

Cloudformation



AWS CloudFormation

¿Qué es AWS CloudFormation?

AWS CloudFormation es un servicio que permite modelar, aprovisionar y gestionar recursos de AWS y de terceros mediante la infraestructura como código.

Desafíos de implementación en la nube

Realizar implementaciones en varios lugares geográficos

Depurar implementaciones

Documentar todos los cambios

Actualizar servidores en vivo

Tener la capacidad de administrar una restauración

Administrar dependencias de sistemas y subsistemas

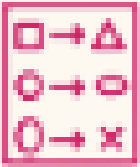
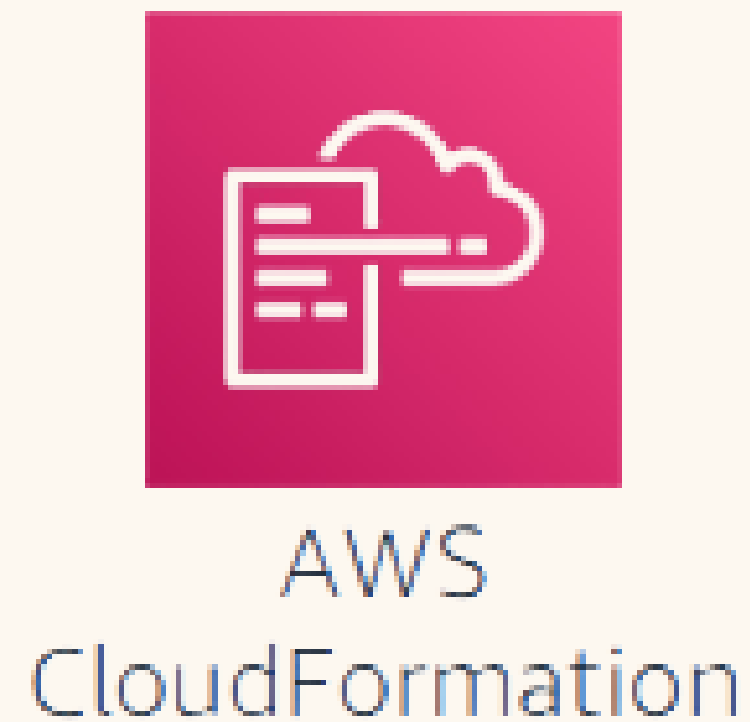
Implementar entornos repetibles e idénticos

Beneficios de AWS CloudFormation

- **Automatización**: Provisiona infraestructura de forma automatizada.
- **Escalabilidad**: Facilita la gestión de entornos grandes y complejos.
- **Reproducibilidad**: Define recursos de manera consistente en distintos entornos.
- **Gestor de cambios**: Permite visualizar cambios antes de aplicarlos.
- **Auditoría y control**: Se integra con AWS CloudTrail para rastrear cambios.

Conceptos Clave

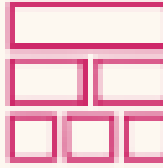
Las plantillas en CloudFormation son archivos JSON o YAML que definen la infraestructura a aprovisionar:



Conjunto de
cambios

The icon for "Conjunto de cambios" (Change Set), showing a pink square with three rows of symbols: a square to a triangle, a circle to a circle, and a circle to an 'x'. Below the icon, the text "Conjunto de cambios" is displayed in a dark blue font.

