

Lección 3

Automatización de la infraestructura



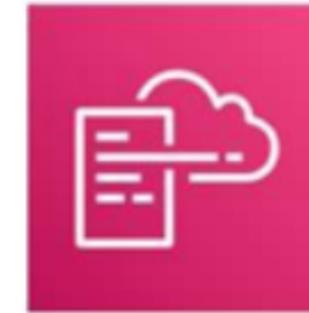
AWS CloudFormation ofrece una forma simplificada de modelar, crear y administrar una colección de recursos de AWS

- Una colección de recursos se conoce como una pila de AWS CloudFormation
- Sin cargo adicional (usted paga solo por los recursos que crea)

Puede crear, actualizar y eliminar pilas

Permite el aprovisionamiento ordenado y predecible y la actualización de recursos

Permite el control de versiones de las implementaciones de recursos de AWS



AWS
CloudFormation

 AWS CloudFormation aprovisiona recursos de forma repetible. Le permite crear y reconstruir su infraestructura y aplicaciones sin necesidad de realizar acciones manuales o escribir scripts personalizados. Con AWS CloudFormation, creará un documento que describe cuál debe ser su infraestructura, incluidos todos los recursos de AWS que deberían formar parte de la implementación. Piense en este documento como un modelo. A continuación, utiliza el modelo para crear la realidad, ya que AWS CloudFormation puede crear los recursos en su cuenta.



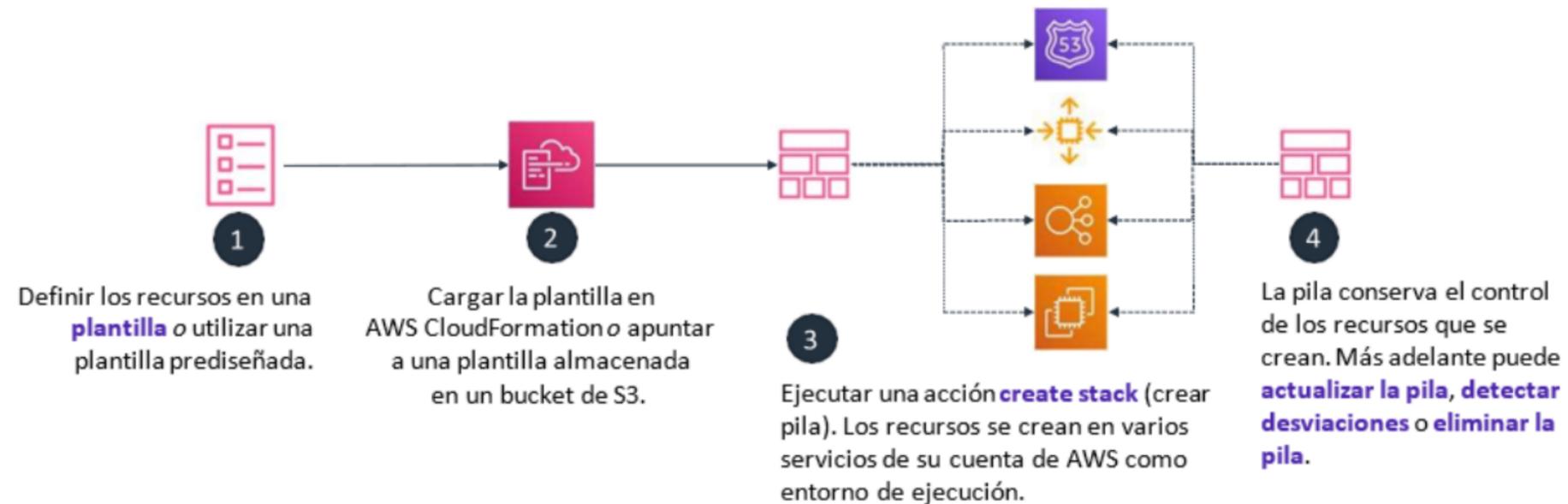
Cuando utiliza AWS CloudFormation para crear recursos, esto se conoce como una pila de AWS CloudFormation. Puede crear una pila, actualizar o eliminar una pila. Por lo tanto, puede aprovisionar recursos de manera ordenada y predecible.



El uso de AWS CloudFormation le permite tratar a su infraestructura como código (Infrastructure as Code, IaC). Puede utilizar cualquier editor de código para escribirla, insertarla en un sistema de control de versiones, como GitHub o AWS CodeCommit, y revisar los archivos con los miembros del equipo antes de implementarlos en los entornos correspondientes. Si el documento de AWS CloudFormation que crea para modelar su implementación ya está registrado en un sistema de control de versiones, siempre puede eliminar una pila, retirar una versión anterior del documento y crear una pila a partir de dicha versión. Con el control de versiones, puede utilizar las capacidades esenciales de restauración.



Información general de AWS CloudFormation



Este diagrama muestra cómo funciona AWS CloudFormation. Primero, define los recursos de AWS que desea crear. En el ejemplo que se muestra aquí, crea algunas instancias EC2, un balanceador de carga, un grupo de Auto Scaling y una zona alojada en Amazon Route 53. Los recursos se definen en una plantilla de AWS CloudFormation. Puede crear la plantilla desde cero o utilizar una plantilla prediseñada. Hay muchas plantillas de ejemplo disponibles.

1

Aunque AWS CloudFormation ofrece una amplia compatibilidad con los servicios de AWS, no todos los recursos se pueden crear con AWS CloudFormation. Para obtener más información, consulte los recursos admitidos por AWS CloudFormation.

2

Luego, debe cargar la plantilla en AWS CloudFormation. También puede almacenar la plantilla en Amazon S3 e indicarle a AWS CloudFormation la ubicación donde está almacenada.

3

En tercer lugar, debe ejecutar la acción create stack (crear pila). Cuando realice esta acción, el servicio AWS CloudFormation lee lo que se especifica en la plantilla y crea los recursos deseados en su cuenta de AWS. Una única pila puede crear y configurar recursos en una única región a través de varios servicios de AWS.

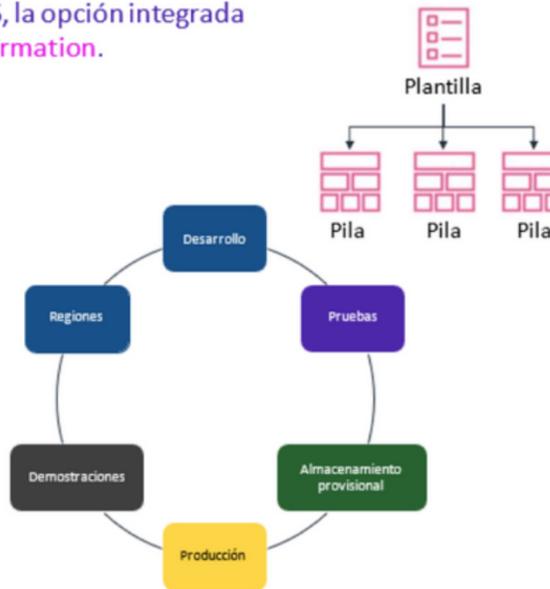
Por último, puede observar el progreso del proceso de creación de la pila. Una vez que la pila se haya creado correctamente, los recursos de AWS que creó existirán en su cuenta. El objeto de la pila permanece y actúa como un timón para todos los recursos que creó. Esto es útil cuando desea realizar acciones posteriores. Por ejemplo, es posible que desee actualizar la pila (para crear recursos adicionales de AWS o modificar recursos existentes) o eliminar la pila (lo que limpiará y eliminará los recursos creados por la pila).

