Whitepaper da AWS

Integração e entrega contínuas para redes 5G na AWS

Integração e entrega contínuas para redes 5G na AWS: Whitepaper da AWS

Table of Contents

Resumo i
Resumo 1
Introdução 2
Integração contínua e entrega contínua 4
Integração contínua 4
Entrega e implantação contínuas 4
Infraestrutura como código 4
CI/CD na AWS 5
Redes 5G na AWS 9
CI/CD em redes 5G 9
Etapas detalhadas de CI/CD 11
Configuração de rede 12
Implantação de infraestrutura 12
Implantação de função de rede nativa da nuvem 13
Entrega contínua de CNF 14
Segurança 17
Capacidade de observação 18
Orquestração de CI/CD com ferramentas de terceiros e de código aberto 21
Terraform 21
Implantação de infraestrutura 22
Implantação e configuração da função de rede 23
Teste 25
CI/CD e orquestração 27
Conclusão 28
Colaboradores 29
Revisões do documento 30
Leitura adicional 31
Siglas 32
Avisos 34

Integração e entrega contínuas para redes 5G na AWS

Data de publicação: 8 de março de 2021 (Revisões do documento)

Resumo

Este whitepaper apresenta a integração e a entrega contínuas (CI/CD) para redes 5G e explica como as ferramentas e os serviços da Amazon Web Services (AWS) podem ser usados para automatizar totalmente a implantação e os upgrades de funções de redes 5G. Ele fornece uma descrição detalhada dos diferentes estágios de CI/CD para funções de rede 5G, incluindo configuração de rede, implantação de infraestrutura, implantação de funções de rede nativas da nuvem e atualizações contínuas de funções de rede. Ele também fornece detalhes sobre a integração com ferramentas de código aberto e de terceiros para teste, capacidade de observação e orquestração.

Este whitepaper é destinado a provedores de serviços de comunicação (CSPs), bem como provedores independentes de software (ISVs).

Introdução

Historicamente, o desenvolvimento, os testes de integração de laboratório e campo e a implantação de produção de novos nós de rede ou novos recursos em uma rede celular levavam semanas ou até meses para garantir a estabilidade dos serviços de telecomunicações (telecomunicações) de missão e negócios críticos. O longo ciclo de implantação foi causado pela arquitetura monolítica dos nós de rede tradicionais, um ambiente de vários fornecedores e muitas interfaces ponto a ponto entre entidades nas redes móveis 2G, 3G e 4G.

Conforme apresentado no whitepaper 5G Network Evolution with AWS (Evolução da rede 5G com a AWS), as redes móveis 5G, conforme padronizado pela 3GPP, agora são compatíveis com uma arquitetura nativa da nuvem habilitada por virtualização e conteinerização. Mais especificamente, as redes 5G introduzem e são compatíveis com um novo paradigma da arquitetura de microsserviços, sem estado e baseada em serviços.

Essa arquitetura 5G significa que diferentes funções de rede podem funcionar como serviços independentes com acoplamento fraco que se comunicam entre si por meio de interfaces e APIs bem definidas. Mais importante ainda, cada função de rede pode ser atualizada de forma independente. Essa mudança de arquitetura no 5G permite que os CSPs obtenham mais agilidade e eficiência operacional, facilitando a implantação de atualizações para funções de rede com mais frequência, mantendo os testes, os requisitos de segurança e os padrões por meio da automação.

A integração e a implantação de novos recursos para um CSP geralmente começam quando o fornecedor da função de rede lança um novo pacote de software de função de rede, como uma imagem do Docker em uma função de rede baseada em contêiner ou um novo arquivo de configuração, como um chart do Helm no caso da aplicação Kubernetes. (Um chart do Helm é uma coleção de arquivos que descrevem um conjunto relacionado de recursos do Kubernetes).

A ideia de usar o paradigma de CI/CD para implantação da função de rede 5G está ganhando força, mas a realização prática dessa ideia tem sido um desafio no setor de telecomunicações.

A AWS foi pioneira no desenvolvimento de novas ferramentas de CI/CD para entrega de software a fim de ajudar uma ampla gama de setores a desenvolver e implementar mudanças de software rapidamente, mantendo a estabilidade e a segurança dos sistemas. Essas ferramentas incluem um conjunto de serviços de desenvolvimento e operações de software (DevOps), como AWS CodeStar, CodeCommit, CodePipeline, CodeBuild e CodeDeploy.

A AWS também promove a ideia de Infraestrutura como Código (IaC) com o uso do AWS Cloud Development Kit (AWS CDK), do AWS CloudFormation e de ferramentas de terceiros baseadas em

API, como o Terraform. Com o uso dessas ferramentas, a AWS pode armazenar os processos de implantação da função de rede na AWS como código-fonte e manter esse código-fonte do IaC no pipeline de CI/CD para realizar a entrega contínua.

Este whitepaper descreve processos detalhados para utilizar as ferramentas de IaC e CI/CD da AWS para a implantação e atualização da função de rede 5G. Além disso, ele aborda a integração com ferramentas de terceiros para teste, capacidade de observação e orquestração.

As ferramentas de CI/CD da AWS não estão restritas às funções de rede 5G. Elas também são empregadas para automatizar a implantação de redes 4G, o que permite que os CSPs implantem e atualizem de forma rápida e eficiente as funções de rede 4G. A maioria das funções de rede 4G é baseada na Função de Rede Virtual (VNF). Conjuntos de ferramentas de CI/CD da AWS AWS CloudFormation podem ser usados para automatizar a implantação de VNFs 4G, trazendo escala e eficiência em termos de tempo para implantações de rede 4G.

Integração contínua e entrega contínua

Integração contínua

Integração contínua (CI) é um processo de software no qual os desenvolvedores enviam regularmente seu código para um repositório central, como AWS CodeCommit ou GitHub. Cada envio de código aciona uma construção automatizada, seguida pela execução de testes. O principal objetivo da CI é descobrir problemas de código em um estágio inicial, melhorar a qualidade dele e reduzir o tempo necessário para validar e lançar novas atualizações de software.

Entrega e implantação contínuas

Entrega contínua (CD) é um processo de software no qual os artefatos são implantados no ambiente de teste, no ambiente de preparação e no ambiente de produção. A entrega contínua pode ser totalmente automatizada ou ter estágios de aprovação em pontos críticos. Isso garante que todas as aprovações necessárias, como a aprovação do gerenciamento de versões, sejam realizadas antes da implantação. Quando a integração contínua for implementada corretamente, os desenvolvedores sempre terão um artefato de criação pronto para ser implantado, e que passou por um processo de teste padronizado.

Com a implantação contínua, as revisões são implantadas em um ambiente de produção automaticamente, sem aprovação explícita de um desenvolvedor, automatizando todo o processo de lançamento de software. Isso permite um circuito de feedback contínuo do cliente no início do ciclo de vida do produto.

Com a implantação contínua, todas as alterações confirmadas e aprovadas nos testes automatizados são liberadas para produção automaticamente. A entrega contínua não se destina a liberar todas as alterações confirmadas e passar os testes automatizados para a produção imediatamente, mas, sim, garantir que todas as alterações estejam prontas para irem para produção.

Infraestrutura como código

Conforme detalhado no whitepaper 5G Network Evolution with AWS (Evolução da rede 5G com a AWS), a IaC é um fator importante para automatizar o processo de provisionamento e o gerenciamento do ciclo de vida da aplicação e de seu ambiente. Em vez de confiar em etapas realizadas manualmente, os administradores e desenvolvedores de rede/TI podem instanciar a

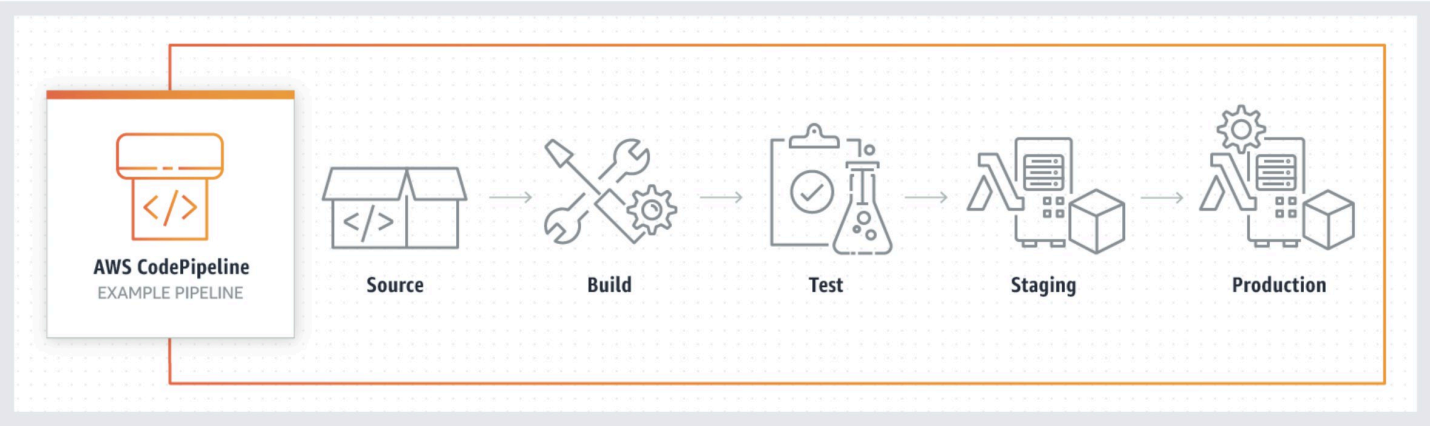
infraestrutura usando arquivos de configuração. A IaC trata esses arquivos de configuração como código de software. Esses arquivos podem ser usados para produzir um conjunto de artefatos, isto é, serviços de computação, armazenamento, rede e aplicações que compõem um ambiente operacional. A IaC elimina o desvio da configuração por meio da automação, aumentando assim a velocidade e a agilidade das implantações de infraestrutura.

No caso da implementação da virtualização de funções de rede (NFV) na AWS, essa framework de IaC agrega valor do ponto de vista da orquestração. Desde a criação da nuvem privada virtual (VPC) até a implantação da função de rede, cada etapa pode ser programada, gerenciada como código-fonte e mantida com controle de versão no AWS CodeCommit.

