Arquitecturas de nube con AWS

Ing. Fernando Lichtschein
Ing. Mora Villa Abrille

11. Automatización de infraestructura

UNIVERSIDAD AUSTRAL

Objetivos

Reconocer cuándo y por qué se utiliza la automatización de arquitectura.

Identificar cómo se usa la infraestructura como código (IaC) para crear y administrar recursos de nube.

Identificar cómo modelar, crear y gestionar una colección de recursos usando AWS CloudFormation.

Comprender cómo usar *templates* de AWS Quick Start CloudFormation para definir una arquitectura.

Reconocer los usos de Amazon Q Developer.

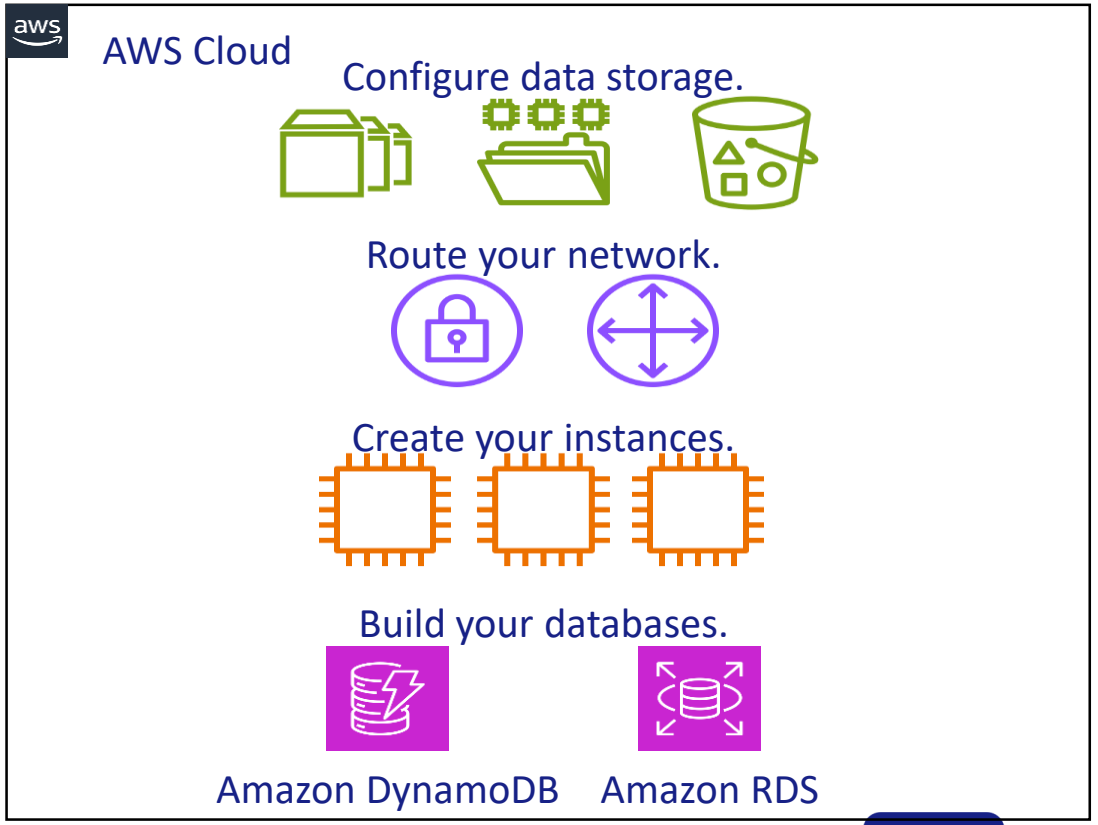
Aplicar los principios del AWS Well-Architected Framework al diseño de estrategias de automatización.