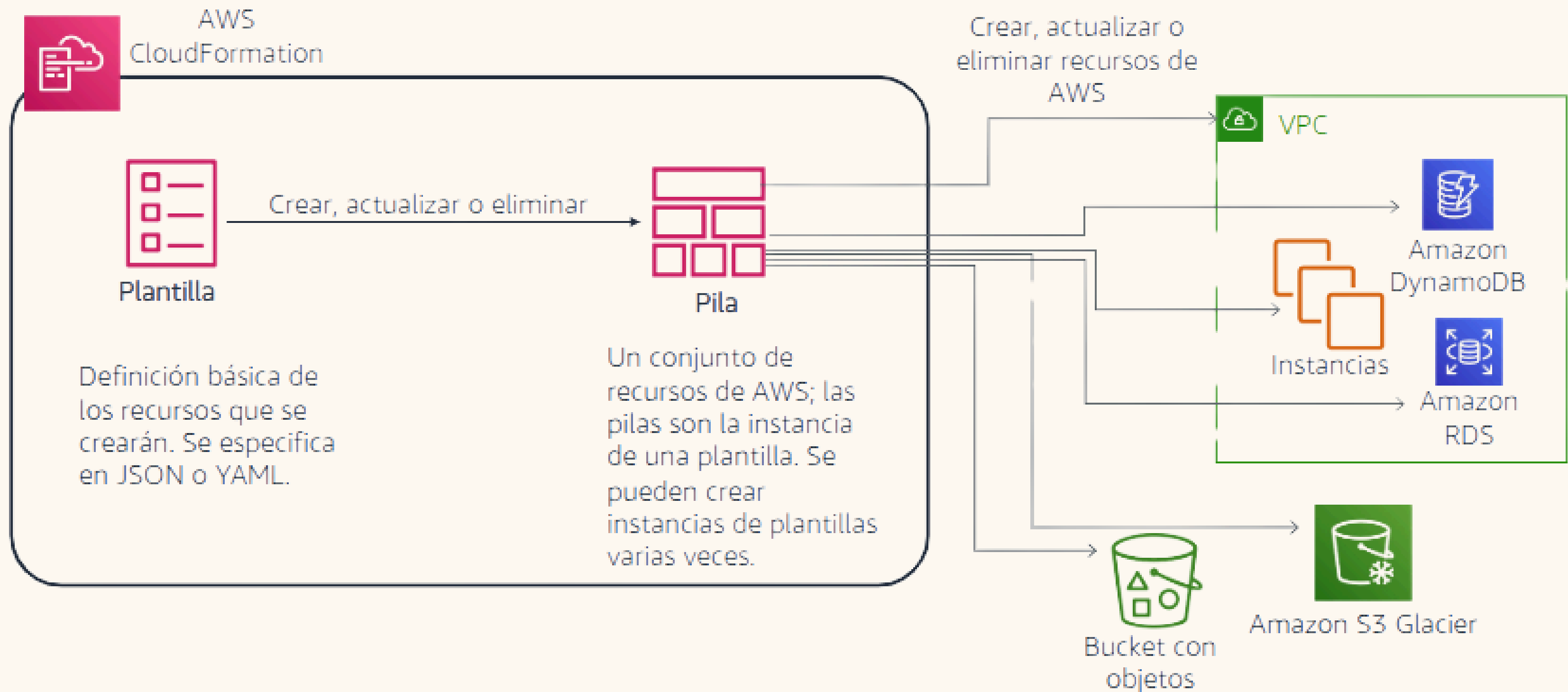
Plantilla

The icon for "Plantilla" (Template), showing a pink square with three rows of a square followed by a horizontal line. Below the icon, the text "Plantilla" is displayed in a dark blue font.

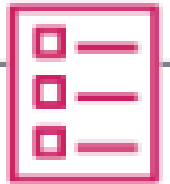
Pila

The icon for "Pila" (Stack), showing a pink square with a grid of four smaller squares arranged in two rows and two columns. Below the icon, the text "Pila" is displayed in a dark blue font.

Terminología de AWS CloudFormation



Estructura de la plantilla



Plantilla

- **Parameters (Parámetros)**
- **Mappings (Mapeo)**
- **Resources (Recursos)**
 - ◆ AWS::EC2::Instance
 - + Properties/UserData
 - + Metadata/AWS::CloudFormation::Init
 - Paquetes
 - Grupos
 - Usuarios
 - Fuentes
 - Archivos
 - Comandos
 - Servicios
 - ◆ AWS::CloudFormation::WaitCondition
 - DependsOn: EC2 Instance
 - ◆ AWS::CloudFormation::WaitConditionHandle
- **Outputs (Resultados)**

Parameters (Parámetros): entradas en la plantilla

Mappings (Mapeo): variables estáticas, normalmente Amazon Machine Image (AMI) más recientes

