



Documento técnico de AWS

Introducción DevOps a AWS



Introducción DevOps a AWS: Documento técnico de AWS

Copyright © 2025 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 e introducción	i
Introducción	1
¿Tiene Well-Architected?	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
Implementación azul/verde	9
Implementación de valores controlados	10
Despliegue lineal	10
All-at-once despliegue	10
Matriz de estrategias de despliegue	11
AWS Elastic Beanstalk estrategias de despliegue	11
Infraestructura como código	13
CloudFormation	14
AWS Serverless Application Model	15
AWS Cloud Development Kit	16
Kit de desarrollo en la nube de AWS para Kubernetes	16
Kit de desarrollo en la nube de AWS para Terraform	16
API de control de nube de AWS	17
Automatizar y herramientas	18
AWS OpsWorks	19
AWS Elastic Beanstalk	20
EC2 Image Builder	20
AWS Proton	21
AWS Service Catalog	21
AWS Cloud9	22
AWS CloudShell	22
Amazon CodeGuru	22

Monitoreo y observabilidad	23
CloudWatch Métricas de Amazon	23
CloudWatch Alarmas Amazon	23
Amazon CloudWatch Logs	24
Información sobre Amazon CloudWatch Logs	24
CloudWatch Eventos de Amazon	24
Amazon EventBridge	25
AWS CloudTrail	25
Amazon DevOps Guru	26
AWS X-Ray	26
Servicio administrado por Amazon para Prometheus	27
Amazon Managed Grafana	27
Comunicación y colaboración	28
Seguridad	29
AWS Modelo de responsabilidad compartida	29
Identity and Access Management	30
Conclusión	32
Revisiones del documento	33
Colaboradores	34
Avisos	35
	xxxvi

Introducción DevOps a AWS

Fecha de publicación: 7 de abril de 2023 ([Revisões del documento](#))

Hoy más que nunca, las empresas emprenden su viaje de transformación digital para establecer conexiones más profundas con sus clientes y lograr un valor empresarial sostenible y duradero. Organizaciones de todas las formas y tamaños están revolucionando a sus competidores y entrando en nuevos mercados al innovar más rápido que nunca. Para estas organizaciones, es importante centrarse en la innovación y la disruptión del software, por lo que es fundamental agilizar su entrega de software. Las organizaciones que acortan el tiempo desde la idea hasta la producción, haciendo de la velocidad y la agilidad una prioridad, podrían ser las disruptoras del mañana.

Si bien hay varios factores a tener en cuenta para convertirse en el próximo disruptor digital, este documento técnico se centra en DevOps los servicios y funciones de la plataforma Amazon Web Services (AWS) que ayudarán a aumentar la capacidad de una organización para ofrecer aplicaciones y servicios a gran velocidad.

Introducción

DevOps es la combinación de filosofías culturales, prácticas de ingeniería y herramientas que aumentan la capacidad de una organización para ofrecer aplicaciones y servicios a gran velocidad y con mejor calidad. Con el tiempo, han surgido varias prácticas esenciales a la hora de DevOps adoptarlas: la integración continua (CI), la entrega continua (CD), la infraestructura como código (IaC) y la supervisión y el registro.

Este paper destaca AWS las capacidades que le ayudan a acelerar su DevOps viaje y cómo AWS los servicios pueden ayudar a eliminar el trabajo pesado indiferenciado asociado a la DevOps adaptación. También describe cómo desarrollar una capacidad de integración y entrega continuas sin administrar servidores ni construir nodos, y cómo usar la IaC para aprovisionar y administrar los recursos de la nube de manera coherente y repetible.

- Integración continua: práctica de desarrollo de software en la que los desarrolladores combinan periódicamente los cambios de código en un repositorio central, tras lo cual se ejecutan compilaciones y pruebas automatizadas.
- Entrega continua: práctica de desarrollo de software en la que los cambios de código se crean, prueban y preparan automáticamente para su lanzamiento a producción.

- La infraestructura como código: práctica en la que la infraestructura se aprovisiona y administra mediante técnicas de desarrollo de código y 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 del usuario final de sus productos.
- Comunicación y colaboración: se establecen prácticas para acercar a los equipos, crear flujos de trabajo y distribuir las responsabilidades correspondientes DevOps.
- Seguridad: debe ser una preocupación transversal. Sus canalizaciones de integración y entrega continuas (CI/CD) y los servicios relacionados deben estar protegidos y deben configurarse los permisos de control de acceso adecuados.

Un examen de cada uno de estos principios revela una estrecha relación con las ofertas disponibles en AWS

¿Usa Well-Architected?

El [AWS Well-Architected Framework](#) le ayuda a comprender las ventajas y desventajas de las decisiones que toma al crear sistemas en la nube. Los seis pilares del marco le permitirán aprender las prácticas recomendadas de arquitectura para diseñar y utilizar sistemas fiables, seguros, eficientes, rentables y sostenibles. Con la [herramienta AWS Well-Architected Tool](#), disponible de forma gratuita en la [Consola de administración de AWS](#), puede comparar sus cargas de trabajo con estas prácticas recomendadas respondiendo a una serie de preguntas para cada pilar.

Integración continua

La integración continua (CI) es una práctica de desarrollo de software en la que los desarrolladores combinan periódicamente los cambios de código en un repositorio de código central, tras lo cual se ejecutan compilaciones y pruebas automatizadas. La integración integrada ayuda a encontrar y corregir errores con mayor rapidez, a mejorar la calidad del software y a reducir el tiempo que se tarda en validar y publicar nuevas actualizaciones de software.

AWS ofrece los siguientes servicios para una integración continua:

Temas

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

AWS CodeCommit

[AWS CodeCommit](#) es un servicio de control de código fuente gestionado, seguro y altamente escalable que aloja repositorios de git privados. CodeCommit reduce la necesidad de utilizar su propio sistema de control de código fuente y no es necesario aprovisionar ni escalar hardware ni instalar, configurar y operar software. Se puede utilizar CodeCommit para almacenar cualquier cosa, desde código hasta binarios, y es compatible con la funcionalidad estándar de GitHub, lo que le permite funcionar sin problemas con las herramientas actuales basadas en Git. Tu equipo también puede usar las herramientas CodeCommit de código en línea para buscar, editar y colaborar en proyectos. AWS CodeCommit tiene varias ventajas:

- Colaboración: AWS CodeCommit está diseñada para el desarrollo colaborativo de software. Puedes asignar, ramificar y combinar fácilmente tu código, lo que te ayuda a mantener fácilmente el control de los proyectos de tu equipo. CodeCommit también es compatible con las solicitudes de extracción, que proporcionan un mecanismo para solicitar revisiones de código y discutir el código con los colaboradores.
- Cifrado: puedes transferir tus archivos de ida y vuelta AWS CodeCommit mediante HTTPS o SSH, según prefieras. Los repositorios también se cifran automáticamente en reposo mediante [AWS Key Management Service](#)(AWS KMS) mediante claves específicas del cliente.
- Control de acceso: AWS CodeCommit utiliza [AWS Identity and Access Management](#)(IAM) para controlar y supervisar quién puede acceder a sus datos, además de cómo, cuándo y dónde puede

acceder a ellos. CodeCommit también te ayuda a monitorizar tus repositorios a través [AWS CloudTrail](#) de [Amazon CloudWatch](#).

