



Documento técnico de AWS

# Integración y entrega continuas para redes 5G en AWS



# Integración y entrega continuas para redes 5G en AWS: Documento técnico de AWS

Copyright © 2026 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Las marcas comerciales y la imagen comercial de Amazon no se pueden utilizar en relación con ningún producto o servicio que no sea de Amazon, de ninguna manera que pueda causar confusión entre los clientes y que menosprecie o desacredite a Amazon. Todas las demás marcas registradas que no son propiedad de Amazon son propiedad de sus respectivos propietarios, que pueden o no estar afiliados, conectados o patrocinados por Amazon.

# Table of Contents

Resumen .....	i
Resumen .....	1
Introducción .....	2
Integración y entrega continuas .....	4
Integración continua .....	4
Entrega e implementación continuas .....	4
Infraestructura como código .....	4
CI/CD activado AWS .....	5
Redes 5G activadas AWS .....	9
CI/CD en redes 5G .....	9
Pasos detallados de CI/CD .....	11
Configuración de la red .....	12
Despliegue de infraestructura .....	12
Despliegue de funciones de red nativas de la nube .....	13
Entrega continua de CNF .....	14
Seguridad .....	17
Observabilidad .....	18
Orquestación de CI/CD con herramientas de terceros y de código abierto .....	21
Terraform .....	21
Despliegue de infraestructura .....	22
Despliegue y configuración de funciones de red .....	23
Testeo .....	25
CI/CD y orquestación .....	27
Conclusión .....	28
Colaboradores .....	29
Revisiones del documento .....	30
Documentación adicional .....	31
Acrónimos .....	32
Avisos .....	34
.....	xxxv

# Integración y entrega continuas para redes 5G en AWS

Fecha de publicación: 8 de marzo de 2021 () [Revisiones del documento](#)

## Resumen

Este documento técnico presenta la integración y la entrega continuas (CI/CD) for 5G networks, and how Amazon Web Services (AWS) tools and services can be used to fully automate the deployment and upgrades of 5G network functions. The whitepaper provides detailed description of the different stages of CI/CD para las funciones de la red 5G), incluida la configuración de la red, el despliegue de la infraestructura, el despliegue de las funciones de red nativas de la nube y las actualizaciones continuas de las funciones de la red. También proporciona detalles sobre la integración con herramientas de código abierto y de terceros para las pruebas, la observabilidad y la organización.

Este documento técnico está dirigido a los proveedores de servicios de comunicación (CSPs) y a los proveedores de software independientes (). ISVs

# Introducción

Históricamente, el desarrollo, las pruebas de integración en laboratorio y sobre el terreno y el despliegue en producción de nuevos nodos de red o nuevas funciones en una red celular llevaban semanas o incluso meses para garantizar la estabilidad de los servicios de telecomunicaciones (telecomunicaciones) críticos para la misión y la empresa. El largo ciclo de despliegue se debió a la arquitectura monolítica de los nodos de red tradicionales, a un entorno de varios proveedores y a point-to-point las numerosas interfaces entre las entidades de red de las redes móviles 2G, 3G y 4G.

Como se presentó en el documento técnico [La evolución de la red 5G con AWS](#), las redes móviles 5G, estandarizadas por el 3GPP, ahora admiten una arquitectura nativa de la nube habilitada por la virtualización y la contenedorización. Más específicamente, las redes 5G introducen y respaldan un nuevo paradigma de arquitectura de microservicios, sin estado y basada en servicios.

Esta arquitectura 5G significa que las diferentes funciones de la red pueden funcionar como servicios independientes acoplados de forma flexible que se comunican entre sí a través de interfaces bien definidas y APIs. Y lo que es más importante, cada función de la red se puede actualizar de forma independiente. Este cambio de arquitectura en el 5G permite a los CSPs lograr una mayor agilidad y eficiencia operativa al facilitar la implementación de actualizaciones para las funciones de la red con mayor frecuencia y, al mismo tiempo, mantener las pruebas, los requisitos de seguridad y los estándares mediante la automatización.

La integración y el despliegue de nuevas funciones para un CSP generalmente comienzan cuando el proveedor de funciones de red lanza un nuevo paquete de software para funciones de red, como una imagen de [Docker](#) en una función de red basada en contenedores, o un nuevo archivo de configuración, como un diagrama de [Helm](#) en el caso de la aplicación [Kubernetes](#). (Un gráfico de Helm es una colección de archivos que describen un conjunto relacionado de recursos de Kubernetes).

La idea de utilizar el paradigma del despliegue CI/CD de funciones de red 5G está ganando terreno, pero la materialización práctica de esta idea ha supuesto un desafío en la industria de las telecomunicaciones.

AWS ha sido pionera en el desarrollo de nuevas CI/CD herramientas para la entrega de software a fin de ayudar a un amplio espectro de industrias a desarrollar e implementar cambios de software con rapidez, manteniendo al mismo tiempo la estabilidad y la seguridad de los sistemas. Estas herramientas incluyen un conjunto de servicios de desarrollo y operaciones de software (DevOps) [AWS CodeStar](#), como [CodeCommit](#), [CodePipeline](#), [CodeBuild](#), y [CodeDeploy](#).

[AWS también promueve la idea de la infraestructura como código \(IaC\) mediante el Kit de Desarrollo en la Nube \(AWS CDK\) de AWS, AWS y herramientas de terceros basadas en API CloudFormation, como Terraform.](#) Con estas herramientas, AWS puede almacenar los procesos de implementación de la función de red AWS como código fuente y mantener este código fuente de IaC en proceso para ofrecer una entrega continua. CI/CD

Este documento técnico describe los procesos detallados para aprovechar el AWS IaC y CI/CD las herramientas para el despliegue y la actualización de la función de red 5G. Además, este documento técnico cubre la integración con herramientas de terceros para las pruebas, la observabilidad y la organización.

AWS CI/CD tools are not restricted to 5G network functions. They are also employed for automating the deployment of 4G networks, which enables CSPs to rapidly and efficiently deploy and update 4G network functions. Most 4G network functions are Virtual Network Function (VNF) based. AWS CI/CD Estos conjuntos de herramientas se AWS CloudFormation pueden utilizar para automatizar el despliegue de la red 4G VNFs, lo que aporta escalabilidad y eficiencia en el tiempo a los despliegues de redes 4G.

# Integración y entrega continuas

## Integración continua

La integración continua (CI) es un proceso de software en el que los desarrolladores envían regularmente su código a un repositorio central, como [AWS CodeCommit](#) o [GitHub](#). Cada inserción de código desencadena una compilación automática, seguida de la ejecución de pruebas. El objetivo principal de la CI es descubrir los problemas de código en una fase temprana, mejorar la calidad del código y reducir el tiempo necesario para validar y publicar nuevas actualizaciones de software.

## Entrega e implementación continuas

La entrega continua (CD) es un proceso de software en el que los artefactos se despliegan en el entorno de prueba, el entorno de ensayo y el entorno de producción. La entrega continua puede estar completamente automatizada o tener etapas de aprobación en puntos críticos. Esto garantiza que se obtengan todas las aprobaciones necesarias antes de la implementación, como la aprobación de la administración de versiones. Cuando la entrega continua se implementa correctamente, los desarrolladores siempre disponen de un artefacto de construcción listo para la implementación que ha pasado por un proceso de pruebas estandarizado.

Con la implementación continua, las revisiones se implementan automáticamente en un entorno de producción sin la aprobación explícita de un desarrollador, lo que automatiza todo el proceso de publicación del software. Esto permite disponer de un ciclo continuo de comentarios de los clientes en las primeras etapas del ciclo de vida del producto.

Con el despliegue continuo, todos los cambios que se realicen y superen las pruebas automatizadas se ponen en producción automáticamente. El objetivo de la entrega continua no es publicar todos los cambios introducidos y pasar inmediatamente las pruebas automatizadas a producción, sino garantizar que todos los cambios estén listos para su puesta en producción.

## Infraestructura como código

Como se detalla en el documento técnico sobre la [evolución de la red 5G con AWS](#), la IaC es un factor clave para automatizar el proceso de aprovisionamiento y la gestión del ciclo de vida tanto de la aplicación como de su entorno. En lugar de confiar en los pasos que se realizan manualmente, tanto los network/IT administradores como los desarrolladores pueden crear instancias de la infraestructura mediante archivos de configuración. IaC trata estos archivos de configuración como

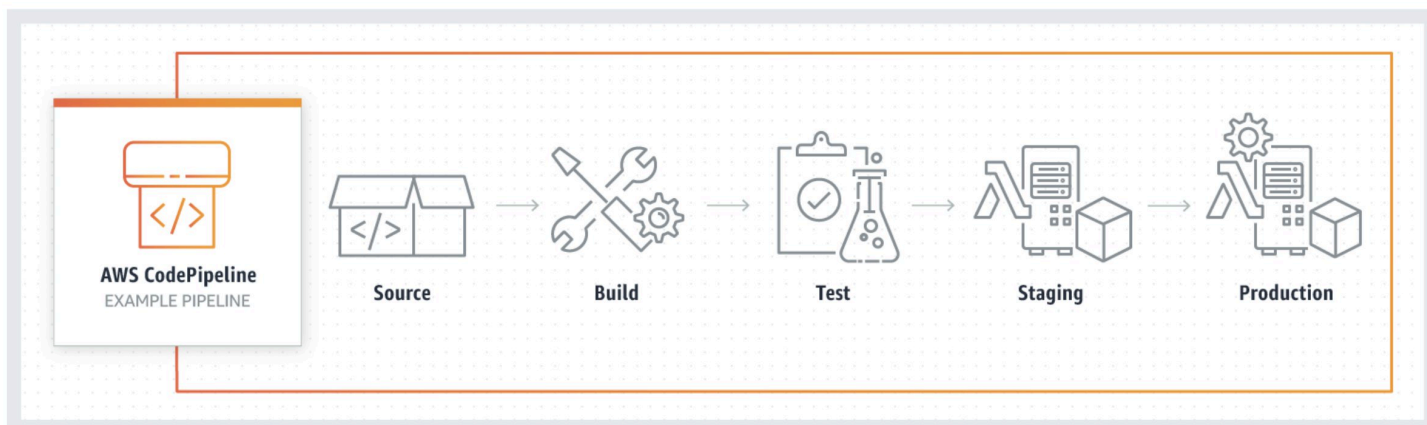
código de software. Estos archivos se pueden usar para producir un conjunto de artefactos: a saber, los servicios de cómputo, almacenamiento, red y aplicaciones que componen un entorno operativo. La IaC elimina las desviaciones de configuración provocadas por la automatización, lo que aumenta la velocidad y la agilidad de las implementaciones de infraestructura.