Automatización de la arquitectura

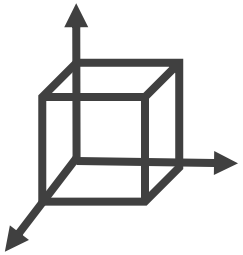
UNIVERSIDAD AUSTRAL

Sin automatización

La construcción de la arquitectura es un proceso largo y manual



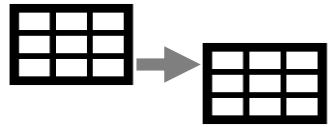
Riesgos del proceso manual



No es repetible a escala

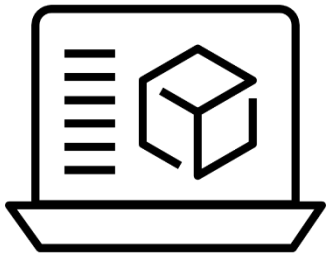
No tiene control de versiones

No tiene registro de auditoría

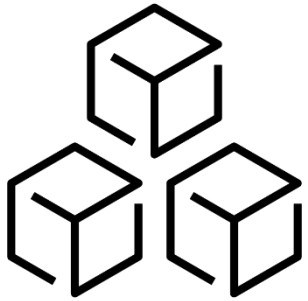


No asegura la consistencia de las configuraciones

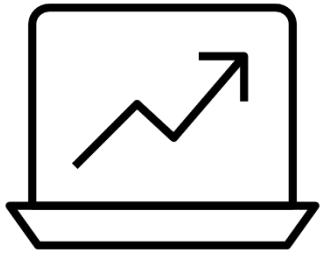
Beneficios de la automatización



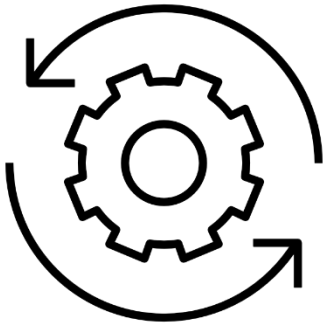
Reduce el acceso y la intervención manual



Permite crear entornos reproducibles



Aumenta la productividad



Automatiza las pruebas y el escalamiento

Resumen

Los procesos manuales están sujetos a error y son inadecuados para soportar un negocio ágil.

Los procesos manuales crean riesgos para las aplicaciones y los entornos.

La automatización ayuda a eliminar los procesos manuales y construir con rapidez.