Resources (Recursos): recursos de AWS que se crearán

Init: aplicaciones personalizadas que se ejecutarán durante el inicio

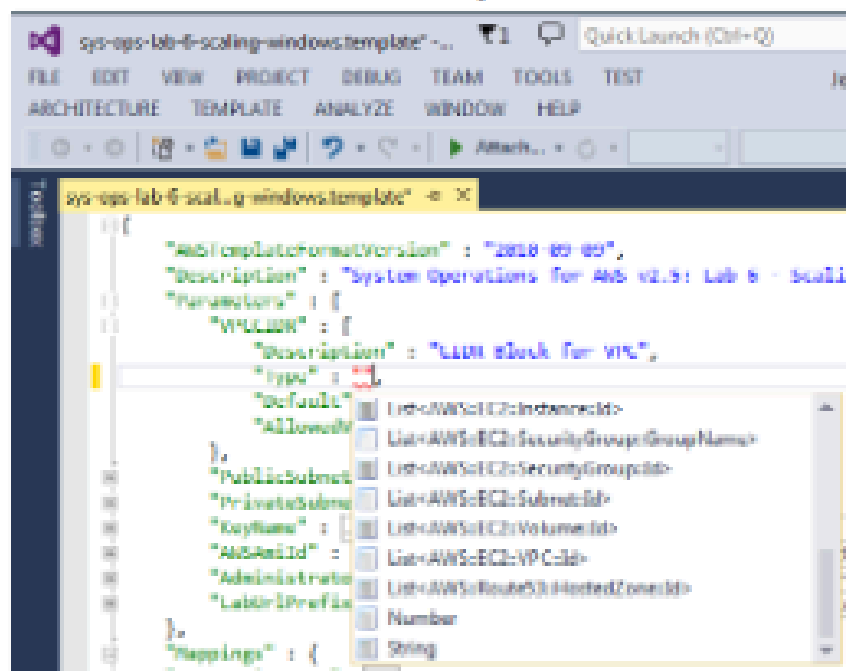
WaitCondition: señal de la instancia de que se han terminado de ejecutar los datos de usuario

Outputs (Resultados): valores de recursos personalizados que se crearon con la plantilla (URL, nombres de usuario, etc.)

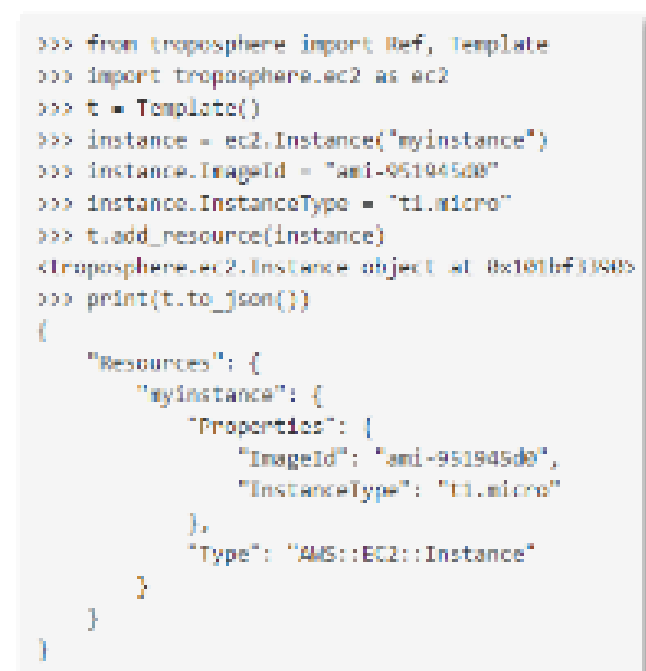
Editar plantillas de AWS CloudFormation

Para editar se debe usar un editor que sea compatible con AWS CloudFormation (o un editor de JSON o de YAML) para aprovechar características como las capacidades de análisis, finalización automática y comprobación de sintaxis