Essa framework de IaC para função de rede resulta em uma infraestrutura repetível e confiável e na criação e na implantação de funções de rede, que podem ser estendidas para a automação de ponta a ponta (E2E) do gerenciamento de fatia de rede e gerenciamento do ciclo de vida do serviço. A AWS oferece um conjunto de ferramentas abrangente para criar, manter e implantar uma infraestrutura de forma programática, descritiva e declarativa usando serviços, como AWS CloudFormation, AWS CDK, AWS CDK para Kubernetes e exposição de APIs de todos os serviços da AWS.

CI/CD na AWS

A CI/CD pode ser representada como um pipeline, em que o novo código é enviado em uma extremidade, testado em uma série de estágios (fonte, construção, teste, preparação e produção) e, então, publicado como código pronto para produção.



Visão geral do pipeline de CI/CD

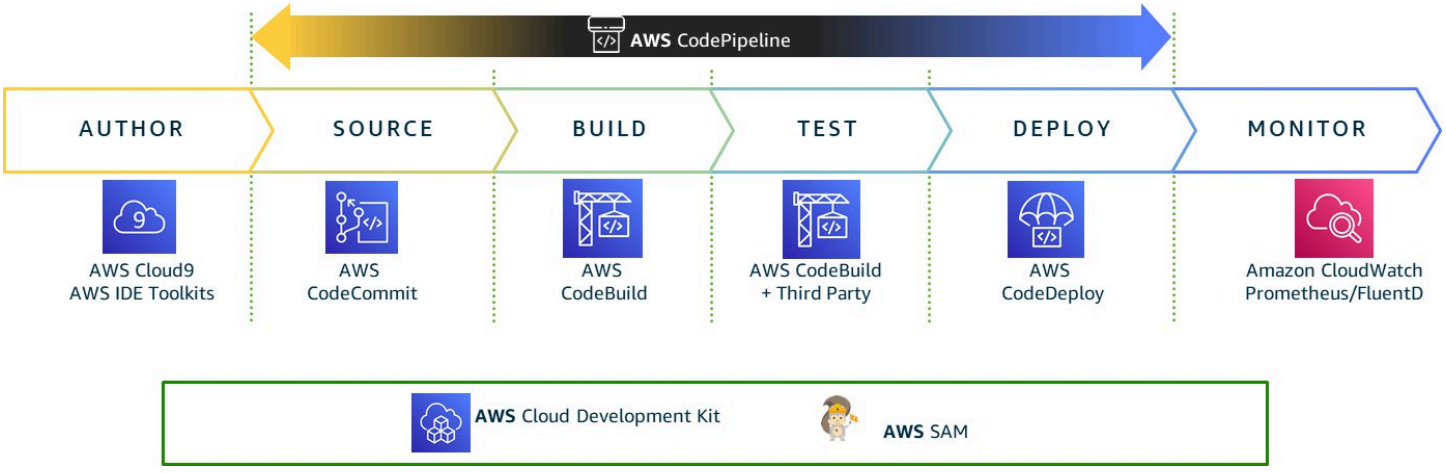
Cada estágio do pipeline de CI/CD é estruturado como uma unidade lógica no processo de entrega. Cada estágio atua como um portão que examina um determinado aspecto do código. À medida

que o código avança pelo pipeline, a suposição é que a qualidade do código é maior nos estágios posteriores, porque mais aspectos dele continuam a ser verificados. Problemas descobertos em um estágio inicial impedem que o código progrida pelo pipeline. Os resultados dos testes são enviados imediatamente à equipe, e todas as outras construções e lançamentos serão interrompidos se o software não passar do estágio.

A AWS traz um conjunto completo de ferramentas de desenvolvedor de CI/CD para acelerar o desenvolvimento de software e os ciclos de lançamento. O AWS CodePipeline automatiza as fases de construção, teste e implantação do processo de lançamento sempre que há uma alteração de código, com base no modelo de versão definido. Isso permite a entrega rápida e confiável de recursos e atualizações.

Os pipelines de código podem se integrar a outros serviços. Podem ser serviços da AWS, como o Amazon Simple Storage Service (Amazon S3), ou produtos de terceiros, como o GitHub. O AWS CodePipeline pode resolver uma variedade de casos de uso de desenvolvimento e operação, entre eles:

- Compilação, construção e teste de código com AWS CodeBuild
- Entrega contínua de aplicações baseadas em contêiner para a nuvem
- Validação pré-implantação de artefatos (como descritores e imagens de contêiner) necessários para o serviço de rede ou funções de rede específicas nativas da nuvem
- Testes funcionais, de integração e de performance para função de rede conteinerizada/função de rede virtual (CNF/VNF), incluindo testes de linha de base e regressão
- Teste de confiabilidade e recuperação de desastres (DR).



Componentes do pipeline do AWS CICD

A AWS pode configurar pipelines de CI/CD usando as seguintes ferramentas do desenvolvedor da AWS:

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

A criação de pipeline de CI/CD pode ser automatizada usando AWS CDK e AWS CloudFormation. No domínio NFV, essa automação nativa da AWS pode ser integrada a uma framework de gerenciamento e orquestração (MANO) e à framework de orquestração de serviços do CSP.

Esse processo de CI/CD inclui as seguintes etapas:

- Configuração de rede: AWS CDK e AWS CloudFormation iniciam a criação dos pré-requisitos de rede:
    - Pilha de rede (VPC, sub-redes, gateway de conversão de endereço de rede (NAT), tabela de rotas e internet gateway)
- Implantação de infraestrutura: AWS CDK e AWS CloudFormation iniciam a criação das seguintes pilhas de recursos:
    - Pilha de computação (criação de cluster do Amazon Elastic Kubernetes Service (Amazon EKS), nós do operador do EKS, AWS Lambda)
    - Pilha de armazenamento (buckets do Amazon S3, volumes do Amazon Elastic Block Store (Amazon EBS) e Amazon Elastic File System (Amazon EFS)
    - Pilha de monitoramento (CloudWatch ,Amazon OpenSearch Service (OpenSearch Service)
    - Pilha de segurança (AWS Identity and Access Management (AWS IAM), grupos de segurança do Amazon Elastic Compute Cloud (Amazon EC2), listas de controle de acesso à rede (NACLs) da VPC

- Implantação da Cloud Network Function (CNF): neste estágio, a CNF é implantada em clusters do EKS usando as ferramentas de chart do Kubectl e do Helm. Essa etapa também implanta todas as aplicações ou ferramentas específicas de que as CNFs precisam para funcionar de forma eficiente (como Prometheus ou Fluentd). As CNFs podem ser implantadas por meio de funções do Lambda ou com o AWS CodeBuild.
- Atualizações e implantação contínuas: são uma sequência de etapas executadas iterativamente para implantar alterações que fazem parte das alterações de contêiner/configuração que resultam em atualizações. Semelhante ao caso de implantação das CNFs, as atualizações e a implantação contínuas podem ser automatizadas com o uso dos serviços da AWS, com o acionador do AWS CodeCommit, o Amazon Elastic Container Registry (Amazon ECR) ou um sistema de origem de terceiros, como o GitLab Webhooks .

