CI) adalah proses perangkat lunak yang memungkinkan developer secara rutin mendorong kode mereka ke repositori pusat seperti AWS CodeCommit atau GitHub. Setiap kode yang didorong akan memicu pembangunan otomatis, yang diikuti dengan menjalankan pengujian. Tujuan utama CI adalah menemukan masalah kode pada tahap awal, meningkatkan kualitas kode, dan mengurangi waktu yang diperlukan untuk memvalidasi dan merilis pembaruan perangkat lunak baru.

Pengiriman dan Deployment Berkelanjutan

Pengiriman Berkelanjutan (CD) adalah proses perangkat lunak yang memungkinkan artefak di-deploy ke lingkungan pengujian, lingkungan penahapan, dan lingkungan produksi. Pengiriman berkelanjutan dapat sepenuhnya otomatis, atau memiliki tahap persetujuan pada titik-titik kritis. Hal ini memastikan bahwa semua persetujuan yang diperlukan sebelum deployment, seperti persetujuan manajemen rilis, diterapkan. Jika pengiriman berkelanjutan diimplementasikan dengan benar, developer akan selalu memiliki artefak pembangunan yang siap di-deploy yang telah melewati proses pengujian terstandardisasi.

Dengan Deployment Berkelanjutan, revisi akan otomatis di-deploy ke lingkungan produksi tanpa persetujuan eksplisit dari developer, sehingga menjadikan seluruh proses rilis perangkat lunak terotomatisasi. Hal ini memungkinkan adanya putaran umpan balik pelanggan yang berkelanjutan di awal siklus hidup produk.

Dengan Deployment Berkelanjutan, setiap perubahan yang dilakukan dan lulus pengujian otomatis akan dirilis ke produksi secara otomatis. Pengiriman Berkelanjutan tidak dimaksudkan untuk merilis setiap perubahan yang dilakukan dan lulus pengujian otomatis ke produksi dengan segera, tetapi untuk memastikan bahwa setiap perubahan siap dikirim ke produksi.

Infrastruktur sebagai Kode (IaC)

Sebagaimana diuraikan dalam laporan resmi Evolusi Jaringan 5G denganAWS, IaC adalah penggerak utama untuk mengotomatisasi proses penyediaan dan manajemen siklus hidup untuk

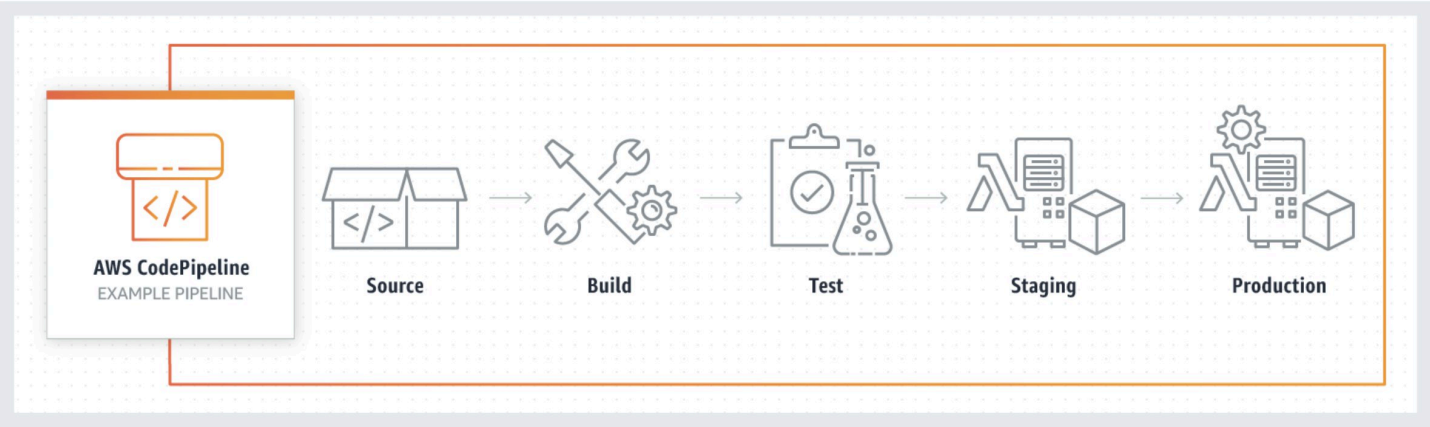
aplikasi dan lingkungannya. Alih-alih mengandalkan langkah-langkah yang dilakukan secara manual, administrator jaringan/IT dan developer dapat menginstansiasikan infrastruktur menggunakan file konfigurasi. IaC memperlakukan file konfigurasi ini sebagai kode perangkat lunak. File-file ini dapat digunakan untuk menghasilkan satu set artefak: layanan komputasi, penyimpanan, jaringan, dan aplikasi yang membentuk lingkungan operasi. IaC menghilangkan configuration drift melalui otomatisasi, sehingga meningkatkan kecepatan dan ketangkasan deployment infrastruktur.

Dalam kasus implementasi Network Function Virtualization (NFV) di AWS, kerangka kerja IaC ini menghadirkan nilai dari perspektif orkestrasi. Dari pembuatan Virtual Private Cloud (VPC) hingga deployment fungsi jaringan, setiap langkah dapat diprogram, dikelola sebagai kode sumber, dan dipelihara dengan kontrol versi di AWS CodeCommit.

Kerangka kerja IaC untuk fungsi jaringan ini menghasilkan pembuatan dan deployment fungsi infrastruktur dan jaringan yang dapat diulang dan dapat diandalkan, yang dapat diperluas ke otomatisasi ujung ke ujung (E2E) untuk manajemen slice jaringan dan manajemen siklus hidup layanan. AWS menyediakan set alat komprehensif untuk membuat, memelihara, dan men-deploy infrastruktur dengan cara terprogram, deskriptif, dan deklaratif, menggunakan layanan seperti AWS CloudFormationAWS CDK, AWS CDK for Kubernetes, dan eksposur API untuk semua Layanan AWS.

CI/CD di AWS

CI/CD dapat digambarkan sebagai alur yang memungkinkan kode baru dikirimkan di salah satu ujung, diuji melalui serangkaian tahapan (sumber, pembangunan, pengujian, penahapan, dan produksi), lalu dipublikasikan sebagai kode siap produksi.



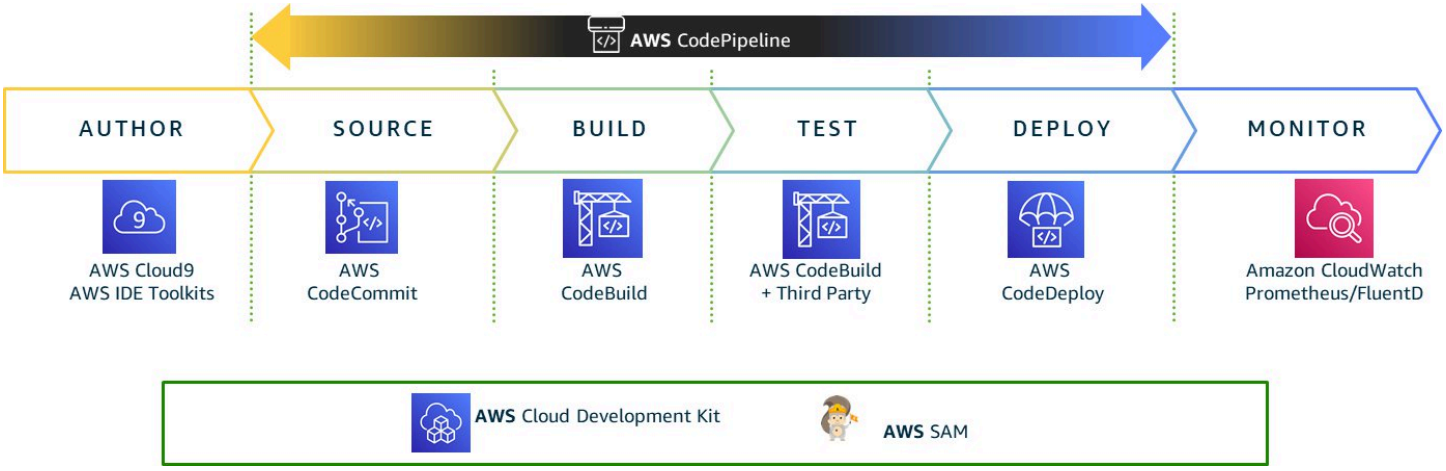
Gambaran umum alur CICD

Setiap tahap alur CI/CD terstruktur sebagai unit logis dalam proses pengiriman. Setiap tahap bertindak sebagai gerbang yang memeriksa aspek tertentu dari kode. Saat kode diproses dalam sebuah alur, dapat diasumsikan bahwa kualitas kode tersebut akan lebih baik di tahap berikutnya karena makin banyak aspeknya yang terus diverifikasi. Masalah-masalah yang ditemukan di tahap awal akan mencegah pemrosesan kode dalam alur. Hasil pengujian segera dikirim ke tim, dan semua pembangunan serta peluncuran lebih lanjut akan dihentikan jika perangkat lunak gagal melewati tahapan ini.

AWS menghadirkan serangkaian alat developer CI/CD lengkap untuk mempercepat siklus pengembangan dan rilis perangkat lunak. AWS CodePipeline mengotomatiskan tahap pembangunan, pengujian, dan deployment dalam proses rilis setiap kali ada perubahan kode, berdasarkan model rilis yang ditetapkan. Hal ini memungkinkan pengiriman fitur dan pembaruan yang cepat dan dapat diandalkan.

Alur kode dapat berintegrasi dengan layanan lain. Ini bisa berupa Layanan AWS, seperti Amazon Simple Storage Service (Amazon S3), atau produk pihak ketiga, seperti GitHub. AWS CodePipeline dapat menangani berbagai kasus penggunaan pengembangan dan operasi termasuk:

- Kompilasi, pembangunan, dan pengujian kode dengan AWS CodeBuild
- Pengiriman berkelanjutan terhadap aplikasi berbasis kontainer ke cloud
- Validasi pra-deployment terhadap artefak (seperti deskriptor dan image kontainer) yang diperlukan untuk layanan jaringan atau fungsi jaringan cloud native tertentu
- Pengujian fungsi, integrasi, dan performa untuk fungsi jaringan terkontainerisasi/fungsi jaringan virtual (CNF/VNF), termasuk pengujian acuan dasar dan regresi
- Pengujian keandalan dan pemulihan bencana (DR).



