



Documento técnico de AWS

# Introduction a DevOps en AWS



---

# Introduction a DevOps en AWS: Documento técnico de AWS

Copyright © 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 de ninguna manera que menosprecie o desacredite a Amazon. Todas las demás marcas comerciales 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 .....	1
Resumen .....	1
Introducción .....	2
Integración continua .....	3
AWS CodeCommit .....	3
AWS CodeBuild .....	4
AWS CodeArtifact .....	5
Entrega continua .....	6
AWS CodeDeploy .....	6
AWS CodePipeline .....	7
Estrategias de implementación .....	9
Implementaciones in situ .....	9
Implementaciones azul/verde .....	9
Implementaciones de valores controlados .....	10
Implementaciones lineales .....	10
Implementaciones todo a la vez .....	10
Matriz de estrategias de implementación .....	11
Estrategias de implementación de AWS Elastic Beanstalk .....	11
Infraestructura como código .....	13
AWS CloudFormation .....	14
Kit de desarrollo en la nube de AWS .....	15
Kit de desarrollo en la nube de AWS para Kubernetes .....	16
Automatización .....	17
AWS OpsWorks .....	18
AWS Elastic Beanstalk .....	19
Supervisión y registro .....	20
Amazon CloudWatch .....	20
Alarmas de Amazon CloudWatch .....	20
Amazon CloudWatch Logs .....	21
Amazon CloudWatch Logs Insights .....	21
Amazon CloudWatch Events .....	21
Amazon EventBridge .....	22
AWS CloudTrail .....	22
Comunicación y colaboración .....	23

---

Equipos de dos pizzas .....	23
Seguridad .....	24
Modelo de responsabilidad compartida de AWS .....	24
Identity and Access Management .....	25
Conclusión .....	26
Revisiones del documento .....	27
Colaboradores .....	28
Avisos .....	29

# Introduction a DevOps en AWS

Fecha de publicación: 16 de octubre de 2020 ([Revisiones del documento](#))

## Resumen

Hoy más que nunca, las empresas se embarcan en su trayectoria de transformación digital para construir conexiones más profundas con sus clientes y lograr un valor empresarial sostenible y duradero. Organizaciones de todas las formas y tamaños logran desestabilizar a la competencia e ingresar a nuevos mercados porque su ritmo de innovación es más ágil y eficiente. Para estas organizaciones, es importante centrarse en la innovación y la disrupción del software, por lo que es fundamental optimizar su entrega de software. Las organizaciones que acorten el plazo desde la idea hasta la producción, haciendo que la velocidad y la agilidad sean una prioridad, podrían ser los disruptores del mañana.

Si bien deben tenerse en cuenta varios factores para convertirse en el próximo disruptor digital, este documento técnico se centra en DevOps y en los servicios y características de la plataforma de AWS que ayudarán a aumentar la capacidad de una organización para entregar aplicaciones y servicios a gran velocidad.

# Introducción

DevOps es la combinación de herramientas y prácticas y patrones culturales y de ingeniería que aumentan la capacidad de una organización de entregar aplicaciones y servicios a alta velocidad y con mejor calidad. Con el tiempo, han surgido varias prácticas esenciales al adoptar DevOps: integración continua, entrega continua, infraestructura como código y supervisión y registro.

Este documento destaca las capacidades de AWS que contribuirán a acelerar su transición a DevOps y cómo los servicios de AWS pueden ayudar a eliminar el trabajo pesado indiscriminado asociado con la adaptación a DevOps. También destacamos cómo crear una capacidad de integración y entrega continuas sin administrar servidores ni construir nodos, y cómo aprovechar la infraestructura como código para aprovisionar y administrar sus recursos en la nube de manera coherente y repetible.

- Integración continua: es una práctica de desarrollo de software mediante la cual los desarrolladores combinan los cambios en el código en un repositorio central de forma periódica, tras lo cual se ejecutan versiones y pruebas automáticas.
- Entrega continua: es una práctica de desarrollo de software mediante la cual se compilan, prueban y preparan automáticamente los cambios en el código y se entregan a la fase de producción.
- Infraestructura como código: es una práctica mediante la que se aprovisiona y administra infraestructura con técnicas de desarrollo de código y de software, como el control de versiones y la integración continua.
- Supervisión y registro: permite a las organizaciones ver cómo el rendimiento de las aplicaciones y la infraestructura afecta a la experiencia que el usuario final tiene de su producto.
- Comunicación y colaboración: se establecen prácticas para acercar a los equipos y crear flujos de trabajo y distribuir las responsabilidades de DevOps.
- Seguridad: debe ser una preocupación transversal. Es preciso proteger sus canalizaciones de integración y entrega continuas (CI/CD) y los servicios relacionados y se deben configurar permisos de control de acceso adecuados.

Un examen de cada uno de estos principios revela una estrecha relación con las ofertas disponibles en Amazon Web Services (AWS).

# Integración continua

La integración continua (CI) es un método de desarrollo de software en el que los desarrolladores fusionan periódicamente sus cambios de código en un repositorio de código central y, con posterioridad, se ejecutan compilaciones y pruebas automatizadas. La CI ayuda a encontrar y arreglar errores con mayor rapidez, mejorar la calidad del software y reducir el tiempo que se tarda en validar y lanzar nuevas actualizaciones de software.

AWS ofrece los siguientes servicios para la integración continua:

Temas

- [AWS CodeCommit](#)
- [AWS CodeBuild](#)
- [AWS CodeArtifact](#)

## AWS CodeCommit

[AWS CodeCommit](#) es un servicio de control de código fuente administrado, seguro y altamente escalable que aloja repositorios Git privados. CodeCommit elimina la necesidad de operar su propio sistema de control de código fuente y no es necesario aprovisionar ni escalar hardware ni instalar, configurar ni operar software. CodeCommit le permite almacenar desde código hasta binarios y soporta la funcionalidad estándar de Git, por lo que puede trabajar con sus herramientas basadas en Git sin problemas. Su equipo también puede utilizar las herramientas de código en línea de CodeCommit para colaborar en proyectos, examinarlos y editarlos. AWS CodeCommit ofrece varias ventajas:

**Colaboración:** AWS CodeCommit se diseñó para el desarrollo colaborativo de software. Puede confirmar, bifurcar y combinar el código para mantener el control sobre los proyectos del equipo de forma sencilla. CodeCommit también es compatible con las solicitudes de incorporación de cambios, que brindan un mecanismo para solicitar revisiones del código y debatir acerca del código con colaboradores.