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

- Notificaciones y scripts personalizados: ahora puede recibir notificaciones de los eventos que afecten a sus repositorios. Las notificaciones se presentarán como notificaciones de [Amazon Simple Notification Service](#) (Amazon SNS). Cada notificación incluirá un mensaje de estado y un enlace a los recursos cuyo evento generó esa notificación. Además, con las señales del AWS CodeCommit repositorio, puede enviar notificaciones y crear webhooks HTTP con Amazon SNS o [AWS Lambda](#) invocar funciones en 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 necesita aprovisionar, administrar ni escalar sus propios servidores de compilación. CodeBuild puede utilizar GitHub Enterprise GitHub BitBucket AWS CodeCommit, Amazon S3 o Amazon S3 como proveedor de origen.

CodeBuild escala de forma continua y puede procesar múltiples compilaciones al mismo tiempo. CodeBuild ofrece varios entornos preconfigurados para varias versiones de Microsoft Windows y Linux. Los clientes también pueden traer 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 vista visual de cuántos casos de prueba se ejecutaron y cuántos se aprobaron o no. El proceso de creación también se puede ejecutar dentro de una [Amazon Virtual Private Cloud](#) (Amazon VPC), lo que puede resultar útil si los servicios de integración o las bases de datos se implementan dentro de una VPC.

AWS CodeArtifact

[AWS CodeArtifact](#)es un servicio de repositorio de artefactos totalmente gestionado que las organizaciones pueden utilizar para almacenar, publicar y compartir de forma segura los paquetes de software utilizados en su proceso de desarrollo de software. CodeArtifact se puede configurar para recuperar automáticamente los paquetes de software y las dependencias de los repositorios de artefactos públicos para que los desarrolladores tengan acceso a las versiones más recientes.

Los equipos de desarrollo de software confían cada vez más en los paquetes de código abierto para realizar tareas habituales en sus paquetes de aplicaciones. Se ha vuelto fundamental que los equipos de desarrollo de software mantengan el control sobre una versión concreta del software de código abierto para garantizar que el software esté libre de vulnerabilidades. Con CodeArtifact, puede configurar controles para hacer cumplir esto.

CodeArtifact funciona con los gestores de paquetes y herramientas de creación más utilizados, como Maven, Gradle, npm, yarn, twine y pip, lo que facilita su integración en los flujos de trabajo de desarrollo existentes.

Entrega continua

La entrega continua (CD) es una práctica de desarrollo de software en la que los cambios de código se preparan automáticamente para su lanzamiento a producción. La entrega continua, uno de los pilares del desarrollo de aplicaciones moderno, amplía la integración continua al implementar todos los cambios de código en un entorno de prueba o de producción una vez finalizada la fase de creación. Cuando se implementan correctamente, los desarrolladores siempre dispondrán de un artefacto de construcción listo para la implementación que haya pasado por un proceso de pruebas estandarizado.

La entrega continua permite a los desarrolladores automatizar las pruebas más allá de las pruebas unitarias para poder verificar las actualizaciones de las aplicaciones en múltiples dimensiones antes de distribuirlas a los clientes.

Estas pruebas pueden incluir pruebas de interfaz de usuario, pruebas de carga, pruebas de integración, pruebas de confiabilidad de API y más. Esto ayuda a los desarrolladores a validar las actualizaciones de forma más exhaustiva y a detectar problemas de forma preventiva. Al utilizar la nube, es fácil y rentable automatizar la creación y la replicación de varios entornos para realizar pruebas, algo que antes era difícil de realizar de forma local.

AWS ofrece los siguientes servicios de entrega continua:

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

Temas

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

AWS CodeDeploy

[AWS CodeDeploy](#) es un servicio de implementación totalmente gestionado que automatiza las implementaciones de software en una variedad de servicios informáticos, [AWS Fargate](#) como [Amazon Elastic Compute Cloud](#) EC2 (Amazon) y sus servidores locales. AWS Lambda AWS CodeDeploy le facilita el lanzamiento rápido de nuevas funciones, le ayuda a evitar el tiempo

de inactividad durante la implementación de las aplicaciones y gestiona la complejidad de la actualización de las aplicaciones. Se puede utilizar CodeDeploy para automatizar las implementaciones de software, lo que reduce la necesidad de realizar operaciones manuales propensas a errores. El servicio se amplía para adaptarse a sus necesidades de implementación.

CodeDeploy tiene varios beneficios que se alinean con el DevOps principio del despliegue continuo:

- Implementaciones automatizadas: automatiza CodeDeploy completamente las implementaciones de software, lo que le permite implementarlas de manera confiable y rápida.
- Control centralizado: CodeDeploy le permite iniciar y realizar un seguimiento fácilmente del estado de las implementaciones de sus aplicaciones durante el o el. Consola de administración de AWS AWS CLI CodeDeploy proporciona un informe detallado que le permite ver cuándo y dónde se implementó cada revisión de la aplicación. También puede crear notificaciones automáticas para recibir actualizaciones en tiempo real sobre sus implementaciones.
- Minimice el tiempo de inactividad: CodeDeploy ayuda a maximizar la disponibilidad de las aplicaciones durante el proceso de implementación del software. Introduce los cambios de forma gradual y realiza un seguimiento del estado de las aplicaciones de acuerdo con reglas configurables. Las implementaciones de software se pueden detener y revertir fácilmente si hay errores.
- Fácil de adoptar: CodeDeploy funciona con cualquier aplicación y ofrece la misma experiencia en diferentes plataformas e idiomas. Puede reutilizar fácilmente el código de configuración existente. CodeDeploy también se puede integrar con su proceso de lanzamiento de software existente o con su cadena de herramientas de entrega continua (por ejemplo,, AWS CodePipeline GitHub, Jenkins).

AWS CodeDeploy admite múltiples opciones de implementación. Para obtener más información, consulte la sección [Estrategias de despliegue](#) de este documento.

