Mejores prácticas para usar el proveedor de Terraform AWS

AWS Guía prescriptiva

AWS Guía prescriptiva: Mejores prácticas para usar el proveedor de Terraform AWS

Table of Contents

Mejores prácticas para usar el proveedor de Terraform AWS

Michael Begin, DevOps consultor sénior de Amazon Web Services (AWS)

Agosto de 2025 (historial del documento)

La gestión de la infraestructura como código (IaC) con Terraform on AWS ofrece importantes beneficios, como una mayor coherencia, seguridad y agilidad. Sin embargo, a medida que la configuración de Terraform aumenta en tamaño y complejidad, se vuelve fundamental seguir las mejores prácticas para evitar problemas.

Esta guía proporciona las mejores prácticas recomendadas para utilizar el Terraform Provider desde AWS. HashiCorp Le explica el control de versiones, los controles de seguridad, los backends remotos, la estructura de la base de código y los proveedores comunitarios adecuados para optimizar Terraform. AWS En cada sección se profundiza en más detalles sobre los aspectos específicos de la aplicación de estas mejores prácticas:

- Seguridad
- Backends
- Estructura y organización de la base de código
- AWS Administración de versiones del proveedor
- Módulos comunitarios

Objetivos

Esta guía le ayuda a adquirir conocimientos operativos sobre el AWS proveedor de Terraform y aborda los siguientes objetivos empresariales que puede alcanzar siguiendo las mejores prácticas de la IaC en materia de seguridad, confiabilidad, cumplimiento y productividad de los desarrolladores.

- Mejore la calidad y la coherencia del código de infraestructura en todos los proyectos de Terraform.
- Acelere la incorporación de los desarrolladores y la capacidad de contribuir al código de infraestructura.
- Aumente la agilidad empresarial mediante cambios de infraestructura más rápidos.
- Reduzca los errores y el tiempo de inactividad relacionados con los cambios en la infraestructura.
- Optimice los costos de infraestructura siguiendo las mejores prácticas de la IaC.

- Refuerce su postura general de seguridad mediante la implementación de las mejores prácticas.

Público objetivo

El público objetivo de esta guía incluye líderes técnicos y gerentes que supervisan los equipos que utilizan Terraform para la IaC en adelante. AWS Otros lectores potenciales son los ingenieros de infraestructuras, los DevOps ingenieros, los arquitectos de soluciones y los desarrolladores que utilizan Terraform de forma activa para gestionar la infraestructura. AWS

Seguir estas mejores prácticas ahorrará tiempo y ayudará a aprovechar las ventajas de la IaC para estas funciones.

Información general

Los proveedores de Terraform son complementos que permiten a Terraform interactuar con diferentes API. El AWS proveedor de Terraform es el complemento oficial para administrar la AWS infraestructura como código (IaC) con Terraform. Traduce la sintaxis de Terraform en llamadas a la AWS API para crear, leer, actualizar y eliminar recursos. AWS

El AWS proveedor se encarga de la autenticación, traduce la sintaxis de Terraform en llamadas a la AWS API y aprovisiona los recursos. AWS Utiliza un bloque de `provider` código de Terraform para configurar el complemento del proveedor que Terraform utiliza para interactuar con la API. AWS Puedes configurar varios bloques AWS de proveedores para administrar los recursos en diferentes Cuentas de AWS regiones.

A continuación, se muestra un ejemplo de configuración de Terraform que usa varios bloques de AWS proveedores con alias para administrar una base de datos de Amazon Relational Database Service (Amazon RDS) que tiene una réplica en una región y una cuenta diferentes. Los proveedores principal y secundario asumen diferentes funciones AWS Identity and Access Management (IAM):

```
# Configure the primary AWS Provider
provider "aws" {
  region = "us-west-1"
  alias  = "primary"
}

# Configure a secondary AWS Provider for the replica Region and account
provider "aws" {
  region      = "us-east-1"
  alias       = "replica"
  assume_role {
    role_arn     = "arn:aws:iam::<replica-account-id>:role/<role-name>"
    session_name = "terraform-session"
  }
}

# Primary Amazon RDS database
resource "aws_db_instance" "primary" {
  provider = aws.primary

  # ... RDS instance configuration
}
```

```
# Read replica in a different Region and account
resource "aws_db_instance" "read_replica" {
  provider = aws.replica

  # ... RDS read replica configuration
  replicate_source_db = aws_db_instance.primary.id
}
```

En este ejemplo:

- El primer `provider` bloque configura el AWS proveedor principal de la `us-west-1` región con el alias. `primary`
- El segundo `provider` bloque configura un AWS proveedor secundario en la `us-east-1` región con el alias. `replica` Este proveedor se utiliza para crear una réplica de lectura de la base de datos principal en una región y una cuenta diferentes. El `assume_role` bloque se utiliza para asumir una función de IAM en la cuenta de réplica. `role_arn`Especifica el nombre de recurso de Amazon (ARN) de la función de IAM que se va a asumir y `session_name` es un identificador único para la sesión de Terraform.
- El `aws_db_instance.primary` recurso crea la base de datos principal de Amazon RDS mediante el `primary` proveedor de la `us-west-1` región.
- El `aws_db_instance.read_replica` recurso crea una réplica de lectura de la base de datos principal de la `us-east-1` región mediante el `replica` proveedor. El `replicate_source_db` atributo hace referencia al ID de la `primary` base de datos.

Prácticas recomendadas de seguridad

Administrar adecuadamente la autenticación, los controles de acceso y la seguridad es fundamental para un uso seguro del Terraform Provider. AWS En esta sección se describen las mejores prácticas en torno a:

- Funciones y permisos de IAM para el acceso con privilegios mínimos
- Proteger las credenciales para evitar el acceso no autorizado a las cuentas y los recursos AWS
- Cifrado remoto de estado para ayudar a proteger los datos confidenciales
- Escaneo de infraestructura y código fuente para identificar errores de configuración
- Controles de acceso para el almacenamiento de estado remoto
- Aplicación de políticas centinela para implementar barreras de gobernabilidad

Seguir estas prácticas recomendadas le ayudará a reforzar su postura de seguridad cuando utilice Terraform para gestionar la infraestructura. AWS

Siga el principio del mínimo privilegio

El privilegio mínimo es un principio de seguridad fundamental que se refiere a conceder solo los permisos mínimos necesarios para que un usuario, proceso o sistema desempeñe las funciones previstas. Es un concepto fundamental del control de acceso y una medida preventiva contra el acceso no autorizado y las posibles violaciones de datos.

En esta sección se hace hincapié en el principio del mínimo privilegio porque está directamente relacionado con la forma en que Terraform autentica y emprende acciones contra los proveedores de servicios en la nube, por ejemplo. AWS

Cuando utilizas Terraform para aprovisionar y administrar AWS recursos, actúa en nombre de una entidad (usuario o función) que requiere los permisos adecuados para realizar llamadas a la API. No respetar el mínimo privilegio conlleva importantes riesgos de seguridad:

- Si Terraform tiene permisos excesivos más allá de lo necesario, un error de configuración no intencionado podría provocar cambios o eliminaciones no deseados.
- Las concesiones de acceso excesivamente permisivas aumentan el alcance del impacto si los archivos estatales o las credenciales de Terraform se ven comprometidos.

- No respetar el más mínimo privilegio va en contra de las mejores prácticas de seguridad y de los requisitos de cumplimiento normativo a la hora de conceder el acceso mínimo requerido.

Uso de roles de IAM

Utilice funciones de IAM en lugar de usuarios de IAM siempre que sea posible para mejorar la seguridad con Terraform AWS Provider. Los roles de IAM proporcionan credenciales de seguridad temporales que se cambian automáticamente, lo que elimina la necesidad de administrar las claves de acceso a largo plazo. Los roles también ofrecen controles de acceso precisos mediante políticas de IAM.

Otorgue el acceso con privilegios mínimos mediante políticas de IAM

Elabore cuidadosamente las políticas de IAM para garantizar que los roles y los usuarios solo tengan el conjunto mínimo de permisos necesarios para su carga de trabajo. Comience con una política vacía y añada de forma iterativa los servicios y acciones permitidos. Para lograrlo:

- Habilite IAM Access Analyzer para evaluar las políticas y destacar los permisos no utilizados que se pueden eliminar.
- Revise manualmente las políticas para eliminar cualquier funcionalidad que no sea esencial para la responsabilidad prevista del puesto.
- Utilice las variables y etiquetas de las políticas de IAM para simplificar la administración de permisos.

Las políticas bien diseñadas otorgan el acceso justo para cumplir con las responsabilidades de la carga de trabajo y nada más. Defina las acciones a nivel operativo y permita las llamadas solo cuando se requieran APIs recursos específicos.

Seguir esta mejor práctica reduce el alcance del impacto y sigue los principios de seguridad fundamentales de la separación de funciones y el acceso con privilegios mínimos. Inicie el acceso estricto y abierto gradualmente según sea necesario, en lugar de empezar de forma abierta e intentar restringir el acceso más adelante.

Asuma las funciones de IAM para la autenticación local

Cuando ejecute Terraform de forma local, evite configurar claves de acceso estáticas. En su lugar, utilice las funciones de IAM para conceder un acceso privilegiado de forma temporal sin exponer las credenciales a largo plazo.