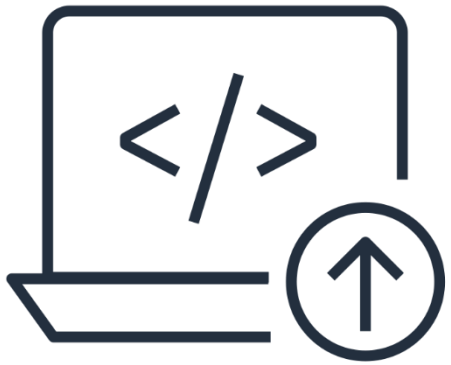
Infraestructura como código

UNIVERSIDAD AUSTRAL

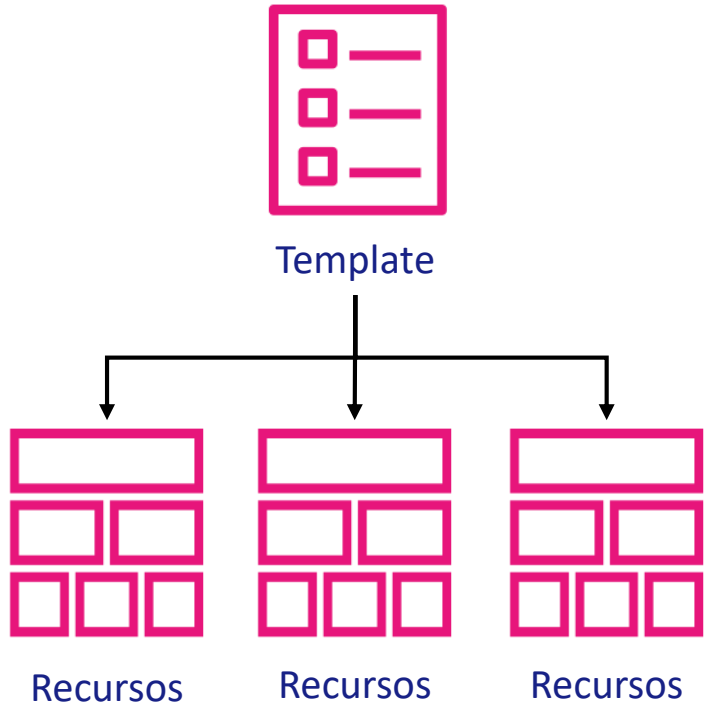
IaC

IaC es el proceso de escribir un template que crea y mantiene recursos de nube.

Legible para las personas



Consumible por una máquina



Con IaC, es posible replicar, volver a implementar y reutilizar infraestructura.