AWS CodePipeline

[AWS CodePipeline](#)es un servicio de entrega continua que puede utilizar para modelar, visualizar y automatizar los pasos necesarios para lanzar su software. Con él AWS CodePipeline, puede modelar el proceso de lanzamiento completo para crear su código, implementarlo en entornos de preproducción, probar su aplicación y lanzarla a producción. AWS CodePipeline a continuación, compila, prueba e implementa la aplicación de acuerdo con el flujo de trabajo definido cada vez que se produce un cambio de código. Puede integrar las herramientas de sus socios y sus propias

herramientas personalizadas en cualquier etapa del proceso de lanzamiento para crear una solución de entrega end-to-end continua.

AWS CodePipeline tiene varios beneficios que se alinean con el DevOps principio del despliegue continuo:

- Entrega rápida: AWS CodePipeline automatiza el proceso de publicación de software, lo que le permite lanzar rápidamente nuevas funciones para sus usuarios. Con CodePipeline él, puede modificar rápidamente los comentarios y ofrecer nuevas funciones a sus usuarios con mayor rapidez.
- Calidad mejorada: al automatizar sus procesos de creación, prueba y publicación, AWS CodePipeline le permite aumentar la velocidad y la calidad de las actualizaciones de software al ejecutar todos los cambios nuevos mediante un conjunto coherente de controles de calidad.
- Fácil de integrar: se AWS CodePipeline puede ampliar fácilmente para adaptarse a sus necesidades específicas. Puedes usar los complementos prediseñados o tus propios complementos personalizados en cualquier paso del proceso de lanzamiento. Por ejemplo, puedes extraer el código fuente GitHub, usar tu servidor de compilación Jenkins local, ejecutar pruebas de carga con un servicio de terceros o transferir la información de implementación a tu panel de operaciones personalizado.
- Flujo de trabajo configurable: AWS CodePipeline le permite modelar las diferentes etapas del proceso de lanzamiento de software mediante la interfaz de la consola AWS CLI[CloudFormation](#), o AWS SDKs. 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 despliegue definen la forma en que desea entregar el software. Las organizaciones siguen diferentes estrategias de despliegue en función de su modelo de negocio. Algunas optan por ofrecer un software totalmente probado, mientras que otras prefieren que sus usuarios envíen sus comentarios y les permitan evaluar las funciones que se encuentran en fase de desarrollo (como las versiones beta). En la siguiente sección, se analizan varias estrategias de implementación.

Implementaciones in situ

En esta estrategia, se detiene la versión anterior de la aplicación en cada recurso informático, se instala la aplicación más reciente y se inicia y valida la nueva versión de la aplicación. Esto permite que las implementaciones de las aplicaciones se realicen con una perturbación mínima de la infraestructura subyacente. Con una implementación local, puede implementar su aplicación sin crear una nueva infraestructura; sin embargo, la disponibilidad de la aplicación puede verse afectada durante estas implementaciones. Este enfoque también minimiza los costos de infraestructura y los gastos generales de administración asociados con la creación de nuevos recursos. 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 all-at-once, suponiendo una interrupción del servicio, o como una actualización progresiva. AWS CodeDeploy y [AWS Elastic Beanstalk](#) ofrecen configuraciones de implementación one-at-a-time para, y half-at-a-time all-at-once

Implementación azul/verde

La implementación [azul/verde, a veces denominada implementación red/black deployment](#), es a technique for releasing applications by shifting traffic between two identical environments running differing versions of the application. Blue/green

Las implementaciones azul/verde le permiten lanzar una nueva versión (verde) de la aplicación junto con la versión anterior (azul), y supervisar y probar la nueva versión antes de redirigir el tráfico a ella, lo que reduce la probabilidad de detectar problemas.

Implementación de valores controlados

El objetivo de un [despliegue canario](#) es reducir el riesgo de implementar una nueva versión que afecte a la carga de trabajo. El método implementará la nueva versión de forma gradual, haciéndola visible para los nuevos usuarios de forma lenta. A medida que vaya adquiriendo confianza en la implementación, la desplegará para reemplazar la versión actual en su totalidad.

Despliegue lineal

El despliegue lineal significa que el tráfico se desplaza en incrementos iguales con un número igual 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.

All-at-once despliegue

All-at-once despliegue significa que todo el tráfico se desplaza del entorno original al entorno de reemplazo de una sola vez.

Matriz de estrategias de despliegue

En la siguiente matriz se enumeran las estrategias de implementación compatibles para [Amazon Elastic Container Service](#) (Amazon ECS) y Amazon EC2 /on-premises. AWS Lambda

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

Estrategia de implementación	Amazon ECS	AWS Lambda	Amazon EC2 / local
In situ	✓	✓	✓
Azul/verde	✓	✓	✓*
Valor controlado	✓	✓	X
Lineal	✓	✓	X
Un ll-at-once	✓	✓	X

 Note

Blue/green deployment with EC2/on-premises solo funciona con EC2 instancias.

AWS Elastic Beanstalk estrategias de despliegue

[AWS Elastic Beanstalk](#) admite los siguientes tipos de estrategias de despliegue:

- A ll-at-once Realiza el despliegue in situ en todas las instancias.
- Rolling divide las instancias en lotes y las despliega en un lote a la vez.
- La integración con un lote adicional divide las implementaciones en lotes, pero para el primer lote crea nuevas EC2 instancias en lugar de implementarlas en las instancias existentes. EC2

- Inmutable si necesita realizar la implementación con una instancia nueva en lugar de usar una instancia existente.
- La división del tráfico Realiza un despliegue inmutable y, a continuación, reenvía el porcentaje del tráfico a las nuevas instancias durante un período de tiempo predeterminado. Si las instancias se mantienen en buen estado, reenvía todo el tráfico a las instancias nuevas y cierra las antiguas.

Infraestructura como código

Un principio fundamental 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 está escrito 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 del desarrollo del código, los cambios y las correcciones de errores. 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 fiable.

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

Tradicionalmente, la infraestructura se aprovisionaba mediante una combinación de scripts y procesos manuales. A veces, estos scripts 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 estos scripts o sigue los libros de ejecución. Si estos scripts o manuales de ejecución no se actualizan con frecuencia, pueden convertirse en un éxito en las implementaciones. Esto hace que la creación de nuevos entornos no siempre sea repetible, fiable o coherente.

Por el contrario, AWS proporciona una forma DevOps centrada de crear y mantener la infraestructura. Similar a la forma en que los desarrolladores de software escriben el código de las aplicaciones, AWS proporciona servicios que permiten la creación, el despliegue y el mantenimiento de la infraestructura de forma programática, descriptiva y declarativa. Estos servicios proporcionan rigor, claridad y fiabilidad. Los AWS servicios analizados en este paper son fundamentales para una DevOps metodología y constituyen la base de numerosos AWS DevOps principios y prácticas de nivel superior.