En primer lugar, cree un rol de IAM con los permisos mínimos necesarios y añada una relación de confianza que permita que su cuenta de usuario o identidad federada asuma el rol de IAM. Esto autoriza el uso temporal del rol.

Ejemplo de política de relaciones de confianza:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::111122223333:role/terraform-execution"
      },
      "Action": "sts:AssumeRole"
    }
  ]
}
```

A continuación, ejecute el AWS CLI comando aws sts assume-role para recuperar las credenciales efímeras del rol. Estas credenciales suelen ser válidas durante una hora.

AWS CLI ejemplo de comando:

```
aws sts assume-role --role-arn arn:aws:iam::111122223333:role/terraform-execution --
role-session-name terraform-session-example
```

El resultado del comando contiene una clave de acceso, una clave secreta y un token de sesión que puede usar para autenticarse AWS en:

```
{
    "AssumedRoleUser": {
        "AssumedRoleId": "AROA3XFRBF535PLBIFPI4:terraform-session-example",
        "Arn": "arn:aws:sts::111122223333:assumed-role/terraform-execution/terraform-
session-example"
```

```
    },
    "Credentials": {
        "SecretAccessKey": " wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY",
        "SessionToken": " AQoEXAMPLEH4aoAH0gNCAPyJxz4BlCFFxWNE1OPTgk5TthT
+FvwqnKwRcOIfrRh3c/LTo6UDdyJwOOvEVPvLXCrrrUtdnniCEXAMPLE/
IvU1dYUg2RVAJBanLiHb4IgRmpRV3zrkuWJOgQs8IZZaIv2BXIa2R4OlgkBN9bkUDNCJiBeb/
AXlzBBko7b15fjrBs2+cTQtpZ3CYWFXG8C5zqx37wnOE49mRl/+OtkIKGO7fAE",
        "Expiration": "2024-03-15T00:05:07Z",
        "AccessKeyId": "ASIAIOSFODNN7EXAMPLE"
    }
}
```

El AWS proveedor también puede encargarse automáticamente de asumir el rol.

Ejemplo de configuración del proveedor para asumir una función de IAM:

```
provider "aws" {
  assume_role {
    role_arn     = "arn:aws:iam::111122223333:role/terraform-execution"
    session_name = "terraform-session-example"
  }
}
```

Esto otorga privilegios elevados estrictamente durante la sesión de Terraform. Las claves temporales no se pueden filtrar porque caducan automáticamente después de la duración máxima de la sesión.

Las principales ventajas de esta práctica recomendada son la mejora de la seguridad en comparación con las claves de acceso de larga duración, los controles de acceso detallados sobre el rol con menos privilegios y la posibilidad de revocar fácilmente el acceso modificando los permisos del rol. Al utilizar las funciones de IAM, también se evita tener que almacenar los secretos directamente de forma local en scripts o en el disco, lo que permite compartir la configuración de Terraform de forma segura con todo el equipo.

Utilice funciones de IAM para la autenticación de Amazon EC2

Cuando ejecute Terraform desde instancias de Amazon Elastic Compute Cloud (Amazon EC2), evite almacenar localmente las credenciales de larga duración. En su lugar, usa roles de IAM y perfiles de instancia para conceder automáticamente los permisos con privilegios mínimos.

En primer lugar, cree un rol de IAM con los permisos mínimos y asígnelo al perfil de la instancia. El perfil de instancia permite a las instancias de EC2 heredar los permisos definidos en el rol. A

continuación, lance las instancias especificando ese perfil de instancia. La instancia se autenticará a través del rol adjunto.

Antes de ejecutar cualquier operación de Terraform, verifica que el rol esté presente en los metadatos de la instancia para confirmar que las credenciales se heredaron correctamente.

```
TOKEN=$(curl -s -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-
metadata-token-ttl-seconds: 21600")

