Cómo elegir la GitOps herramienta adecuada para su clúster de Amazon EKS

AWS Guía prescriptiva

AWS Guía prescriptiva: Cómo elegir la GitOps herramienta adecuada para su clúster de Amazon EKS

Table of Contents

Cómo elegir la GitOps herramienta adecuada para su clúster de Amazon EKS

Pradip Kumar Pandey y Pratap Kumar Nanda, Amazon Web Services ()AWS

Abril de 2025 (historia del documento)

En el panorama en rápida evolución de las tecnologías nativas de la nube, se GitOps ha convertido en una poderosa metodología para administrar e implementar aplicaciones e infraestructuras. Si utiliza Amazon Elastic Kubernetes Service (Amazon EKS) GitOps , la implementación de principios puede mejorar considerablemente los procesos de implementación, aumentar la confiabilidad y agilizar las operaciones. Hay una variedad de GitOps herramientas disponibles, y elegir la adecuada para su clúster de EKS es una decisión fundamental que puede afectar a la eficiencia de su equipo y al éxito general de sus prácticas. DevOps

La selección de una GitOps herramienta adecuada para su entorno Amazon EKS implica considerar detenidamente varios factores, incluidos sus requisitos específicos, la experiencia del equipo, las necesidades de escalabilidad y las capacidades de integración con las existentes Servicios de AWS. Cada herramienta viene con su propio conjunto de características, puntos fuertes y posibles limitaciones, por lo que es esencial alinear su elección con los objetivos y el contexto operativo de su organización.

En esta guía se analizan las consideraciones clave a la hora de seleccionar GitOps herramientas para Amazon EKS, se comparan las opciones que se utilizan con más frecuencia y se proporciona información que le ayudará a tomar una decisión fundamentada. Abarca nueve GitOps herramientas populares:

- Argo CD
- Flujo
- Tejido GitOps
- Jenkins X
- GitLab CI/CD
- Spinnaker
- Flota de rancheros
- Codefresh

- Pulumi

Resultados empresariales específicos

La siguiente lista analiza los posibles objetivos y resultados al elegir una herramienta para implementar GitOps principios en sus procesos de desarrollo y operaciones.

Integración perfecta con Amazon EKS

GitOps La herramienta debe integrarse sin problemas con Amazon EKS y ser compatible con las funciones y optimizaciones específicas de Amazon EKS.

- Soporte nativo de Amazon EKS: busque herramientas que ofrezcan soporte integrado para Amazon EKS, incluida una fácil conexión y administración de clústeres.
- Servicio de AWS integración: asegúrese de que la herramienta pueda interactuar con otras, Servicios de AWS como AWS Identity and Access Management (IAM), Amazon Elastic Container Registry (Amazon ECR) y Amazon. CloudWatch
- Compatibilidad con los complementos de Amazon EKS: confirme que la herramienta es compatible con los complementos de Amazon EKS y que puede gestionarlos de forma eficaz.

Escalabilidad y rendimiento

Su GitOps herramienta debe poder gestionar la escala de sus operaciones de Amazon EKS, desde clústeres pequeños hasta entornos grandes con varios clústeres.

- Eficiencia de los recursos: evalúe el consumo de recursos de la herramienta y su impacto en el rendimiento del clúster.
- Operaciones a gran escala: evalúe la capacidad de la herramienta para gestionar numerosas aplicaciones y clústeres de forma simultánea.
- Rendimiento bajo carga: tenga en cuenta el rendimiento de la herramienta durante las actualizaciones de alta frecuencia y las implementaciones a gran escala.

Seguridad y conformidad

Las características de seguridad y las capacidades de cumplimiento son cruciales, especialmente en los sectores regulados o cuando se manejan datos confidenciales.

- Control de acceso: busque funciones sólidas de control de acceso basado en roles (RBAC) que se integren con la IAM.
- Gestión de secretos: evalúe cómo la herramienta gestiona la información confidencial y cómo se integra con otras soluciones. AWS Secrets Manager
- Registros de auditoría: asegúrese de que la herramienta proporcione capacidades integrales de registro y auditoría para garantizar el cumplimiento y la solución de problemas.
- Escaneo de seguridad: considere herramientas que ofrezcan un escaneo de seguridad integrado para detectar vulnerabilidades en las implementaciones.

Facilidad de uso y curva de aprendizaje

La herramienta debe ser fácil de usar y estar alineada con las habilidades de su equipo para garantizar una adopción rápida y un uso eficiente.

- Interfaz de usuario: evalúe la intuitividad de las funciones de la interfaz de línea de comandos (CLI) y de la interfaz gráfica de usuario (GUI).
- Calidad de la documentación: busque up-to-date documentación y tutoriales completos.
- Recursos de aprendizaje: considere la disponibilidad de materiales de capacitación, cursos y recursos comunitarios.

Apoyo de la comunidad y de la red

Una comunidad y una red sólidas pueden proporcionar recursos valiosos, complementos y sostenibilidad a largo plazo.

- Desarrollo activo: compruebe la frecuencia de las actualizaciones y la capacidad de respuesta de los responsables del mantenimiento.
- Tamaño de la comunidad: tenga en cuenta el tamaño y la actividad de la comunidad de usuarios para obtener apoyo e intercambiar conocimientos.
- Integraciones de terceros: evalúa la disponibilidad de los complementos y las integraciones con otras herramientas de tu lista.

Capacidades de administración de varios clústeres

Si tiene varios clústeres de EKS, la capacidad de gestionarlos de forma eficiente es crucial.

- Administración centralizada: busque funciones que permitan administrar varios clústeres desde un único plano de control.
- Federación de clústeres: considere herramientas que admitan la federación de Kubernetes para aplicaciones con varios clústeres.
- Paridad del entorno: evalúe en qué medida la herramienta mantiene la coherencia en los diferentes entornos, como el desarrollo, la puesta en escena y la producción.

Observabilidad y supervisión

La herramienta debe proporcionar información clara sobre el estado de las implementaciones y el estado del clúster.

- Visibilidad de la implementación: busque funciones que ofrezcan una visión clara del estado y el historial de la implementación.
- Integración con las herramientas de monitoreo: considere qué tan bien se integra la herramienta con las soluciones de monitoreo populares, como Prometheus y Grafana.
- Capacidades de alerta: evalúe la capacidad de la herramienta para configurar y gestionar alertas en caso de problemas de implementación o desviaciones.

Flexibilidad y personalización

La capacidad de adaptar la herramienta a sus flujos de trabajo y requisitos específicos es importante para una satisfacción a largo plazo.

- Extensibilidad: busque arquitecturas de complementos o APIs que le permitan ampliar la funcionalidad de la herramienta.
- Soporte de recursos personalizados: confirme que la herramienta puede gestionar los recursos personalizados de Kubernetes de forma eficaz.
- Personalización del flujo de trabajo: evalúa la facilidad con la que puedes adaptar los GitOps flujos de trabajo a las necesidades de tu equipo.

Soporte de entrega continua e implementación progresiva

Las estrategias de implementación avanzadas suelen ser cruciales para minimizar el riesgo y garantizar que las actualizaciones se realicen sin problemas.

- Implementaciones de Canary: busque un soporte integrado para las versiones de Canary.
- Blue/green deployments: Assess the tool's capabilities for blue/greenestrategias de despliegue.
- Mecanismos de reversión: garantice funciones sólidas y de easy-to-use reversión para una recuperación rápida en caso de implementaciones fallidas.

Rentabilidad y utilización de los recursos

Tenga en cuenta el costo total de adoptar y mantener la herramienta, incluidos los costos directos e indirectos.

- Costes de licencia: compare las opciones de código abierto con las soluciones comerciales y considere las funciones empresariales y de soporte.
- Gastos generales operativos: evalúe los costos operativos adicionales en términos de administración y mantenimiento.
- Consumo de recursos: evalúe la eficiencia de la herramienta en términos de los recursos informáticos y de almacenamiento que se necesitarían.

Al considerar detenidamente estos resultados y sus aspectos, puede tomar una decisión informada sobre la GitOps herramienta más adecuada para su clúster de EKS y asegurarse de que la herramienta se ajuste a las necesidades, las capacidades y la estrategia a largo plazo de su organización.

GitOps herramientas para clústeres EKS

Actualmente, hay varias GitOps herramientas para Kubernetes disponibles en el mercado. Esta es una lista de algunas de las opciones más utilizadas:

- Argo (CD)
- Flujo
- Tejido GitOps
- Jenkins X
- GitLab CI/CD
- Spinnaker
- Flota Rancher
- Codefresh
- Pulumi

Siga los enlaces para ver información detallada sobre cómo estas herramientas implementan las GitOps prácticas. Cada herramienta tiene puntos fuertes y casos de uso. La elección depende de factores como los requisitos específicos, la infraestructura existente, la experiencia del equipo y las funciones deseadas. Es importante evaluar estas herramientas en función de las necesidades de su organización y de la complejidad de su entorno de Kubernetes.

Argo CD

Argo CD es una herramienta de entrega GitOps continua (CD) muy utilizada en Kubernetes que cumple con varios principios clave. GitOps

GitOps soporte

Área	Capacidades de la herramienta
Configuración declarativa	Argo CD usa configuraciones declarativas que se almacenan en los repositorios de Git. El estado deseado de la aplicación y la infraestr uctura se define en los archivos YAML. Estas

Área	Capacidades de la herramienta
	configuraciones describen lo que se debe implementar, no cómo implementarlo.
El sistema de control de versiones como única fuente de información	Los repositorios de Git son la única fuente de información fiable para todo el sistema. Todos los cambios en la aplicación y la infraestructura se realizan a través de Git. Esto garantiza un registro de auditoría completo y la posibilidad de volver a cualquier estado anterior.
Sincronización automatizada	Argo CD monitorea continuamente el repositor io de Git para detectar cambios. Cuando se detectan cambios, sincroniza automáticamente el estado real del clúster con el estado deseado definido en Git. Esto garantiza que el clúster siempre refleje el estado que se describe en el repositorio.
Nativo de Kubernetes	Argo CD está diseñado específicamente para entornos de Kubernetes. Aprovecha la naturaleza declarativa y los recursos personali zados de Kubernetes para gestionar las aplicaciones.
Recuperación automática y detección de desviaciones	Argo CD compara periódicamente el estado activo del clúster con el estado deseado en Git. Si detecta alguna desviación (diferencias entre el estado real y el deseado), puede corregir automáticamente estas discrepancias.
Soporte para múltiples clústeres y múltiples inquilinos	Argo CD puede gestionar varios clústeres de Kubernetes desde una sola instancia. Es compatible con la multitenencia, por lo que los diferentes equipos pueden gestionar sus aplicaciones de forma independiente.

Área	Capacidades de la herramienta
Definición de aplicación	Las aplicaciones de Argo CD se definen mediante la CRD de aplicaciones (definición de recursos personalizada). Esto permite una forma nativa de Kubernetes de definir qué se debe implementar y cómo.
Separación del despliegue y el lanzamiento	Argo CD separa la distribución del código de su publicación para los usuarios. Esto se logra mediante diversas estrategias de despliegue, como blue/green los despliegues canarios.
Observabilidad y auditabilidad	Argo CD proporciona una interfaz de usuario web y una CLI para observar el estado de las aplicaciones y los clústeres. Todas las acciones se registran para proporcionar un registro de auditoría claro de los cambios y las implementaciones.
Seguridad y RBAC	Argo CD se integra con el control de acceso basado en roles (RBAC) de Kubernetes. Admite la integración del inicio de sesión único para la autenticación y la autorización.
Arquitectura conectable	Argo CD es compatible con varios sistemas de gestión de control de código fuente, gráficos de Helm, Kustomize y otros formatos de manifiesto de Kubernetes. Esta flexibilidad le permite adaptarse a diversos entornos y flujos de trabajo.
Entrega continua (CD)	Si bien Argo CD se centra en la entrega continua, se puede integrar con herramientas de integración continua (CI) para crear una CI/CD cartera completa.

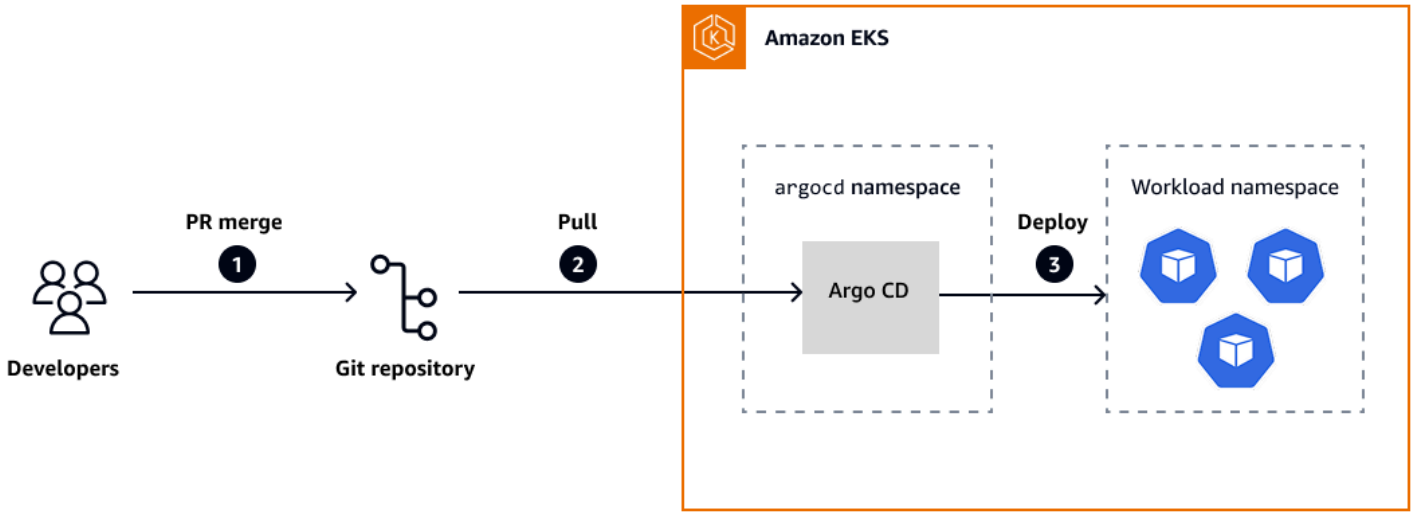
Al cumplir con estos GitOps principios, Argo CD proporciona una forma sólida, escalable y segura de gestionar las implementaciones de Kubernetes. Garantiza que el estado operativo del sistema esté siempre sincronizado con el estado deseado definido en el repositorio de Git y promueve la coherencia, la fiabilidad y la facilidad de administración en entornos complejos de Kubernetes.

Para ver los escenarios y requisitos que Argo CD puede abordar, consulta los casos de uso de Argo CD más adelante en esta guía. Para ver una comparación entre Argo CD y Flux, consulta la comparación de funciones más adelante en esta guía.

Para obtener información adicional, consulta la documentación de Argo CD.

Arquitectura

El siguiente diagrama ilustra un flujo de trabajo de CD GitOps controlado por componentes que utiliza Argo CD dentro de un clúster de EKS. Para obtener información detallada, consulte la documentación de Argo CD.



donde:

- Paso 1: Fusión mediante solicitud de extracción (PR). Un desarrollador confirma los cambios en los manifiestos de Kubernetes o en los gráficos de Helm almacenados en un repositorio de Git. Cuando el PR se ha revisado y fusionado con la rama principal, el estado deseado de la aplicación se actualiza en el control de código fuente.
- Paso 2: Sincronización del repositorio. Argo CD se ejecuta dentro de un espacio de nombres dedicado (`argocd`) en el clúster de EKS y monitorea continuamente el repositorio de Git configurado. Cuando detecta cambios, extrae las últimas actualizaciones para conciliar el estado declarado.

- Paso 3: Implementación en el espacio de nombres de destino. Argo CD compara el estado deseado de Git con el estado activo del clúster. A continuación, aplica los cambios necesarios al espacio de nombres de la carga de trabajo de destino para que la aplicación se despliegue o actualice en consecuencia. Esto incluye la administración de los recursos de Kubernetes, como las implementaciones, los servicios y los secretos ConfigMaps, para mantener la coherencia de los clústeres con la fuente de información de Git.

Flujo

Flux es otra herramienta para Kubernetes que implementa los GitOps principios de una manera única.

GitOps soporte

Área	Capacidades de la herramienta
Git como única fuente de información	Flux utiliza los repositorios de Git como fuente definitiva para definir el estado deseado del sistema. Toda la configuración de las aplicacio nes y la infraestructura se almacena en Git.
Configuración declarativa	Flux trabaja con descripciones declarativas del estado deseado de su clúster. Estas descripci ones suelen ser manifiestos de Kubernete s, gráficos de Helm o superposiciones de Kustomize.
Sincronización automatizada	Flux monitorea continuamente el repositorio de Git para detectar cambios. Cuando detecta cambios, los aplica automáticamente al clúster.
Nativo de Kubernetes	Flux está diseñado como un conjunto de controladores de Kubernetes y recursos personalizados. Utiliza los mecanismos de extensión de Kubernetes para proporcionar capacidades. GitOps

Área	Capacidades de la herramienta
Modelo de despliegue basado en extracciones	A diferencia de los CI/CD sistemas tradicion ales basados en la tracción, Flux utiliza un modelo basado en la tracción. El clúster extrae el estado deseado de Git en lugar de utilizar un sistema externo para enviar los cambios.
Reconciliación continua	Flux compara constantemente el estado real del clúster con el estado deseado en Git. Corrige automáticamente cualquier desviación que se detecte entre estos estados.
Tenencia múltiple	Flux apoya la multitenencia a través de sus conceptos de personalización y. HelmRelea ses Los diferentes equipos pueden gestionar sus propias partes de la configuración de forma independiente.
Entrega progresiva	Flux admite estrategias de implementación avanzadas, como las versiones y las A/B pruebas de Canary, a través de su componente Flagger.
Integración de Helm	Flux incluye soporte nativo para Helm, por lo que puedes gestionar fácilmente las versiones de Helm GitOps.
Automatización de la actualización de imágenes	Flux puede actualizar automáticamente las imágenes del contenedor en Git cuando hay nuevas versiones disponibles en el registro del contenedor.
Personaliza el soporte	Puedes usar el soporte nativo que ofrece Flux para Kustomize para personalizar y parchear los manifiestos de Kubernetes.

Área	Capacidades de la herramienta
Seguridad y RBAC	Flux se integra con el RBAC de Kubernetes para el control de acceso. Soporta la gestión de secretos a través de varios backends.
Observabilidad	Flux proporciona información de estado y métricas sobre la reconciliación y las operacion es. Se integra con las herramientas de monitoreo para mejorar la observabilidad.
Arquitectura basada en eventos	Flux utiliza un enfoque basado en eventos para implementar conciliaciones y actualizaciones.
Extensibilidad	La herramienta está diseñada para ser extensible, por lo que puede agregar controlad ores y recursos personalizados.
Sincronización entre clústeres	Flux admite la gestión de varios clústeres desde un único conjunto de repositorios.
Administración de dependencias	Permite definir las dependencias entre las diferentes partes del sistema y garantiza el orden correcto de las operaciones.
Receptores Webhook	Puedes configurar Flux para recibir webhooks de proveedores de Git u otros sistemas para iniciar la reconciliación inmediata.

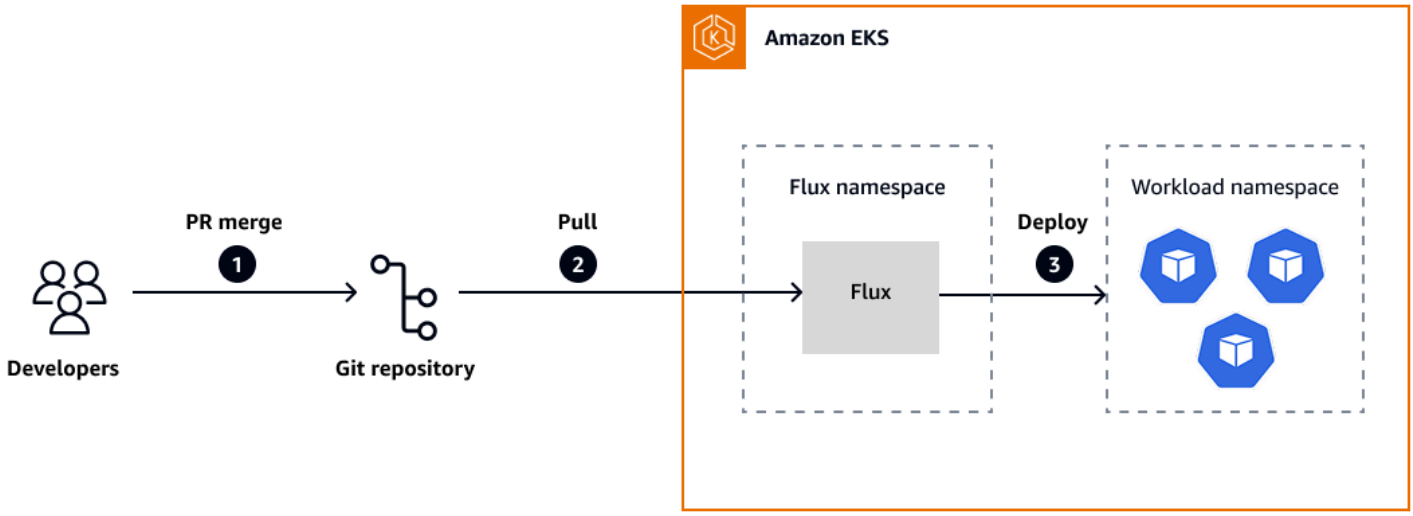
Al implementar estos GitOps principios, Flux proporciona un sistema sólido y flexible para administrar los clústeres y las aplicaciones de Kubernetes. Garantiza que tu infraestructura y tus aplicaciones estén siempre sincronizadas con tus repositorios de Git y proporciona coherencia, confiabilidad y facilidad de administración en entornos complejos de Kubernetes. El enfoque nativo de Kubernetes de la herramienta y su enfoque en la automatización la hacen especialmente adecuada para entornos nativos de la nube.