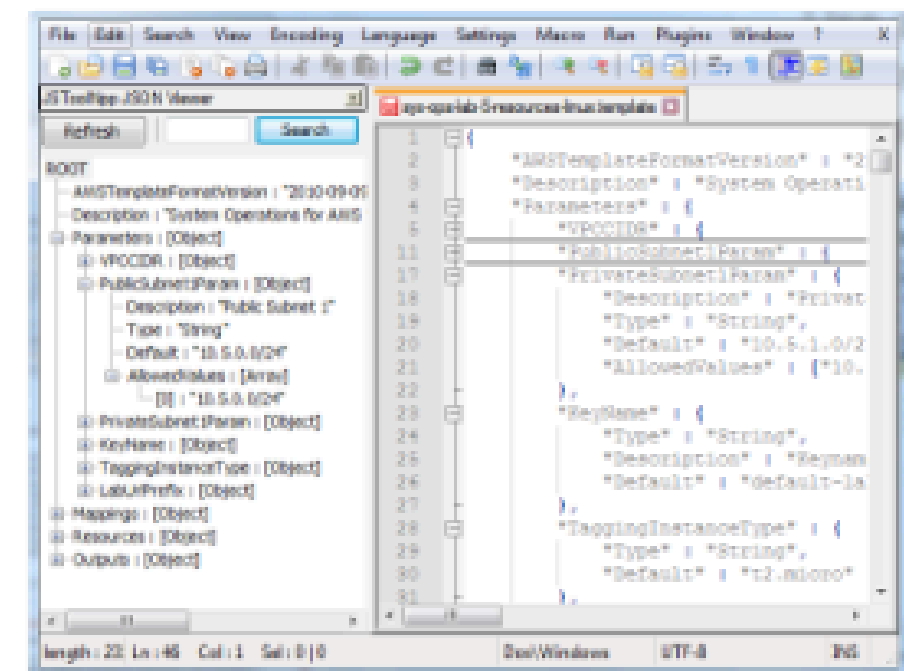
Editor de plantillas de AWS CloudFormation para Visual Studio o Eclipse



Herramientas de terceros



Editor de texto con complementos de JavaScript o JSON

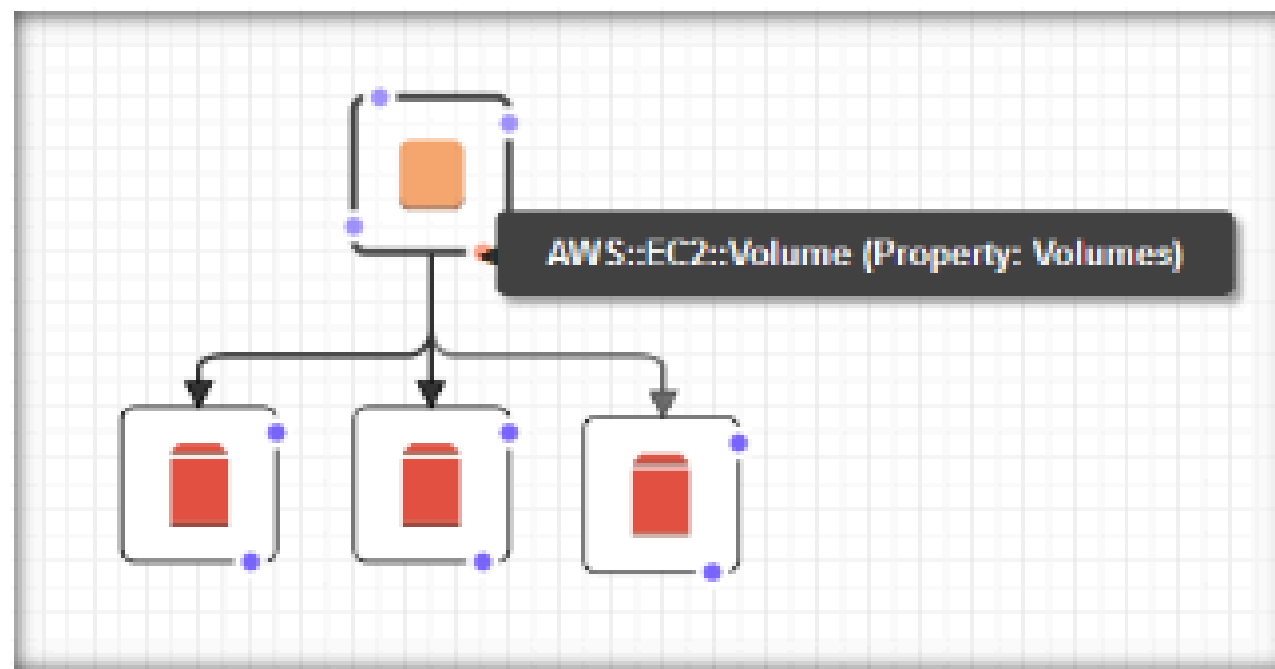


Diseñar plantillas

AWS CloudFormation Designer es una herramienta visual que permite crear y modificar plantillas de AWS CloudFormation mediante una interfaz en la que se puede arrastrar y soltar