Komponen alur CI/CD AWS

AWS dapat menyiapkan alur CI/CD menggunakan Alat Developer AWS berikut:

- AWS CodeCommit
- AWS CodeBuild
- AWS CodePipeline
- AWS CodeDeploy
- Amazon Elastic Container Registry
- AWS CodeStar

Pembuatan alur CI/CD dapat diotomatisasi menggunakan AWS CDK dan AWS CloudFormation. Dalam domain NFV, otomatisasi native AWS ini dapat diintegrasikan ke dalam kerangka kerja Management and Orchestration (MANO) dan kerangka kerja orkestrasi layanan CSP.

Proses CI/CD mencakup langkah-langkah berikut:

- Penyiapan jaringan – AWS CDK dan AWS CloudFormation menginisiasi pembuatan prasyarat jaringan:
    - Tumpukan jaringan (VPC, subnet, gateway terjemahan alamat jaringan (NAT), tabel rute, dan gateway internet)
- Deployment infrastruktur – AWS CDK dan AWS CloudFormation menginisiasi pembuatan tumpukan sumber daya berikut:
    - Tumpukan komputasi (pembuatan klaster Amazon Elastic Kubernetes Service (Amazon EKS), node Pekerja EKS, AWS Lambda)
    - Tumpukan penyimpanan (bucket Amazon S3, volume Amazon Elastic Block Store (Amazon EBS), dan Amazon Elastic File System (Amazon EFS)
    - Tumpukan pemantauan (CloudWatch , Amazon OpenSearch Service (OpenSearch Service)
    - Tumpukan keamanan (AWS Identity and Access Management (AWS IAM), grup keamanan Amazon Elastic Compute Cloud (Amazon EC2), daftar kontrol akses jaringan (NACL) VPC

- Deployment Cloud Network Function (CNF) — Pada tahap ini, CNF di-deploy ke klaster EKS menggunakan alat bagan Kubectl dan Helm. Tahap ini juga men-deploy aplikasi atau alat tertentu yang dibutuhkan oleh CNF untuk bekerja secara efisien (seperti Prometheus atau Fluentd). CNF dapat di-deploy melalui fungsi Lambda atau dengan AWS CodeBuild.
- Pembaruan dan deployment berkelanjutan – Ini adalah urutan langkah yang dilakukan secara iteratif untuk men-deploy perubahan yang merupakan bagian dari perubahan kontainer/konfigurasi yang menghasilkan upgrade. Mirip dengan kasus deployment CNF, pembaruan dan deployment berkelanjutan dapat diotomatisasi menggunakan Layanan AWS, dengan pemicu dari AWS CodeCommit, Amazon Elastic Container Registry (Amazon ECR), atau sistem sumber pihak ketiga seperti GitLab Webhook.

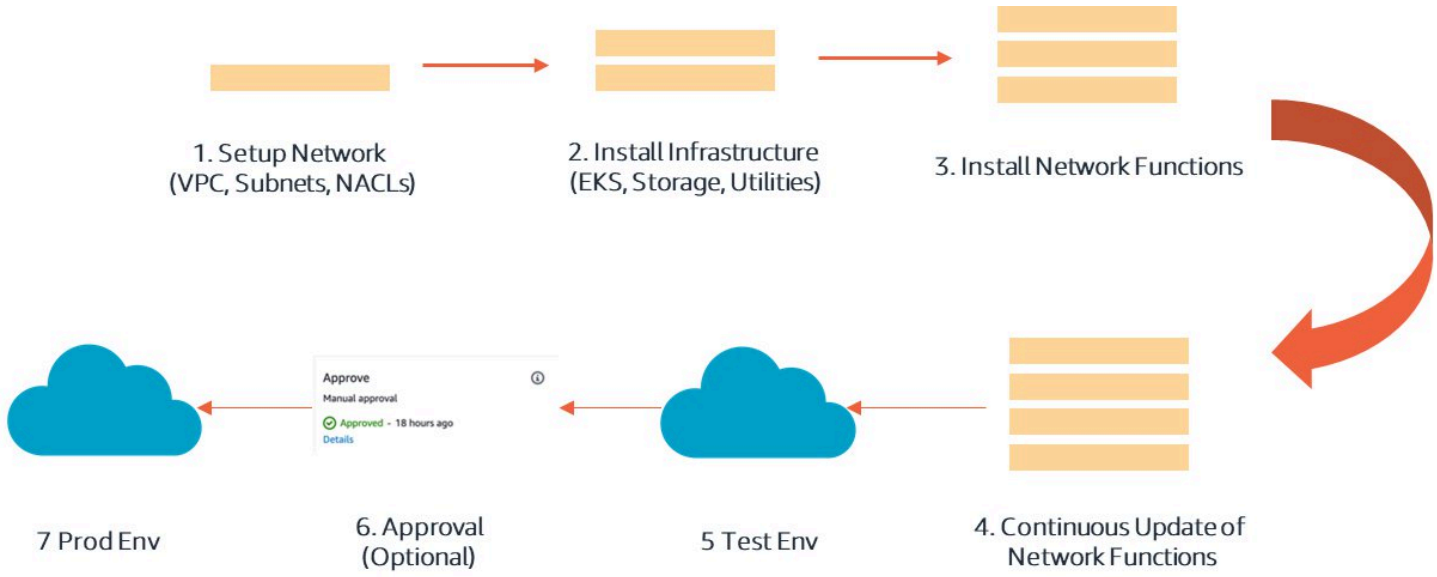


Diagram rangkaian alur CI/CD AWS

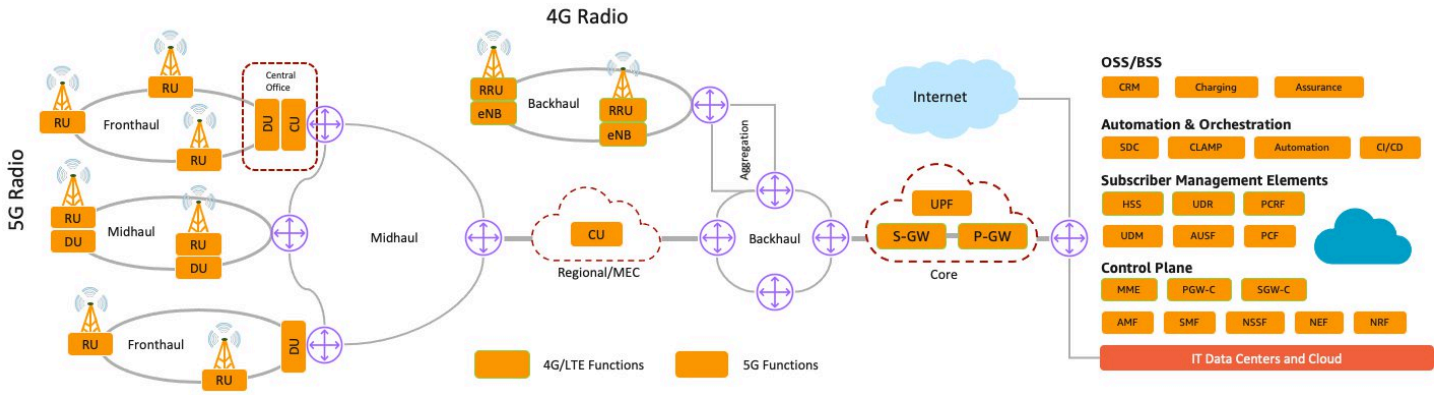
Alur CI/CD dibuat menggunakan AWS CodePipeline, dan memanfaatkan layanan pengiriman berkelanjutan yang memodelkan, memvisualisasikan, dan mengotomatisasi langkah-langkah yang diperlukan untuk merilis perangkat lunak. Dengan mendefinisikan tahapan dalam alur, Anda dapat mengambil kode dari repositori kode sumber, membangun kode sumber itu menjadi artefak yang dapat dirilis, menguji artefak, dan men-deploy artefak ke produksi. Hanya kode yang berhasil melewati semua tahap ini akan di-deploy. Anda memiliki opsi untuk menambahkan persyaratan lain ke alur Anda, seperti persetujuan manual, untuk membantu memastikan bahwa hanya perubahan yang disetujui yang di-deploy ke produksi.

Jaringan 5G di AWS

Model standar infrastruktur jaringan 5G terdiri dari situs radio 4G/5G, jaringan fronthaul/midhaul/backhaul, situs jaringan inti, dan pusat data telekomunikasi/IT. CSP dapat menggunakan Layanan AWS untuk menciptakan infrastruktur jaringan 5G yang dapat diskalakan dan fleksibel sambil mengurangi biaya investasi di muka. AWS dapat digunakan untuk mengimplementasikan Pusat Operasi Jaringan (NOC) virtual di Wilayah yang meng-host Sistem Dukungan Operasi/Sistem Dukungan Bisnis (OSS/BSS) dan sebagian besar fungsi jaringan inti bidang kontrol.

AWS juga dapat dimanfaatkan untuk mengimplementasikan kantor pusat (CO) lokal atau pusat data terdistribusi dengan armada instans AWS Outposts yang meng-host sebagian besar fungsi bidang pengguna seperti UPF (fungsi bidang pengguna), unit pusat (CU) RAN, dan Komputasi Edge Multi-Akses (MEC). Penjelasan lebih mendetail tentang arsitektur referensi dan manfaat penerapan jaringan 5G di AWS dijelaskan dalam laporan resmi Evolusi Jaringan 5G di AWS.

Ketika saatnya menerapkan jaringan 5G di AWS, alat CI/CD AWS, yang diperkenalkan di bagian berikut dalam laporan resmi ini, dapat memfasilitasi otomatisasi penuh deployment, upgrade, dan manajemen siklus hidup fungsi jaringan 5G.