En el caso de la implementación de la virtualización de funciones de red (NFV) AWS, este marco IaC aporta un valor desde la perspectiva de la organización. Desde la creación de la Nube Privada Virtual (VPC) hasta el despliegue de las funciones de red, cada paso se puede programar, gestionar como código fuente y mantener con el control de versiones incorporado. [AWS CodeCommit](#)

Este marco de IaC para el funcionamiento de la red permite crear y desplegar infraestructuras y funciones de red de forma repetible y fiable, lo que puede ampliarse a la automatización end-to-end (E2E) de la gestión de los segmentos de red y la gestión del ciclo de vida de los servicios. AWS proporciona un conjunto completo de herramientas para crear, mantener e implementar la infraestructura de forma programática, descriptiva y declarativa, utilizando servicios como, AWS CDK para Kubernetes, y la exposición a las CloudFormation API AWS CDK de todos los servicios. AWS

## CI/CD activado AWS

El CI/CD puede imaginarse como un proceso en el que el código nuevo se envía por un extremo, se prueba en una serie de etapas (fuente, compilación, prueba, puesta en escena y producción) y, a continuación, se publica como código listo para la producción.



### Descripción general de la canalización de CI/CD

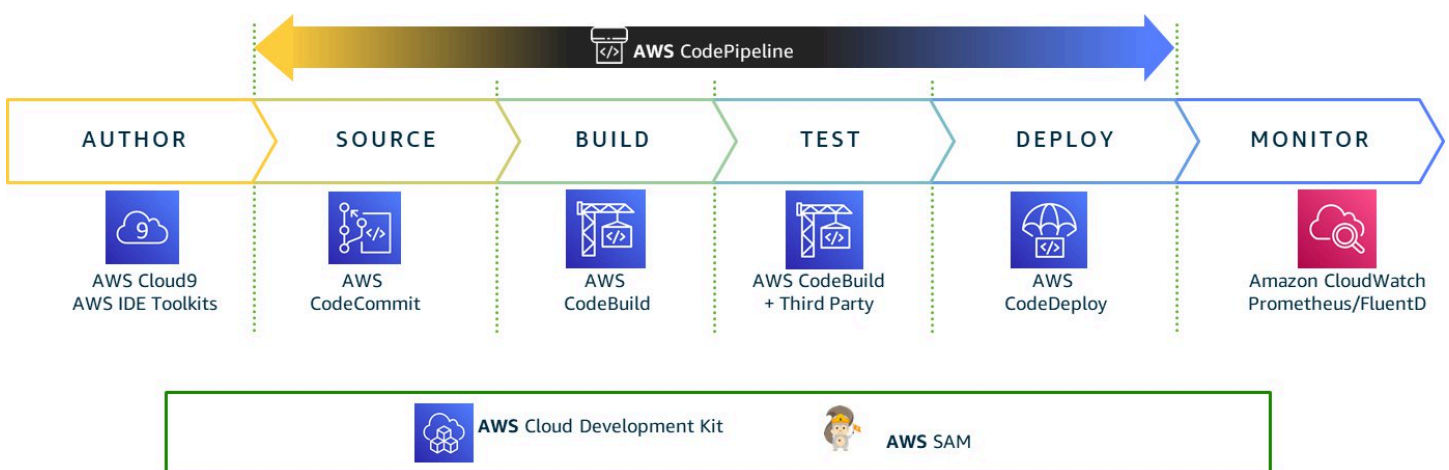
Cada etapa de la CI/CD canalización está estructurada como una unidad lógica en el proceso de entrega. Cada etapa actúa como una puerta que controla un aspecto determinado del código. A medida que el código avanza en el proceso, se supone que la calidad del código es mayor en las etapas posteriores, ya que se siguen verificando más aspectos del mismo. Los problemas que se

descubren en una fase temprana impiden que el código avance en el proceso. Los resultados de las pruebas se envían inmediatamente al equipo y, si el software no pasa la fase de creación y publicación, se detienen todas las versiones y versiones posteriores.

AWS incluye un conjunto completo de herramientas de CI/CD desarrollo para acelerar los ciclos de desarrollo y lanzamiento del software. [AWS CodePipeline](#) automatiza las fases de creación, prueba e implementación del proceso de publicación cada vez que se produce un cambio de código, en función del modelo de publicación definido. Esto permite la entrega rápida y confiable de funciones y actualizaciones.

Las canalizaciones de código se pueden integrar con otros servicios. Pueden ser AWS servicios, como [Amazon Simple Storage Service](#) (Amazon S3), o productos de terceros, como GitHub. AWS CodePipeline puede abordar una variedad de casos de uso de desarrollo y operación, que incluyen:

- Compilar, compilar y probar código con [AWS CodeBuild](#)
- Entrega continua de aplicaciones basadas en contenedores a la nube
- Se requiere la validación previa al despliegue de los artefactos (como descriptores e imágenes de contenedores) para el servicio de red o para funciones de red específicas nativas de la nube
- Pruebas funcionales, de integración y de rendimiento para el funcionamiento de la red de función/ virtual red en contenedores (CNF/VNF), incluidas las pruebas de referencia y de regresión
- Pruebas de confiabilidad y recuperación ante desastres (DR).



## AWS Componentes de la tubería CICD

AWS puede configurar CI/CD canalizaciones mediante las siguientes herramientas de AWS desarrollador:

- [AWS CodeCommit](#)
- [AWS CodeBuild](#)
- [AWS CodePipeline](#)
- [AWS CodeDeploy](#)
- [Amazon Elastic Container Registry](#)
- [AWS CodeStar](#)

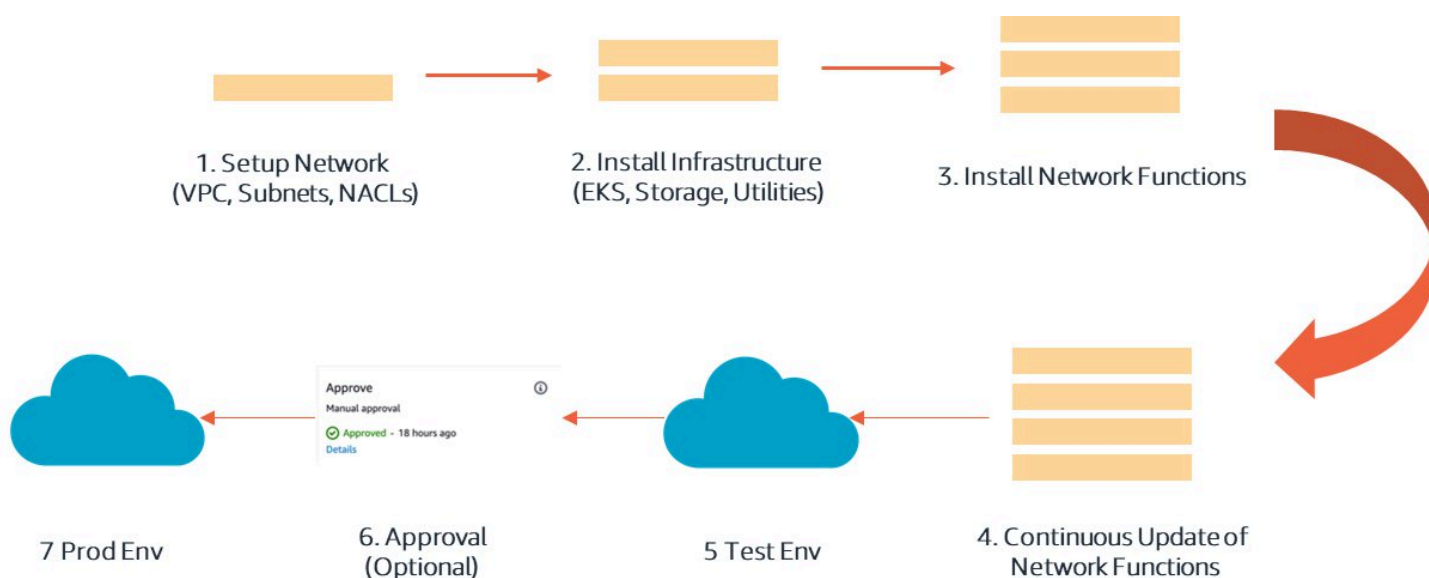
La creación de canalizaciones de CI/CD se puede automatizar mediante y. [AWS CDKCloudFormation](#) En el ámbito de la NFV, esta automatización AWS nativa se puede integrar en un marco de gestión y orquestación (MANO) y en el marco de orquestación de servicios del CSP.

El CI/CD proceso incluye los siguientes pasos:

- Configuración de la red AWS CDK e CloudFormation iniciar la creación de los requisitos previos de la red:
  - Pila de redes (VPC, subredes, puerta de enlace de traducción de direcciones de red (NAT), tabla de enrutamiento y puerta de enlace de Internet)
- Despliegue de la infraestructura AWS CDK e CloudFormation inicie la creación de las siguientes pilas de recursos:
  - Pila de cómputo (creación de clústeres de [Amazon Elastic Kubernetes Service](#) (Amazon EKS), nodos EKS Worker) [AWS Lambda](#)
  - Pila de almacenamiento (depósitos de Amazon S3, volúmenes de [Amazon Elastic Block Store](#) (Amazon EBS) y [Amazon Elastic File System](#) (Amazon EFS))
  - Pila de monitorización ( ) [CloudWatch](#), [Amazon OpenSearch Service](#) (OpenSearch Servicio)
  - Pila de seguridad [AWS Identity and Access Management](#)(AWS IAM), grupos de seguridad de [Amazon Elastic Compute Cloud](#) (Amazon EC2), listas de [control de acceso a la red de VPC](#) ( ) NACLs
- Implementación de la función de red en la nube (CNF): en esta etapa, la CNF se implementa en los clústeres de EKS mediante las herramientas de gráficos de [KubectI](#) y Helm. [Esta etapa también](#)

implementa cualquier aplicación o herramienta específica que necesite CNFs para trabajar de manera eficiente (como Prometheus o Fluentd). CNFs se puede implementar mediante funciones Lambda o con. AWS CodeBuild

- Actualizaciones e implementación continuas: se trata de una secuencia de pasos que se llevan a cabo de forma iterativa para implementar los cambios que forman parte de container/configuration los cambios que dan lugar a las actualizaciones. Al igual que en el caso de implementación de CNF, las actualizaciones y el despliegue continuos se pueden automatizar mediante AWS los servicios, con el activador de AWS CodeCommit Amazon Elastic Container Registry (Amazon ECR) o de un sistema fuente de terceros, como GitLab Webhooks.