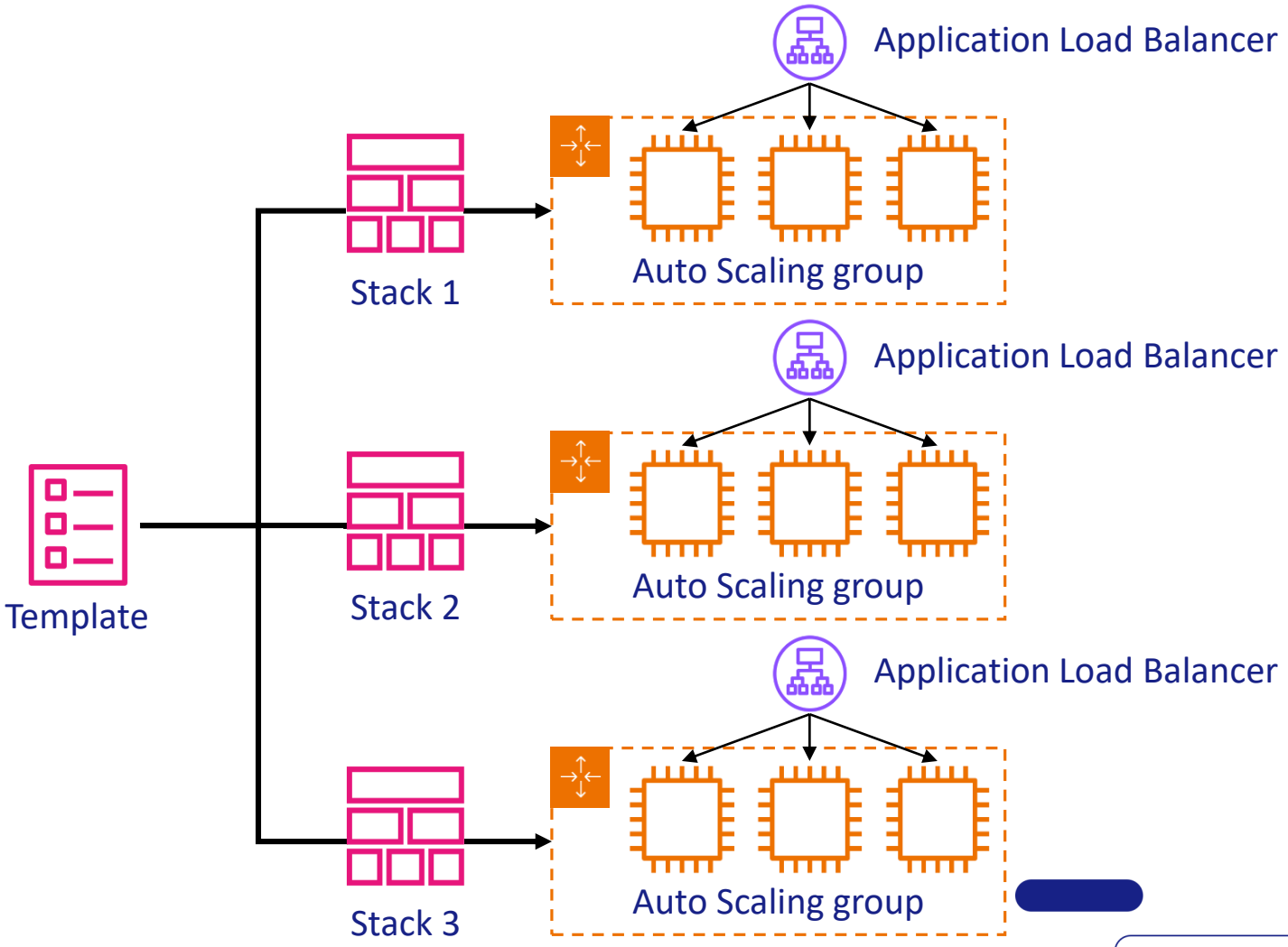
Beneficios de IaC

Velocidad de implementación y consistencia.

Propagación de cambios a todo el *stack,* modificando solo el template

Limpieza: al eliminar el *stack*, se borran todos los recursos creados.

Reusable, repetible y más fácil de mantener.



CloudFormation

CloudFormation

Es una forma simplificada de modelar, crear y administrar recursos de AWS.

- Una colección de recursos se llama un *stack* de CloudFormation
- Se pagan los recursos que se usan, no el empleo de CloudFormation

Permite crear, actualizar y eliminar stacks.

Permite la creación y actualización ordenadas y predecibles de recursos.

Brinda un control de versiones de las implementaciones de recursos de AWS.

Servicios de AWS que usan CloudFormation

UNIVERSIDAD AUSTRAL

AWS Elastic Beanstalk

Servicio gestionado que crea automáticamente un entorno de AWS con el código de la aplicacíon del cliente.

AWS Quick Starts

Arquitecturas de referencia automáticas, basadas en templates de CloudFormation

AWS Serverless Application

Extensión de CloudFormation con una sintaxis simplificada para construir infraestructura *serverless* común

AWS Amplify

Framework de desarrollo para aplicaciones web y mobile que requiere menos código y menos conocimiento de las integraciones con el backend

AWS Cloud Development Kit

Framework de código abierto para crear recursos de CloudFormation usando lenguajes de programación familiares

AWS CloudFormation

Resumen

IaC es el proceso de crear y administrar recursos de nube mediante archivos de template.

Permite implementar entornos complejos rápidamente, y volver a construirlos o actualizarlos repetidamente.

La solución de IaC más apropiada se selecciona en función de: el caso de uso, el balance adecuado entre conveniencia y control, y las habilidades del equipo de trabajo.