**Cifrado:** puede transferir archivos desde y hacia AWS CodeCommit con HTTPS u SSH, en función de sus preferencias. Los repositorios también se cifran automáticamente en reposo con [AWS Key Management Service](#) (AWS KMS) mediante el uso de claves específicas de los clientes.

Control de acceso: AWS CodeCommit utiliza [AWS Identity and Access Management](#) (IAM) para controlar y supervisar quién puede obtener acceso a los datos y cómo, cuándo y dónde pueden obtener acceso a ellos. CodeCommit también le ayuda a supervisar los repositorios a través de [AWS CloudTrail](#) y [Amazon CloudWatch](#).

Alta disponibilidad y durabilidad: AWS CodeCommit almacena los repositorios en [Amazon Simple Storage Service](#) (Amazon S3) y [Amazon DynamoDB](#). Los datos cifrados se almacenan de manera redundante en varias instalaciones. Esta arquitectura aumenta la disponibilidad y durabilidad de los datos del repositorio.

Notificaciones y secuencias de comandos personalizadas: ahora puede recibir notificaciones para los eventos que afecten a sus repositorios. Las notificaciones aparecerán como notificaciones de [Amazon Simple Notification Service](#) (Amazon SNS). Cada notificación incluirá un mensaje de estado, así como un enlace a los recursos cuyo evento generó esa notificación. Además, mediante el uso de los activadores de repositorios de AWS CodeCommit, puede enviar notificaciones y crear webhooks HTTP con Amazon SNS o invocar funciones de [AWS Lambda](#) como respuesta a los eventos del repositorio que elija.

## AWS CodeBuild

[AWS CodeBuild](#) es un servicio de integración continua completamente administrado que compila código fuente, ejecuta pruebas y produce paquetes de software listos para su implementación. No es necesario aprovisionar, administrar ni escalar sus propios servidores de compilación. CodeBuild puede utilizar GitHub, GitHub Enterprise, BitBucket, AWS CodeCommit o Amazon S3 como proveedor de código fuente.

CodeBuild escala continuamente y procesa varias compilaciones a la vez. CodeBuild ofrece varios entornos preconfigurados para varias versiones de Microsoft Windows y Linux. Los clientes también pueden llevar sus entornos de compilación personalizados como contenedores Docker. CodeBuild también se integra con herramientas de código abierto como Jenkins y Spinnaker.

CodeBuild también puede crear informes para pruebas unitarias, funcionales o de integración. Estos informes proporcionan una panorámica de cuántos casos de pruebas se ejecutaron y cuántos se aprobaron o no aprobaron. El proceso de compilación también se puede ejecutar dentro de una [Amazon Virtual Private Cloud](#) (Amazon VPC), lo que puede resultar útil si sus servicios de integración o bases de datos se implementan dentro de una VPC.



# AWS CodeArtifact

[AWS CodeArtifact](#) es un servicio de repositorio de artefactos completamente administrado que las organizaciones pueden emplear para almacenar, publicar y compartir de forma segura paquetes utilizados en sus procesos de desarrollo de software. CodeArtifact puede configurarse para recuperar automáticamente paquetes de software y dependencias de repositorios de artefactos públicos para que los desarrolladores accedan a las versiones más recientes.

Los equipos de desarrollo de software confían cada vez más en paquetes de código abierto para realizar tareas comunes en su paquete de aplicaciones. Ahora ha llegado a ser fundamental para los equipos de desarrollo de software mantener el control sobre una versión particular del software de código abierto que esté libre de vulnerabilidades. Con CodeArtifact, puede configurar controles para aplicar todo lo anterior.

CodeArtifact funciona con administradores de paquetes y herramientas de compilación de uso habitual como Maven, Gradle, npm, yarn, twine y pip, lo que hacen que sea fácil de integrar en flujos de trabajo de desarrollo existentes.

# Entrega continua

La entrega continua es una práctica de desarrollo de software mediante la cual se preparan automáticamente los cambios en el código y se entregan a la fase de producción. Fundamental para el desarrollo de aplicaciones modernas, la entrega continua amplía la integración continua al implementar todos los cambios en el código en un entorno de pruebas o de producción después de la fase de compilación. Cuando la entrega continua se implementa de manera adecuada, los desarrolladores dispondrán siempre de un artefacto de compilación listo para su implementación que se ha sometido a un proceso de pruebas estandarizado.

La entrega continua permite a los desarrolladores automatizar las pruebas más allá de las pruebas de unidades, por lo que pueden verificar actualizaciones en las aplicaciones en varias dimensiones antes de enviarlas a los clientes. Las pruebas pueden incluir pruebas de la UI, de carga, de integración, de fiabilidad de la API, etc. De este modo, los desarrolladores pueden validar las actualizaciones de forma más exhaustiva y descubrir problemas por anticipado. Con la nube, resulta sencillo y rentable automatizar la creación y replicación de varios entornos de pruebas, algo que anteriormente era complicado localmente.

AWS ofrece los siguientes servicios para la entrega continua:

- [AWS CodeBuild](#)
- [AWS CodeDeploy](#)
- [AWS CodePipeline](#)

Temas

- [AWS CodeDeploy](#)
- [AWS CodePipeline](#)

## AWS CodeDeploy

[AWS CodeDeploy](#) es un servicio de implementación completamente administrado que automatiza las implementaciones de software en diferentes servicios de computación, como [Amazon Elastic Compute Cloud](#) (Amazon EC2), [AWS Fargate](#), AWS Lambda y sus servidores locales. AWS CodeDeploy facilita el lanzamiento rápido de nuevas características, ayuda a evitar tiempos de inactividad durante la implementación de una aplicación y administra la compleja actualización de las

aplicaciones. Puede usar AWS CodeDeploy para automatizar las implementaciones de software y eliminar así las operaciones manuales propensas a errores. El servicio se adapta a sus necesidades de implementación.

CodeDeploy tiene varios beneficios que se alinean con el principio de DevOps de implementación continua:

**Implementaciones automatizadas:** AWS CodeDeploy automatiza íntegramente las implementaciones de software, lo que permite realizarlas con fiabilidad y rapidez.