Infraestructure as Code (IaC)



Para el desarrollo de la nube de AWS, la opción integrada para IaC es **AWS CloudFormation**.

- Infraestructure as Code o IaC es el proceso de aprovisionar y administrar sus recursos en la nube mediante la escritura de un archivo de plantilla que presenta las siguientes características:
 - Es legible por humanos
 - Es consumible por las máquinas
- Es una infraestructura que puede replicar, reimplementar, reutilizar
- Puede restaurarla al último estado favorable en caso de fallos

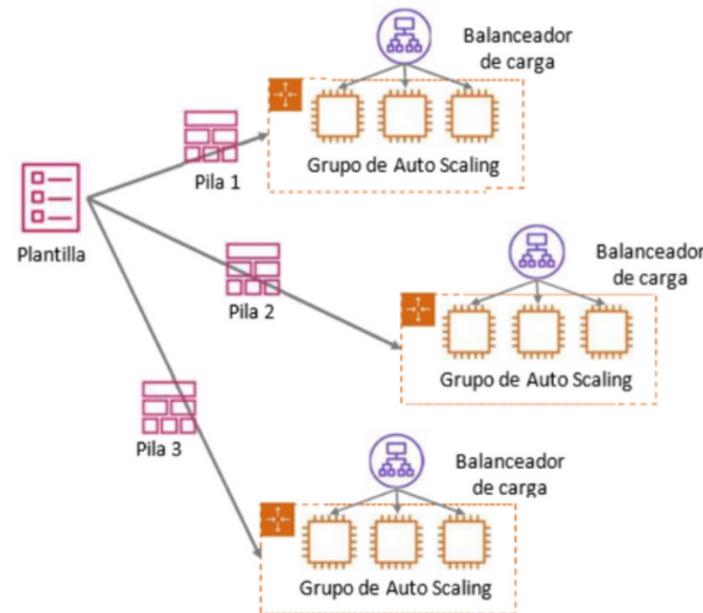


Infraestructure as Code (IaC) es un término de la industria que se refiere al proceso de aprovisionar y administrar recursos en la nube definiéndolos en un archivo de plantilla que es legible por humanos y consumible por las máquinas.

La popularidad de la IaC continúa creciendo porque proporciona una solución viable para desafíos como cómo replicar, volver a implementar y reutilizar la infraestructura de manera fácil, confiable y coherente.

La naturaleza transaccional de AWS CloudFormation es una de sus mayores ventajas desde el punto de vista del cliente. AWS CloudFormation es transaccional: el servicio se restaurará al último estado correcto en caso de fallos.

Infraestructure as code: Beneficios



Reduce los entornos equivalentes múltiples

- Implementación rápida de entornos complejos
- Coherencia de configuración
- Limpieza simple en cualquier momento (cuando se elimina la pila, se eliminan los recursos creados)
- Facilidad para propagar un cambio a todas las pilas
 - Modifique la plantilla, ejecute la actualización de pila en todas las pilas

Beneficios

- Reutilización
- Repetibilidad
- Mantenimiento

Ahora, considere algunos de los beneficios de la IaC con más detalle. Si construye una infraestructura con código, obtendrá beneficios como la capacidad de implementar entornos complejos rápidamente. Con solo una plantilla (o una combinación de plantillas), puede replicar entornos complejos una y otra vez.

En el ejemplo de aquí, una sola plantilla se puede utilizar para crear tres pilas diferentes. Cada pila se puede crear rápidamente, generalmente en cuestión de minutos. Cada pila replica los detalles de configuración compleja de forma coherente.



Si la pila 2 es su entorno de prueba y la pila 3 es su entorno de producción, puede confiar en que, si sus trabajos de prueba funcionan bien en el entorno de prueba, también funcionarán bien en el entorno de producción. La plantilla minimiza el riesgo de que el entorno de prueba esté configurado de manera diferente del entorno de producción.

Además, si tiene que realizar una actualización de configuración en el entorno de prueba, puede actualizar la plantilla con el cambio y actualizar todas las pilas. Este proceso ayuda a garantizar que las modificaciones realizadas en un solo entorno se propaguen de forma fiable a todos los entornos que deberían recibir la actualización.

Otro beneficio es que es más fácil limpiar todos los recursos que se crearon en su cuenta para admitir un entorno de prueba cuando ya no los necesite. Esto ayuda a reducir los costos que no están asociados a recursos que ya no necesita y ayuda a mantener su cuenta libre de códigos mal diseñados, complicados o no deseados.



Sintaxis de una plantilla de AWS CloudFormation

```
{  
  "AWSTemplateFormatVersion": "2010-09-09",  
  "Resources" : {  
    "awsexamplebucket1" : {  
      "Type" : "AWS::S3::Bucket"  
    }  
  }  
}
```

Ejemplo de JSON

```
AWSTemplateFormatVersion: 2010-09-09  
Resources:  
  awsexamplebucket1:  
    Type: AWS::S3::Bucket
```

Ejemplo de YAML

Las plantillas también se pueden crear en [AWS CloudFormation Designer](#), una interfaz de diseño gráfico en la consola de administración de AWS.

Plantillas de AWS CloudFormation

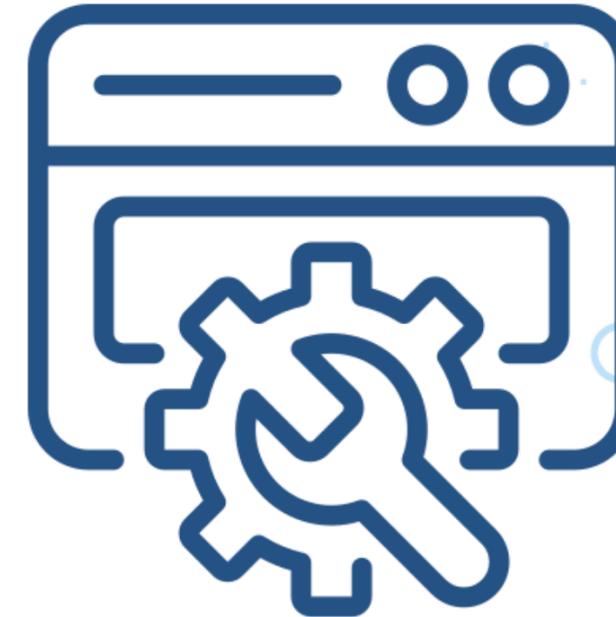
- Se crean en el lenguaje de notación de objetos JavaScript (JSON) o YAML Ain't Markup Language (YAML)
- Ventajas de YAML:
 - Menos detallado (sin {}, "", caracteres)
 - Es compatible con comentarios incrustados
- Ventajas de JSON:
 - Ampliamente utilizado por otros sistemas informáticos (por ejemplo, API)
- Recomendación: Trate a las plantillas como código fuente
 - Almacene las plantillas en un repositorio de código