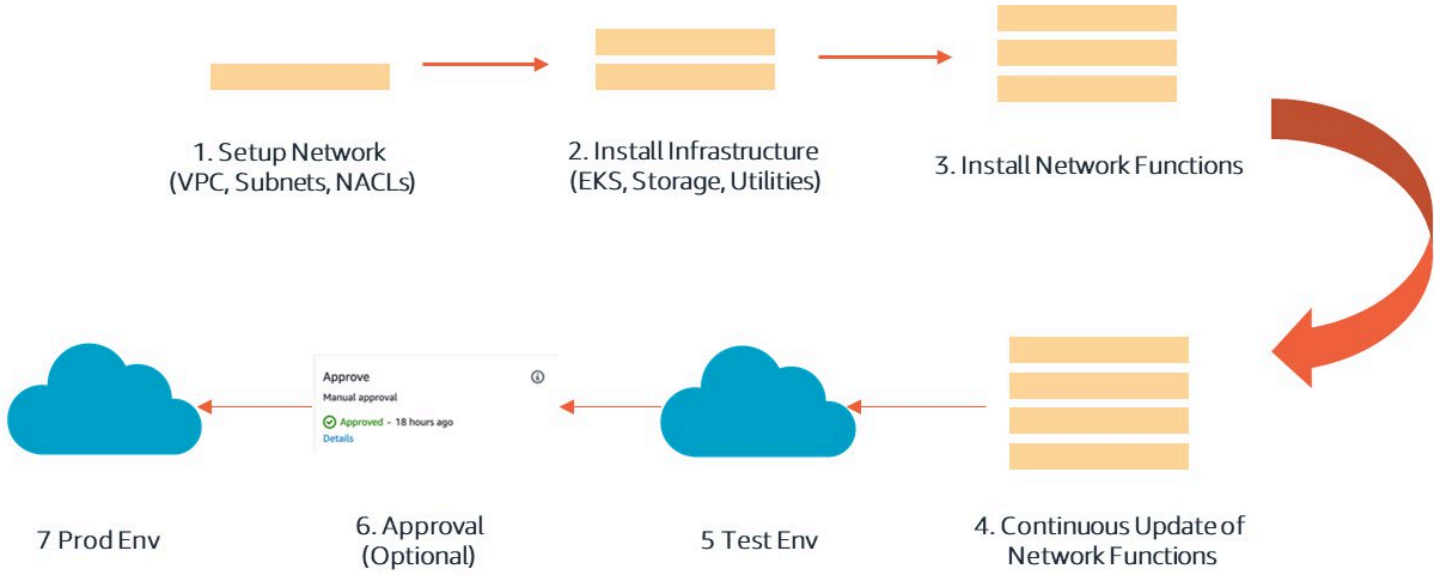


Diagrama de fluxo do pipeline do AWS CICD

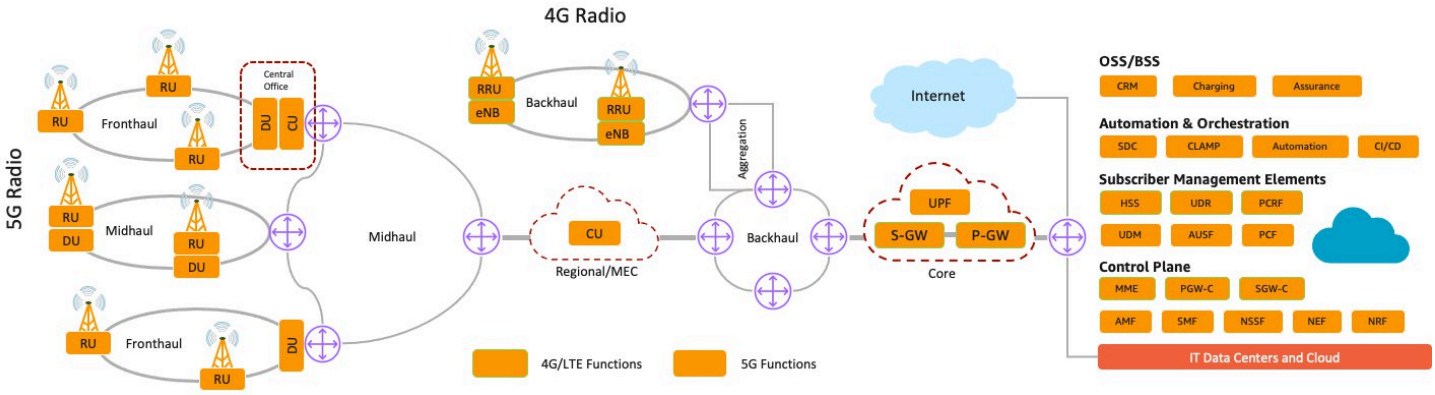
O pipeline de CI/CD é criado usando o AWS CodePipeline e utiliza um serviço de entrega contínua que modela, visualiza e automatiza as etapas necessárias para liberar o software. Ao definir estágios em um pipeline, você pode recuperar o código de um repositório de código-fonte, criar esse código-fonte em um artefato liberável, testar o artefato e implantá-lo na produção. Somente o código que passar com êxito por todos esses estágios será implantado. Você também pode adicionar outros requisitos ao seu pipeline, como aprovações manuais, para ajudar a garantir que apenas as alterações aprovadas sejam implantadas na produção.

Redes 5G na AWS

O modelo típico de infraestrutura de rede 5G é composto por um site de rádio 4G/5G, uma rede fronthaul/midhaul/backhaul, um site de rede principal e um datacenter de telecomunicações/TI. Os CSPs podem usar os serviços da AWS para criar uma infraestrutura de rede 5G escalável e flexível e, ao mesmo tempo, reduzir o custo inicial do investimento. A AWS pode ser usada para implementar o Network Operation Center (NOC) virtual na região que hospeda o Operations Support System/ Business Support System (OSS/BSS) e a maioria das funções de rede central do plano de controle.

A AWS também pode ser utilizada para implementar o escritório central (CO) local ou o datacenter distribuído com uma frota de instâncias do AWS Outposts que hospedam principalmente funções de plano de usuário, como UPF (função de plano de usuário), unidade central (CU) de RAN e computação de borda de acesso múltiplo (MEC). Há uma explicação mais detalhada da arquitetura de referência e os benefícios da implementação da rede 5G na AWS no whitepaper 5G Network Evolution with AWS (Evolução da rede 5G com a AWS).

Quando chegar a hora de implementar a rede 5G na AWS, as ferramentas de CI/CD da AWS, apresentadas nas seções a seguir deste whitepaper, podem facilitar a automação completa da implantação, atualização e gerenciamento do ciclo de vida das funções de rede 5G.

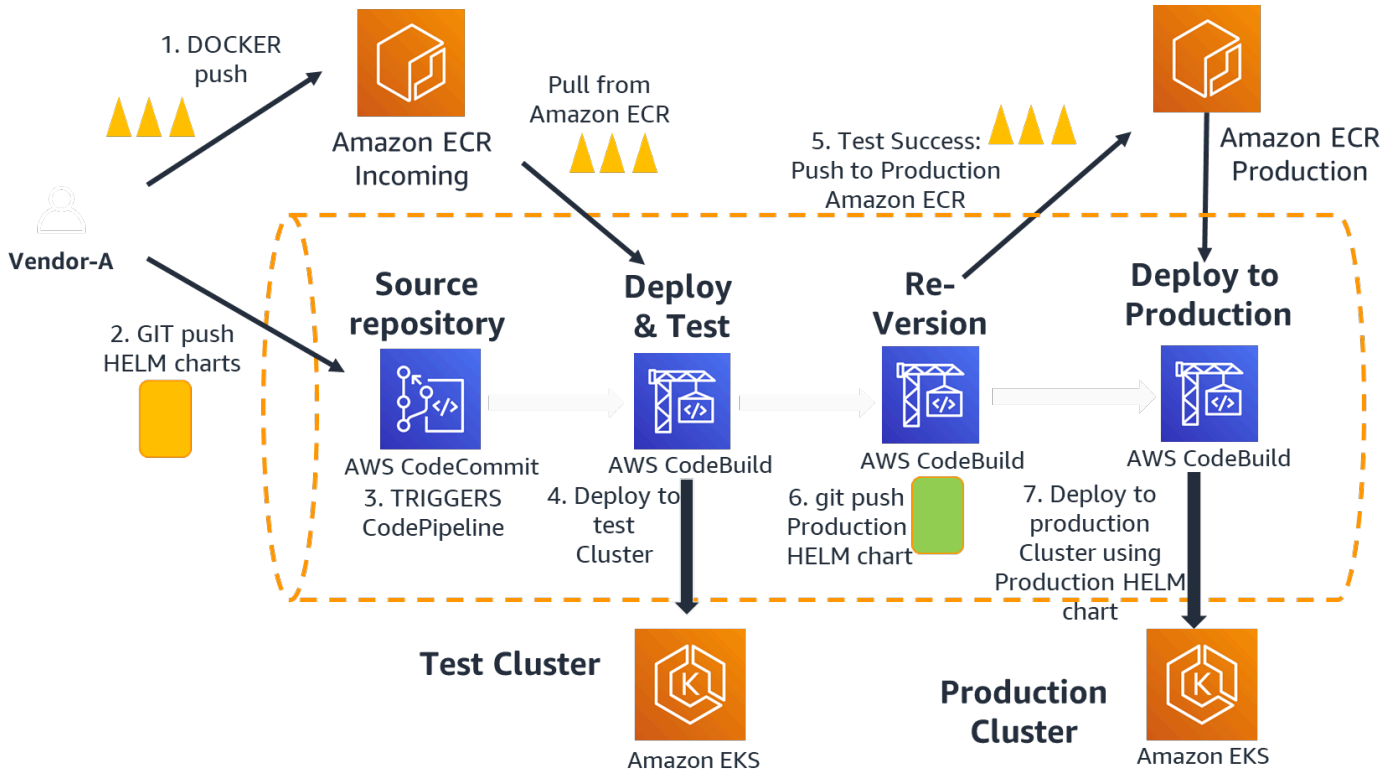


Arquitetura E2E da rede 5G

CI/CD em redes 5G

A construção de design da infraestrutura é armazenada na forma de código usando linguagem declarativa. Isso permite que o CSP tenha uma reprodução repetível da infraestrutura com o mesmo comportamento esperado conforme necessário. O código é mantido no repositório de código e um pipeline é configurado para orquestrar atualizações para as pilhas implantadas (por exemplo, AWS

CDK e AWS CloudFormation). A AWS pode ajudar a criar a Infraestrutura como código (IaC) para integração ágil de funções de provedores independentes de software (ISV).



Fluxo do pipeline de código

As alterações nas configurações de funções de rede nativas da nuvem por meio de chart do Helm são consideradas gatilhos para a execução automática de um pipeline de CI/CD para funções de rede.

O AWS CodeCommit pode ser usado para manter arquivos de configuração, e o Amazon ECR pode ser usado para preservar imagens de contêiner.

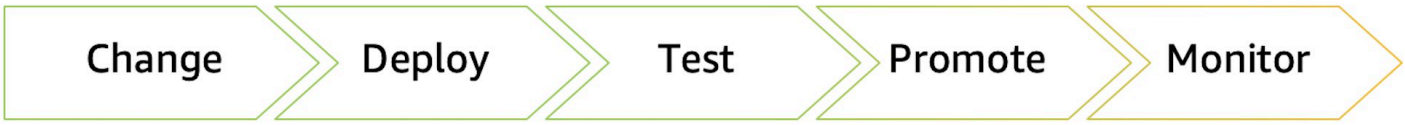
Conforme mostrado na figura Fluxo do pipeline de código, quando o ISV envia novas alterações de código para o repositório de código (chart do Helm, arquivos de configuração ou um arquivo de propriedades), o pipeline de código é acionado. O pipeline de código extrai a imagem do ECR e usa o chart do Helm para implantar a aplicação. Os testes da nova aplicação podem ser integrados à framework de automação de teste de terceiros. Com base no resultado, os CSPs podem aprovar a implantação de produção.

O estágio de origem do CodePipeline procura alterações nos arquivos de configuração. Os provedores válidos para o estágio de origem são CodeCommit, Amazon S3, GitHub ou AWS CloudFormation. Sistemas de origem alternativos podem ser integrados usando funções do Lambda

para implementar Webhooks, o que permite a integração orientada por eventos entre o Gitlab e o AWS CodePipeline. Consulte os links a seguir para obter um guia de implementação detalhado.

- Webhooks com GitLab
- Integrações de registro de contêiner

O projeto do pipeline de CI/CD deve levar em conta as etapas críticas de implantação, como implantação inicial, teste e promoção para produção após os resultados do teste estarem alinhados com as expectativas e verificados em relação à linha de base. Cada estágio do processo de pipeline fornece artefatos de dados, que permitem a comparação e decisões orientadas por dados.



Etapas do pipeline de CI/CD da aplicação

Cada estágio pode ser considerado uma tarefa separada, permitindo a incorporação de fluxos de trabalho de validação e implantação adequados para dar suporte ao serviço de rede e às funções de rede nativas da nuvem. As tarefas de execução podem incorporar ferramentas adicionais de terceiros, como geradores de tráfego e simuladores, permitindo a validação do serviço de rede de ponta a ponta.