Para ver los escenarios y requisitos que Flux puede abordar, consulte los casos de uso de Flux más adelante en esta guía. Para ver una comparación entre Argo CD y Flux, consulta la comparación de funciones más adelante en esta guía.

Para obtener información adicional, consulta la documentación de Flux.

Arquitectura

El siguiente diagrama ilustra un flujo de trabajo de CD GitOps controlado por componentes que utiliza Flux dentro de un clúster de EKS. Para obtener información detallada, consulte la documentación de Flux.



donde:

- Paso 1: Fusión mediante solicitud de extracción (PR). Un desarrollador confirma los cambios en los manifiestos de Kubernetes o en los gráficos de Helm almacenados en un repositorio de Git. Cuando el PR se ha revisado y fusionado con la rama principal, el estado deseado de la aplicación se actualiza en el control de código fuente.
- Paso 2: Sincronización del repositorio. Flux se ejecuta dentro de un espacio de nombres dedicado en el clúster de EKS y monitorea continuamente el repositorio de Git configurado. Cuando detecta cambios, extrae las últimas actualizaciones para conciliar el estado declarado.
- Paso 3: Implementación en el espacio de nombres de destino. Flux compara el estado deseado de Git con el estado activo del clúster. A continuación, aplica los cambios necesarios al espacio de nombres de la carga de trabajo de destino para que la aplicación se despliegue o actualice en consecuencia.

Tejer GitOps

Weave GitOps fue desarrollado por Weaveworks, que es la empresa que introdujo el término. GitOps Esta herramienta proporciona una GitOps solución integral que se basa en los principios básicos. GitOps

GitOps apoyo

Área	Capacidades de la herramienta
Git como única fuente de información	Weave GitOps usa los repositorios de Git como fuente autorizada para definir el estado deseado del sistema. Todas las configura ciones, incluidos los manifiestos de aplicación, las definiciones de infraestructura y las políticas , se almacenan en Git.
Configuración declarativa	El sistema se basa en descripciones declarati vas de todo el estado del sistema. Estas descripciones suelen ser manifiestos de Kubernetes, gráficos de Helm u otros formatos declarativos.
Sincronización automatizada	Weave monitorea GitOps continuamente los repositorios de Git para detectar cambios. Cuando detecta cambios, los aplica automátic amente al entorno de destino.
Arquitectura nativa de Kubernetes	Weave GitOps se creó como un conjunto de controladores y recursos personalizados de Kubernetes. Utiliza los mecanismos de extensión de Kubernetes para proporcionar capacidades. GitOps
Reconciliación continua	Esta herramienta compara constantemente el estado real del clúster con el estado deseado definido en Git. Corrige automáticamente

Área	Capacidades de la herramienta
	cualquier desviación detectada entre estos estados.
Administración de varios clústeres	Weave GitOps admite la administración de varios clústeres de Kubernetes desde un único plano de control. Permite una implementación uniforme de las aplicaciones en diferentes entornos.
La política como código	Weave GitOps incorpora el concepto de política como código para hacer cumplir las normas de seguridad y cumplimiento. Las políticas están controladas por versiones junto con el código de la aplicación y las definiciones de infraestr uctura.
Entrega progresiva	Esta herramienta es compatible con estrategi as de implementación avanzadas, como las versiones e blue/green implementaciones estándar. Se integra con Flagger para una entrega automática y progresiva.
Observabilidad y cuadros de mando	Weave GitOps proporciona paneles integrado s para monitorear el estado de las aplicaciones y los clústeres. Ofrece información sobre los procesos de reconciliación y el estado de los clústeres.
Seguro por diseño	La herramienta implementa las mejores prácticas de seguridad, incluida la integraci ón del RBAC y la gestión de secretos. Es compatible con varios métodos de autentica ción y se integra con los proveedores de identidad empresariales.

Área	Capacidades de la herramienta
Extensibilidad e integración	La herramienta está diseñada para funcionar con una amplia gama de herramientas nativas de la nube. Es compatible con herramientas populares como Flux, Helm y Kustomize.
Plataformas de autoservicio para desarroll adores	Weave GitOps permite la creación de plataform as de autoservicio para desarrolladores. Proporciona plantillas y barandillas para la implementación de aplicaciones.
GitOps Automation	La herramienta automatiza muchos aspectos del GitOps flujo de trabajo, incluida la generación de solicitudes de cambios para las actualizaciones.
Canalizaciones de entrega continua	Se integra con CI/CD los sistemas para crear canalizaciones end-to-end de entrega.
Auditoría y conformidad	Weave DevOps proporciona un registro de auditoría completo de todos los cambios y acciones. Le ayuda a cumplir los requisitos de conformidad mediante el control de versiones y los procesos automatizados.
Escalabilidad	La herramienta está diseñada para ampliarse desde proyectos pequeños hasta grandes implementaciones de nivel empresarial.
Colaboración en equipo	Weave GitOps facilita la colaboración entre los equipos de desarrollo y operaciones a través de flujos de trabajo basados en Git.
GitOps como servicio	Esta herramienta se ofrece GitOps como un servicio gestionado, lo que simplifica la adopción y la gestión.

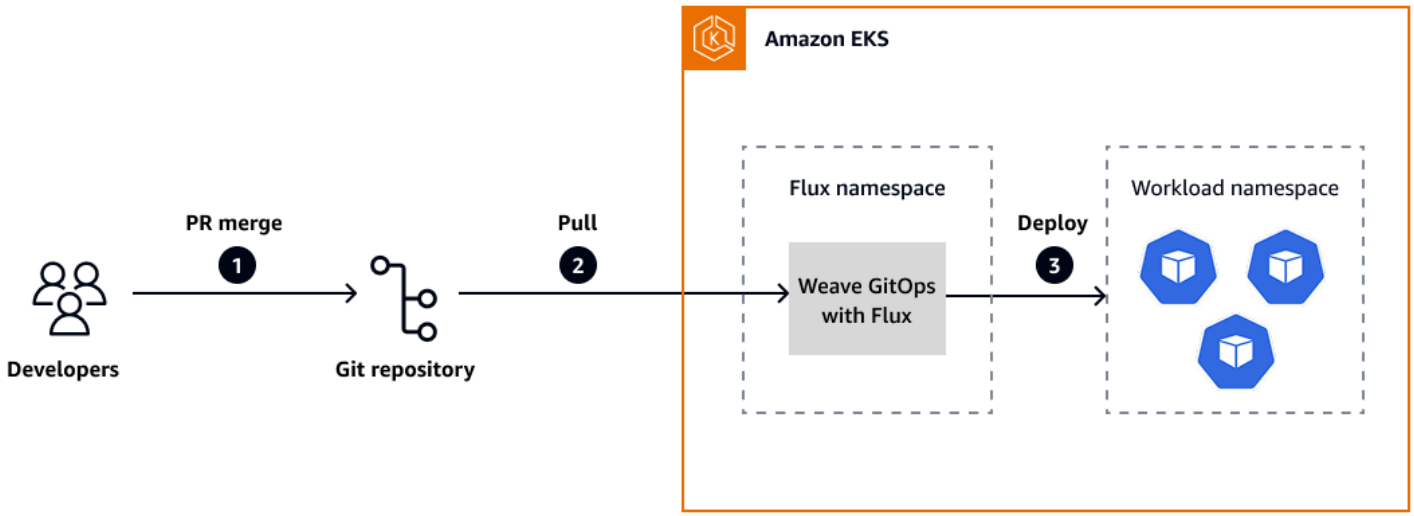
Área	Capacidades de la herramienta
Soporte híbrido y multinube	Weave GitOps permite una administración uniforme en diferentes proveedores de nube y entornos locales.
Seguridad continua	La herramienta integra el análisis de seguridad y la aplicación de políticas durante todo el proceso de implementación.

Weave GitOps implementa estos principios para proporcionar una GitOps solución integral que va más allá de la automatización básica de la implementación. Su objetivo es crear un modelo operativo completo para las aplicaciones nativas de la nube que se centre en la seguridad, la escalabilidad y la facilidad de uso. Al cumplir con estos GitOps principios, Weave GitOps ayuda a las organizaciones a lograr una gestión coherente, auditable y eficiente de sus entornos de Kubernetes en varios clústeres y proveedores de nube.

Para obtener más información, consulta la documentación de Weave. GitOps

Arquitectura

El siguiente diagrama ilustra un flujo de trabajo de CD GitOps controlado por componentes que utiliza Weave GitOps dentro de un clúster de EKS. Para obtener información detallada, consulte el repositorio de GitOpsWeave.



donde:

- Paso 1: Fusión mediante solicitud de extracción (PR). Un desarrollador confirma los cambios en los manifiestos de Kubernetes o en los gráficos de Helm almacenados en un repositorio de Git. Cuando el PR se ha revisado y fusionado con la rama principal, el estado deseado de la aplicación se actualiza en el control de código fuente.
- Paso 2: Sincronización del repositorio. Weave GitOps se ejecuta en el espacio de nombres Flux del clúster EKS y monitorea continuamente el repositorio Git configurado. Cuando detecta cambios, extrae las últimas actualizaciones para conciliar el estado declarado.
- Paso 3: Implementación en el espacio de nombres de destino. Weave GitOps compara el estado deseado de Git con el estado activo del clúster. A continuación, aplica los cambios necesarios al espacio de nombres de la carga de trabajo de destino para que la aplicación se despliegue o actualice en consecuencia.

Jenkins X

Jenkins X es una CI/CD plataforma de código abierto nativa de la nube que implementa GitOps principios para los entornos de Kubernetes. Aunque Jenkins X no es exclusivamente una GitOps herramienta como Argo CD o Flux, incorpora prácticas en sus flujos de trabajo. GitOps

GitOps soporte

Área	Capacidades de la herramienta
Flujo de trabajo centrado en Git	Jenkins X usa los repositorios de Git como fuente principal de información tanto para el código como para la configuración de la aplicación. Todos los cambios en las aplicacio nes y la infraestructura se realizan a través de Git.
El entorno como código (eAC)	Los entornos (como la puesta en escena y la producción) se definen como código en los repositorios de Git. Esto permite controlar las versiones y revisar las configuraciones del entorno.

Área	Capacidades de la herramienta
CI/CD Canalizaciones automatizadas	Jenkins X configura automáticamente las CI/CD canalizaciones para los proyectos. Estas canalizaciones se definen como código (canalización como código) y se almacenan en Git.
Nativo de Kubernetes	Jenkins X está diseñado específicamente para entornos de Kubernetes. Utiliza recursos de Kubernetes y definiciones de recursos personalizadas (). CRDs
Previsualice los entornos	Jenkins X crea automáticamente entornos temporales para las solicitudes de incorpora ción de cambios. Permite revisar y probar fácilmente los cambios antes de las fusiones.
Promoción entre entornos	Jenkins X utiliza un GitOps enfoque para promover las aplicaciones entre entornos (por ejemplo, desde la puesta en escena hasta la producción). Las promociones se gestionan mediante solicitudes de selección para garantizar que los procesos de revisión y aprobación sean adecuados.
Gestión de gráficos de Helm	Jenkins X usa los gráficos de Helm para empaquetar e implementar aplicaciones. Los gráficos están controlados por versiones en los repositorios de Git.
Control de versiones automatizado	Jenkins X gestiona automáticamente el control de versiones de aplicaciones y versiones. Utiliza el control de versiones semántico y genera notas de lanzamiento.

Área	Capacidades de la herramienta
ChatOps integración	Jenkins X admite ChatOps operaciones comunes. Esto se alinea con los GitOps principios de automatización y colaboración.
Extensibilidad	Esta herramienta proporciona un sistema de complementos para ampliar la funcionalidad. Permite la integración con varias herramientas nativas de la nube.
Infraestructura como código (IaC)	Jenkins X es compatible con Terraform y otras herramientas de IaC para definir y administrar la infraestructura. CloudFormation AWS Cloud Development Kit (AWS CDK) Las definiciones de infraestructura se controlan por versiones junto con el código de la aplicación.
Reversiones automatizadas	Jenkins X admite la reversión automática si se detectan problemas después de la implement ación.
Administración de secretos	La herramienta se integra con soluciones de gestión de secretos externas para gestionar la información confidencial de forma segura.
Observabilidad	Jenkins X proporciona integración con herramientas de monitoreo y registro para la observabilidad.
Soporte multinube	Jenkins X está diseñado para funcionar en diferentes proveedores de nube y entornos locales.
Colaboración en equipo	Esta herramienta fomenta la colaboración a través de flujos de trabajo y solicitudes de cambios basados en Git.

Área	Capacidades de la herramienta
Retroalimentación continua	La herramienta proporciona información rápida sobre los cambios mediante entornos automatizados de pruebas y vistas previas.
DevOps mejores prácticas	Jenkins X implementa las DevOps mejores prácticas de forma predeterminada, incluidos GitOps los principios.
Configuración declarativa	La herramienta utiliza configuraciones declarativas para definir aplicaciones y entornos.
Actualizaciones automatizadas	Jenkins X proporciona herramientas para automatizar las actualizaciones de la propia plataforma Jenkins X.

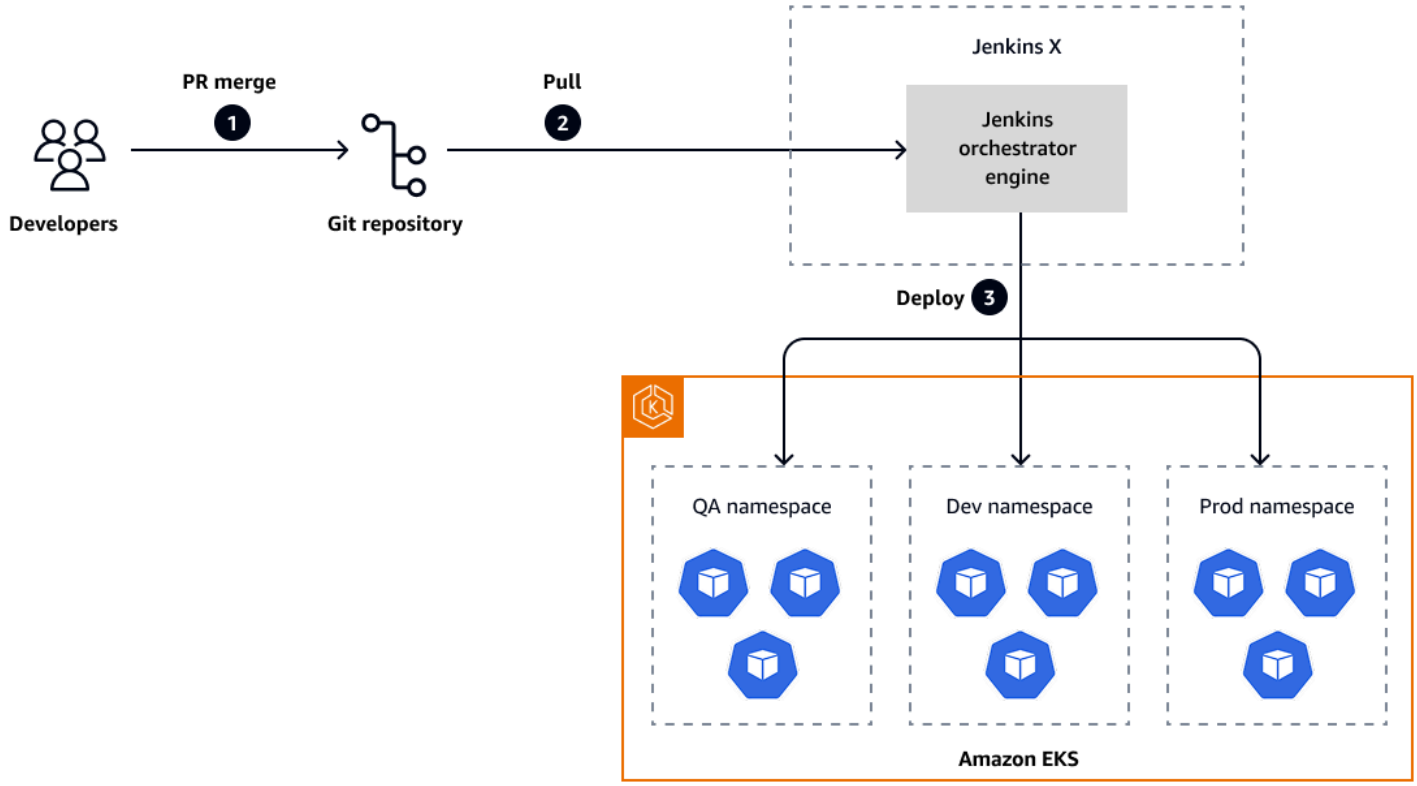
Jenkins X implementa estos GitOps principios para crear una solución integral de CI/CD para Kubernetes. Su objetivo es automatizar y agilizar todo el proceso de entrega de software, desde la confirmación del código hasta la implementación en producción, sin dejar de cumplir con las prácticas. GitOps De este modo, ayuda a los equipos a lograr despliegues más rápidos, fiables y consistentes en entornos nativos de la nube.

La diferencia clave entre Jenkins X y herramientas como Argo CD o Flux es que Jenkins X ofrece una CI/CD solución más completa, que incluye la automatización de la construcción y la gestión de los procesos, sin dejar de incorporar GitOps principios para la gestión del entorno y el despliegue. Esto lo hace especialmente adecuado para los equipos que necesitan una all-in-one solución que abarque los aspectos de CI y CD en un solo marco. GitOps

Para obtener más información, consulte la documentación de Jenkins X.

Arquitectura

El siguiente diagrama ilustra un flujo de trabajo de CD GitOps controlado por componentes que utiliza Jenkins X. Para obtener información detallada, consulte la documentación de Jenkins X.



donde:

- Paso 1: Fusión mediante solicitud de extracción (PR). Un desarrollador crea una solicitud de cambios en los manifiestos de Kubernetes, los gráficos de Helm o el código de la aplicación que está almacenado en un repositorio de Git. Tras su revisión y aprobación, el PR se fusiona con la rama principal y actualiza el estado deseado en el control de código fuente.
- Paso 2: Sincronización del repositorio. Jenkins X activa automáticamente una CI/CD canalización cuando detecta el cambio. La canalización crea, prueba y promueve la aplicación en diferentes entornos (por ejemplo, puesta en escena y producción) utilizando GitOps principios.
- Paso 3: Implementación en los espacios de nombres de destino. Jenkins X actualiza los repositorios del entorno (de puesta en escena y de producción) con las nuevas versiones de la aplicación. El clúster reconcilia automáticamente los cambios mediante la extracción de los últimos manifiestos de Git y el despliegue de la aplicación en los espacios de nombres correspondientes.

GitLab CI/CD

GitLab CI/CD is an integrated part of the GitLab platform that provides continuous integration, delivery, and deployment capabilities. Although GitLab CI/CDno es exclusivamente una GitOps

herramienta, sino que puede configurarla para implementar GitOps principios, especialmente cuando la usa para despliegues de Kubernetes.

GitOps soporte

Área	Capacidades de la herramienta
Git como única fuente de información	GitLab CI/CD usa los repositorios de Git para almacenar tanto el código de la aplicación como las configuraciones de infraestructura. Todos los cambios en el sistema se realizan a través de Git, lo que garantiza un historial y un registro de auditoría completos.
Configuración declarativa	GitLab Las canalizaciones de CI/CD se definen en un archivo.gitlab-ci.yml, que es una configuración declarativa almacenad a en el repositorio de Git. Los manifiestos de Kubernetes, los gráficos de Helm u otros archivos de infraestructura como código (IaC) se pueden almacenar en el mismo repositorio para definir el estado deseado de la infraestr uctura.
Canalizaciones automatizadas	GitLab El CI/CD activa automáticamente las canalizaciones cuando los cambios se envían al repositorio. Estas canalizaciones pueden incluir etapas para crear, probar e implementar aplicaciones.
Integración con Kubernetes	GitLab La CI/CD proporciona una integraci ón nativa con Kubernetes y admite implement aciones de estilo propio en clústeres de Kubernetes. GitOps Puede crear y gestionar automáticamente los recursos de Kubernetes en función de la configuración de Git.

Área	Capacidades de la herramienta
Gestión del entorno	GitLab La CI/CD admite la definición de varios entornos (como la puesta en escena y la producción) como código. Las implement aciones en estos entornos se pueden automatizar o pueden requerir una aprobació n manual, de conformidad con las prácticas. GitOps
Revise las solicitudes	GitLab puede crear automáticamente entornos temporales para las solicitudes de fusión, de forma similar a los entornos de previsualización de otras GitOps herramientas. Esto permite revisar y probar fácilmente los cambios antes de las fusiones.
Implementación continua	GitLab El CI/CD se puede configurar para implementar automáticamente los cambios en los clústeres de Kubernetes cuando los cambios se fusionen en ramas específicas.
IaC	GitLab El CI/CD permite la integración con herramientas como Terraform y permite gestionar la infraestructura como código. CloudFormation Las definiciones de infraestr uctura se pueden controlar por versiones junto con el código de la aplicación.
Observabilidad y monitoreo	GitLab El CI/CD proporciona funciones integradas de monitoreo y observabilidad, incluida la integración con Prometheus y Grafana.