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

Servicios

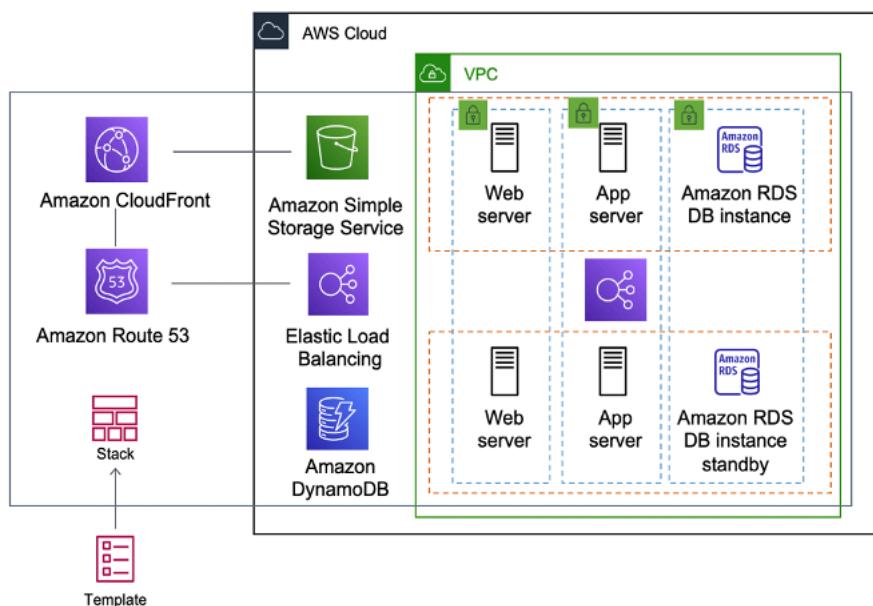
- [CloudFormation](#)
- [AWS Serverless Application Model](#)

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

CloudFormation

AWS CloudFormation es un servicio que permite a los desarrolladores crear AWS recursos de forma ordenada y predecible. Los recursos se escriben en archivos de texto en formato JSON o YAML. Las plantillas requieren una sintaxis y una estructura específicas que dependen de los tipos de recursos que se crean y administran. Puedes crear tus recursos en JSON o YAML con cualquier editor de código, por ejemplo [AWS Cloud9](#), incorporarlos a un sistema de control de versiones y, a continuación, CloudFormation compilar los servicios especificados de forma segura y repetible.

Una CloudFormation plantilla se implementa en el AWS entorno como una pila. Puede administrar las pilas a través de Consola de administración de AWS, AWS Command Line Interface, o CloudFormation APIs. Si necesita realizar cambios en los recursos en ejecución de 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 los cambios pueden afectar a sus recursos en ejecución, especialmente en el caso de los recursos críticos, antes de implementarlos.



AWS CloudFormation crear un entorno completo (pila) a partir de una 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 un nivel de gobierno que es importante para muchas organizaciones.

Al crear o actualizar una pila en la CloudFormation consola, se muestran los eventos que muestran el estado de la configuración. Si se produce un error, de forma predeterminada, la pila vuelve a su estado anterior. Amazon SNS proporciona notificaciones sobre eventos. Por ejemplo, puede usar Amazon SNS para realizar un seguimiento del progreso de creación y eliminación de pilas mediante el correo electrónico e integrarlo con otros procesos mediante programación.

AWS CloudFormation facilita la organización y el despliegue de un conjunto de AWS recursos y le permite describir cualquier dependencia o introducir parámetros especiales cuando se configura la pila.

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

AWS Serverless Application Model

El [AWS Serverless Application Model](#)(AWS SAM) es un marco de código abierto que puede utilizar para crear [aplicaciones sin servidor](#). AWS

AWS SAM se integra con otros AWS servicios, por lo que la creación de aplicaciones sin servidor AWS SAM ofrece las siguientes ventajas:

- Configuración de implementación única: AWS SAM facilita la organización de los componentes y recursos relacionados y el funcionamiento en una sola pila. Puede utilizarla AWS SAM para compartir la configuración (como la memoria y los tiempos de espera) entre los recursos e implementar todos los recursos relacionados juntos como una sola entidad versionada.
- Extensión de CloudFormation: debido a que AWS SAM es una extensión de CloudFormation, obtiene las capacidades de implementación confiables de CloudFormation. Puede definir los recursos utilizándolos CloudFormation en su AWS SAM plantilla.
- Mejores prácticas integradas: puede utilizarlas AWS SAM para definir e implementar su IaC. Esto le permite utilizar y aplicar las mejores prácticas, como las revisiones de código.

AWS Cloud Development Kit (AWS CDK)

[AWS Cloud Development Kit \(AWS CDK\)](#) Se trata de un marco de desarrollo de software de código abierto para modelar y aprovisionar los recursos de sus aplicaciones en la nube utilizando lenguajes de programación conocidos. AWS CDK le permite modelar la infraestructura de aplicaciones mediante Python TypeScript, Java y .NET. Los desarrolladores pueden aprovechar su entorno de desarrollo integrado (IDE) existente, utilizando herramientas como el autocompletado y la documentación en línea para acelerar el desarrollo de la infraestructura.

AWS CDK se utiliza CloudFormation en segundo plano para aprovisionar recursos de forma segura y repetible. Las construcciones son los componentes básicos del código CDK. Una construcción representa un componente de la nube y encapsula todo lo CloudFormation necesario para crear el componente. AWS CDK Incluye la [biblioteca AWS Construct](#), que contiene construcciones que representan muchos AWS servicios. Al combinar componentes, puede crear arquitecturas complejas de forma rápida y sencilla para su implementación. AWS

Kit de desarrollo en la nube de AWS para Kubernetes

El [AWS Cloud Development Kit para Kubernetes](#) es un marco de desarrollo de software de código abierto para definir las aplicaciones de Kubernetes mediante lenguajes de programación de uso general.

Una vez que haya definido la aplicación en un lenguaje de programación (a partir de la fecha de esta publicación, solo TypeScript son compatibles con Python), cdk8s convertirá la descripción de la aplicación en un formato YAML anterior a Kubernetes. A continuación, cualquier clúster de Kubernetes que se ejecute en cualquier lugar podrá consumir este archivo YAML. Como la estructura está definida en un lenguaje de programación, puedes usar las numerosas funciones que ofrece el lenguaje de programación. Puede utilizar la función de abstracción del lenguaje de programación para crear su propio código repetitivo y reutilizarlo en todas las implementaciones.