A AWS fornece o serviço sofisticado AWS Step Function (máquina de estado nativa da nuvem) que se integra nativamente a outros serviços da AWS e também tem a capacidade de se integrar a sistemas externos, como o Jira ou uma framework de automação de teste.

Etapas detalhadas de CI/CD

A CI/CD pode ser representada como um pipeline, em que o novo código é enviado em uma extremidade, testado em uma série de estágios (fonte, construção, teste, preparação e produção) e, então, publicado como código pronto para produção.

Veja as etapas de implantação e teste. A implantação e a configuração são divididas em quatro seções principais:

- Configuração de rede

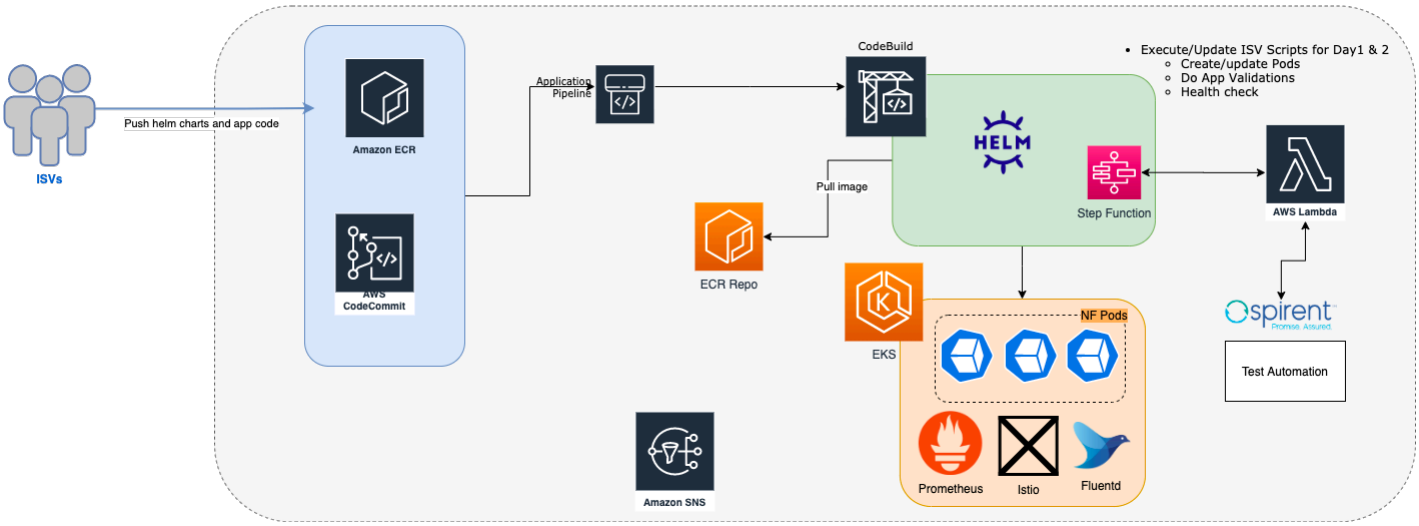
- Implantação de infraestrutura
- Implantação de função de rede nativa da nuvem
- Entrega contínua de CNF

Configuração de rede

Concentre-se na configuração dos pré-requisitos de infraestrutura. Isso inclui a criação da VPC, redes, sub-redes, NACLs, etc. Projete o plano de rede IP considerando os ISVs e o plano de implementação do cliente (como alocações de multilocação e estáticas em comparação com dinâmicas). Esse plano pode ser codificado em AWS CDK ou AWS CloudFormation. A execução desse código implanta os pré-requisitos de rede da infraestrutura de nuvem.

Implantação de infraestrutura

A implantação da infraestrutura provisiona quaisquer componentes da infraestrutura. Ela inclui a geração do cluster do EKS e da infraestrutura de suporte, como EFS, nós do operador do EKS, ELBs e a configuração do cluster de acordo com os requisitos da função de rede nativa da nuvem. Com base nos requisitos da CNF, a AWS também implanta interfaces de rede extras para os nós, incluindo interfaces Multus. A maioria das etapas de implantação e configuração corresponde a um esforço único para uma aplicação e é atualizada somente quando necessário como uma atualização para a aplicação.

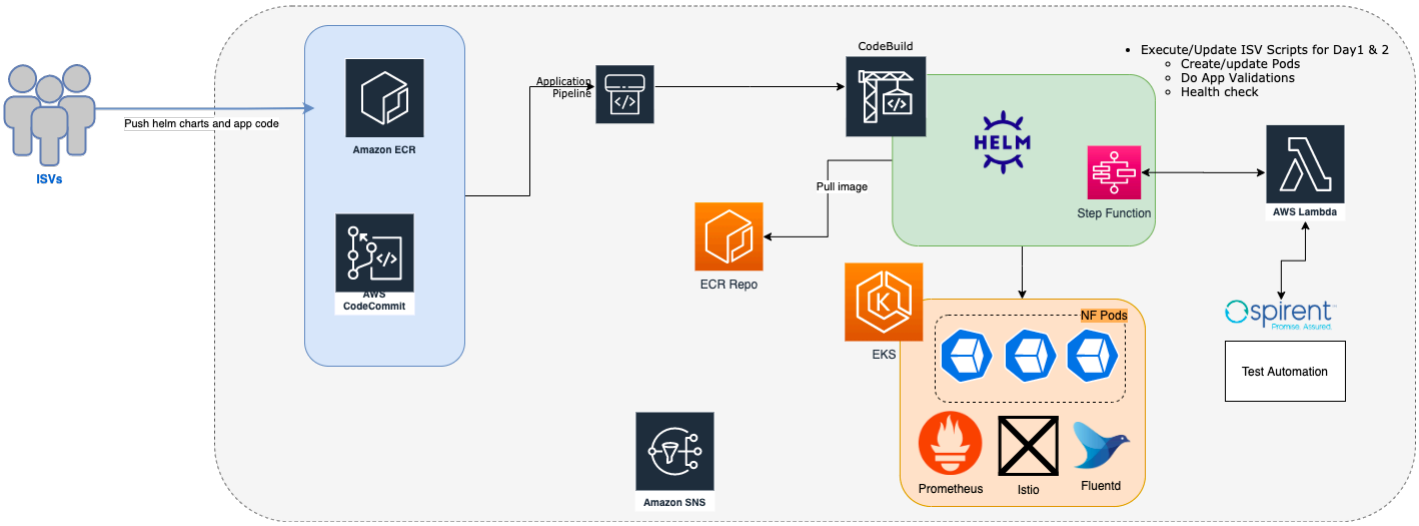


Implantação da infraestrutura com CDK

Implantação de função de rede nativa da nuvem

A implantação da CNF é relativa à implantação da aplicação. Como parte da implantação da CNF, os charts do Helm para a aplicação são implementados por meio do pipeline de código CI/CD. É incorporado o retorno de chamada para executar scripts específicos de aplicações individuais envolvendo principalmente pré e pós-verificações. Os charts do Helm são implementados em uma sequência de acordo com as necessidades da aplicação e verificam o status dos PODs do Kubernetes antes de passar para a próxima etapa da implantação. Muitas vezes, os ISVs fornecem um script wrapper para executar os charts do Helm e as verificações de sanidade. Esses scripts ISV são invocados de dentro do AWS CodePipeline. Como parte dessa fase, os agentes de registro em log e monitoramento, como Prometheus e Fluentd, são implantados além do Amazon CloudWatch, que registra e monitora a infraestrutura de nuvem da aplicação.

O pipeline de código é integrado à framework de automação de teste de terceiros. O pipeline de código pode chamar diretamente as APIs da framework de automação de teste para executar o teste na aplicação implantada, consultar os resultados do teste e analisar o resultado. Isso simplifica a implantação e o teste da aplicação.

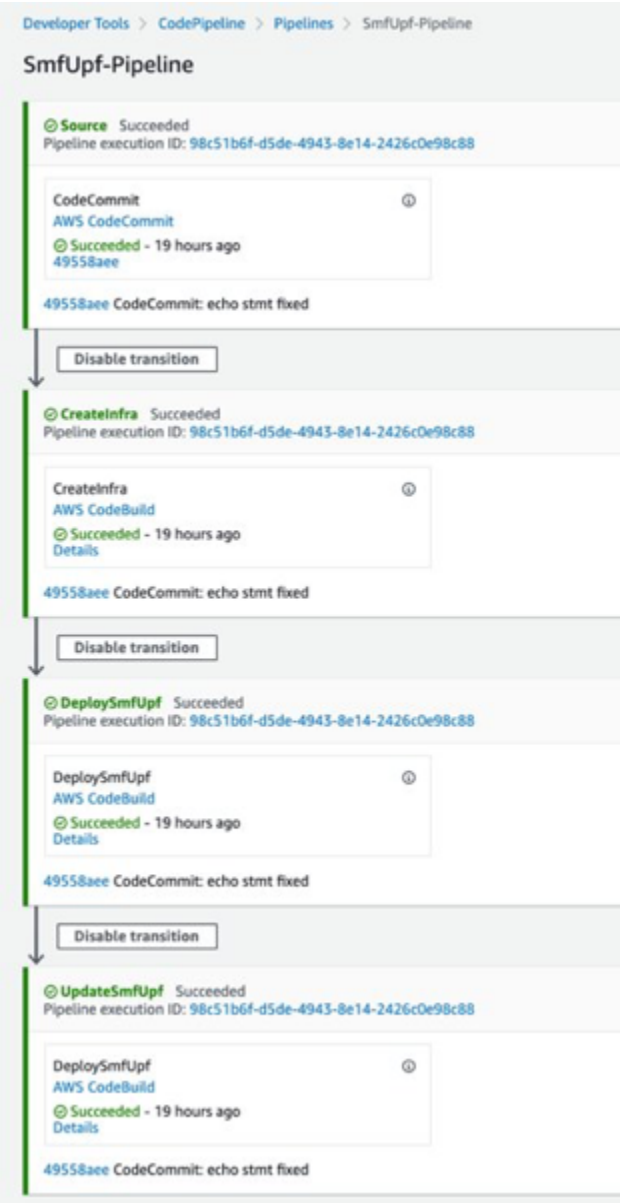


Implantação e atualização de aplicações

Veja um exemplo da implantação da função de plano do usuário/função de gerenciamento de sessão (UPF/SMF) CNF pelo AWS CodePipeline.

