La guía de AWS CDK capas

AWS Guía prescriptiva

AWS Guía prescriptiva: La guía de AWS CDK capas

Table of Contents

Guía de las capas AWS CDK

Steven Guggenheimer, Amazon Web Services (AWS)

Diciembre de 2023 (historial de documentos)

Uno de los conceptos principales detrás de AWS Cloud Development Kit (AWS CDK) se parece mucho al concepto de mantenerse en calor en un día frío. Ese concepto se llama estratificación. En un día frío se pone una camisa, una chaqueta y, a veces, una chaqueta aún más grande según del frío que haga. Luego, si entra y la calefacción está encendida, puede quitarse una o ambas capas de la chaqueta para que no tenga demasiado calor. AWS CDK utiliza capas para proporcionar distintos niveles de abstracción al utilizar componentes de la nube. La estratificación garantiza que nunca tenga que escribir demasiado código ni tener muy poco acceso a las propiedades de los recursos al implementar las pilas de infraestructura como código (IaC).

Si no utiliza AWS CDK, tiene que escribir las plantillas de AWS CloudFormation a mano. Es decir, utilizará solo una capa, lo que lo obligará a escribir mucho más código del que normalmente es necesario. Por otro lado, si AWS CDK tuviera que abstraer en CloudFormation todo lo que, por lo general, no es necesario escribir, no podría gestionar ningún caso extremo.

Para solucionar este problema, AWS CDK divide el aprovisionamiento de recursos en tres capas independientes y distintas:

- Capa 1 (la capa de CloudFormation): la capa más básica en la que el recurso de CloudFormation y el recurso de AWS CDK son casi idénticos.
- Capa 2 (la capa seleccionada): la capa en la que los recursos de CloudFormation se abstraen en clases programáticas que simplifican gran parte de la sintaxis repetitiva de CloudFormation. Esta capa constituye la mayor parte de AWS CDK.
- Capa 3 (la capa de patrón): la capa más abstracta en la que puede utilizar los componentes básicos proporcionados por las capas 1 y 2 para personalizar el código para el caso de uso específico.

Cada elemento de cada capa es una instancia de una clase especial de AWS CDK llamada `Construct`. Según la documentación de AWS, los constructos son “los componentes básicos de las aplicaciones de AWS CDK. Un constructo representa un ‘componente de nube’ y encapsula todo lo que AWS CloudFormation necesita para crear el componente”. Los constructos en estas capas se conocen como constructos de la capa 1, capa 2 y capa 3 según la capa a la que pertenezcan. En

esta guía, revisaremos cada capa de AWS CDK para averiguar para qué se utilizan y por qué son importantes.

Esta guía está dirigida a los gerentes técnicos, líderes y desarrolladores interesados en profundizar en los conceptos básicos que hacen que funcione AWS CDK. AWS CDK es una herramienta popular, pero es muy común que los equipos no utilicen gran parte de lo que ofrece. Cuando empiece a comprender los conceptos descritos en esta guía, podrá descubrir todo un mundo nuevo de posibilidades y optimizar los procesos de aprovisionamiento de los recursos de sus equipos.

En esta guía:

Constructos de la capa 1

Los constructos de la capa 1 son los componentes básicos de AWS CDK y se distinguen de manera fácil de otros constructos por el prefijo `Cfn`. Por ejemplo, el paquete de Amazon DynamoDB en AWS CDK contiene un constructo `Table`, que es un constructo de la capa 2. El constructo correspondiente de la capa 1 se llama `CfnTable` y representa de manera directa un constructo `Table` de DynamoDB de CloudFormation Es imposible utilizar AWS CDK sin acceder a esta primera capa, aunque una aplicación de AWS CDK, en general, nunca utiliza un constructo de la capa 1 de manera directa. Sin embargo, en la mayoría de los casos, los constructos de la capa 2 y 3 a los que los desarrolladores están acostumbrados a utilizar dependen en gran medida de los constructos de la capa 1. Por lo tanto, puede pensar en los constructos de la capa 1 como el puente entre CloudFormation y AWS CDK.

La finalidad única de AWS CDK es generar plantillas de CloudFormation mediante el uso de lenguajes de codificación estándar. Tras ejecutar el comando de la CLI cdk synth y generar las plantillas de CloudFormation resultantes, el trabajo de AWS CDK estará completo. El comando cdk deploy está ahí solo por practicidad, pero lo que hace al ejecutar ese comando sucede de principio a fin en CloudFormation. La pieza del rompecabezas que traduce el código AWS CDK al formato que entiende CloudFormation es el constructo de la capa 1.

AWS CDK: el ciclo de vida de CloudFormation para los constructos de la capa 1

El proceso de creación y uso de los constructos de la capa 1 consta de los pasos siguientes:

1. El proceso de creación de AWS CDK convierte las especificaciones de CloudFormation en código programático en forma de constructos de la capa 1.
2. Los desarrolladores escriben código que hace referencia directa o indirectamente a los constructos de la capa 1 como parte de una aplicación. de AWS CDK.
3. Los desarrolladores ejecutan el comando cdk synth para convertir el código programático al formato dictado por las especificaciones (plantillas) de CloudFormation.
4. Los desarrolladores ejecutan el comando cdk deploy para implementar las pilas de CloudFormation que se incluyen en estas plantillas en los entornos de las cuentas de AWS.

Hagamos un ejercicio pequeño. Vaya al repositorio de código abierto de AWS CDK en GitHub, elija un producto de AWS al azar y, luego, vaya al paquete de AWS CDK de ese servicio (ubicado en la carpeta `packages`, `aws-cdk-lib`, `aws-<servicename>`, `lib`). Para este ejemplo, vamos a utilizar Amazon S3, pero funciona para cualquier servicio. Si mira el archivo index.ts principal de ese paquete, verá una línea que dice:

```
export * from './s3.generated';
```

Sin embargo, no verá el archivo `s3.generated` en ningún lugar del directorio correspondiente. Esto se debe a que los constructos de la capa 1 se generan de manera automática a partir de la especificación de recursos de CloudFormation durante el proceso de creación de AWS CDK. Verá `s3.generated` en el paquete solo después de ejecutar el comando de creación de AWS CDK del paquete.

Especificación de recursos de AWS CloudFormation

La especificación del recurso AWS CloudFormation define la infraestructura como código (IaC) de AWS y determina cómo el código de las plantillas de CloudFormation se convierte en recursos en una cuenta de AWS. Esta especificación define los recursos de AWS en formato JSON por región. A cada recurso se le asigna un nombre de tipo de recurso único que sigue el formato `provider::service::resource`. Por ejemplo, el nombre del tipo de recurso de un bucket de Amazon S3 sería `AWS::S3::Bucket` y el nombre del tipo de recurso de un punto de acceso de Amazon S3 sería `AWS::S3::AccessPoint`. Estos tipos de recursos se pueden representar en una plantilla de CloudFormation mediante la sintaxis definida en la especificación del recurso AWS CloudFormation. Cuando se ejecuta el proceso de compilación del AWS CDK, cada tipo de recurso también se convierte en un constructo de la capa 1.