curl -H "X-aws-ec2-metadata-token: $TOKEN" -s http://169.254.169.254/latest/meta-data/
iam/security-credentials/
```

Este enfoque evita codificar AWS las claves permanentes en los scripts o en la configuración de Terraform dentro de la instancia. Las credenciales temporales se ponen a disposición de Terraform de forma transparente a través del rol y el perfil de la instancia.

Los principales beneficios de esta mejor práctica incluyen una mayor seguridad en comparación con las credenciales a largo plazo, una reducción de la sobrecarga de administración de credenciales y la coherencia entre los entornos de desarrollo, prueba y producción. La autenticación de roles de IAM simplifica las ejecuciones de Terraform desde instancias EC2 y, al mismo tiempo, impone el acceso con privilegios mínimos.

Utilice credenciales dinámicas para los espacios de trabajo de HCP Terraform

HCP Terraform es un servicio gestionado HashiCorp que ayuda a los equipos a utilizar Terraform para aprovisionar y gestionar la infraestructura en varios proyectos y entornos. Cuando ejecute Terraform en HCP Terraform, utilice credenciales dinámicas para simplificar y proteger la autenticación. AWS Terraform intercambia automáticamente las credenciales temporales en cada ejecución sin necesidad de asumir el rol de IAM.

Los beneficios incluyen una rotación de secretos más sencilla, una gestión centralizada de las credenciales en todos los espacios de trabajo, los permisos con los mínimos privilegios y la eliminación de las claves codificadas. Confiar en claves efímeras cifradas mejora la seguridad en comparación con las claves de acceso de larga duración.

Utilice las funciones de IAM en AWS CodeBuild

En AWS CodeBuild, ejecuta tus compilaciones con una función de IAM asignada al CodeBuild proyecto. Esto permite que cada compilación herede automáticamente las credenciales temporales del rol en lugar de utilizar claves de larga duración.

Ejecute GitHub acciones de forma remota en HCP Terraform

Configure los flujos de trabajo de GitHub Actions para ejecutar Terraform de forma remota en los espacios de trabajo de HCP Terraform. Confíe en las credenciales dinámicas y en el bloqueo remoto del estado en lugar de en la gestión de secretos. GitHub

Utilice GitHub las acciones con el OIDC y configure la AWS acción Credenciales

Utilice el estándar OpenID Connect (OIDC) para federar GitHub la identidad de Actions a través de IAM. Utilice la acción Configurar AWS credenciales para cambiar el GitHub token por AWS credenciales temporales sin necesidad de claves de acceso de larga duración.

Úselo GitLab con el OIDC y el AWS CLI

Utilice el estándar OIDC para federar GitLab identidades a través de IAM para el acceso temporal. Al confiar en el OIDC, se evita tener que gestionar directamente las claves de acceso a largo plazo desde su interior. AWS GitLab Se intercambian las credenciales just-in-time, lo que mejora la seguridad. Los usuarios también obtienen el acceso con los privilegios mínimos según los permisos de la función de IAM.

Utilice usuarios de IAM exclusivos con herramientas de automatización antiguas

Si dispone de scripts y herramientas de automatización que no admiten de forma nativa el uso de funciones de IAM, puede crear usuarios de IAM individuales para conceder acceso mediante programación. Se sigue aplicando el principio del privilegio mínimo. Minimice los permisos de las políticas y utilice funciones distintas para cada proceso o secuencia de comandos. A medida que vaya migrando a herramientas o scripts más modernos, comience a desempeñar las funciones de soporte de forma nativa y vaya pasando gradualmente a ellas.

> **Warning**
>
> Los usuarios de IAM tienen credenciales de larga duración, lo que supone un riesgo para la seguridad. Para ayudar a mitigar este riesgo, le recomendamos que brinde a estos usuarios únicamente los permisos que necesitan para realizar la tarea y que los elimine cuando ya no los necesiten.

Utilice el complemento Jenkins Credentials AWS

Usa el complemento AWS Credentials de Jenkins para configurar e inyectar AWS credenciales en las compilaciones de forma centralizada y dinámica. Esto evita tener que registrar los secretos en el control de código fuente.

Supervise, valide y optimice continuamente los privilegios mínimos

Con el tiempo, es posible que se concedan permisos adicionales que pueden superar las políticas mínimas requeridas. Analice continuamente el acceso para identificar y eliminar cualquier derecho innecesario.

Supervise continuamente el uso de las claves de acceso

Si no puede evitar el uso de claves de acceso, utilice los informes de credenciales de IAM para encontrar las claves de acceso no utilizadas que tengan más de 90 días de antigüedad y revoque las claves inactivas tanto en las cuentas de usuario como en las funciones de la máquina. Avise a los administradores para que confirmen manualmente la retirada de las claves de los empleados y sistemas activos.

Supervisar el uso de las claves le ayuda a optimizar los permisos, ya que puede identificar y eliminar los derechos no utilizados. Si sigue esta práctica recomendada con la rotación de claves de acceso, se limita la vida útil de las credenciales y se impone el acceso con privilegios mínimos.

AWS proporciona varios servicios y funciones que puede utilizar para configurar alertas y notificaciones para los administradores. Estas son algunas de las opciones:

- AWS Config: Puede usar AWS Config reglas para evaluar los ajustes de configuración de sus AWS recursos, incluidas las claves de acceso de IAM. Puede crear reglas personalizadas para comprobar si hay condiciones específicas, como claves de acceso no utilizadas que tengan una

antigüedad superior a un número específico de días. Cuando se infringe una regla, AWS Config puede iniciar una evaluación para corregirla o enviar notificaciones a un tema del Amazon Simple Notification Service (Amazon SNS).

- AWS Security Hub CSPM: Security Hub CSPM proporciona una visión completa del estado de seguridad de su AWS cuenta y puede ayudarle a detectar y notificarle posibles problemas de seguridad, incluidas las claves de acceso de IAM inactivas o no utilizadas. Security Hub CSPM puede integrarse con Amazon EventBridge y Amazon SNS o Amazon Q Developer en aplicaciones de chat para enviar notificaciones a los administradores.
- AWS Lambda: Las funciones Lambda se pueden llamar mediante varios eventos, incluidos Amazon CloudWatch Events o AWS Config reglas. Puede escribir funciones Lambda personalizadas para evaluar el uso de las claves de acceso de IAM, realizar comprobaciones adicionales y enviar notificaciones mediante servicios como Amazon SNS o Amazon Q Developer en aplicaciones de chat.

Valide continuamente las políticas de IAM

Utilice IAM Access Analyzer para evaluar las políticas asociadas a las funciones e identificar los servicios no utilizados o las acciones excesivas que se hayan otorgado. Implemente revisiones de acceso periódicas para verificar manualmente que las políticas cumplan con los requisitos actuales.

Compare la política existente con la política generada por IAM Access Analyzer y elimine los permisos innecesarios. También debe proporcionar informes a los usuarios y revocar automáticamente los permisos no utilizados tras un período de gracia. Esto ayuda a garantizar que las políticas mínimas permanezcan en vigor.

La revocación proactiva y frecuente del acceso obsoleto minimiza las credenciales que podrían estar en riesgo en caso de producirse una infracción. La automatización proporciona una optimización sostenible y a largo plazo de la higiene de las credenciales y los permisos. Seguir esta mejor práctica limita el alcance del impacto al aplicar de forma proactiva el mínimo privilegio en todas AWS las identidades y los recursos.

Almacenamiento remoto seguro

El almacenamiento de estado remoto se refiere a almacenar el archivo de estado de Terraform de forma remota en lugar de hacerlo localmente en la máquina en la que se ejecuta Terraform. El archivo de estado es crucial porque realiza un seguimiento de los recursos aprovisionados por Terraform y sus metadatos.

Si no se protege el estado remoto, se pueden producir problemas graves, como la pérdida de datos de estado, la imposibilidad de gestionar la infraestructura, la eliminación accidental de recursos y la exposición de información confidencial que podría estar presente en el archivo de estado. Por esta razón, proteger el almacenamiento remoto remoto es crucial para el uso de Terraform a nivel de producción.

Habilite el cifrado y los controles de acceso

Utilice el cifrado del lado del servidor (SSE) de Amazon Simple Storage Service (Amazon S3) para cifrar el estado remoto en reposo.

Limite el acceso directo a los flujos de trabajo colaborativos

- Estructura los flujos de trabajo de colaboración en HCP Terraform o en una CI/CD canalización dentro de tu repositorio de Git para limitar el acceso directo al estado.
- Confíe en las solicitudes de incorporación de datos, las aprobaciones de ejecución, las comprobaciones de políticas y las notificaciones para coordinar los cambios.

Seguir estas pautas ayuda a proteger los atributos confidenciales de los recursos y evita conflictos con los cambios de los miembros del equipo. El cifrado y las estrictas protecciones de acceso ayudan a reducir la superficie de ataque, y los flujos de trabajo de colaboración aumentan la productividad.

Utilice AWS Secrets Manager

Hay muchos recursos y fuentes de datos en Terraform que almacenan valores secretos en texto plano en el archivo estatal. Evite almacenar los secretos en el estado AWS Secrets Manager; utilícelos en su lugar.

En lugar de intentar cifrar manualmente los valores confidenciales, confíe en el soporte integrado de Terraform para la gestión de estados confidenciales. Al exportar valores confidenciales a la salida, asegúrese de que los valores estén marcados como confidenciales.

Escanee continuamente la infraestructura y el código fuente

Escanee proactivamente la infraestructura y el código fuente de forma continua para detectar riesgos, como credenciales expuestas o errores de configuración, a fin de reforzar su postura de seguridad. Aborde los hallazgos rápidamente reconfigurando o parcheando los recursos.

Utilice los AWS servicios de escaneo dinámico

Usa herramientas AWS nativas como Amazon Inspector AWS Security Hub CSPM, Amazon Detective y Amazon GuardDuty para supervisar la infraestructura aprovisionada en todas las cuentas y regiones. Programe escaneos periódicos en Security Hub CSPM para realizar un seguimiento de los cambios en la implementación y la configuración. Escanee instancias EC2, funciones Lambda, contenedores, depósitos S3 y otros recursos.

Realice un análisis estático

Integre analizadores estáticos como Checkov directamente en CI/CD las canalizaciones para escanear el código de configuración (HCL) de Terraform e identificar los riesgos de forma preventiva antes del despliegue. De este modo, los controles de seguridad se trasladan a una fase anterior del proceso de desarrollo (lo que se denomina desplazamiento a la izquierda) y se evita que la infraestructura esté mal configurada.

Garantice una pronta reparación

En el caso de todos los resultados del análisis, asegúrese de que se corrijan rápidamente actualizando la configuración de Terraform, aplicando parches o reconfigurando los recursos manualmente, según proceda. Reduzca los niveles de riesgo abordando las causas fundamentales.

Al utilizar tanto el escaneo de infraestructura como el escaneo de códigos, se obtiene información detallada sobre las configuraciones de Terraform, los recursos aprovisionados y el código de la aplicación. Esto maximiza la cobertura del riesgo y el cumplimiento mediante controles preventivos, de detección y reactivos, al tiempo que incorpora la seguridad en una fase más temprana del ciclo de vida del desarrollo del software (SDLC).

Aplique controles de políticas

Utilice marcos de código, como las políticas de HashiCorp Sentinel, para proporcionar barreras de gobernanza y plantillas estandarizadas para el aprovisionamiento de infraestructuras con Terraform.

Las políticas de Sentinel pueden definir los requisitos o restricciones de la configuración de Terraform para alinearlos con los estándares y las mejores prácticas de la organización. Por ejemplo, puede usar las políticas de Sentinel para:

- Exija etiquetas en todos los recursos.

- Restrinja los tipos de instancias a una lista aprobada.
- Aplica las variables obligatorias.
- Impedir la destrucción de los recursos de producción.

La integración de las comprobaciones de políticas en los ciclos de vida de la configuración de Terraform permite la aplicación proactiva de los estándares y las directrices de arquitectura. Sentinel proporciona una lógica de políticas compartida que ayuda a acelerar el desarrollo y, al mismo tiempo, evita prácticas no aprobadas.

Mejores prácticas de backend

El uso de un servidor remoto adecuado para almacenar el archivo de estado es fundamental para permitir la colaboración, garantizar la integridad de los archivos estatales mediante el bloqueo, proporcionar copias de seguridad y recuperación confiables, integrarse con los CI/CD flujos de trabajo y aprovechar las funciones avanzadas de seguridad, gobierno y administración que ofrecen los servicios gestionados como HCP Terraform.

Terraform admite varios tipos de backend, como Kubernetes, Consul y HTTP. HashiCorp Sin embargo, esta guía se centra en Amazon S3, que es una solución de backend óptima para la mayoría de AWS los usuarios.

Como servicio de almacenamiento de objetos totalmente gestionado que ofrece una alta durabilidad y disponibilidad, Amazon S3 proporciona un backend seguro, escalable y de bajo coste para gestionar el estado de Terraform. AWS La presencia global y la resiliencia de Amazon S3 superan lo que la mayoría de los equipos pueden lograr mediante la autogestión del almacenamiento estatal. Además, al estar integrado de forma nativa con los controles de AWS acceso, las opciones de cifrado, las capacidades de control de versiones y otros servicios, convierte a Amazon S3 en una cómoda opción de backend.

Esta guía no proporciona orientación de backend para otras soluciones, como Kubernetes o Consul, porque el público objetivo principal son los clientes. AWS Para los equipos que están totalmente ocupados Nube de AWS, Amazon S3 suele ser la opción ideal en lugar de los clústeres de Kubernetes o HashiCorp Consul. La simplicidad, la resiliencia y la estrecha AWS integración del almacenamiento estatal de Amazon S3 proporcionan una base óptima para la mayoría de los usuarios que siguen las AWS mejores prácticas. Los equipos pueden aprovechar la durabilidad, las protecciones de respaldo y la disponibilidad de AWS los servicios para mantener una alta resiliencia del estado remoto de Terraform.

Seguir las recomendaciones del backend de esta sección permitirá que las bases de código de Terraform sean más colaborativas y, al mismo tiempo, se limitará el impacto de los errores o las modificaciones no autorizadas. Al implementar un backend remoto bien diseñado, los equipos pueden optimizar los flujos de trabajo de Terraform.

Mejores prácticas:

- Utilice Amazon S3 para el almacenamiento remoto
- Facilite la colaboración en equipo

- Separe los backends de cada entorno
- Supervise activamente la actividad de estado remoto

Utilice Amazon S3 para el almacenamiento remoto

El almacenamiento remoto del estado de Terraform en Amazon S3 y la implementación del bloqueo del estado y la comprobación de coherencia mediante Amazon DynamoDB ofrecen ventajas importantes en comparación con el almacenamiento local de archivos. El estado remoto permite la colaboración en equipo, el seguimiento de los cambios, las protecciones de respaldo y el bloqueo remoto para aumentar la seguridad.

El uso de Amazon S3 con la clase de almacenamiento estándar S3 (predeterminada) en lugar de soluciones de almacenamiento local efímero o autogestionadas proporciona protecciones de durabilidad del 99,19% y disponibilidad del 99,99% para evitar la pérdida accidental de datos por estado. AWS los servicios gestionados, como Amazon S3 y DynamoDB, ofrecen acuerdos de nivel de servicio SLAs () que superan lo que la mayoría de las organizaciones pueden lograr cuando autogestionan el almacenamiento. Confíe en estas protecciones para mantener accesibles los backends remotos.

Habilite el bloqueo de estado remoto

El bloqueo por estado restringe el acceso para evitar operaciones de escritura simultáneas y reduce los errores derivados de las modificaciones simultáneas realizadas por varios usuarios. Terraform admite dos mecanismos de bloqueo para los backends de Amazon S3:

- Bloqueo de estado nativo de Amazon S3 (recomendado): disponible desde Terraform 1.10.0; utiliza las capacidades de bloqueo nativas de Amazon S3
- Bloqueo de estado de DynamoDB (obsoleto): enfoque heredado que se eliminará en futuras versiones de Terraform