**Control centralizado:** AWS CodeDeploy le permite lanzar y supervisar el estado de las implementaciones de aplicaciones con facilidad a través de la consola de administración de AWS o AWS CLI. CodeDeploy le ofrece un informe detallado que le permite ver cuándo y dónde se implementó cada revisión de aplicación. También puede crear notificaciones push para recibir actualizaciones en tiempo real sobre sus implementaciones.

**Minimizar el tiempo de inactividad:** AWS CodeDeploy le ayuda a maximizar la disponibilidad de la aplicación durante el proceso de implementación de software. Introduce cambios progresivamente y supervisa el estado de las aplicaciones en función de reglas configurables. Las implementaciones de software se pueden detener y revertir con facilidad si ocurren errores.

**Adopción sencilla:** CodeDeploy funciona con cualquier aplicación y ofrece la misma experiencia en diferentes plataformas y lenguajes. Puede volver a utilizar con facilidad el código de configuración existente. CodeDeploy también se puede integrar con el proceso de lanzamiento de software existente o con la cadena de herramientas de entrega constante (por ejemplo, AWS CodePipeline, GitHub, Jenkins).

AWS CodeDeploy admite varias opciones de implementación. Para obtener más información, consulte [Estrategias de implementación](#).

## AWS CodePipeline

[AWS CodePipeline](#) es un servicio de entrega continua que le permite modelar, visualizar y automatizar los pasos necesarios para lanzar su software. Con AWS CodePipeline, puede modelar el proceso de lanzamiento completo para compilar el código, realizar implementaciones en entornos de preproducción, probar la aplicación y lanzarla en producción. A continuación, AWS CodePipeline compila, prueba e implementa la aplicación según el flujo de trabajo definido cada vez que se produce una modificación en el código. Puede integrar herramientas de socios de APN y

herramientas personalizadas propias en cualquier fase del proceso de lanzamiento para crear una solución integral de entrega continua.

AWS CodePipeline tiene varios beneficios que se alinean con el principio DevOps de implementación continua:

**Entrega rápida:** AWS CodePipeline automatiza el proceso de lanzamiento de software para que pueda poner a disposición de los usuarios las nuevas características con rapidez. Con CodePipeline, puede iterar rápidamente los comentarios y poner a disposición de los usuarios nuevas características con mayor rapidez.

**Calidad mejorada:** mediante la automatización de los procesos de compilación, prueba y lanzamiento, AWS CodePipeline le permite aumentar la velocidad y calidad de las actualizaciones de software mediante la ejecución de todos los cambios nuevos a través de un conjunto coherente de comprobaciones de calidad.

**Integración sencilla:** AWS CodePipeline se puede ampliar con facilidad para adaptarse a las necesidades específicas de cada usuario. Puede utilizar los complementos precompilados o complementos personalizados propios en cualquier paso del proceso de lanzamiento. Por ejemplo, puede extraer el código fuente de GitHub, utilizar el servidor de compilación de Jenkins local, ejecutar pruebas de carga con un servicio de terceros o transferir información de implementación al panel de operaciones personalizado.

**Flujo de trabajo configurable:** AWS CodePipeline le permite modelar las diferentes fases del proceso de lanzamiento de software mediante el uso de la interfaz de la consola, AWS CLI, [AWS CloudFormation](#) o los SDK de AWS. Puede especificar fácilmente las pruebas que se van a ejecutar y personalizar los pasos para implementar la aplicación y sus dependencias.

# Estrategias de implementación

Las estrategias de implementación definen cómo quiere entregar su software. Las organizaciones siguen diferentes estrategias de implementación en función de su modelo de negocio. Algunas pueden optar por entregar software que está completamente probado, y otras pueden querer que sus usuarios proporcionen comentarios y dejan que sus usuarios evalúen las características en desarrollo (por ejemplo, versiones beta). En la siguiente sección hablaremos sobre varias estrategias de implementación.

## Temas

- [Implementaciones in situ](#)
- [Implementaciones azul/verde](#)
- [Implementaciones de valores controlados](#)
- [Implementaciones lineales](#)
- [Implementaciones todo a la vez](#)

## Implementaciones in situ

En esta estrategia, la implementación se hace en consonancia con el cierre de la aplicación de cada instancia del grupo de implementación, se instala la última revisión de la aplicación y se inicia y valida la nueva versión de la aplicación. Puede utilizar un equilibrador de carga de modo que se cancele el registro de cada instancia durante su implementación y, a continuación, vuelva a ponerse en servicio una vez completada la implementación. Las implementaciones in situ pueden realizarse todas a la vez, con una interrupción del servicio o pueden realizarse como una actualización continua. AWS CodeDeploy y [AWS Elastic Beanstalk](#) ofrecen configuraciones de implementación de una en una, la mitad a la vez y todas a la vez. Estas mismas estrategias de implementación para implementaciones in situ están disponibles para las implementaciones azules/verdes.

## Implementaciones azul/verde

La implementación azul/verde, a veces denominada rojo/negro, es una técnica de publicación de aplicaciones que pasa el tráfico entre dos entornos idénticos que ejecutan diferentes versiones de la aplicación. Las implementaciones azul/verde le ayudan a minimizar el tiempo de inactividad durante las actualizaciones de las aplicaciones, lo que mitiga los riesgos relacionados con el tiempo de inactividad. Las implementaciones azul/verde le permiten lanzar una nueva versión (verde) de su

aplicación junto con la versión anterior (azul), y supervisar y probar la nueva versión antes de redirigir el tráfico a ella, revirtiendo la implementación si se detectan problemas.

## Implementaciones de valores controlados

El tráfico se desvía en dos incrementos. Una implementación de valores controlados es una estrategia azul/verde que es más reacia al riesgo, en la que se utiliza un enfoque gradual. Puede ser de dos etapas o lineal, en la que el nuevo código de la aplicación se implementa y se expone para probarlo y, una vez aceptado, se implementa en el resto del entorno o de manera lineal.

## Implementaciones lineales

En las implementaciones lineales el tráfico se desvía en incrementos iguales con el mismo número de minutos entre cada incremento. Puede elegir opciones lineales predefinidas que especifiquen el porcentaje de tráfico desviado en cada incremento y el número de minutos entre cada incremento.

## Implementaciones todo a la vez

En las implementaciones todo a la vez todo el tráfico se desvía del entorno original al entorno de reemplazo al mismo tiempo.

# Matriz de estrategias de implementación

En la siguiente matriz se enumeran las estrategias de implementación compatibles con [Amazon Elastic Container Service](#) (Amazon ECS), AWS Lambda y Amazon EC2/local.