En consecuencia, cada constructo de la capa 1 es una imagen reflejada programática de su recurso de CloudFormation correspondiente. Todas las propiedades que aplicaría en una plantilla de CloudFormation están disponibles al crear una instancia de un constructo de la capa 1, y todas las propiedades de CloudFormation necesarias también son necesarias como argumento al crear una instancia del constructo de la capa 1 correspondiente. En la siguiente tabla, se compara un bucket de S3 representado en una plantilla de CloudFormation con el mismo bucket de S3 definido como constructo de la capa 1 de AWS CDK.

Plantilla de CloudFormation	Constructo de la capa 1

```
"amzns3demobucket": {
    "Type": "AWS::S3::Bucket",
    "Properties": {
      "BucketName": "amzn-s3-demo-
bucket",
      "BucketEncryption": {
        "ServerSideEncryptionConfig
uration": [
          {
            "ServerSideEncrypt
ionByDefault": {
              "SSEAlgorithm": "AES256"
            }
          }
        ]
      },
      "MetricsConfigurations": [
        {
          "Id": "myConfig"
        }
      ],
      "OwnershipControls": {
        "Rules": [
          {
            "ObjectOwnership":
 "BucketOwnerPreferred"
          }
        ]
      },
      "PublicAccessBlockConfigura
tion": {
        "BlockPublicAcls": true,
        "BlockPublicPolicy": true,
        "IgnorePublicAcls": true,
        "RestrictPublicBuckets": true
      },
      "VersioningConfiguration": {
        "Status": "Enabled"
      }
    }
  }
```

```
new CfnBucket(this, "amzns3de
mobucket", {
  bucketName: "amzn-s3-demo-bucket",
  bucketEncryption: {
    serverSideEncryptionConfigu
ration: [
      {
        serverSideEncryptionByDefau
lt: {
          sseAlgorithm: "AES256"
        }
      }
    ]
  },
  metricsConfigurations: [
    {
      id: "myConfig"
    }
  ],
  ownershipControls: {
    rules: [
      {
        objectOwnership: "BucketOw
nerPreferred"
      }
    ]
  },
  publicAccessBlockConfiguration: {
    blockPublicAcls: true,
    blockPublicPolicy: true,
    ignorePublicAcls: true,
    restrictPublicBuckets: true
  },
  versioningConfiguration: {
    status: "Enabled"
  }
});
```

Como puede ver, el constructo de la capa 1 es el manifiesto exacto en el código del recurso de CloudFormation. No hay atajos ni simplificaciones, por lo que la cantidad de texto repetitivo que se debe escribir es casi la misma. Sin embargo, se supone que una de las grandes ventajas de usar AWS CDK es que ayuda a eliminar gran parte de esa sintaxis repetitiva de CloudFormation. ¿Y cómo sucede eso? Ahí es donde entra en juego el constructo de la capa 2.

Constructos de la capa 2

El repositorio de código abierto de AWS CDK se escribe sobre todo con el lenguaje de programación TypeScript y consta de numerosos paquetes y módulos. La biblioteca de paquetes principal, llamada `aws-cdk-lib`, se divide más o menos en un paquete por servicio de AWS, aunque no siempre es así. Como se mencionó anteriormente, los constructos de la capa 1 se generan de manera automática durante el proceso de compilación. Entonces, ¿qué es todo ese código que ve cuando revisa el repositorio? Se trata de constructos de la capa 2, que son abstracciones de los constructos de la capa 1.

Los paquetes también contienen una colección de tipos, enumeraciones e interfaces de TypeScript, así como clases auxiliares que agregan funcionalidad, pero todos esos elementos sirven para los constructos de la capa 2. Todos los constructos de la capa 2 llaman a sus constructos correspondientes de la capa 1 en sus constructores tras la creación de instancias. Se puede acceder al constructo de la capa 1 resultante que se crea desde la capa 2 de la manera siguiente:

```
const role = new Bucket(this, "amzn-s3-demo-bucket", {/*...BucketProps*/});
const cfnBucket = role.node.defaultChild;
```

El constructo de la capa 2 toma las propiedades predeterminadas, los métodos prácticos y otros elementos sintácticos y los aplica al constructo de la capa 1. Esto elimina gran parte de la repetición y la verbosidad necesarias para aprovisionar los recursos de manera directa en CloudFormation.

Todos los constructos de la capa 2 crean sus constructos correspondientes de la capa 1 de manera clandestina. Sin embargo, los constructos de la capa 2 en realidad no amplían los constructos de la capa 1. Los constructos de la capa 1 y de la capa 2 heredan una clase especial denominada Construct. En la versión 1 de AWS CDK, la clase `Construct` estaba integrada en el kit de desarrollo, pero en la versión 2, se trata un paquete independiente. Esto es para que otros paquetes, como Cloud Development Kit for Terraform (CDKTF), puedan incluirla como una dependencia. Cualquier clase que herede la clase `Construct` es un constructo de la capa 1, 2 y 3. Los constructos de la capa 2 amplían esta clase de manera directa, mientras que los constructos de la capa 1 amplían una clase llamada `CfnResource`, como se muestra en la tabla siguiente.

Árbol de herencia de la capa 1	Árbol de herencia de la capa 2
Constructo de la capa 1	Constructo de la capa 2
→ clase CfnResource	→ clase Construct

→→ clase abstracta CfnRefElement

→→→ clase abstracta CfnElement

→→→→ clase Construct

Si los constructos de la capa 1 y 2 heredan la clase `Construct`, ¿por qué los constructos de la capa 2 no solo amplían la capa 1? Las clases entre la clase `Construct` y la capa 1 bloquean el constructo de la capa 1 en su lugar como una imagen reflejada del recurso de CloudFormation. Contienen métodos abstractos (métodos que las clases posteriores deben incluir), como `_toCloudFormation`, que obligan al constructo a generar de manera directa la sintaxis de CloudFormation. Los constructos de la capa 2 omiten esas clases y amplían la clase `Construct` de manera directa. Esto les da la flexibilidad de abstraer gran parte del código necesario para los constructos de la capa 1 al crearlos por separado en sus constructores.