Kit de desarrollo en la nube de AWS para Terraform

Basado en la [biblioteca JSII](#) de código abierto, [CDK for Terraform \(CDKTF\)](#) le permite escribir configuraciones de Terraform en C#, Python, Java o Go de su elección y, aun así TypeScript, beneficiarse del ecosistema completo de proveedores y módulos de Terraform. Puede importar cualquier proveedor o módulo existente del registro de Terraform a su aplicación, y CDKTF generará clases de recursos con las que podrá interactuar en el lenguaje de programación de destino.

Con el CDKTF, los desarrolladores pueden configurar su IaC sin tener que cambiar el contexto de su lenguaje de programación habitual y utilizar las mismas herramientas y sintaxis para aprovisionar los recursos de infraestructura de forma similar a la lógica empresarial de la aplicación. Los equipos pueden colaborar con una sintaxis conocida y, al mismo tiempo, utilizar la potencia del ecosistema de Terraform y desplegar sus configuraciones de infraestructura a través de los canales de despliegue de Terraform establecidos.

API de control de nube de AWS

[API de control de nube de AWS](#)es una nueva AWS capacidad que introduce un conjunto común de funciones de creación, lectura, actualización, eliminación y lista (CRUDL) APIs para ayudar a los desarrolladores a gestionar su infraestructura de nube de forma sencilla y coherente. La API de control de nube común APIs permite a los desarrolladores administrar de manera uniforme el ciclo de vida de AWS y los servicios de terceros.

Como desarrollador, es posible que prefiera simplificar la forma en que administra el ciclo de vida de todos sus recursos. Puedes usar el modelo uniforme de configuración de recursos de la API de Cloud Control con un formato predefinido para estandarizar la configuración de los recursos de la nube. Además, te beneficiarás de un comportamiento uniforme de la API (elementos de respuesta y errores) a la vez que administras tus recursos.

Por ejemplo, te resultará fácil depurar los errores durante las operaciones de CRUDL mediante códigos de error uniformes que aparecen en la API de Cloud Control y que son independientes de los recursos con los que operas. Con la API de Cloud Control, también te resultará sencillo configurar las dependencias entre recursos. Además, ya no tendrás que crear y mantener código personalizado en herramientas de varios proveedores ni utilizar recursos de terceros APIs a AWS la vez.

Automatizar y herramientas

Otra filosofía y práctica fundamentales DevOps es la automatización. La automatización se centra en la instalación, la configuración, el despliegue y el soporte de la infraestructura y las aplicaciones que se ejecutan en ella. Al utilizar la automatización, puede configurar los entornos con mayor rapidez de forma estandarizada y repetible. La eliminación de los procesos manuales es clave para el éxito de una DevOps estrategia. Históricamente, la configuración del servidor y el despliegue de las aplicaciones han sido predominantemente un proceso manual. Los entornos dejan de ser estándares y 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 depende en gran medida de la automatización para proporcionar las características principales de elasticidad y escalabilidad.

Los procesos manuales son propensos a errores, poco fiables e inadecuados para respaldar una empresa ágil. Con frecuencia, una organización puede utilizar recursos altamente cualificados para realizar la configuración manual, cuando podría emplearse mejor el tiempo en apoyar otras actividades más críticas y de mayor valor dentro de la empresa.

Los entornos operativos modernos suelen depender de la automatización total para eliminar la intervención manual o el acceso a los entornos de producción. Esto incluye todas las versiones de software, la configuración de la máquina, la aplicación de parches al sistema operativo, la solución de problemas o la corrección de errores. Se pueden utilizar varios niveles de prácticas de automatización en conjunto para proporcionar un proceso end-to-end automatizado de mayor nivel.

La automatización tiene las siguientes ventajas clave:

- Cambios rápidos
- Mejora de la productividad
- Configuraciones repetibles
- Entornos reproducibles
- Elasticidad
- Escalado automático
- Pruebas automatizadas

La automatización es la piedra angular de AWS los servicios y cuenta con el respaldo interno de todos los servicios, funciones y ofertas.

Temas

- [AWS OpsWorks](#)
- [AWS Elastic Beanstalk](#)
- [EC2 Image Builder](#)
- [AWS Proton](#)
- [AWS Service Catalog](#)
- [AWS Cloud9](#)
- [AWS CloudShell](#)
- [Amazon CodeGuru](#)

AWS OpsWorks

[AWS OpsWorks](#) lleva los principios de DevOps incluso más allá AWS Elastic Beanstalk. Puede considerarse un servicio de administración de aplicaciones y no simplemente un contenedor de aplicaciones. OpsWorks proporciona incluso más niveles de automatización, con funciones adicionales, como la integración con el software de gestión de la configuración (Chef) y la gestión del ciclo de vida de las aplicaciones. Puede utilizar la gestión del ciclo de vida de las aplicaciones para definir cuándo se instalan, configuran, despliegan, desimplementan o finalizan los recursos.

Para mayor flexibilidad AWS OpsWorks , defina su aplicación en pilas configurables. También puede seleccionar pilas de aplicaciones predefinidas. Los paquetes de aplicaciones contienen todo el aprovisionamiento de los recursos de AWS que necesita su aplicación, incluidos los servidores de aplicaciones, los servidores web, las bases de datos y los平衡adores de carga.

Las pilas de aplicaciones se organizan en capas arquitectónicas para que las pilas se puedan mantener de forma independiente. Las capas de ejemplo podrían incluir el nivel web, el nivel de aplicación y el nivel de base de datos. Desde el primer momento, AWS OpsWorks también simplifica la configuración de los grupos de [Auto Scaling de AWS](#) y los balanceadores de [carga de Elastic Load Balancing](#) (ELB), lo que ilustra aún más el principio de la DevOps automatización. Al igual que AWS Elastic Beanstalk OpsWorks , AWS admite el control de versiones de aplicaciones, la implementación continua y la administración de la configuración de la infraestructura



OpsWorks que muestra DevOps las características y la arquitectura

AWS OpsWorks también es compatible con las DevOps prácticas de monitoreo y registro (que se describen en la siguiente sección). El soporte de monitoreo lo proporciona Amazon CloudWatch. Todos los eventos del ciclo de vida se registran y un registro de Chef independiente documenta todas las recetas de Chef que se estén ejecutando, con cualquier excepción.