AWS CloudFormation Designer

Superficie de diseño



Editor de JSON o YAML

```
Properties Metadata CreationPolicy DeletionPolicy DependsOn
HNO6Y
1 {
2   "Resources": {
3     "Instance": {
4       "Type": "AWS::EC2::Instance",
5       "Properties": {
6         "InstanceType": "t2.micro",
7         "DisableApiTermination": true,
8         "Monitoring": true,
9       "Volumes": [
10        {
11          "VolumeId": {
12            "Ref": "vol1"
13          }
14        }
15      ]
16    }
17  }
18 }
```

Herramientas de terceros



Lanzar y eliminar pilas en AWS CloudFormation

Puedes lanzar una plantilla como pila a través de:

- **Consola de administración de AWS**
- **AWS CLI**
- **API de AWS**

Errores y restauración de recursos

- **Si hay un error al lanzar una plantilla, los recursos se restauran automáticamente (rollback).**
- **Al eliminar una pila, se eliminan los recursos creados, salvo que se haya habilitado protección de terminación.**

Definir parámetros en una plantilla

- Utilice la sección opcional Parameters (Parámetros) para personalizar sus plantillas.
- Los parámetros permiten ingresar valores personalizados a la plantilla cada vez que crea o actualiza una pila

```
"Parameters": {  
  "VPCCIDR": {  
    "Description": "CIDR Block for VPC",  
    "Type": "String",  
    "Default": "10.200.0.0/20",  
    "AllowedValues": [ "10.200.0.0/20" ]  
  }  
}
```

Nombre lógico (único dentro de la plantilla)

Se debe especificar un tipo de datos admitido (*String*, *Number*, *List*, *CommaDelimitedList*, un parámetro específico de AWS o un parámetro de AWS Systems Manager)

Referenciar un parámetro

- Utilice la función intrínseca *Ref* para referenciar un parámetro designado.
- Se convierte en una cadena, independientemente del tipo en la declaración *Parameter*.

Parámetro

```
"KeyName": {"Type": "AWS::EC2::KeyPair::KeyName"}
```

Referenciar
un parámetro

```
"KeyName": {"Ref": "KeyName"},
```

Utilice la función `Fn::Select` para seleccionar valores en una lista delimitada por comas.

```
"AvailabilityZone" : { "Fn::Select" : [ "0", { "Ref" : "AvailableAZs" } ] }
```

Ref y otras funciones intrínsecas

Permite referenciar componentes definidos en una plantilla de AWS CloudFormation. Es necesaria para lo siguiente:

- Hacer referencia a parámetros
- Usar mapas
- Unir cadenas
- Usar otras funciones

```
"MyEIP" : {  
  "Type" : "AWS::EC2::EIP",  
  "Properties" : {  
    "InstanceId" : { "Ref" : "MyEC2Instance"  
  }  
}
```

Pseudoparámetros

Puede usar la función intrínseca *Ref* para acceder a información acerca de su entorno en tiempo de ejecución.

Ejemplos: región, nombre de pila, identificador de la cuenta de AWS

Los pseudoparámetros se definen previamente: no es necesario especificarlos en la sección Parameters (Parámetros) de la plantilla.

```
"Outputs" {  
  "MyStacksRegion" : { "Value" : { "Ref" : "AWS::Region" } }  
}
```

Definir mapeos en una plantilla

Para definir mapeos en la plantilla:

Defina tablas de búsqueda de pares de nombre

- valor en un mapa de dos niveles.
- Combine varias tablas de mapeo para crear búsquedas anidadas.
- Se utiliza con mayor frecuencia para buscar los valores actuales de Amazon Machine Image (AMI).