- Amazon ECS es un servicio de orquestación completamente administrado.
- AWS Lambda le permite ejecutar código sin aprovisionar ni administrar servidores.
- Amazon EC2 le permite ejecutar una capacidad de computación segura y de tamaño variable en la nube.

	A	B	C	D
1	Matriz de estrategias de implementación	Amazon ECS	AWS Lambda	Amazon EC2/local
2	In situ	✓	✓	✓
3	Azul/verde	✓	✓	✓*
4	Valor controlado	✓	✓	X
5	Lineal	✓	✓	X
6	Todo a la vez	✓	✓	X

**Note**

La implementación azul/verde con EC2/local solo funciona con instancias de EC2.

## Estrategias de implementación de AWS Elastic Beanstalk

AWS Elastic Beanstalk admite los siguientes tipos de estrategias de implementación:

- Todo a la vez: realiza una implementación in situ en todas las instancias.

- Continua: divide las instancias en lotes e implementa en un lote cada vez.
- Continua con un lote adicional: divide las implementaciones en lotes, pero crea nuevas instancias de EC2 para el primer lote en lugar de implementarlo en instancias de EC2 existentes.
- Inmutable: si necesita realizar la implementación con una nueva instancia en lugar de usar una instancia existente.
- División de tráfico: realiza una implementación inmutable y, a continuación, reenvía un porcentaje de tráfico a las nuevas instancias durante un período de tiempo predeterminado. Si las instancias se mantienen en buen estado, reenvía todo el tráfico a instancias nuevas y cierra las instancias antiguas.



# Infraestructura como código

Un principio fundamental de DevOps es tratar la infraestructura de la misma manera que los desarrolladores tratan el código. El código de la aplicación tiene un formato y una sintaxis definidos. Si el código no se escribe de acuerdo con las reglas del lenguaje de programación, no se pueden crear aplicaciones. El código se almacena en un sistema de administración de versiones o control de código fuente que registra un historial de desarrollo, cambios y correcciones de errores del código. Cuando el código se compila o se integra en las aplicaciones, esperamos que se cree una aplicación coherente y que la compilación sea repetible y confiable.

La práctica de la infraestructura como código significa aplicar el mismo rigor de desarrollo del código de la aplicación al aprovisionamiento de la infraestructura. Todas las configuraciones deben definirse de forma declarativa y almacenarse en un sistema de control de código fuente como [AWS CodeCommit](#), lo mismo que el código de la aplicación. El aprovisionamiento, la orquestación y la implementación de la infraestructura también deben admitir el uso de la infraestructura como código.

La infraestructura se aprovisionaba tradicionalmente mediante una combinación de secuencias de comandos y procesos manuales. A veces, estas secuencias de comandos se almacenaban en sistemas de control de versiones o se documentaban paso a paso en archivos de texto o libros de ejecución. A menudo, la persona que escribe los libros de ejecución no es la misma persona que ejecuta las secuencias de comandos o sigue los libros de ejecución. Si estas secuencias de comandos o libros de ejecución no se actualizan con frecuencia, pueden convertirse en un obstáculo para las implementaciones. Esto hace que la creación de nuevos entornos no siempre sea repetible, fiable o coherente.

A diferencia de lo anterior, AWS proporciona una forma centrada en DevOps de crear y mantener la infraestructura. Similar a la forma en que los desarrolladores de software escriben el código de la aplicación, AWS proporciona servicios que permiten la creación, implementación y mantenimiento de la infraestructura de manera programática, descriptiva y declarativa. Estos servicios brindan rigor, claridad y fiabilidad. Los servicios de AWS que se analizan en este documento son fundamentales para una metodología de DevOps y constituyen la base de numerosos principios y prácticas de DevOps de AWS de nivel superior.

AWS ofrece los siguientes servicios para definir la infraestructura como código.

- [AWS CloudFormation](#)
- [AWS Cloud Development Kit \(AWS CDK\)](#)
- [Kit de desarrollo en la nube de AWS para Kubernetes](#)

# AWS CloudFormation

AWS CloudFormation es un servicio que permite a los desarrolladores crear recursos de AWS de forma ordenada y predecible. Los recursos se escriben en archivos de texto con formato JavaScript Object Notation (JSON) o Yet Another Markup Language (YAML). Las plantillas requieren una sintaxis y una estructura específicas que dependen de los tipos de recursos que se crean y administran. Usted crea sus recursos en JSON o YAML con cualquier editor de código como [AWS Cloud9](#), los registra en un sistema de control de versiones y, a continuación, CloudFormation crea los servicios especificados de manera segura y repetible.

Se implementa una plantilla de CloudFormation en el entorno de AWS como una pila. Puede administrar pilas a través de la consola de administración de AWS, la interfaz de línea de comandos de AWS o las API de AWS CloudFormation. Si necesita realizar cambios en los recursos que están ejecutándose en una pila, actualice la pila. Antes de realizar cambios en los recursos, puede generar un conjunto de cambios, que es un resumen de los cambios propuestos. Los conjuntos de cambios le permiten ver cómo afectan los cambios a sus recursos en ejecución, en especial para los recursos cruciales, antes de implementarlos.

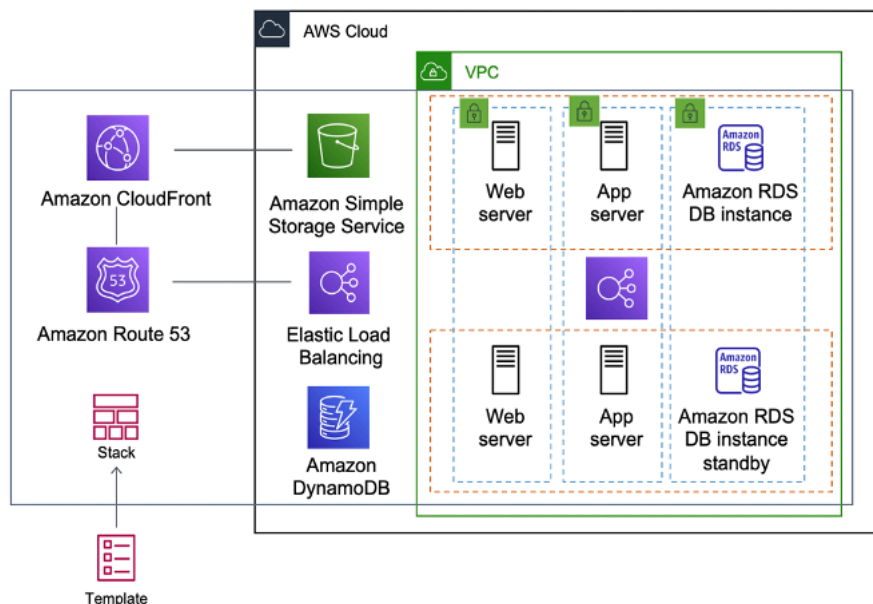


Figura 1: AWS CloudFormation crea un entorno completo (pila) a partir de un flujo de trabajo de plantilla