AWS Elastic Beanstalk

[AWS Elastic Beanstalk](#) es un servicio para implementar y escalar rápidamente aplicaciones web desarrolladas 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 superior a EC2 Amazon, Amazon EC2 Auto Scaling, y simplifica la implementación al ofrecer funciones adicionales como la clonación, las blue/green implementaciones, la [interfaz de línea de comandos \(EB CLI\) de Elastic Beanstalk](#) y la integración con el kit de herramientas de AWS para Visual Studio, Visual [Studio Code, Eclipse e IntelliJ](#) para aumentar la productividad de los desarrolladores.

EC2 Image Builder

[EC2 Image Builder](#) es un AWS servicio totalmente gestionado que le ayuda a automatizar la creación, el mantenimiento, la validación, el uso compartido y la implementación de AMI

personalizadas, seguras y personalizadas para up-to-date Linux o Windows. EC2 Image Builder también se puede utilizar para crear imágenes de contenedores. Puedes usar el Consola de administración de AWS AWS CLI, el o APIs para crear imágenes personalizadas en tu AWS cuenta.

EC2 Image Builder reduce significativamente el esfuerzo de mantener up-to-date las imágenes seguras al proporcionar una interfaz gráfica sencilla, automatización integrada y ajustes AWS de seguridad proporcionados. Con EC2 Image Builder, no hay pasos manuales para actualizar una imagen ni tienes que crear tu propia canalización de automatización.

AWS Proton

[AWS Proton](#)permite a los equipos de plataformas conectar y coordinar todas las diferentes herramientas que sus equipos de desarrollo necesitan para el aprovisionamiento de infraestructuras, la implementación de código, la supervisión y las actualizaciones. AWS Proton permite una infraestructura automatizada como el aprovisionamiento de código y el despliegue de aplicaciones basadas en contenedores y sin servidor.

AWS Proton permite a los equipos de plataformas definir su infraestructura y sus herramientas de despliegue, al tiempo que proporciona a los desarrolladores una experiencia de autoservicio para obtener infraestructura e implementar código. De este AWS Proton modo, los equipos de plataformas proporcionan recursos compartidos y definen las pilas de aplicaciones, incluidas CI/CD las canalizaciones y las herramientas de observación. A continuación, puede gestionar qué funciones de infraestructura e implementación están disponibles para los desarrolladores.

AWS Service Catalog

[AWS Service Catalog](#)permite a las organizaciones crear y administrar catálogos de servicios de TI AWS aprobados. Estos servicios de TI pueden incluir desde imágenes de máquinas virtuales, servidores, software, bases de datos, etc., hasta arquitecturas de aplicaciones completas de varios niveles. AWS Service Catalog le permite gestionar de forma centralizada los servicios, las aplicaciones, los recursos y los metadatos de TI implementados para lograr una gobernanza uniforme de sus plantillas de IaC.

Con ello AWS Service Catalog, puede cumplir con sus requisitos de conformidad y, al mismo tiempo, asegurarse de que sus clientes puedan implementar rápidamente los servicios de TI aprobados que necesitan. Los usuarios finales pueden implementar rápidamente solo los servicios de TI aprobados que necesitan, de acuerdo con las limitaciones establecidas por su organización.

AWS Cloud9

[AWS Cloud9](#) es un IDE basado en la nube que le permite escribir, ejecutar y depurar el código con solo un navegador. Incluye un editor de código, un depurador y un terminal. AWS Cloud9 viene preempaquetado con herramientas esenciales para los lenguajes de programación populares JavaScript, incluidos Python, PHP y más, por lo que no necesita instalar archivos ni configurar su máquina de desarrollo para iniciar nuevos proyectos. Como su AWS Cloud9 IDE está basado en la nube, puede trabajar en sus proyectos desde su oficina, casa o cualquier lugar mediante una máquina conectada a Internet.

AWS CloudShell

[AWS CloudShelles](#) un shell basado en un navegador que facilita la administración, la exploración y la interacción con sus recursos de forma segura. AWS AWS CloudShell está preautenticado con las credenciales de la consola. Las herramientas de desarrollo y operaciones más comunes vienen preinstaladas, por lo que no es necesario instalar ni configurar el software en el equipo local.

Amazon CodeGuru

[Amazon CodeGuru](#) es una herramienta para desarrolladores que proporciona recomendaciones inteligentes para mejorar la calidad del código e identificar las líneas de código más caras de una aplicación. Intégrela CodeGuru en su flujo de trabajo de desarrollo de software actual para automatizar las revisiones del código durante el desarrollo de la aplicación y monitorizar continuamente el rendimiento de la aplicación durante la producción, además de ofrecer recomendaciones y pistas visuales sobre cómo mejorar la calidad del código y el rendimiento de las aplicaciones y reducir los costes generales. CodeGuru tiene dos componentes:

- Amazon CodeGuru Reviewer: [Amazon CodeGuru Reviewer](#) es un servicio de revisión de código automatizado que identifica los defectos críticos y las desviaciones de las prácticas recomendadas de codificación para el código de Java y Python. Escanea las líneas de código incluidas en una solicitud de extracción y proporciona recomendaciones inteligentes basadas en los estándares aprendidos en los principales proyectos de código abierto, así como en el código base de Amazon.
- Amazon CodeGuru Profiler: [Amazon CodeGuru Profiler](#) analiza el perfil de tiempo de ejecución de la aplicación y proporciona recomendaciones y visualizaciones inteligentes que guían a los desarrolladores sobre cómo mejorar el rendimiento de las partes más relevantes de su código.

Monitoreo y observabilidad

La comunicación y la colaboración son fundamentales en una DevOps filosofía. Para facilitar esto, la retroalimentación es fundamental. Estos comentarios los proporciona nuestro conjunto de servicios de monitoreo y observabilidad.

AWS proporciona los siguientes servicios de monitoreo y registro:

Temas

- [CloudWatch Métricas de Amazon](#)
- [CloudWatch Alarmas Amazon](#)
- [Amazon CloudWatch Logs](#)
- [Información sobre Amazon CloudWatch Logs](#)
- [CloudWatch Eventos de Amazon](#)
- [Amazon EventBridge](#)
- [AWS CloudTrail](#)
- [Amazon DevOps Guru](#)
- [AWS X-Ray](#)
- [Servicio administrado por Amazon para Prometheus](#)
- [Amazon Managed Grafana](#)

CloudWatch Métricas de Amazon

[CloudWatch Las métricas de Amazon](#) recopilan automáticamente datos de AWS servicios como las EC2 instancias de Amazon, los volúmenes de Amazon EBS y las instancias de bases de datos (DB) de Amazon RDS. Luego, estas métricas se pueden organizar como paneles y se pueden crear alarmas o eventos para activar eventos o realizar acciones de Amazon EC2 Auto Scaling.