Área	Capacidades de la herramienta
Análisis de seguridad	GitLab CI/CD includes built-in security scanning tools that can be integrated into the CI/CDcanalización para reforzar la seguridad como parte del flujo de trabajo. GitOps
Registro de contenedores	GitLab El CI/CD incluye un registro de contenedores integrado para una integración perfecta de la administración de imágenes de contenedores en el GitOps flujo de trabajo.
Automático DevOps	La DevOps función automática en las GitLab CI/CD can automatically configure CI/CD canalizaciones que siguen los GitOps principios de las implementaciones de Kubernetes.
flujos de trabajo de aprobación	GitLab El CI/CD admite los procesos de aprobación de las implementaciones, lo que permite ascensos controlados entre entornos.
Administración de secretos	GitLab CI/CD provides features to securely manage and use secrets within CI/CDoleoductos.
Control de versiones y lanzamientos	GitLab CI/CD supports automatic versioning and release management as part of the CI/CDproceso.
Reversiones	GitLab La tecnología CI/CD permite volver fácilmente a versiones anteriores si se detectan problemas después de la implementación.
Registros de auditoría	GitLab El CI/CD proporciona registros de auditoría completos para todas las acciones a fin de respaldar el aspecto de la trazabilidad. GitOps

Área	Capacidades de la herramienta
Canalizaciones de varios proyectos	GitLab El CI/CD admite GitOps flujos de trabajo complejos que abarcan varios proyectos o repositorios.
ChatOps	GitLab CI/CD admite ChatOps las integraci ones, que proporcionan colaboración y operaciones a través de interfaces de chat.
Administración de clústeres de Kubernetes	GitLab La CI/CD proporciona funciones para administrar los clústeres de Kubernetes directamente desde la interfaz. GitLab

Sin embargo GitLab CI/CD is not exclusively designed for GitOps, it can be used effectively to implement GitOps practices, especially for teams that already use GitLab as their primary development platform. Its integrated approach, which combines source control, CI/CD, y la administración de Kubernetes, lo convierten en una herramienta poderosa para implementar flujos de trabajo. GitOps

La diferencia clave entre las capacidades. GitLab CI/CD and dedicated GitOps tools such as Argo CD or Flux is that GitLab provides a more comprehensive platform that includes source control management, issue tracking, and other development tools along with its CI/CD Esto lo hace especialmente adecuado para los equipos que necesitan una all-in-one solución que pueda implementar GitOps prácticas dentro de un sistema de desarrollo más amplio.

Para obtener más información sobre la GitLab CI/CD y su arquitectura, consulte la documentación de la GitLab CI/CD.

Espinaquer

Si bien Spinnaker no está diseñado exclusivamente como una GitOps herramienta, puede configurarlo para implementar GitOps principios, especialmente cuando lo usa para implementaciones nativas de la nube y de Kubernetes.

GitOps soporte

Área	Capacidades de la herramienta
Configuración declarativa	Spinnaker utiliza definiciones de canalización declarativas, que normalmente se almacenan como archivos JSON o YAML. Estas definicio nes de canalización se pueden controlar por versiones en los repositorios de Git, de acuerdo con las prácticas. GitOps
IaC	Spinnaker admite la definición de las configura ciones de infraestructura e implementación en forma de código. Estas definiciones se pueden almacenar en los repositorios de Git y pueden servir como única fuente de información.
Implementaciones multinube	Spinnaker está diseñado para funcionar en varios proveedores de nube y clústeres de Kubernetes. Permite prácticas coherentes GitOps en diversos entornos.
La canalización como código	Las canalizaciones de Spinnaker pueden definirse como código y almacenarse en los repositorios de Git. Esto permite el control de versiones y la revisión de los procesos de despliegue.
Implementaciones automatizadas	Puedes configurar Spinnaker para que inicie automáticamente las implementaciones en función de los cambios en los repositorios de Git. La herramienta es compatible con las prácticas de despliegue continuo que son fundamentales para. GitOps
Infraestructura inmutable	Spinnaker promueve el uso de una infraestr uctura inmutable, que es un concepto clave

Área	Capacidades de la herramienta
	en. GitOps Fomenta el despliegue de nuevas instancias en lugar de modificar las existentes.
Reversiones y control de versiones	Spinnaker ofrece sólidas capacidades de reversión y reversión rápida a estados buenos conocidos anteriormente. Soporta el control de versiones de las implementaciones, de conformidad con los principios de trazabilidad. GitOps
flujos de trabajo de aprobación	Spinnaker incluye etapas de evaluación manual en los procesos para respaldar los ascensos controlados entre entornos. Esto respalda GitOps las prácticas de separación entre las implementaciones y las versiones.
Canary y despliegues blue/green	Spinnaker apoya estrategias de despliegu e avanzadas que se alinean con GitOps las prácticas de liberación segura y controlada.
Integración con los sistemas de control de versiones	Spinnaker puede integrarse con varios proveedores de Git para iniciar canalizaciones en función de los eventos del repositorio.
Integración con Kubernetes	Spinnaker ofrece soporte nativo para Kubernetes y permite gestionar los recursos de Kubernetes de forma similar. GitOps
Gestión de artefactos	Spinnaker admite la gestión de artefactos y el control de versiones, que son fundamentales para mantener un flujo de trabajo. GitOps
Observabilidad y monitoreo	Spinnaker ofrece la integración con herramien tas de monitoreo para respaldar el aspecto de observabilidad de. GitOps

Área	Capacidades de la herramienta
Registro de auditoría	Spinnaker proporciona registros e historial de implementación detallados, que respaldan el principio de auditabilidad de. GitOps
Control de acceso basado en roles (RBAC)	Esta herramienta implementa el RBAC para controlar minuciosamente quién puede realizar qué acciones, de acuerdo con las prácticas de seguridad. GitOps
Creación de plantillas y parametrización	Spinnaker admite la creación de plantillas en las definiciones de canalización para permitir despliegues reutilizables y parametrizados.
Promoción del medio ambiente	Spinnaker facilita la promoción de aplicacio nes entre entornos (por ejemplo, desde la puesta en escena hasta la producción) de forma controlada.
Integración con herramientas de CI	Spinnaker puede integrarse con varias herramientas de integración continua (CI) para proporcionar una CI/CD cartera completa que cumpla con los principios. GitOps
Etapas y extensiones personalizadas	Esta herramienta admite etapas y extensiones personalizadas, para que los equipos puedan implementar GitOps flujos de trabajo que se adapten a sus necesidades.
Administración centralizada	Spinnaker proporciona una plataforma centraliz ada para gestionar las implementaciones en varios entornos y proveedores de nube.

Aunque Spinnaker no se comercializa principalmente como una GitOps herramienta, su flexibilidad y su sólido conjunto de funciones lo hacen capaz de implementar GitOps flujos de trabajo, especialmente en entornos complejos de múltiples nubes. La diferencia clave entre Spinnaker y

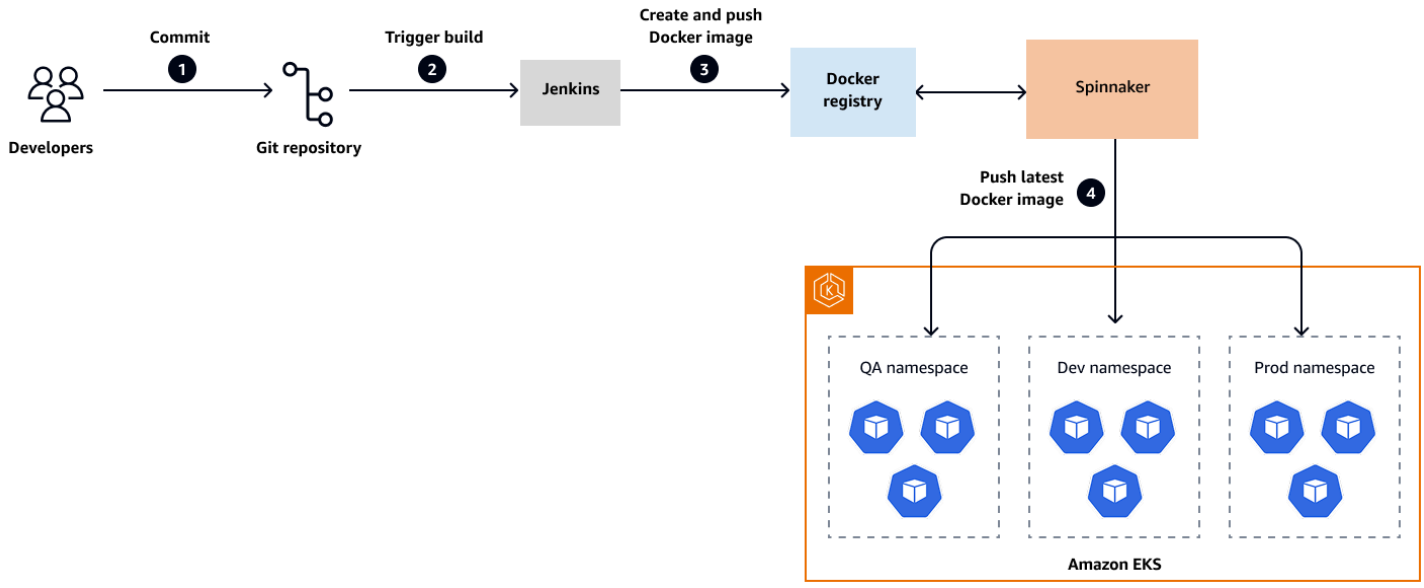
GitOps las herramientas dedicadas, como Argo CD o Flux, es que Spinnaker ofrece una plataforma de entrega continua más completa con estrategias de despliegue avanzadas y compatibilidad con múltiples nubes.

La fortaleza de Spinnaker reside en su capacidad para gestionar escenarios de despliegue complejos en varios proveedores de servicios en la nube y en su apoyo a las estrategias de despliegue avanzadas. Cuando Spinnaker se configura correctamente, puede implementar los principios de forma eficaz. GitOps Esto lo convierte en una herramienta poderosa para las organizaciones que desean adoptar GitOps prácticas en entornos diversos y complejos.

Para obtener más información, consulte la documentación de Spinnaker.

Arquitectura

El siguiente diagrama ilustra un flujo de trabajo GitOps de CD basado en discos compactos que utiliza Spinnaker y Jenkins X. Para obtener información detallada, consulte la documentación de Spinnaker.



donde:

- Paso 1: confirmación del código. Los desarrolladores confirman los cambios en el código de la aplicación en un repositorio de Git. Estos cambios podrían incluir actualizaciones de la propia aplicación, de Dockerfiles o de los manifiestos de Kubernetes.
- Paso 2: Compilación y creación de imágenes de Jenkins. El repositorio de Git activa Jenkins automáticamente mediante un webhook o una encuesta. Jenkins crea la aplicación, crea una

imagen de Docker y la envía a un registro de Docker configurado (como Amazon ECR o Docker Hub).

- Paso 3: Monitorización de imágenes y activación de canalizaciones mediante Spinnaker. Spinnaker monitorea continuamente el registro de Docker en busca de nuevas imágenes. Cuando se detecta una nueva versión de la imagen, Spinnaker activa automáticamente una canalización para iniciar el proceso de despliegue.
- Paso 4: Despliegue en los espacios de nombres de destino. Spinnaker implementa la nueva imagen de Docker en Amazon EKS. Según las configuraciones de canalización, la imagen se implementa en los espacios de nombres de destino del clúster. Spinnaker se asegura de que se implemente la última versión de la aplicación y, al mismo tiempo, sigue las estrategias de implementación definidas, como las implementaciones alternativas. blue/green

Flota Rancher

Rancher Fleet es una GitOps-at-scale solución diseñada específicamente para administrar varios clústeres de Kubernetes. Se adhiere estrictamente a los GitOps principios y, al mismo tiempo, se centra en la escalabilidad y la gestión de varios clústeres.

GitOps soporte

Área	Capacidades de la herramienta
Git como única fuente de información	Fleet utiliza los repositorios de Git como fuente autorizada para definir el estado deseado de las aplicaciones y los recursos en varios clústeres. Todas las configuraciones, incluidos los manifiestos de Kubernetes, los gráficos de Helm y los recursos personalizados, se almacenan en Git.
Configuración declarativa	Fleet trabaja con descripciones declarativas del estado deseado para las aplicaciones y los recursos. Pueden ser archivos YAML de Kubernetes sin procesar, gráficos de Helm, archivos de Kustomize o recursos personali zados específicos de Fleet.

Área	Capacidades de la herramienta
Sincronización automatizada	Fleet monitorea continuamente los repositorios de Git para detectar cambios. Aplica automáticamente los cambios a los clústeres de destino cuando detecta diferencias entre el estado de Git y el estado del clúster.
Administración de varios clústeres	Fleet está diseñada específicamente para gestionar las implementaciones en varios clústeres de Kubernetes. Puede gestionar miles de clústeres desde un único plano de control.
Arquitectura nativa de Kubernetes	Fleet está diseñada como un conjunto de controladores y recursos personalizados de Kubernetes. Utiliza los mecanismos de extensión de Kubernetes para sus operaciones. GitOps
Reconciliación continua	Fleet compara constantemente el estado real de los clústeres con el estado deseado definido en Git. Corrige automáticamente cualquier desviación detectada entre estos estados.
Agrupación y segmentación de clústeres	Fleet le permite agrupar clústeres y orientar las implementaciones a grupos específicos o clústeres individuales. Soporta un despliegue uniforme de aplicaciones en diferentes entornos y tipos de clústeres.
Configuraciones en capas	Fleet admite configuraciones en capas, que proporcionan configuraciones básicas con superposiciones específicas para el entorno. Esto se alinea con las GitOps prácticas de administración eficiente de múltiples entornos.

Área	Capacidades de la herramienta
Integración de Helm	Fleet proporciona soporte nativo para los gráficos de Helm y proporciona una gestión sencilla de aplicaciones complejas. Puede versionar y gestionar las versiones de Helm mediante GitOps flujos de trabajo.
Definiciones de recursos personalizadas (CRDs)	Fleet utiliza recursos personalizados, como GitRepo un paquete, para definir las implement aciones. Estos CRDs proporcionan una forma nativa de Kubernetes de definir los flujos de trabajo. GitOps
Seguridad y RBAC	Fleet se integra con el RBAC de Kubernetes para el control de acceso. Soporta la gestión segura de información y credenciales confidenc iales.
Observabilidad	Fleet proporciona información de estado sobre el estado de sincronización de los clústeres y las aplicaciones. Ofrece información sobre GitOps los procesos de toda la flota de clústeres.
Escalabilidad	Fleet está diseñada para ampliarse y gestionar miles de clústeres de forma eficiente. Soporta GitOps operaciones a gran escala en entornos empresariales.
Administración de dependencias	Puede definir las dependencias entre diferente s recursos y aplicaciones. Fleet garantiza que se siga el orden correcto de las operaciones en despliegues complejos.

Área	Capacidades de la herramienta
Personalización y extensibilidad	Fleet admite scripts personalizados y enlaces de ciclo de vida para una personalización avanzada de las implementaciones. Permite la integración con las herramientas y los flujos de trabajo existentes.
Soporte sin conexión y sin conexión a Internet	Fleet puede operar en entornos con conectividad a Internet limitada o nula. Soporta GitOps flujos de trabajo en entornos regulados o de alta seguridad.
Implementaciones progresivas	Fleet admite los despliegues escalonados en todos los clústeres, lo que permite aplicar estrategias de despliegue controladas y graduales.
Interfaz de administración unificada	Fleet proporciona una interfaz única para gestionar los GitOps flujos de trabajo en todos los clústeres. Simplifica las operaciones en entornos complejos de varios clústeres.
Integración con otras herramientas de Rancher	Fleet se integra con otras herramientas de Rancher para proporcionar una solución integral de administración de Kubernetes.
Registro de auditoría y cumplimiento	Fleet mantiene un registro de auditoría claro de todos los cambios e implementaciones. Le ayuda a cumplir los requisitos de conformidad mediante operaciones basadas en Git y controladas por versiones.

Rancher Fleet implementa estos GitOps principios con un fuerte enfoque en la escalabilidad y la administración de múltiples clústeres. Su diseño es especialmente adecuado para las organizaciones que administran una gran cantidad de clústeres de Kubernetes en diferentes entornos, centros de datos o proveedores de nube.

El principal elemento diferenciador de Fleet es su capacidad para gestionarlos a escala. GitOps Esta función la hace especialmente valiosa para las grandes empresas o los proveedores de servicios gestionados que gestionan numerosos clústeres. Herramientas como Argo CD o Flux se utilizan a menudo para la gestión de clústeres individuales, mientras que Fleet está diseñada para gestionar GitOps una gran flota de clústeres.

Al seguir estos GitOps principios, Rancher Fleet ofrece una solución para las organizaciones que desean implementar una gestión coherente, escalable y automatizada de las aplicaciones y los recursos en un entorno de Kubernetes diverso y a gran escala.

Para obtener más información, consulte la documentación de Fleet.

Arquitectura

Para obtener información sobre la arquitectura y el flujo de trabajo, consulte el repositorio de Fleet.

Codefresh

Codefresh es una CI/CD plataforma moderna que respalda GitOps los principios, especialmente para las implementaciones de Kubernetes. Codefresh ofrece un conjunto completo de CI/CD funciones y sus capacidades son notables. GitOps

GitOps soporte

Área	Capacidades de la herramienta
Git como única fuente de información	Codefresh utiliza los repositorios de Git como fuente autorizada para el código de las aplicaciones, las definiciones de infraestr uctura y las configuraciones de canalización. Todos los cambios en el sistema se realizan a través de Git, lo que garantiza un historial y un registro de auditoría completos.
Configuración declarativa	Codefresh admite definiciones de canalizac ión declarativas mediante archivos YAML que se almacenan en Git. Los manifiestos de Kubernetes, los gráficos de Helm, las

Área	Capacidades de la herramienta
	CloudFormation plantillas y otros archivos de IaC se pueden controlar desde el punto de vista de las versiones en los mismos repositor ios.
GitOps panel de control	Codefresh proporciona un GitOps panel dedicado para visualizar y gestionar GitOps los flujos de trabajo. Ofrece una visión clara del estado de sincronización entre Git y los estados del clúster.
Sincronización automatizada	Codefresh monitorea continuamente los repositorios de Git para detectar cambios. Inicia automáticamente las canalizaciones para aplicar los cambios a los entornos de destino cuando detecta diferencias.
Integración con Kubernetes	Codefresh ofrece una integración profunda con Kubernetes para admitir despliegues de estilo propio en varios clústeres. GitOps Es compatible con varios recursos de Kubernetes y con definiciones de recursos personalizadas (). CRDs
Gestión del entorno	Puede definir y administrar varios entornos (como el de desarrollo, puesta en escena y producción) como código. Codefresh apoya la promoción entre entornos mediante el uso GitOps de prácticas.
Integración con Argo CD	Codefresh se integra con Argo CD para mejorar las capacidades. GitOps Combina sus capacidades de CI con las ventajas del CD de Argo CD para ofrecer una solución completa. GitOps

Área	Capacidades de la herramienta
Soporte de Helm	Codefresh es compatible con los gráficos de Helm y proporciona una gestión sencilla de aplicaciones complejas. GitOps También ofrece el control de versiones y la promoción de gráficos de Helm.
Entrega progresiva	Codefresh admite estrategias de despliegu e avanzadas, como Canary y blue/green despliegues. Puede implementar y gestionar estas estrategias a través GitOps de flujos de trabajo.
Reversiones y control de versiones	Codefresh permite retroceder fácilmente a versiones anteriores si se detectan problemas después de la implementación. Mantiene el control de versiones de la implementación para garantizar la trazabilidad.
flujos de trabajo de aprobación	Codefresh admite procesos de aprobación manuales y automatizados para las implement aciones. Permite realizar promociones controladas entre entornos, de conformidad con GitOps las prácticas.
IaC	Codefresh admite la integración con herramien tas de iAC como Terraform CloudFormation . Permite el control de versiones de las definicio nes de infraestructura junto con el código de la aplicación.
Observabilidad y monitoreo	Codefresh proporciona funciones integradas de monitoreo y observabilidad. También ofrece integraciones con herramientas de monitoreo externas para mejorar la visibilidad.

Área	Capacidades de la herramienta
Análisis de seguridad	Codefresh incluye capacidades de escaneo de seguridad que se pueden integrar en GitOps los flujos de trabajo. Los controles de seguridad son parte del proceso de despliegue automatiz ado.
Registros de seguimiento de auditoría	Codefresh mantiene registros de auditoría exhaustivos para todas las acciones y cambios. Respalda los aspectos de trazabilidad y cumplimiento de. GitOps
RBAC y control de acceso	Codefresh implementa el control de acceso basado en roles (RBAC) para una gestión de permisos detallada. Esto ayuda a garantizar la seguridad de las operaciones en todos los equipos y entornos. GitOps
GitOps Automation	Codefresh ofrece funciones para automatizar varios aspectos de los GitOps flujos de trabajo, incluida la creación y fusión de solicitudes de extracción (PR).
Implementaciones híbridas y multinube	Codefresh admite GitOps flujos de trabajo en varios proveedores de nube y entornos locales.
Creación de plantillas y parametrización	Codefresh admite plantillas en configuraciones de canalización e implementación. Esto permite flujos de trabajo reutilizables y parametrizados GitOps.
Gestión de imágenes integrada	Codefresh proporciona capacidades integrada s de administración de imágenes de contenedo res. Integra la creación y el despliegue de imágenes en GitOps los flujos de trabajo.