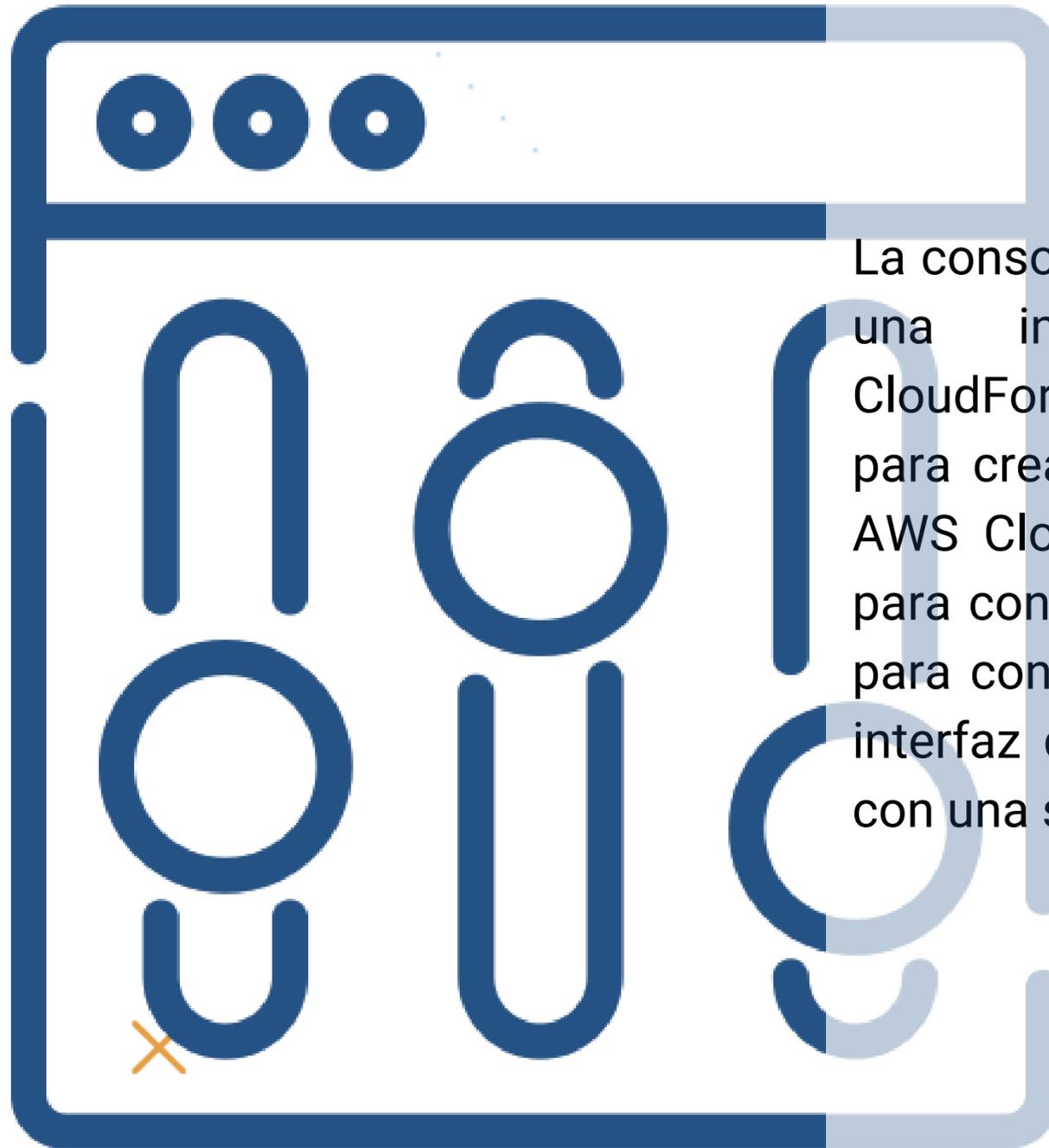
Las plantillas de AWS CloudFormation se pueden escribir en archivos de notación de objetos de JavaScript (JSON) o YAML Ain't Markup Language (YAML).

YAML está optimizado para facilitar la lectura. Los mismos datos que se almacenan en JSON necesitarán menos líneas para almacenarse en YAML, ya que en YAML no se utilizan llaves ({}) y se utilizan menos comillas (""). Otra ventaja de YAML es que admite los comentarios incrustados en forma nativa. Por último, a muchas personas les resulta más fácil depurar documentos YAML que con JSON. Con JSON, puede ser difícil encontrar comas o llaves faltantes o mal colocados. YAML no tiene este problema.



A pesar de todas las ventajas de YAML, JSON ofrece algunas ventajas únicas. En primer lugar, se lo utiliza ampliamente en sistemas informáticos. Su ubicuidad es una ventaja, ya que los datos que se almacenan en JSON pueden usarse con confianza en muchos sistemas sin necesidad de transformarlos. Además, suele ser más fácil generar y analizar JSON que YAML.





La consola de administración de AWS proporciona una interfaz gráfica, denominada AWS CloudFormation Designer, que se puede utilizar para crear o ver el contenido de las plantillas de AWS CloudFormation. También se puede utilizar para convertir una plantilla JSON válida a YAML o para convertir de YAML a JSON. Proporciona una interfaz de arrastrar y soltar para crear plantillas con una salida que puede estar en JSON o YAML.

Plantilla simple: Crear una instancia EC2

```
{  
  "AWSTemplateFormatVersion": "2010-09-09",  
  "Description": "Create EC2 instance",  
  "Parameters": {  
    "KeyPair": {  
      "Description": "SSH Key Pair",  
      "Type": "String"}},  
  "Resources": {  
    "Ec2Instance": {  
      "Type": "AWS::EC2::Instance",  
      "Properties": {  
        "ImageId": "ami-9d23aeaa",  
        "InstanceType": "m3.medium",  
        "KeyName": {"Ref": "KeyPair"}  
      }  
    }  
  },  
  "Outputs": {  
    "InstanceId": {  
      "Description": "InstanceId",  
      "Value": {"Ref": "Ec2Instance"}  
    }  
  }  
}
```

← **Parameters (parámetros)** : especifican qué valores se pueden establecer durante el tiempo de ejecución cuando crea la pila

- Ejemplos de usos: configuración específica de la región o configuración de entorno de producción frente a entorno de prueba

← **Resources (recursos)** : definen lo que se debe crear en la cuenta de AWS

- Ejemplo: crear todos los componentes de una nube virtual privada (VPC) en una región y, a continuación, crear instancias EC2 en la VPC
- Pueden hacer referencia a parámetros

← **Outputs (resultados)** : especifican los valores devueltos después de crear la pila

- Ejemplo de uso: devolver el ID de instancia o la dirección IP pública de una instancia EC2

En este ejemplo, la plantilla de AWS CloudFormation crea una instancia EC2. Aunque este ejemplo no ilustra todas las secciones posibles de una plantilla, sí resalta algunas de las secciones más utilizadas, incluidos parámetros, recursos y salidas.



Parameters (parámetros) es una sección opcional de la plantilla. Los parámetros son valores que se pasan a su plantilla en tiempo de ejecución (al crear o actualizar una pila). Puede hacer referencia a los parámetros de las secciones Resources (recursos) y Outputs (salidas) de la plantilla. El nombre del parámetro y su descripción aparecen en la página Specify Parameters (especificación de parámetros) cuando un usuario utiliza la plantilla en el asistente Create Stack (crear pila).