### Diagrama de flujo de canalización de AWS CICD

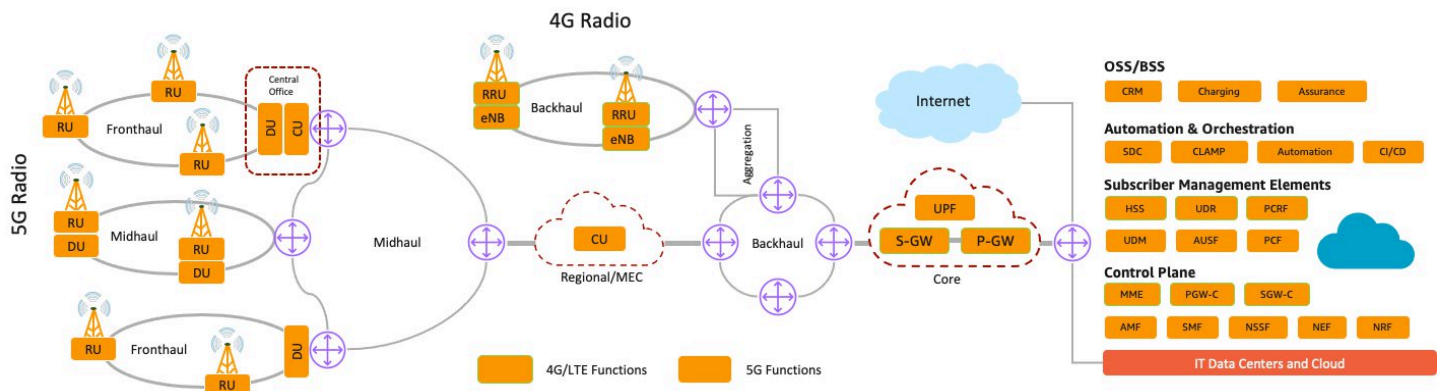
La CI/CD canalización se crea utilizando [AWS CodePipeline](#) utiliza un servicio de entrega continua que modela, visualiza y automatiza los pasos necesarios para lanzar el software. Al definir las etapas de una canalización, puedes recuperar el código de un repositorio de código fuente, convertirlo en un artefacto publicable, probar el artefacto e implementarlo en producción. Solo se implementará el código que supere satisfactoriamente todas estas etapas. Si lo desea, puede añadir otros requisitos a su proceso, como las aprobaciones manuales, para garantizar que solo los cambios aprobados se implementen en producción.

## Redes 5G activadas AWS

El modelo típico de infraestructura de red 5G se compone de un sitio de radio 4G/5G, un centro de fronthaul/midhaul/backhaul network, a core network site, and a telecom/IT datos. CSPs puede usar AWS los Servicios para crear una infraestructura de red 5G flexible y escalable y, al mismo tiempo, reducir el costo de inversión inicial. AWS se puede utilizar para implementar el Centro de Operaciones de Red (NOC) virtual en la región que aloja el Operations System/Business Support System (OSS/BSS) y la mayoría de las funciones principales de la red del plano de control.

AWS también se puede aprovechar para implementar la oficina central (CO) local o el centro de datos distribuido con una flota de instancias de [AWS Outposts](#) que alojan principalmente funciones del plano de usuario, como la UPF (función de plano de usuario), la unidad central RAN (CU) y la computación perimetral de acceso múltiple (MEC). En el documento técnico sobre la [evolución de la red 5G en AWS](#) se explica con más detalle la arquitectura de referencia y las ventajas de la [implementación de la red 5G en ella](#).

Cuando llegue el momento de implementar la red 5G, las herramientas de AWS CI/CD AWS, que se presentan en las siguientes secciones de este documento técnico, pueden facilitar la automatización total del despliegue, la actualización y la gestión del ciclo de vida de las funciones de la red 5G.

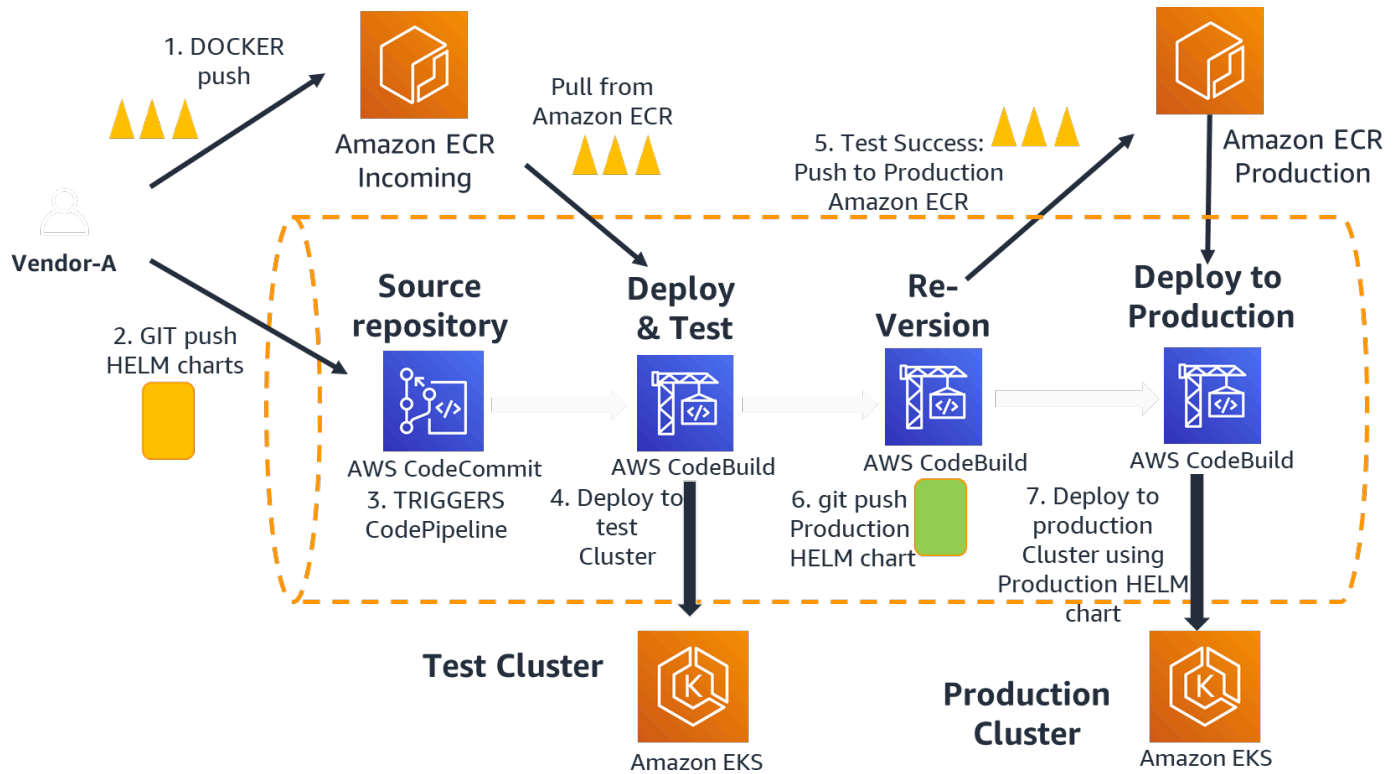


### Arquitectura E2E de la red 5G

## CI/CD en redes 5G

La construcción del diseño de la infraestructura se almacena en forma de código utilizando un lenguaje declarativo. Esto permite al CSP tener una reproducción repetible de la infraestructura con el mismo comportamiento esperado según sea necesario. El código se mantiene en el repositorio de códigos y se configura una canalización para organizar las actualizaciones de las pilas implementadas (por ejemplo, y). AWS CDK CloudFormation AWS puede ayudar a crear la

infraestructura como código (IaC) para la incorporación ágil de las funciones de los proveedores de software independientes (ISV).



### Flujo de canalización de código

Los cambios en las configuraciones de las funciones de red nativas de la nube a través de los gráficos de Helm se consideran desencadenantes de una ejecución automática de las funciones de red por CI/CD canalización.

AWS CodeCommit se puede utilizar para mantener los archivos de configuración y Amazon ECR se puede utilizar para conservar las imágenes de los contenedores.

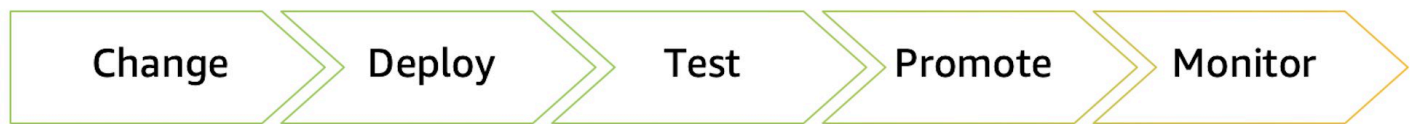
Como se muestra en la figura del flujo de código, cuando el ISV introduce nuevos cambios de código en el repositorio de código (un diagrama de Helm, archivos de configuración o un archivo de propiedades), se activa la canalización de código. La canalización del código extrae la imagen del ECR y utiliza el diagrama de Helm para implementar la aplicación. Las nuevas pruebas de aplicaciones se pueden integrar con el marco de automatización de pruebas de terceros. En función del resultado, CSPs puede aprobar el despliegue en producción.

La etapa CodePipeline de origen busca cambios en los archivos de configuración. Los proveedores válidos para la fase CodeCommit de origen son Amazon S3 o CloudFormation. GitHub Se pueden integrar sistemas fuente alternativos mediante el uso de funciones Lambda para implementar

Webhooks, lo que permite la integración basada en eventos entre Gitlab y. AWS CodePipeline  
Consulta los siguientes enlaces para obtener una guía de implementación detallada.

- [Webhooks con GitLab](#)
- [Integraciones de registro de contenedores](#)

El diseño de la canalización de CI/CD debe tener en cuenta los pasos fundamentales del despliegue, como el despliegue inicial, las pruebas y el ascenso a la producción, una vez que los resultados de las pruebas estén alineados con las expectativas y se hayan verificado con respecto al punto de referencia. Cada etapa del proceso de procesamiento proporciona artefactos de datos que permiten la comparación y la toma de decisiones basadas en datos.



### Pasos del CI/CD proceso de aplicación

Cada etapa puede considerarse una tarea independiente, lo que permite incorporar flujos de trabajo de validación e implementación adecuados para respaldar el servicio de red y las funciones de red nativas de la nube. Las tareas de ejecución pueden incorporar herramientas adicionales de terceros, como generadores y simuladores de tráfico, lo que permite validar los servicios end-to-end de red.