CloudWatch Alarmas Amazon

Puedes configurar las alarmas mediante las alarmas de [Amazon CloudWatch](#) en función de las métricas recopiladas por Amazon CloudWatch Metrics. A continuación, la alarma puede enviar una

notificación al tema de Amazon SNS o iniciar acciones de Amazon EC2 Auto Scaling. Una alarma requiere un período (tiempo que se tarda en evaluar una métrica), un período de evaluación (número de puntos de datos más recientes) y puntos de datos que generar una 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 CloudWatch los registros para que todos los registros se puedan monitorear de forma centralizada. Además, varios otros servicios proporcionan una integración directa con AWS CloudWatch los servicios mencionados anteriormente.

Con CloudWatch Logs puedes:

- Consulta tus datos de registro
- Supervise los registros de las EC2 instancias de Amazon
- AWS CloudTrail Supervise los eventos registrados
- Defina la política de retención de registros

Información sobre Amazon CloudWatch Logs

Amazon CloudWatch Logs Insights escanea sus registros y le permite realizar consultas y visualizaciones interactivas. Comprende varios formatos de registro y descubre automáticamente los campos de los registros JSON.

CloudWatch Eventos de Amazon

[Amazon CloudWatch Events](#) ofrece una transmisión casi en tiempo real de los eventos del sistema que describen los cambios en AWS los recursos. Mediante reglas sencillas que puede configurar rápidamente, puede asignar los eventos y dirigirlos a uno o más flujos o funciones de destino.

CloudWatch Los eventos se dan cuenta de los cambios operativos a medida que se producen. CloudWatch Events responde a estos cambios operativos y toma las medidas correctivas necesarias, mediante el envío de mensajes en respuesta al entorno, la activación de funciones, la realización de cambios y la captura de información de estado.

Puedes configurar reglas en Amazon CloudWatch Events para avisarte de cambios en los AWS servicios e integrar estos eventos con otros sistemas de terceros que utilizan Amazon EventBridge. Los siguientes son los servicios AWS DevOps relacionados que se integran con CloudWatch Events.

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

Amazon EventBridge



Note

Amazon CloudWatch Events y EventBridge son el mismo servicio y API subyacentes, sin embargo, EventBridge proporcionan más funciones.

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

AWS CloudTrail

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

Amazon DevOps Guru

[Amazon DevOps Guru](#) es un servicio basado en el aprendizaje automático (ML) diseñado para facilitar la mejora del rendimiento operativo y la disponibilidad de una aplicación. DevOps Guru ayuda a detectar comportamientos que se desvían de los patrones operativos normales, para que pueda identificar los problemas operativos mucho antes de que afecten a sus clientes.

DevOps Guru utiliza modelos de aprendizaje automático basados en los años de excelencia AWS operativa de Amazon.com para ayudar a identificar el comportamiento anómalo de las aplicaciones (por ejemplo, el aumento de la latencia, las tasas de error, la escasez de recursos, etc.) y detectar problemas críticos que podrían provocar posibles interrupciones del servicio o interrupciones del servicio.

Cuando DevOps Guru identifica un problema crítico, ahorra tiempo de depuración al obtener información relevante y específica de una gran cantidad de fuentes de datos y envía automáticamente una alerta y proporciona un resumen de las anomalías relacionadas y el contexto de cuándo y dónde se produjo el problema.

AWS X-Ray

[AWS X-Ray](#)ayuda a los desarrolladores a analizar y depurar las aplicaciones distribuidas y de producción, como las creadas con una arquitectura de microservicios. Con X-Ray, puede comprender el rendimiento de su aplicación y sus servicios subyacentes para identificar y solucionar la causa raíz de los problemas y errores de rendimiento. X-Ray proporciona una end-to-end vista de las solicitudes a medida que se desplazan por la aplicación y muestra un mapa de los componentes subyacentes de la aplicación. X-Ray le permite:

- Cree un mapa de servicios: al rastrear las solicitudes realizadas a sus aplicaciones, X-Ray puede crear un mapa de los servicios utilizados por su aplicación. Esto le proporciona una vista de las conexiones entre los servicios de su aplicación y le permite crear un árbol de dependencias, detectar la latencia o los errores al trabajar en distintas zonas o regiones de AWS disponibilidad, concentrarse en los servicios que no funcionan según lo esperado, etc.
- Identifique errores y errores: X-Ray puede resaltar automáticamente los errores o errores en el código de su aplicación al analizar el código de respuesta de cada solicitud realizada a su aplicación. Esto permite depurar fácilmente el código de la aplicación sin necesidad de reproducir el error o error.

- Cree sus propias aplicaciones de análisis y visualización: X-Ray proporciona un conjunto de consultas APIs que puede utilizar para crear sus propias aplicaciones de análisis y visualización que utilizan los datos que graba X-Ray.

Servicio administrado por Amazon para Prometheus

[Amazon Managed Service for Prometheus](#) es un servicio de monitorización de métricas sin servidor compatible con Prometheus de código abierto, que le facilita la supervisión y las alertas de forma segura en entornos de contenedores. Amazon Managed Service for Prometheus reduce el trabajo pesado necesario para empezar a monitorizar las aplicaciones en Amazon Elastic Kubernetes Service, Amazon Elastic Container AWS Fargate Service y los clústeres de Kubernetes autogestionados.

Amazon Managed Grafana

[Amazon Managed Grafana](#) es un servicio totalmente gestionado con visualizaciones de datos completas e interactivas que ayudan a los clientes a analizar, monitorear y generar alarmas sobre las métricas, los registros y los rastreos en múltiples fuentes de datos. Puede crear paneles interactivos y compartirlos con cualquier persona de su organización con un servicio escalable automáticamente, de alta disponibilidad y seguro para la empresa.

Comunicación y colaboración

Tanto si estás adoptando DevOps la cultura en tu organización como si estás pasando por una transformación DevOps cultural, la comunicación y la colaboración son una parte importante de tu enfoque. En Amazon, nos dimos cuenta de que era necesario cambiar la mentalidad de nuestros equipos y, por lo tanto, adoptamos el concepto de equipos de dos pizzas.

«Intentamos crear equipos que no sean más grandes de lo que se puede alimentar con dos pizzas», afirma Bezos. «A eso lo llamamos la regla de los equipos de dos pizzas».

Cuanto más pequeño sea el equipo, mejor será la colaboración. La colaboración es muy importante, ya que las versiones de software avanzan más rápido que nunca. Y la capacidad de un equipo para 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 función del producto o corregir un error. Quieres que esto suceda lo más rápido posible, para poder disponer de menos go-to-market tiempo. No quieres que la transformación sea un proceso lento; quieres un enfoque ágil en el que las oleadas de cambios comiencen a tener un impacto.