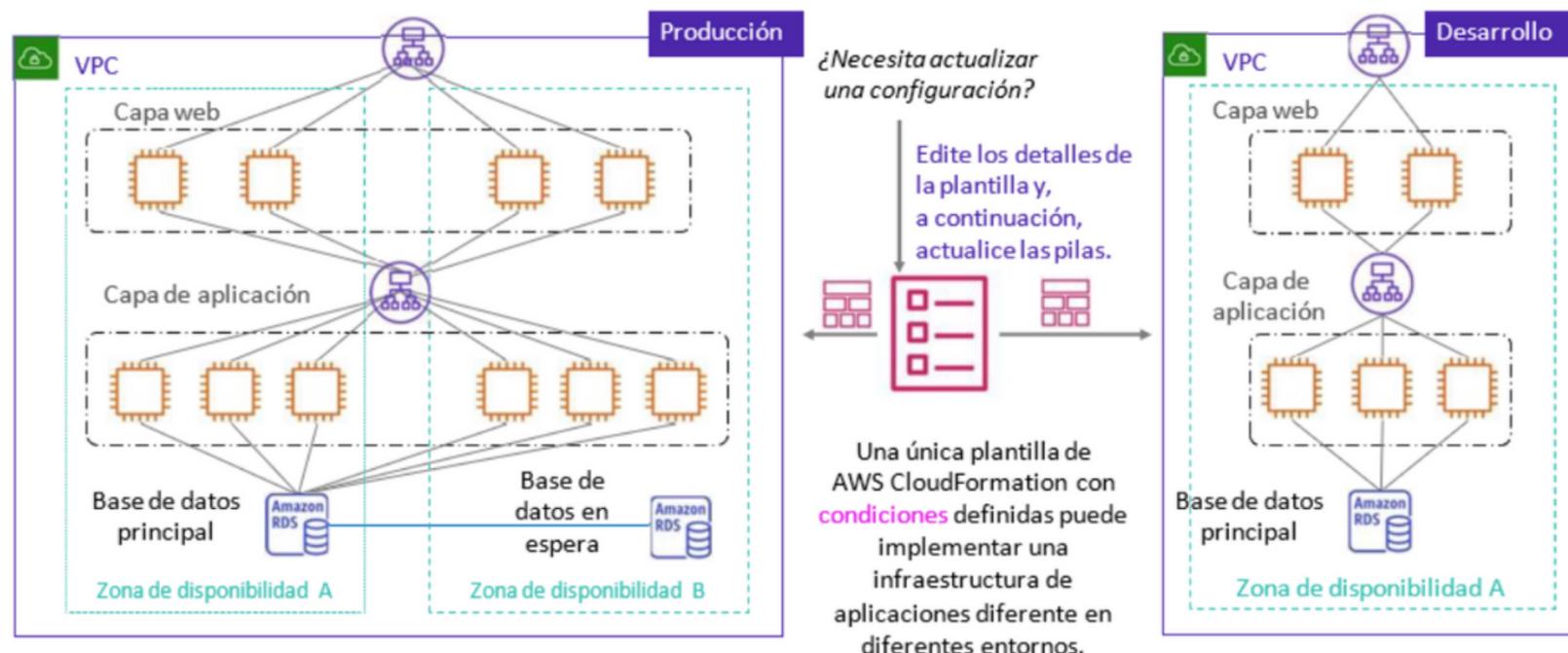
Resources (recursos) es una sección obligatoria de la plantilla. Se utilizan para especificar los recursos de AWS que se crearán, junto con sus propiedades. En este ejemplo, se especifica un recurso de tipo `AWS::EC2::Instance`, que crea una instancia EC2. El recurso de ejemplo incluye propiedades definidas estáticamente (`ImageID` e `InstanceType`) y una referencia al parámetro `KeyPair`.





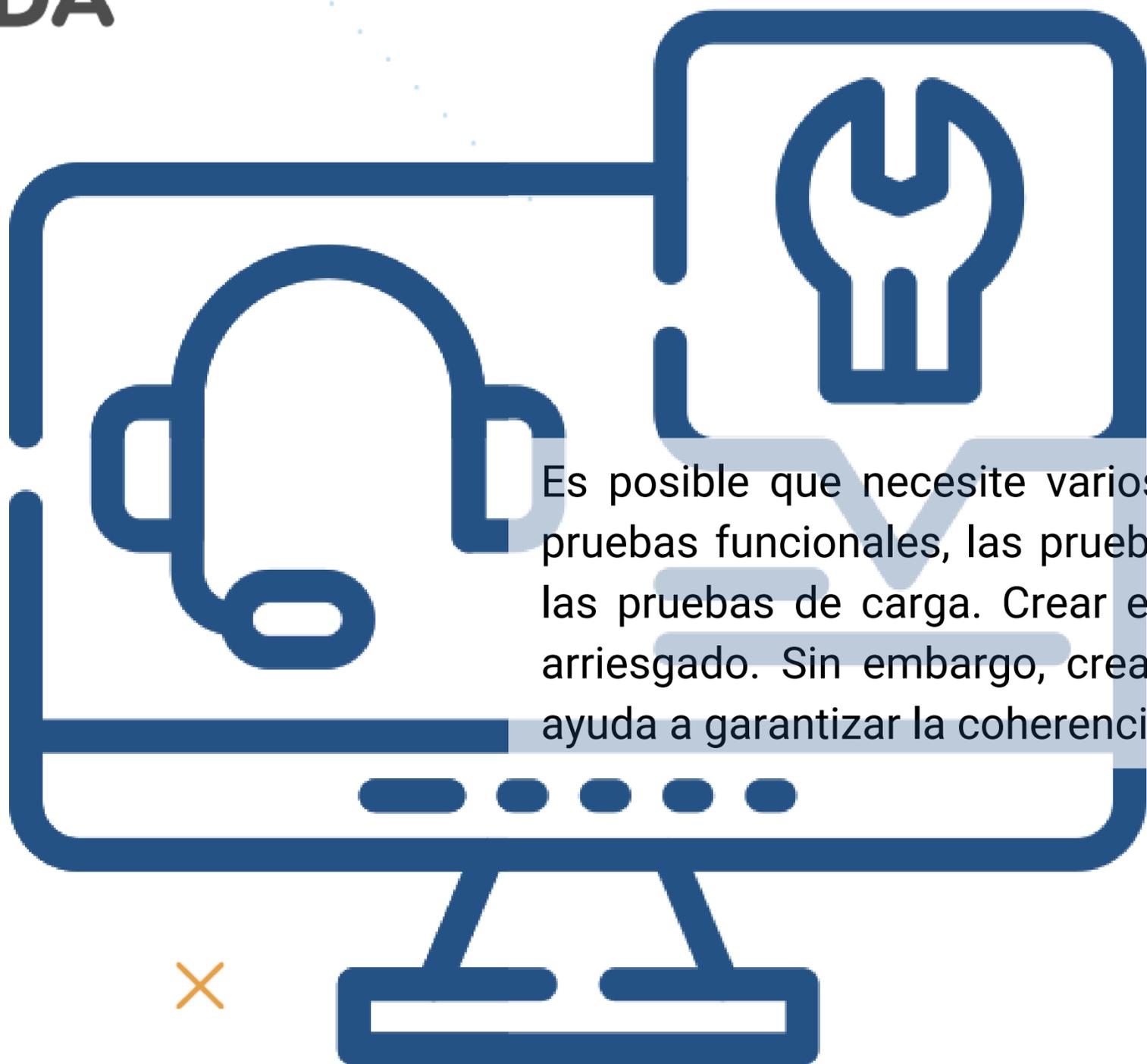
Finalmente, el ejemplo muestra la sección Outputs (salidas). Outputs (salidas) describe los valores que se devuelven cuando ve las propiedades de su pila. En el ejemplo, se declara una salida InstanceID. Después de crear la pila, puede ver este valor en los detalles de la pila en la consola de AWS CloudFormation si ejecuta el comando `aws cloudformation describe-stacks` de la interfaz de línea de comandos de AWS (CLI de AWS) o si utiliza los kits de desarrollo de software (SDK) de AWS para recuperar este valor.