Puede usar una sola plantilla para crear y actualizar un entorno completo o plantillas independientes para administrar varias capas dentro de un entorno. Esto permite modular las plantillas y también proporciona una capa de gobernanza que es importante para muchas organizaciones.

Al crear o actualizar una pila en la consola, aparecen eventos que muestran el estado de la configuración. Si se produce un error, de forma predeterminada, la pila vuelve a su estado anterior. Amazon Simple Notification Service (Amazon SNS) proporciona notificaciones sobre eventos. Por ejemplo, puede utilizar Amazon SNS para realizar un seguimiento del curso de las operaciones de creación y eliminación de pilas mediante correo electrónico e integrar otros procesos de manera programática.

AWS CloudFormation facilita la organización e implementación de una colección de recursos de AWS, además de permitirle describir cualquier dependencia o transmitir parámetros especiales cuando se configura la pila.

Con las plantillas de CloudFormation, puede trabajar con un amplio conjunto de servicios de AWS, como Amazon S3, Auto Scaling, Amazon CloudFront, Amazon DynamoDB, Amazon EC2, Amazon ElastiCache, AWS Elastic Beanstalk, Elastic Load Balancing, IAM, AWS OpsWorks y Amazon VPC. Para obtener la lista más reciente de recursos compatibles, consulte la [Referencia de tipos de recursos y propiedades de AWS](#).

## Kit de desarrollo en la nube de AWS

El [AWS Cloud Development Kit \(AWS CDK\)](#) es un marco de desarrollo de software de código abierto que sirve para modelar y aprovisionar sus recursos destinados a aplicaciones en la nube mediante lenguajes de programación conocidos. AWS CDK permite que modele la infraestructura de la aplicación con TypeScript, Python, Java y .NET. Los desarrolladores pueden aprovechar su entorno de desarrollo integrado (IDE) existente, aprovechando herramientas como el autocompletado y la documentación en línea para acelerar el desarrollo de la infraestructura.

AWS CDK utiliza AWS CloudFormation en segundo plano para aprovisionar recursos de manera segura y repetible. Los componentes fijos son los componentes básicos del código CDK. Un componente fijo representa un componente de la nube y encapsula todo lo que AWS CloudFormation necesita para crear el componente. El AWS CDK incluye la [biblioteca de componentes fijos de AWS](#) que contiene componentes fijos que representan muchos servicios de AWS. Al combinar estos componentes, puede crear arquitecturas complejas de forma rápida y sencilla para su implementación en AWS.

# Kit de desarrollo en la nube de AWS para Kubernetes

El [kit de desarrollo en la nube de AWS para Kubernetes](#) (cdk8s) es un marco de desarrollo de software de código abierto para definir aplicaciones de Kubernetes mediante lenguajes de programación de uso general.

Una vez que haya definido su aplicación en un lenguaje de programación (en la fecha de publicación solo se admiten Python y TypeScript), cdk8s convertirá la descripción de su aplicación en YAML pre-Kubernetes. Este archivo YAML se puede consumir en cualquier clúster de Kubernetes que se ejecute en cualquier lugar. Como la estructura se define en un lenguaje de programación, puede utilizar las características enriquecidas que proporciona el lenguaje de programación. Puede utilizar la característica de abstracción del lenguaje de programación para crear su propio código repetitivo y reutilizarlo en todas las implementaciones.

# Automatización

Otra filosofía y práctica fundamentales de DevOps es la automatización. La automatización se centra en la preparación, la configuración, la implementación y el soporte de la infraestructura y las aplicaciones que se ejecutan en ella. Mediante el uso de la automatización, puede configurar entornos más rápidamente de manera estandarizada y repetible. La eliminación de los procesos manuales es clave para una estrategia de DevOps fructífera. Históricamente, la configuración del servidor y la implementación de la aplicación ha sido predominantemente un proceso manual. Cuando los entornos dejan de ser estándar, es difícil reproducir un entorno cuando surgen problemas.

El uso de la automatización es fundamental para aprovechar todos los beneficios de la nube. Internamente, AWS se basa en gran medida en la automatización para proporcionar las características principales de elasticidad y escalabilidad. Los procesos manuales son propensos a errores, poco fiables e inadecuados para respaldar un negocio ágil. Con frecuencia, una organización puede comprometer recursos altamente cualificados para proporcionar una configuración manual, cuando sería mejor dedicar el mejor a apoyar otras actividades más cruciales y de mayor valor dentro de la empresa.

Los entornos operativos modernos suelen depender de la automatización completa para eliminar la intervención manual o acceder a los entornos de producción. Esto incluye todas las versiones de software, configuración de máquinas, parches del sistema operativo, solución de problemas o corrección de errores. Se pueden usar muchos niveles de prácticas de automatización en conjunto para proporcionar un proceso automatizado de extremo a extremo de mayor nivel.

La automatización tiene los siguientes beneficios clave:

- Cambios rápidos
- Mejora de la productividad
- Configuraciones repetibles
- Entornos reproducibles
- Aprovechamiento de la elasticidad
- Aprovechamiento del escalado automático
- Pruebas automatizadas

La automatización es una piedra angular de los servicios de AWS y se admite internamente en todos los servicios, características y ofertas.

Temas

- [AWS OpsWorks](#)
- [AWS Elastic Beanstalk](#)

## AWS OpsWorks

[AWS OpsWorks](#) lleva los principios de DevOps incluso más lejos que AWS Elastic Beanstalk. Se puede considerar un servicio de administración de aplicaciones en lugar de simplemente un contenedor de aplicaciones. AWS OpsWorks proporciona aún más niveles de automatización con características adicionales como la integración con el software de administración de configuración (Chef) y la administración del ciclo de vida de la aplicación. Puede usar la administración del ciclo de vida de la aplicación para definir cuándo se preparan, configuran e implementan los recursos y cuando se cancela la implementación o se cierran los recursos.