- Automatize o processo completo de CI/CD usando CodeCommit, CodeBuild e CodePipeline.
- As tarefas de criação de infraestrutura e instalação de aplicações são integradas como parte do pipeline.

- Os agentes FluentD e Prometheus são instalados e criados nos painéis do Amazon CloudWatch.



Exemplo de implantação de CNFs UPF/SMF

Entrega contínua de CNF

Consiste em uma sequência de etapas executadas repetidamente para implantar alterações que fazem parte das alterações de contêiner/configuração que resultam em atualizações. A entrega contínua de CNF é automatizada por meio de pipelines e é específica de aplicações individuais. A AWS usa charts do Helm padrão para atualizar CNFs específicas. O pipeline de código tem

verificações prévias e posteriores para o status de atualização da aplicação. O pipeline de CI/CD atualizado também é integrado a uma framework de automação de teste para executar testes automatizados. Essa abstração permite uma implantação limpa das funções de rede.

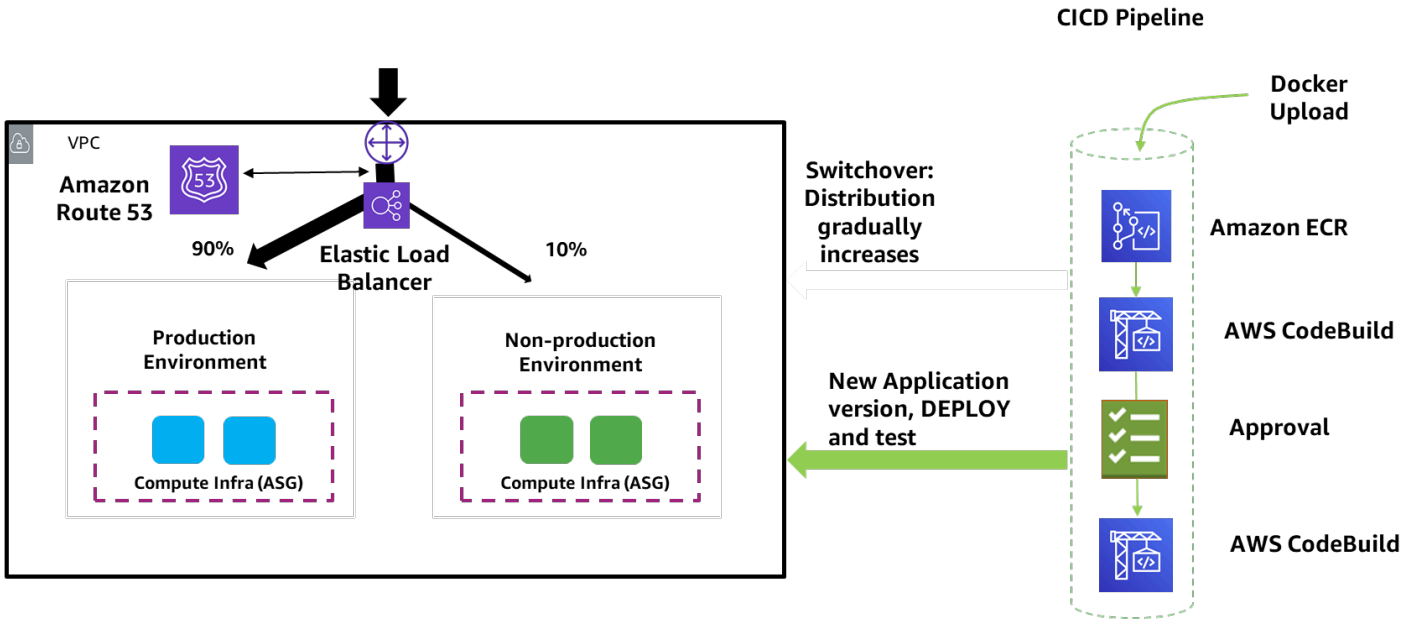
A entrega e a implantação contínuas de CNF podem ser amplamente classificadas nas seguintes categorias:

- Atualizações de aplicações: a maioria das atualizações de aplicações é composta por alterações nos PODs de aplicações do Kuberbetes. Essas atualizações podem ser aplicadas automaticamente por meio do pipeline de código. A maioria das CNFs é compatível com atualizações no local, fornecendo várias instâncias de PODs de aplicações. Várias instâncias permitem uma abordagem de atualização contínua. Nem todas as alterações do POD da aplicação são compatíveis com a atualização do Helm. Os pipelines levam essas variações em consideração e usam a instalação/exclusão do Helm conforme necessário.
- Atualizações principais: as principais atualizações são essencialmente alterações no esquema do banco de dados. Não é possível aplicar essa alteração sem causar algum tempo de inatividade. A abordagem padrão para essas alterações é excluir a aplicação e recriar os pods relevantes. Durante o processo, a aplicação pode não estar disponível. As seguintes ferramentas são usadas para atualizações:

    - O AWS CloudFormation permite que os clientes descrevam e provisionem todos os recursos de infraestrutura em modelos JSON ou YAML. O AWS CloudFormation fornece um avançado mecanismo de extensão por meio de recursos personalizados apoiados pelo Lambda. Os clientes podem estender o AWS CloudFormation além dos recursos da AWS e provisionar o recurso necessário em outros ambientes; por exemplo, recursos on-premises em ambientes híbridos. O AWS CDK oferece aos desenvolvedores a capacidade de criar código usando linguagens de programação familiares de nível superior, como Python, TypeScript, JavaScript, Java e C#, e depois compilar o código em um formato JSON do AWS CloudFormation de nível inferior, que então pode ser implantado.
    - Implantação azul/verde: a AWS é compatível com implantações baseadas em canary e azul/verde em ambientes de teste e de produção. As implantações azul/verde permitem que os clientes testem uma nova versão da aplicação em um ambiente contido. Elas fornecem um método fácil e tranquilo de alternar o tráfego de produção. As implantações baseadas em canary estendem esse conceito, permitindo que o ambiente verde de não produção seja testado com uma pequena proporção do tráfego de produção para descobrir quaisquer problemas causados pelo tráfego

de produção. A nova versão da aplicação é testada em relação ao tráfego de teste simulado interno, bem como pequenas quantidades de tráfego de produção, o que proporciona confiança ao usuário antes de alternar o tráfego de produção. O tráfego de produção é aumentado gradualmente até que a transição seja concluída. A implementação envolve DNS ponderado e grupos de destino ponderados do ELB.

- A automação pode ser alcançada configurando o AWS CodePipeline com os estágios de implantação azul/verde e baseada em canary. O estágio de aprovação inicialmente pode ser conduzido de modo manual durante o provisionamento, mas depois deve ser totalmente automatizado. Nos ambientes de teste, é uma prática recomendada sempre testar com uma ação de reversão a fim de validar a compatibilidade com versões anteriores e posteriores, antes de implantar na produção. A implantação azul/verde em clusters com malha de serviços depende do suporte fornecido pela aplicação final e pelo gateway de roteamento para que a malha de serviços realize uma transição tranquila.
- O AWS Systems Manager fornece uma interface de usuário unificada para que você possa visualizar dados operacionais de vários serviços da AWS usados por funções de rede implantadas por CI/CD. O Systems Manager permite automatizar tarefas operacionais em todos os seus recursos da AWS.



Implantação canary

Segurança

A segurança é um elemento crítico. Veja a seguir uma lista de etapas de segurança que o processo de CI/CD da AWS leva em consideração ao implantar uma aplicação.

- Fonte: o repositório ECR alocado ao fornecedor é configurado com um sinalizador “Scan on Push” ativado, para que qualquer upload de imagens do Docker seja imediatamente submetido a uma verificação de segurança. Quaisquer vulnerabilidades e exposições comuns conhecidas (CVE) serão sinalizadas com notificações. Além do ECR, quando os fornecedores colocam grafos no repositório AWS CodeCommit, é solicitado que eles criptografem todas as senhas usadas com o Secrets Manager em vez de usarem texto simples.
- Integridade dos artefatos: os artefatos usados no pipeline são criptografados em repouso (usando chaves gerenciadas da AWS) ou em trânsito (usando SSL/TLS).
- Usuários e funções do IAM: as permissões fornecidas aos usuários ou recursos são baseadas no princípio de menor privilégio. Deve haver uma relação de confiança entre funções do IAM que talvez precise ser configurada se você estiver operando em recursos em serviços diferentes. Por exemplo, o AWS CodeBuild precisa de permissão para executar comandos em um cluster do Amazon EKS.
- Auditoria: o recurso de auditoria oferecido pelo AWS CloudTrailrastreia cada chamada de API em serviços e operações do usuário e permite a avaliação de eventos passados.
- Verificação de vulnerabilidade de imagem: as imagens de CNF das quais é feito upload no Amazon ECR são verificadas automaticamente em busca de vulnerabilidades de segurança. Um relatório das descobertas da verificação está disponível no AWS Management Console e também pode ser recuperado por meio da API. As descobertas podem então ser enviadas aos operadores do CSP para ação corretiva, incluindo a substituição da imagem de CNF.

As verificações de segurança ocorrem em vários estágios do pipeline para garantir que a imagem recém-carregada seja segura e atenda às verificações de conformidade desejadas para que uma notificação possa ser enviada aos CSPs para aprovação:

- O registro do contêiner verifica se há vulnerabilidades abertas do CVE.
- A configuração é verificada quanto a vazamentos de informações, padrões de informações de identificação pessoal (PII) conhecidos, durante a fase de teste, acionando regras de verificação de conformidade para problemas como portas TCP/UDP abertas inesperadas e vulnerabilidades do DOS.

- A compatibilidade com versões anteriores e posteriores é verificada quanto à segurança de atualização/reversão.

Além da aplicação, é fundamental provisionar a segurança do pipeline garantindo a transferência criptografada de artefatos entre os estágios, seja em repouso ou em trânsito.

Capacidade de observação

A AWS permite a capacidade de observação das CNFs 5G que são implantadas na AWS por padrão. Isso é habilitado pelo Amazon CloudWatch. O CloudWatch oferece visibilidade completa dos seus recursos e aplicações de nuvem.