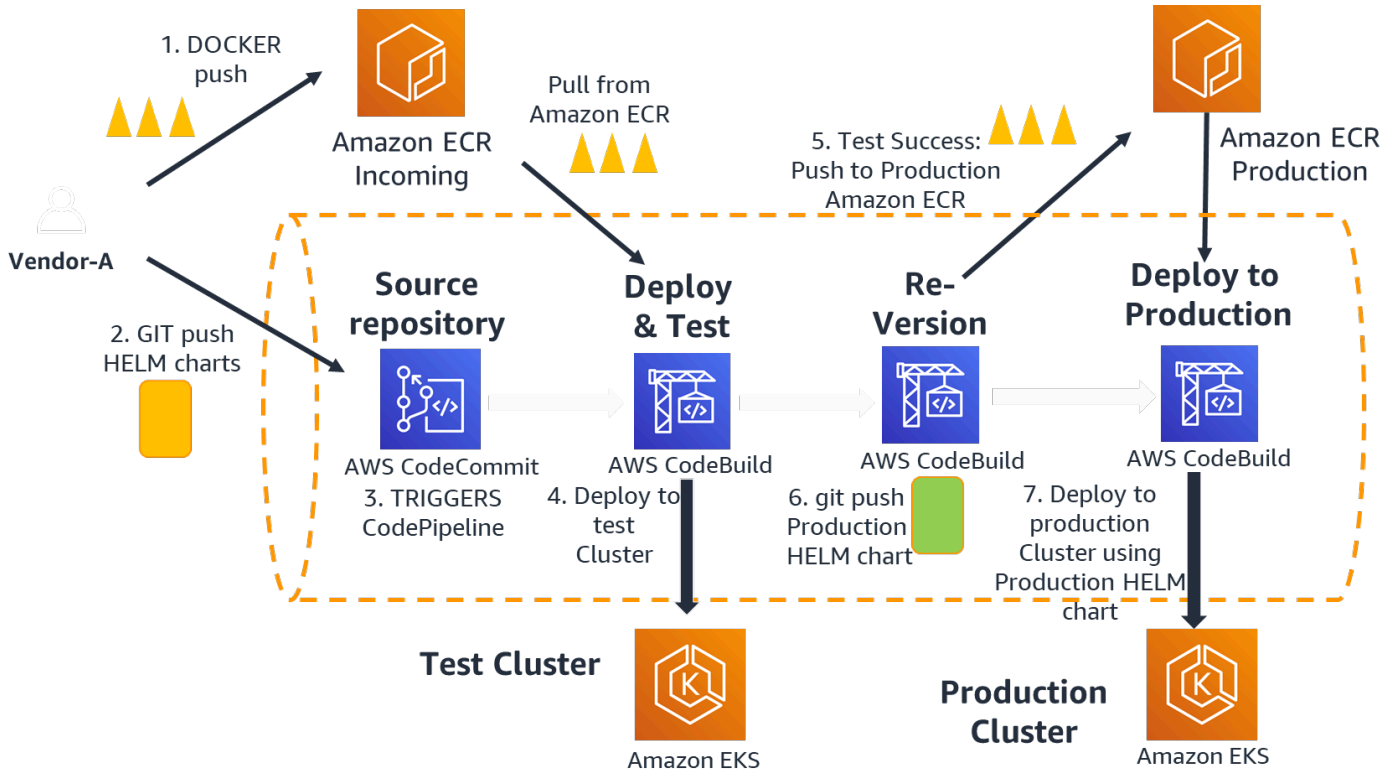


Arsitektur E2E jaringan 5G

CI/CD dalam jaringan 5G

Konstruksi desain infrastruktur disimpan dalam bentuk kode yang menggunakan bahasa deklaratif. Hal ini memungkinkan CSP memiliki proses pembuatan infrastruktur yang berulang dengan perilaku yang sama seperti yang diperlukan. Kode ini dipelihara dalam repositori kode, dan sebuah alur disiapkan untuk menyiapkan pembaruan orkestrasi ke tumpukan yang di-deploy (misalnya, AWS CDK dan AWS CloudFormation). AWS dapat membantu membangun Infrastruktur sebagai

Kode (IaC) untuk melakukan onboarding yang tangkas terhadap fungsi Vendor Perangkat Lunak Independen (ISV).



Rangkaian alur kode

Perubahan konfigurasi fungsi jaringan cloud native melalui bagan Helm dianggap sebagai pemicu untuk eksekusi alur CI/CD otomatis untuk fungsi jaringan.

AWS CodeCommit dapat digunakan untuk mengelola file konfigurasi, dan Amazon ECR dapat digunakan untuk memelihara image kontainer.

Seperti yang ditunjukkan pada gambar Rangkaian alur kode, ketika ISV mendorong perubahan kode baru ke dalam repositori kode (bagan Helm, file konfigurasi, atau file properti), alur kode akan dipicu. Alur kode akan menarik image dari ECR dan menggunakan bagan Helm untuk men-deploy aplikasi. Pengujian aplikasi baru dapat diintegrasikan dengan kerangka kerja otomatisasi pengujian pihak ketiga. Berdasarkan hasilnya, CSP dapat menyetujui deployment produksi.

Tahap sumber CodePipeline akan mencari perubahan dalam file konfigurasi. Penyedia yang valid untuk tahap sumber adalah CodeCommit, Amazon S3, GitHub, atau AWS CloudFormation. Sistem sumber alternatif dapat diintegrasikan dengan menggunakan fungsi Lambda untuk mengimplementasikan Webhook, yang memungkinkan integrasi berbasis peristiwa antara Gitlab dan AWS CodePipeline. Buka tautan berikut untuk panduan implementasi yang terperinci.

- Webhook dengan GitLab
- Integrasi registri kontainer

Desain alur CI/CD harus memperhitungkan langkah-langkah deployment penting seperti deployment awal, pengujian, dan promosi ke produksi setelah hasil pengujian selaras sudah sesuai dengan ekspektasi dan diverifikasi berdasarkan acuan dasar. Setiap tahap proses alur menyediakan artefak data, yang memungkinkan perbandingan dan keputusan yang didorong data.

Langkah alur CI/CD aplikasi

Setiap tahap dapat dianggap sebagai tugas terpisah, sehingga memungkinkan pelaksanaan alur kerja validasi dan deployment yang memadai untuk mendukung layanan jaringan dan fungsi jaringan cloud native. Tugas dapat dijalankan menggunakan alat pihak ketiga tambahan seperti generator dan simulator lalu lintas, sehingga memungkinkan validasi layanan jaringan ujung ke ujung.

AWS menyediakan layanan AWS Step Function (mesin status cloud native) canggih yang terintegrasi secara native dengan Layanan AWS lainnya, dan juga memiliki kemampuan untuk berintegrasi dengan sistem eksternal seperti Jira atau kerangka kerja otomatisasi pengujian.

Langkah mendetail CI/CD

CI/CD dapat digambarkan sebagai alur yang memungkinkan kode baru dikirimkan di salah satu ujung, diuji melalui serangkaian tahapan (sumber, pembangunan, pengujian, penahapan, dan produksi), lalu dipublikasikan sebagai kode siap produksi.

Berikut adalah langkah-langkah deployment dan pengujian. Deployment dan konfigurasi terutama dibagi ke dalam empat bagian utama:

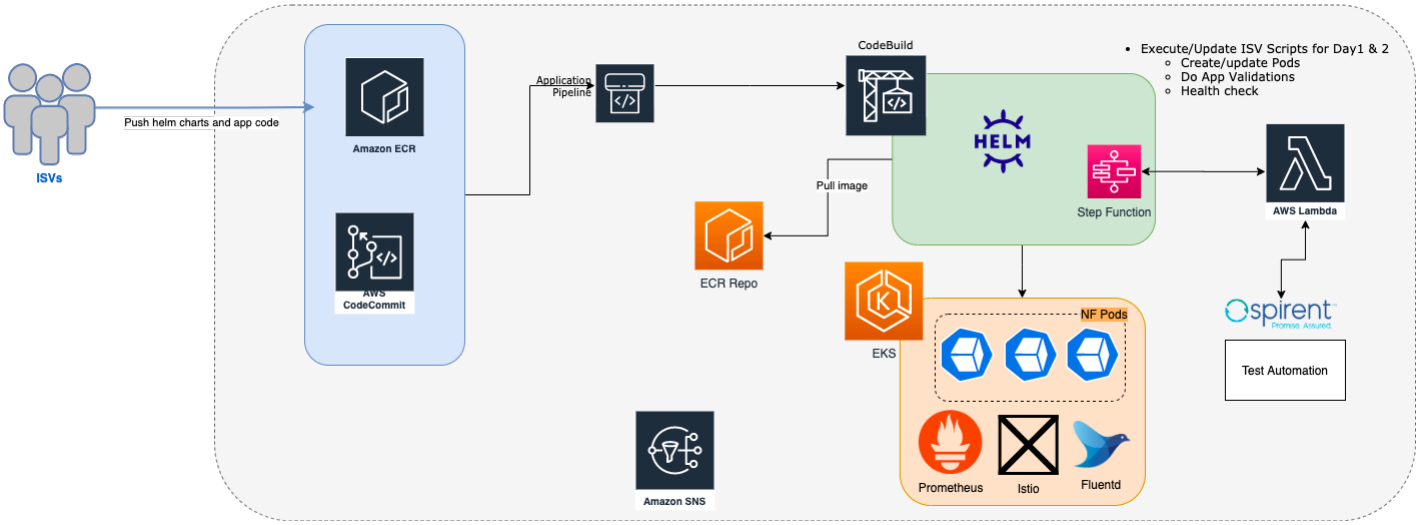
- Penyiapan jaringan
- Deployment infrastruktur
- Deployment fungsi jaringan cloud native
- Pengiriman berkelanjutan CNF

Penyiapan jaringan

Fokus pada penyiapan prasyarat infrastruktur. Ini termasuk pembuatan VPC, jaringan, subnet, NACL, dan sebagainya. Rancang rencana jaringan IP dengan mempertimbangkan ISV dan rencana implementasi pelanggan (seperti multi-penghunian dan alokasi statis vs. dinamis). Rencana ini dapat dikodekan dalam AWS CDK atau AWS CloudFormation. Jalankan kode ini untuk men-deploy prasyarat jaringan infrastruktur cloud.