En la sección anterior se presentó una comparación paralela de un bucket de S3 de una plantilla de CloudFormation y ese mismo bucket de S3 renderizado como un constructo de la capa 1. En esa comparación se mostró que las propiedades y la sintaxis son casi idénticas y que el constructo de la capa 1 guarda solo tres o cuatro líneas en comparación con el constructo de CloudFormation. Ahora, comparemos el constructo de la capa 1 con el constructo de la capa 2 para el mismo bucket de S3:

Constructo de la capa 1 para el bucket de S3

```
new CfnBucket(this, "amzns3de
mobucket", {
  bucketName: "amzn-s3-demo-bucket",
  bucketEncryption: {
    serverSideEncryptionConfigu
ration: [
      {
        serverSideEncryptionByDefau
lt: {
          sseAlgorithm: "AES256"
        }
      }
    ]
  },
  metricsConfigurations: [
    {
```

Constructo de la capa 2 para el bucket de S3

```
new Bucket(this, "amzns3demobucket",
 {
  bucketName: "amzn-s3-demo-bucket",
  encryption: BucketEncryption.S
3_MANAGED,
  metrics: [
    {
      id: "myConfig"
    },
  ],
  objectOwnership: ObjectOwnership.BU
CKET_OWNER_PREFERRED,
  blockPublicAccess: BlockPubl
icAccess.BLOCK_ALL,
  versioned: true
});
```

```
      id: "myConfig"
    }
  ],
  ownershipControls: {
    rules: [
      {
        objectOwnership: "BucketOw
nerPreferred"
      }
    ]
  },
  publicAccessBlockConfiguration: {
    blockPublicAcls: true,
    blockPublicPolicy: true,
    ignorePublicAcls: true,
    restrictPublicBuckets: true
  },
  versioningConfiguration: {
    status: "Enabled"
  }
});
```

Como puede ver, el constructo de la capa 2 es menos de la mitad del tamaño del constructo de la capa 1. Los constructos de la capa 2 utilizan numerosas técnicas para lograr esta consolidación. Algunas de estas técnicas se aplican a un solo constructo de la capa 2, pero otras se pueden reutilizar en varios constructos para separarlos en su propia clase para su reutilización. Los constructos de la capa 2 consolidan la sintaxis de CloudFormation de varias maneras, como se explica en las secciones siguientes.

Propiedades predeterminadas

La manera más sencilla de consolidar el código para aprovisionar un recurso es convertir la configuración de propiedades más común en valores predeterminados. AWS CDK tiene acceso a lenguajes de programación potentes y CloudFormation no, por lo que estos valores predeterminados suelen ser de naturaleza condicional. A veces, se pueden eliminar varias líneas de configuración de CloudFormation del código de AWS CDK porque se pueden deducir de los valores de otras propiedades que se pasan al constructo.

Estructuras, tipos e interfaces

Aunque AWS CDK está disponible en varios lenguajes de programación, está escrito de forma nativa en TypeScript, por lo que el sistema de tipos de ese lenguaje se utiliza para definir los tipos que componen los constructos de la capa 2. Entrar en detalle sobre ese sistema de tipos supera el ámbito de esta guía. Consulte la documentación de TypeScript para más información. En resumen, `type` de TypeScript describe qué tipo de datos contiene una variable concreta. Pueden ser datos básicos, como `string`, o datos más complejos, como `object`. `interface` de TypeScript es otra manera de expresar el tipo de objeto de TypeScript y `struct` es otro nombre para una interfaz.

TypeScript no usa el término estructura, pero si consulta la referencia de la API del AWS CDK, verá que una estructura es en realidad solo otra interfaz de TypeScript en el código. La referencia de la API también se refiere a interfaces determinadas como “interfaces”. Si las estructuras y las interfaces son lo mismo, ¿por qué la documentación de AWS CDK hace una distinción entre estas?

Lo que AWS CDK denomina estructuras son interfaces que representan a los objetos que utiliza un constructo de la capa 2. Esto incluye los tipos de objeto de los argumentos de propiedad que se pasan al constructo de la capa 2 durante la creación de instancias, como `BucketProps` para el constructo de un bucket de S3 y `TableProps` para el constructo de la tabla de DynamoDB, así como otras interfaces de TypeScript que se utilizan en AWS CDK. En resumen, si se trata de una interfaz de TypeScript en AWS CDK y su nombre no tiene el prefijo de la letra `I`, AWS CDK la llama estructura.

Por el contrario, AWS CDK utiliza el término interfaz para representar los elementos básicos. Un objeto simple debería considerarse una representación adecuada de un constructo o clase auxiliar en particular. Es decir, una interfaz describe cuáles deben ser las propiedades públicas de un constructo de la capa 2. Todos los nombres de las interfaces de AWS CDK son los nombres de los constructos o clases auxiliares existentes con el prefijo de la letra `I`. Todos los constructos de la capa 2 amplían la clase `Construct`, pero también implementan su interfaz correspondiente. Por lo tanto, `Bucket` del constructo de la capa 2 implementa la interfaz de `IBucket`.

Métodos estáticos

Cada instancia de un constructo de la capa 2 también es una instancia de su interfaz correspondiente, pero no ocurre lo contrario. Esto es importante al momento de analizar una estructura para ver qué tipos de datos son necesarios. Si una estructura tiene una propiedad llamada `bucket`, que requiere el tipo de datos `IBucket`, puede pasar un objeto que contenga las

propiedades enumeradas en la interfaz de `IBucket` o una instancia de una propiedad `Bucket` de la capa 2. Ambas funcionarían. Sin embargo, si esa propiedad `bucket` llamará a una propiedad `Bucket` de la capa 2, solo podría pasar una instancia de `Bucket` en ese campo.

Esta distinción se vuelve muy importante cuando se importan recursos preexistentes a la pila. Puede crear un constructo de la capa 2 para los recursos que nativos de la pila, pero si tiene que hacer referencia a un recurso que se creó fuera de la pila, tiene que utilizar la interfaz de ese constructo de la capa 2. Esto se debe a que al crear un constructo de la capa 2 se crea un recurso nuevo si aún no existe uno en esa pila. Las referencias a los recursos existentes tienen que ser objetos simples que se ajusten a la interfaz de ese constructo de la capa 2.

Para facilitar esto en la práctica, la mayoría de los constructos de la capa 2 tienen un conjunto de métodos estáticos asociados que devuelven la interfaz de ese constructo de la capa 2. Estos métodos estáticos suelen empezar con la palabra `from`. Los dos primeros argumentos que se pasan a estos métodos son los mismos argumentos `scope` y `id` necesarios para un constructo estándar de la capa 2. Sin embargo, el tercer argumento no es `props`, sino un subconjunto pequeño de propiedades (o, a veces, solo una propiedad) que define una interfaz. Por este motivo, cuando se pasa un constructo de la capa 2, en la mayoría de los casos solo son necesarios los elementos de la interfaz. Esto permite utilizar también los recursos importados, siempre que sea posible.

```
// Example of referencing an external S3 bucket
const preExistingBucket = Bucket.fromBucketName(this, "external-bucket", "name-of-
bucket-that-already-exists");
```

Sin embargo, no debe confiar demasiado en las interfaces. Debe importar los recursos y utilizar las interfaces de manera directa solo cuando sea absolutamente necesario, ya que las interfaces no proporcionan muchas de las propiedades (como los métodos de ayuda) que hacen que un constructo de la capa 2 sea tan potente.