Para mayor flexibilidad AWS OpsWorks le hace definir su aplicación en pilas configurables. También puede seleccionar pilas de aplicaciones predefinidas. Las pilas de aplicaciones contienen todo el aprovisionamiento de los recursos de AWS que su aplicación requiere, incluidos los servidores de aplicaciones, los servidores web, las bases de datos y los equilibradores de carga.

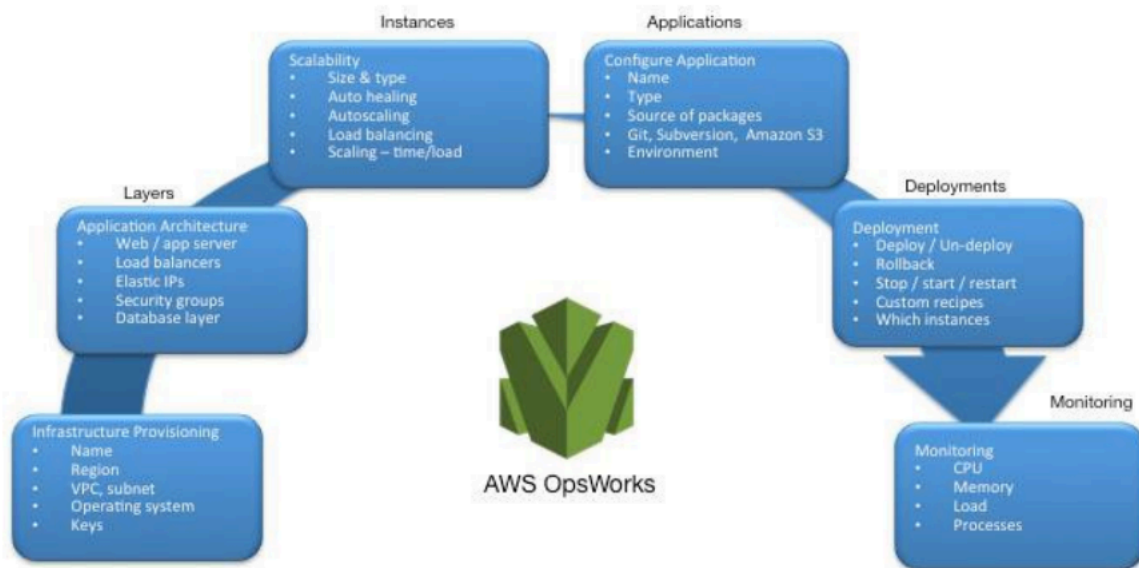


Figura 2: AWS OpsWorks con las características y la arquitectura de DevOps

Las pilas de aplicaciones se organizan en capas arquitectónicas para que las pilas se puedan mantener de forma independiente. Las capas podrían incluir, por ejemplo, nivel web, nivel de aplicación y nivel de base de datos. De forma inmediata, AWS OpsWorks también simplifica la configuración de grupos de Auto Scaling y equilibradores de carga de Elastic Load Balancing, lo que ilustra aún más el principio de automatización de DevOps. Al igual que AWS Elastic Beanstalk, AWS OpsWorks admite el control de versiones de aplicaciones, la implementación continua y la administración de configuración de la infraestructura.

AWS OpsWorks también es compatible con las prácticas de supervisión y registro de DevOps (que se tratan en la siguiente sección). Amazon CloudWatch proporciona soporte para la supervisión. Se registran todos los eventos del ciclo de vida y un registro de Chef independiente documenta las recetas de Chef que se ejecutan, junto con cualquier excepción.

## AWS Elastic Beanstalk

[AWS Elastic Beanstalk](#) es un servicio para implementar y escalar rápidamente aplicaciones web desarrollados con Java, .NET, PHP, Node.js, Python, Ruby, Go y Docker en servidores conocidos, como, por ejemplo, Apache, NGINX, Passenger e IIS.

Elastic Beanstalk es una abstracción añadida a Amazon EC2, Auto Scaling, y simplifica la implementación al proporcionar características adicionales como clonación, implementaciones azul/verde, interfaz de línea de comandos de Elastic Beanstalk (eb cli) e integración con AWS Toolkit for Visual Studio, Visual Studio Code, Eclipse y IntelliJ para aumentar la productividad de los desarrolladores.

# Monitorización y registro

La comunicación y la colaboración son fundamentales en una filosofía de DevOps. Para facilitar esto, la retroalimentación es fundamental. En AWS, la retroalimentación proviene de dos servicios principales: Amazon CloudWatch y AWS CloudTrail. Juntos, proporcionan una sólida infraestructura de supervisión, alertas y auditoría para que los desarrolladores y los equipos de operaciones puedan trabajar juntos de manera estrecha y transparente.

AWS proporciona los siguientes servicios para la supervisión y el registro:

## Temas

- [Amazon CloudWatch](#)
- [Alarmas de Amazon CloudWatch](#)
- [Amazon CloudWatch Logs](#)
- [Amazon CloudWatch Logs Insights](#)
- [Amazon CloudWatch Events](#)
- [Amazon EventBridge](#)
- [AWS CloudTrail](#)

## Amazon CloudWatch

Las métricas de Amazon CloudWatch recopilan automáticamente datos de los servicios de AWS, como instancias de Amazon EC2, volúmenes de Amazon EBS e instancias de base de datos de Amazon RDS. Estas métricas se pueden organizar como paneles y se pueden crear alarmas o eventos para desencadenar eventos o realizar acciones de Auto Scaling.

## Alarmas de Amazon CloudWatch

Puede configurar alarmas en función de las métricas recopiladas por Amazon CloudWatch Metrics. La alarma puede enviar una notificación al tema de Amazon Simple Notification Service (Amazon SNS) o iniciar acciones de Auto Scaling. Una alarma requiere un período (duración del tiempo para evaluar una métrica), un período de evaluación (número de los puntos de datos más recientes) y puntos de datos a alarma (número de puntos de datos dentro del período de evaluación).