Ejemplo de mapeo

```
"Mappings" : {  
  "AWSRegionToAMI" : {  
    "us-east-1" : { "AMI" : "ami-76817c1e" },  
    "us-west-2" : { "AMI" : "ami-d13845e1" },  
    "us-west-1" : { "AMI" : "ami-f0d3d4b5" },  
    "eu-west-1" : { "AMI" : "ami-892fe1fe" }, ...  
  }  
}
```

```
"ImageId" : {  
  "Fn::FindInMap" : [  
    "AWSRegionToAMI",  
    { "Ref" : "AWS::Region" },  
    "AMI"  
  ]  
}
```

Utilice la función intrínseca
Fn::FindInMap para las
búsquedas.

Definir recursos en una plantilla

El tipo de recurso que está declarando

Puede exigir la creación de un recurso específico después de otro

Propiedades para un recurso (opcional)

```
"Resources" : {  
  "MyRDSInstance" : {  
    "Type" : "AWS::RDS::DBInstance",  
    "Properties" : {  
      "AllocatedStorage" : "5",  
      ...  
    }  
  },  
  "WebServer" : {  
    "Type" : "AWS::EC2::Instance",  
    "DependsOn" : [ "MyRDSInstance" ],  
    "Properties" : {
```

CloudFormation::Init


- Es un tipo de recurso que brinda acceso a metadatos de instancias EC2.
- Puede usarlo con el script auxiliar *cfn-init*.
- **Script auxiliar** *cfn-init*:
 - Lee los metadatos de la clave `AWS::CloudFormation::Init` en la plantilla.
 - Con esta información, *cfn-init* puede llevar a cabo las siguientes acciones en las instancias EC2:
 - » Instalar paquetes
 - » Administrar grupos y cuentas de usuario
 - » Escribir archivos en el disco
 - » Ejecutar comandos
 - » Habilitar o desactivar servicios
 - » Iniciar o detener los servicios

Datos del usuario frente a CloudFormation::Init

Datos de usuario en AWS CloudFormation:

- Archivo de script, como con las instancias individuales.
- Ofrece mayor control.
- Ofrece mayor potencial de error. Se debe tener más cuidado.

CloudFormation::Init:

- Se puede restaurar automáticamente en caso de error.
 - Es más fácil de administrar que los campos de datos de usuario en plantillas de CloudFormation.
 - Se pueden adjuntar metadatos a la plantilla.
 - Los Configsets permiten realizar varias configuraciones en un único CloudFormation::Init.
- 