Área	Capacidades de la herramienta
GitOps para la gestión de secretos	Codefresh ofrece formas seguras de gestionar los secretos dentro de los GitOps flujos de trabajo. Se integra con soluciones externas de gestión de secretos.
Funciones de colaboración	Codefresh proporciona funciones para la colaboración en equipo dentro de GitOps los procesos. Estas funciones incluyen comentari os, notificaciones y paneles compartidos.

El enfoque de Codefresh destaca por su integración de las capacidades de CI/CD con las prácticas. GitOps GitOps Su objetivo es proporcionar una plataforma integral que cubra todo el ciclo de vida de la entrega del software y, al mismo tiempo, cumpla con los principios. GitOps

El principal elemento diferenciador de Codefresh en la GitOps zona es su enfoque de plataforma unificada, que combina las capacidades de CI con el CD y las funciones. GitOps Esto lo hace especialmente adecuado para los equipos que desean una all-in-one solución que pueda gestionar CI/CD escenarios complejos y, al mismo tiempo, implementar GitOps prácticas.

Codefresh ofrece una plataforma para las organizaciones que desean adoptar GitOps metodologías en un CI/CD contexto más amplio, especialmente cuando trabajan con Kubernetes y tecnologías nativas de la nube.

Para obtener más información, consulte la documentación de Codefresh.

Pulumi

Pulumi es una plataforma iAC que no está diseñada exclusivamente para ello. GitOps Sin embargo, se puede usar de manera eficaz para implementar GitOps principios, especialmente para la infraestructura de nube y las implementaciones de Kubernetes.

GitOps soporte

Área	Capacidades de la herramienta
IaC	Pulumi le permite definir su infraestructura mediante el uso de lenguajes de programac ión de uso general, TypeScript como Python y Go. Este enfoque basado en código se alinea con el GitOps énfasis en las configuraciones declarativas y versionadas.
Git como única fuente de información	El código de infraestructura de Pulumi se puede almacenar y controlar sus versiones en los repositorios de Git. Esto garantiza que Git sirva como la única fuente de información fiable para las definiciones de infraestructura.
Estado declarativo deseado	Aunque Pulumi utiliza lenguajes de programac ión, sigue describiendo el estado deseado de la infraestructura de forma declarativa. El código define el aspecto que debe tener la infraestr uctura, no el step-by-step proceso para crearla.
Sincronización automatizada	Pulumi se puede integrar con CI/CD canalizac iones para aplicar automáticamente los cambios cuando el código se actualiza en Git. Esto permite el despliegue continuo de los cambios en la infraestructura, lo cual es un principio clave. GitOps
Soporte multinube y Kubernetes	Pulumi es compatible con una amplia gama de proveedores de nube y Kubernetes, por lo que puede seguir las prácticas en diversos entornos. GitOps La herramienta permite una gestión coherente de los recursos en diferentes plataformas.

Área	Capacidades de la herramienta
Administración de estados	Pulumi gestiona el estado de la infraestructura, que se puede almacenar de forma remota y segura. Esta gestión del estado es crucial para GitOps las prácticas y garantiza la coherencia entre el estado definido y el estado real de la infraestructura.
Detección y reconciliación de desviaciones	Pulumi puede detectar diferencias entre el estado deseado (en código) y el estado real de la infraestructura. Reconcilia estas diferencias de acuerdo con el GitOps principio de reconcili ación continua.
La política como código	Puede definir y hacer cumplir las políticas como código mediante Pulumi CrossGuard. Esto permite gestionar las políticas de conformidad y seguridad GitOps al estilo de las versiones.
Administración de secretos	Pulumi proporciona formas seguras de administrar la información confidencial dentro del código de infraestructura. Soporta la integración con sistemas de gestión de secretos externos, lo cual es crucial para las prácticas de GitOps seguridad.
Componentes modulares y reutilizables	Pulumi apoya la creación de componentes y módulos reutilizables. Esta modularidad se alinea con las GitOps prácticas de gestión de despliegues complejos en varios entornos.
Obtenga una vista previa y planifique	Pulumi ofrece la posibilidad de previsualizar los cambios antes de aplicarlos. Esto respalda el GitOps principio de cambios seguros y predecibles en la infraestructura.

Área	Capacidades de la herramienta
Retrocesos e historia	Pulumi mantiene un historial de despliegues y admite la reversión a estados anteriores. Esto se ajusta a los GitOps principios de trazabilidad y reversibilidad.
Entrega continua para la infraestructura	Pulumi se puede integrar en las CI/CD tuberías para ofrecer de forma continua los cambios en la infraestructura. Es compatible con las pruebas y la validación automatizadas del código de infraestructura.
RBAC y control de acceso	Pulumi proporciona un control de acceso basado en roles para administrar quién puede realizar cambios en la infraestructura. Esto respalda las prácticas de GitOps seguridad y gobierno.
Observabilidad y registro	Pulumi ofrece capacidades de registro y monitoreo de los cambios en la infraestructura. Estas capacidades respaldan el aspecto de observabilidad de las prácticas. GitOps
Integración con otras herramientas	Pulumi puede integrarse con varias herramien tas en la nube. Esta flexibilidad permite flujos de trabajo integrales GitOps.
Gestión del entorno	Pulumi admite la administración de múltiples entornos (desarrollo, puesta en escena, producción) mediante el uso de la misma base de código con diferentes configuraciones. Esto se alinea con GitOps las prácticas para una gestión coherente de varios entornos.

Área	Capacidades de la herramienta
Administración de dependencias	Pulumi gestiona las dependencias entre los recursos y garantiza el orden correcto de las operaciones. Esto es crucial para las GitOps implementaciones complejas que implican componentes interdependientes.
Proveedores de recursos personalizados	Pulumi te permite crear proveedores personalizados para gestionar cualquier servicio basado en API. Esto amplía GitOps las prácticas a una amplia gama de recursos más allá de las ofertas en la nube estándar.
Funciones de colaboración	Pulumi apoya la colaboración en equipo mediante controles de acceso y estado compartidos. Esto facilita los GitOps flujos de trabajo en los entornos de equipo.

Al utilizar estas funciones de Pulumi, las organizaciones pueden implementar GitOps prácticas para su infraestructura, especialmente en situaciones en las que necesitan un control detallado o una lógica compleja, o bien desean administrar un conjunto diverso de recursos locales y en la nube dentro de un marco único y coherente.

El enfoque de Pulumi GitOps es único porque aporta la potencia y la flexibilidad de los lenguajes de programación de uso general a la gestión de la infraestructura y, al mismo tiempo, se adhiere a los principios. GitOps Esto puede resultar especialmente ventajoso para los equipos que prefieren trabajar con lenguajes de programación conocidos y desean aplicar las mejores prácticas de ingeniería de software a la gestión de infraestructuras.

El principal elemento diferenciador de Pulumi GitOps es el uso de lenguajes de programación estándar para definir la infraestructura. GitOps Las herramientas tradicionales suelen utilizar lenguajes YAML o de dominios específicos, mientras que Pulumi permite utilizar una lógica más compleja, una mejor reutilización del código y una integración más sencilla con los flujos de trabajo de desarrollo existentes.

Para obtener más información, consulta la documentación de Pulumi.

GitOps comparación de herramientas

Esta es una comparación de las nueve GitOps herramientas que se analizaron en las secciones anteriores. Al elegir una herramienta, tenga en cuenta sus requisitos específicos, la infraestructura existente, la experiencia del equipo y el nivel deseado de control y personalización.

Facilidad de uso

- Por lo general, Argo CD, Flux y Rancher Fleet son más fáciles de configurar.
- Spinnaker y Jenkins X tienen curvas de aprendizaje más pronunciadas.
- GitOps Es posible que Weave requiera más configuración para las funciones avanzadas.
- GitLab CI/CD y Codefresh ofrecen experiencias integradas.

Integración con Kubernetes

- Argo CD, Flux y Rancher Fleet están muy centrados en Kubernetes.
- Jenkins X y Weave ofrecen capacidades más amplias. GitOps DevOps
- Las demás herramientas son compatibles con Kubernetes sin centrarse exclusivamente en él.

Capacidades de CI/CD

- Jenkins X, soluciones. GitLab CI/CD, and Codefresh offer complete CI/CD
- Argo CD, Flux y Weave GitOps se centran más en el aspecto CD del flujo de trabajo y, a menudo, requieren la integración con herramientas de CI independientes.

GitOps pureza

- Argo CD y Flux son herramientas que se centran específicamente en GitOps.
- Las otras herramientas incorporan GitOps principios en diversos grados.

Soporte multinube

- Spinnaker y Pulumi destacan en escenarios multicloud.
- Las demás herramientas pueden funcionar en todas las nubes, pero es posible que requieran una configuración adicional.

Compatibilidad con multiclústeres

- Todas las herramientas admiten despliegues de varios clústeres.
- Argo CD y Weave GitOps tienen funciones de administración de múltiples clústeres más avanzadas.

Integración

- Flux cuenta con el firme respaldo de la Cloud Native Computing Foundation (CNCF).
- Argo CD tiene una comunidad grande y activa.
- Argo CD y Flux tienen una sólida integración con Kubernetes.
- Jenkins X usa el sistema Jenkins, más amplio.
- Weave GitOps es más reciente, pero está creciendo con un fuerte respaldo comercial.
- GitLab CI/CD se integra estrechamente con. GitLab
- Rancher Fleet funciona bien dentro del sistema Rancher.

Comunidad y apoyo

- Flux cuenta con un fuerte respaldo de la CNCF.
- Argo GitLab, CD y Spinnaker tienen grandes comunidades.
- El soporte comercial está disponible para la mayoría de las herramientas.

Funciones empresariales

- De forma predeterminada, Weave GitOps y Jenkins X ofrecen más funciones centradas en la empresa.

- Argo CD y Flux tienen ofertas empresariales o pueden ampliarse para uso empresarial.

Flexibilidad y extensibilidad

- Flux es altamente modular y extensible.
- Argo CD ofrece buenas opciones de personalización.
- Jenkins X es muy extensible, pero puede requerir más esfuerzo.
- Weave GitOps tiene como objetivo proporcionar una solución completa con menos necesidad de extensibilidad.

Escalabilidad

- Spinnaker y GitLab CI/CD son conocidos por su escalabilidad empresarial.
- Argo CD y Flux gestionan bien las implementaciones de Kubernetes a gran escala.

Gestión de infraestructuras

- Pulumi se centra en la gestión de infraestructuras.
- Weave GitOps y Flux ofrecen buenas capacidades de iAC.

Soporte de lenguaje y modelo de programación

- En Pulumi, puede definir la infraestructura mediante lenguajes de programación de uso general, como Python, Go TypeScript, C# y Java. El uso de lenguajes estándar por parte de Pulumi permite la integración del código de infraestructura con flujos de trabajo de desarrollo conocidos, prácticas de prueba y lógica compleja.
- Terraform usa el lenguaje HashiCorp de configuración (HCL).
- CloudFormation usa plantillas JSON y YAML.
- Argo CD, Flux, Rancher Fleet, Weave GitOps, Spinnaker y GitLab CI/CD administran principalmente archivos de configuración declarativa o YAML.
- Jenkins X gestiona canalizaciones basadas en YAML y secuencias de comandos, pero no ofrece de forma nativa programación de uso general para la IaC.

Casos de uso de Argo CD y Flux

Esta sección se centra en dos herramientas, Argo CD y Flux, que proporcionan una funcionalidad pura GitOps. En este contexto, pure GitOps se refiere a un modelo en el que un repositorio de Git sirve como única fuente de información sobre el estado deseado de las aplicaciones y la infraestructura. Todos los cambios se realizan mediante confirmaciones de Git y el sistema sincroniza automáticamente el entorno en vivo para que coincida con el estado definido en el repositorio. No se requiere ninguna intervención manual fuera de las operaciones de Git.

Consideraciones generales

- Puede que prefieras usar Argo CD en entornos en los que la gestión visual y los flujos de trabajo centrados en las aplicaciones sean importantes.
- Puede elegir Flux si necesita soluciones más livianas, una sólida capacidad de arrendamiento múltiple o una integración profunda con la red más amplia de Cloud Native Computing Foundations (CNCF).
- Argo CD suele atraer a los equipos que están pasando de la CI/CD tradicional a la CI/CD tradicional debido a su interfaz de usuario intuitiva. GitOps
- Flux suele ser el preferido en los entornos nativos de la nube, donde los flujos de trabajo basados en CLI y las prácticas de IaC ya están establecidos.

En última instancia, la elección entre Argo CD y Flux suele depender de las necesidades organizativas específicas, de las herramientas existentes y de las preferencias del equipo. Ambas herramientas son capaces de gestionar la mayoría de los GitOps escenarios, por lo que le recomendamos que las evalúe en función de sus casos de uso y requisitos específicos.

Casos de uso de Argo CD

Gestión visual:

- Cuando necesite una interfaz de usuario fácil de usar para gestionar las implementaciones y visualizar los estados de las aplicaciones.
- Para los equipos que prefieren una interfaz gráfica para la supervisión y la solución de problemas.

Enfoque centrado en las aplicaciones:

- Cuando desee gestionar las implementaciones a nivel de aplicación en lugar de gestionar los recursos individuales.
- Para organizaciones que estructuran sus despliegues en torno a conceptos de aplicaciones.

Administración de varios clústeres:

- Cuando la gestión de las implementaciones en varios clústeres es un requisito principal.
- Para entornos complejos y distribuidos con muchos clústeres.

Reversión y sincronización de ondas:

- Cuando necesite un control detallado del proceso de implementación, incluidas las ondas de sincronización y las intervenciones manuales.
- Para escenarios que requieren estrategias de reversión complejas.

Integración con las herramientas existentes:

- Cuando ya utilizas otras herramientas del proyecto Argo, como Argo Workflows y Argo Events.

Entornos empresariales:

- Para grandes empresas que necesitan una sólida integración de RBAC e inicio de sesión único de forma predeterminada.

Casos de uso de Flux

Implementaciones ligeras:

- Cuando necesite una solución más ligera y que consuma menos recursos GitOps.
- Para escenarios de computación perimetral o IoT en los que los recursos pueden estar limitados.

Actualizaciones de imágenes automatizadas:

- Cuando la detección y el despliegue automáticos de nuevas imágenes de contenedores son un requisito clave.
- Para equipos que se centran en un despliegue continuo con actualizaciones frecuentes de las imágenes.

Tenencia múltiple:

- Cuando se necesita un sólido soporte para múltiples inquilinos, especialmente en entornos de clústeres compartidos.
- Para proveedores de servicios o grandes organizaciones que tienen una separación estricta entre equipos o proyectos.

IaC:

- Cuando es importante gestionar las aplicaciones y la infraestructura mediante el mismo GitOps flujo de trabajo.
- Para equipos que apuestan fuertemente por el paradigma de la IaC.

Integración del timón:

- Cuando el uso extensivo de los gráficos de Helm forma parte de su estrategia de despliegue.
- Para entornos que tienen despliegues complejos basados en Helm.

Integración del proyecto CNCF:

- Cuando es importante una estrecha integración con otros proyectos de la CNCF.
- Para organizaciones que se alinean con las tecnologías y los principios de la CNCF.

Arquitectura modular:

- Cuando necesite flexibilidad para usar solo componentes específicos del GitOps kit de herramientas.
- Para equipos que desean crear GitOps flujos de trabajo personalizados mediante componentes modulares.

Entrega progresiva:

- Cuando las estrategias de despliegue avanzadas, como las versiones preliminares o A/B las pruebas, son requisitos fundamentales.

Comparación de características

Área	Argo CD	Flujo
Support for core GitOps principles	✅Sí	✅Sí
Arquitectura	End-to-end aplicación para implementar flujos de trabajo de Kubernetes GitOps	Proporciona CRDs Kubernetes y controladores para GitOps
Configuración	Sencillez	Complejo
Soporte de timón	✅Sí	✅Sí
Personaliza el soporte	✅Sí	✅Sí
GUI integrada	CLI e interfaz de usuario web con todas las funciones	CLI e interfaz web ligera opcional
Soporte RBAC	Control granular	RBAC nativo de Kubernetes
Soporte para varios usuarios y varios clústeres	Excelente soporte para múltiples clústeres	Excelente soporte para la multitenencia
Autenticación mediante inicio de sesión único	✅Sí	✅Sí
Automatización de la sincronización	Posibilidad de sincronizar ventanas	Posibilidad de establecer intervalos de reconciliación
Sincronización parcial	✅Sí	❌No

Área	Argo CD	Flujo
Proceso de reconciliación	Soporta sincronizaciones manuales y automáticas. Hay varias estrategias diferentes disponibles.	Soporta sincronizaciones manuales y automáticas.
Extensibilidad	Soporta plugins personalizados. Opciones de personalización limitadas.	Soporta un mando personalizado. Buena extensibilidad e integraciones de terceros.
Apoyo de la comunidad	Comunidad grande y activa.	Comunidad más pequeña pero en crecimiento.
Escalabilidad	Buena escalabilidad, pero limitada por la velocidad de obtención de datos de la interfaz de usuario web. Los análisis de la comunidad sugieren que son compatibles con decenas de miles de aplicaciones.	Guías claras para la escalabilidad horizontal y vertical, hasta decenas de miles de aplicaciones.

Mejores prácticas para elegir una GitOps herramienta

En esta sección se proporcionan consideraciones, consejos y prácticas recomendadas para elegir una GitOps herramienta para su clúster de EKS. La elección correcta depende del contexto específico, los requisitos y la estrategia a largo plazo. Suele ser beneficioso realizar una prueba de concepto con las mejores opciones antes de tomar una decisión final.

Evalúe las necesidades y capacidades de su organización:

- Ten en cuenta las habilidades actuales de tu equipo y su disposición a aprender nuevas herramientas.
- Evalúe la complejidad de su entorno Amazon EKS. (Por ejemplo, ¿utiliza uno o varios clústeres?)
- Determine sus requisitos específicos de conformidad, seguridad y escalabilidad.

> **Práctica recomendada**
>
> Cree un documento de requisitos detallado que describa las características requeridas y las capacidades útiles, pero no obligatorias.

Evalúe la madurez y la adopción de la herramienta:

- Investigue la madurez de GitOps las posibles herramientas y sus tasas de adopción en la industria.
- Busque herramientas que tengan un historial comprobado en los entornos de Amazon EKS.

> **Práctica recomendada**
>
> Priorice las herramientas que se hayan adoptado ampliamente y que tengan una fuerte presencia en la red de la Cloud Native Computing Foundation (CNCF).

Considere la posibilidad de integrarla con su cadena de herramientas actual:

- Evalúe qué tan bien se integra la GitOps herramienta con su CI/CD cartera actual, sus soluciones de monitoreo y otras herramientas operativas.
- Busque integraciones nativas con Servicios de AWS IAM, Amazon ECR y. CloudWatch

> **Práctica recomendada**
>
> Cree una prueba de concepto para probar las capacidades de integración antes de tomar una decisión final.

Evalúe las características de seguridad:

- Priorice las herramientas que tengan sólidas capacidades de control de acceso basado en roles (RBAC) y que se integren bien con la IAM.
- Busque funciones que respalden la gestión segura de los secretos y la aplicación de políticas.

> **Práctica recomendada**
>
> Elija una herramienta que respalde las prácticas de seguridad GitOps basadas en las normas, incluidas las políticas en forma de código y las comprobaciones de cumplimiento automatizadas.

Evalúe la escalabilidad y el rendimiento:

- Tenga en cuenta el rendimiento de la herramienta con un gran número de aplicaciones y clústeres.
- Evalúe su impacto en el rendimiento del clúster y el consumo de recursos.

> **Práctica recomendada**
>
> Realice pruebas de rendimiento con cargas de trabajo similares a las de su entorno de producción para asegurarse de que la herramienta puede adaptarse a su escala.

Considere la posibilidad de ofrecer soporte para varios clústeres y entornos:

- Si tiene o planea tener varios clústeres de EKS, priorice las herramientas que tengan sólidas capacidades de administración de varios clústeres.
- Busque funciones que admitan implementaciones uniformes en diferentes entornos (como el desarrollo, la puesta en escena y la producción).

> **Práctica recomendada**
>
> Elija una herramienta que permita la administración centralizada de varios clústeres y, al mismo tiempo, mantenga las configuraciones específicas del entorno.

Evalúe las capacidades de observabilidad y monitoreo:

- Busque herramientas que ofrezcan una visibilidad clara del estado de sus despliegues y del estado del clúster.
- Considere qué tan bien se integra la herramienta con sus soluciones de monitoreo y registro existentes.

> **Práctica recomendada**
>
> Priorice las herramientas que ofrecen paneles y mecanismos de alerta personalizables para la detección proactiva de problemas.

Evalúe la curva de aprendizaje y la documentación:

- Evalúe la calidad y la exhaustividad de la documentación de la herramienta.
- Considere la disponibilidad de recursos de capacitación y el apoyo de la comunidad.

> **Práctica recomendada**
>
> Elija una herramienta que tenga una documentación bien mantenida, foros comunitarios activos y programas de formación o certificaciones oficiales.

Tenga en cuenta el costo y la utilización de los recursos:

- Evalúe tanto los costos directos (como las licencias y el soporte) como los costos indirectos (como los gastos generales operativos y los costos de capacitación) de la adopción de la herramienta.
- Evalúe la eficiencia de la herramienta en términos de consumo de recursos informáticos y de almacenamiento.

> **Práctica recomendada**
>
> Realice un análisis del costo total de propiedad (TCO) que incluya los costos a corto y largo plazo.

Evalúe las opciones de flexibilidad y personalización:

- Busque herramientas que le permitan personalizar los flujos de trabajo para adaptarlos a sus necesidades específicas.
- Tenga en cuenta la extensibilidad de la herramienta mediante complementos o APIs.