AWS proporciona un sofisticado servicio [AWS Step Function](#) (máquina de estado nativa de la nube) que se integra de forma nativa con otros AWS servicios y también tiene la capacidad de integrarse con sistemas externos como Jira o un marco de automatización de pruebas.

## Pasos detallados de CI/CD

La CI/CD puede imaginarse como un proceso en el que el código nuevo se envía por un extremo, se prueba en una serie de etapas (fuente, compilación, prueba, puesta en escena y producción) y, a continuación, se publica como código listo para la producción.

Estos son los pasos de la implementación y las pruebas. La implementación y la configuración se dividen principalmente en cuatro secciones principales:

- Configuración de la red
- Despliegue de infraestructura

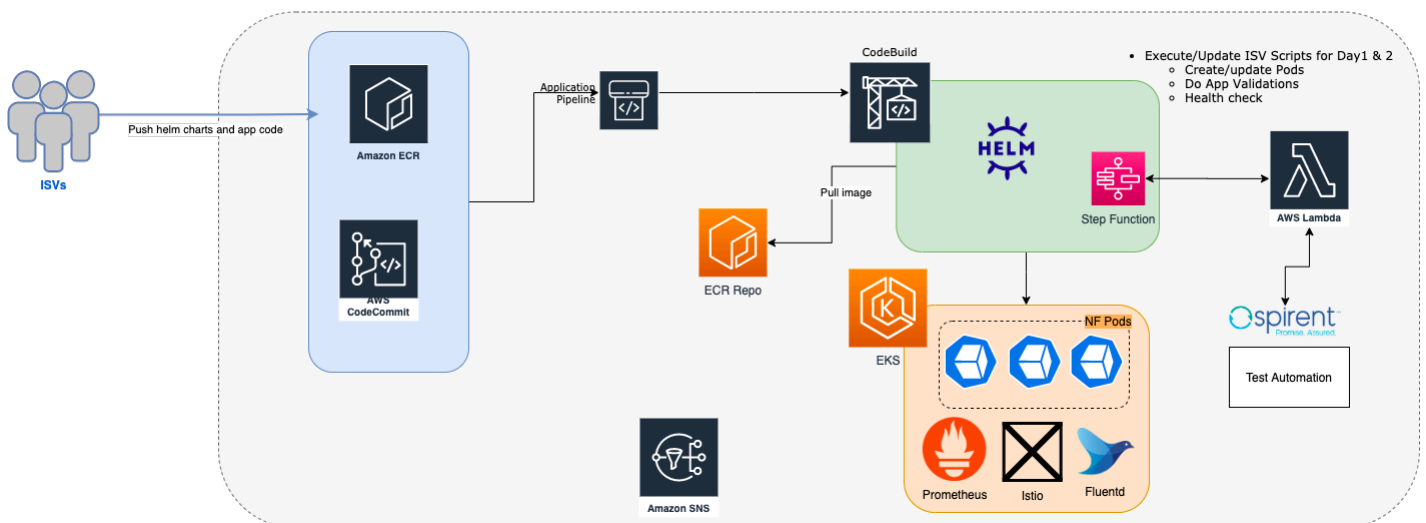
- Despliegue de funciones de red nativas de la nube
- Entrega continua de CNF

## Configuración de la red

Concéntrate en la configuración de los requisitos previos de la infraestructura. Esto incluye la creación de la VPC, las redes, las subredes NACLs, etc. Diseña el plan de red IP teniendo en cuenta ISVs el plan de implementación del cliente (por ejemplo, la multitenencia y las asignaciones estáticas frente a las dinámicas). Este plan se puede codificar en AWS CDK o CloudFormation para ejecutar este código, se implementan los requisitos previos de la red de infraestructura de nube.

## Despliegue de infraestructura

El despliegue de la infraestructura aprovisiona todos los componentes de la infraestructura. Incluye generar el clúster EKS y la infraestructura de soporte, como los nodos EFS y EKS Worker ELBs, y configurar el clúster de acuerdo con los requisitos de función de red nativos de la nube. [En función de los requisitos de la CNF, AWS también implementa interfaces de red adicionales para los nodos, incluidas las interfaces Multus.](#) La mayoría de los pasos de implementación y configuración se realizan una sola vez para una aplicación y solo se actualizan cuando es necesario como una actualización de la aplicación.

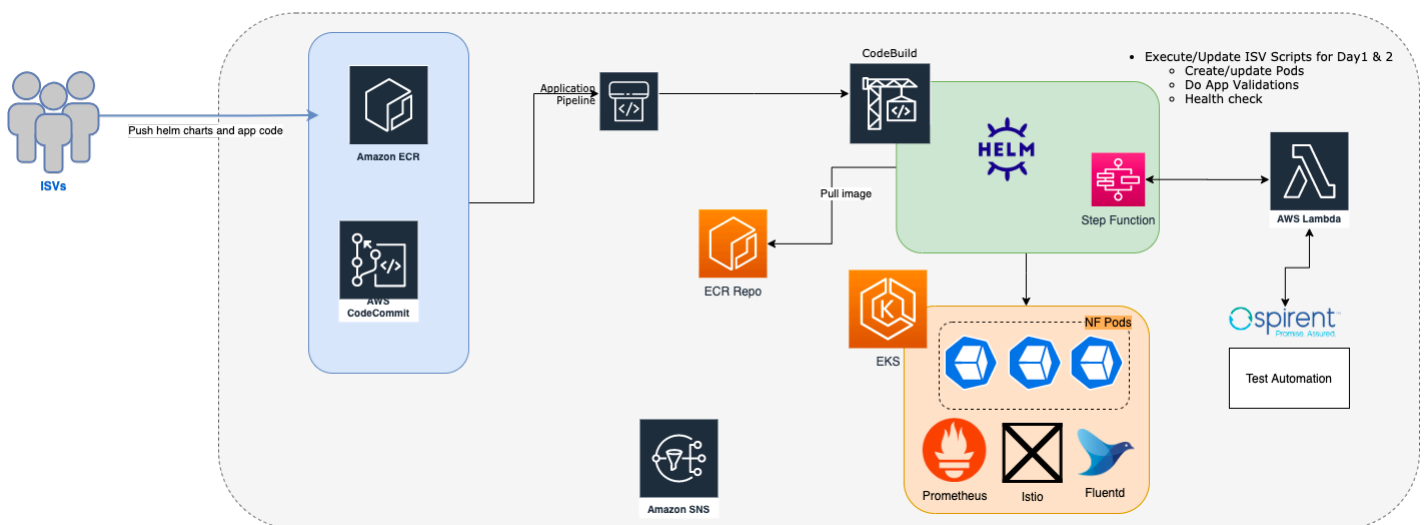


## Despliegue de infraestructura con CDK

## Despliegue de funciones de red nativas de la nube

El despliegue de la CNF tiene que ver con el despliegue de la aplicación. Como parte del despliegue de la CNF, los gráficos de Helm de la aplicación se implementan mediante la canalización de CI/CD código. Se incorpora la función de retrollamada para ejecutar scripts individuales específicos de cada aplicación, que implican principalmente comprobaciones previas y posteriores. Los gráficos de Helm se implementan en una secuencia según las necesidades de la aplicación y comprueban el estado de Kubernetes PODs antes de pasar al siguiente paso de la implementación. A menudo, ISVs proporciona un script contenedor para ejecutar los gráficos de Helm y las comprobaciones de cordura. Estos scripts ISV se invocan desde dentro. AWS CodePipeline Como parte de esta fase, se implementan los agentes de registro y monitoreo, como Prometheus y Fluentd, además de CloudWatch Amazon, que registra y monitorea la infraestructura de nube de la aplicación.

La canalización de códigos está integrada con el marco de automatización de pruebas de terceros. La canalización de código puede llamar directamente al marco de automatización de pruebas APIs para ejecutar la prueba en la aplicación implementada, consultar los resultados de las pruebas y analizarlos. Esto simplifica la implementación y las pruebas de la aplicación.

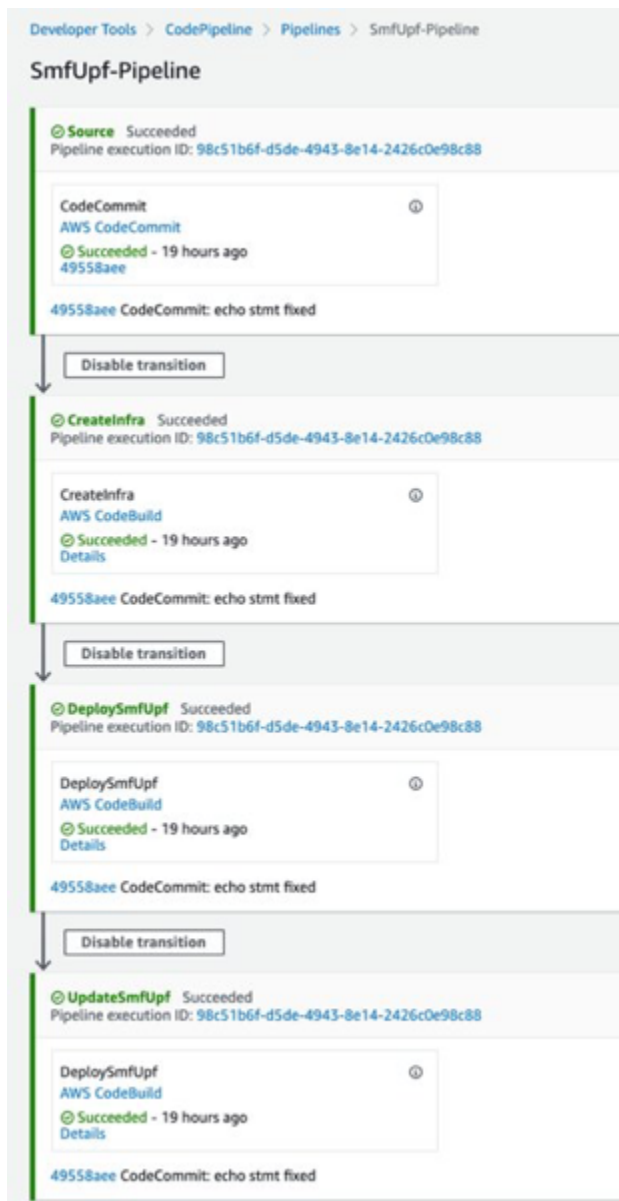


### Despliegue y actualización de aplicaciones

A continuación se muestra un ejemplo del despliegue del CNF function/session management function (UPF/SMF (plano de usuario) mediante AWS CodePipeline.