O Amazon CloudWatch tem quatro etapas principais durante esse processo:

1. Coletar: colete métricas e logs de todos os seus recursos, aplicações e serviços da AWS executados na AWS e em servidores on-premises.
2. Monitorar: visualize aplicações e infraestrutura com painéis do CloudWatch, correlacione logs e métricas lado a lado para solucionar problemas e defina alertas com os Alarmes do CloudWatch .
3. Agir: automatize a resposta a alterações operacionais com o CloudWatch Events e AWS Auto Scaling.
4. Analisar: métricas de até um segundo, retenção de dados estendida (15 meses) e análise em tempo real com o CloudWatch Metric Math .

O agente do Amazon CloudWatch é instalado no cluster Kubernetes do cliente. O agente é compatível com os recursos de configuração, descoberta e pull de métrica do Prometheus, enriquecendo e publicando todas as métricas e metadados do Prometheus de alta fidelidade como Embedded Metric Format (EMF) no CloudWatch Logs.

O Amazon CloudWatch Container Insights automatiza a descoberta e a coleta de métricas do Prometheus a partir de aplicações conteinerizadas. Ele coleta, filtra e cria automaticamente métricas personalizadas agregadas do CloudWatch visualizadas em painéis.

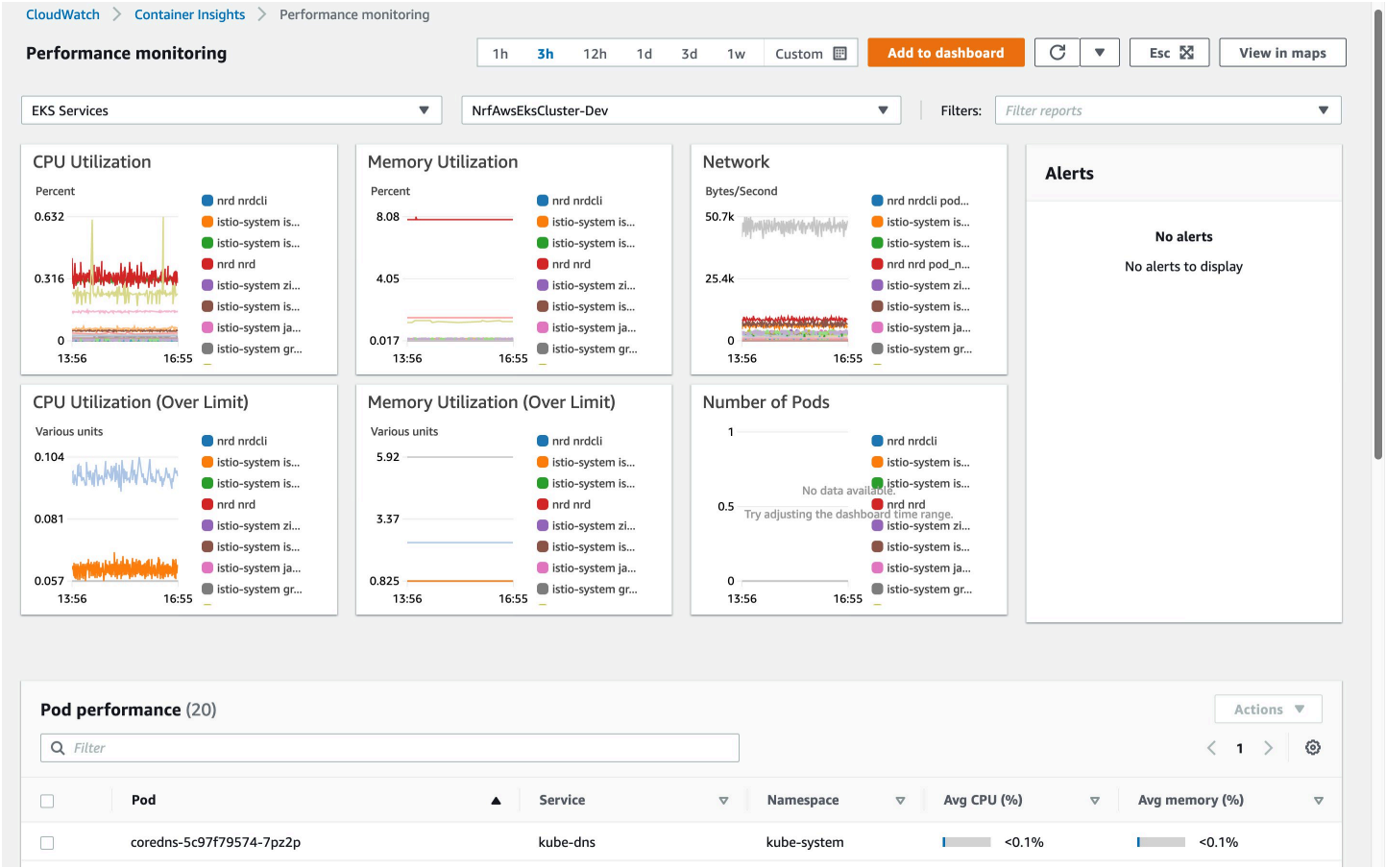
Cada evento cria pontos de dados de métrica como métricas personalizadas do CloudWatch para um conjunto de dimensões de métrica com curadoria que é totalmente configurável. A publicação de métricas agregadas do Prometheus como estatísticas de métricas personalizadas do CloudWatch reduz o número de métricas necessárias para monitorar, alertar e solucionar problemas e falhas de performance. Você também pode analisar as métricas de alta fidelidade do Prometheus usando

a linguagem de consulta do CloudWatch Logs Insights para isolar pods e rótulos específicos que afetam a integridade e a performance de seus ambientes em contêineres.

O AWS CloudTrail oferece essa visibilidade, registrando todas as chamadas de API nos serviços. O AWS Config oferece capacidade para validação de conformidade. A AWS oferece aos clientes opções adicionais de monitoramento de métricas, logs, eventos para a aplicação, infraestrutura e pipelines, usando vários serviços, como o AWS X-Ray e o AWS CloudTrail.

- A AWS pode integrar nativamente ferramentas de métrica de código aberto, como Prometheus, Fluentd, etc.
- As métricas do Prometheus podem ser ingeridas no Amazon CloudWatch ou no OpenSearch Service para análise posterior.
- A AWS usa o fluentD como um mecanismo padrão para coletar logs de vários sistemas. Esse mesmo mecanismo é usado e configurado para este projeto.

Para obter detalhes sobre como configurar esse mecanismo, consulte Configurar o FluentD como um DaemonSet para enviar logs para o CloudWatch Logs.



Exemplo de métricas monitoradas do Amazon CloudWatch

Orquestração de CI/CD com ferramentas de terceiros e de código aberto

A camada de orquestração usa o IaC para implantar e configurar a infraestrutura subjacente necessária para executar as funções de rede 5G. Essa camada deve ser projetada para ser modular, portátil e reutilizável.

A infraestrutura segue as práticas recomendadas nativas da nuvem, sendo altamente disponível, redundante e escalável.

Conforme demonstrado nas seções anteriores, a implantação da infraestrutura subjacente pode ser obtida com uso do AWS Cloud Development Kit. Isso pode ser feito com o uso do Terraform da Hashicorp.

Terraform

O Terraform é uma ferramenta de software IaC de código aberto que fornece um fluxo de trabalho consistente de interface de linha de comando (CLI) para gerenciar centenas de serviços de nuvem. O Terraform codifica APIs de nuvem em arquivos de configuração declarativos.

Para implantação com o Terraform, use os mesmos princípios usados no CDK. O código é estruturado em módulos que permitem que os componentes de rede sejam personalizados e reutilizados de acordo com os requisitos do fornecedor.

A configuração é toda parametrizada, o que permite que as implantações sejam totalmente adaptadas de acordo com as recomendações de provedores e ISV.

A implantação das funções de rede é separada em duas fases:

- A infraestrutura necessária da AWS é criada e gerenciada por meio de um repositório central.
- A configuração e o código são armazenados centralmente em um repositório do GitHub.

Depois que os pré-requisitos são criados, a função de rede está pronta para ser implantada usando um pipeline de aplicação que foi definido no estágio anterior.

Implantação de infraestrutura

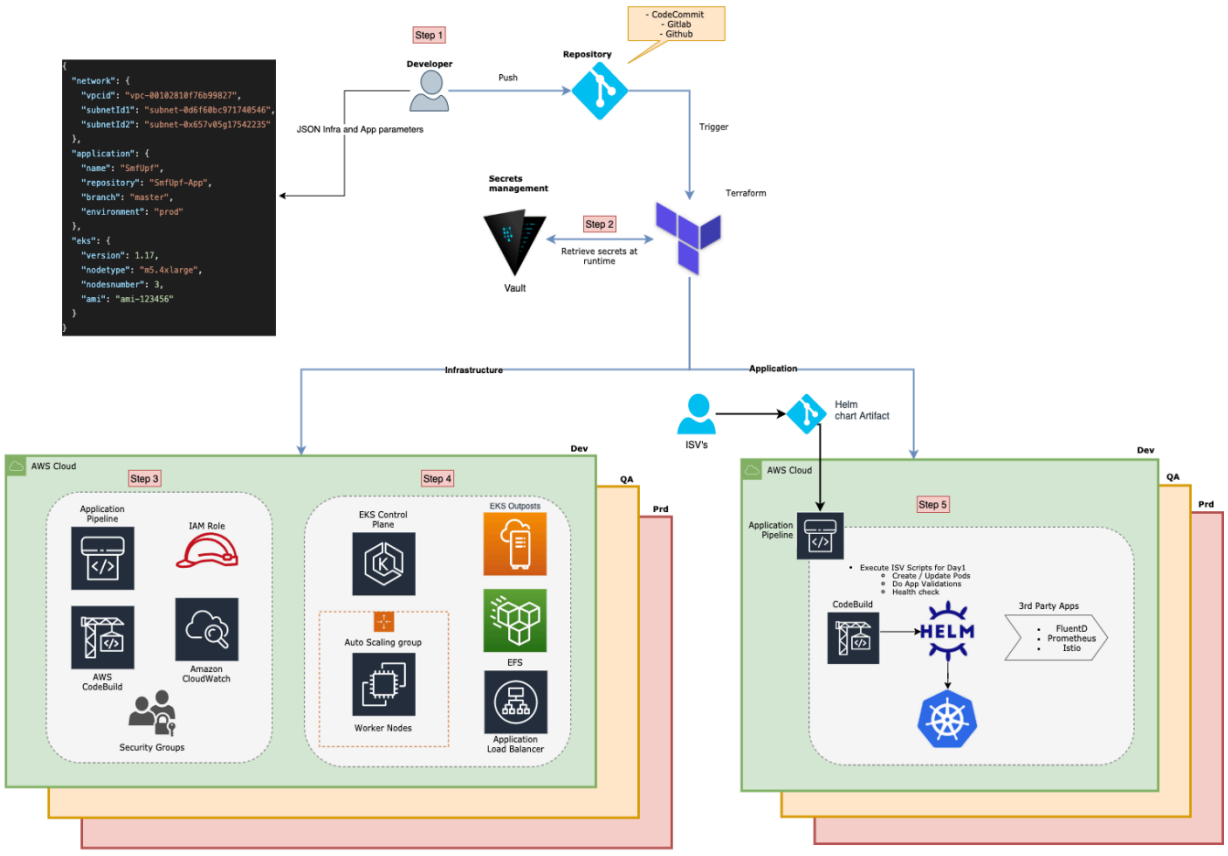
A implantação de infraestrutura inclui todos os pré-requisitos para que a função de rede seja implantada e configurada com êxito.