> **Práctica recomendada**
>
> Elija una herramienta que equilibre la funcionalidad predeterminada con la posibilidad de personalizarla según sus requisitos únicos.

Evalúe las capacidades de entrega continua e implementación progresiva:

- Busque herramientas que respalden estrategias de implementación avanzadas, como las versiones e blue/green implementaciones estándar.
- Evalúe la facilidad de implementar y administrar estas estrategias.

> **Práctica recomendada**
>
> Priorice las herramientas que ofrecen soporte integrado para los patrones de entrega progresivos a fin de minimizar el riesgo en sus implementaciones.

Considere la posibilidad de depender de un proveedor y de portabilidad:

- Evalúe las dependencias de la herramienta con respecto a proveedores o tecnologías de nube específicos.
- Considere la facilidad de migrar a una herramienta diferente en el futuro si es necesario.

> **Práctica recomendada**
>
> Prefiera las herramientas que utilizan estándares abiertos y ofrecen capacidades de exportación para sus GitOps configuraciones.

Evalúe el apoyo y las extensiones de la comunidad:

- Observe el tamaño y la actividad de la comunidad de usuarios.
- Evalúe la disponibilidad de integraciones y complementos de terceros.

> **Práctica recomendada**
>
> Únase a los foros de la comunidad o a los grupos de usuarios para obtener experiencias de primera mano de otros usuarios antes de tomar una decisión.

Tenga en cuenta los requisitos de conformidad y auditoría:

- Evalúe en qué medida la herramienta responde a sus necesidades de cumplimiento, incluidos los registros de auditoría y los informes.
- Busque funciones que le ayuden a mantener y demostrar el cumplimiento.

> **Práctica recomendada**
>
> Elija una herramienta que proporcione registros de auditoría completos y permita generar informes de conformidad.

Evalúe las capacidades de reversión y recuperación ante desastres:

- Evalúe la facilidad y confiabilidad de los mecanismos de reversión.
- Considere cómo la herramienta admite los escenarios de recuperación ante desastres.

Práctica recomendada

Pruebe minuciosamente los procesos de reversión y recuperación como parte de su evaluación.

Preguntas frecuentes

P: ¿Cuáles son las GitOps herramientas más populares para Amazon EKS?

R: Las GitOps herramientas más populares de Amazon EKS incluyen Argo CD, Flux, Jenkins X y GitLab CI/CD. Cada herramienta tiene puntos fuertes, pero Argo CD y Flux son especialmente apreciadas por su enfoque nativo de Kubernetes y su sólido apoyo por parte de la comunidad.

P: ¿Cómo se mejora la gestión de clústeres de EKS? GitOps

R: GitOps mejora la administración de los clústeres de EKS al proporcionar control de versiones para la infraestructura, implementaciones automatizadas, mayor seguridad mediante configuraciones declarativas, reversiones más sencillas y mejor auditabilidad. También mejora la colaboración y reduce los errores humanos en las implementaciones.

P: ¿Qué características clave debo buscar en una GitOps herramienta para Amazon EKS?

R: Entre las principales características a tener en cuenta se incluyen: una integración perfecta con Amazon EKS, un RBAC sólido, compatibilidad con varios clústeres, funciones de observabilidad, compatibilidad con estrategias de entrega progresiva, escalabilidad e integración con IAM y Amazon Servicios de AWS ECR.

P: ¿Cómo garantizo la seguridad al realizar la implementación GitOps en Amazon EKS?

R: Para garantizar la seguridad, elige una herramienta que tenga una sólida integración de RBAC con IAM, una gestión segura de secretos, soporte para repositorios Git cifrados y la capacidad de implementar políticas de seguridad como código. Además, compruebe que la herramienta proporciona registros de auditoría completos.

P: ¿Pueden GitOps las herramientas gestionar entornos Amazon EKS de varios clústeres?

R: Sí, GitOps herramientas como Argo CD y Flux cuentan con sólidas capacidades de administración de varios clústeres. Permiten gestionar varios clústeres de EKS desde un único plano de control, lo que garantiza la coherencia en todos los entornos.

P: ¿Cómo se integran GitOps las herramientas con las canalizaciones de CI/CD existentes?

R: Por lo general, GitOps las herramientas se integran con las canalizaciones de CI/CD existentes y actúan como la etapa de despliegue de la canalización. Las herramientas de CI pueden activarlos

cuando los cambios se envían al repositorio de Git y automatizan el proceso de implementación en los clústeres de EKS.

P: ¿Cuáles son los desafíos de la implementación GitOps en Amazon EKS?

R: Los desafíos más comunes incluyen administrar los secretos de forma segura, garantizar los controles de acceso adecuados, gestionar las aplicaciones con estado, gestionar la desviación entre Git y el estado del clúster y adaptar los flujos de trabajo del equipo al GitOps modelo.

P: ¿Cómo gestionan GitOps las herramientas las reversiones en Amazon EKS?

R: GitOps Las herramientas suelen gestionar las reversiones volviendo a una confirmación anterior en el repositorio de Git. Esto desencadena automáticamente un despliegue en el estado correcto conocido anteriormente, lo que se traduce en reversiones rápidas y fiables.

P: ¿Pueden GitOps las herramientas gestionar los complementos y otros AWS recursos de Amazon EKS?

R: Muchas GitOps herramientas pueden administrar los complementos y algunos AWS recursos de Amazon EKS, especialmente cuando se combinan con herramientas de iAC como Terraform o. CloudFormation Sin embargo, el alcance de esta capacidad puede variar; consulte la sección de GitOps herramientas para obtener información específica sobre cada herramienta.

P: ¿Cómo cumplen GitOps las herramientas los requisitos de conformidad de Amazon EKS?

R: GitOps Las herramientas respaldan el cumplimiento al proporcionar un registro de auditoría claro de todos los cambios, hacer cumplir los procesos de aprobación, implementar la política como código para las comprobaciones de cumplimiento automatizadas y ofrecer funciones detalladas de registro e informes.

P: ¿Cuál es la curva de aprendizaje necesaria para la implementación GitOps en Amazon EKS?

R: La curva de aprendizaje puede variar en función de la herramienta y de los conocimientos actuales de su equipo. Por lo general, los equipos que están familiarizados con Git, Kubernetes y Amazon EKS se adaptarán más rápido que otros. Las herramientas más populares ofrecen una amplia documentación y recursos de formación para facilitar su adopción.

P: ¿Cómo gestionan GitOps las herramientas la administración de secretos en Amazon EKS?

R: GitOps Las herramientas suelen integrarse con soluciones externas de administración de secretos, como AWS Secrets Manager HashiCorp Vault. Algunas herramientas también ofrecen cifrado integrado para los secretos que se almacenan en los repositorios de Git.

P: ¿ GitOps Las herramientas pueden funcionar tanto con aplicaciones sin estado como con aplicaciones con estado en Amazon EKS?

R: Sí, GitOps las herramientas pueden funcionar tanto con aplicaciones con estado como con aplicaciones con estado. Sin embargo, la administración de aplicaciones con estado a menudo requiere consideraciones adicionales, como la gestión de los volúmenes persistentes y la garantía de la coherencia de los datos durante las actualizaciones.

P: ¿Cómo son compatibles GitOps las herramientas con Canary o blue/green las implementaciones en Amazon EKS?

R: Muchas GitOps herramientas ofrecen soporte integrado para estrategias de implementación avanzadas. Pueden gestionar el lanzamiento gradual de nuevas versiones, supervisar los problemas y revertirlos automáticamente si se detectan problemas. Todas estas operaciones se definen como código en el repositorio de Git.

P: ¿Cuál es la diferencia entre usar una GitOps herramienta y usarla **`kubectl apply`** con una CI/CD canalización?

R: Las GitOps herramientas ofrecen ventajas en comparación con `kubectl apply` los comandos simples, como la detección y la conciliación automatizadas de las desviaciones, la mejora de la seguridad mediante las implementaciones basadas en la extracción, una mejor auditabilidad y estrategias de implementación más sofisticadas. También proporcionan un enfoque más integral para gestionar todo el estado del clúster.

Recursos

Los siguientes recursos proporcionan documentación oficial, guías prácticas, estudios de casos y análisis detallados que pueden ayudarle a tomar una decisión informada a la hora de elegir una GitOps herramienta para su clúster de EKS. Abarcan varios aspectos GitOps, como las estrategias de implementación, las mejores prácticas, las comparaciones entre diferentes herramientas y las experiencias del mundo real.

AWS recursos:

- Documentación de Amazon EKS
- Automatizar Amazon EKS con GitOps (AWS entrada de blog)
- Introducción a un GitOps EKS con Weaveworks (taller)AWS
- Laboratorio Flux (taller de Amazon EKS)
- Laboratorio Argo CD (taller de Amazon EKS)

GitOps y documentación de la herramienta:

- GitOps Mejores prácticas para un despliegue continuo y una seguridad progresiva (seminario web a DevOps pedido en línea)
- Documentación de Kubernetes
- Documentación de Argo CD
- Documentación de Flux
- Documentación de Weave GitOps
- Documentación de Jenkins X
- GitLab CI/CD documentación
- Documentación de Spinnaker
- Documentación de Rancher Fleet
- Documentación de Codefresh

Historial de documentos

En la siguiente tabla, se describen cambios significativos de esta guía. Si quiere recibir notificaciones de futuras actualizaciones, puede suscribirse a las notificaciones RSS.

Cambio	Descripción	Fecha
Publicación inicial	—	30 de abril de 2025

AWS Glosario de orientación prescriptiva

Los siguientes son términos de uso común en las estrategias, guías y patrones proporcionados por la Guía AWS prescriptiva. Para sugerir entradas, utilice el enlace Enviar comentarios al final del glosario.

Números

Las 7 R

Siete estrategias de migración comunes para trasladar aplicaciones a la nube. Estas estrategias se basan en las 5 R que Gartner identificó en 2011 y consisten en lo siguiente:

- Refactorizar/rediseñar: traslade una aplicación y modifique su arquitectura mediante el máximo aprovechamiento de las características nativas en la nube para mejorar la agilidad, el rendimiento y la escalabilidad. Por lo general, esto implica trasladar el sistema operativo y la base de datos. Ejemplo: Migrar la base de datos de Oracle en las instalaciones a Amazon Aurora PostgreSQL-Compatible Edition.
- Redefinir la plataforma (transportar y redefinir): traslade una aplicación a la nube e introduzca algún nivel de optimización para aprovechar las capacidades de la nube. Ejemplo: Migrar la base de datos Oracle en las instalaciones a Amazon Relational Database Service (Amazon RDS) para Oracle en la nube de Nube de AWS.
- Recomprar (readquirir): cambie a un producto diferente, lo cual se suele llevar a cabo al pasar de una licencia tradicional a un modelo SaaS. Ejemplo: Migrar el sistema de administración de las relaciones con los clientes (CRM) a Salesforce.com.
- Volver a alojar (migrar mediante lift-and-shift): traslade una aplicación a la nube sin realizar cambios para aprovechar las capacidades de la nube. Ejemplo: Migrar la base de datos de Oracle en las instalaciones a Oracle en una instancia de EC2 en la Nube de AWS.
- Reubicar: (migrar el hipervisor mediante lift and shift): traslade la infraestructura a la nube sin comprar equipo nuevo, reescribir aplicaciones o modificar las operaciones actuales. Los servidores se migran de una plataforma en las instalaciones a un servicio en la nube para la misma plataforma. Ejemplo: migrar una Microsoft Hyper-V aplicación a AWS.
- Retener (revisitar): conserve las aplicaciones en el entorno de origen. Estas pueden incluir las aplicaciones que requieren una refactorización importante, que desee posponer para más adelante, y las aplicaciones heredadas que desee retener, ya que no hay ninguna justificación empresarial para migrarlas.

- Retirar: retire o elimine las aplicaciones que ya no sean necesarias en un entorno de origen.

A

ABAC

Consulte control de acceso basado en atributos.

servicios abstractos

Consulte servicios administrados.

ACID

Consulte atomicidad, consistencia, aislamiento, durabilidad.

migración activa-activa

Método de migración de bases de datos en el que las bases de datos de origen y destino se mantienen sincronizadas (mediante una herramienta de replicación bidireccional o mediante operaciones de escritura doble) y ambas bases de datos gestionan las transacciones de las aplicaciones conectadas durante la migración. Este método permite la migración en lotes pequeños y controlados, en lugar de requerir una transición única. Es más flexible, pero requiere más trabajo que una migración activa-pasiva.

migración activa-pasiva

Método de migración de bases de datos en el que las bases de datos de origen y destino se mantienen sincronizadas, pero solo la de origen gestiona las transacciones de las aplicaciones conectadas, mientras los datos se replican en la de destino. La base de datos de destino no acepta ninguna transacción durante la migración.

función de agregación

Función SQL que actúa en un grupo de filas y calcula un único valor de devolución para el grupo. Entre los ejemplos de funciones de agregación se incluyen `SUM` y `MAX`.

IA

Consulte inteligencia artificial.

AIOps

Consulte operaciones de inteligencia artificial

anonimización

El proceso de eliminar permanentemente la información personal de un conjunto de datos. La anonimización puede ayudar a proteger la privacidad personal. Los datos anonimizados ya no se consideran datos personales.

antipatrones

Una solución que se utiliza con frecuencia para un problema recurrente en el que la solución es contraproducente, ineficaz o menos eficaz que una alternativa.

control de aplicaciones

Enfoque de seguridad que permite usar de manera exclusiva aplicaciones aprobadas para ayudar a proteger un sistema contra el malware.

cartera de aplicaciones

Recopilación de información detallada sobre cada aplicación que utiliza una organización, incluido el costo de creación y mantenimiento de la aplicación y su valor empresarial. Esta información es clave para el proceso de detección y análisis de la cartera y ayuda a identificar y priorizar las aplicaciones que se van a migrar, modernizar y optimizar.

inteligencia artificial (IA)

El campo de la informática que se dedica al uso de tecnologías informáticas para realizar funciones cognitivas que suelen estar asociadas a los seres humanos, como el aprendizaje, la resolución de problemas y el reconocimiento de patrones. Para más información, consulte ¿Qué es la inteligencia artificial?

operaciones de inteligencia artificial (AIOps)

El proceso de utilizar técnicas de machine learning para resolver problemas operativos, reducir los incidentes operativos y la intervención humana, y mejorar la calidad del servicio. Para obtener más información sobre cómo AIOps se utiliza en la estrategia de AWS migración, consulte la guía de integración de operaciones.

cifrado asimétrico

Algoritmo de cifrado que utiliza un par de claves, una clave pública para el cifrado y una clave privada para el descifrado. Puede compartir la clave pública porque no se utiliza para el descifrado, pero el acceso a la clave privada debe estar sumamente restringido.

atomicidad, consistencia, aislamiento, durabilidad (ACID)

Conjunto de propiedades de software que garantizan la validez de los datos y la fiabilidad operativa de una base de datos, incluso en caso de errores, cortes de energía u otros problemas.

control de acceso basado en atributos (ABAC)

La práctica de crear permisos detallados basados en los atributos del usuario, como el departamento, el puesto de trabajo y el nombre del equipo. Para obtener más información, consulte ABAC AWS en la documentación AWS Identity and Access Management (IAM).

origen de datos fidedigno

Ubicación en la que se almacena la versión principal de los datos, que se considera la fuente de información más fiable. Puede copiar los datos del origen de datos autorizado a otras ubicaciones con el fin de procesarlos o modificarlos, por ejemplo, anonimizarlos, redactarlos o seudonimizarlos.

Zona de disponibilidad

Una ubicación distinta dentro de una Región de AWS que está aislada de los fallos en otras zonas de disponibilidad y que proporciona una conectividad de red económica y de baja latencia a otras zonas de disponibilidad de la misma región.

AWS Marco de adopción de la nube (AWS CAF)

Un marco de directrices y mejores prácticas AWS para ayudar a las organizaciones a desarrollar un plan eficiente y eficaz para migrar con éxito a la nube. AWS CAF organiza la orientación en seis áreas de enfoque denominadas perspectivas: negocios, personas, gobierno, plataforma, seguridad y operaciones. Las perspectivas empresariales, humanas y de gobernanza se centran en las habilidades y los procesos empresariales; las perspectivas de plataforma, seguridad y operaciones se centran en las habilidades y los procesos técnicos. Por ejemplo, la perspectiva humana se dirige a las partes interesadas que se ocupan de los Recursos Humanos (RR. HH.), las funciones del personal y la administración de las personas. Desde esta perspectiva, AWS CAF proporciona orientación para el desarrollo, la formación y la comunicación de las personas a fin de preparar a la organización para una adopción exitosa de la nube. Para obtener más información, consulte la Página web de AWS CAF y el Documento técnico de AWS CAF.

AWS Marco de calificación de la carga de trabajo (AWS WQF)

Herramienta que evalúa las cargas de trabajo de migración de bases de datos, recomienda estrategias de migración y proporciona estimaciones de trabajo. AWS WQF se incluye con AWS

Schema Conversion Tool ().AWS SCT Analiza los esquemas de bases de datos y los objetos de código, el código de las aplicaciones, las dependencias y las características de rendimiento y proporciona informes de evaluación.

B

bot malicioso

Bot destinado a causar interrupciones o daños a personas u organizaciones.

BCP

Consulte planificación de la continuidad del negocio.

gráfico de comportamiento

Una vista unificada e interactiva del comportamiento de los recursos y de las interacciones a lo largo del tiempo. Puede utilizar un gráfico de comportamiento con Amazon Detective para examinar los intentos de inicio de sesión fallidos, las llamadas sospechosas a la API y acciones similares. Para obtener más información, consulte Datos en un gráfico de comportamiento en la documentación de Detective.

sistema big-endian

Un sistema que almacena primero el byte más significativo. Consulte también endianidad.

clasificación binaria

Un proceso que predice un resultado binario (una de las dos clases posibles). Por ejemplo, es posible que su modelo de ML necesite predecir problemas como “¿Este correo electrónico es spam o no es spam?” o “¿Este producto es un libro o un automóvil?”.

filtro de floración

Estructura de datos probabilística y eficiente en términos de memoria que se utiliza para comprobar si un elemento es miembro de un conjunto.

implementación azul/verde

Estrategia de implementación en la que se crean dos entornos separados, pero idénticos. La versión actual de la aplicación se ejecuta en un entorno (azul) y la nueva versión de la aplicación se ejecuta en el otro entorno (verde). Esta estrategia lo ayuda a hacer reversiones rápidas con un impacto mínimo.

bot

Aplicación de software que ejecuta tareas automatizadas a través de Internet y simula la actividad o interacción humana. Algunos bots son útiles o beneficiosos, como los rastreadores web que indexan la información de Internet. Otros bots, conocidos como bots maliciosos, tienen como objetivo causar interrupciones o daños a personas u organizaciones.

botnet

Redes de bots infectadas por malware y que están bajo el control de una sola parte, conocida como pastor de bots u operador de bots. Las botnets son el mecanismo más conocido para escalar los bots y su impacto.

branch

Área contenida de un repositorio de código. La primera rama que se crea en un repositorio es la rama principal. Puede crear una rama nueva a partir de una rama existente y, a continuación, desarrollar características o corregir errores en la rama nueva. Una rama que se genera para crear una característica se denomina comúnmente rama de característica. Cuando la característica se encuentra lista para su lanzamiento, se vuelve a combinar la rama de característica con la rama principal. Para obtener más información, consulte Acerca de las sucursales (GitHub documentación).

acceso de emergencia

En circunstancias excepcionales y mediante un proceso aprobado, es una forma rápida de que un usuario pueda acceder a un Cuenta de AWS sitio al que normalmente no tiene permisos de acceso. Para más información, consulte el indicador Implement break-glass procedures en la guía de AWS Well-Architected.

estrategia de implementación sobre infraestructura existente

La infraestructura existente en su entorno. Al adoptar una estrategia de implementación sobre infraestructura existente para una arquitectura de sistemas, se diseña la arquitectura en función de las limitaciones de los sistemas y la infraestructura actuales. Si está ampliando la infraestructura existente, puede combinar las estrategias de implementación sobre infraestructuras existentes y de implementación desde cero.

caché de búfer

El área de memoria donde se almacenan los datos a los que se accede con más frecuencia.

capacidad empresarial

Lo que hace una empresa para generar valor (por ejemplo, ventas, servicio al cliente o marketing). Las arquitecturas de microservicios y las decisiones de desarrollo pueden estar impulsadas por las capacidades empresariales. Para obtener más información, consulte la sección Organizado en torno a las capacidades empresariales del documento técnico Ejecutar microservicios en contenedores en AWS.

planificación de la continuidad del negocio (BCP)

Plan que aborda el posible impacto de un evento disruptivo, como una migración a gran escala en las operaciones y permite a la empresa reanudar las operaciones rápidamente.

C

CAF

Consulte AWS Cloud Adoption Framework.

implementación canario

Lanzamiento lento e incremental de una versión para los usuarios finales. Cuando tenga mayor confianza en la nueva versión, la implementa y reemplaza la versión actual en su totalidad.

CCoE

Consulte Centro de excelencia en la nube.

CDC

Consulte captura de datos de cambios.

captura de datos de cambio (CDC)

Proceso de seguimiento de los cambios en un origen de datos, como una tabla de base de datos, y registro de los metadatos relacionados con el cambio. Puede utilizar los CDC para diversos fines, como auditar o replicar los cambios en un sistema de destino para mantener la sincronización.

ingeniería del caos

Introducción intencionada de fallos o eventos disruptivos para poner a prueba la resiliencia de un sistema. Puedes usar AWS Fault Injection Service (AWS FIS) para realizar experimentos que estresen tus AWS cargas de trabajo y evalúen su respuesta.