Métodos auxiliares

Un constructo de la capa 2 es una clase programática y no un objeto simple, por lo que puede exponer los métodos de clase que permiten manipular la configuración de los recursos una vez creada la instancia. Un buen ejemplo de ello es el constructo Role de la capa 2 de AWS Identity and Access Management (IAM). Los fragmentos siguientes muestran dos maneras de crear el mismo rol de IAM mediante el constructo `Role` de la capa 2.

Sin un método auxiliar:

```
const role = new Role(this, "my-iam-role", {
   assumedBy: new FederatedPrincipal('my-identity-provider.com'),
   managedPolicies: [
      ManagedPolicy.fromAwsManagedPolicyName("ReadOnlyAccess")
   ],
   inlinePolicies: {
      lambdaPolicy: new PolicyDocument({
         statements: [
            new PolicyStatement({
               effect: Effect.ALLOW,
               actions: [ 'lambda:UpdateFunctionCode' ],
               resources: [ 'arn:aws:lambda:us-east-1:123456789012:function:my-
function' ]
            })
         ]
      })
   }
});
```

Con un método auxiliar:

```
const role = new Role(this, "my-iam-role", {
   assumedBy: new FederatedPrincipal('my-identity-provider.com')
});

role.addManagedPolicy(ManagedPolicy.fromAwsManagedPolicyName("ReadOnlyAccess"));
role.attachInlinePolicy(new Policy(this, "lambda-policy", {
   policyName: "lambdaPolicy",
   statements: [
      new PolicyStatement({
         effect: Effect.ALLOW,
         actions: [ 'lambda:UpdateFunctionCode' ],
         resources: [ 'arn:aws:lambda:us-east-1:123456789012:function:my-function' ]
      })
   ]
 }));
```

La capacidad de utilizar los métodos de la instancia para manipular la configuración de los recursos después de la creación de instancias proporciona a los constructos de la capa 2 mucha flexibilidad adicional con respecto a la capa anterior. Los constructos de la capa 1 también heredan algunos métodos de los recursos (por ejemplo `addPropertyOverride`), pero no se obtienen métodos diseñados en específico para ese recurso y sus propiedades hasta la segunda capa.

Enums

La sintaxis de CloudFormation a menudo requiere que se especifiquen muchos detalles para aprovisionar un recurso de manera correcta. Sin embargo, la mayoría de los casos de uso suelen incluir solo unas pocas configuraciones. La representación de esas configuraciones mediante una serie de valores enumerados puede reducir de manera considerable la cantidad de código necesaria.

En el ejemplo de código de la capa 2 del bucket de S3 que aparece anteriormente en esta sección, debe utilizar la propiedad `bucketEncryption` de la plantilla de CloudFormation para proporcionar todos los detalles, tal como el nombre del algoritmo de cifrado que se va a utilizar. En su lugar, AWS CDK proporciona la enumeración `BucketEncryption`, que utiliza las cinco maneras más comunes de cifrado de buckets y permite expresarlas mediante nombres de variables individuales.

¿Qué sucede con los casos extremos que no se cubren mediante las enumeraciones? Uno de los objetivos de un constructo de la capa 2 es simplificar la tarea de aprovisionamiento de un recurso de la capa 1, por lo que es posible que algunos casos extremos que se utilizan con menos frecuencia no se admitan en la capa 2. Para admitir estos casos extremos, AWS CDK permite manipular las propiedades de los recursos subyacentes de CloudFormation de manera directa mediante el método addPropertyOverride. Para más información sobre las anulaciones de las propiedades, consulte la sección Prácticas recomendadas de esta guía y la sección Abstracciones y vías de escape de la documentación. AWS CDK

Clases auxiliares

En algunas ocasiones, una enumeración no puede cumplir con la lógica programática necesaria para configurar un recurso de un caso de uso determinado. En estas situaciones, AWS CDK a menudo ofrece una clase auxiliar en su lugar. Una enumeración es un objeto simple que ofrece una serie de pares clave-valor, mientras que una clase auxiliar ofrece todas las funcionalidades de una clase de TypeScript. Una clase auxiliar aún puede actuar como una enumeración al exponer las propiedades estáticas, pero esas propiedades podrían tener sus valores establecidos de manera interna con lógica condicional en el constructor de la clase auxiliar o en un método auxiliar.

Por lo tanto, aunque la enumeración `BucketEncryption` puede reducir la cantidad de código necesaria para configurar un algoritmo de cifrado en un bucket de S3, esa misma estrategia no funcionaría al establecer duraciones temporales, ya que hay demasiados valores posibles entre los que elegir. Crear una enumeración para cada valor sería mucho más complicado que lo que vale la pena. Por este motivo, se utiliza una clase auxiliar para los valores predeterminados de la configuración de Bloqueo de objetos de S3 de un bucket de S3, representados por la clase

ObjectLockRetention. `ObjectLockRetention` contiene dos métodos estáticos: uno para la retención del cumplimiento y otro para la retención de la gobernanza. Ambos métodos utilizan una instancia de la clase auxiliar Duration como argumento para expresar el tiempo durante el que se debe configurar el bloqueo.

Otro ejemplo es la clase auxiliar Runtime de AWS Lambda. A simple vista, podría parecer que las propiedades estáticas asociadas a esta clase podrían gestionarse mediante una enumeración. Sin embargo, en realidad, el valor de cada propiedad representa una instancia de la propia clase `Runtime`, por lo que la lógica que se utiliza en el constructor de la clase no podría lograrse en una enumeración.

Constructos de la capa 3

Si los constructos de la capa 1 hacen una traducción literal de los recursos de CloudFormation a código programático y los constructos de la capa 2 sustituyen gran parte de la sintaxis detallada de CloudFormation por métodos auxiliares y lógica personalizada, ¿qué hacen los constructos de la capa 3? Solo su imaginación limita la respuesta a esa pregunta. Puede crear la capa 3 para satisfacer cualquier caso de uso concreto. Si su proyecto necesita un recurso que tenga un subconjunto específico de propiedades, puede crear un constructo de la capa 3 reutilizable para satisfacer esa necesidad.