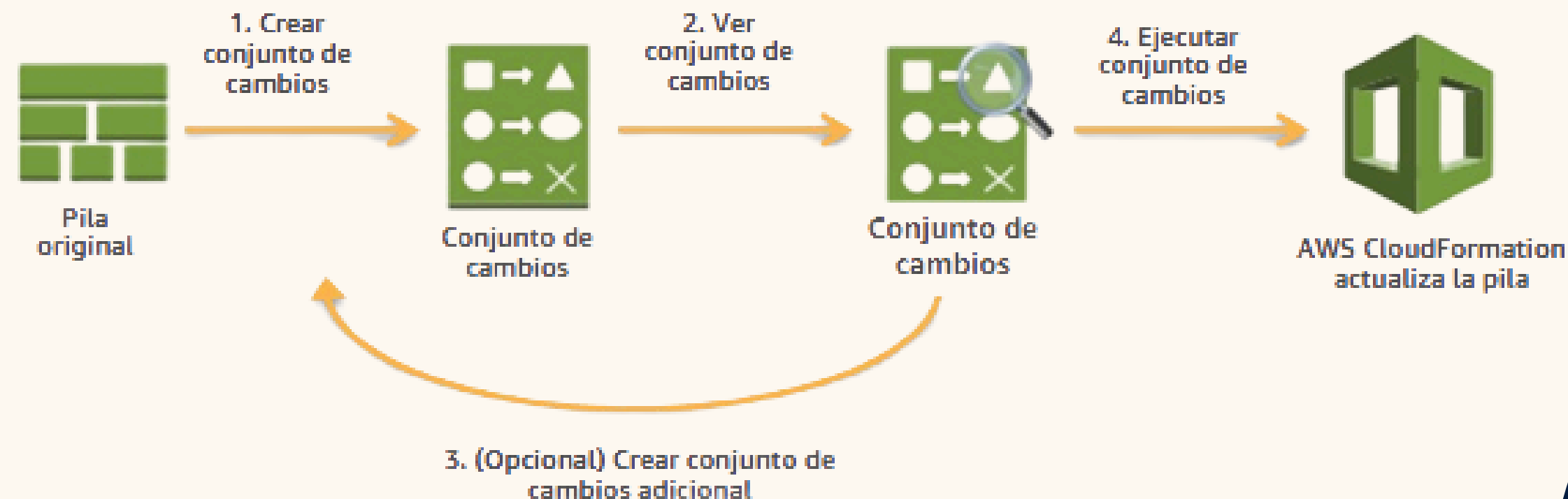
WaitCondition y WaitConditionHandle

`waitCondition` y `waitConditionHandle` se utilizan cuando se realiza el arranque de instancias. Además:

- `waitCondition` bloquea el estado de finalización de una pila hasta que se llame a `waitConditionHandle` o hasta que se alcance el límite de tiempo de espera.
 - Se puede especificar la cantidad de señales correctas que se deben recibir para que se cumpla `waitCondition`
 - Predeterminado: 1
- Puede usar el comando `cfn-signal` para indicar el éxito o el error de la señal.

Conjuntos de cambios

- Los conjuntos de cambios muestran los cambios propuestos en una pila antes de cualquier implementación
- AWS CloudFormation compara los cambios enviados con la pila.
- Los ingenieros pueden ver los cambios para ver qué recursos se agregarán, modificarán o eliminarán.
- El conjunto de cambios se puede aplicar a la pila para implementar los cambios.



Prácticas Recomendadas para Gestionar Pilas en CloudFormation

1. Organización de Pilas por Ciclo de Vida y Propietario

Descripción: Organiza las pilas por su ciclo de vida y asigna propietarios.

Beneficios:

- Responsabilidad: Claridad en las responsabilidades.
- Eficiencia: Identificación rápida de problemas y gestión de cambios.

2. Reutilización de Plantillas para Replicar Pilas en Varios Entornos

Descripción: Usa plantillas reutilizables para diferentes entornos (desarrollo, prueba, producción).

Beneficios:

- Consistencia: Configuraciones coherentes entre entornos.
- Rapidez: Acelera despliegues sin crear nuevas plantillas.

3. Verificación de los Límites de Todos los Tipos de Recursos

Descripción: Verifica los límites de recursos antes de realizar actualizaciones.

Beneficios:

- Prevención de Errores: Evita fallos por superación de límites.
 - Planificación Estratégica: Mejora la planificación de escalabilidad.
- 

Beneficios de las Mejores Prácticas en AWS CloudFormation

Consistencia

- Centraliza la gestión para evitar desviaciones en la configuración.
- Facilita la gestión de versiones y actualizaciones.

Automatización

- Minimiza los errores humanos.
- Mejora la eficiencia operativa.

Prevención de Errores

- Revisa y valida los cambios antes de aplicarlos.
- Identifica problemas antes de que afecten la producción.

Transparencia

- Proporciona claridad sobre las modificaciones propuestas.
 - Permite entender el impacto de las actualizaciones.
- 



Beneficios de las Mejores Prácticas en AWS CloudFormation

Control de Acceso

- Solo usuarios autorizados pueden realizar cambios.
- Mejora la seguridad y el cumplimiento.

Auditoría Completa

- Registra todas las acciones para la revisión y detección de actividades anómalas.
- Soporta requisitos de cumplimiento y forenses.

Calidad del Código

- Mejora la calidad de las plantillas con revisiones y pruebas automáticas.
- Reduce errores de configuración.

Colaboración y Mejora Continua

- Fomenta el intercambio de conocimientos entre los equipos.
- Facilita la mejora de procesos y prácticas.



En resumen

- CloudFormation te permite definir tu infraestructura como código.
- Mappings se usan para guardar valores estáticos.
- Parameters permiten personalizar la plantilla.
- Ref y FindInMap se usan para buscar valores dentro de la plantilla.
- Resources define la infraestructura a crear.
- cfn-init ayuda a configurar instancias EC2.
- WaitCondition permite esperar a que ciertos procesos se completen.
- Outputs devuelve valores útiles como URLs o IDs de recursos.
- AWS CloudFormation Designer permite diseñar visualmente la plantilla.



Documentación