# Amazon CloudWatch Logs

[Amazon CloudWatch Logs](#) es un servicio de agregación y supervisión de registros. AWS CodeBuild, CodeCommit, CodeDeploy y CodePipeline proporcionan integraciones con los registros de CloudWatch para poder supervisar todos los registros de forma centralizada. Además, los servicios mencionados anteriormente y otros servicios de AWS proporcionan una integración directa con CloudWatch.

Con CloudWatch Logs, puede:

- Consulta los datos de registro
- Supervisar registros de instancias de Amazon EC2
- Supervisar eventos registrados de AWS CloudTrail
- Definir la política de conservación

## Amazon CloudWatch Logs Insights

Amazon CloudWatch Logs Insights analiza sus registros y le permite realizar consultas y visualizaciones interactivas. Comprende varios campos de detección automática y formatos de registro de los registros JSON.

## Amazon CloudWatch Events

Amazon CloudWatch Events proporciona una secuencia de eventos de sistema casi en tiempo real que describen cambios en los recursos de AWS. Mediante reglas sencillas que puede configurar rápidamente, puede asignar los eventos y dirigirlos a uno o más flujos o funciones de destino. CloudWatch Events conoce los cambios operativos a medida que se producen. CloudWatch Events responde a estos cambios operativos y toma medidas correctoras según sea necesario, enviando mensajes para responder al entorno, activando funciones, realizando cambios y captando información de estado.

Con CloudWatch Events puede configurar reglas que le alertan de los cambios en los servicios de AWS e integrar estos eventos con otros sistemas de terceros mediante Amazon EventBridge. A continuación, se enumeran los servicios relacionados con AWS DevOps que se integran con CloudWatch Events.

- [Eventos de Application Auto Scaling](#)

- [Eventos de CodeBuild](#)
- [Eventos de CodeCommit](#)
- [Eventos de CodeDeploy](#)
- [Eventos de CodePipeline](#)

## Amazon EventBridge

Amazon CloudWatch Events y EventBridge son el mismo servicio subyacente y la misma API, pero EventBridge ofrece más características.

[Amazon EventBridge](#) es un bus de eventos sin servidor que permite integraciones entre servicios de AWS, software como servicio (SaaS) y sus aplicaciones. Además de crear aplicaciones basadas en eventos, EventBridge se puede usar para notificar sobre eventos de los servicios como CodeBuild, CodeDeploy, CodePipeline y CodeCommit.

## AWS CloudTrail

Para adoptar los principios de colaboración, comunicación y transparencia de DevOps, es importante entender quién lleva a cabo las modificaciones en su infraestructura. En AWS el servicio [AWS CloudTrail](#) proporciona esta transparencia. Todas las interacciones de AWS se gestionan a través de llamadas a la API de AWS que AWS CloudTrail supervisa y registra. Todos los archivos de registro generados se almacenan en el bucket de Amazon S3 que usted defina. Los archivos de registro se cifran mediante el [cifrado del lado del servidor \(SSE\) de Amazon S3](#). Se registran todas las llamadas a la API, tanto si proceden directamente de un usuario como si son en nombre de un usuario mediante un servicio de AWS. Numerosos grupos pueden beneficiarse de los registros de CloudTrail, incluidos los equipos de operaciones para el soporte, los equipos de seguridad para la gobernanza y los equipos financieros para la facturación.

# Comunicación y colaboración

Tanto si adopta la cultura de DevOps en su organización como si está en una etapa de comunicación de la transformación cultural de DevOps, la colaboración es una parte importante de su estrategia. En Amazon, nos hemos dado cuenta de que es necesario cambiar la mentalidad de los equipos y, por lo tanto, adoptamos el concepto del equipos de dos pizzas.

## Temas

- [Equipos de dos pizzas](#)

## Equipos de dos pizzas

«Tratamos de crear equipos que sean lo suficientemente pequeños como para alimentarlos con solo dos pizzas», dijo Bezos. «A eso lo llamamos la regla del equipo de dos pizzas».

Cuanto más pequeño sea el equipo, mejor será la colaboración. La colaboración también es muy importante porque las versiones de software avanzan más rápido que nunca. Y la capacidad de un equipo de ofrecer el software puede ser un factor diferenciador para su organización frente a la competencia. Imagine una situación en la que es necesario lanzar una nueva característica del producto o corregir un error; quiere que esto suceda lo más rápido posible para poder introducir la actualización en el mercado lo antes posible. Además tampoco desea que la transformación sea un proceso lento en lugar de un enfoque ágil cuando las oleadas de cambios comiencen a tener un impacto.

La comunicación entre los equipos también es fundamental: avanzamos hacia el modelo de responsabilidad compartida y comenzamos a salir del enfoque de desarrollo en silos. Esto trae el concepto de propiedad al equipo, lo que cambia su perspectiva para ver este proceso de principio a fin. Su equipo no debe pensar en sus entornos de producción como cajas negras donde no tienen visibilidad.

La transformación cultural también es crucial, ya que puede que le interese crear un equipo de DevOps común. La otra opción es tener uno o más miembros centrados en DevOps en su equipo. Ambos enfoques introducen la responsabilidad compartida en el equipo.

# Seguridad

Tanto si está pasando por una transformación de DevOps o implementando los principios de DevOps por primera vez, debe pensar en la seguridad como un elemento integrado en sus procesos de DevOps. Esto debería ser una preocupación transversal en todas las etapas de compilación, prueba e implementación.

Antes de hablar de la seguridad en DevOps en AWS, echemos un vistazo al modelo de responsabilidad compartida de AWS.

## Temas

- [Modelo de responsabilidad compartida de AWS](#)
- [Identity and Access Management](#)

## Modelo de responsabilidad compartida de AWS

Los asuntos relacionados con la seguridad son una responsabilidad compartida entre AWS y el cliente. A continuación se explican las diferentes partes del modelo de responsabilidad compartida:

- Responsabilidad de AWS en relación con la seguridad de la nube: AWS es responsable de proteger la infraestructura que ejecuta todos los servicios provistos en la nube de AWS. Esta infraestructura está compuesta por el hardware, el software, las redes y las instalaciones que ejecutan los servicios de la nube de AWS.
- Responsabilidad del cliente en relación con la seguridad en la nube: la responsabilidad del cliente estará determinada por los servicios de la nube de AWS que el cliente seleccione. Esto determina el alcance del trabajo de configuración a cargo del cliente como parte de sus responsabilidades de seguridad.