La comunicación entre los equipos también es importante a medida que se avanza hacia el modelo de responsabilidad compartida y se empieza a dejar atrás el enfoque de desarrollo aislado. Esto aporta al equipo el concepto de propiedad y cambia su perspectiva para considerar el proceso como una end-to-end empresa. Su equipo no debe pensar en sus entornos de producción como cajas negras en las que no hay visibilidad.

La transformación cultural también es importante, ya que es posible que estés creando un DevOps equipo común o que tengas un miembro DevOps centrado en tu equipo. Ambos enfoques introducen la responsabilidad compartida en el equipo.

Seguridad

Tanto si está realizando una DevOps transformación como si está implementando DevOps principios por primera vez, debe pensar en la seguridad como algo integrado en sus DevOps procesos. Esto debería ser una preocupación transversal en todas las etapas de creación y prueba e implementación.

Antes de profundizar DevOps en la seguridad AWS, este paper analiza el modelo de responsabilidad AWS compartida.

Temas

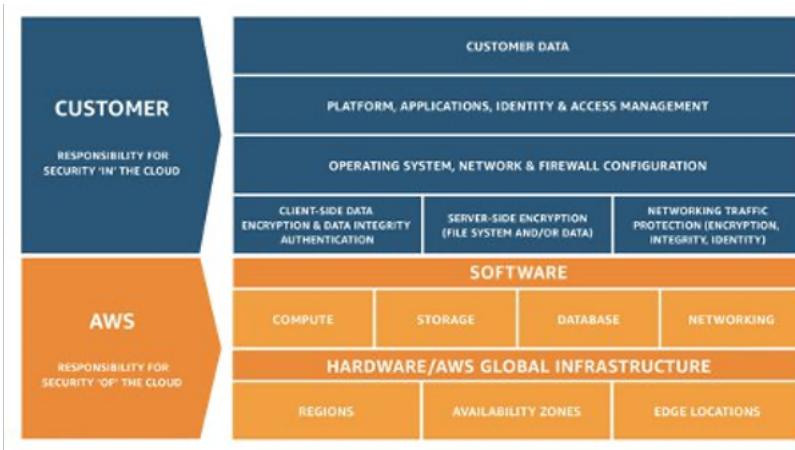
- [AWS Modelo de responsabilidad compartida](#)
- [Identity and Access Management](#)

AWS Modelo de responsabilidad compartida

La seguridad es una responsabilidad compartida entre el cliente AWS y el cliente. Las diferentes partes del modelo de responsabilidad compartida son:

- Responsabilidad de AWS: «Seguridad de la nube»: AWS es responsable de proteger la infraestructura en la que se ejecutan todos los servicios que se ofrecen en Nube de AWS. Esta infraestructura está compuesta por el hardware, el software, las redes y las instalaciones que ejecutan Nube de AWS los servicios.
- Responsabilidad del cliente «Seguridad en la nube»: la responsabilidad del cliente viene determinada por los Nube de AWS servicios que selecciona el cliente. Esto determina la cantidad de trabajo de configuración que el cliente debe llevar a cabo como parte de sus responsabilidades de seguridad.

Este modelo compartido puede ayudar a aliviar la carga operativa del cliente, ya que AWS opera, gestiona y controla los componentes, desde el sistema operativo anfitrión y la capa de virtualización hasta la seguridad física de las instalaciones en las que opera el servicio. Esto es fundamental en los casos en los que el cliente quiere entender la seguridad de sus entornos de construcción.



Modelo de responsabilidad compartida de AWS

Para ello DevOps, asigne los permisos en función del modelo de permisos [con privilegios mínimos](#). Este modelo establece que «un usuario (o servicio) debe tener los derechos de acceso exactos necesarios para cumplir con las responsabilidades de su función, ni más ni menos».

Los permisos se mantienen en IAM. Puede usar IAM para controlar quién está autenticado (ha iniciado sesión) y quién está autorizado (tiene permisos) para usar los recursos.

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. AWS Con IAM, puede crear usuarios y grupos y definir los permisos para varios servicios. DevOps

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

La IAM es un componente de la infraestructura de AWS seguridad. Con IAM, puede gestionar de forma centralizada los grupos, los usuarios, las funciones de servicio y las credenciales de seguridad, como contraseñas, claves de acceso y políticas de permisos que controlan a qué servicios y recursos de AWS pueden acceder los usuarios. La [política de IAM](#) le permite definir el conjunto de permisos. Luego, esta política se puede adjuntar a un [rol](#), [usuario](#) o [servicio](#) para definir su permiso.

También puede utilizar la IAM para crear funciones que se utilicen ampliamente dentro de la DevOps estrategia que desee. En algunos casos, puede tener mucho sentido hacerlo mediante programación [AssumeRole](#) en lugar de obtener los permisos directamente. Cuando un servicio o un usuario asume funciones, recibe credenciales temporales para acceder a un servicio al que normalmente no tiene acceso.

Conclusión

Para que el viaje a la nube sea fluido, eficiente y efectivo, las empresas de tecnología deben adoptar DevOps principios y prácticas. Estos principios están integrados en AWS numerosos AWS servicios y son la piedra angular de ellos, especialmente los que se incluyen en las ofertas de implementación y monitoreo.

Comience por definir su infraestructura como código mediante el servicio AWS CloudFormation o AWS CDK. A continuación, defina la forma en que sus aplicaciones van a utilizar el despliegue continuo 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 ECS o [Amazon Elastic Kubernetes](#) Service (Amazon EKS). OpsWorks Utilícelos para simplificar la configuración de arquitecturas comunes. El uso de estos servicios también facilita la inclusión de otros servicios importantes, como Amazon EC2 Auto Scaling y ELB.

Por último, utilice la DevOps estrategia de monitoreo, como Amazon CloudWatch, y prácticas de seguridad sólidas, como IAM.

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

Revisões del documento

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

Cambio	Descripción	Fecha
<u>Actualizado</u>	Actualizado	7 de abril de 2023
<u>Secciones actualizadas para incluir nuevos servicios</u>	Secciones actualizadas para incluir nuevos servicios	16 de octubre de 2020
<u>Publicación inicial</u>	Documento técnico publicado por primera vez	1 de diciembre de 2014

Colaboradores

Los colaboradores de este documento son:

- Abhra Sinha, arquitecta de soluciones
- Anil Nadiminti, arquitecto de soluciones
- Muhammad Mansoor, arquitecto de soluciones
- Ajit Zadgaonkar, líder tecnológico mundial en 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, gerente de desarrollo de software
- Tara McCann, gerente, arquitecta de soluciones

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.

© 2023 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.