CI/CD

Consulte integración continua y entrega continua.

clasificación

Un proceso de categorización que permite generar predicciones. Los modelos de ML para problemas de clasificación predicen un valor discreto. Los valores discretos siempre son distintos entre sí. Por ejemplo, es posible que un modelo necesite evaluar si hay o no un automóvil en una imagen.

cifrado del cliente

Cifrado de datos localmente, antes de que el objetivo los Servicio de AWS reciba.

Centro de excelencia en la nube (CCoE)

Equipo multidisciplinario que impulsa los esfuerzos de adopción de la nube en toda la organización, incluido el desarrollo de las prácticas recomendadas en la nube, la movilización de recursos, el establecimiento de plazos de migración y la dirección de la organización durante las transformaciones a gran escala. Para obtener más información, consulte las publicaciones de CCo E en el blog de estrategia Nube de AWS empresarial.

computación en la nube

La tecnología en la nube que se utiliza normalmente para la administración de dispositivos de IoT y el almacenamiento de datos de forma remota. La computación en la nube suele estar relacionada con la tecnología de computación de periferia.

modelo operativo en la nube

En una organización de TI, el modelo operativo que se utiliza para crear, madurar y optimizar uno o más entornos de nube. Para obtener más información, consulte Creación de su modelo operativo de nube.

etapas de adopción de la nube

Las siguientes son las cuatro fases por las que suelen pasar las empresas cuando migran a la Nube de AWS:

- Proyecto: ejecución de algunos proyectos relacionados con la nube con fines de prueba de concepto y aprendizaje
- Fundamento: realizar inversiones fundamentales para escalar su adopción de la nube (p. ej., crear una landing zone, definir una CCo E, establecer un modelo de operaciones)

- Migración: migración de aplicaciones individuales
- Reinvención: optimización de productos y servicios e innovación en la nube

Stephen Orban definió estas etapas en la entrada del blog The Journey Toward Cloud-First & the Stages of Adoption en el blog Nube de AWS Enterprise Strategy. Para obtener información sobre su relación con la estrategia de AWS migración, consulte la guía de preparación para la migración.

CMDB

Consulte base de datos de administración de configuración.

repositorio de código

Una ubicación donde el código fuente y otros activos, como documentación, muestras y scripts, se almacenan y actualizan mediante procesos de control de versiones. Algunos repositorios en la nube comunes son GitHub o Bitbucket Cloud. Cada versión del código se denomina rama. En una estructura de microservicios, cada repositorio se encuentra dedicado a una única funcionalidad. Una sola canalización de CI/CD puede utilizar varios repositorios.

caché en frío

Una caché de búfer que está vacía no está bien poblada o contiene datos obsoletos o irrelevantes. Esto afecta al rendimiento, ya que la instancia de la base de datos debe leer desde la memoria principal o el disco, lo que es más lento que leer desde la memoria caché del búfer.

datos fríos

Datos a los que se accede con poca frecuencia y que suelen ser históricos. Al consultar este tipo de datos, normalmente se aceptan consultas lentas. Trasladar estos datos a niveles o clases de almacenamiento de menor rendimiento y menos costosos puede reducir los costos.

visión artificial (CV)

Campo de la IA que utiliza el machine learning para analizar y extraer información de formatos visuales, como imágenes y videos digitales. Por ejemplo, Amazon SageMaker AI proporciona algoritmos de procesamiento de imágenes para CV.

deriva de configuración

En el caso de una carga de trabajo, un cambio en la configuración con respecto al estado esperado. Podría provocar que la carga de trabajo deje de cumplir las normas y, por lo general, es gradual e involuntaria.

base de datos de administración de configuración (CMDB)

Repositorio que almacena y administra información sobre una base de datos y su entorno de TI, incluidos los componentes de hardware y software y sus configuraciones. Por lo general, los datos de una CMDB se utilizan en la etapa de detección y análisis de la cartera de productos durante la migración.

paquete de conformidad

Un conjunto de AWS Config reglas y medidas correctivas que puede reunir para personalizar sus controles de conformidad y seguridad. Puede implementar un paquete de conformidad como una entidad única en una región Cuenta de AWS y, o en una organización, mediante una plantilla YAML. Para obtener más información, consulta los paquetes de conformidad en la documentación. AWS Config

integración y entrega continuas (CI/CD)

El proceso de automatización de las etapas de origen, compilación, prueba, puesta en escena y producción del proceso de publicación del software. CI/CD se describe comúnmente como una canalización. CI/CD puede ayudarlo a automatizar los procesos, mejorar la productividad, mejorar la calidad del código y entregar más rápido. Para obtener más información, consulte Beneficios de la entrega continua. CD también puede significar implementación continua. Para obtener más información, consulte Entrega continua frente a implementación continua.

CV

Consulte visión artificial.

D

datos en reposo

Datos que están estacionarios en la red, como los datos que se encuentran almacenados.

clasificación de datos

Un proceso para identificar y clasificar los datos de su red en función de su importancia y sensibilidad. Es un componente fundamental de cualquier estrategia de administración de riesgos de ciberseguridad porque lo ayuda a determinar los controles de protección y retención adecuados para los datos. La clasificación de datos es un componente del pilar de seguridad

del AWS Well-Architected Framework. Para obtener más información, consulte Clasificación de datos.

deriva de datos

Una variación significativa entre los datos de producción y los datos que se utilizaron para entrenar un modelo de machine learning, o un cambio significativo en los datos de entrada a lo largo del tiempo. La deriva de datos puede reducir la calidad, la precisión y la imparcialidad generales de las predicciones de los modelos de machine learning.

datos en tránsito

Datos que se mueven de forma activa por la red, por ejemplo, entre los recursos de la red.

malla de datos

Marco de arquitectura que proporciona una propiedad de datos distribuida y descentralizada con una administración y una gobernanza centralizadas.

minimización de datos

El principio de recopilar y procesar solo los datos estrictamente necesarios. Practicar la minimización de los datos Nube de AWS puede reducir los riesgos de privacidad, los costos y la huella de carbono de la analítica.

perímetro de datos

Un conjunto de barreras preventivas en su AWS entorno que ayudan a garantizar que solo las identidades confiables accedan a los recursos confiables desde las redes esperadas. Para obtener más información, consulte Crear un perímetro de datos sobre. AWS

preprocesamiento de datos

Transformar los datos sin procesar en un formato que su modelo de ML pueda analizar fácilmente. El preprocesamiento de datos puede implicar eliminar determinadas columnas o filas y corregir los valores faltantes, incoherentes o duplicados.

procedencia de los datos

El proceso de rastrear el origen y el historial de los datos a lo largo de su ciclo de vida, por ejemplo, la forma en que se generaron, transmitieron y almacenaron los datos.

titular de los datos

Persona cuyos datos se recopilan y procesan.

almacenamiento de datos

Sistema de administración de datos que respalda la inteligencia empresarial, como los análisis. Los almacenes de datos suelen contener grandes cantidades de datos históricos y, por lo general, se utilizan para las consultas y los análisis.

lenguaje de definición de datos (DDL)

Instrucciones o comandos para crear o modificar la estructura de tablas y objetos de una base de datos.

lenguaje de manipulación de datos (DML)

Instrucciones o comandos para modificar (insertar, actualizar y eliminar) la información de una base de datos.

DDL

Consulte lenguaje de definición de bases de datos.

conjunto profundo

Combinar varios modelos de aprendizaje profundo para la predicción. Puede utilizar conjuntos profundos para obtener una predicción más precisa o para estimar la incertidumbre de las predicciones.

aprendizaje profundo

Un subcampo del ML que utiliza múltiples capas de redes neuronales artificiales para identificar el mapeo entre los datos de entrada y las variables objetivo de interés.

defense-in-depth

Un enfoque de seguridad de la información en el que se distribuyen cuidadosamente una serie de mecanismos y controles de seguridad en una red informática para proteger la confidencialidad, la integridad y la disponibilidad de la red y de los datos que contiene. Al adoptar esta estrategia AWS, se añaden varios controles en diferentes capas de la AWS Organizations estructura para ayudar a proteger los recursos. Por ejemplo, un defense-in-depth enfoque podría combinar la autenticación multifactorial, la segmentación de la red y el cifrado.

administrador delegado

En AWS Organizations, un servicio compatible puede registrar una cuenta de AWS miembro para administrar las cuentas de la organización y gestionar los permisos de ese servicio. Esta

cuenta se denomina administrador delegado para ese servicio. Para obtener más información y una lista de servicios compatibles, consulte Servicios que funcionan con AWS Organizations en la documentación de AWS Organizations .

Implementación

El proceso de hacer que una aplicación, características nuevas o correcciones de código se encuentren disponibles en el entorno de destino. La implementación abarca implementar cambios en una base de código y, a continuación, crear y ejecutar esa base en los entornos de la aplicación.

entorno de desarrollo

Consulte entorno.

control de detección

Un control de seguridad que se ha diseñado para detectar, registrar y alertar después de que se produzca un evento. Estos controles son una segunda línea de defensa, ya que lo advierten sobre los eventos de seguridad que han eludido los controles preventivos establecidos. Para obtener más información, consulte Controles de detección en Implementación de controles de seguridad en AWS.

asignación de flujos de valor para el desarrollo (DVSM)

Proceso que se utiliza para identificar y priorizar las restricciones que afectan negativamente a la velocidad y la calidad en el ciclo de vida del desarrollo de software. DVSM amplía el proceso de asignación del flujo de valor diseñado originalmente para las prácticas de fabricación ajustada. Se centra en los pasos y los equipos necesarios para crear y transferir valor a través del proceso de desarrollo de software.

gemelo digital

Representación virtual de un sistema del mundo real, como un edificio, una fábrica, un equipo industrial o una línea de producción. Los gemelos digitales son compatibles con el mantenimiento predictivo, la supervisión remota y la optimización de la producción.

tabla de dimensiones

En un esquema en estrella, tabla más pequeña que contiene los atributos de datos sobre los datos cuantitativos en una tabla de hechos. Los atributos de la tabla de dimensiones suelen ser campos de texto o números discretos que se comportan como texto. Estos atributos se suelen utilizar para restringir consultas, filtrarlas y etiquetar los conjuntos de resultados.

desastre

Un evento que impide que una carga de trabajo o un sistema cumplan sus objetivos empresariales en su ubicación principal de implementación. Estos eventos pueden ser desastres naturales, fallos técnicos o el resultado de acciones humanas, como una configuración incorrecta involuntaria o un ataque de malware.

recuperación de desastres (DR)

Estrategia y proceso que utiliza para minimizar el tiempo de inactividad y la pérdida de datos a causa de un desastre. Para obtener más información, consulte Recuperación ante desastres de cargas de trabajo en AWS: Recovery in the Cloud in the AWS Well-Architected Framework.

DML

Consulte lenguaje de manipulación de bases de datos.

diseño basado en el dominio

Un enfoque para desarrollar un sistema de software complejo mediante la conexión de sus componentes a dominios en evolución, o a los objetivos empresariales principales, a los que sirve cada componente. Este concepto lo introdujo Eric Evans en su libro, Diseño impulsado por el dominio: abordando la complejidad en el corazón del software (Boston: Addison-Wesley Professional, 2003). Para obtener información sobre cómo utilizar el diseño basado en dominios con el patrón de higos estranguladores, consulte Modernización gradual de los servicios web antiguos de Microsoft ASP.NET (ASMX) mediante contenedores y Amazon API Gateway.

DR

Consulte recuperación ante desastres.

Detección de desviaciones

Seguimiento de las desviaciones con respecto a una configuración con línea de base. Por ejemplo, puedes usarlo AWS CloudFormation para detectar desviaciones en los recursos del sistema o puedes usarlo AWS Control Tower para detectar cambios en tu landing zone que puedan afectar al cumplimiento de los requisitos de gobierno.

DVSM

Consulte asignación de flujos de valor para el desarrollo.

E

EDA

Consulte análisis de datos de tipo exploratorio.

EDI

Consulte intercambio electrónico de datos.

computación en la periferia

La tecnología que aumenta la potencia de cálculo de los dispositivos inteligentes en la periferia de una red de IoT. En comparación con la computación en la nube, la computación de periferia puede reducir la latencia de la comunicación y mejorar el tiempo de respuesta.

intercambio electrónico de datos (EDI)

Intercambio automatizado de documentos comerciales entre organizaciones. Para más información, consulte ¿Qué es el intercambio electrónico de datos?

cifrado

Proceso de computación que transforma datos de texto plano, que son legibles por humanos, en texto cifrado.

clave de cifrado

Cadena criptográfica de bits aleatorios que se genera mediante un algoritmo de cifrado. Las claves pueden variar en longitud y cada una se ha diseñado para ser impredecible y única.

endianidad

El orden en el que se almacenan los bytes en la memoria del ordenador. Los sistemas big-endianos almacenan primero el byte más significativo. Los sistemas Little-Endian almacenan primero el byte menos significativo.

punto de conexión

Consulte punto de conexión de servicio.

servicio de punto de conexión

Servicio que puede alojar en una nube privada virtual (VPC) para compartir con otros usuarios. Puede crear un servicio de punto final AWS PrivateLink y conceder permisos a otros Cuentas de AWS o a responsables AWS Identity and Access Management (de IAM). Estas cuentas o entidades principales pueden conectarse a su servicio de punto de conexión de forma privada

mediante la creación de puntos de conexión de VPC de interfaz. Para obtener más información, consulte Creación de un servicio de punto de conexión en la documentación de Amazon Virtual Private Cloud (Amazon VPC).

planificación de recursos empresariales (ERP)

Sistema que automatiza y administra los procesos empresariales clave (como la contabilidad, MES y la administración de proyectos) de una empresa.

cifrado de sobre

El proceso de cifrar una clave de cifrado con otra clave de cifrado. Para obtener más información, consulte el cifrado de sobres en la documentación de AWS Key Management Service (AWS KMS).

entorno

Una instancia de una aplicación en ejecución. Los siguientes son los tipos de entornos más comunes en la computación en la nube:

- entorno de desarrollo: instancia de una aplicación en ejecución que solo se encuentra disponible para el equipo principal responsable del mantenimiento de la aplicación. Los entornos de desarrollo se utilizan para probar los cambios antes de promocionarlos a los entornos superiores. Este tipo de entorno a veces se denomina entorno de prueba.
- entornos inferiores: todos los entornos de desarrollo de una aplicación, como los que se utilizan para las compilaciones y pruebas iniciales.
- entorno de producción: instancia de una aplicación en ejecución a la que pueden acceder los usuarios finales. En un CI/CD proceso, el entorno de producción es el último entorno de implementación.
- entornos superiores: todos los entornos a los que pueden acceder usuarios que no sean del equipo de desarrollo principal. Esto puede incluir un entorno de producción, entornos de preproducción y entornos para las pruebas de aceptación por parte de los usuarios.

epopeya

En las metodologías ágiles, son categorías funcionales que ayudan a organizar y priorizar el trabajo. Las epopeyas brindan una descripción detallada de los requisitos y las tareas de implementación. Por ejemplo, las epopeyas AWS de seguridad de CAF incluyen la gestión de identidades y accesos, los controles de detección, la seguridad de la infraestructura, la protección de datos y la respuesta a incidentes. Para obtener más información sobre las epopeyas en la estrategia de migración de AWS , consulte la Guía de implementación del programa.

ERP

Consulte planificación de recursos empresariales.

análisis de datos de tipo exploratorio (EDA)

El proceso de analizar un conjunto de datos para comprender sus características principales. Se recopilan o agregan datos y, a continuación, se realizan las investigaciones iniciales para encontrar patrones, detectar anomalías y comprobar las suposiciones. El EDA se realiza mediante el cálculo de estadísticas resumidas y la creación de visualizaciones de datos.

F

tabla de hechos

Tabla central de un esquema en estrella. Almacena datos cuantitativos sobre operaciones empresariales. Por lo general, una tabla de hechos contiene dos tipos de columnas: las que contienen medidas y las que contienen una clave externa para una tabla de dimensiones.

Fail Fast

Filosofía que utiliza pruebas frecuentes e incrementales para reducir el ciclo de vida del desarrollo. Es una parte fundamental de los enfoques ágiles.

límite de aislamiento de errores

En el Nube de AWS, un límite, como una zona de disponibilidad Región de AWS, un plano de control o un plano de datos, que limita el efecto de una falla y ayuda a mejorar la resiliencia de las cargas de trabajo. Para más información, consulte AWS Fault Isolation Boundaries.

rama de característica

Consulte rama.

características

Los datos de entrada que se utilizan para hacer una predicción. Por ejemplo, en un contexto de fabricación, las características pueden ser imágenes que se capturan periódicamente desde la línea de fabricación.

importancia de las características

La importancia que tiene una característica para las predicciones de un modelo. Por lo general, esto se expresa como una puntuación numérica que se puede calcular mediante diversas

técnicas, como las explicaciones aditivas de Shapley (SHAP) y los gradientes integrados. Para obtener más información, consulte Interpretabilidad del modelo de aprendizaje automático con. AWS

transformación de funciones

Optimizar los datos para el proceso de ML, lo que incluye enriquecer los datos con fuentes adicionales, escalar los valores o extraer varios conjuntos de información de un solo campo de datos. Esto permite que el modelo de ML se beneficie de los datos. Por ejemplo, si divide la fecha del “27 de mayo de 2021 00:15:37” en “jueves”, “mayo”, “2021” y “15”, puede ayudar al algoritmo de aprendizaje a aprender patrones matizados asociados a los diferentes componentes de los datos.

peticiones con pocos pasos

Proporcionar a un LLM una pequeña cantidad de ejemplos que demuestren la tarea y el resultado deseado antes de pedirle que lleve a cabo una tarea similar. Esta técnica es una aplicación del aprendizaje contextual, mediante el que los modelos aprenden a partir de ejemplos (pasos) incrustados en las peticiones. La técnica de peticiones con pocos pasos puede ser eficaz para las tareas que requieren un formato, un razonamiento o un conocimiento del dominio específicos. Consulte también peticiones desde cero.

FGAC

Consulte control de acceso detallado.

control de acceso preciso (FGAC)

El uso de varias condiciones que tienen por objetivo permitir o denegar una solicitud de acceso.

migración relámpago

Método de migración de bases de datos que utiliza la replicación continua de datos mediante la captura de datos de cambio para migrar los datos en el menor tiempo posible, en lugar de utilizar un enfoque gradual. El objetivo es reducir al mínimo el tiempo de inactividad.

FM

Consulte modelo fundacional.

Modelo fundacional (FM)

Una gran red neuronal de aprendizaje profundo que se ha estado entrenando con conjuntos de datos masivos de datos generalizados y sin etiquetar. FMs son capaces de realizar una

amplia variedad de tareas generales, como comprender el lenguaje, generar texto e imágenes y conversar en lenguaje natural. Para más información, consulte ¿Qué son los modelos fundacionales?

G

IA generativa

Subconjunto de modelos de IA que se entrenaron con grandes cantidades de datos y que pueden utilizar una simple petición de texto para crear contenido y artefactos nuevos, como imágenes, videos, texto y audio. Para más información, consulte ¿Qué es la IA generativa?

bloqueo geográfico

Consulte restricciones geográficas.

restricciones geográficas (bloqueo geográfico)

En Amazon CloudFront, una opción para impedir que los usuarios de países específicos accedan a las distribuciones de contenido. Puede utilizar una lista de permitidos o bloqueados para especificar los países aprobados y prohibidos. Para obtener más información, consulta la sección Restringir la distribución geográfica del contenido en la CloudFront documentación.

Flujo de trabajo de Gitflow

Un enfoque en el que los entornos inferiores y superiores utilizan diferentes ramas en un repositorio de código fuente. El flujo de trabajo de Gitflow se considera heredado, mientras que el flujo de trabajo basado en enlaces troncales es el enfoque moderno preferido.

imagen dorada

Instantánea de un sistema o software que se usa como plantilla para implementar nuevas instancias de ese sistema o software. Por ejemplo, en la fabricación, una imagen dorada se puede utilizar para aprovisionar software en varios dispositivos y ayuda a mejorar la velocidad, la escalabilidad y la productividad de las operaciones de fabricación de dispositivos.

estrategia de implementación desde cero

La ausencia de infraestructura existente en un entorno nuevo. Al adoptar una estrategia de implementación desde cero para una arquitectura de sistemas, puede seleccionar todas las tecnologías nuevas sin que estas deban ser compatibles con una infraestructura existente, lo que también se conoce como implementación sobre infraestructura existente. Si está

ampliando la infraestructura existente, puede combinar las estrategias de implementación sobre infraestructuras existentes y de implementación desde cero.

barrera de protección

Una regla de alto nivel que ayuda a regular los recursos, las políticas y el cumplimiento en todas las unidades organizativas (OUs). Las barreras de protección preventivas aplican políticas para garantizar la alineación con los estándares de conformidad. Se implementan mediante políticas de control de servicios y límites de permisos de IAM. Las barreras de protección de detección detectan las vulneraciones de las políticas y los problemas de conformidad, y generan alertas para su corrección. Se implementan mediante Amazon AWS Config AWS Security Hub CSPM GuardDuty AWS Trusted Advisor, Amazon Inspector y AWS Lambda cheques personalizados.