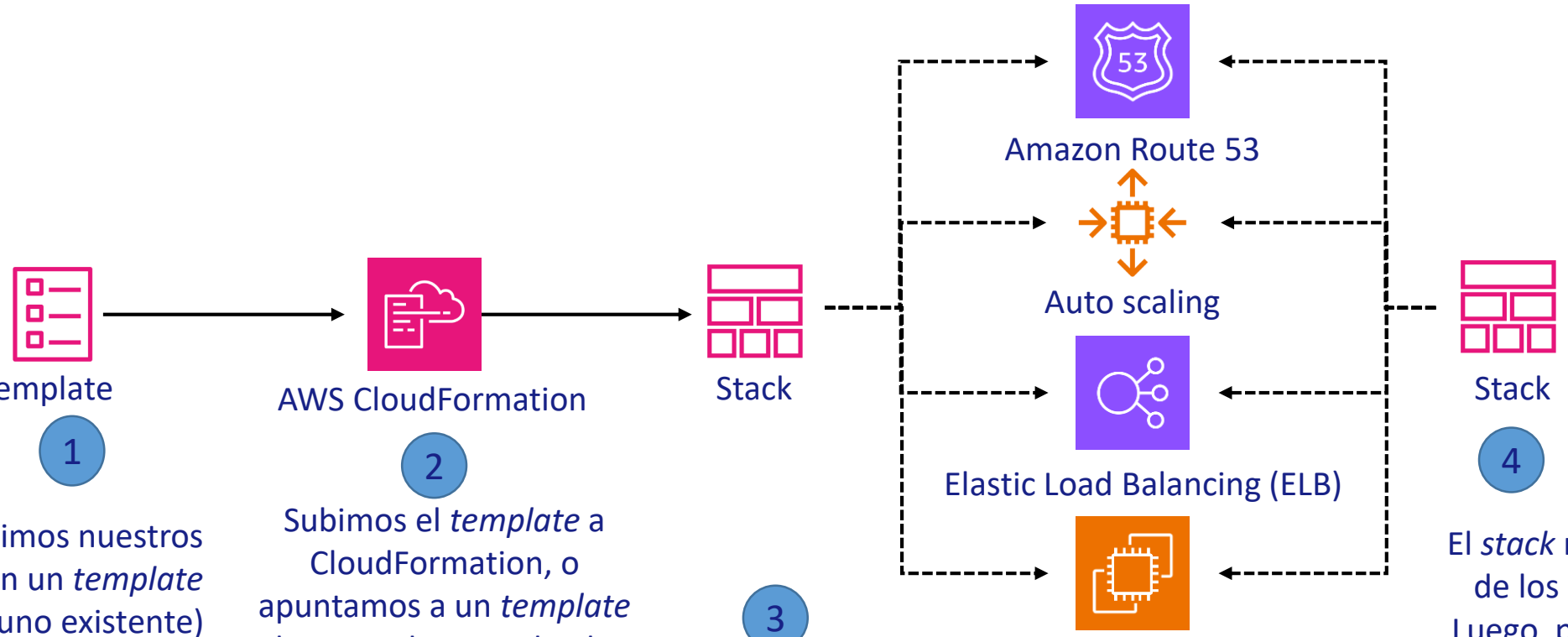
CloudFormation es un servicio de IaC que permite crear, actualizar y eliminar servicios y arquitecturas.

Personalización con CloudFormation

UNIVERSIDAD AUSTRAL

CloudFormation

¿Cómo funciona?



Definimos nuestros recursos en un *template* (o usamos uno existente)

Subimos el *template* a CloudFormation, o apuntamos a un *template* almacenado en un bucket de S3

Ejecutamos la acción **create stack**. Los recursos se crean en distintos servicios de nuestra cuenta de AWS-

El *stack* mantiene el control de los recursos creados. Luego, podemos **actualizar *el stack*, detectar desvíos o borrar el *stack*.**

CloudFormation

Templates

Template

YAML example

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

JSON example

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

Ventajas de YAML

Está optimizado para que sea legible
Menos texto
Permite incluir comentarios

JSON advantages

Uso más difundido en otros sistemas (por ejemplo, API)
Suele ser menos complejo de generar y analizar

CloudFormation

Templates

Un *template* puede incluir distintas secciones, que se definen en función de las cargas de trabajo que queremos crear.

La única sección obligatoria es **Resources**. El resto son opcionales.

El ejemplo muestra un fragmento de un template en YAML con el orden sugerido de las secciones.

```
---
AWSTemplateFormatVersion: "version date"
Description:
        String
Metadata:
        template metadata
Parameters:
        set of parameters
Rules:
        set of rules
Mappings:
        set of mappings
Conditions:
        set of conditions
Transform:
        set of transforms
Resources:
        set of resources
Outputs:
        set of outputs
```