- Automatice todo el CI/CD proceso mediante CodeCommit CodeBuild, y CodePipeline.
- Las tareas de creación de infraestructuras e instalación de aplicaciones se integran como parte del proceso.

- Los agentes FluentD y Prometheus se instalan y crean en los paneles de Amazon. CloudWatch



Ejemplo de despliegue de UPF/SMF CNFs

## Entrega continua de CNF

Este paso consiste en una secuencia de pasos que se llevan a cabo repetidamente para implementar los cambios que forman parte de los container/configuration cambios que dan lugar a actualizaciones. La entrega continua de CNF se automatiza a través de canalizaciones y es específica para aplicaciones individuales. AWS utiliza gráficos de Helm estándar para actualizar

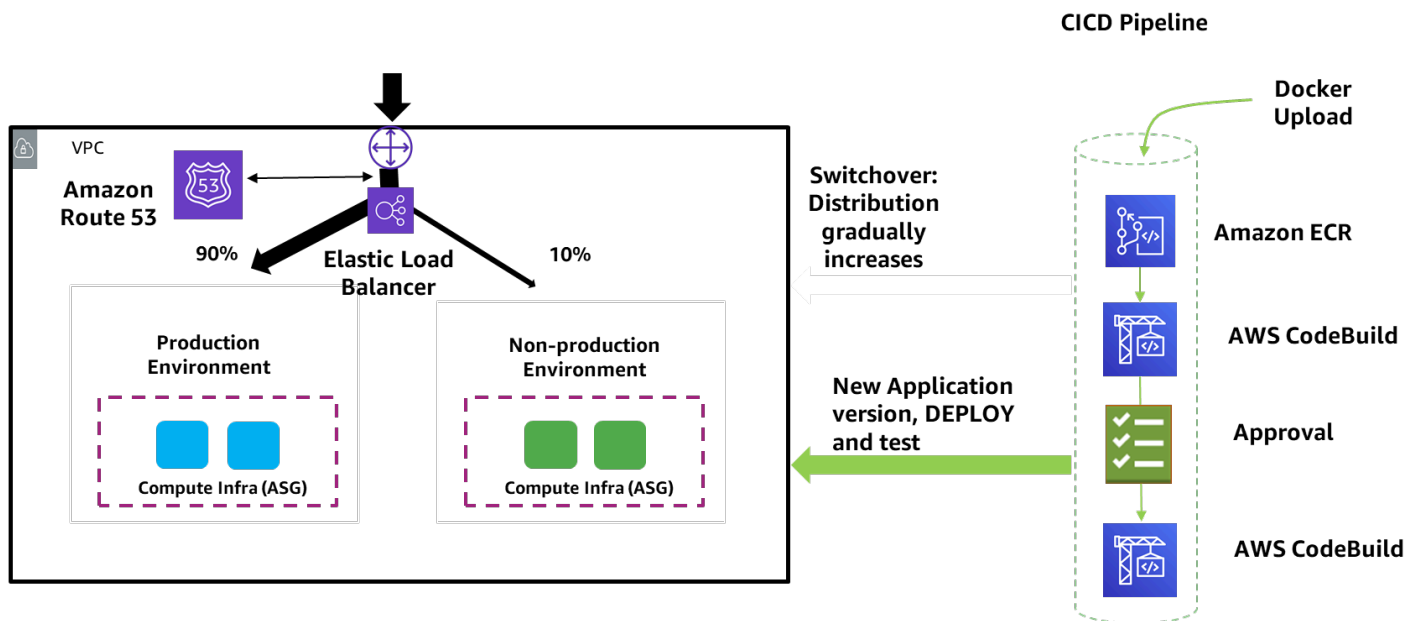
gráficos específicos. CNFs La canalización de códigos realiza comprobaciones previas y posteriores del estado de actualización de la aplicación. La CI/CD canalización actualizada también está integrada con un marco de automatización de pruebas para ejecutar pruebas automatizadas. Esta abstracción permite un despliegue limpio de las funciones de la red.

La entrega y el despliegue continuos de CNF se pueden clasificar en términos generales en las siguientes categorías:

- **Actualizaciones de aplicaciones:** la mayoría de las actualizaciones de aplicaciones son cambios en la aplicación de Kubernetes. PODs Estas actualizaciones se pueden aplicar automáticamente a través de la canalización del código. La mayoría CNFs admiten actualizaciones locales al proporcionar varias instancias de aplicación PODs. Múltiples instancias permiten un enfoque de actualización gradual. No todos los cambios en el POD de las aplicaciones admiten la actualización de Helm. Las canalizaciones tienen en cuenta estas variaciones y utilizan Helm install/delete según sea necesario.
- **Mejoras importantes:** las actualizaciones principales consisten principalmente en cambios en el esquema de la base de datos. Este cambio no se puede aplicar sin provocar algún tiempo de inactividad. El enfoque estándar para estos cambios consiste en eliminar la aplicación y volver a crear los pods correspondientes. Durante el proceso, es posible que la aplicación no esté disponible. Para las actualizaciones se utilizan las siguientes herramientas:
  - [AWS CloudFormation](#) permite a los clientes describir y aprovisionar todos los recursos de infraestructura en plantillas JSON o YAML. CloudFormation proporciona un potente mecanismo de extensión a través de recursos personalizados respaldados por Lambda. Los clientes pueden AWS CloudFormation ir más allá de los recursos de AWS y aprovisionar el recurso necesario en otros entornos, por ejemplo, recursos locales en entornos híbridos. AWS CDK ofrece a los desarrolladores la posibilidad de crear código utilizando lenguajes de programación conocidos de nivel superior, como Python, TypeScript JavaScript, Java y C#, y luego compilar el código en un formato CloudFormation JSON de nivel inferior, que luego se puede implementar.
  - **BlueGreen implementación:** AWS admite blue/green y recomienda las implementaciones basadas en canary tanto en entornos de prueba como en entornos de producción. [Blue/green las implementaciones](#) permiten a los clientes probar una nueva versión de la aplicación en un entorno contenido. Proporcionan un método sencillo y elegante para cambiar el tráfico de producción. [Los despliegues basados en Canarias](#) amplían este concepto al permitir probar el entorno ecológico

no productivo con una pequeña proporción del tráfico de producción para descubrir cualquier problema causado por el tráfico de producción. La nueva versión de la aplicación se prueba con tráfico de prueba simulado interno y con pequeñas cantidades de tráfico de producción, lo que proporciona confianza al usuario antes de cambiar el tráfico de producción. El tráfico de producción se incrementa gradualmente hasta que se complete la transición. La implementación implica grupos objetivo ponderados de DNS y ELB ponderados.

- La automatización se puede lograr configurando AWS CodePipeline con las etapas de implementación blue/green y basadas en canary. La etapa de aprobación puede realizarse manualmente inicialmente durante el aprovisionamiento, pero más adelante debería automatizarse por completo. En los entornos de prueba, se recomienda probar siempre con una acción de reversión para validar la compatibilidad con versiones anteriores y posteriores, antes de implementarlos en producción. La blue/green implementación en clústeres con malla de servicios depende del soporte que brinden la aplicación final y la pasarela de enrutamiento de la malla de servicios para lograr una transición sin problemas.
- [AWS Systems Manager](#) proporciona una interfaz de usuario unificada para que pueda ver los datos operativos de varios servicios de AWS utilizados por las funciones de red implementadas por CI/CD. Systems Manager le permite automatizar las tareas operativas en todos sus AWS recursos.



## Despliegue canario

# Seguridad

La seguridad es un elemento fundamental. A continuación se incluye una lista de los pasos de seguridad que el AWS CI/CD proceso tiene en cuenta al implementar una aplicación.

- Fuente: el repositorio de ECR asignado al proveedor está configurado con el indicador «Scan on Push» activado, de modo que cualquier carga de imágenes de Docker se someta inmediatamente a un análisis de seguridad. Todas las vulnerabilidades y exposiciones comunes (CVE) conocidas se marcarán con notificaciones. Además del ECR, cuando los proveedores colocan gráficos en el AWS CodeCommit repositorio, se les pide que cifren las contraseñas utilizadas con Secrets Manager en lugar de hacerlo con texto plano.
- Integridad de los artefactos: los artefactos utilizados en todo el proceso se cifran, ya sea en reposo (mediante claves AWS gestionadas) o en tránsito (mediante SSL/TLS).
- Usuarios y funciones: los permisos que se proporcionan a los usuarios o recursos se basan en el principio del privilegio mínimo. Debe existir una relación de confianza entre funciones y funciones que puede ser necesario configurar si se utilizan recursos de distintos servicios. Por ejemplo, AWS CodeBuild necesita permiso para ejecutar comandos en un clúster de Amazon EKS.
- Auditoría: la capacidad de auditoría que ofrece hace un [AWS CloudTrail](#) seguimiento de todas y cada una de las llamadas a la API en todos los servicios y operaciones de los usuarios, y permite evaluar los eventos pasados.
- Escaneo de vulnerabilidades de imagen: las imágenes CNF que se cargan en Amazon ECR se escanean automáticamente para detectar vulnerabilidades de seguridad. En la API se encuentra disponible un informe con los resultados del escaneo [Consola de administración de AWS](#), que también se puede consultar a través de la API. Luego, los resultados se pueden enviar a los operadores del CSP para que tomen las medidas correctivas, incluida la sustitución de la imagen CNF.

Los controles de seguridad se llevan a cabo en varias etapas del proceso para garantizar que la imagen recién cargada sea segura y cumpla con las comprobaciones de conformidad deseadas, de modo que se pueda enviar una notificación CSPs para su aprobación:

- El registro del contenedor busca cualquier vulnerabilidad de CVE abierta.
- Durante la fase de prueba, la configuración se comprueba para detectar filtraciones de información (patrones conocidos de información de identificación personal (PII), lo que activa las normas de control de la conformidad para detectar problemas como la apertura inesperada de TCP/UDP puertos o las vulnerabilidades de DOS.

- Para upgrade/rollback garantizar la seguridad, se verifica la compatibilidad con versiones anteriores y posteriores.

Además de la aplicación, es fundamental garantizar la seguridad de la canalización garantizando la transferencia cifrada de los artefactos entre etapas, ya sea en reposo o en tránsito.

## Observabilidad

AWS permite la observabilidad de las redes 5G CNFs que están desplegadas de forma AWS predeterminada. Esto lo habilita Amazon CloudWatch. CloudWatch brinda una visibilidad completa de sus recursos y aplicaciones en la nube.