Deployment infrastruktur

Deployment infrastruktur akan menyediakan komponen infrastruktur apa pun. Ini termasuk melakukan spawning klaster EKS dan infrastruktur pendukung, seperti EFS, node pekerja EKS, ELB, dan mengonfigurasi klaster sesuai dengan persyaratan fungsi jaringan cloud native. Berdasarkan persyaratan CNF, AWS juga men-deploy antarmuka jaringan tambahan untuk node, termasuk antarmuka Multus. Sebagian besar langkah deployment dan konfigurasi adalah upaya satu kali untuk sebuah aplikasi, dan diperbarui hanya jika diperlukan sebagai pembaruan untuk aplikasi.



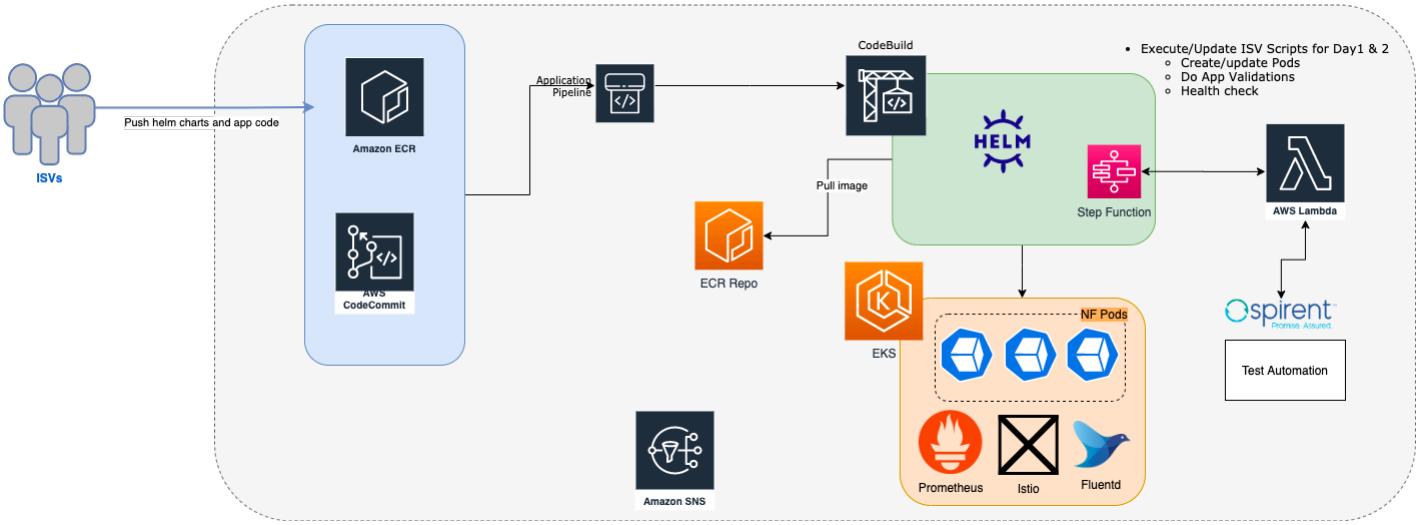
Deployment infrastruktur dengan CDK

Deployment fungsi jaringan cloud native

Deployment CNF berkaitan dengan deployment aplikasi. Sebagai bagian dari deployment CNF, bagan Helm untuk aplikasi diimplementasikan melalui alur kode CI/CD. Callback untuk menjalankan skrip khusus aplikasi individual yang terutama memerlukan pra- dan pasca-pemeriksaan akan diterapkan. Bagan Helm diimplementasikan secara berurutan sesuai kebutuhan aplikasi, dan

memeriksa status POD Kubernetes sebelum beralih ke langkah berikutnya dalam deployment. Sering kali, ISV menyediakan skrip wrapper untuk menjalankan bagan Helm dan sanity check. Skrip ISV ini dipanggil dari dalam AWS CodePipeline. Sebagai bagian dari fase ini, agen pencatatan log dan pemantauan seperti Prometheus dan Fluentd akan di-deploy di samping Amazon CloudWatch, yang mencatat dan memantau infrastruktur cloud aplikasi.

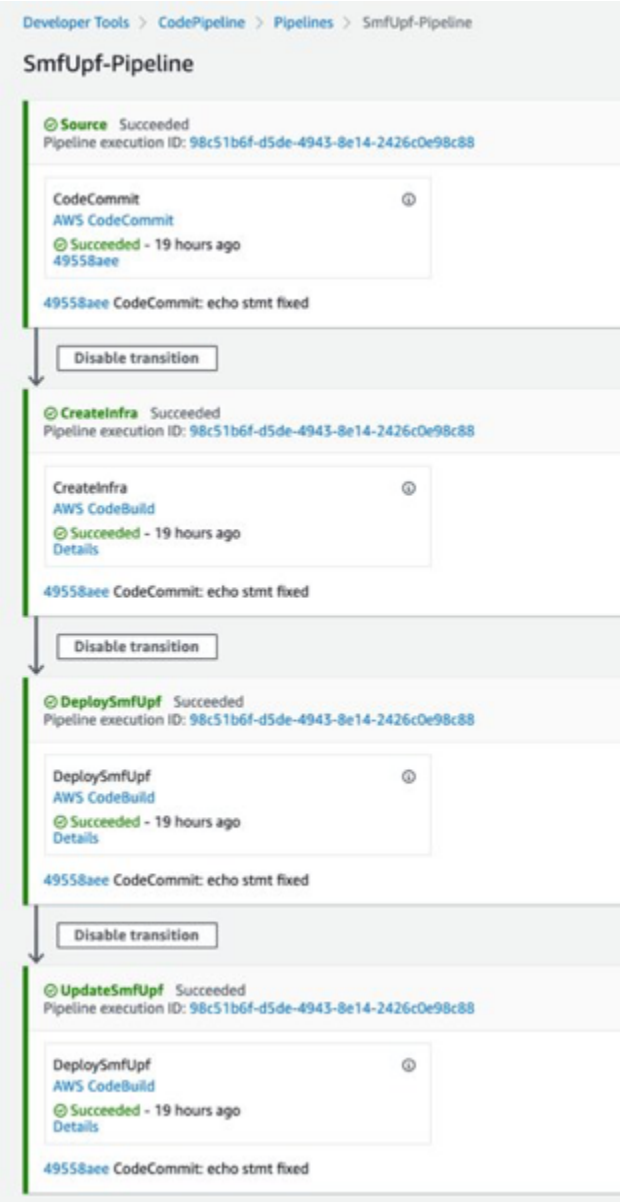
Alur kode terintegrasi dengan kerangka kerja otomatisasi pengujian pihak ketiga. Alur kode dapat langsung memanggil API kerangka kerja otomatisasi pengujian untuk menjalankan pengujian pada aplikasi yang di-deploy, mengkueri hasil pengujian, dan menganalisis hasilnya. Hal ini menyederhanakan deployment dan pengujian aplikasi.



Deployment dan pembaruan aplikasi

Berikut ini adalah contoh dari deployment CNF fungsi bidang pengguna/fungsi manajemen sesi (UPF/SMF) melalui AWS CodePipeline.

- Otomatiskan proses CI/CD lengkap menggunakan CodeCommit, CodeBuild, dan CodePipeline.
- Tugas pembuatan infrastruktur dan instalasi aplikasi terintegrasi sebagai bagian dari alur.
- Agen FluentD dan Prometheus diinstal dan dibuat di dasbor Amazon CloudWatch.



Contoh deployment CNF UPF/SMF

Pengiriman berkelanjutan CNF

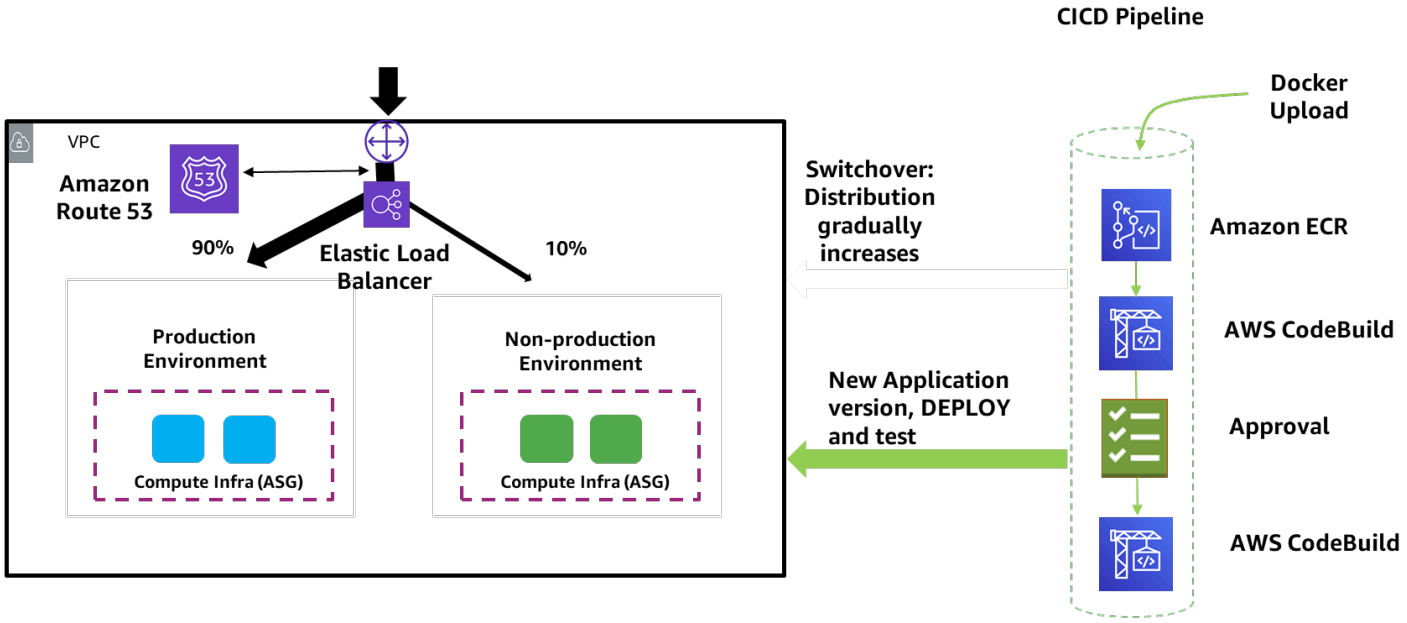
Langkah ini terdiri dari urutan langkah-langkah yang dilakukan berulang kali untuk men-deploy perubahan yang merupakan bagian dari perubahan kontainer/konfigurasi yang menghasilkan upgrade. Pengiriman berkelanjutan CNF diotomatisasi melalui alur dan bersifat spesifik untuk aplikasi individual. AWS menggunakan bagan Helm standar untuk memperbarui CNF tertentu. Alur kode memiliki pra- dan pasca-pemeriksaan untuk status pembaruan aplikasi. Alur CI/CD yang diperbarui juga terintegrasi dengan kerangka kerja otomatisasi pengujian untuk menjalankan pengujian otomatis. Penyederhanaan ini memungkinkan deployment fungsi jaringan yang jelas.