Los constructos de la capa 3 se denominan patrones en AWS CDK. Un patrón es cualquier objeto que amplía la clase `Construct` en AWS CDK (o amplía una clase que amplía la clase `Construct`) para hacer cualquier lógica abstracta más allá de la capa 2. Cuando utiliza la CLI de AWS CDK para ejecutar cdk init e iniciar un nuevo proyecto de AWS CDK, debe elegir entre tres tipos de aplicaciones de AWS CDK: `app`, `lib` y `sample-app`.

```
Available templates:
* app: Template for a CDK Application
   └─ cdk init app --language=[csharp|fsharp|go|java|javascript|python|typescript]
* lib: Template for a CDK Construct Library
   └─ cdk init lib --language=typescript
* sample-app: Example CDK Application with some constructs
   └─ cdk init sample-app --language=[csharp|fsharp|go|java|javascript|python|typescript]
```

`app` y `sample-app` representan las aplicaciones clásicas de AWS CDK en las que se crean e implementan pilas de CloudFormation en entornos de AWS. Al elegir `lib`, elige crear un constructo de la capa 3 completamente nuevo. `app` y `sample-app` le permiten elegir cualquier lenguaje que AWS CDK admita, pero solo puede elegir TypeScript con `lib`. Esto se debe a que AWS CDK está escrito de manera nativa en TypeScript y utiliza un sistema de código abierto llamado JSii para traducir el código original a los demás idiomas admitidos. Al elegir `lib` para iniciar el proyecto, elige crear una ampliación para AWS CDK.

Cualquier clase que amplíe la clase `Construct` puede ser un constructo de la capa 3, pero los casos de uso más comunes de la capa 3 son las interacciones de recursos, las ampliaciones de recursos y los recursos personalizados. La mayoría de los constructos de la capa 3 utilizan uno o varios de estos tres casos para ampliar la funcionalidad de AWS CDK.

Interacciones de los recursos

Por lo general, una solución utiliza varios servicios de AWS que funcionan juntos. Por ejemplo, una distribución de Amazon CloudFront suele utilizar un bucket de S3 como origen y AWS WAF como protección contra las vulnerabilidades de seguridad más comunes. AWS AppSync y Amazon API Gateway suelen utilizar tablas de Amazon DynamoDB como orígenes de datos para sus API. Una canalización de AWS CodePipeline suele utilizar Amazon S3 como origen y AWS CodeBuild para sus etapas de creación. En estos casos, suele ser útil crear un constructo único de la capa 3 que se encargue del aprovisionamiento de dos o varios constructos de la capa 2 interconectadas.

A continuación, se muestra un ejemplo de un constructo de la capa 3 que aprovisiona una distribución de CloudFront junto con su origen de S3, un AWS WAF para ponerlo delante, un registro de Amazon Route 53 y un certificado de AWS Certificate Manager (ACM) para agregar un punto de conexión personalizado con cifrado en tránsito, todo en un constructo reutilizable:

```
// Define the properties passed to the L3 construct
export interface CloudFrontWebsiteProps {
     distributionProps: DistributionProps
     bucketProps: BucketProps
     wafProps: CfnWebAclProps
     zone: IHostedZone
}

// Define the L3 construct
export class CloudFrontWebsite extends Construct {
     public distribution: Distribution

     constructor(
        scope: Construct,
        id: string,
        props: CloudFrontWebsiteProps
     ) {
         super(scope, id);

         const certificate = new Certificate(this, "Certificate", {
            domainName: props.zone.zoneName,
            validation: CertificateValidation.fromDns(props.zone)
         });
         const defaultBehavior = {
            origin: new S3Origin(new Bucket(this, "bucket", props.bucketProps))
         }
```

```
        const waf = new CfnWebACL(this, "waf", props.wafProps);
        this.distribution = new Distribution(this, id, {
           ...props.distributionProps,
           defaultBehavior,
           certificate,
           domainNames: [this.domainName],
           webAclId: waf.attrArn,
         });
    }
}
```

Observe que CloudFront, Amazon S3, Route 53 y ACM utilizan constructos de la capa 2, pero la ACL web (que define las reglas para gestionar las solicitudes web) utiliza un constructo de la capa 1. Esto se debe a que AWS CDK es un paquete de código abierto en evolución que no está completo del todo y aún no existe un constructo de la capa 2 para `WebAcl`. Sin embargo, cualquiera puede contribuir a AWS CDK al crear nuevos constructos de la capa 2. Por lo tanto, hasta que AWS CDK no ofrezca un constructo de la capa 2 para `WebAcl`, tendrá que utilizar un constructo de la capa 1. Para crear un nuevo sitio web mediante el constructo de la capa 3 `CloudFrontWebsite`, utilice el código siguiente:

```
const siteADotCom = new CloudFrontWebsite(stack, "siteA", siteAProps);
const siteBDotCom = new CloudFrontWebsite(stack, "siteB", siteBProps);
const siteCDotCom = new CloudFrontWebsite(stack, "siteC", siteCProps);
```

En este ejemplo, el constructo de la capa 2 `Distribution` de CloudFront se expone como una propiedad pública del constructo de la capa 3. Seguirán existiendo casos en los que deba exponer propiedades de la capa 3 como esta, según sea necesario. De hecho, volveremos a ver `Distribution` más adelante, en la sección Recursos personalizados.

AWS CDK incluye algunos ejemplos de patrones de interacción de recursos como este. Además del paquete de `aws-ecs` que contiene los constructos de la capa 2 para Amazon Elastic Container Service (Amazon ECS), AWS CDK tiene un paquete llamado aws-ecs-patterns. Este paquete contiene varios constructos de la capa 3 que combinan Amazon ECS con equilibradores de carga de aplicaciones, equilibradores de carga de red y grupos de destinos, además de que ofrece distintas versiones predefinidas para Amazon Elastic Compute Cloud (Amazon EC2) y AWS Fargate. Dado que muchas aplicaciones sin servidor utilizan Amazon ECS solo con Fargate, estos constructos de la capa 3 ofrecen una comodidad que permite a los desarrolladores ahorrar tiempo y dinero a los clientes.

Extensiones de recursos

Algunos casos de uso requieren que los recursos tengan una configuración predeterminada específica que no sea nativa del constructo de la capa 2. Para las pilas, esto se puede gestionar mediante los aspectos, pero otra manera práctica de dotar a un constructo nuevo de la capa 2 de nuevos valores predeterminados consiste en ampliar la capa 2. Como un constructo es cualquier clase que hereda la clase `Construct` y los constructos de la capa 2 amplían esa clase, también puede crear un constructo de la capa 3 mediante la ampliación directa de un constructo de la capa 2.