AWS CloudFormation



Puede utilizar la misma plantilla de AWS CloudFormation para crear su entorno de producción y nuestro entorno de desarrollo. Este enfoque puede ayudar a garantizar (por ejemplo) que se utilicen los mismos binarios de aplicación, la misma versión de Java y la misma versión de base de datos en el desarrollo y en la producción. Por lo tanto, la plantilla puede ayudar a garantizar que su aplicación se comporte en la producción del mismo modo que se comportaba en el entorno de desarrollo.

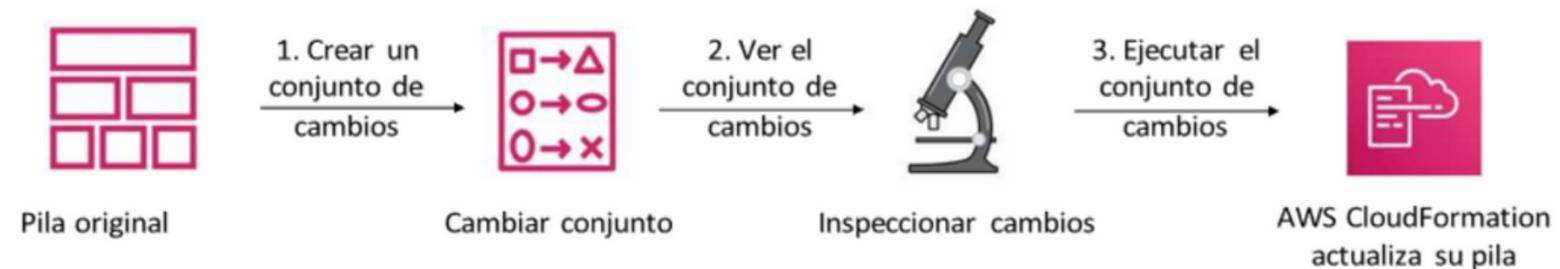
En el ejemplo, verá que los entornos de producción y desarrollo se crean a partir de la misma plantilla. Sin embargo, el entorno de producción está configurado para ejecutarse en dos zonas de disponibilidad y el entorno de desarrollo se ejecuta en una única zona de disponibilidad. Este tipo de diferencias específicas de implementación se pueden lograr mediante el uso de condiciones. Puede utilizar una declaración de Conditions (condiciones) en las plantillas de AWS CloudFormation para asegurarse de que, aunque sean diferentes en tamaño y alcance, los entornos de desarrollo, prueba y producción estén configurados de forma idéntica.



Es posible que necesite varios entornos de prueba para las pruebas funcionales, las pruebas de aceptación del usuario y las pruebas de carga. Crear esos entornos manualmente es arriesgado. Sin embargo, crearlos con AWS CloudFormation ayuda a garantizar la coherencia y la repetibilidad.

Conjuntos de cambios de AWS CloudFormation

Los conjuntos de cambios permiten visualizar una versión preliminar de los cambios antes de implementarlos.



Utilice el atributo `DeletionPolicy` para conservar o realizar una copia de seguridad de un recurso cuando se elimine o actualice su pila.

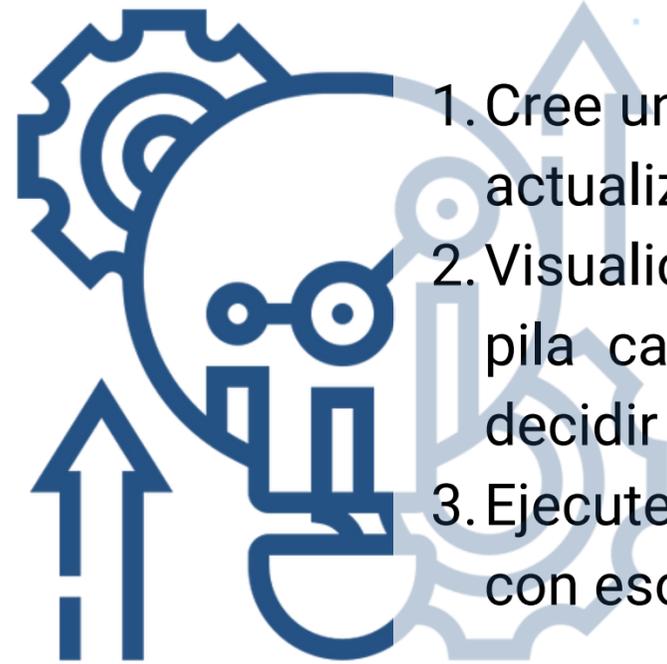
Una forma de actualizar una pila (y, por lo tanto, actualizar los recursos de AWS) es actualizar la plantilla de AWS CloudFormation que utilizó para crear la pila y, a continuación, ejecutar la opción Update Stack (actualizar pila).

Sin embargo, es posible que desee obtener información adicional sobre los cambios específicos que AWS CloudFormation implementará si ejecuta ese comando, antes de ejecutar realmente una actualización. Si desea este tipo de información, puede utilizar un conjunto de cambios.

Los conjuntos de cambios le permiten visualizar previamente los cambios, comprobar que se alinean con sus expectativas y, a continuación, aprobar las actualizaciones antes de continuar.



Siga este flujo de trabajo básico para utilizar los conjuntos de cambios de AWS CloudFormation:

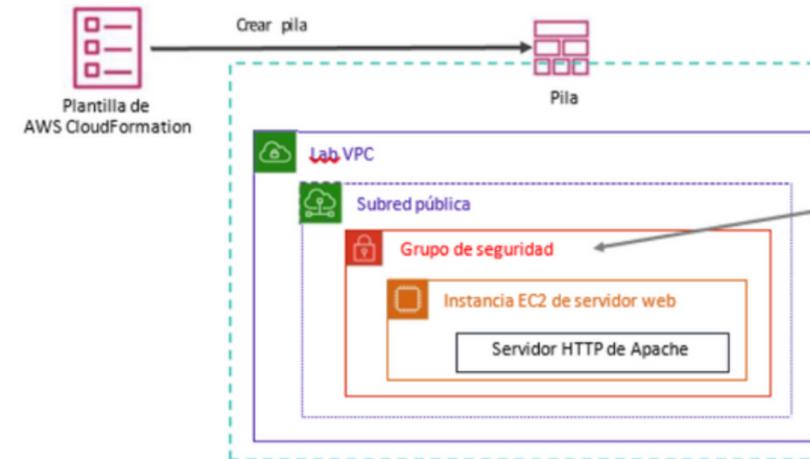


1. Cree un conjunto de cambios enviando cambios para la pila que desea actualizar.
2. Visualice el conjunto de cambios para ver qué ajustes y recursos de pila cambiarán. Si desea tener en cuenta otros cambios antes de decidir qué cambios hacer, cree conjuntos de cambios adicionales.
3. Ejecute el conjunto de cambios. AWS CloudFormation actualiza su pila con esos cambios.