Amazon CloudWatch tiene cuatro pasos principales durante este proceso:

1. Recopile: recopile métricas y registros de todos sus AWS recursos, aplicaciones y servicios que se ejecutan en servidores locales AWS y locales.
2. Supervise: [visualice las aplicaciones y la infraestructura con CloudWatch paneles, correlacione los registros y las métricas uno al lado del otro para solucionar problemas y configure alertas con alarmas. CloudWatch](#)
3. Actúe: [automatice la respuesta a los cambios operativos con CloudWatch Events y. AWS Auto Scaling](#)
4. Analice: métricas de hasta un segundo, mayor retención de datos (15 meses) y análisis en tiempo real con [CloudWatch Metric Math](#).

El CloudWatch agente de Amazon está instalado en el clúster de Kubernetes del cliente. [El agente es compatible con las funciones de configuración, descubrimiento y extracción de métricas de Prometheus, lo que enriquece y publica todas las métricas y metadatos de Prometheus de alta fidelidad en formato métrico integrado \(EMF\) en los registros. CloudWatch](#)

[Amazon CloudWatch Container Insights](#) automatiza el descubrimiento y la recopilación de métricas de Prometheus de aplicaciones en contenedores. Recopila, filtra y crea automáticamente métricas personalizadas CloudWatch agregadas que se visualizan en paneles de control.

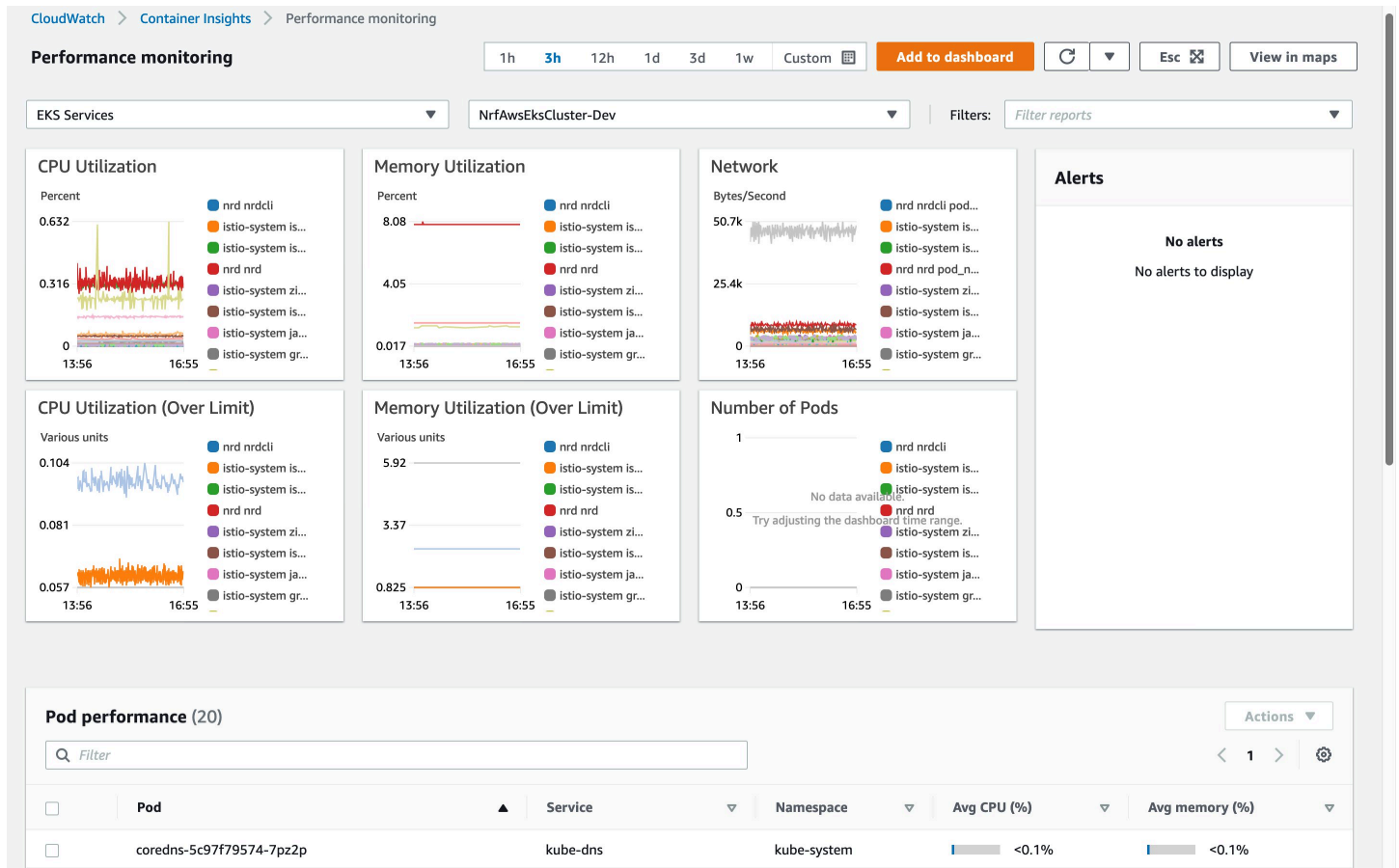
Cada evento crea puntos de datos métricos como métricas CloudWatch personalizadas para un conjunto seleccionado de dimensiones métricas que es totalmente configurable. La publicación de métricas agregadas de Prometheus CloudWatch como estadísticas de métricas personalizadas

reduce la cantidad de métricas necesarias para monitorear, emitir alarmas y solucionar problemas y fallas de rendimiento. También puede analizar las métricas de Prometheus de alta fidelidad [CloudWatch mediante el lenguaje de consultas de Logs Insights](#) para aislar cápsulas y etiquetas específicas que afectan al estado y el rendimiento de sus entornos en contenedores.

AWS CloudTrail ofrece esta visibilidad al registrar todas las llamadas a la API en todos los servicios. [AWS Config](#) ofrece la capacidad de validar el cumplimiento. AWS ofrece a los clientes opciones de monitoreo adicionales de métricas, registros y eventos para la aplicación, la infraestructura y las canalizaciones, mediante varios servicios como [AWS X-Ray](#) y [AWS CloudTrail](#).

- AWS puede integrar de forma nativa herramientas métricas de código abierto como Prometheus, Fluentd, etc.
- Las métricas de [Prometheus](#) se pueden incorporar posteriormente a CloudWatch Amazon OpenSearch o Service para analizarlas más a fondo.
- AWS utiliza FluentD como mecanismo estándar para recopilar registros de varios sistemas. Ese mismo mecanismo se usa y configura para este proyecto.

Para obtener detalles sobre cómo configurar este mecanismo, consulte [Configurar FluentD como para enviar registros DaemonSet a registros](#). CloudWatch



Ejemplo de métricas CloudWatch monitorizadas por Amazon

# Orquestación de CI/CD con herramientas de terceros y de código abierto

La capa de orquestación utiliza la IaC para implementar y configurar la infraestructura subyacente necesaria para ejecutar las funciones de la red 5G. Esta capa debe diseñarse para que sea modular, portátil y reutilizable.

La infraestructura sigue las mejores prácticas nativas de la nube y es de alta disponibilidad, redundante y escalable.

Como se demostró en las secciones anteriores, el despliegue de la infraestructura subyacente se puede lograr mediante el [AWS Cloud Development Kit \(AWS CDK\)](#). Esto se puede lograr utilizando [Terraform](#) de Hashicorp.

## Terraform

Terraform es una herramienta de software de IaC que proporciona un flujo de trabajo de interfaz de línea de comandos (CLI) coherente para administrar cientos de servicios en la nube. Terraform codifica la nube APIs en archivos de configuración declarativos.

Para la implementación con Terraform, utilice los mismos principios que se utilizan en CDK. El código está estructurado en módulos que permiten personalizar y reutilizar los componentes de la red de acuerdo con los requisitos del proveedor.

Toda la configuración está parametrizada, lo que permite personalizar completamente las implementaciones de acuerdo con las recomendaciones de los proveedores y los ISV.

El despliegue de las funciones de red se divide en dos fases:

- La AWS infraestructura requerida se crea y administra a través de un repositorio central.
- La configuración y el código se almacenan de forma centralizada en un GitHub repositorio.

Una vez creados los requisitos previos, la función de red está lista para implementarse mediante una canalización de aplicaciones que se estableció en la etapa anterior.

# Despliegue de infraestructura

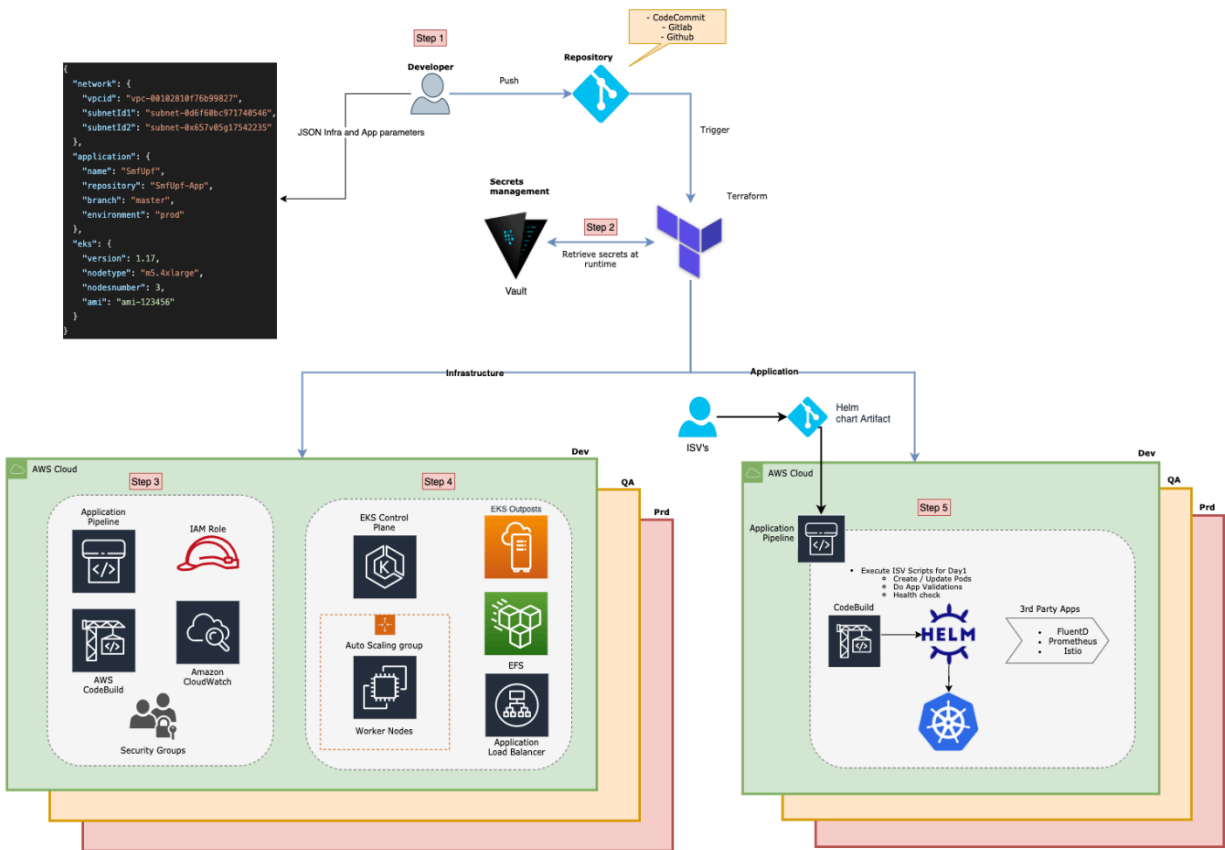
El despliegue de la infraestructura incluye todos los requisitos previos para que la función de red se despliegue y configure correctamente.