Esto puede resultar muy útil para la lógica empresarial personalizada que respalde las necesidades concretas de un cliente. Supongamos que una empresa tiene un repositorio que almacena todo su código de la función de AWS Lambda en un único directorio llamado `src/lambda` y que la mayoría de las funciones de Lambda reutilizan el mismo tiempo de ejecución y el mismo nombre de controlador cada vez. En lugar de configurar la ruta del código cada vez que configure una nueva función de Lambda, podría crear un constructo nuevo de la capa 3:

```
export class MyCompanyLambdaFunction extends Function {
      constructor(
        scope: Construct,
        id: string,
        props: Partial<FunctionProps> = {}
     ) {
         super(scope, id, {
            handler: 'index.handler',
            runtime: Runtime.NODEJS_LATEST,
            code: Code.fromAsset(`src/lambda/${props.functionName || id}`),
            ...props
     });
}
```

A continuación, puede reemplazar el constructo de la capa 2 `Function` en todas partes del repositorio de la manera siguiente:

```
new MyCompanyLambdaFunction(this, "MyFunction");
new MyCompanyLambdaFunction(this, "MyOtherFunction");
new MyCompanyLambdaFunction(this, "MyThirdFunction", {
   runtime: Runtime.PYTHON_3_11
});
```

Los valores predeterminados le permiten crear nuevas funciones de Lambda en una sola línea, y el constructo de la capa 3 está configurada para que pueda anular las propiedades predeterminadas si es necesario.

La extensión directa de los constructos de la capa 2 funciona mejor cuando solo se desean agregar nuevos valores predeterminados a los constructos de la capa 2 existentes. Si también necesita otra lógica personalizada, es mejor ampliar la clase `Construct`. El motivo de esto se debe al método `super`, al que se llama desde el constructor. En las clases que amplían otras clases, el método `super` se utiliza para llamar al constructor de la clase principal. Esto debe ser lo primero que suceda en el constructor. Esto significa que las manipulaciones de los argumentos pasados u otra lógica personalizada solo puede producirse después de que se haya creado el constructo de la capa 2 original. Si necesita hacer alguna de estas lógicas personalizadas antes de crear una instancia del constructo de la capa 2, es mejor seguir el patrón descrito anteriormente en la sección Interacciones entre recursos.

Recursos personalizados

Los recursos personalizados son una característica eficaz de CloudFormation que permite ejecutar una lógica personalizada desde una función de Lambda que se activa durante la implementación de la pila. Siempre que necesite algún proceso durante la implementación que no sea compatible directamente con CloudFormation, puede utilizar un recurso personalizado para lograrlo. AWS CDK ofrece clases que también le permiten crear recursos personalizados mediante programación. Al utilizar los recursos personalizados en un constructor de la capa 3, puede hacer un constructo a partir de casi cualquier cosa.

Una de las ventajas de utilizar Amazon CloudFront es su funcionalidad sólida de almacenamiento en caché global. Si desea restablecer de manera manual la caché para que el sitio web refleje de inmediato los cambios nuevos hechos en el origen, puede utilizar una invalidación de CloudFront. Sin embargo, las invalidaciones son procesos que se ejecutan en una distribución de CloudFront en lugar de ser propiedades de una distribución de CloudFront. Se pueden crear y aplicar a una distribución existente en cualquier momento, por lo que no forman parte de manera nativa del proceso de aprovisionamiento e implementación.

En este escenario, es posible que quiera crear y ejecutar una invalidación después de cada actualización del origen de una distribución. Debido a los recursos personalizados, puede crear un constructo de la capa 3 similar a la siguiente:

```
export interface CloudFrontInvalidationProps {
```

```
     distribution: Distribution
     region?: string
     paths?: string[]
}

export class CloudFrontInvalidation extends Construct {
      constructor(
        scope: Construct,
        id: string,
        props: CloudFrontInvalidationProps
     ) {
          super(scope, id);
          const policy = AwsCustomResourcePolicy.fromSdkCalls({
               resources:AwsCustomResourcePolicy.ANY_RESOURCE
          });
          new AwsCustomResource(scope, `${id}Invalidation`, {
             policy,
             onUpdate: {
                  service: 'CloudFront',
                  action: 'createInvalidation',
                  region: props.region || 'us-east-1',
                  physicalResourceId:
 PhysicalResourceId.fromResponse('Invalidation.Id'),
                  parameters: {
                     DistributionId: props.distribution.distributionId,
                     InvalidationBatch: {
                         Paths: {
                            Quantity: props.paths?.length || 1,
                            Items: props.paths || ['/*']
                         },
                         CallerReference: crypto.randomBytes(5).toString('hex')
                     }
                  }
             }
          }
     }
}
```

Con la distribución que se creó anteriormente en el constructo `CloudFrontWebsite` de la capa 3, podría hacerlo de manera muy fácil:

```
new CloudFrontInvalidation(this, 'MyInvalidation', {
       distribution: siteADotCom.distribution
```

```
});
```

Este constructo de la capa 3 utiliza un constructo AWS CDK de la capa 3 llamado AwsCustomResource para crear un recurso personalizado que ejecuta una lógica personalizada. `AwsCustomResource` es muy práctico cuando se debe hacer exactamente una llamada al SDK de AWS, ya que le permite hacerla sin tener que escribir ningún código de Lambda. Si tiene requisitos más complejos y desea implementar su propia lógica, puede utilizar de manera directa la clase básica CustomResource.

Otro buen ejemplo de AWS CDK en el que se utiliza un constructo de la capa 3 del recurso personalizado es la implementación de buckets de S3. La función de Lambda creada por el recurso personalizado del constructor de este constructo de la capa 3 agrega una funcionalidad que CloudFormation no podría gestionar. Agrega y actualiza objetos en un bucket de S3. Sin la implementación de un bucket de S3, no podría colocar contenido en el bucket que acaba de crear como parte de su pila, lo que sería muy inconveniente.

El mejor ejemplo de cómo AWS CDK elimina la necesidad de escribir montones de sintaxis de CloudFormation es este constructo `S3BucketDeployment` básico:

```
new BucketDeployment(this, 'BucketObjects', {
  sources: [Source.asset('./path/to/amzn-s3-demo-bucket')],
  destinationBucket: amzn-s3-demo-bucket
});
```