Pengiriman dan deployment berkelanjutan CNF dapat secara luas diklasifikasikan ke dalam kategori berikut:

- Upgrade aplikasi — Sebagian besar upgrade aplikasi adalah perubahan dalam POD aplikasi Kuberbetes. Pembaruan ini dapat diterapkan secara otomatis melalui alur kode. Sebagian besar CNF mendukung upgrade di tempat dengan menyediakan sejumlah instans POD aplikasi. Dengan adanya sejumlah instans, pendekatan upgrade bergulir dapat dimungkinkan. Tidak semua perubahan POD aplikasi mendukung upgrade Helm. Alur akan memperhitungkan variasi ini dan menggunakan instalasi/penghapusan Helm sesuai kebutuhan.
- Upgrade besar — Upgrade besar terutama merupakan perubahan skema basis data. Perubahan ini tidak dapat diterapkan tanpa menyebabkan waktu henti. Pendekatan standar terhadap perubahan ini adalah menghapus aplikasi dan membuat kembali pod yang relevan. Selama prosesnya, aplikasi mungkin tidak tersedia. Alat berikut digunakan untuk upgrade:

    - AWS CloudFormation memungkinkan pelanggan mendeskripsikan dan menyediakan semua sumber daya infrastruktur dalam templat JSON atau YAML. AWS CloudFormation menyediakan mekanisme ekstensi yang efektif melalui sumber daya kustom yang didukung Lambda. Pelanggan dapat memperluas AWS CloudFormation di luar sumber daya AWS, dan menyediakan sumber daya yang diperlukan di lingkungan lain; misalnya, sumber daya on-premise di lingkungan hibrid. AWS CDK memberi developer kemampuan untuk membangun kode menggunakan bahasa pemrograman tingkat lebih tinggi yang sudah dikenal seperti Python, TypeScript, JavaScript, Java, dan C#, lalu mengompilasi kode ini ke dalam format JSON AWS CloudFormation tingkat lebih rendah, yang kemudian dapat di-deploy.
    - Deployment Biru/Hijau — AWS mendukung dan merekomendasikan deployment berbasis biru/hijau dan canary dalam pengujian serta di lingkungan produksi. Deployment biru/hijau memungkinkan pelanggan menguji versi aplikasi baru di lingkungan yang dibatasi. Deployment ini menyediakan metode yang mudah dan lancar untuk mengalihkan lalu lintas produksi. Deployment berbasis canary memperluas konsep ini dengan memungkinkan lingkungan hijau non-produksi diuji dengan sebagian kecil lalu lintas produksi untuk mengungkap masalah apa pun yang disebabkan oleh lalu lintas produksi. Versi aplikasi baru diuji dengan lalu lintas pengujian simulasi internal serta sejumlah kecil lalu lintas produksi, yang memberikan keyakinan bagi pengguna sebelum mereka mengalihkan lalu lintas produksi. Lalu lintas produksi secara bertahap meningkat sampai peralihan selesai. Implementasi ini memerlukan grup target DNS berbobot dan ELB berbobot.

- Otomatisasi dapat dicapai dengan mengonfigurasi AWS CodePipeline dengan tahap deployment berbasis biru/hijau dan canary. Tahap persetujuan awalnya dapat didorong secara manual selama penyediaan, tetapi kemudian harus sepenuhnya otomatis. Di lingkungan pengujian, praktik yang baik adalah selalu menguji dengan tindakan rollback untuk memvalidasi kompatibilitas maju dan mundur, sebelum men-deploy ke dalam produksi. Deployment biru/hijau pada klaster dengan mesh layanan bergantung pada dukungan yang diberikan oleh aplikasi akhir dan gateway perutean untuk mesh layanan untuk mewujudkan transisi yang lancar.
- AWS Systems Manager menyediakan antarmuka pengguna terpadu sehingga Anda dapat melihat data operasional dari sejumlah Layanan AWS yang digunakan oleh fungsi jaringan yang di-deploy melalui CI/CD. Systems Manager memungkinkan Anda mengotomatisasi tugas operasional untuk seluruh sumber daya AWS Anda.



Deployment canary

Keamanan

Keamanan adalah elemen penting. Berikut ini adalah daftar langkah-langkah keamanan yang dipertimbangkan proses CI/CD AWS saat men-deploy aplikasi.

- Sumber — Repositori ECR yang dialokasikan ke vendor akan dikonfigurasi dengan mengaktifkan flag “Scan on Push”, sehingga setiap unggahan image Docker akan segera menjalani pemindaian

keamanan. Setiap kelemahan dan eksposur umum (CVE) yang diketahui akan di-flag dengan notifikasi. Terlepas dari ECR, ketika vendor memasukkan bagan ke dalam repositori AWS CodeCommit, mereka akan diminta untuk mengenkripsi kata sandi yang digunakan dengan Secrets Manager daripada dengan teks biasa.

- Integritas artefak — Artefak yang digunakan di seluruh alur akan dienkripsi secara at rest (menggunakan kunci terkelola AWS) atau in transit (menggunakan SSL/TLS).
- Pengguna IAM dan IAM role — Izin yang diberikan ke pengguna atau sumber daya akan didasarkan pada prinsip hak akses paling rendah. Harus ada hubungan kepercayaan lintas IAM role yang mungkin perlu dikonfigurasi jika Anda beroperasi menggunakan berbagai sumber daya dalam layanan yang berbeda-beda. Misalnya, AWS CodeBuild memerlukan izin untuk menjalankan perintah pada klaster Amazon EKS.
- Audit — Kemampuan audit yang ditawarkan oleh AWS CloudTrailakan melacak setiap panggilan API di seluruh layanan dan operasi pengguna, dan memungkinkan evaluasi peristiwa terdahulu.
- Pemindaian kelemahan image — Image CNF yang diunggah ke Amazon ECR secara otomatis dipindai untuk menemukan kelemahan keamanan. Sebuah laporan dari temuan pemindaian tersedia di AWS Management Console, dan juga dapat diambil melalui API. Temuan tersebut kemudian dapat dikirim ke operator CSP untuk tindakan korektif, termasuk penggantian image CNF.

Pemeriksaan keamanan terjadi pada berbagai tahap alur untuk memastikan bahwa image yang baru diunggah aman dan lulus pemeriksaan kepatuhan yang diinginkan sehingga notifikasi dapat dikirim ke CSP untuk meminta persetujuan:

- Registri kontainer memindai kelemahan CVE yang terbuka.
- Konfigurasi akan diperiksa untuk menemukan kebocoran informasi dan pola informasi pengenal pribadi (PII) yang diketahui, selama tahap pengujian, dengan memicu aturan pemeriksaan kepatuhan untuk masalah seperti port TCP/UDP yang terbuka secara tidak terduga dan kelemahan DOS.
- Kompatibilitas mundur dan maju diverifikasi untuk keamanan upgrade/rollback.

Terlepas dari aplikasinya, sangat penting untuk menyediakan keamanan alur dengan memastikan transfer artefak yang terenkripsi di seluruh tahap, baik at rest atau in transit.

Kemampuan Pengamatan

AWS memungkinkan Kemampuan Pengamatan untuk CNF 5G yang di-deploy di AWS secara default. Hal ini diaktifkan oleh Amazon CloudWatch. CloudWatch menghadirkan visibilitas lengkap terhadap sumber daya dan aplikasi cloud Anda.

Amazon CloudWatch memiliki empat langkah besar selama proses ini:

1. Kumpulkan — Kumpulkan metrik dan log dari semua sumber daya, aplikasi, dan layanan AWS Anda yang berjalan di AWS dan server on-premise.
2. Pantau— Visualisasikan aplikasi dan infrastruktur dengan dasbor CloudWatch, korelasikan log dan metrik secara berdampingan untuk memecahkan masalah, dan atur pemberitahuan dengan CloudWatch Alarms .
3. Bertindak — Otomatiskan respons terhadap perubahan operasional dengan CloudWatch Events dan AWS Auto Scaling.
4. Analisis — Metrik hingga satu detik, retensi data yang diperpanjang (15 bulan), dan analisis waktu nyata dengan CloudWatch Metric Math .

Agen Amazon CloudWatch diinstal di klaster Kubernetes pelanggan. Agen ini mendukung fitur konfigurasi, penemuan, dan pull metrik Prometheus, sehingga memperkaya dan memublikasikan semua metrik dan metadata Prometheus berketelitian tinggi sebagai Embedded Metric Format (EMF) ke CloudWatch Logs.

Amazon CloudWatch Container Insights mengotomatiskan penemuan dan pengumpulan metrik Prometheus dari aplikasi terkontainerisasi. Layanan ini secara otomatis mengumpulkan, memfilter, dan membuat metrik CloudWatch kustom gabungan yang divisualisasikan di dasbor.