Alguns dos componentes criados como parte dessa fase são:

- Rede: VPC, sub-redes públicas e privadas, rotas, balanceadores de carga
- Computação: Kubernetes (Vmware Tanzu, Amazon EKS ou AWS Outposts), instâncias do Amazon EC2, nós primários e de trabalho, grupo do Auto Scaling
- Armazenamento: Amazon EFS, Amazon EBS, bucket do Amazon S3
- Segurança: funções do IAM, grupos de segurança
- Pipeline: CodePipeline, CodeBuild
- Capacidade de observação: CloudWatch, Prometheus, FluentD

Veja a sequência de infraestrutura orquestrada pelo Terraform e explicada na figura a seguir:

1. Um desenvolvedor preenche um arquivo JSON armazenado em um repositório central com o código IaC. O arquivo contém informações sobre a configuração de infraestrutura desejada, como tamanho das instâncias, versão do Kubernetes, informações de rede e detalhes do repositório da aplicação.
2. Recupera segredos do HashiCorp Vault ou do AWS Secrets Manager em tempo de execução.
3. Implanta e configura os componentes da infraestrutura (redes, computação, armazenamento e segurança).
4. Um cluster do Amazon EKS com nós de processamento que hospedam os pods de função de rede é implantado. O Amazon EKS também pode ser implantado no AWS Outposts para oferecer suporte a workloads que exigem proximidade com um datacenter.
5. Um pipeline de aplicação é criado e configurado para escutar alterações no repositório de funções de rede. Sempre que o código é enviado para a ramificação do repositório configurada, o pipeline aciona automaticamente a construção, o teste e a implantação da função de rede.
6. Ferramentas de capacidade de observação que coletam e centralizam logs e métricas são implantadas como serviços em todos os nós e fornecem dados quase em tempo real que podem ser visualizados no Grafana ou no OpenSearch Dashboards



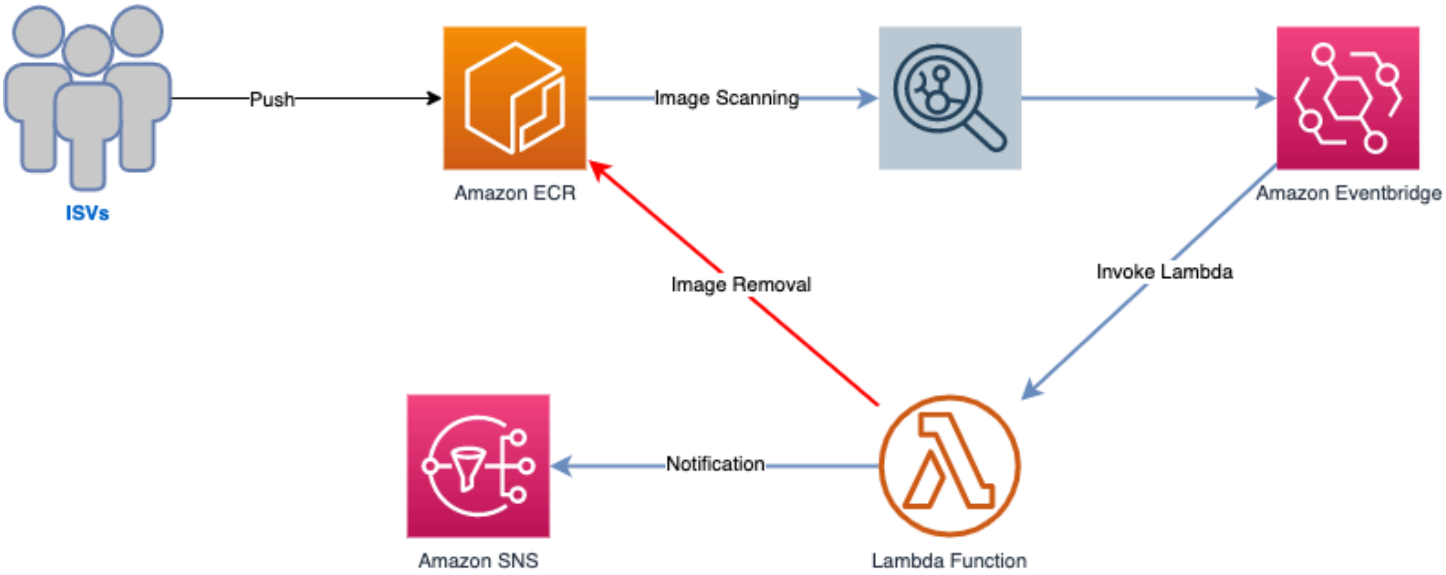
Implantação e configuração da função de rede

Implantação e configuração da função de rede

O pipeline criado no estágio anterior permite que ISVs e provedores descentralizem e otimizem a implantação de funções de rede. O pipeline é conectado e escuta as alterações no repositório da aplicação, que é configurado no arquivo JSON na etapa 1 da figura anterior.

Para examinar as imagens publicadas por terceiros, uma solução de verificação de vulnerabilidades que auxilia na identificação de vulnerabilidades de software em imagens de contêiner é implantada e configurada. A solução de verificação examina automaticamente todas as novas imagens enviadas para o Amazon ECR. Para obter mais informações sobre a verificação de imagens ECR, consulte Verificação de imagens.

A figura a seguir demonstra a arquitetura da solução de verificação de vulnerabilidades de imagem.



Arquitetura da solução de verificação de vulnerabilidades de imagem

O pipeline da aplicação pode ser configurado para ser acionado por alterações na imagem após o resultado da verificação ou por alterações diretas no repositório. Por exemplo, quando uma nova imagem do Helm é criada.

A lista a seguir é a sequência que cria/atualiza a função de rede, conforme representado nesta figura:

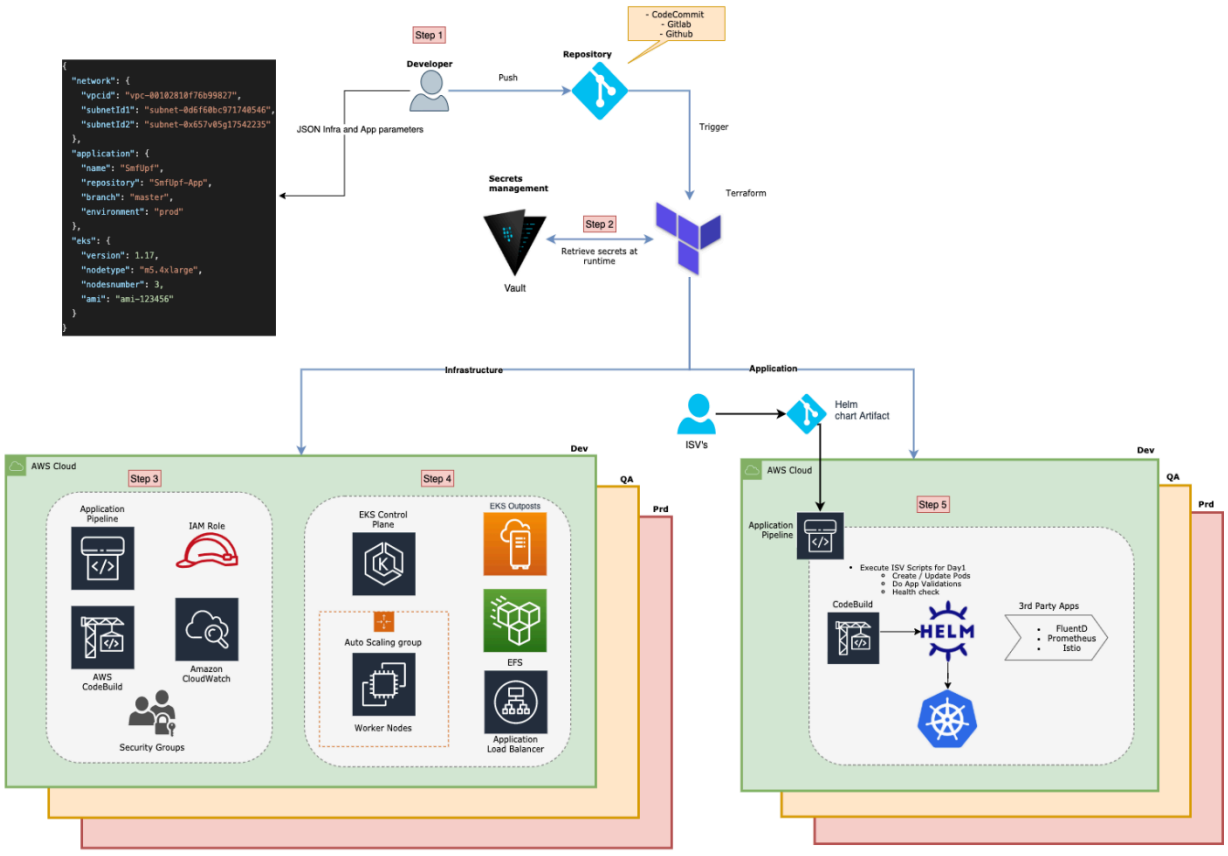
Os ISVs publicam novas imagens no Amazon ECR. (Se a imagem for aprovada, o pipeline da aplicação será acionada.