Compárelo con el código de CloudFormation que tendría que escribir para lograr lo mismo:

```
"lambdapolicyA5E98E09": {
   "Type": "AWS::IAM::Policy",
   "Properties": {
    "PolicyDocument": {
     "Statement": [
      {
       "Action": "lambda:UpdateFunctionCode",
       "Effect": "Allow",
       "Resource": "arn:aws:lambda:us-east-1:123456789012:function:my-function"
      }
     ],
     "Version": "2012-10-17"
    },
    "PolicyName": "lambdaPolicy",
    "Roles": [
```

```
      {
       "Ref": "myiamroleF09C7974"
      }
     ]
    },
    "Metadata": {
     "aws:cdk:path": "CdkScratchStack/lambda-policy/Resource"
    }
   },
   "BucketObjectsAwsCliLayer8C081206": {
    "Type": "AWS::Lambda::LayerVersion",
    "Properties": {
     "Content": {
      "S3Bucket": {
       "Fn::Sub": "cdk-hnb659fds-assets-${AWS::AccountId}-${AWS::Region}"
      },
      "S3Key": "e2277687077a2abf9ae1af1cc9565e6715e2ebb62f79ec53aa75a1af9298f642.zip"
     },
     "Description": "/opt/awscli/aws"
    },
    "Metadata": {
     "aws:cdk:path": "CdkScratchStack/BucketObjects/AwsCliLayer/Resource",
     "aws:asset:path":
"asset.e2277687077a2abf9ae1af1cc9565e6715e2ebb62f79ec53aa75a1af9298f642.zip",
     "aws:asset:is-bundled": false,
     "aws:asset:property": "Content"
    }
   },
   "BucketObjectsCustomResourceB12E6837": {
    "Type": "Custom::CDKBucketDeployment",
    "Properties": {
     "ServiceToken": {
      "Fn::GetAtt": [
       "CustomCDKBucketDeployment8693BB64968944B69AAFB0CC9EB8756C81C01536",
       "Arn"
      ]
     },
     "SourceBucketNames": [
      {
       "Fn::Sub": "cdk-hnb659fds-assets-${AWS::AccountId}-${AWS::Region}"
      }
     ],
     "SourceObjectKeys": [
      "f888a9d977f0b5bdbc04a1f8f07520ede6e00d4051b9a6a250860a1700924f26.zip"
```

```
    ],
    "DestinationBucketName": {
     "Ref": "amzn-s3-demo-bucket77F80CC0"
    },
    "Prune": true
   },
   "UpdateReplacePolicy": "Delete",
   "DeletionPolicy": "Delete",
   "Metadata": {
    "aws:cdk:path": "CdkScratchStack/BucketObjects/CustomResource/Default"
   }
  },
  "CustomCDKBucketDeployment8693BB64968944B69AAFB0CC9EB8756CServiceRole89A01265": {
   "Type": "AWS::IAM::Role",
   "Properties": {
    "AssumeRolePolicyDocument": {
     "Statement": [
      {
       "Action": "sts:AssumeRole",
       "Effect": "Allow",
       "Principal": {
        "Service": "lambda.amazonaws.com"
       }
      }
     ],
     "Version": "2012-10-17"
    },
    "ManagedPolicyArns": [
     {
      "Fn::Join": [
       "",
       [
        "arn:",
        {
         "Ref": "AWS::Partition"
        },
        ":iam::aws:policy/service-role/AWSLambdaBasicExecutionRole"
       ]
      ]
     }
    ]
   },
   "Metadata": {
```

```
    "aws:cdk:path": "CdkScratchStack/
Custom::CDKBucketDeployment8693BB64968944B69AAFB0CC9EB8756C/ServiceRole/Resource"
   }
  },

 "CustomCDKBucketDeployment8693BB64968944B69AAFB0CC9EB8756CServiceRoleDefaultPolicy88902FDF":
 {
   "Type": "AWS::IAM::Policy",
   "Properties": {
    "PolicyDocument": {
     "Statement": [
      {
       "Action": [
        "s3:GetBucket*",
        "s3:GetObject*",
        "s3:List*"
       ],
       "Effect": "Allow",
       "Resource": [
        {
         "Fn::Join": [
          "",
          [
           "arn:",
           {
            "Ref": "AWS::Partition"
           },
           ":s3:::",
           {
            "Fn::Sub": "cdk-hnb659fds-assets-${AWS::AccountId}-${AWS::Region}"
           },
           "/*"
          ]
         ]
        },
        {
         "Fn::Join": [
          "",
          [
           "arn:",
           {
            "Ref": "AWS::Partition"
           },
           ":s3:::",
```

```
          {
           "Fn::Sub": "cdk-hnb659fds-assets-${AWS::AccountId}-${AWS::Region}"
          }
         ]
        ]
       }
      ]
     },
     {
      "Action": [
       "s3:Abort*",
       "s3:DeleteObject*",
       "s3:GetBucket*",
       "s3:GetObject*",
       "s3:List*",
       "s3:PutObject",
       "s3:PutObjectLegalHold",
       "s3:PutObjectRetention",
       "s3:PutObjectTagging",
       "s3:PutObjectVersionTagging"
      ],
      "Effect": "Allow",
      "Resource": [
       {
        "Fn::GetAtt": [
         "amzns3demobucket77F80CC0",
         "Arn"
        ]
       },
       {
        "Fn::Join": [
         "",
         [
          {
           "Fn::GetAtt": [
            "amzns3demobucket77F80CC0",
            "Arn"
           ]
          },
          "/*"
         ]
        ]
       }
      ]
```

```
      }
     ],
     "Version": "2012-10-17"
    },
    "PolicyName":
 "CustomCDKBucketDeployment8693BB64968944B69AAFB0CC9EB8756CServiceRoleDefaultPolicy88902FDF",
    "Roles": [
     {
      "Ref":
 "CustomCDKBucketDeployment8693BB64968944B69AAFB0CC9EB8756CServiceRole89A01265"
     }
    ]
   },
   "Metadata": {
    "aws:cdk:path": "CdkScratchStack/
Custom::CDKBucketDeployment8693BB64968944B69AAFB0CC9EB8756C/ServiceRole/DefaultPolicy/
Resource"
   }
  },
  "CustomCDKBucketDeployment8693BB64968944B69AAFB0CC9EB8756C81C01536": {
   "Type": "AWS::Lambda::Function",
   "Properties": {
    "Code": {
     "S3Bucket": {
      "Fn::Sub": "cdk-hnb659fds-assets-${AWS::AccountId}-${AWS::Region}"
     },
     "S3Key": "9eb41a5505d37607ac419321497a4f8c21cf0ee1f9b4a6b29aa04301aea5c7fd.zip"
    },
    "Role": {
     "Fn::GetAtt": [
      "CustomCDKBucketDeployment8693BB64968944B69AAFB0CC9EB8756CServiceRole89A01265",
      "Arn"
     ]
    },
    "Environment": {
     "Variables": {
      "AWS_CA_BUNDLE": "/etc/pki/ca-trust/extracted/pem/tls-ca-bundle.pem"
     }
    },
    "Handler": "index.handler",
    "Layers": [
     {
      "Ref": "BucketObjectsAwsCliLayer8C081206"
     }
```

```
    ],
    "Runtime": "python3.9",
    "Timeout": 900
   },
   "DependsOn": [

 "CustomCDKBucketDeployment8693BB64968944B69AAFB0CC9EB8756CServiceRoleDefaultPolicy88902FDF",
    "CustomCDKBucketDeployment8693BB64968944B69AAFB0CC9EB8756CServiceRole89A01265"
   ],
   "Metadata": {
    "aws:cdk:path": "CdkScratchStack/
Custom::CDKBucketDeployment8693BB64968944B69AAFB0CC9EB8756C/Resource",
    "aws:asset:path":
 "asset.9eb41a5505d37607ac419321497a4f8c21cf0ee1f9b4a6b29aa04301aea5c7fd",
    "aws:asset:is-bundled": false,
    "aws:asset:property": "Code"
   }
  }
```

4 líneas contra 241 es una diferencia enorme. Y esto es solo un ejemplo de lo que es posible cuando se aprovecha la capa 3 para personalizar las pilas.