Si utiliza conjuntos de cambios, es posible que desee establecer una política de eliminación en algunos recursos. Con el atributo DeletionPolicy, puede conservar o (en algunos casos) crear una copia de seguridad de un recurso cuando se elimina o actualiza su pila. Si un recurso no tiene ningún atributo DeletionPolicy, AWS CloudFormation elimina el recurso.

Detección de desviaciones

Considere esta situación: Se crea un entorno de aplicación ejecutando una pila de AWS CloudFormation. A continuación, alguien decide modificar manualmente la configuración del entorno implementado.



Caso:

1. Una pila de AWS CloudFormation crea un entorno de aplicación.
2. Más tarde, alguien **modifica manualmente** el grupo de seguridad y abre un nuevo puerto TCP entrante.
3. La **detección de desviaciones** se ejecuta en la pila.
4. Todos los recursos excepto el grupo de seguridad muestran el resultado **IN_SYNC** (sincronizado), pero el grupo de seguridad muestra el estado **MODIFIED** (modificado), con detalles.

Pregunta: En este caso, ¿cuál sería un mejor enfoque si el equipo desea modificar la configuración del grupo de seguridad?

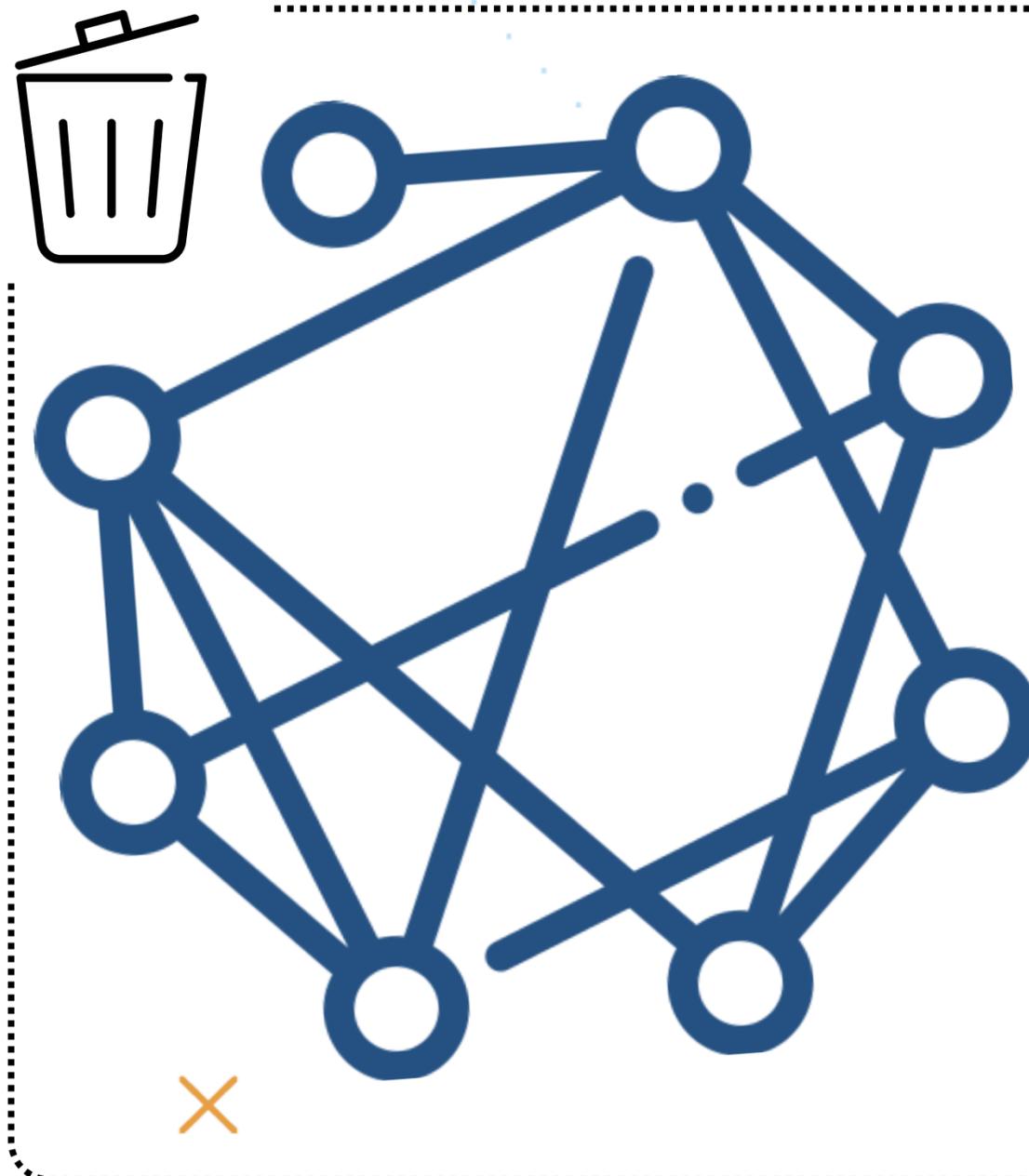
Respuesta: Modificar la configuración del grupo de seguridad de la plantilla de AWS CloudFormation. A continuación, ejecutar la acción Update Stack. AWS CloudFormation actualizará el grupo de seguridad. Mantiene la implementación del *modelo* sincronizada con la implementación real.

Crean una nueva regla de entrada en el grupo de seguridad creado por la pila. Sin embargo, realizan este cambio fuera del contexto de AWS CloudFormation (por ejemplo, mediante la consola de Amazon EC2). Como arquitecto de esta aplicación, querrá saber que su entorno implementado ya no coincide con el entorno del modelo definido en la plantilla de AWS CloudFormation.

¿Cómo sabría qué recursos se modificaron y ya no se ajustan exactamente a las especificaciones en la pila?