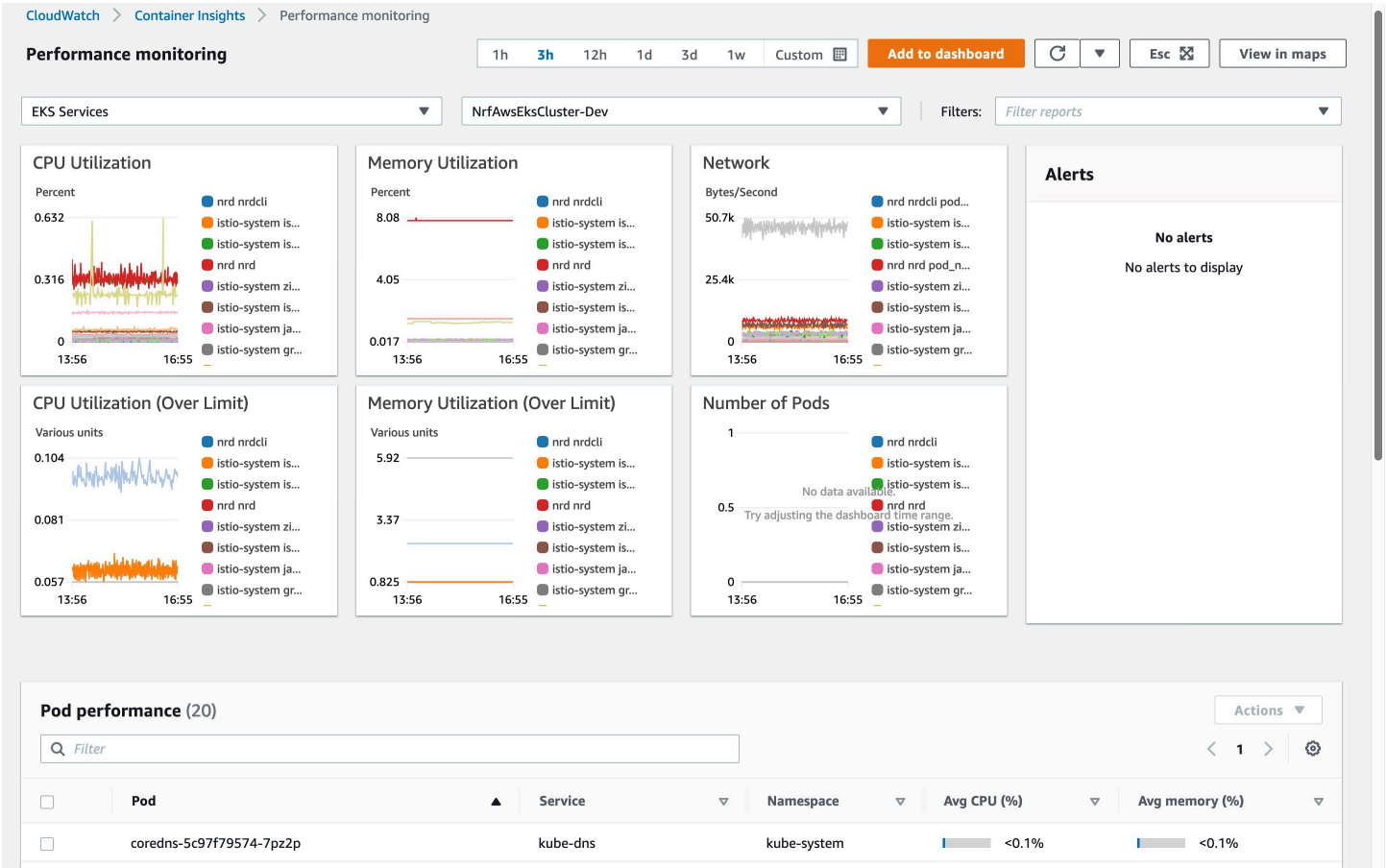
Setiap peristiwa membuat titik data metrik sebagai metrik kustom CloudWatch untuk serangkaian dimensi metrik terkurasi yang dapat dikonfigurasi sepenuhnya. Memublikasikan metrik Prometheus gabungan sebagai statistik metrik kustom CloudWatch akan mengurangi jumlah metrik yang diperlukan untuk memantau, memicu alarm tentang, dan memecahkan masalah performa dan kegagalan. Anda juga dapat menganalisis metrik Prometheus berketelitian tinggi menggunakan bahasa kueri CloudWatch Logs Insights untuk mengisolasi pod dan label tertentu yang memengaruhi kondisi dan performa lingkungan terkontainerisasi.

AWS CloudTrail menawarkan visibilitas ini, dengan merekam setiap panggilan API di seluruh layanan. AWS Config menawarkan kemampuan untuk validasi kepatuhan. AWS memberi pelanggan

opsi pemantauan metrik, log, peristiwa untuk aplikasi, infrastruktur, dan alur, menggunakan berbagai layanan seperti AWS X-Ray dan AWS CloudTrail.

- AWS dapat mengintegrasikan alat metrik sumber terbuka seperti Prometheus, Fluentd, dan sebagainya.
- Metrik Prometheus dapat diserap lebih lanjut ke Amazon CloudWatch atau OpenSearch Service untuk analisis lebih lanjut.
- AWS menggunakan FluentD sebagai mekanisme standar untuk mengumpulkan log dari berbagai sistem. Mekanisme yang sama tersebut digunakan dan dikonfigurasi untuk proyek ini.

Untuk detail tentang cara mengonfigurasi mekanisme ini, lihat Menyiapkan FluentD sebagai DaemonSet untuk Mengirim Log ke CloudWatch Logs.



Contoh metrik yang dipantau Amazon CloudWatch

Orkestrasi CI/CD dengan alat pihak ketiga dan sumber terbuka

Lapisan orkestrasi menggunakan IaC untuk men-deploy dan mengonfigurasi infrastruktur dasar yang diperlukan untuk menjalankan fungsi jaringan 5G. Lapisan ini harus dirancang agar modular, portabel, dan dapat digunakan kembali.

Infrastrukturnya mengikuti praktik terbaik cloud native, sangat tersedia, redundan, dan dapat diskalakan.

Seperti yang ditunjukkan pada bagian sebelumnya, deployment infrastruktur yang mendasari dapat dicapai dengan menggunakan AWS Cloud Development Kit. Hal ini dapat dicapai dengan menggunakan Terraform oleh Hashicorp.

Terraform

Terraform adalah perangkat lunak IaC sumber terbuka yang menyediakan alur kerja antarmuka baris perintah (CLI) yang konsisten untuk mengelola ratusan layanan cloud. Terraform mengkodifikasi API cloud ke dalam file konfigurasi deklaratif.

Untuk deployment dengan Terraform, gunakan prinsip yang sama yang digunakan dalam CDK. Kode ini terstruktur dalam modul yang memungkinkan komponen jaringan disesuaikan dan digunakan kembali sesuai dengan persyaratan vendor.

Konfigurasi ini semuanya diparameterisasi, yang memungkinkan deployment sepenuhnya disesuaikan dengan rekomendasi penyedia dan ISV.

Deployment fungsi jaringan dipisahkan dalam dua fase:

- Infrastruktur AWS yang diperlukan dibuat dan dikelola melalui repositori pusat.
- Konfigurasi dan kode disimpan secara terpusat dalam repositori GitHub.

Setelah prasyarat dibuat, fungsi jaringan siap di-deploy dengan menggunakan alur aplikasi yang ditetapkan pada tahap sebelumnya.

Deployment infrastruktur

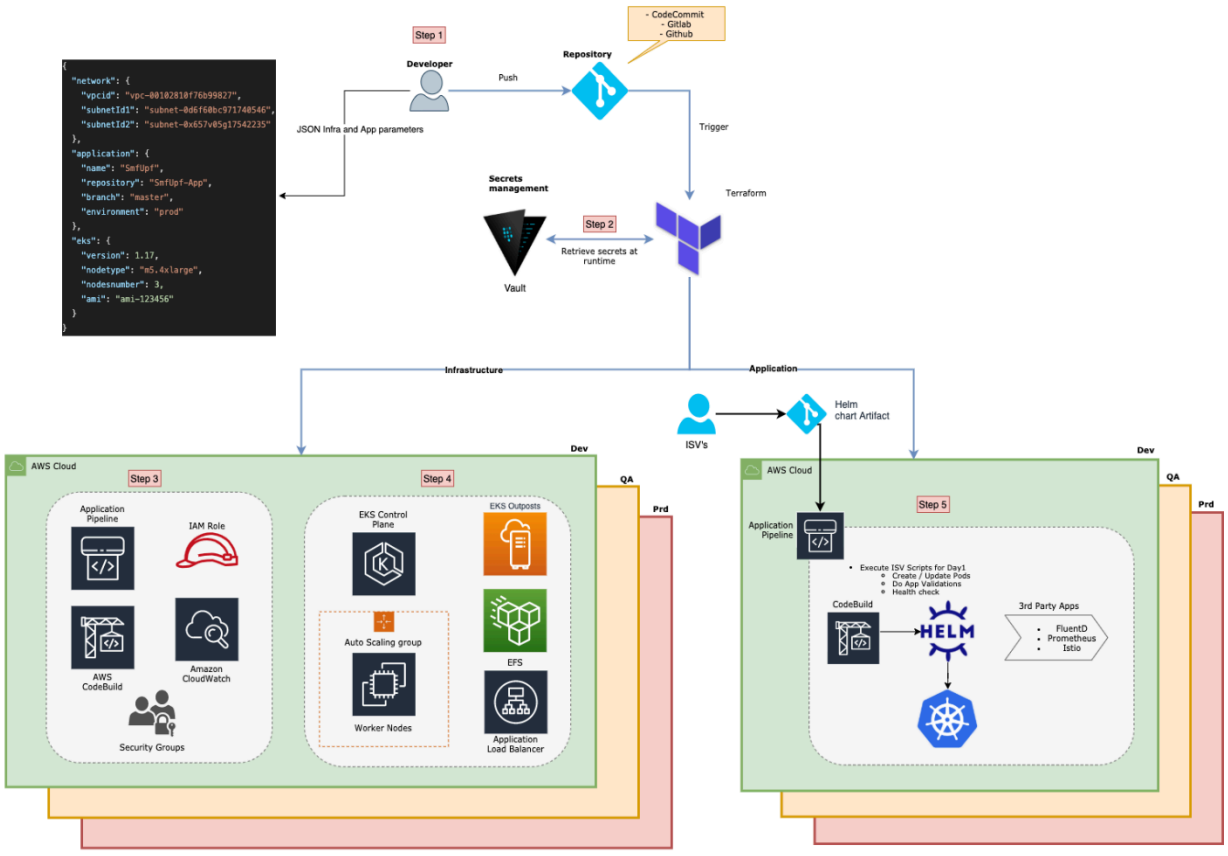
Deployment infrastruktur mencakup semua prasyarat agar fungsi jaringan berhasil di-deploy dan dikonfigurasi.