```
terraform {
  backend "s3" {
    bucket       = "myorg-terraform-states"
    key          = "myapp/production/tfstate"
    region       = "us-east-1"
    use_lockfile = true
  }
```

```
}
```

Para garantizar la compatibilidad con versiones anteriores durante la migración, puede configurar el bloqueo de Amazon S3 y DynamoDB simultáneamente. Sin embargo, le recomendamos que migre al bloqueo nativo de Amazon S3, ya que el bloqueo basado en DynamoDB está en desuso.

Habilite el control de versiones y las copias de seguridad automáticas

Para una protección adicional, habilite el control de versiones y las copias de seguridad automáticas mediante AWS Backup el uso de los backends de Amazon S3. El control de versiones preserva todas las versiones anteriores del estado siempre que se realicen cambios. También le permite restaurar las instantáneas de los estados de trabajo anteriores si es necesario para revertir los cambios no deseados o recuperarse de un accidente.

Restaure las versiones anteriores si es necesario

Los buckets de estado versionados de Amazon S3 facilitan la reversión de los cambios mediante la restauración de una instantánea anterior de buen estado conocida. Esto ayuda a proteger contra cambios accidentales y proporciona capacidades de respaldo adicionales.

Utilice HCP Terraform

HCP Terraform ofrece una alternativa de backend totalmente gestionada a la configuración de su propio almacenamiento estatal. HCP Terraform gestiona automáticamente el almacenamiento seguro del estado y el cifrado y, al mismo tiempo, desbloquea funciones adicionales.

Al utilizar HCP Terraform, el estado se almacena de forma remota de forma predeterminada, lo que permite compartir y bloquear el estado en toda la organización. Los controles detallados de las políticas le ayudan a restringir el acceso y los cambios estatales.

Las capacidades adicionales incluyen integraciones de control de versiones, políticas de protección, automatización del flujo de trabajo, administración de variables e integraciones de inicio de sesión único con SAML. También puedes usar la política de Sentinel como código para implementar controles de gobierno.

Si bien HCP Terraform requiere el uso de una plataforma de software como servicio (SaaS), para muchos equipos las ventajas en torno a la seguridad, los controles de acceso, las comprobaciones automatizadas de políticas y las funciones de colaboración la convierten en una opción óptima en lugar del almacenamiento estatal autogestionado con Amazon S3 o DynamoDB.

La fácil integración con servicios como GitHub , por ejemplo, GitLab con configuraciones menores también atrae a los usuarios que utilizan plenamente las herramientas SaaS y en la nube para mejorar los flujos de trabajo en equipo.

Facilite la colaboración en equipo

Usa backends remotos para compartir los datos de estado entre todos los miembros de tu equipo de Terraform. Esto facilita la colaboración porque brinda a todo el equipo visibilidad de los cambios en la infraestructura. Los protocolos de backend compartidos, combinados con la transparencia del historial estatal, simplifican la gestión de los cambios internos. Todos los cambios en la infraestructura pasan por el proceso establecido, lo que aumenta la agilidad empresarial en toda la empresa.

Mejore la responsabilidad mediante el uso de AWS CloudTrail

Intégrelo AWS CloudTrail con el bucket de Amazon S3 para capturar las llamadas a la API realizadas al bucket de estado. Filtre CloudTrail los eventos para realizar un seguimiento `PutObject` `DeleteObject,` y otras llamadas relevantes.

CloudTrail Los registros muestran la AWS identidad del director que hizo que cada API solicitara un cambio de estado. La identidad del usuario puede compararse con la de una cuenta de máquina o con la de los miembros del equipo que interactúan con el almacenamiento interno.

Combine CloudTrail los registros con el control de versiones estatales de Amazon S3 para vincular los cambios de infraestructura con el director que los aplicó. Al analizar varias revisiones, puede atribuir las actualizaciones a la cuenta de la máquina o al miembro responsable del equipo.

Si se produce un cambio imprevisto o perjudicial, el control de versiones por estado proporciona funciones de reversión. CloudTrail rastrea el cambio hasta el usuario para que pueda analizar las mejoras preventivas.

También le recomendamos que aplique los permisos de IAM para limitar el acceso a los buckets estatales. En general, el control de versiones y la CloudTrail supervisión de S3 permiten la auditoría de todos los cambios de infraestructura. Los equipos adquieren mejores capacidades de rendición de cuentas, transparencia y auditoría a lo largo de la historia del estado de Terraform.

Separe los backends de cada entorno

Utilice distintos backends de Terraform para cada entorno de aplicación. Los backends separados aíslan el estado entre el desarrollo, las pruebas y la producción.

Reduzca el alcance del impacto

El estado de aislamiento ayuda a garantizar que los cambios en los entornos inferiores no afecten a la infraestructura de producción. Los accidentes o los experimentos en los entornos de desarrollo y prueba tienen un impacto limitado.

Limite el acceso a la producción

Limite los permisos del backend del estado de producción para que la mayoría de los usuarios puedan acceder a ellos en modo de solo lectura. Limite las funciones de quién puede modificar la infraestructura de producción a la CI/CD cartera y romper el vidrio.

Simplifique los controles de acceso

La administración de los permisos a nivel de back-end simplifica el control de acceso entre entornos. El uso de buckets S3 distintos para cada aplicación y entorno permite conceder amplios permisos de lectura o escritura en todos los buckets de backend.

Evite los espacios de trabajo compartidos

Si bien puedes usar los espacios de trabajo de Terraform para separar los estados de los entornos, los distintos backends proporcionan un mayor aislamiento. Si ha compartido espacios de trabajo, los accidentes pueden afectar a varios entornos.

Mantener los backends del entorno completamente aislados minimiza el impacto de cualquier fallo o infracción individual. Los backends independientes también alinean los controles de acceso con el nivel de sensibilidad del entorno. Por ejemplo, puede proporcionar protección contra escritura para el entorno de producción y un acceso más amplio para los entornos de desarrollo y prueba.

Supervise activamente la actividad de estado remoto

La supervisión continua de la actividad del estado remoto es fundamental para detectar posibles problemas de forma temprana. Busque desbloqueos, cambios o intentos de acceso anómalos.

Recibe alertas sobre desbloqueos sospechosos

La mayoría de los cambios de estado deberían realizarse por CI/CD procesos. Genera alertas si los desbloqueos de estado se producen directamente en las estaciones de trabajo de los desarrolladores, lo que podría indicar cambios no autorizados o no probados.

Supervise los intentos de acceso

Los errores de autenticación en los depósitos de estado pueden indicar una actividad de reconocimiento. Observa si varias cuentas están intentando acceder al estado o si aparecen direcciones IP inusuales, lo que indica que las credenciales están comprometidas.

Mejores prácticas para la estructura y organización de la base de código

La estructura y la organización adecuadas de la base de código son fundamentales a medida que el uso de Terraform crece en grandes equipos y empresas. Una base de código bien diseñada permite la colaboración a escala y, al mismo tiempo, mejora la capacidad de mantenimiento.

Esta sección proporciona recomendaciones sobre la modularidad de Terraform, las convenciones de nomenclatura, la documentación y los estándares de codificación que respaldan la calidad y la coherencia.

Las directrices incluyen dividir la configuración en módulos reutilizables por entorno y componentes, establecer convenciones de nomenclatura mediante el uso de prefijos y sufijos, documentar los módulos y explicar con claridad las entradas y salidas, y aplicar reglas de formato coherentes mediante comprobaciones de estilo automatizadas.

Otras buenas prácticas incluyen la organización lógica de los módulos y recursos en una jerarquía estructurada, la catalogación de los módulos públicos y privados en la documentación y la abstracción de los detalles de implementación innecesarios en los módulos para simplificar el uso.

Al implementar pautas de estructura de código base en torno a la modularidad, la documentación, los estándares y la organización lógica, puede respaldar una amplia colaboración entre los equipos y, al mismo tiempo, mantener Terraform mantenible a medida que el uso se extiende por toda la organización. Al aplicar las convenciones y los estándares, puede evitar la complejidad de una base de código fragmentada.

Prácticas recomendadas:

- Implemente una estructura de repositorio estándar
- Estructura de modularidad
- Siga las convenciones de nomenclatura
- Utilice los recursos de archivos adjuntos
- Usa etiquetas predeterminadas
- Cumple con los requisitos de registro de Terraform
- Utilice las fuentes de módulos recomendadas
- Siga los estándares de codificación

Implemente una estructura de repositorio estándar