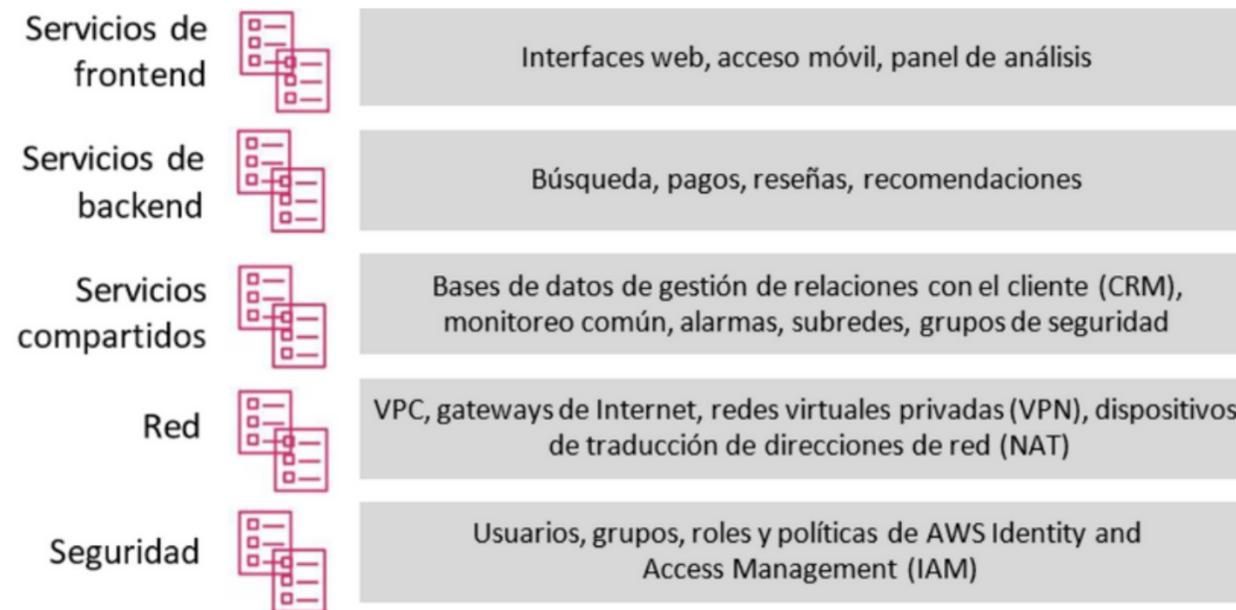
La detección de desviaciones se puede ejecutar en una pila seleccionando Detect Drift (Detectar desviaciones) en el menú Stack actions (Acciones de pila) de la consola. Llevar a cabo una operación de detección de desviaciones sobre una pila determina si la pila se ha desviado de su configuración de plantilla prevista. La detección de desviaciones devuelve información detallada sobre el estado de desviación de cada recurso de la pila que admite detección de desviaciones.





Cuando elimina una pila que tiene desviaciones, el proceso de limpieza de recursos de AWS CloudFormation no controla las desviaciones. Si la pila tiene dependencias de recursos sin resolver, podrían provocar un error en la acción de eliminación de la pila. En tal caso, es posible que tenga que resolver el problema manualmente. Para obtener más información al respecto, consulte la lista de recursos que admiten la detección de desviaciones.

Alcance y organización de plantillas



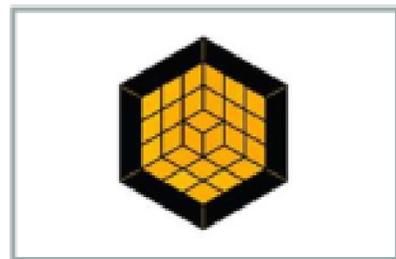
El diagrama ofrece algunas ideas sobre cómo puede organizar las plantillas para que sean más fáciles de mantener y para que se puedan utilizar en combinación, de una manera que tenga sentido. Una buena estrategia consiste en agrupar las definiciones de recursos en plantillas, de manera similar a la forma en que organizaría la funcionalidad de una aplicación empresarial grande en diferentes partes.

A medida que su organización comience a utilizar más plantillas de AWS CloudFormation, será importante que tener una estrategia para las plantillas. Esta estrategia definirá el alcance de lo que debe crear una sola plantilla y las características generales que podrían hacer que quiera definir su infraestructura de AWS en más de una plantilla.

Piense en los componentes más estrechamente conectados de su infraestructura y considere colocarlos en las mismas plantillas. En este ejemplo, las plantillas de AWS CloudFormation tienen un alcance para crear y mantener recursos de AWS en una de las cinco áreas: servicios de frontend, servicios de backend, servicios compartidos, red y seguridad. En cada área, puede mantener una plantilla que se ajuste a las necesidades de una sola aplicación o de un solo departamento.

Independientemente de cómo organice y programe el alcance de cada plantilla de AWS CloudFormation, trate las plantillas como a un código que necesita control de versiones. Almacene sus plantillas en un sistema de control de código fuente.

AWS Quick Starts



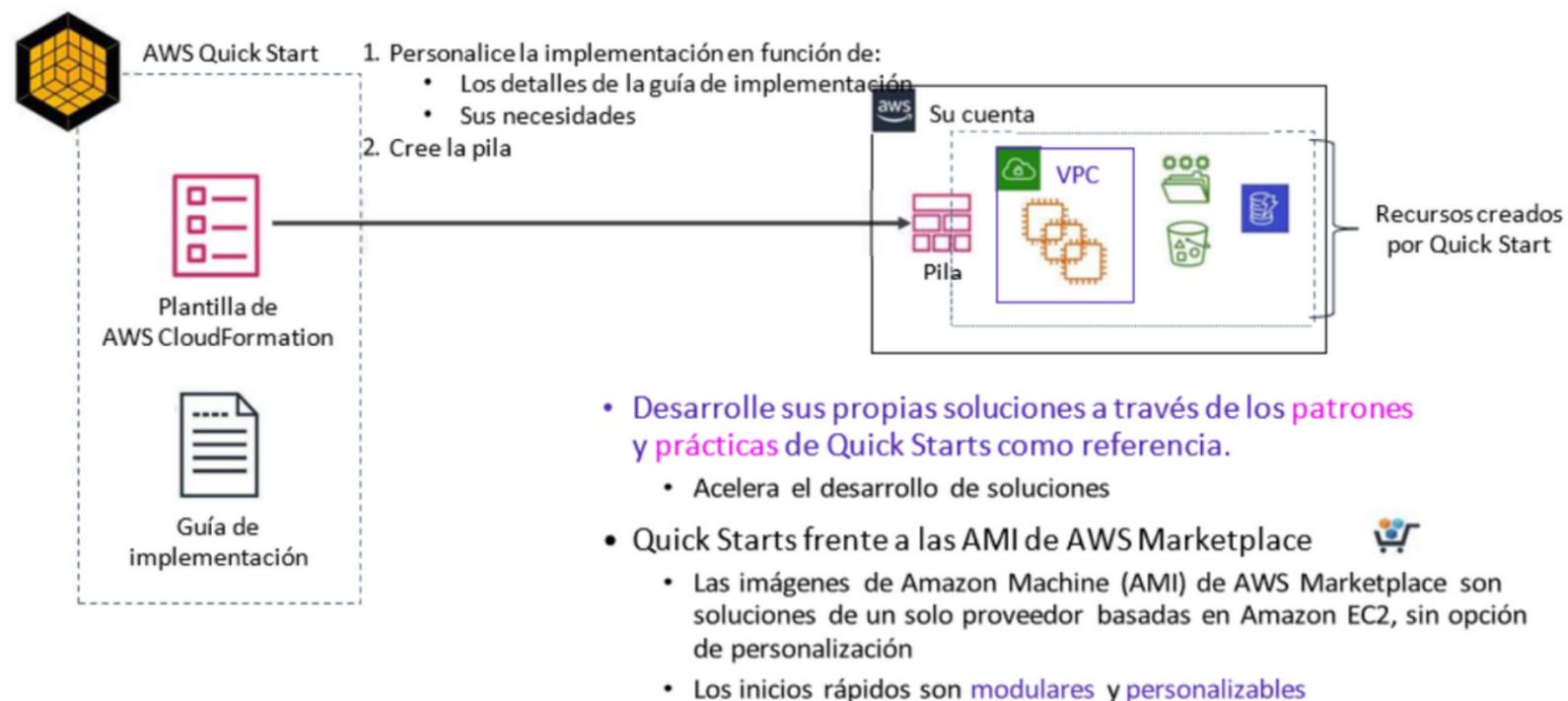
Plantillas de AWS CloudFormation
creadas por arquitectos de soluciones
de AWS