H

HA

Consulte alta disponibilidad.

migración heterogénea de bases de datos

Migración de la base de datos de origen a una base de datos de destino que utilice un motor de base de datos diferente (por ejemplo, de Oracle a Amazon Aurora). La migración heterogénea suele ser parte de un esfuerzo de rediseño de la arquitectura y convertir el esquema puede ser una tarea compleja. AWS ofrece AWS SCT, lo cual ayuda con las conversiones de esquemas.

alta disponibilidad (HA)

La capacidad de una carga de trabajo para funcionar de forma continua, sin intervención, en caso de desafíos o desastres. Los sistemas de alta disponibilidad están diseñados para realizar una conmutación por error automática, ofrecer un rendimiento de alta calidad de forma constante y gestionar diferentes cargas y fallos con un impacto mínimo en el rendimiento.

modernización histórica

Un enfoque utilizado para modernizar y actualizar los sistemas de tecnología operativa (TO) a fin de satisfacer mejor las necesidades de la industria manufacturera. Un histórico es un tipo de base de datos que se utiliza para recopilar y almacenar datos de diversas fuentes en una fábrica.

datos de reserva

Parte de los datos históricos etiquetados que se ocultan de un conjunto de datos que se utiliza para entrenar un modelo de machine learning. Puede utilizar los datos de reserva para evaluar el rendimiento del modelo mediante la comparación de las predicciones del modelo con los datos de reserva.

migración homogénea de bases de datos

Migración de la base de datos de origen a una base de datos de destino que comparte el mismo motor de base de datos (por ejemplo, Microsoft SQL Server a Amazon RDS para SQL Server). La migración homogénea suele formar parte de un esfuerzo para volver a alojar o redefinir la plataforma. Puede utilizar las utilidades de bases de datos nativas para migrar el esquema.

datos recientes

Datos a los que se accede con frecuencia, como datos en tiempo real o datos traslacionales recientes. Por lo general, estos datos requieren un nivel o una clase de almacenamiento de alto rendimiento para proporcionar respuestas rápidas a las consultas.

hotfix

Una solución urgente para un problema crítico en un entorno de producción. Debido a su urgencia, una revisión suele realizarse fuera del flujo de trabajo de DevOps publicación típico.

periodo de hiperatención

Periodo, inmediatamente después de la transición, durante el cual un equipo de migración administra y monitorea las aplicaciones migradas en la nube para solucionar cualquier problema. Por lo general, este periodo dura de 1 a 4 días. Al final del periodo de hiperatención, el equipo de migración suele transferir la responsabilidad de las aplicaciones al equipo de operaciones en la nube.

I

IaC

Consulte infraestructura como código.

políticas basadas en identidades

Política asociada a uno o más directores de IAM que define sus permisos en el entorno. Nube de AWS

aplicación inactiva

Aplicación que utiliza un promedio de CPU y memoria de entre 5 y 20 por ciento durante un periodo de 90 días. En un proyecto de migración, es habitual retirar estas aplicaciones o mantenerlas en las instalaciones.

IIoT

Consulte Internet de las cosas industrial.

infraestructura inmutable

Modelo que implementa una nueva infraestructura para las cargas de trabajo de producción en lugar de actualizar o modificar la infraestructura existente o aplicarle revisiones. Las infraestructuras inmutables son de manera intrínseca más coherentes, fiables y predecibles que las infraestructuras mutables. Para más información, consulte la práctica recomendada Implementación mediante una infraestructura inmutable en el Marco de AWS Well-Architected.

VPC entrante (de entrada)

En una arquitectura de AWS cuentas múltiples, una VPC que acepta, inspecciona y enruta las conexiones de red desde fuera de una aplicación. La arquitectura AWS de referencia de seguridad recomienda configurar la cuenta de red con entradas, salidas e inspección VPCs para proteger la interfaz bidireccional entre la aplicación y el resto de Internet.

migración gradual

Estrategia de transición en la que se migra la aplicación en partes pequeñas en lugar de realizar una transición única y completa. Por ejemplo, puede trasladar inicialmente solo unos pocos microservicios o usuarios al nuevo sistema. Tras comprobar que todo funciona correctamente, puede trasladar microservicios o usuarios adicionales de forma gradual hasta que pueda retirar su sistema heredado. Esta estrategia reduce los riesgos asociados a las grandes migraciones.

Industria 4.0

Término que introdujo Klaus Schwab en 2016 para referirse a la modernización de los procesos de fabricación mediante los avances en la conectividad, los datos en tiempo real, la automatización, el análisis, la IA y el ML.

infraestructura

Todos los recursos y activos que se encuentran en el entorno de una aplicación.

infraestructura como código (IaC)

Proceso de aprovisionamiento y administración de la infraestructura de una aplicación mediante un conjunto de archivos de configuración. La IaC se ha diseñado para ayudarlo a centralizar la administración de la infraestructura, estandarizar los recursos y escalar con rapidez a fin de que los entornos nuevos sean repetibles, fiables y consistentes.

Internet de las cosas industrial (T) IIo

El uso de sensores y dispositivos conectados a Internet en los sectores industriales, como el productivo, el eléctrico, el automotriz, el sanitario, el de las ciencias de la vida y el de la agricultura. Para obtener más información, consulte Creación de una estrategia de transformación digital de la Internet de las cosas (IIoT) industrial.

VPC de inspección

En una arquitectura de AWS cuentas múltiples, una VPC centralizada que gestiona las inspecciones del tráfico de red VPCs entre Internet y las redes locales (en una misma o Regiones de AWS diferente). La arquitectura AWS de referencia de seguridad recomienda configurar su cuenta de red con entrada, salida e inspección VPCs para proteger la interfaz bidireccional entre la aplicación e Internet en general.

Internet de las cosas (IoT)

Red de objetos físicos conectados con sensores o procesadores integrados que se comunican con otros dispositivos y sistemas a través de Internet o de una red de comunicación local. Para obtener más información, consulte ¿Qué es IoT?.

interpretabilidad

Característica de un modelo de machine learning que describe el grado en que un ser humano puede entender cómo las predicciones del modelo dependen de sus entradas. Para obtener más información, consulte Interpretabilidad del modelo de aprendizaje automático con. AWS

IoT

Consulte Internet de las cosas.

biblioteca de información de TI (ITIL)

Conjunto de prácticas recomendadas para ofrecer servicios de TI y alinearlos con los requisitos empresariales. La ITIL proporciona la base para la ITSM.

administración de servicios de TI (ITSM)

Actividades asociadas con el diseño, la implementación, la administración y el soporte de los servicios de TI para una organización. Para obtener información sobre la integración de las operaciones en la nube con las herramientas de ITSM, consulte la Guía de integración de operaciones.

ITIL

Consulte biblioteca de información de TI.

ITSM

Consulte administración de servicios de TI.

L

control de acceso basado en etiquetas (LBAC)

Una implementación del control de acceso obligatorio (MAC) en la que a los usuarios y a los propios datos se les asigna explícitamente un valor de etiqueta de seguridad. La intersección entre la etiqueta de seguridad del usuario y la etiqueta de seguridad de los datos determina qué filas y columnas puede ver el usuario.

zona de aterrizaje

Una landing zone es un AWS entorno multicuenta bien diseñado, escalable y seguro. Este es un punto de partida desde el cual las empresas pueden lanzar e implementar rápidamente cargas de trabajo y aplicaciones con confianza en su entorno de seguridad e infraestructura. Para obtener más información sobre las zonas de aterrizaje, consulte Configuración de un entorno de AWS seguro y escalable con varias cuentas.

modelo de lenguaje de gran tamaño (LLM)

Modelo de IA de aprendizaje profundo que se entrenó previamente con una gran cantidad de datos. Un LLM puede llevar a cabo varias tareas, como responder preguntas, resumir documentos, traducir textos a otros idiomas y completar oraciones. Para obtener más información, consulte Qué son. LLMs

migración grande

Migración de 300 servidores o más.

LBAC

Consulte control de acceso basado en etiquetas.

privilegio mínimo

La práctica recomendada de seguridad que consiste en conceder los permisos mínimos necesarios para realizar una tarea. Para obtener más información, consulte Aplicar permisos de privilegio mínimo en la documentación de IAM.

migrar mediante lift-and-shift

Consulte Las 7 R.

sistema little-endian

Un sistema que almacena primero el byte menos significativo. Consulte también endianidad.

LLM

Consulte modelo de lenguaje de gran tamaño.

entornos inferiores

Consulte entorno.

M

machine learning (ML)

Un tipo de inteligencia artificial que utiliza algoritmos y técnicas para el reconocimiento y el aprendizaje de patrones. El ML analiza y aprende de los datos registrados, como los datos del Internet de las cosas (IoT), para generar un modelo estadístico basado en patrones. Para más información, consulte Machine learning.

rama principal

Consulte rama.

malware

Software diseñado para comprometer la seguridad o la privacidad de la computadora. El malware podría interrumpir los sistemas informáticos, filtrar información confidencial u obtener acceso

no autorizado. Algunos ejemplos de malware son los virus, los gusanos, el ransomware, los troyanos, el spyware y los registradores de pulsaciones de teclas.

Servicios administrados

Servicios de AWS para lo cual AWS opera la capa de infraestructura, el sistema operativo y las plataformas, y se accede a los puntos finales para almacenar y recuperar datos. Amazon Simple Storage Service (Amazon S3) y Amazon DynamoDB son ejemplos de servicios administrados. También se conocen como servicios abstractos.

sistema de ejecución de fabricación (MES)

Sistema de software para seguir, supervisar, documentar y controlar los procesos de producción que convierten las materias primas en productos acabados en la zona de producción.

MAP

Consulte Programa de aceleración de la migración.

mecanismo

Proceso completo mediante el que se crea una herramienta, se impulsa su adopción y, a continuación, se inspeccionan los resultados para hacer ajustes. Un mecanismo es un ciclo que se refuerza y mejora por sí mismo a medida que funciona. Para obtener más información, consulte Creación de mecanismos en el AWS Well-Architected Framework.

cuenta de miembro

Todas las Cuentas de AWS demás cuentas, excepto la de administración, que forman parte de una organización. AWS Organizations Una cuenta no puede pertenecer a más de una organización a la vez.

MES

Consulte sistema de ejecución de fabricación.

Message Queuing Telemetry Transport (MQTT)

Un protocolo de comunicación ligero machine-to-machine (M2M), basado en el patrón de publicación/suscripción, para dispositivos de IoT con recursos limitados.

microservicio

Un servicio pequeño e independiente que se comunica a través de una red bien definida APIs y que, por lo general, es propiedad de equipos pequeños e independientes. Por ejemplo,

un sistema de seguros puede incluir microservicios que se adapten a las capacidades empresariales, como las de ventas o marketing, o a subdominios, como las de compras, reclamaciones o análisis. Los beneficios de los microservicios incluyen la agilidad, la escalabilidad flexible, la facilidad de implementación, el código reutilizable y la resiliencia. Para obtener más información, consulte Integrar microservicios mediante AWS servicios sin servidor.

arquitectura de microservicios

Un enfoque para crear una aplicación con componentes independientes que ejecutan cada proceso de la aplicación como un microservicio. Estos microservicios se comunican a través de una interfaz bien definida mediante un uso ligero. APIs Cada microservicio de esta arquitectura se puede actualizar, implementar y escalar para satisfacer la demanda de funciones específicas de una aplicación. Para obtener más información, consulte Implementación de microservicios en. AWS

Programa de aceleración de la migración (MAP)

Un AWS programa que proporciona soporte de consultoría, formación y servicios para ayudar a las organizaciones a crear una base operativa sólida para migrar a la nube y para ayudar a compensar el costo inicial de las migraciones. El MAP incluye una metodología de migración para ejecutar las migraciones antiguas de forma metódica y un conjunto de herramientas para automatizar y acelerar los escenarios de migración más comunes.

migración a escala

Proceso de transferencia de la mayoría de la cartera de aplicaciones a la nube en oleadas, con más aplicaciones desplazadas a un ritmo más rápido en cada oleada. En esta fase, se utilizan las prácticas recomendadas y las lecciones aprendidas en las fases anteriores para implementar una fábrica de migración de equipos, herramientas y procesos con el fin de agilizar la migración de las cargas de trabajo mediante la automatización y la entrega ágil. Esta es la tercera fase de la estrategia de migración de AWS.

fábrica de migración

Equipos multifuncionales que agilizan la migración de las cargas de trabajo mediante enfoques automatizados y ágiles. Los equipos de las fábricas de migración suelen incluir a analistas y propietarios de operaciones, empresas, ingenieros de migración, desarrolladores y DevOps profesionales que trabajan a pasos agigantados. Entre el 20 y el 50 por ciento de la cartera de aplicaciones empresariales se compone de patrones repetidos que pueden optimizarse mediante un enfoque de fábrica. Para obtener más información, consulte la discusión sobre las fábricas de migración y la Guía de fábricas de migración a la nube en este contenido.

metadatos de migración

Información sobre la aplicación y el servidor que se necesita para completar la migración. Cada patrón de migración requiere un conjunto diferente de metadatos de migración. Algunos ejemplos de metadatos de migración son la subred de destino, el grupo de seguridad y AWS la cuenta.

patrón de migración

Tarea de migración repetible que detalla la estrategia de migración, el destino de la migración y la aplicación o el servicio de migración utilizados. Ejemplo: rehospede la migración a Amazon EC2 AWS con Application Migration Service.

Migration Portfolio Assessment (MPA)

Herramienta en línea que proporciona información a fin de validar los argumentos comerciales necesarios para migrar a la Nube de AWS. La MPA ofrece una evaluación detallada de la cartera (adecuación del tamaño de los servidores, precios, comparaciones del costo total de propiedad, análisis de los costos de migración), así como una planificación de la migración (análisis y recopilación de datos de aplicaciones, agrupación de aplicaciones, priorización de la migración y planificación de oleadas). La herramienta MPA (requiere iniciar sesión) está disponible de forma gratuita para todos los AWS consultores y consultores de los socios de APN.

Evaluación de la preparación para la migración (MRA)

Proceso que consiste en obtener información sobre el estado de preparación de una organización para la nube, identificar sus puntos fuertes y débiles y elaborar un plan de acción para cerrar las brechas identificadas mediante el AWS CAF. Para obtener más información, consulte la Guía de preparación para la migración. La MRA es la primera fase de la estrategia de migración de AWS.

estrategia de migración

Enfoque utilizado para migrar una carga de trabajo a la Nube de AWS. Para más información, consulte la entrada Las 7 R de este glosario y también Mobilize your organization to accelerate large-scale migrations.

ML

Consulte machine learning.

modernización

Transformar una aplicación obsoleta (antigua o monolítica) y su infraestructura en un sistema ágil, elástico y de alta disponibilidad en la nube para reducir los gastos, aumentar la eficiencia

y aprovechar las innovaciones. Para más información, consulte Strategy for modernizing applications in the Nube de AWS.

evaluación de la preparación para la modernización

Evaluación que ayuda a determinar la preparación para la modernización de las aplicaciones de una organización; identifica los beneficios, los riesgos y las dependencias; y determina qué tan bien la organización puede soportar el estado futuro de esas aplicaciones. El resultado de la evaluación es un esquema de la arquitectura objetivo, una hoja de ruta que detalla las fases de desarrollo y los hitos del proceso de modernización y un plan de acción para abordar las brechas identificadas. Para más información, consulte Evaluating modernization readiness for applications in the Nube de AWS.

aplicaciones monolíticas (monolitos)

Aplicaciones que se ejecutan como un único servicio con procesos estrechamente acoplados. Las aplicaciones monolíticas presentan varios inconvenientes. Si una característica de la aplicación experimenta un aumento en la demanda, se debe escalar toda la arquitectura. Agregar o mejorar las características de una aplicación monolítica también se vuelve más complejo a medida que crece la base de código. Para solucionar problemas con la aplicación, puede utilizar una arquitectura de microservicios. Para obtener más información, consulte Descomposición de monolitos en microservicios.

MPA

Consulte Migration Portfolio Assessment.

MQTT

Consulte Message Queuing Telemetry Transport.

clasificación multiclase

Un proceso que ayuda a generar predicciones para varias clases (predice uno de más de dos resultados). Por ejemplo, un modelo de ML podría preguntar “¿Este producto es un libro, un automóvil o un teléfono?” o “¿Qué categoría de productos es más interesante para este cliente?”.

infraestructura mutable

Modelo que actualiza y modifica la infraestructura actual para las cargas de trabajo de producción. Para mejorar la coherencia, la fiabilidad y la previsibilidad, el AWS Well-Architected Framework recomienda el uso de una infraestructura inmutable como práctica recomendada.

O

OAC

Consulte control de acceso de origen.

OAI

Consulte identidad de acceso de origen.

OCM

Consulte administración del cambio organizacional.

migración fuera de línea

Método de migración en el que la carga de trabajo de origen se elimina durante el proceso de migración. Este método implica un tiempo de inactividad prolongado y, por lo general, se utiliza para cargas de trabajo pequeñas y no críticas.

OI

Consulte integración de operaciones.

OLA

Consulte acuerdo de nivel operativo.

migración en línea

Método de migración en el que la carga de trabajo de origen se copia al sistema de destino sin que se desconecte. Las aplicaciones que están conectadas a la carga de trabajo pueden seguir funcionando durante la migración. Este método implica un tiempo de inactividad nulo o mínimo y, por lo general, se utiliza para cargas de trabajo de producción críticas.

OPC-UA

Consulte Open Process Communications: arquitectura unificada.

Open Process Communications: arquitectura unificada (OPC-UA)

Un protocolo de machine-to-machine comunicación (M2M) para la automatización industrial. OPC-UA establece un estándar de interoperabilidad con esquemas de autenticación, autorización y cifrado de datos.

acuerdo de nivel operativo (OLA)

Acuerdo que aclara lo que los grupos de TI operativos se comprometen a ofrecerse entre sí, para respaldar un acuerdo de nivel de servicio (SLA).

revisión de la preparación operativa (ORR)

Lista de comprobación de preguntas y prácticas recomendadas asociadas que son útiles para comprender, evaluar, prevenir o reducir el alcance de los incidentes y posibles errores. Para más información, consulte Operational Readiness Reviews (ORR) en el Marco de AWS Well-Architected.

tecnología operativa (TO)

Sistemas de hardware y software que funcionan con el entorno físico para controlar las operaciones, los equipos y la infraestructura industriales. En el sector de la fabricación, la integración de los sistemas de TO y tecnología de la información (TI) es un enfoque clave para las transformaciones de la industria 4.0.

integración de operaciones (OI)

Proceso de modernización de las operaciones en la nube, que implica la planificación de la preparación, la automatización y la integración. Para obtener más información, consulte la Guía de integración de las operaciones.

registro de seguimiento organizativo

Un registro creado por y AWS CloudTrail que registra todos los eventos para todos los miembros Cuentas de AWS de una organización. AWS Organizations Este registro de seguimiento se crea en cada Cuenta de AWS que forma parte de la organización y realiza un seguimiento de la actividad en cada cuenta. Para obtener más información, consulte Crear un registro para una organización en la CloudTrail documentación.

administración del cambio organizacional (OCM)

Marco para administrar las transformaciones empresariales importantes y disruptivas desde la perspectiva de las personas, la cultura y el liderazgo. La OCM ayuda a las empresas a prepararse para nuevos sistemas y estrategias y a realizar la transición a ellos, al acelerar la adopción de cambios, abordar los problemas de transición e impulsar cambios culturales y organizacionales. En la estrategia de AWS migración, este marco se denomina aceleración de personal, debido a la velocidad de cambio que requieren los proyectos de adopción de la nube. Para obtener más información, consulte la Guía de OCM.

control de acceso de origen (OAC)

En CloudFront, una opción mejorada para restringir el acceso y proteger el contenido del Amazon Simple Storage Service (Amazon S3). El OAC admite todos los buckets de S3 Regiones de AWS, el cifrado del lado del servidor AWS KMS (SSE-KMS) y las solicitudes dinámicas `PUT` y `DELETE` dirigidas al bucket de S3.

identidad de acceso de origen (OAI)

En CloudFront, una opción para restringir el acceso y proteger el contenido de Amazon S3. Cuando utiliza OAI, CloudFront crea un principal con el que Amazon S3 puede autenticarse. Los directores autenticados solo pueden acceder al contenido de un bucket de S3 a través de una distribución específica. CloudFront Consulte también el OAC, que proporciona un control de acceso más detallado y mejorado.

ORR

Consulte revisión de la preparación operativa.

OT

Consulte tecnología operativa.

VPC saliente (de salida)

En una arquitectura de AWS cuentas múltiples, una VPC que gestiona las conexiones de red que se inician desde una aplicación. La arquitectura AWS de referencia de seguridad recomienda configurar la cuenta de red con entradas, salidas e inspección VPCs para proteger la interfaz bidireccional entre la aplicación e Internet en general.

P

límite de permisos

Una política de administración de IAM que se adjunta a las entidades principales de IAM para establecer los permisos máximos que puede tener el usuario o el rol. Para obtener más información, consulte Límites de permisos en la documentación de IAM.

información de identificación personal (PII)

Información que, vista directamente o combinada con otros datos relacionados, puede utilizarse para deducir de manera razonable la identidad de una persona. Algunos ejemplos de información de identificación personal son los nombres, las direcciones y la información de contacto.

PII

Consulte información de identificación personal.

manual de estrategias

Conjunto de pasos predefinidos que capturan el trabajo asociado a las migraciones, como la entrega de las funciones de operaciones principales en la nube. Un manual puede adoptar la forma de scripts, manuales de procedimientos automatizados o resúmenes de los procesos o pasos necesarios para operar un entorno modernizado.

PLC

Consulte controlador lógico programable.