Se recomienda implementar el siguiente diseño de repositorio. La estandarización de estas prácticas de coherencia en todos los módulos mejora la capacidad de descubrimiento, la transparencia, la organización y la confiabilidad, al tiempo que permite la reutilización en muchas configuraciones de Terraform.

- Módulo o directorio raíz: este debería ser el punto de entrada principal tanto para los módulos raíz como para los reutilizables de Terraform y se espera que sea único. Si tienes una arquitectura más compleja, puedes usar módulos anidados para crear abstracciones ligeras. Esto le ayuda a describir la infraestructura en términos de su arquitectura en lugar de hacerlo directamente, en términos de objetos físicos.
- README: el módulo raíz y cualquier módulo anidado deben tener archivos README. Este archivo debe tener un nombre. `README.md` Debe contener una descripción del módulo y para qué se debe utilizar. Si desea incluir un ejemplo del uso de este módulo con otros recursos, colóquelo en un `examples` directorio. Considere la posibilidad de incluir un diagrama que describa los recursos de infraestructura que el módulo podría crear y sus relaciones. Utilice terraform-docs para generar automáticamente entradas o salidas del módulo.
- main.tf: Este es el punto de entrada principal. Para un módulo simple, todos los recursos pueden crearse en este archivo. En el caso de un módulo complejo, la creación de recursos puede estar distribuida en varios archivos, pero cualquier llamada a un módulo anidado debería estar en el `main.tf` archivo.
- variables.tf y outputs.tf: estos archivos contienen las declaraciones de las variables y las salidas. Todas las variables y salidas deben tener descripciones de una o dos oraciones que expliquen su propósito. Estas descripciones se utilizan como documentación. Para obtener más información, consulte la HashiCorp documentación sobre la configuración de variables y la configuración de salida.
  - Todas las variables deben tener un tipo definido.
  - La declaración de variables también puede incluir un argumento predeterminado. Si la declaración incluye un argumento predeterminado, la variable se considera opcional y se utiliza el valor predeterminado si no se establece un valor al llamar al módulo o ejecutar Terraform. El argumento predeterminado requiere un valor literal y no puede hacer referencia a otros objetos de la configuración. Para hacer que una variable sea obligatoria, omita un valor predeterminado en la declaración de la variable y considere si la configuración tiene `nullable = false` sentido.

  - Para las variables que tienen valores independientes del entorno (por ejemplo,`disk_size`), proporcione los valores predeterminados.
  - En el caso de las variables que tienen valores específicos del entorno (por ejemplo,`project_id`), no proporciones valores predeterminados. En este caso, el módulo de llamada debe proporcionar valores significativos.
  - Utilice valores predeterminados vacíos para variables como cadenas o listas vacías solo cuando dejar la variable vacía sea una preferencia válida que las variables subyacentes APIs no rechacen.
  - Sea prudente en el uso de las variables. Parametrice los valores solo si deben variar para cada instancia o entorno. Cuando decida si desea exponer una variable, asegúrese de contar con un caso de uso concreto para cambiarla. Si solo hay una pequeña posibilidad de que se necesite una variable, no la expongas.
    - Añadir una variable con un valor predeterminado es compatible con versiones anteriores.
    - La eliminación de una variable no es compatible con versiones anteriores.
    - En los casos en los que un literal se reutilice en varios lugares, debe utilizar un valor local sin exponerlo como una variable.
  - No pases los resultados directamente a través de las variables de entrada, ya que al hacerlo evitarás que se agreguen correctamente a la gráfica de dependencias. Para garantizar que se creen dependencias implícitas, asegúrese de que las salidas hagan referencia a los atributos de los recursos. En lugar de hacer referencia directamente a una variable de entrada para una instancia, pasa el atributo.
- locals.tf: este archivo contiene valores locales que asignan un nombre a una expresión, por lo que un nombre se puede usar varias veces dentro de un módulo en lugar de repetir la expresión. Los valores locales son como las variables locales temporales de una función. Las expresiones de los valores locales no se limitan a constantes literales; también pueden hacer referencia a otros valores del módulo, como variables, atributos de recursos u otros valores locales, para combinarlos.
- providers.tf: este archivo contiene el bloque terraform y los bloques provider. `provider`Los consumidores de los módulos deben declarar los bloques únicamente en los módulos raíz.

  Si utilizas HCP Terraform, añade también un bloque de nube vacío. El `cloud` bloque debe configurarse completamente mediante variables de entorno y credenciales de variables de entorno como parte de una CI/CD canalización.

- versions.tf: este archivo contiene el bloque required_providers. Todos los módulos de Terraform deben declarar qué proveedores requieren para que Terraform pueda instalar y usar estos proveedores.
- data.tf: Para una configuración sencilla, coloque las fuentes de datos junto a los recursos que hacen referencia a ellas. Por ejemplo, si vas a buscar una imagen para usarla al lanzar una instancia, colócala junto a la instancia en lugar de recopilar los recursos de datos en su propio archivo. Si el número de fuentes de datos es demasiado grande, plantéate moverlos a un `data.tf` archivo dedicado.
- Archivos.tfvars: en el caso de los módulos raíz, puede proporcionar variables no sensibles mediante un archivo. `.tfvars` Para mantener la coherencia, asigne un nombre a los archivos de variables. `terraform.tfvars` Coloque los valores comunes en la raíz del repositorio y los valores específicos del entorno en la `envs/` carpeta.
- Módulos anidados: los módulos anidados deben estar en el subdirectorio. `modules/` Cualquier módulo anidado que tenga un `README.md` se considera utilizable por un usuario externo. Si `README.md` no existe, el módulo se considera únicamente para uso interno. Los módulos anidados deben usarse para dividir el comportamiento complejo en varios módulos pequeños que los usuarios puedan seleccionar y elegir cuidadosamente.

  Si el módulo raíz incluye llamadas a módulos anidados, estas llamadas deben utilizar rutas relativas, por ejemplo, `./modules/sample-module` para que Terraform las considere parte del mismo repositorio o paquete en lugar de volver a descargarlas por separado.

  Si un repositorio o paquete contiene varios módulos anidados, lo ideal sería que la persona que los llamara pudiera componerlos en lugar de llamarse directamente entre sí y crear un árbol de módulos muy anidado.
- Ejemplos: deberían existir ejemplos del uso de un módulo reutilizable en el `examples/` subdirectorio de la raíz del repositorio. Para cada ejemplo, puedes añadir un archivo README para explicar el objetivo y el uso del ejemplo. Los ejemplos de submódulos también deben colocarse en el directorio raíz`examples/`.

  Como los ejemplos suelen copiarse en otros repositorios para su personalización, los bloques de módulos deben tener su origen establecido en la dirección que utilizaría una persona externa, no en una ruta relativa.
- Archivos con nombres de servicio: los usuarios suelen querer separar los recursos de Terraform por servicio en varios archivos. Esta práctica debe desalentarse en la medida de lo posible y, en `main.tf` su lugar, deberían definirse los recursos. Sin embargo, si un conjunto de recursos (por

ejemplo, las funciones y políticas de IAM) supera las 150 líneas, es razonable dividirlo en sus propios archivos, por ejemplo. `iam.tf` De lo contrario, todo el código de recursos debería estar definido en. `main.tf`

- Secuencias de comandos personalizadas: utilice las secuencias de comandos solo cuando sea necesario. Terraform no tiene en cuenta ni administra el estado de los recursos que se crean mediante scripts. Utilice scripts personalizados solo cuando los recursos de Terraform no admitan el comportamiento deseado. Coloque los scripts personalizados invocados por Terraform en un `scripts/` directorio.
- Secuencias de comandos auxiliares: Organice las secuencias de comandos auxiliares a las que Terraform no haya llamado en un directorio. `helpers/` Documente los scripts auxiliares en el `README.md` archivo con explicaciones y ejemplos de invocaciones. Si los scripts auxiliares aceptan argumentos, proporcione la comprobación y el resultado de los argumentos. `--help`
- Archivos estáticos: los archivos estáticos a los que Terraform hace referencia pero que no ejecuta (como los scripts de inicio cargados en instancias de EC2) deben organizarse en un directorio. `files/` Coloca los documentos extensos en archivos externos, separados de su HCL. Haga referencia a ellos con la función file ().
- Plantillas: para los archivos que lee la función templatefile de Terraform, utilice la extensión de archivo. `.tftpl` Las plantillas se deben colocar en un directorio. `templates/`

Estructura del módulo raíz

Terraform siempre se ejecuta en el contexto de un único módulo raíz. Una configuración completa de Terraform consta de un módulo raíz y un árbol de módulos secundarios (que incluye los módulos a los que llama el módulo raíz, cualquier módulo llamado por esos módulos, etc.).

Ejemplo básico del diseño del módulo raíz de Terraform:

```
.
### data.tf
### envs
#   ### dev
#   #   ### terraform.tfvars
#   ### prod
#   #   ### terraform.tfvars
#   ### test
#       ### terraform.tfvars
### locals.tf
### main.tf
```

```
### outputs.tf
### providers.tf
### README.md
### terraform.tfvars
### variables.tf
### versions.tf
```

Estructura de módulo reutilizable

Los módulos reutilizables siguen los mismos conceptos que los módulos raíz. Para definir un módulo, cree un nuevo directorio para él y coloque los `.tf` archivos en su interior, del mismo modo que definiría un módulo raíz. Terraform puede cargar módulos desde rutas relativas locales o desde repositorios remotos. Si espera que un módulo se reutilice en muchas configuraciones, colóquelo en su propio repositorio de control de versiones. Es importante mantener el árbol de módulos relativamente plano para que sea más fácil reutilizar los módulos en diferentes combinaciones.

Ejemplo básico de diseño de módulos reutilizables de Terraform:

```
.
### data.tf
### examples
#   ### multi-az-new-vpc
#   #   ### data.tf
#   #   ### locals.tf
#   #   ### main.tf
#   #   ### outputs.tf
#   #   ### providers.tf
#   #   ### README.md
#   #   ### terraform.tfvars
#   #   ### variables.tf
#   #   ### versions.tf
#   #   ### vpc.tf
#   ### single-az-existing-vpc
#   #   ### data.tf
#   #   ### locals.tf
#   #   ### main.tf
#   #   ### outputs.tf
#   #   ### providers.tf
#   #   ### README.md
#   #   ### terraform.tfvars
#   #   ### variables.tf
#   #   ### versions.tf
```

```
### iam.tf
### locals.tf
### main.tf
### outputs.tf
### README.md
### variables.tf
### versions.tf
```

Estructura de modularidad

En principio, puede combinar cualquier recurso y otras construcciones en un módulo, pero el uso excesivo de módulos anidados y reutilizables puede hacer que la configuración general de Terraform sea más difícil de entender y mantener, así que utilice estos módulos con moderación.

Cuando tenga sentido, divida su configuración en módulos reutilizables que aumenten el nivel de abstracción. Para ello, describa un nuevo concepto de su arquitectura que se construya a partir de tipos de recursos.

Al modular la infraestructura en definiciones reutilizables, opte por conjuntos lógicos de recursos en lugar de componentes individuales o colecciones demasiado complejas.

No agrupe recursos individuales

No debes crear módulos que sean envoltorios delgados alrededor de otros tipos de recursos únicos. Si tienes problemas para encontrar un nombre para tu módulo que sea diferente del nombre del tipo de recurso principal que contiene, es probable que tu módulo no esté creando una nueva abstracción, sino que esté añadiendo complejidad innecesaria. En su lugar, usa el tipo de recurso directamente en el módulo de llamada.

Encapsula las relaciones lógicas

Agrupe conjuntos de recursos relacionados, como los fundamentos de las redes, los niveles de datos, los controles de seguridad y las aplicaciones. Un módulo reutilizable debe encapsular las partes de la infraestructura que funcionan juntas para habilitar una capacidad.

Mantenga la herencia estable

Cuando anide módulos en subdirectorios, evite profundizar más de uno o dos niveles. Las estructuras de herencia profundamente anidadas complican las configuraciones y la solución de problemas. Los módulos deben basarse en otros módulos, no construir túneles a través de ellos.

Al centrar los módulos en agrupaciones de recursos lógicos que representan patrones de arquitectura, los equipos pueden configurar rápidamente bases de infraestructura confiables. Equilibre la abstracción sin sobreingeniería ni simplificación excesivas.

Recursos de referencia en los productos

Para cada recurso que se defina en un módulo reutilizable, incluya al menos un resultado que haga referencia al recurso. Las variables y los resultados permiten inferir las dependencias entre los módulos y los recursos. Sin ningún resultado, los usuarios no pueden ordenar correctamente su módulo en relación con sus configuraciones de Terraform.

Los módulos bien estructurados que proporcionan la coherencia del entorno, las agrupaciones con objetivos específicos y las referencias de recursos exportadas permiten la colaboración a escala de Terraform en toda la organización. Los equipos pueden armar la infraestructura a partir de componentes reutilizables.

No configure los proveedores

Si bien los módulos compartidos heredan los proveedores de los módulos de llamada, los módulos no deberían configurar los ajustes del proveedor por sí mismos. Evite especificar bloques de configuración del proveedor en los módulos. Esta configuración solo debe declararse una vez a nivel mundial.

Declare los proveedores obligatorios

Si bien las configuraciones de los proveedores se comparten entre los módulos, los módulos compartidos también deben declarar sus propios requisitos de proveedor. Esta práctica permite a Terraform asegurarse de que haya una única versión del proveedor que sea compatible con todos los módulos de la configuración y especificar la dirección de origen que sirve como identificador global (independiente del módulo) del proveedor. Sin embargo, los requisitos del proveedor específicos del módulo no especifican ninguno de los ajustes de configuración que determinan a qué puntos finales remotos accederá el proveedor, como un. Región de AWS

Al declarar los requisitos de versión y evitar la configuración de proveedores codificada, los módulos proporcionan portabilidad y reutilización en todas las configuraciones de Terraform que utilizan proveedores compartidos.

En el caso de los módulos compartidos, defina las versiones de proveedor mínimas requeridas en un bloque required_providers. `versions.tf`

Para declarar que un módulo requiere una versión determinada del AWS proveedor, usa un `required_providers` bloque dentro de un bloque: `terraform`

```
terraform {
  required_version = ">= 1.0.0"

  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = ">= 4.0.0"
    }
  }
}
```

Si un módulo compartido solo admite una versión específica del AWS proveedor, utilice el operador de restricción pesimista (~>), que permite que solo se incremente el componente de la versión que se encuentra más a la derecha:

```
terraform {
  required_version = ">= 1.0.0"

  required_providers {
    aws = {
      source  = "hashicorp/aws"
      version = "~> 4.0"
    }
  }
}
```

En este ejemplo, `~> 4.0` permite la instalación de y, pero no. `4.57.1` `4.67.0` `5.0.0` Para obtener más información, consulte Sintaxis de restricciones de versión en la HashiCorp documentación.

Siga las convenciones de nomenclatura

Los nombres descriptivos y claros simplifican la comprensión de las relaciones entre los recursos del módulo y el propósito de los valores de configuración. La coherencia con las pautas de estilo mejora la legibilidad tanto para los usuarios como para los encargados del mantenimiento del módulo.

Siga las pautas para la denominación de los recursos

- Usa snake_case (donde los términos en minúscula están separados por guiones bajos) en todos los nombres de los recursos para que coincidan con los estándares de estilo de Terraform. Esta práctica garantiza la coherencia con la convención de nomenclatura para los tipos de recursos, los tipos de fuentes de datos y otros valores predefinidos. Esta convención no se aplica a los argumentos de los nombres.
- Para simplificar las referencias a un recurso que es el único de su tipo (por ejemplo, un único balanceador de carga para todo un módulo), asigne un nombre al recurso `main` o `this` para mayor claridad.
- Usa nombres significativos que describan el propósito y el contexto del recurso y que ayuden a diferenciar recursos similares (por ejemplo, para la base de datos principal y `primary` `read_replica` para una réplica de lectura de la base de datos).
- Use nombres en singular, no en plural.
- No repitas el tipo de recurso en el nombre del recurso.

Siga las pautas para la denominación de las variables

- Agregue unidades a los nombres de las entradas, variables locales y salidas que representen valores numéricos como el tamaño del disco o el tamaño de la RAM (por ejemplo, `ram_size_gb` para el tamaño de la RAM en gigabytes). Esta práctica aclara la unidad de entrada esperada para los responsables del mantenimiento de la configuración.
- Utilice unidades binarias como MiB y GiB para los tamaños de almacenamiento y unidades decimales como MB o GB para otras métricas.
- Asigne a las variables booleanas nombres positivos, como. `enable_external_access`

Utilice los recursos de archivos adjuntos

Algunos recursos tienen pseudorecursos integrados como atributos. Siempre que sea posible, debes evitar el uso de estos atributos de recursos integrados y, en su lugar, usar el recurso único para adjuntar ese pseudorecurso. Estas relaciones de recursos pueden provocar cause-and-effect problemas exclusivos para cada recurso.

Uso de un atributo incrustado (evite este patrón):

```
resource "aws_security_group" "allow_tls" {
  ...
  ingress {
    description      = "TLS from VPC"
    from_port        = 443
    to_port          = 443
    protocol         = "tcp"
    cidr_blocks      = [aws_vpc.main.cidr_block]
    ipv6_cidr_blocks = [aws_vpc.main.ipv6_cidr_block]
  }

  egress {
    from_port        = 0
    to_port          = 0
    protocol         = "-1"
    cidr_blocks      = ["0.0.0.0/0"]
    ipv6_cidr_blocks = ["::/0"]
  }
}
```

Uso de recursos de archivos adjuntos (preferido):