Prácticas recomendadas

Constructos de capa 1

- No siempre se puede evitar el uso directo de los constructos de capa 1, pero hay que evitarlos siempre que sea posible. Si un constructo de capa 2 específico no es compatible con su caso particular extremo, puede explorar estas dos opciones en lugar de utilizar el constructo de capa 1 de manera directa:
  - Acceso`defaultChild`: si la propiedad CloudFormation que necesita no está disponible en un constructo de capa 2, puede acceder al constructo de capa 1 subyacente mediante `L2Construct.node.defaultChild`. Puede actualizar las propiedades públicas del constructo de capa 1. Para ello, acceda a este a través de esta propiedad, en lugar de tener que crear el constructo de capa 1.
  - Utilice anulaciones de propiedades: ¿Qué sucede si la propiedad que quiere actualizar no es pública? La mejor vía de escape que permite a AWS CDK hacer cualquier cosa que pueda hacer una plantilla de CloudFormation es utilizar un método disponible en todos los constructos de capa 1: addPropertyOverride. Puede manipular la pila a nivel de plantilla de CloudFormation al pasar el nombre y el valor de la propiedad de CloudFormation de manera directa a este método.

Constructos de capa 2

- Recuerde aprovechar los métodos auxiliares que suelen ofrecer los constructos de capa 2. Con la capa 2, no tiene que pasar cada propiedad tras la creación de instancias. Los métodos auxiliares de la capa 2 pueden hacer que el aprovisionamiento de recursos sea exponencialmente más práctico, sobre todo cuando es necesaria la lógica condicional. Uno de los métodos auxiliares más prácticos se deriva de la clase Grant. Esta clase no se utiliza de manera directa, pero muchos constructos de la capa 2 la utilizan para proporcionar métodos auxiliares que facilitan mucho la implementación de los permisos. Por ejemplo, si quiere dar permiso a una función de Lambda de la capa 2 para acceder a un bucket de la capa 2 de S3, puede llamar a `s3Bucket.grantReadWrite(lambdaFunction)` en lugar de crear un rol y política nuevos.

Constructos de capa 3

- Si bien los constructos de la capa 3 pueden resultar muy prácticos si quiere que las pilas sean más reutilizables y personalizables, le recomendamos utilizarlas con cuidado. Considere qué tipo de constructo de la capa 3 necesita o si necesita algún constructo de nivel 3:

- Si no interactúa de manera directa con los recursos de AWS, suele ser más adecuado crear una clase auxiliar en lugar de ampliar la case `Construct`. Esto se debe a que la clase `Construct` realiza muchas acciones de manera predeterminada que solo son necesarias si interactúa de manera directa con los recursos de AWS. Por lo tanto, si no es necesario que se realicen esas acciones, es más eficiente evitarlas.
- Si determina que es adecuado crear un constructo nuevo de la capa 3, en la mayoría de los casos será conveniente ampliar la clase `Construct` de manera directa. Amplíe otros constructos de la capa 2 solo cuando desee actualizar las propiedades predeterminadas del constructo. Si están implicados otros constructos de la capa 2 o una lógica personalizada, amplíe `Construct` de manera directa y cree una instancia de todos los recursos del constructor.

Preguntas frecuentes

¿No puedo usar las capas AWS CDK sin entender?

Por supuesto que sí puede. Pero como ocurre con las herramientas más poderosas, AWS CDK se vuelven más poderosas cuanto más se sabe sobre ellas. Al aprender cómo interactúan las capas, se obtiene un nuevo nivel de comprensión que ayuda a simplificar las implementaciones de pilas mucho más allá de lo que se puede hacer con solo unos conocimientos básicos AWS CDK . AWS CDK

¿Se pueden crear constructos de la capa 2 a partir de la capa 1 de la misma manera que hago los constructos de la capa 3 a partir de la capa 2?

Si un recurso ya tiene un constructo de la capa 2, le recomendamos utilizar ese constructo y haga las personalizaciones en la capa 3. Esto se debe a que ya se hicieron muchas investigaciones para determinar las mejores maneras de configurar los constructos existentes de la capa 2 para un recurso concreto. Sin embargo, hay varios constructos de la capa 1 cuyos constructos de la capa 2 aún no existen. En esos casos, lo animamos a crear sus propios constructos de la capa 2 y a compartirlos con otras personas, lo que lo convierte en colaborador de la biblioteca de código abierto de AWS CDK . Encontrará todo lo que necesita para empezar en las directrices de contribución del AWS CDK.

¿Qué AWS recursos aún no tienen construcciones oficiales de L2?

La cantidad de AWS recursos que no tienen construcciones de L2 disminuye día a día, pero si estás interesado en ayudar a crear una construcción de L2 para uno de estos recursos, consulta la referencia de la API.AWS CDK Vea la lista de recursos en el panel izquierdo. Los recursos que tienen el superíndice 1 junto a sus nombres no tienen constructos de la capa 2 oficiales.

¿Puedo crear una construcción L2 o L3 en cualquier idioma que sea compatible? AWS CDK

AWS CDK Es compatible con varios lenguajes de programación TypeScript JavaScript, incluidos Python, Java, C# y Go. Puede crear sus construcciones L3 personales utilizando el AWS CDK

código compilado en el lenguaje correspondiente. Sin embargo, si desea contribuir a las AWS CDK construcciones nativas AWS CDK o crearlas, debe utilizarlas. TypeScript Esto se debe a que TypeScript es el único idioma nativo de AWS CDK. Las AWS CDK versiones para otros idiomas se crean a partir del TypeScript código nativo mediante una AWS biblioteca llamada JSii.

¿Dónde puedo encontrar constructos de la capa 3 existentes fuera de AWS CDK?

Hay demasiadas ubicaciones para compartirlas aquí, pero puede encontrar muchas de las versiones más populares en el sitio web de AWS Solutions Constructs y en la AWS CDK sección del Construct Hub.

Recursos

- AWS CDK Referencia de la API de
- AWS CloudFormation Especificación de recursos de
- Documentación de los constructos de AWS CDK
- Abstracciones y vías de escape de AWS CDK
- Leverage L2 constructs to reduce the complexity of your AWS CDK application (entrada del blog de AWS)
- Recursos personalizados de AWS CloudFormation
- AWS Solutions Constructs
- Construct Hub
- Ejemplos de AWS CDK (repositorio de GitHub)

Historial de documentos

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

Cambio	Descripción	Fecha
Publicación inicial	—	4 de diciembre de 2023

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. Para más información, consulte la guía Cuantificación de la incertidumbre en los sistemas de aprendizaje profundo.

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.