PLM

Consulte administración del ciclo de vida del producto.

policy

Objeto que puede definir permisos (consulte política basada en identidad), especificar las condiciones de acceso (consulte política basada en recursos) o definir los permisos máximos para todas las cuentas de una organización de AWS Organizations (consulte política de control de servicio).

persistencia políglota

Elegir de forma independiente la tecnología de almacenamiento de datos de un microservicio en función de los patrones de acceso a los datos y otros requisitos. Si sus microservicios tienen la misma tecnología de almacenamiento de datos, pueden enfrentarse a desafíos de implementación o experimentar un rendimiento deficiente. Los microservicios se implementan más fácilmente y logran un mejor rendimiento y escalabilidad si utilizan el almacén de datos que mejor se adapte a sus necesidades.

evaluación de cartera

Proceso de detección, análisis y priorización de la cartera de aplicaciones para planificar la migración. Para obtener más información, consulte la Evaluación de la preparación para la migración.

predicate

Condición de consulta que devuelve `true` o `false`. En general, se encuentra en una cláusula `WHERE`.

inserción de predicados

Técnica de optimización de consultas en bases de datos que filtra los datos de la consulta antes de transferirlos. Esta técnica reduce la cantidad de datos de la base de datos relacional que se tienen que recuperar y procesar. Además, mejora el rendimiento de las consultas.

control preventivo

Un control de seguridad diseñado para evitar que ocurra un evento. Estos controles son la primera línea de defensa para evitar el acceso no autorizado o los cambios no deseados en la red. Para obtener más información, consulte Controles preventivos en Implementación de controles de seguridad en AWS.

entidad principal

Una entidad AWS que puede realizar acciones y acceder a los recursos. Esta entidad suele ser un usuario raíz para un Cuenta de AWS rol de IAM o un usuario. Para obtener más información, consulte Entidad principal en Términos y conceptos de roles en la documentación de IAM.

Privacidad desde el diseño

Enfoque de ingeniería de sistemas que tiene en cuenta la privacidad durante todo el proceso de desarrollo.

zonas alojadas privadas

Un contenedor que contiene información sobre cómo desea que Amazon Route 53 responda a las consultas de DNS de un dominio y sus subdominios dentro de uno o más VPCs. Para obtener más información, consulte Uso de zonas alojadas privadas en la documentación de Route 53.

control proactivo

Control de seguridad que se diseñó para evitar la implementación de recursos que no cumplan con la normativa. Estos controles analizan los recursos antes de aprovisionarlos. Si el recurso no cumple con los requisitos del control, no se aprovisiona. Para obtener más información, consulte la guía de referencia de controles en la AWS Control Tower documentación y consulte Controles proactivos en la sección Implementación de controles de seguridad en AWS.

administración del ciclo de vida del producto (PLM)

Administración de los datos y los procesos de un producto a lo largo de todo su ciclo de vida, desde el diseño, el desarrollo y el lanzamiento, pasando por el crecimiento y la madurez, hasta la reducción de su uso y su retirada.

entorno de producción

Consulte entorno.

controlador lógico programable (PLC)

En el sector de la fabricación, computadora adaptable y altamente fiable que supervisa las máquinas y automatiza los procesos de fabricación.

encadenamiento de peticiones

Uso de la salida de una petición de LLM como entrada para la siguiente petición a fin de generar mejores respuestas. Esta técnica se utiliza para dividir una tarea compleja en tareas secundarias o para refinar o ampliar de forma iterativa una respuesta preliminar. Ayuda a mejorar la precisión y la relevancia de las respuestas de un modelo y permite obtener resultados más detallados y personalizados.

seudonimización

El proceso de reemplazar los identificadores personales de un conjunto de datos por valores de marcadores de posición. La seudonimización puede ayudar a proteger la privacidad personal. Los datos seudonimizados siguen considerándose datos personales.

publish/subscribe (pub/sub)

Patrón que permite establecer comunicaciones asíncronas entre microservicios para mejorar la escalabilidad y la capacidad de respuesta. Por ejemplo, en un MES basado en microservicios, un microservicio puede publicar mensajes de eventos en un canal al que se pueden suscribir otros microservicios. El sistema puede agregar nuevos microservicios sin cambiar el servicio de publicación.

Q

plan de consulta

Serie de pasos, como instrucciones, que se utilizan para acceder a los datos de un sistema de base de datos relacional SQL.

regresión del plan de consulta

El optimizador de servicios de la base de datos elige un plan menos óptimo que antes de un cambio determinado en el entorno de la base de datos. Los cambios en estadísticas,

restricciones, configuración del entorno, enlaces de parámetros de consultas y actualizaciones del motor de base de datos PostgreSQL pueden provocar una regresión del plan.

R

Matriz RACI

Consulte responsable, fiable, consultada e informada (RACI).

RAG

Consulte generación aumentada por recuperación.

ransomware

Software malicioso que se ha diseñado para bloquear el acceso a un sistema informático o a los datos hasta que se efectúe un pago.

Matriz RASCI

Consulte responsable, fiable, consultada e informada (RACI).

RCAC

Consulte control de acceso por filas y columnas.

réplica de lectura

Una copia de una base de datos que se utiliza con fines de solo lectura. Puede enrutar las consultas a la réplica de lectura para reducir la carga en la base de datos principal.

rediseñar

Consulte Las 7 R.

objetivo de punto de recuperación (RPO)

La cantidad de tiempo máximo aceptable desde el último punto de recuperación de datos. Esto determina qué se considera una pérdida de datos aceptable entre el último punto de recuperación y la interrupción del servicio.

objetivo de tiempo de recuperación (RTO)

La demora máxima aceptable entre la interrupción del servicio y el restablecimiento del servicio.

refactorizar

Consulte Las 7 R.

Region

Conjunto de AWS recursos en un área geográfica. Cada uno Región de AWS está aislado e independiente de los demás para proporcionar tolerancia a las fallas, estabilidad y resiliencia. Para más información, consulte Specify which Regiones de AWS your account can use.

regresión

Una técnica de ML que predice un valor numérico. Por ejemplo, para resolver el problema de "¿A qué precio se venderá esta casa?", un modelo de ML podría utilizar un modelo de regresión lineal para predecir el precio de venta de una vivienda en función de datos conocidos sobre ella (por ejemplo, los metros cuadrados).

volver a alojar

Consulte Las 7 R.

versión

En un proceso de implementación, el acto de promover cambios en un entorno de producción.

reubicar

Consulte Las 7 R.

redefinir la plataforma

Consulte Las 7 R.

recomprar

Consulte Las 7 R.

resiliencia

Capacidad de una aplicación para resistir interrupciones o recuperarse de ellas. Al planificar la resiliencia en la Nube de AWS, la alta disponibilidad y la recuperación ante desastres son consideraciones comunes. Para más información, consulte Resiliencia en la Nube de AWS.

política basada en recursos

Una política asociada a un recurso, como un bucket de Amazon S3, un punto de conexión o una clave de cifrado. Este tipo de política especifica a qué entidades principales se les permite el acceso, las acciones compatibles y cualquier otra condición que deba cumplirse.

matriz responsable, confiable, consultada e informada (RACI)

Una matriz que define las funciones y responsabilidades de todas las partes involucradas en las actividades de migración y las operaciones de la nube. El nombre de la matriz se deriva de los tipos de responsabilidad definidos en la matriz: responsable (R), contable (A), consultado (C) e informado (I). El tipo de soporte (S) es opcional. Si incluye el soporte, la matriz se denomina matriz RASCI y, si la excluye, se denomina matriz RACI.

control receptivo

Un control de seguridad que se ha diseñado para corregir los eventos adversos o las desviaciones con respecto a su base de seguridad. Para obtener más información, consulte Controles receptivos en Implementación de controles de seguridad en AWS.

retain

Consulte Las 7 R.

retirar

Consulte Las 7 R.

Generación aumentada de recuperación (RAG)

Tecnología de IA generativa mediante la que un LLM hace referencia a un origen de datos autorizado que se encuentra fuera de sus orígenes de datos de entrenamiento antes de generar una respuesta. Por ejemplo, un modelo de RAG podría hacer una búsqueda semántica en la base de conocimientos o en los datos personalizados de una organización. Para más información, consulte ¿Qué es RAG (generación aumentada por recuperación)?

rotación

Proceso mediante el que periódicamente se actualiza un secreto para que resulte más difícil que un atacante pueda acceder a las credenciales.

control de acceso por filas y columnas (RCAC)

El uso de expresiones SQL básicas y flexibles que tienen reglas de acceso definidas. El RCAC consta de permisos de fila y máscaras de columnas.

RPO

Consulte objetivo de punto de recuperación.

RTO

Consulte objetivo de tiempo de recuperación.

manual de procedimientos

Conjunto de procedimientos manuales o automatizados necesarios para realizar una tarea específica. Por lo general, se diseñan para agilizar las operaciones o los procedimientos repetitivos con altas tasas de error.

S

SAML 2.0

Un estándar abierto que utilizan muchos proveedores de identidad (IdPs). Esta función permite el inicio de sesión único (SSO) federado, de modo que los usuarios pueden iniciar sesión Consola de administración de AWS o llamar a las operaciones de la AWS API sin tener que crear un usuario en IAM para todos los miembros de la organización. Para obtener más información sobre la federación basada en SAML 2.0, consulte Acerca de la federación basada en SAML 2.0 en la documentación de IAM.

SCADA

Consulte control de supervisión y adquisición de datos.

SCP

Consulte política de control de servicio.

secreta

En AWS Secrets Manager, información confidencial o restringida, como una contraseña o credenciales de usuario, que se almacena de forma cifrada. Se compone del valor del secreto y de sus metadatos. El valor del secreto puede ser binario, una sola cadena o varias cadenas. Para más información, consulte What's in a Secrets Manager secret? en la documentación de Secrets Manager.

seguridad desde el diseño

Enfoque de ingeniería de sistemas que tiene en cuenta la seguridad durante todo el proceso de desarrollo.

control de seguridad

Barrera de protección técnica o administrativa que impide, detecta o reduce la capacidad de un agente de amenazas para aprovechar una vulnerabilidad de seguridad. Existen cuatro tipos de controles de seguridad principales: preventivos, de detección, de respuesta y proactivos.

refuerzo de la seguridad

Proceso de reducir la superficie expuesta a ataques para hacerla más resistente a los ataques. Esto puede incluir acciones, como la eliminación de los recursos que ya no se necesitan, la implementación de prácticas recomendadas de seguridad consistente en conceder privilegios mínimos o la desactivación de características innecesarias en los archivos de configuración.

sistema de información sobre seguridad y administración de eventos (SIEM)

Herramientas y servicios que combinan sistemas de administración de información sobre seguridad (SIM) y de administración de eventos de seguridad (SEM). Un sistema de SIEM recopila, monitorea y analiza los datos de servidores, redes, dispositivos y otras fuentes para detectar amenazas y brechas de seguridad y generar alertas.

automatización de la respuesta de seguridad

Acción predefinida y programada que está diseñada para responder automáticamente a un evento de seguridad o corregirlo. Estas automatizaciones sirven como controles de seguridad preventivos o adaptables que le ayudan a implementar las mejores prácticas AWS de seguridad. La modificación de un grupo de seguridad de VPC, la aplicación de revisiones a una instancia de Amazon EC2 o la rotación de credenciales son algunos ejemplos de acciones de respuesta automatizadas.

cifrado del servidor

Cifrado de los datos en su destino, por parte de Servicio de AWS quien los recibe.

política de control de servicio (SCP)

Política que proporciona un control centralizado de los permisos de todas las cuentas de una organización en AWS Organizations. SCPs defina barreras o establezca límites a las acciones que un administrador puede delegar en usuarios o roles. Puede utilizarlas SCPs como listas de permitidos o rechazados para especificar qué servicios o acciones están permitidos o prohibidos. Para obtener más información, consulte las políticas de control de servicios en la AWS Organizations documentación.

punto de enlace de servicio

La URL del punto de entrada de un Servicio de AWS. Para conectarse mediante programación a un servicio de destino, puede utilizar un punto de conexión. Para obtener más información, consulte Puntos de conexión de Servicio de AWS en Referencia general de AWS.

acuerdo de nivel de servicio (SLA)

Acuerdo que aclara lo que un equipo de TI se compromete a ofrecer a los clientes, como el tiempo de actividad y el rendimiento del servicio.

indicador de nivel de servicio (SLI)

Medición de un aspecto del rendimiento de un servicio, como la tasa de errores, la disponibilidad o el rendimiento.

objetivo de nivel de servicio (SLO)

Métrica objetivo que representa el estado de un servicio medido mediante un indicador de nivel de servicio.

modelo de responsabilidad compartida

Un modelo que describe la responsabilidad con AWS la que compartes la seguridad y el cumplimiento de la nube. AWS es responsable de la seguridad de la nube, mientras que usted es responsable de la seguridad en la nube. Para obtener más información, consulte el Modelo de responsabilidad compartida.

SIEM

Consulte sistema de administración de eventos e información de seguridad.

único punto de error (SPOF)

Error en un único componente crítico de una aplicación que puede interrumpir el sistema.

SLA

Consulte acuerdo de nivel de servicio.

SLI

Consulte indicador de nivel de servicio.

SLO

Consulte objetivo de nivel de servicio.

split-and-seed modelo

Un patrón para escalar y acelerar los proyectos de modernización. A medida que se definen las nuevas funciones y los lanzamientos de los productos, el equipo principal se divide para

crear nuevos equipos de productos. Esto ayuda a ampliar las capacidades y los servicios de su organización, mejora la productividad de los desarrolladores y apoya la innovación rápida. Para más información, consulte Phased approach to modernizing applications in the Nube de AWS.

SPOF

Consulte único punto de error.

esquema en estrella

Estructura organizativa de una base de datos que utiliza una tabla de hechos de gran tamaño para almacenar datos transaccionales o medidos y una o varias tablas dimensionales más pequeñas para almacenar los atributos de los datos. Esta estructura está diseñada para utilizarse en un almacén de datos o con fines de inteligencia empresarial.

patrón de higo estrangulador

Un enfoque para modernizar los sistemas monolíticos mediante la reescritura y el reemplazo gradual de las funciones del sistema hasta que se pueda desmantelar el sistema heredado. Este patrón utiliza la analogía de una higuera que crece hasta convertirse en un árbol estable y, finalmente, se apodera y reemplaza a su host. El patrón fue presentado por Martin Fowler como una forma de gestionar el riesgo al reescribir sistemas monolíticos. Para ver un ejemplo con la aplicación de este patrón, consulte Modernización gradual de los servicios web antiguos de Microsoft ASP.NET (ASMX) mediante contenedores y Amazon API Gateway.

subred

Un intervalo de direcciones IP en la VPC. Una subred debe residir en una sola zona de disponibilidad.

control de supervisión y adquisición de datos (SCADA)

En el sector de la fabricación, sistema que utiliza hardware y software para supervisar los activos físicos y las operaciones de producción.

cifrado simétrico

Un algoritmo de cifrado que utiliza la misma clave para cifrar y descifrar los datos.

pruebas sintéticas

Prueba de un sistema de manera que simule las interacciones de los usuarios para detectar posibles problemas o supervisar el rendimiento. Puede usar Amazon CloudWatch Synthetics para crear estas pruebas.

petición del sistema

Técnica para proporcionar contexto, instrucciones o pautas a un LLM para dirigir su comportamiento. Las peticiones del sistema ayudan a establecer el contexto y las reglas para las interacciones con los usuarios.

T

etiquetas

Pares clave-valor que actúan como metadatos para organizar los recursos. AWS Las etiquetas pueden ayudar a administrar, identificar, organizar, buscar y filtrar recursos de . Para obtener más información, consulte Etiquetado de los recursos de AWS.

variable de destino

El valor que intenta predecir en el ML supervisado. Esto también se conoce como variable de resultado. Por ejemplo, en un entorno de fabricación, la variable objetivo podría ser un defecto del producto.

lista de tareas

Herramienta que se utiliza para hacer un seguimiento del progreso mediante un manual de procedimientos. La lista de tareas contiene una descripción general del manual de procedimientos y una lista de las tareas generales que deben completarse. Para cada tarea general, se incluye la cantidad estimada de tiempo necesario, el propietario y el progreso.

entorno de prueba

Consulte entorno.

entrenamiento

Proporcionar datos de los que pueda aprender su modelo de ML. Los datos de entrenamiento deben contener la respuesta correcta. El algoritmo de aprendizaje encuentra patrones en los datos de entrenamiento que asignan los atributos de los datos de entrada al destino (la respuesta que desea predecir). Genera un modelo de ML que captura estos patrones. Luego, el modelo de ML se puede utilizar para obtener predicciones sobre datos nuevos para los que no se conoce el destino.

puerta de enlace de tránsito

Un centro de tránsito de red que puede usar para interconectar sus redes con VPCs las locales. Para obtener más información, consulte Qué es una pasarela de tránsito en la AWS Transit Gateway documentación.

flujo de trabajo basado en enlaces troncales

Un enfoque en el que los desarrolladores crean y prueban características de forma local en una rama de característica y, a continuación, combinan esos cambios en la rama principal. Luego, la rama principal se adapta a los entornos de desarrollo, preproducción y producción, de forma secuencial.

acceso de confianza

Otorgar permisos a un servicio que especifique para realizar tareas en su organización AWS Organizations y en sus cuentas en su nombre. El servicio de confianza crea un rol vinculado al servicio en cada cuenta, cuando ese rol es necesario, para realizar las tareas de administración por usted. Para obtener más información, consulte AWS Organizations Utilización con otros AWS servicios en la AWS Organizations documentación.

ajuste

Cambiar aspectos de su proceso de formación a fin de mejorar la precisión del modelo de ML. Por ejemplo, puede entrenar el modelo de ML al generar un conjunto de etiquetas, incorporar etiquetas y, luego, repetir estos pasos varias veces con diferentes ajustes para optimizar el modelo.

equipo de dos pizzas

Un DevOps equipo pequeño al que puedes alimentar con dos pizzas. Un equipo formado por dos integrantes garantiza la mejor oportunidad posible de colaboración en el desarrollo de software.

U

incertidumbre

Un concepto que hace referencia a información imprecisa, incompleta o desconocida que puede socavar la fiabilidad de los modelos predictivos de ML. Hay dos tipos de incertidumbre: la incertidumbre epistémica se debe a datos limitados e incompletos, mientras que la incertidumbre aleatoria se debe al ruido y la aleatoriedad inherentes a los datos.

tareas indiferenciadas

También conocido como tareas arduas, es el trabajo que es necesario para crear y operar una aplicación, pero que no proporciona un valor directo al usuario final ni proporciona una ventaja competitiva. Algunos ejemplos de tareas indiferenciadas son la adquisición, el mantenimiento y la planificación de la capacidad.

entornos superiores

Consulte entorno.

V

succión

Una operación de mantenimiento de bases de datos que implica limpiar después de las actualizaciones incrementales para recuperar espacio de almacenamiento y mejorar el rendimiento.

control de versión

Procesos y herramientas que realizan un seguimiento de los cambios, como los cambios en el código fuente de un repositorio.

Emparejamiento de VPC

Una conexión entre dos VPCs que le permite enrutar el tráfico mediante direcciones IP privadas. Para obtener más información, consulte ¿Qué es una interconexión de VPC? en la documentación de Amazon VPC.

vulnerabilidad

Defecto de software o hardware que pone en peligro la seguridad del sistema.

W

caché caliente

Un búfer caché que contiene datos actuales y relevantes a los que se accede con frecuencia. La instancia de base de datos puede leer desde la caché del búfer, lo que es más rápido que leer desde la memoria principal o el disco.

datos templados

Datos a los que el acceso es infrecuente. Al consultar este tipo de datos, normalmente se aceptan consultas moderadamente lentas.

función de ventana

Función SQL que hace un cálculo en un grupo de filas que se relacionan de alguna manera con el registro actual. Las funciones de ventana son útiles para las tareas de procesamiento, como calcular una media móvil o acceder al valor de las filas en función de la posición relativa de la fila actual.

carga de trabajo

Conjunto de recursos y código que ofrece valor comercial, como una aplicación orientada al cliente o un proceso de backend.

flujo de trabajo

Grupos funcionales de un proyecto de migración que son responsables de un conjunto específico de tareas. Cada flujo de trabajo es independiente, pero respalda a los demás flujos de trabajo del proyecto. Por ejemplo, el flujo de trabajo de la cartera es responsable de priorizar las aplicaciones, planificar las oleadas y recopilar los metadatos de migración. El flujo de trabajo de la cartera entrega estos recursos al flujo de trabajo de migración, que luego migra los servidores y las aplicaciones.

WORM

Consulte escritura única y lectura múltiple.

WQF

Consulte AWS Workload Qualification Framework.

escritura única y lectura múltiple (WORM)

Modelo de almacenamiento que escribe los datos una sola vez y evita que se eliminen o modifiquen. Los usuarios autorizados pueden leer los datos tantas veces como sea necesario, pero no los pueden cambiar. Esta infraestructura de almacenamiento de datos se considera inmutable.

Z

ataque de día cero

Ataque, normalmente de malware, que se aprovecha de una vulnerabilidad de día cero.

vulnerabilidad de día cero

Un defecto o una vulnerabilidad sin mitigación en un sistema de producción. Los agentes de amenazas pueden usar este tipo de vulnerabilidad para atacar el sistema. Los desarrolladores suelen darse cuenta de la vulnerabilidad a raíz del ataque.

peticiones desde cero

Proporcionar a un LLM instrucciones para llevar a cabo una tarea, pero sin ejemplos (pasos) que puedan ayudar a guiarlo. El LLM debe usar los conocimientos del entrenamiento previo para llevar a cabo la tarea. La eficacia de la petición desde cero depende de la complejidad de la tarea y de la calidad de la petición. Consulte también peticiones con pocos pasos.

aplicación zombi

Aplicación que utiliza un promedio de CPU y memoria menor al 5 por ciento. En un proyecto de migración, es habitual retirar estas aplicaciones.

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