O CodePipeline extrai a nova imagem do Amazon ECR e usa o CodeBuild para implantar a imagem no Kubernetes. Os comandos do Helm podem ser usados para atualizar a função de rede.

Depois que a imagem é implantada, o Teste como serviço (TaS) é acionado. O TaS valida a nova implantação e centraliza dados e métricas sobre o desempenho das funções de rede sob estresse.

Logs e métricas são coletados e centralizados no OpenSearch e no Grafana. Terceiros, como Datadog, Istio e Prometheus, também podem ser configurados para fornecer capacidade de observação adicional.

Um MANO que pode coordenar recursos de rede também pode ser implantado e integrado à solução. Ele usa os dados coletados para realizar ações automatizadas, como divisão de rede e autoescalabilidade de Qualidade de Serviço (QoS).



Pipeline de aplicação

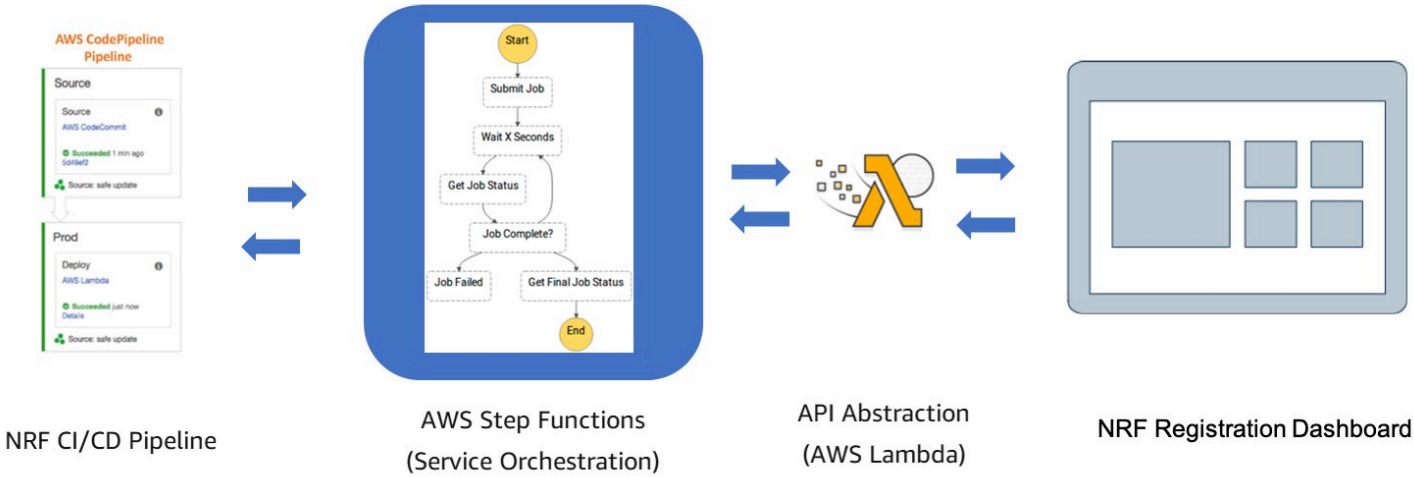
Teste

A framework de automação de teste específica de telecomunicações pode ser integrada ao pipeline de código. O pipeline de código é integrado ao Step Functions para orquestrar a integração com uma framework de automação de teste. O AWS Step Functions usa várias funções do Lambda para invocar a framework de automação de teste (ferramentas de terceiros) por meio de chamadas de API. A função de etapa obterá inicialmente o ID do teste, executará o teste e obterá os resultados da framework de automação de teste. Depois, a função de etapa analisa os resultados e os passa para o pipeline de código para aprovação da aplicação para implantação de produção. A aprovação pode ser automatizada ou mantida manualmente no pipeline de código, conforme necessário. Essa é uma etapa importante para os CSPs promoverem a implantação do ambiente de teste para a produção. As APIs de alto nível necessárias para a integração são categorizadas da seguinte maneira:

- Obter contexto
- Executar um caso de teste específico

- Interromper o caso de teste
- Obter resultados

A complexidade da chamada de APIs REST externas é modelada usando o AWS Step Functions, que permitem que construções padrão invoquem fluxos paralelos, aguardem resultados, ramifiquem com base em condições e integrem a API REST com o AWS CodePipeline.



Fluxo de teste

CI/CD e orquestração

A integração e a entrega contínuas fazem parte de uma filosofia de automação geral que acompanha a arquitetura nativa da nuvem e como ela se aplica ao 5G. A orquestração é outro aspecto dessa filosofia que deve ser dinâmico e reativo a quaisquer mudanças que ocorram na rede. A orquestração e a CI/CD devem ser estreitamente acopladas para garantir um serviço íntegro e minimizar a interrupção do serviço. A integração entre CI/CD e orquestração devem ocorrer em duas frentes:

- A aplicação de patches e as atualizações no sistema precisam ser gerenciadas e orquestradas de forma a minimizar a interrupção dos serviços ativos. Por exemplo, a orquestração pode determinar dinamicamente o melhor momento em que uma atualização deve ser lançada.
- A orquestração com reconhecimento de CI/CD permite que o tráfego seja deslocado durante a implantação de atualizações com base na estratégia do modelo de implantação adotada (canary, linear ou tudo de uma vez).

Normalmente, as soluções de orquestração são executadas sobre pipelines de CI/CD para permitir que a orquestração introduza fases de governança nesses pipelines e tenha exposição a ciclos de atualização contínuos.

Conclusão

A CI/CD fornece um caminho claro e eficiente para desenvolvedores e equipes de aplicações implantarem novos códigos de aplicações em questão de minutos. A AWS tem uma variedade de ferramentas que podem ajudar os desenvolvedores na integração, no teste e na implantação de novos códigos, incluindo AWS CodePipeline, AWS CodeCommit, AWS CodeBuild, AWS CodeDeploy e muitos outros. Este documento analisou como os serviços da AWS podem ser usados para criar um processo de CI/CD para implantar a função de rede 5G de forma totalmente automatizada, incluindo as diferentes etapas necessárias para concluir a implantação do novo código. Ele também analisou como integrar uma framework de automação de teste de terceiros ao processo de CI/CD, bem como usar ferramentas de terceiros, como o Terraform.

Colaboradores

Os colaboradores desse documento incluem:

- Hisham Elshaer, consultor sênior, AWS Telecom, Amazon Web Services
- Vara Prasad Talari, consultora principal, AWS Telecom, Amazon Web Services
- Rabi Abdel, consultor principal, AWS Telecom, Amazon Web Services
- Franco Bontorin, consultor sênior de entrega compartilhada, Amazon Web Services
- Pragtideep Singh, consultor de entrega compartilhada, Amazon Web Services
- Subbarao Duggisetty, arquiteto de infraestrutura de nuvem, contas globais, Amazon Web Services
- Young Jung, arquiteto sênior de soluções de parceiros, AWS Telecom, Amazon Web Services

Revisões do documento

Para ser notificado sobre atualizações deste whitepaper, inscreva-se no RSS feed.

update-history-change	update-history-description	update-history-date
Publicação inicial	Primeira publicação do whitepaper	8 de março de 2021

Outras fontes de leitura

Para obter informações adicionais, consulte:

- Prática de integração e entrega contínuas na AWS (whitepaper)
- Rede de núcleo de pacotes móveis de nível de operadora na AWS (whitepaper)
- 5G Network Evolution with AWS (whitepaper)

Siglas

- AMF: função de gerenciamento de acesso e mobilidade
- API: interface do programa da aplicação
- AUSF: função de servidor de autenticação
- BSS: sistema de suporte empresarial
- CDK: Cloud Development Kit
- CI/CD: integração e entrega contínuas
- CLI: Command Line Interface
- CNF: função de rede nativa da nuvem ou conteinerizada
- CSP: provedor de serviços de comunicação
- CU: unidade central RAN
- CVE: vulnerabilidades e exposições comuns
- DoS: negação de serviço
- DR: recuperação de desastres
- DU: unidade distribuída RAN
- E2E: ponta a ponta
- ECR: Elastic Container Registry
- EFS: Elastic File System
- EKS: Elastic Kubernetes Service
- EPC: núcleo de pacotes evoluído
- IaC: infraestrutura como código
- ISV: provedores independentes de software
- MANO: gerenciamento e orquestração
- MEC: computação de borda de acesso múltiplo
- NACL: lista de controle de acesso da rede
- NAT: conversão de endereço de rede
- NF: função de rede
- NFV: virtualização de funções de rede
- NFVO: orquestrador de virtualização de funções de rede

- NOC: centro de operações de rede
- NRF: função de repositório de rede
- OSS: sistema de suporte de operações
- PII: informações de identificação pessoal
- QoS: qualidade de serviço
- RAN: rede de acesso por rádio
- SBI: interface baseada em serviço
- SMF: função de gerenciamento de sessão
- SSL: Secure Sockets Layer
- TaS: Teste como serviço
- TCP: Transmission Control Protocol
- TLS: Transport Layer Security
- UDM: gerenciamento unificado de dados
- UDP — User Datagram Protocol
- UPF: função de plano de usuário
- VIM: gerente de infraestrutura virtualizada
- VNF: função de rede virtual
- VPC: nuvem privada virtual

Avisos

Os clientes são responsáveis por fazer sua própria avaliação independente das informações neste documento. Este documento é: (a) fornecido apenas para fins informativos, (b) representa as ofertas e práticas de produtos atuais da AWS, que estão sujeitas a alterações sem aviso prévio e (c) não cria nenhum compromisso ou garantia da AWS e suas afiliadas, fornecedores ou licenciadores. Os produtos ou serviços da AWS são fornecidos no “estado em que se encontram”, sem garantias, declarações ou condições de qualquer tipo, explícitas ou implícitas. As responsabilidades e obrigações da AWS com seus clientes são regidas por contratos da AWS, e este documento não modifica nem faz parte de nenhum contrato entre a AWS e seus clientes.