```
resource "aws_security_group" "allow_tls" {
  ...
}

resource "aws_security_group_rule" "example" {
  type              = "ingress"
  description      = "TLS from VPC"
  from_port        = 443
  to_port          = 443
  protocol         = "tcp"
  cidr_blocks      = [aws_vpc.main.cidr_block]
  ipv6_cidr_blocks = [aws_vpc.main.ipv6_cidr_block]
  security_group_id = aws_security_group.allow_tls.id
}
```

Usa etiquetas predeterminadas

Asigne etiquetas a todos los recursos que puedan aceptarlas. El AWS proveedor de Terraform tiene una fuente de datos aws_default_tags que debe usar dentro del módulo raíz.

Considere agregar las etiquetas necesarias a todos los recursos creados por un módulo de Terraform. Esta es una lista de posibles etiquetas para adjuntar:

- Nombre: nombre del recurso legible por humanos
- AppId: el ID de la aplicación que usa el recurso
- AppRole: la función técnica del recurso; por ejemplo, «servidor web» o «base de datos»
- AppPurpose: El propósito comercial del recurso; por ejemplo, «interfaz de usuario» o «procesador de pagos»
- Entorno: el entorno de software, como desarrollo, prueba o producción
- Proyecto: los proyectos que utilizan el recurso
- CostCenter: ¿A quién facturar por el uso de los recursos

Cumple con los requisitos de registro de Terraform

Un repositorio de módulos debe cumplir con todos los requisitos siguientes para poder publicarse en un registro de Terraform.

Siempre debe seguir estos requisitos, incluso si no planea publicar el módulo en un registro a corto plazo. De este modo, puede publicar el módulo en un registro más adelante sin tener que cambiar la configuración y la estructura del repositorio.

- Nombre del repositorio: en el caso de un repositorio de módulos, utilice el nombre dividido en tres partes`terraform-aws-<NAME>`, donde se `<NAME>` refleje el tipo de infraestructura que administra el módulo. El `<NAME>` segmento puede contener guiones adicionales (por ejemplo,). `terraform-aws-iam-terraform-roles`
- Estructura de módulos estándar: el módulo debe cumplir con la estructura de repositorio estándar. Esto permite que el registro inspeccione el módulo y genere documentación, realice un seguimiento del uso de los recursos y mucho más.
  - Tras crear el repositorio de Git, copia los archivos del módulo en la raíz del repositorio. Te recomendamos que coloques cada módulo que vaya a ser reutilizable en la raíz de su propio repositorio, pero también puedes hacer referencia a los módulos de los subdirectorios.
  - Si utilizas HCP Terraform, publica los módulos que quieres compartir en el registro de tu organización. El registro gestiona las descargas y controla el acceso con los tokens de la API de HCP Terraform, por lo que los consumidores no necesitan acceder al repositorio fuente del módulo incluso cuando ejecutan Terraform desde la línea de comandos.

- Ubicación y permisos: el repositorio debe estar en uno de sus proveedores de sistemas de control de versiones (VCS) configurados y la cuenta de usuario del VCS de HCP Terraform debe tener acceso de administrador al repositorio. El registro necesita acceso de administrador para crear los webhooks e importar nuevas versiones de los módulos.
- Etiquetas x.y.z para las versiones: debe haber al menos una etiqueta de publicación para poder publicar un módulo. El registro utiliza etiquetas de publicación para identificar las versiones de los módulos. Los nombres de las etiquetas de publicación deben utilizar el control de versiones semántico, al que, si lo desea, puede añadir como prefijo un `v` (por ejemplo, `v1.1.0` y). `1.1.0` El registro ignora las etiquetas que no se parecen a los números de versión. Para obtener más información sobre la publicación de módulos, consulte la documentación de Terraform.

Para obtener más información, consulte Preparación de un repositorio de módulos en la documentación de Terraform.

Utilice las fuentes de módulos recomendadas

Terraform usa el `source` argumento de un bloque de módulos para buscar y descargar el código fuente de un módulo secundario.

Le recomendamos que utilice rutas locales para los módulos estrechamente relacionados que tengan como objetivo principal descartar los elementos de código repetidos, y que utilice un registro de módulos nativo de Terraform o un proveedor de VCS para los módulos que estén destinados a ser compartidos por varias configuraciones.

Los siguientes ejemplos ilustran los tipos de fuentes más comunes y recomendados para compartir módulos. Los módulos de registro admiten el control de versiones. Siempre debe proporcionar una versión específica, como se muestra en los siguientes ejemplos.

Registro

Registro de Terraform:

```
module "lambda" {
  source = "github.com/terraform-aws-modules/terraform-aws-lambda.git?
ref=e78cdf1f82944897ca6e30d6489f43cf24539374" #--> v4.18.0

  ...
```

```
}
```

Al fijar los hashes de confirmación, puede evitar desviarse de los registros públicos, que son vulnerables a los ataques a la cadena de suministro.

HCP Terraform:

```
module "eks_karpenter" {
  source = "app.terraform.io/my-org/eks/aws"
  version = "1.1.0"

  ...

  enable_karpenter = true
}
```

Terraform Enterprise:

```
module "eks_karpenter" {
  source = "terraform.mydomain.com/my-org/eks/aws"
  version = "1.1.0"

  ...

  enable_karpenter = true
}
```

Proveedores de VCS

Los proveedores de VCS apoyan el `ref` argumento a favor de seleccionar una revisión específica, como se muestra en los siguientes ejemplos.

GitHub (HTTPS):

```
module "eks_karpenter" {
  source = "github.com/my-org/terraform-aws-eks.git?ref=v1.1.0"

  ...

  enable_karpenter = true
}
```

Repositorio Git genérico (HTTPS):

```
module "eks_karpenter" {
  source = "git::https://example.com/terraform-aws-eks.git?ref=v1.1.0"

  ...

  enable_karpenter = true
}
```

Repositorio Git genérico (SSH):

> **Warning**
>
> Debes configurar las credenciales para acceder a los repositorios privados.

```
module "eks_karpenter" {
  source = "git::ssh://username@example.com/terraform-aws-eks.git?ref=v1.1.0"

  ...