Algunos de los componentes creados como parte de esta fase son:

- Redes: VPC, subredes públicas y privadas, rutas, balanceadores de carga
- Computación: Kubernetes ([Vmware Tanzu](#), Amazon EKS AWS Outposts o), instancias Amazon EC2, nodos principales y de trabajo, grupo de autoescalado
- Almacenamiento: Amazon EFS, Amazon EBS, Amazon S3 bucket
- Seguridad: grupos [de seguridad](#)
- Oleoducto — CodePipeline, CodeBuild
- Observabilidad — CloudWatch, Prometheus, FluentD

Esta es la secuencia de infraestructura orquestada por Terraform y explicada en la siguiente figura:

1. Un desarrollador rellena un archivo JSON que está almacenado en un repositorio central con el código IaC. El archivo contiene información sobre la configuración de infraestructura deseada, como el tamaño de las instancias, la versión de Kubernetes, la información de la red y los detalles del repositorio de aplicaciones.
2. Recupera los secretos de HashiCorp Vault o [AWS Secrets Manager](#) en tiempo de ejecución.
3. Implementa y configura los componentes de la infraestructura (redes, cómputo, almacenamiento y seguridad).
4. Se implementa un clúster de Amazon EKS con nodos de trabajo que alojan los pods de funciones de red. Amazon EKS también se puede implementar [AWS Outposts](#) para soportar cargas de trabajo que requieren proximidad a un centro de datos.
5. Se crea y configura una canalización de aplicaciones para detectar cambios en el repositorio de funciones de red. Cada vez que se envía código a la rama de repositorio configurada, la canalización activa automáticamente la creación, la prueba y el despliegue de la función de red.
6. [Las herramientas de observabilidad que recopilan y centralizan registros y métricas se implementan como servicios en todos los nodos y proporcionan datos casi en tiempo real que se pueden visualizar en Grafana o en Dashboards OpenSearch](#)



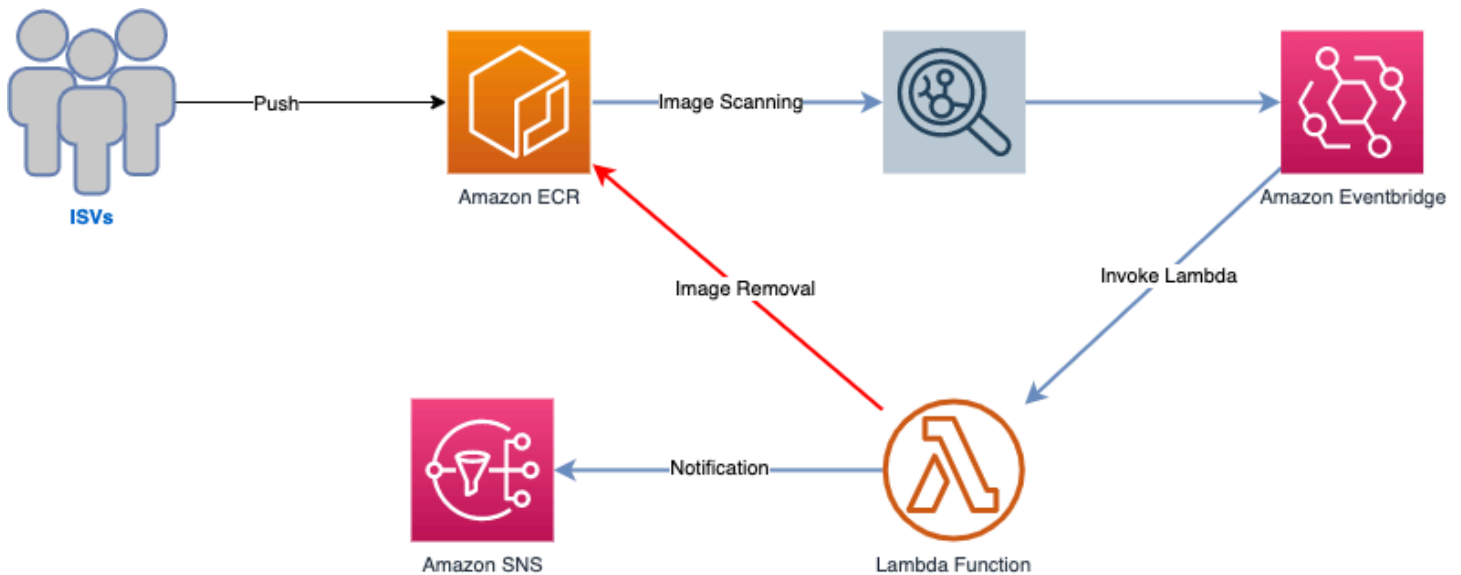
## Despliegue y configuración de funciones de red

## Despliegue y configuración de funciones de red

La canalización creada en la etapa anterior permite tanto ISVs a los proveedores como a los proveedores descentralizar y optimizar el despliegue de las funciones de la red. La canalización está conectada y escucha los cambios en el repositorio de aplicaciones, que se configura en el archivo JSON en el paso 1 de la figura anterior.

Para examinar las imágenes publicadas por terceros, se ha implementado y configurado una solución de escaneo de vulnerabilidades que ayuda a identificar las vulnerabilidades de software en las imágenes de los contenedores. La solución de digitalización comprueba automáticamente todas las imágenes nuevas que se envían a [Amazon ECR](#). Para obtener más información sobre el escaneo de imágenes mediante ECR, consulte Digitalización de [imágenes](#).

La siguiente figura muestra la arquitectura de la solución de escaneo de vulnerabilidades de imagen.



### Arquitectura de la solución de escaneo de vulnerabilidades de imagen

La canalización de aplicaciones se puede configurar para que se active por cambios en la imagen tras el resultado del escaneo o por cambios directos en el repositorio. Por ejemplo, cuando se crea una nueva imagen de Helm.

La siguiente lista es la secuencia en creates/upgrades la que funciona la red, tal como se muestra en la siguiente figura:

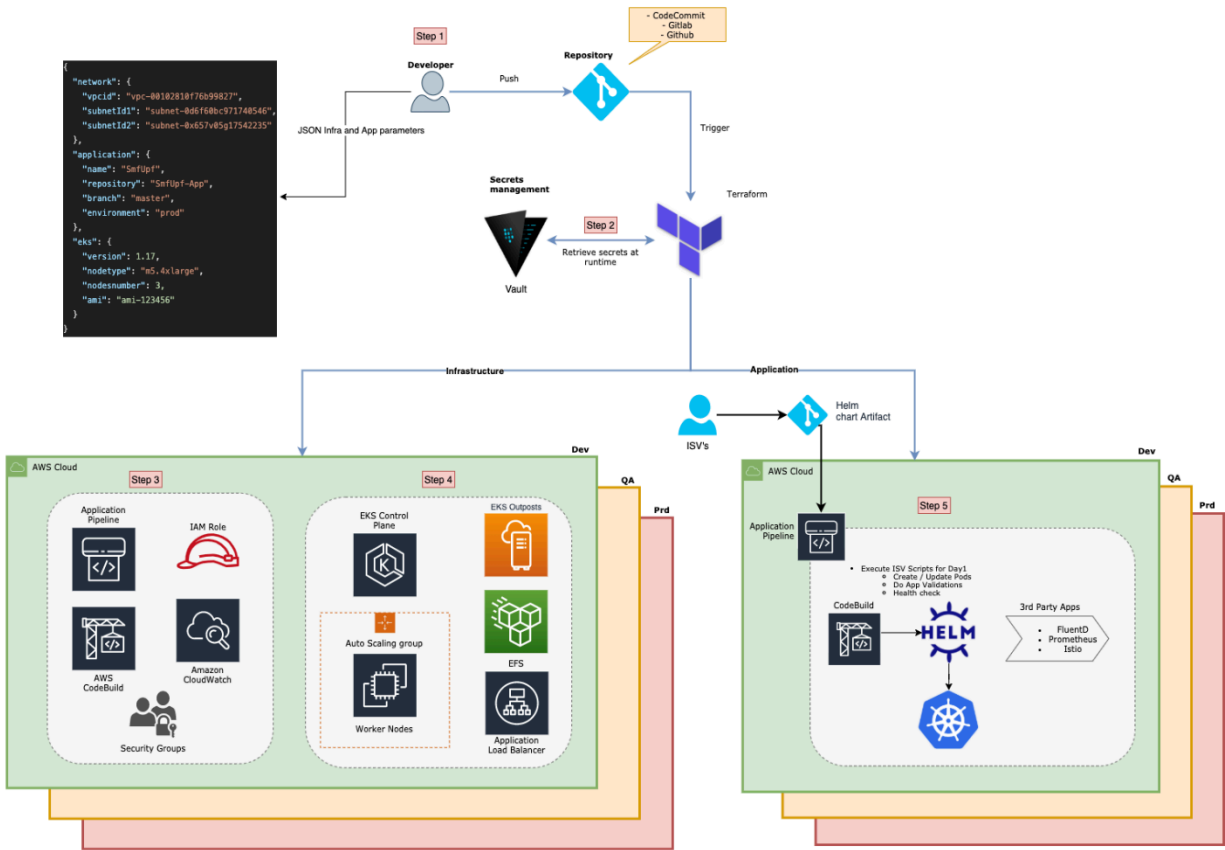
ISVs publique nuevas imágenes en Amazon ECR. (Si se aprueba la imagen, se activa la canalización de la solicitud.