Beberapa komponen yang dibuat sebagai bagian dari fase ini adalah:

- Jaringan — VPC, subnet publik dan privat, rute, penyeimbang beban
- Komputasi — Kubernetes (Vmware Tanzu , Amazon EKS, atau AWS Outposts), node primer dan pekerja instans Amazon EC2, grup penskalaan otomatis
- Penyimpanan — Amazon EFS, Amazon EBS, bucket Amazon S3
- Keamanan — IAM role , grup keamanan
- Alur — CodePipeline, CodeBuild
- Kemampuan Pengamatan — CloudWatch, Prometheus, FluentD

Berikut adalah urutan infrastruktur yang diorkestrasi oleh Terraform dan dijelaskan pada gambar berikut:

1. Seorang developer mengisi file JSON yang disimpan dalam repositori pusat dengan kode IaC. File ini berisi informasi tentang konfigurasi infrastruktur yang diinginkan seperti ukuran instans, versi Kubernetes, informasi jaringan, dan detail repositori aplikasi.
2. Mengambil rahasia dari HashiCorp Vault atau AWS Secrets Manager pada saat runtime.
3. Men-deploy dan mengonfigurasi komponen infrastruktur (jaringan, komputasi, penyimpanan, dan keamanan).
4. Klaster Amazon EKS dengan node pekerja yang meng-host pod fungsi jaringan di-deploy. Amazon EKS juga dapat di-deploy di AWS Outposts untuk mendukung beban kerja yang memerlukan kedekatan dengan pusat data.
5. Alur aplikasi dibuat dan dikonfigurasi untuk mendengarkan perubahan dalam repositori fungsi jaringan. Setiap kali kode didorong ke cabang repositori yang dikonfigurasi, alur ini secara otomatis memicu pembangunan, pengujian, dan deployment fungsi jaringan.
6. Alat pengamatan yang mengumpulkan dan mensentralisasi log dan metrik di-deploy sebagai layanan di semua node, dan menyediakan data hampir waktu nyata yang dapat divisualisasikan di Grafana atau OpenSearch Dashboards



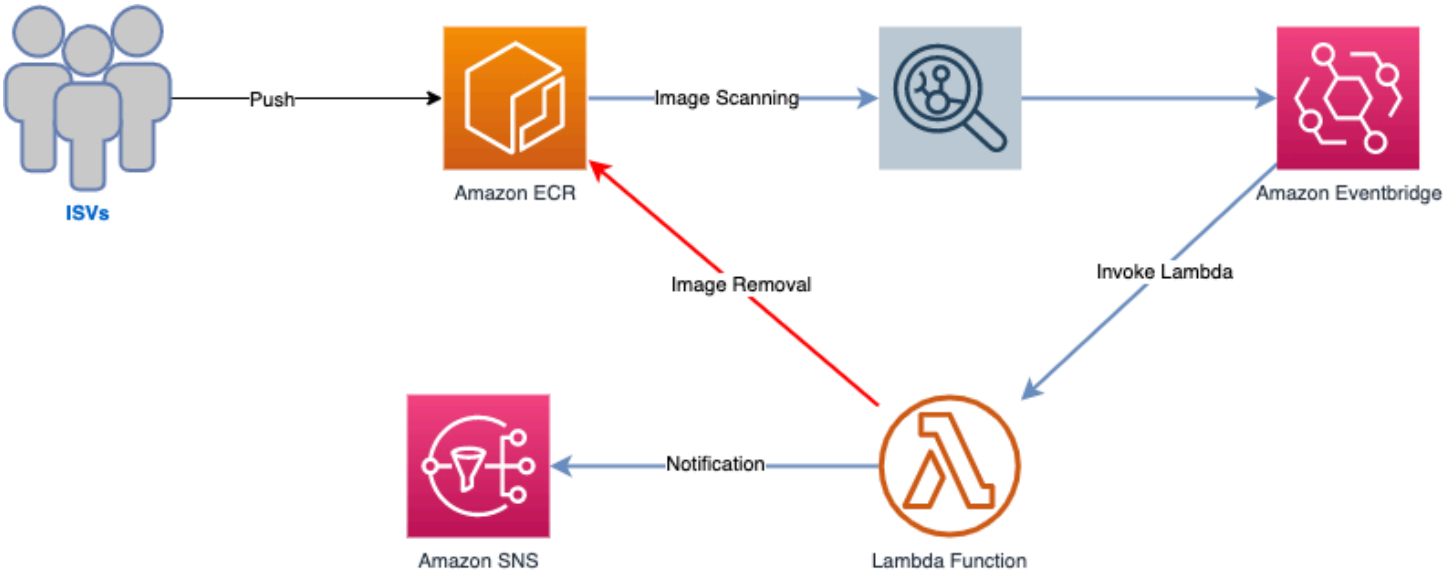
Deployment dan konfigurasi fungsi jaringan

Deployment dan konfigurasi fungsi jaringan

Alur yang dibuat pada tahap sebelumnya memungkinkan ISV dan penyedia mendesentralisasi dan mengoptimalkan deployment fungsi jaringan. Alur ini terhubung dan mendengarkan perubahan dalam repositori aplikasi, yang dikonfigurasi dalam file JSON pada langkah 1 dari gambar sebelumnya.

Untuk memeriksa image yang dipublikasikan oleh pihak ketiga, solusi pemindaian kelemahan yang membantu mengidentifikasi kelemahan perangkat lunak dalam image kontainer di-deploy dan dikonfigurasi. Solusi pemindaian secara otomatis memeriksa semua image baru yang didorong ke Amazon ECR. Untuk informasi selengkapnya tentang pemindaian image ECR, lihat Pemindaian image.

Gambar berikut menampilkan arsitektur solusi Pemindaian Kelemahan Image.



Arsitektur solusi Pemindaian Kelemahan Image

Alur aplikasi dapat dikonfigurasi agar dipicu oleh perubahan image setelah hasil pemindaian, atau oleh perubahan langsung di repositori. Misalnya, ketika image Helm baru dibuat.

Daftar berikut adalah urutan yang membuat/meng-upgrade fungsi jaringan, seperti yang direpresentasikan pada gambar berikut:

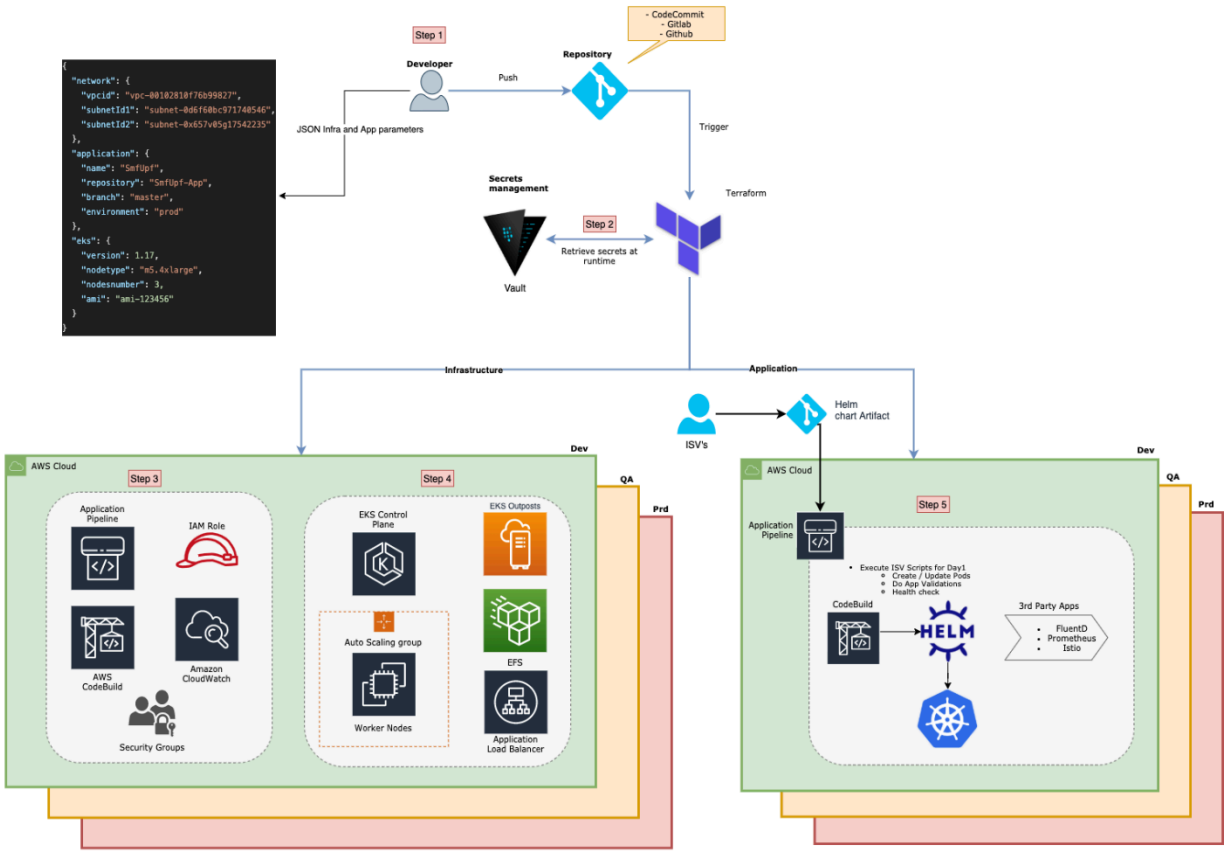
ISV mempublikasikan image baru ke Amazon ECR. (Jika image disetujui, alur aplikasi dipicu.)

CodePipeline menarik image baru dari Amazon ECR dan menggunakan CodeBuild untuk men-deploy image ke Kubernetes. Perintah Helm dapat digunakan untuk meng-upgrade fungsi jaringan.

Setelah image di-deploy, Test as Service (TaS) dipicu. TaS memvalidasi deployment baru dan mensentralisasi data dan metrik tentang performa fungsi jaringan di bawah tekanan.

Log dan metrik dikumpulkan dan disentralisasi di OpenSearch dan Grafana. Pihak ketiga seperti Datadog, Istio, dan Prometheus juga dapat dikonfigurasi untuk memberikan kemampuan pengamatan tambahan.