- Son implementaciones de referencia
- Están basadas en las prácticas recomendadas de AWS en materia de seguridad y alta disponibilidad
- Se pueden utilizar para crear arquitecturas enteras con un solo clic en menos de una hora
- Se pueden utilizar para experimentación y como base para sus propias arquitecturas

AWS Quick Starts ofrece plantillas de AWS CloudFormation. Los socios y arquitectos de soluciones de AWS crean inicios rápidos para ayudarlo a implementar soluciones populares en AWS, basadas en las prácticas recomendadas de AWS para seguridad y alta disponibilidad. Estas implementaciones de referencia se pueden aprovisionar en su cuenta de AWS en menos de una hora, muchas veces en solo minutos. Le ayudan a crear un entorno de prueba o producción de buena arquitectura en unos pocos pasos. Puede utilizarlas para crear arquitecturas enteras o utilizarlas para experimentar con nuevos enfoques de implementación.

Para obtener más información al respecto, diríjase a la página de AWS Quick Starts.

Cómo utilizar AWS Quick Starts



Cada inicio rápido consta de una plantilla de AWS CloudFormation y una guía de implementación. La guía proporciona detalles sobre las opciones de implementación y cómo configurar la implementación para que se ajuste a sus necesidades.

Personalice la implementación para que se ajuste a sus necesidades y cree la pila. Dependiendo de los recursos de AWS que se deban crear, la plantilla de Quick Starts finalizará la implementación en cuestión de minutos o en unas pocas horas.

Incluso si no utiliza las plantillas de Quick Starts, puede que le resulte útil examinar algunas de ellas para ver los tipos de patrones y prácticas que siguen. Podría encontrarlas útiles para acelerar el desarrollo de su propia plantilla si toma prestada alguna función de una plantilla de Quick Start y la incrusta en su propia plantilla.



Las Imágenes de Amazon Machine (AMI) de AWS Marketplace son otra solución que las personas utilizan a veces. Estos recursos se pueden lanzar desde la consola de Amazon EC2. Las AMI de AWS Marketplace ofrecen soluciones con un solo proveedor que se ejecutan en instancias EC2. Por el contrario, AWS Quick Starts son soluciones modulares y más personalizables que pueden (o no) utilizar Amazon EC2.

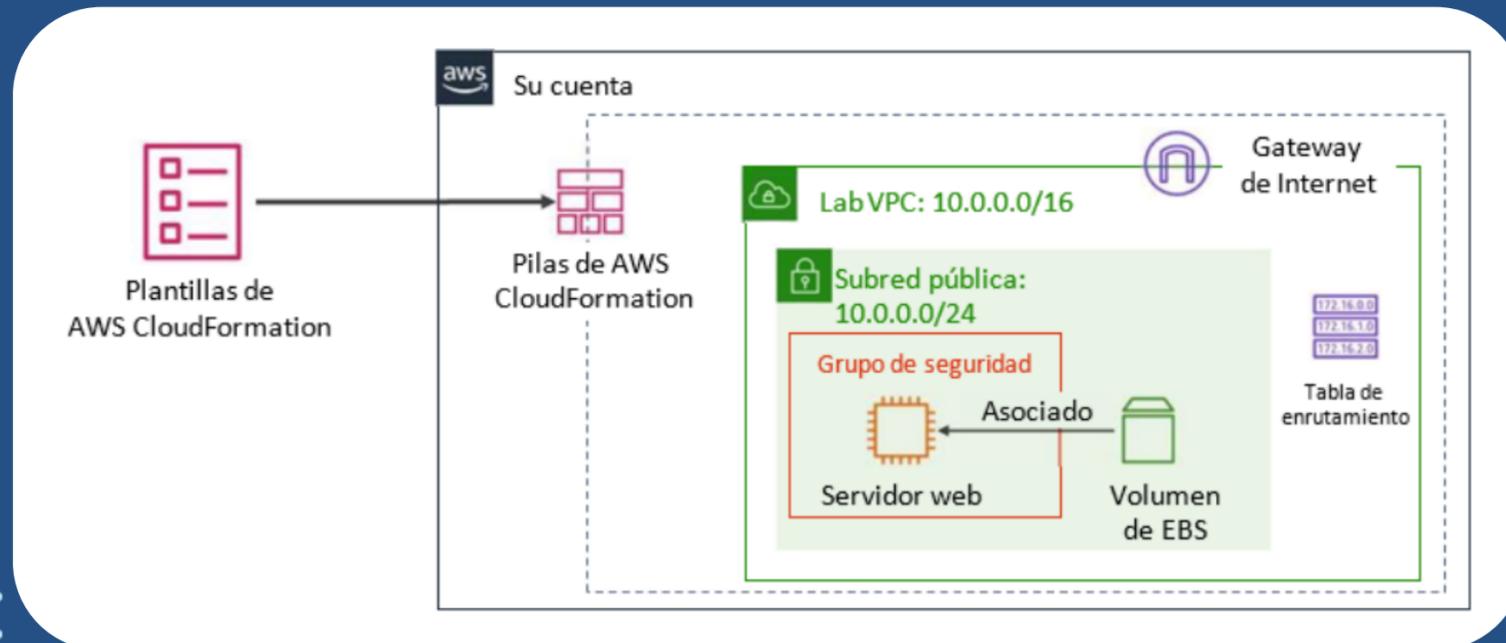
El instructor puede elegir demostrar la estructura de una plantilla de AWS CloudFormation y, a continuación, crear una pila de AWS CloudFormation utilizando la plantilla.

Ahora completará el laboratorio guiado de la unidad 5:
Automatización de la implementación de infraestructura con
AWS CloudFormation.

En este laboratorio guiado, realizará las siguientes tareas:

1. Implementar una capa de red
2. Implementar una capa de aplicación
3. Actualizar una pila
4. Explorar las plantillas con AWS CloudFormation Designer
5. Eliminar la pila





Al final de este laboratorio guiado, habrá utilizado AWS CloudFormation para crear los recursos del diagrama. Los recursos de la capa de red se crean cuando crea la primera pila. La instancia EC2, el grupo de seguridad y el volumen de Amazon Elastic Block Store (Amazon EBS) se crean cuando crea la segunda pila.

A continuación, la configuración del grupo de seguridad se actualiza en la plantilla que se utiliza para crear la segunda pila. Esta modificación se aplica al recurso del grupo de seguridad cuando se ejecuta la acción update stack (actualizar pila).

Llegó la hora de iniciar el laboratorio guiado

Estos son algunos de los aprendizajes clave de esta de la unidad:

- AWS CloudFormation es un servicio de Infrastructure as Code (IaC) que le permite modelar, crear y administrar una colección de recursos de AWS
- Infrastructure as Code de AWS CloudFormation se define en plantillas creadas en JSON o YAML
- Una pila es lo que crea cuando utiliza una plantilla para crear recursos de AWS
- Las acciones que están disponibles en una pila existente incluyen update stack, detect drift y delete stack
- AWS Quick Starts proporciona plantillas de AWS CloudFormation que han sido creadas por arquitectos de soluciones y reflejan las prácticas recomendadas de AWS

INICIO