Este modelo compartido puede aliviar la carga operativa del cliente, ya que AWS opera, administra y controla los componentes desde el sistema operativo host y la capa de virtualización hasta los sistemas de seguridad física de las instalaciones en las que operan los servicios. Esto es fundamental en los casos en los que el cliente quiere entender la seguridad de sus entornos de creación.

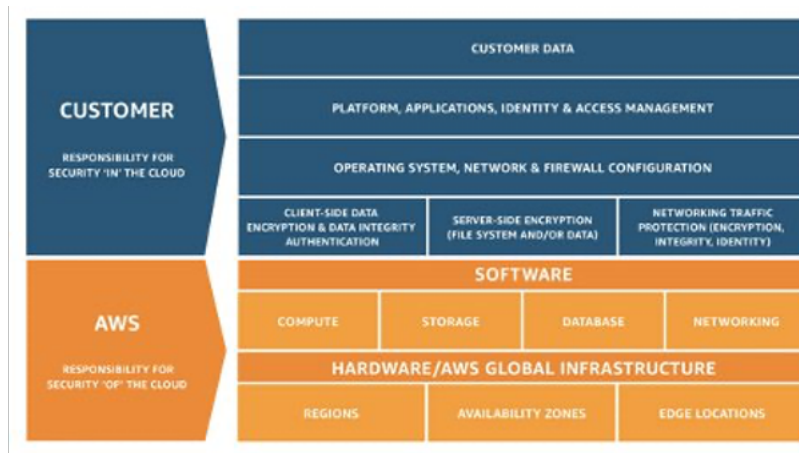


Figura 3: Modelo de responsabilidad compartida de AWS

## Identity and Access Management

[AWS Identity and Access Management \(IAM\)](#) define los controles y las políticas que se utilizan para administrar el acceso a los recursos de AWS. Con IAM, puede crear usuarios y grupos y definir permisos para varios servicios de DevOps.

Además de los usuarios, es posible que varios servicios también necesiten acceso a los recursos de AWS. Por ejemplo, su proyecto de CodeBuild puede necesitar acceso para almacenar imágenes de Docker en [Amazon Elastic Container Registry \(Amazon ECR\)](#) y permisos para escribir en Amazon ECR. Estos tipos de permisos se definen mediante un rol de tipo especial conocido como rol de servicio.

IAM es un componente de la infraestructura de seguridad de AWS. Con IAM, puede gestionar de forma centralizada los grupos, los usuarios, los roles de servicio y las credenciales de seguridad (por ejemplo, las contraseñas, las claves de acceso) y las políticas de permisos que controlan los servicios y recursos de AWS a los que pueden acceder los usuarios. La [política de IAM](#) le permite definir el conjunto de permisos. Esta política se puede adjuntar a un [rol](#), [usuario](#) o [servicio](#) para definir su permiso. También puede usar IAM para crear roles que se usan ampliamente dentro de la estrategia de DevOps que desee. En algunos casos, puede tener sentido crear [AssumeRole](#) programáticamente en lugar de obtener los permisos directamente. Cuando un servicio o usuario asume roles, se le otorgan credenciales temporales para acceder a un servicio al que normalmente no tiene acceso.

# Conclusión

Para que la transición a la nube sea fluida, eficiente y efectiva, las empresas tecnológicas deben adoptar los principios y las prácticas de DevOps. Estos principios están integrados en la plataforma de AWS. De hecho, constituyen la piedra angular de numerosos servicios de AWS, especialmente los de las ofertas de implementación y supervisión.

Comience por definir su infraestructura como código mediante el servicio AWS CloudFormation o AWS Cloud Development Kit (AWS CDK). A continuación, defina la forma en que sus aplicaciones van a utilizar la implementación continua con la ayuda de servicios como AWS CodeBuild, AWS CodeDeploy, AWS CodePipeline y AWS CodeCommit. A nivel de aplicación, utilice contenedores como AWS Elastic Beanstalk, Amazon Elastic Container Service (Amazon ECS) o Amazon Elastic Kubernetes Service (Amazon EKS) y AWS OpsWorks para simplificar la configuración de arquitecturas comunes. El uso de estos servicios también facilita la inclusión de otros servicios importantes, como Auto Scaling y Elastic Load Balancing. Por último, utilice una estrategia de supervisión de DevOps, como Amazon CloudWatch, y prácticas de seguridad sólidas, como IAM de AWS.

Con AWS como socio, sus principios de DevOps aportan agilidad a su empresa y organización de TI y aceleran su transición a la nube.

# Revisiones del documento

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

update-history-change

[Se ha restaurado la sección Colaboradores omitida.](#)

update-history-description

Se ha restaurado la sección Colaboradores omitida y se han hecho pequeños cambios de texto.

update-history-date

21 de noviembre de 2020

[Secciones actualizadas para incluir nuevos servicios](#)

Secciones actualizadas para incluir nuevos servicios

16 de octubre de 2020

[Publicación inicial](#)

Primera publicación del documento técnico

1 de diciembre de 2014

# Colaboradores

Entre los colaboradores de este documento, están las siguientes personas:

- Muhammad Mansoor, arquitecto de soluciones
- Ajit Zadgaonkar, líder mundial en tecnología, modernización
- Juan Lamadrid, arquitecto de soluciones
- Darren Ball, arquitecto de soluciones
- Rajeswari Malladi, arquitecto de soluciones
- Pallavi Nargund, arquitecto de soluciones
- Bert Zahniser, arquitecto de soluciones
- Abdullahi Olaoye, arquitecto de soluciones en la nube
- Mohamed Kiswani, director de desarrollo de software
- Tara McCann, arquitecto de soluciones jefe



# Avisos

Los clientes son responsables de realizar sus propias evaluaciones de la información contenida en este documento. Este documento: (a) solo tiene fines informativos, (b) representa las prácticas y las ofertas de productos vigentes de AWS, que están sujetas a cambios sin previo aviso, y (c) no crea ningún compromiso ni garantía de AWS y sus empresas afiliadas, proveedores o concesionarios de licencias. Los productos o servicios de AWS se proporcionan “tal cual”, sin garantías, representaciones ni condiciones de ningún tipo, ya sean explícitas o implícitas. Las responsabilidades y obligaciones de AWS en relación con sus clientes se rigen por los acuerdos de AWS, y este documento no modifica ni forma parte de ningún acuerdo entre AWS y sus clientes.

© 2020 Amazon Web Services, Inc. o sus empresas afiliadas. Todos los derechos reservados.