MANO yang dapat mengoordinasikan sumber daya jaringan juga dapat di-deploy dan diintegrasikan dengan solusi ini. Bahasa pemrograman ini menggunakan data yang dikumpulkan untuk mengambil tindakan otomatis seperti slicing jaringan dan penskalaan otomatis Quality of Service (QoS).



Alur aplikasi

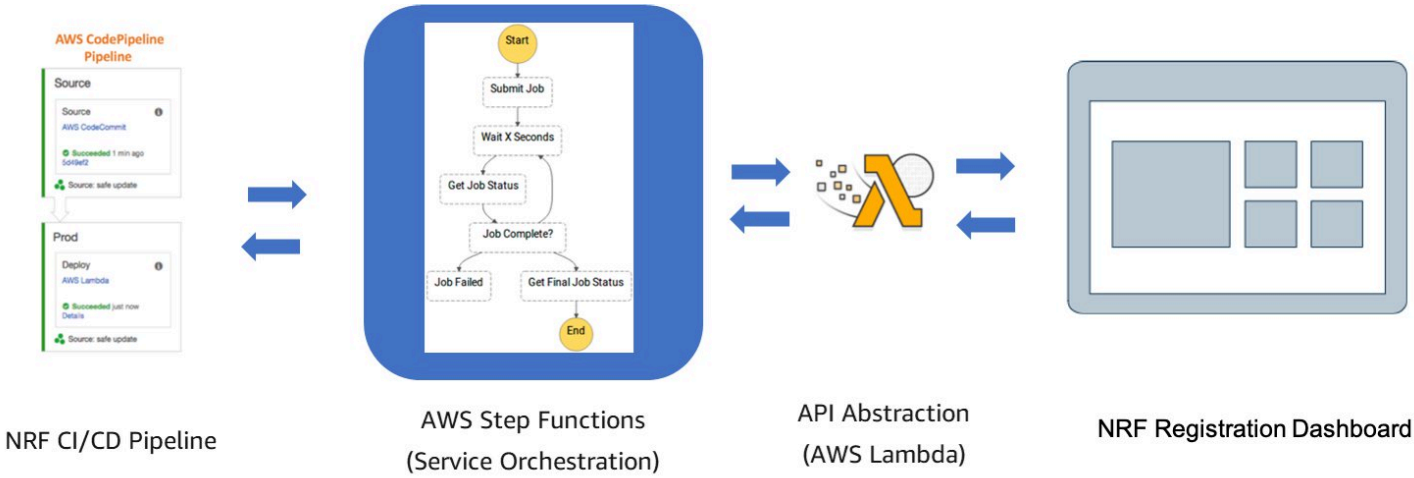
Pengujian

Kerangka kerja otomatisasi pengujian khusus telekomunikasi dapat diintegrasikan dalam alur kode. Alur kode ini terintegrasi dengan Step Functions untuk mengorkestrasi integrasi dengan kerangka kerja otomatisasi pengujian. AWS Step Functions menggunakan sejumlah fungsi Lambda untuk memanggil kerangka kerja otomatisasi pengujian (alat pihak ketiga) melalui panggilan API. Step Functions awalnya akan mendapatkan ID pengujian, menjalankan pengujian, dan mendapatkan hasil dari kerangka kerja otomatisasi pengujian. Step Functions kemudian menganalisis hasil dan meneruskannya ke alur kode untuk persetujuan aplikasi untuk deployment produksi. Persetujuan dapat otomatis atau tetap manual dalam alur kode sesuai kebutuhan. Ini merupakan langkah penting bagi CSP untuk mempromosikan deployment dari lingkungan pengujian ke produksi. API tingkat tinggi yang diperlukan untuk integrasi dikategorikan dengan cara berikut:

- Dapatkan konteks
- Eksekusi kasus pengujian tertentu

- Hentikan kasus pengujian
- Dapatkan hasil

Kompleksitas dalam memanggil API REST eksternal dimodelkan menggunakan AWS Step Functions, yang memungkinkan konstruksi standar memanggil alur paralel, menunggu hasil, melakukan pencabangan berdasarkan kondisi, dan mengintegrasikan API REST dengan AWS CodePipeline.



Alur pengujian

CI/CD dan orkestrasi

Integrasi Berkelanjutan dan Pengiriman Berkelanjutan adalah bagian dari filosofi otomatisasi keseluruhan yang hadir dengan arsitektur cloud native dan cara penerapannya untuk 5G. Orkestrasi adalah aspek lain dari filosofi ini yang harus dinamis dan reaktif terhadap setiap perubahan yang terjadi pada jaringan. Orkestrasi dan CI/CD harus saling terkait secara erat untuk menjamin layanan yang berkondisi baik dan meminimalkan gangguan layanan. Integrasi antara CI/CD dan orkestrasi harus berada di dua area:

- Penerapan patch dan upgrade ke dalam sistem perlu dikelola dan diorkestrasi dengan cara yang meminimalkan gangguan pada setiap layanan yang sedang berjalan. Misalnya, orkestrasi dapat secara dinamis menentukan waktu terbaik pembaruan harus diluncurkan.
- Orkestrasi yang mempertimbangkan CI/CD memungkinkan lalu lintas digeser selama deployment upgrade berdasarkan strategi model deployment yang diadopsi (canary, linier, atau sekaligus).

Biasanya, solusi orkestrasi berjalan di alur CI/CD untuk memungkinkan orkestrasi memperkenalkan fase tata kelola ke dalam alur tersebut dan memiliki eksposur ke siklus upgrade yang sedang berlangsung.

Kesimpulan

CI/CD menyediakan jalur yang jelas dan efisien bagi developer dan tim aplikasi untuk men-deploy kode aplikasi baru dalam hitungan menit. AWS memiliki banyak alat yang dapat membantu developer dalam integrasi, pengujian, dan deployment kode baru, termasuk, AWS CodePipeline, AWS CodeCommit, AWS CodeBuildAWS CodeDeploy, dan banyak lainnya. Dokumen ini membahas cara Layanan AWS dapat digunakan untuk membuat proses CI/CD untuk men-deploy fungsi jaringan 5G dengan cara yang sepenuhnya otomatis, termasuk langkah-langkah berbeda yang diperlukan untuk menyelesaikan deployment kode baru. Laporan resmi ini juga membahas cara mengintegrasikan kerangka kerja otomatisasi pengujian pihak ketiga ke dalam proses CI/CD, serta menggunakan alat pihak ketiga seperti Terraform.

Kontributor

Kontributor dokumen ini meliputi:

- Hisham Elshaer, Konsultan Senior (Senior Consultant), AWS Telecom, Amazon Web Services
- Vara Prasad Talari, Konsultan Utama (Principal Consultant), AWS Telecom, Amazon Web Services
- Rabi Abdel, Konsultan Utama (Principal Consultant), AWS Telecom, Amazon Web Services
- Franco Bontorin, Konsultan Senior (Senior Consultant), Shared Delivery, Amazon Web Services
- Pragtideep Singh, Konsultan (Consultant), Shared Delivery, Amazon Web Services
- Subbarao Duggisetty, Arsitek Infrastruktur Cloud (Cloud Infra Architect), Global Accounts, Amazon Web Services
- Young Jung, Arsitek Solusi Partner Senior (Senior Partner Solutions Architect), AWS Telecom, Amazon Web Services

Revisi dokumen

Untuk mendapatkan notifikasi tentang pembaruan laporan resmi ini, silakan berlangganan umpan RSS.

update-history-change	update-history-description	update-history-date
Publikasi awal	Laporan resmi pertama kali dipublikasikan	8 Maret 2021

Bacaan lebih lanjut

Untuk informasi tambahan, lihat:

- Praktik Integrasi Berkelanjutan dan Pengiriman Berkelanjutan di AWS (laporan resmi)
- Jaringan Inti Paket Seluler Kelas Operator di AWS (laporan resmi)
- Evolusi Jaringan 5G dengan AWS (laporan resmi)

Akronim

- AMF — Access & Mobility Management Function
- API — Application Programming Interface
- AUSF — Authentication Server Function
- BSS — Business Support System
- CDK — Cloud Development Kit
- CI/CD — Continuous Integration/Continuous Delivery
- CLI — Command Line Interface
- CNF — Cloud-native/Containerized Network Function
- CSP — Communication Service Provider
- CU — RAN Central Unit
- CVE — Common Vulnerabilities and Exposures
- DoS — Denial of Service
- DR — Disaster Recovery
- DU — RAN Distributed Unit
- E2E — End to End
- ECR — Elastic Container Registry
- EFS — Elastic File System
- EKS — Elastic Kubernetes Service
- EPC — Evolved Packet Core
- IaC — Infrastructure as Code
- ISV — Independent Software Vendor
- MANO — Management and Orchestration
- MEC — Multi-Access Edge Computing
- NACL — Network Access Control List
- NAT — Network Address Translation
- NF — Network Function
- NFV — Network Function Virtualization
- NFVO — Network Function Virtualization Orchestrator

- NOC — Network Operations Centre
- NRF — Network Repository Function
- OSS — Operations Support System
- PII — Personally Identifiable Information
- QoS — Quality of Service
- RAN — Radio Access Network
- SBI — Service Based Interface
- SMF — Session Management Function
- SSL — Secure Sockets Layer
- TaS — Test as Service
- TCP — Transmission Control Protocol
- TLS — Transport Layer Security
- UDM — Unified Data Management
- UDP — User Datagram Protocol
- UPF — User Plane Function
- VIM — Virtualized Infrastructure Manager
- VNF — Virtual Network Function
- VPC — Virtual Private Cloud

Pemberitahuan

Pelanggan bertanggung jawab untuk membuat penilaian independen mereka sendiri atas informasi dalam dokumen ini. Dokumen ini: (a) hanya disediakan sebagai informasi, (b) berisi penawaran produk dan praktik AWS saat ini, yang dapat berubah tanpa pemberitahuan, dan (c) tidak menjadi komitmen atau jaminan apa pun dari AWS dan afiliasi, pemasok, atau pemberi lisensinya. Produk atau layanan AWS disediakan “sebagaimana adanya” tanpa jaminan, representasi, atau ketentuan apa pun, baik tersurat maupun tersirat. Tanggung jawab dan kewajiban AWS kepada pelanggannya dikendalikan oleh perjanjian AWS, dan dokumen ini bukan bagian dari, juga tidak mengubah, perjanjian apa pun antara AWS dan pelanggannya.