CodePipeline extrae la nueva imagen de Amazon ECR y la utiliza CodeBuild para implementar la imagen en Kubernetes. Los comandos Helm se pueden utilizar para actualizar la función de red.

Una vez desplegada la imagen, se activa la prueba como servicio (TaS). TaS valida la nueva implementación y centraliza los datos y las métricas sobre el rendimiento de las funciones de la red en situaciones de stress.

Los registros y las métricas se recopilan y centralizan en OpenSearch Grafana. También se pueden configurar terceros como [Datadog](#), [Istio](#) y Prometheus para proporcionar una observabilidad adicional.

También se puede implementar e integrar con la solución un MANO que pueda coordinar los recursos de la red. Utiliza los datos recopilados para realizar acciones automatizadas, como la segmentación de la red y el escalado automático de la calidad de servicio (QoS).



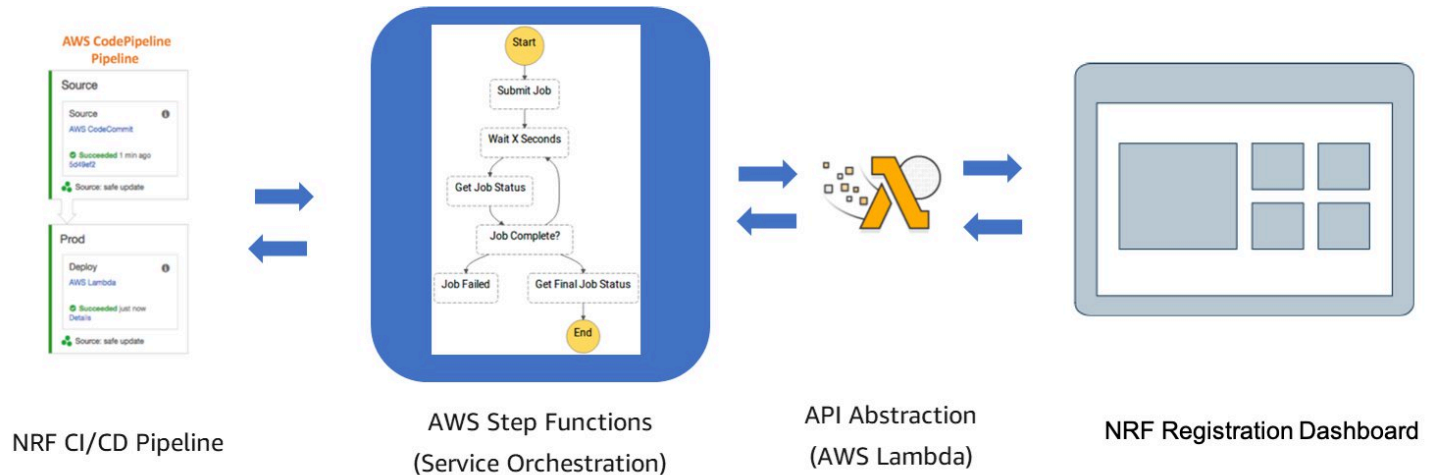
### Canalización de aplicaciones

## Testeo

El marco de automatización de pruebas específico para telecomunicaciones se puede integrar en la canalización de códigos. La canalización de código está integrada con funciones escalonadas para organizar la integración con un marco de automatización de pruebas. Las funciones Step de AWS utilizan varias funciones de Lambda para invocar el marco de automatización de pruebas (herramientas de terceros) mediante llamadas a la API. La función step obtendrá inicialmente el ID de la prueba, la ejecutará y obtendrá los resultados del marco de automatización de la prueba. A continuación, la función escalonada analiza los resultados y los pasa a la canalización de códigos para que se apruebe la aplicación para su despliegue en producción. La aprobación puede automatizarse o mantenerse de forma manual en el proceso de codificación, según sea necesario. Este es un paso importante CSPs para promover el despliegue desde el entorno de prueba hasta el de producción. Los niveles altos APIs necesarios para la integración se clasifican de la siguiente manera:

- Obtenga el contexto
- Ejecute un caso de prueba específico
- Detenga el caso de prueba
- Obtenga resultados

La complejidad de llamar a un REST externo APIs se modela mediante funciones escalonadas de AWS, que permiten a las construcciones estándar invocar flujos paralelos, esperar los resultados, ramificar en función de las condiciones e integrar la API REST con ella. AWS CodePipeline



## Flujo de pruebas

## CI/CD y orquestación

La integración y la entrega continuas forman parte de una filosofía de automatización general que acompaña a la arquitectura nativa de la nube y a su aplicación a la 5G. La orquestación es otro aspecto de esta filosofía que se espera que sea dinámico y reaccione ante cualquier cambio que se produzca en la red. Se espera que la organización y la coordinación CI/CD estén estrechamente vinculadas para garantizar un servicio saludable y minimizar las interrupciones del servicio. Se espera que la integración CI/CD y la orquestación se produzcan en dos frentes:

- La aplicación de parches y actualizaciones al sistema debe gestionarse y organizarse de forma que se minimice la interrupción de cualquier servicio activo. Por ejemplo, la organización puede determinar de forma dinámica el mejor momento para implementar una actualización.
- La orquestación compatible con CI/CD permite desplazar el tráfico durante la implementación de las actualizaciones en función de la estrategia del modelo de implementación adoptada (canaria, lineal o). all-at-once

Por lo general, las soluciones de CI/CD orquestación se ejecutan por etapas para permitir que la organización introduzca fases de gobierno en esas canalizaciones y esté expuesta a los ciclos de actualización en curso.

## Conclusión

La CI/CD proporciona un camino claro y eficiente para que los desarrolladores y los equipos de aplicaciones desplieguen un nuevo código de aplicación en cuestión de minutos. AWS cuenta con una gran cantidad de herramientas que pueden ayudar a los desarrolladores a integrar, probar e implementar código nuevo AWS CodePipeline, incluidas AWS CodeCommit, AWS CodeBuild, AWS CodeDeploy, y muchas otras. En este documento, se analizó cómo se pueden utilizar los AWS servicios para crear un CI/CD proceso de implementación de la función de red 5G de forma totalmente automatizada, incluidos los diferentes pasos necesarios para completar la implementación del nuevo código. También se analizó cómo integrar un marco de automatización de pruebas de terceros en el CI/CD proceso, así como el uso de herramientas de terceros, como Terraform.

# Colaboradores

Los colaboradores de este documento son:

- Hisham Elshaer, consultor sénior de AWS Telecom, Amazon Web Services
- Vara Prasad Talari, consultora principal de AWS Telecom, Amazon Web Services
- Rabi Abdel, consultor principal de AWS Telecom, Amazon Web Services
- Franco Bontorin, consultor sénior de Shared Delivery, Amazon Web Services
- Pragtideep Singh, consultor de Shared Delivery, Amazon Web Services
- Subbarao Duggisetty, arquitecto de infraestructura de nube, cuentas globales, Amazon Web Services
- Young Jung, socio principal de arquitectura de soluciones, AWS Telecom, Amazon Web Services

## Revisiones del documento

Para recibir notificaciones sobre las actualizaciones de este documento técnico, suscríbase a la fuente RSS.

Cambio	Descripción	Fecha
<a href="#">Publicación inicial</a>	Documento técnico publicado por primera vez	8 de marzo de 2021

# Documentación adicional

Para obtener información adicional, consulte:

- [Cómo practicar la integración y la entrega continuas en AWS](#) (documento técnico)
- [Red central de paquetes móviles de nivel de operador en AWS](#) (documento técnico)
- [Evolución de la red 5G con AWS](#) (documento técnico)

# Acrónimos

- AMF: función de gestión de acceso y movilidad
- API: interfaz de programación de aplicaciones
- AUSF: función de servidor de autenticación
- BSS — Business Support System
- CDK — Kit de desarrollo en la nube
- CI/CD: integración continua y entrega continua
- CLI: interfaz de línea de comandos
- CNF: función de red nativa de la nube o en contenedores
- CSP: proveedor de servicios de comunicación
- CU — Unidad central RAN
- CVE: vulnerabilidades y exposiciones comunes
- DoS — Denegación de servicio
- DR: recuperación ante desastres
- Unidad distribuida DU — RAN
- E2E: extremo a extremo
- ECR: Elastic Container Registry
- EFS: sistema de archivos elástico
- EKS: servicio de Elastic Kubernetes
- EPC: núcleo de paquetes evolucionado
- IaC: infraestructura como código
- ISV: proveedor de software independiente
- MANO: gestión y organización
- MEC: computación perimetral de acceso múltiple
- NACL: Lista de control de acceso a la red
- NAT: traducción de direcciones de red
- NF: función de red
- NFV: virtualización de funciones de red
- NFVO: orquestador de virtualización de funciones de red

- NOC: Centro de operaciones de red
- NRF: función de repositorio de red
- OSS — Operations Support System
- PII: información de identificación personal
- QoS: calidad de servicio
- RAN: red de acceso por radio
- SBI: interfaz basada en servicios
- SMF: función de gestión de sesiones
- SSL: capa de sockets seguros
- TaS: prueba como servicio
- TCP: Protocolo de control de transmisión
- TLS: seguridad de la capa de transporte
- UDM: gestión de datos unificada
- UDP: protocolo de datagramas de usuario
- UPF: función de plano de usuario
- VIM: administrador de infraestructura virtualizada
- VNF: función de red virtual
- VPC: nube privada virtual

# Avisos

Es responsabilidad de los clientes realizar su propia evaluación independiente de la información que contiene este documento. El presente documento: (a) tiene solo fines informativos, (b) representa las ofertas y prácticas actuales de los productos de AWS, que están sujetas a cambios sin previo aviso, y (c) no supone ningún compromiso ni garantía por parte de AWS y sus filiales, proveedores o licenciantes. Los productos o servicios de AWS se proporcionan “tal cual” sin garantías, declaraciones ni condiciones de ningún tipo, ya sean expresas o implícitas. Las responsabilidades y obligaciones de AWS con respecto a sus clientes se controlan mediante los acuerdos de AWS y este documento no forma parte ni modifica ningún acuerdo entre AWS y sus clientes.

© 2021 Amazon Web Services, Inc. o sus filiales. Todos los derechos reservados.

Las traducciones son generadas a través de traducción automática. En caso de conflicto entre la traducción y la versión original de inglés, prevalecerá la versión en inglés.