  enable_karpenter = true
}
```

Siga los estándares de codificación

Aplique reglas y estilos de formato de Terraform consistentes en todos los archivos de configuración. Haga cumplir los estándares mediante el uso de comprobaciones de estilo automatizadas en las CI/CD canalizaciones. Cuando incorporas las mejores prácticas de codificación en los flujos de trabajo de los equipos, las configuraciones siguen siendo legibles, fáciles de mantener y colaborativas, ya que su uso se extiende por toda la organización.

Sigue las pautas de estilo

- Formatee todos los archivos de Terraform (`.tf`archivos) con el comando terraform fmt para que coincidan HashiCorp con los estándares de estilo.
- Utilice el comando terraform validate para verificar la sintaxis y la estructura de su configuración.

- Analice estáticamente la calidad del código mediante. TFLint Este linter comprueba las mejores prácticas de Terraform más allá del simple formateo y falla en las compilaciones cuando encuentra errores.

Configura enlaces previos a la confirmación

Configura enlaces de preconfirmación del lado del cliente que se ejecuten `terraform fmt` `tflintcheckov`, y otros escaneos de código y comprobaciones de estilo antes de permitir las confirmaciones. Esta práctica te ayuda a validar el cumplimiento de los estándares de forma más temprana en los flujos de trabajo de los desarrolladores.

Utilice marcos de preconfirmación, como la confirmación previa, para añadir el linting, el formateo y el escaneo de código de Terraform a modo de gancho en su máquina local. Los Hooks se ejecutan en cada confirmación de Git y fallan en la confirmación si las comprobaciones no se aprueban.

Al trasladar los controles de estilo y calidad a los enlaces locales previos a la confirmación, los desarrolladores reciben información rápida antes de introducir los cambios. Los estándares pasan a formar parte del flujo de trabajo de codificación.

Mejores prácticas para la administración AWS de versiones de Provider

Administrar cuidadosamente las versiones del AWS proveedor y los módulos de Terraform asociados es fundamental para la estabilidad. En esta sección se describen las mejores prácticas en relación con las restricciones y actualizaciones de las versiones.

Mejores prácticas:

- Añada comprobaciones de versiones automatizadas
- Supervise las nuevas versiones
- Contribuya a los proveedores

Añada comprobaciones de versiones automatizadas

Añada comprobaciones de versiones para los proveedores de Terraform a sus procesos de CI/CD para validar la fijación de versiones y, si la versión no está definida, no podrá compilar correctamente.

- Añade comprobaciones de TFlint en las canalizaciones de CI/CD para buscar versiones de proveedores que no tengan definidas las restricciones de versiones principales o secundarias fijas. Utilice el complemento de conjunto de reglas TFlint para Terraform AWS Provider, que proporciona reglas para detectar posibles errores y comprueba las mejores prácticas en relación con los recursos. AWS
- Falla las ejecuciones de CI que detectan versiones de proveedores no fijadas para evitar que las actualizaciones implícitas lleguen a producción.

Supervise las nuevas versiones

- Supervise las notas de las versiones de los proveedores y los feeds de los registros de cambios. Recibe notificaciones sobre las nuevas versiones principales o secundarias.
- Evalúe las notas de la versión para detectar posibles cambios importantes y evalúe su impacto en la infraestructura existente.
- Actualice primero las versiones secundarias en entornos que no sean de producción para validarlas antes de actualizar el entorno de producción.

Al automatizar las comprobaciones de versiones en los procesos y supervisar las nuevas versiones, puede detectar las actualizaciones no compatibles con antelación y dar tiempo a sus equipos para evaluar el impacto de las nuevas versiones principales o secundarias antes de actualizar los entornos de producción.

Contribuya a los proveedores

Contribuya activamente al HashiCorp AWS proveedor informando sobre los defectos o solicitando características en caso de GitHub problemas:

- Abre las publicaciones bien documentadas en el repositorio AWS de Provider para detallar cualquier error que hayas encontrado o alguna funcionalidad que falte. Proporcione pasos reproducibles.
- Solicite y vote sobre las mejoras para ampliar las capacidades del AWS proveedor para administrar nuevos servicios.
- Haga referencia a las solicitudes de cambios emitidas cuando aporte propuestas para corregir los defectos o mejoras del proveedor. Enlace a temas relacionados.
- Siga las pautas de contribución del repositorio para conocer las convenciones de codificación, los estándares de prueba y la documentación.

Al retribuir a los proveedores que utilizas, puedes contribuir directamente a su hoja de ruta y ayudar a mejorar su calidad y sus capacidades para todos los usuarios.

Mejores prácticas para los módulos comunitarios

El uso eficaz de los módulos es clave para gestionar las configuraciones complejas de Terraform y promover la reutilización. Esta sección proporciona las mejores prácticas en torno a los módulos, las dependencias, las fuentes, la abstracción y las contribuciones de la comunidad.

Mejores prácticas:

- Descubra los módulos comunitarios
- Comprenda las dependencias
- Utilice fuentes confiables
- Contribuya a los módulos comunitarios

Descubra los módulos comunitarios

Busque en el registro de Terraform y en otras fuentes los AWS módulos existentes que puedan resolver su caso de uso antes de crear un módulo nuevo. GitHub Busca opciones populares que tengan actualizaciones recientes y que se mantengan activamente.

Usa variables para la personalización

Cuando utilice módulos comunitarios, transfiera las entradas a través de las variables en lugar de bifurcar o modificar directamente el código fuente. Anule los valores predeterminados cuando sea necesario en lugar de cambiar las partes internas del módulo.

La bifurcación debe limitarse a aportar correcciones o funciones al módulo original para beneficiar a la comunidad en general.

Comprenda las dependencias

Antes de usar el módulo, revise su código fuente y su documentación para identificar las dependencias:

- Proveedores obligatorios: anote las versiones de AWS Kubernetes u otros proveedores que requiere el módulo.
- Módulos anidados: compruebe si hay otros módulos utilizados internamente que introduzcan dependencias en cascada.

- Fuentes de datos externas: anote las API, los complementos personalizados o las dependencias de infraestructura en las que se basa el módulo.

Al trazar el árbol completo de dependencias directas e indirectas, puede evitar sorpresas al utilizar el módulo.

Utilice fuentes confiables

Adquirir módulos de Terraform de editores desconocidos o no verificados implica un riesgo significativo. Utilice módulos únicamente de fuentes confiables.

- Prefiera los módulos certificados del Registro de Terraform publicados por creadores AWS o HashiCorp socios verificados.
- Para los módulos personalizados, revisa el historial del editor, los niveles de soporte y la reputación de uso, incluso si el módulo pertenece a tu propia organización.

Al no permitir módulos de fuentes desconocidas o no examinadas, puede reducir el riesgo de introducir vulnerabilidades o problemas de mantenimiento en su código.

Suscribirse a las notificaciones de

Suscríbase a las notificaciones de nuevos lanzamientos de módulos de editores de confianza:

- Mire los repositorios de GitHub módulos para recibir alertas sobre las nuevas versiones del módulo.
- Supervise los blogs y registros de cambios de los editores para ver si hay actualizaciones.
- Recibe notificaciones proactivas sobre las nuevas versiones de fuentes verificadas y altamente valoradas, en lugar de incluir las actualizaciones de forma implícita.

Consumir módulos únicamente de fuentes confiables y monitorear los cambios proporcionan estabilidad y seguridad. Los módulos aprobados mejoran la productividad y minimizan el riesgo de la cadena de suministro.

Contribuya a los módulos comunitarios

Envíe correcciones y mejoras para los módulos de la comunidad que están alojados en GitHub:

- Abre solicitudes de cambios en los módulos para corregir los defectos o limitaciones que encuentres en su uso.
- Solicita nuevas configuraciones recomendadas para añadirlas a los módulos OSS existentes creando problemas.

La contribución a los módulos comunitarios mejora los patrones codificados y reutilizables para todos los profesionales de Terraform.

Preguntas frecuentes

P: ¿Por qué centrarse en el AWS proveedor?

R. El AWS proveedor es uno de los proveedores más utilizados y complejos para el aprovisionamiento de infraestructura en Terraform. Seguir estas mejores prácticas ayuda a los usuarios a optimizar su uso del proveedor para el AWS entorno.

P: Soy nuevo en Terraform. ¿Puedo usar esta guía?

R: La guía es para personas que recién comienzan a usar Terraform, así como para profesionales más avanzados que desean mejorar sus habilidades. Las prácticas mejoran los flujos de trabajo de los usuarios en cualquier etapa del aprendizaje.

P: ¿Cuáles son algunas de las mejores prácticas clave que se incluyen?

R. Entre las mejores prácticas clave se incluyen el uso de funciones de IAM en lugar de las claves de acceso, la fijación de versiones, la incorporación de pruebas automatizadas, el bloqueo remoto por estado, la rotación de credenciales, la contribución a los proveedores y la organización lógica de las bases de códigos.

P: ¿Dónde puedo obtener más información sobre Terraform?

R. La sección de Recursos incluye enlaces a la documentación oficial de HashiCorp Terraform y a los foros de la comunidad. Utilice los enlaces para obtener más información sobre los flujos de trabajo avanzados de Terraform.

Siguientes pasos

Estos son algunos posibles pasos a seguir después de leer esta guía:

- Si ya tiene una base de código de Terraform, revise su configuración e identifique las áreas que podrían mejorarse según las recomendaciones que se proporcionan en esta guía. Por ejemplo, revise las mejores prácticas para implementar backends remotos, separar el código en módulos, utilizar la fijación de versiones, etc., y valídelas en su configuración.
- Si no tiene una base de código de Terraform existente, utilice estas mejores prácticas al estructurar su nueva configuración. Siga los consejos sobre la administración del estado, la autenticación, la estructura del código, etc. desde el principio.
- Intente utilizar algunos de los módulos HashiCorp comunitarios a los que se hace referencia en esta guía para comprobar si simplifican sus patrones de arquitectura. Los módulos permiten niveles más altos de abstracción, por lo que no es necesario reescribir los recursos comunes.
- Habilite la creación de perfiles, los escaneos de seguridad, las comprobaciones de políticas y las herramientas de pruebas automatizadas para reforzar algunas de las mejores prácticas en materia de seguridad, cumplimiento y calidad del código. Herramientas como TFlint, tfsec y Checkov pueden ayudar.
- Consulte la documentación más reciente AWS de Provider para ver si hay nuevos recursos o funciones que puedan ayudarle a optimizar su uso de Terraform. Manténgase actualizado sobre las nuevas versiones del AWS proveedor.
- Para obtener orientación adicional, consulte la documentación, la guía de mejores prácticas y la guía de estilo de Terraform en el HashiCorp sitio web.

Recursos

Referencias

Los siguientes enlaces proporcionan material de lectura adicional sobre el AWS proveedor de Terraform y sobre el uso de Terraform para iAC en adelante. AWS

- Terraform Provider AWS (documentación) HashiCorp
- Módulos de Terraform para AWS servicios (Terraform Registry)
- The AWS and HashiCorp Partnership (HashiCorp entrada de blog)
- Credenciales dinámicas con el AWS proveedor (documentación de HCP Terraform)
- Bloqueo de estado de DynamoDB (documentación de Terraform)
- Haga cumplir la política con Sentinel (documentación de Terraform)

Herramientas

Las siguientes herramientas ayudan a mejorar la calidad del código y la automatización de las configuraciones de Terraform AWS, tal como se recomienda en esta guía de mejores prácticas.

Calidad del código:

- Checkov: escanea el código de Terraform para identificar errores de configuración antes del despliegue.
- TFlint: identifica los posibles errores, la sintaxis obsoleta y las declaraciones no utilizadas. Este linter también puede aplicar las AWS mejores prácticas y las convenciones de nomenclatura.
- terraform-docs: genera documentación a partir de los módulos de Terraform en varios formatos de salida.

Herramientas de automatización:

- HCP Terraform: ayuda a los equipos a versionar, colaborar y crear flujos de trabajo de Terraform con controles de políticas y plazos de aprobación.
- Atlantis: una herramienta de automatización de solicitudes de extracción de Terraform de código abierto para validar los cambios de código.

- CDK para Terraform: un marco que le permite usar lenguajes conocidos como TypeScript Python, Java, C# y Go en lugar del lenguaje de HashiCorp configuración (HCL) para definir, aprovisionar y probar su infraestructura de Terraform como código.

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
Actualización del bloqueo de estado remoto	Se actualizó la sección de mejores prácticas de backend para reflejar el bloqueo por estado nativo de Amazon S3, que ahora es el enfoque recomendado.	26 de agosto de 2025
Publicación inicial	—	28 de mayo de 2024

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.