CloudFormation

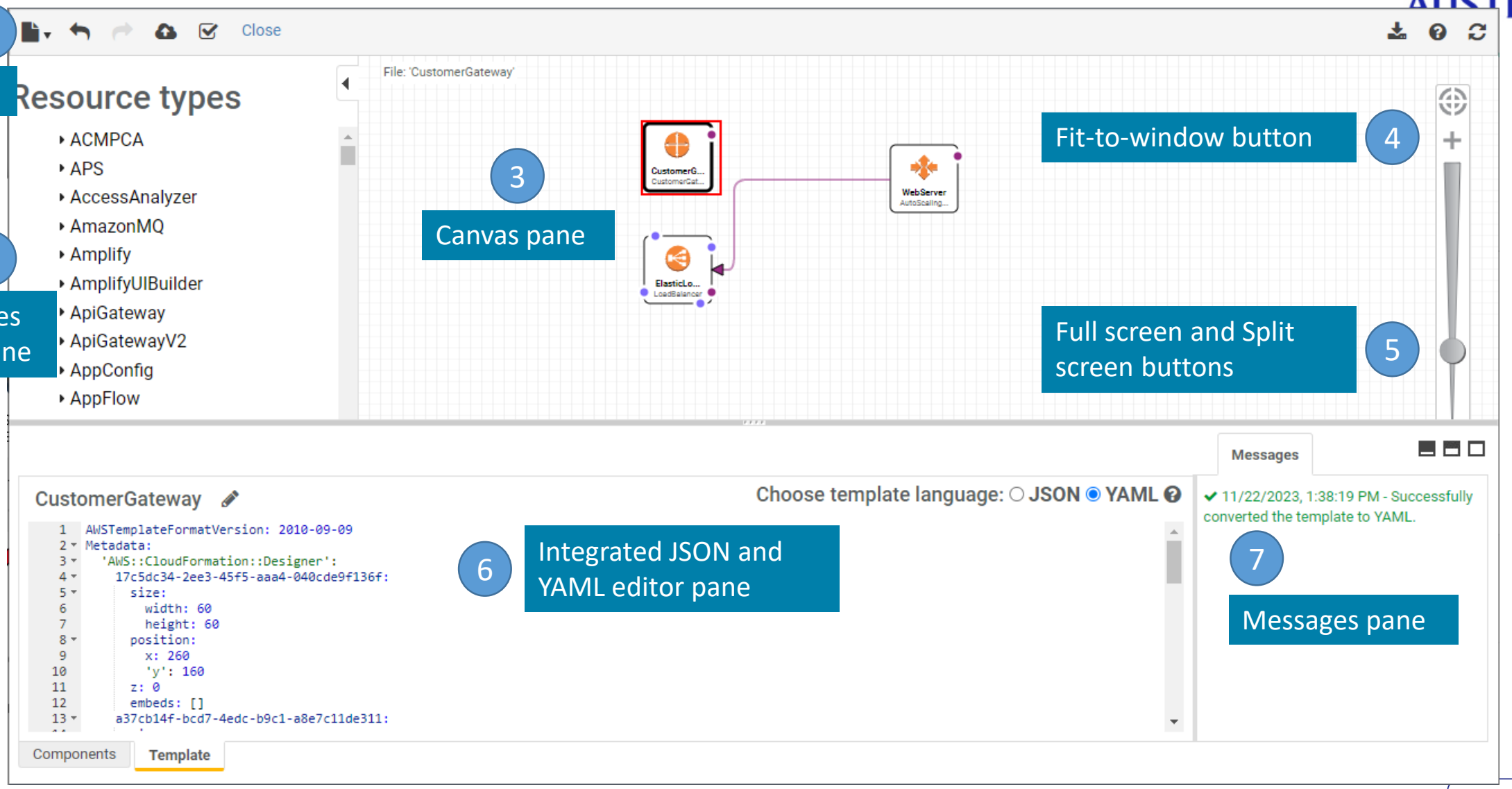
Ejemplo de un template

```
{
                "Resources": {
        "Ec2Instance": {
        "Type": "AWS::EC2::Instance",
        "Properties": {
        "ImageId": "ami-9d23aeea",
        "InstanceType": "m3.medium",
        "KeyName": {"Ref": "KeyPair}
                }},
                "Outputs": {
        "InstanceId": {
        "Description": "InstanceId",
        "Value": {"Ref": "Ec2Instance"}
        }
        }
}
```

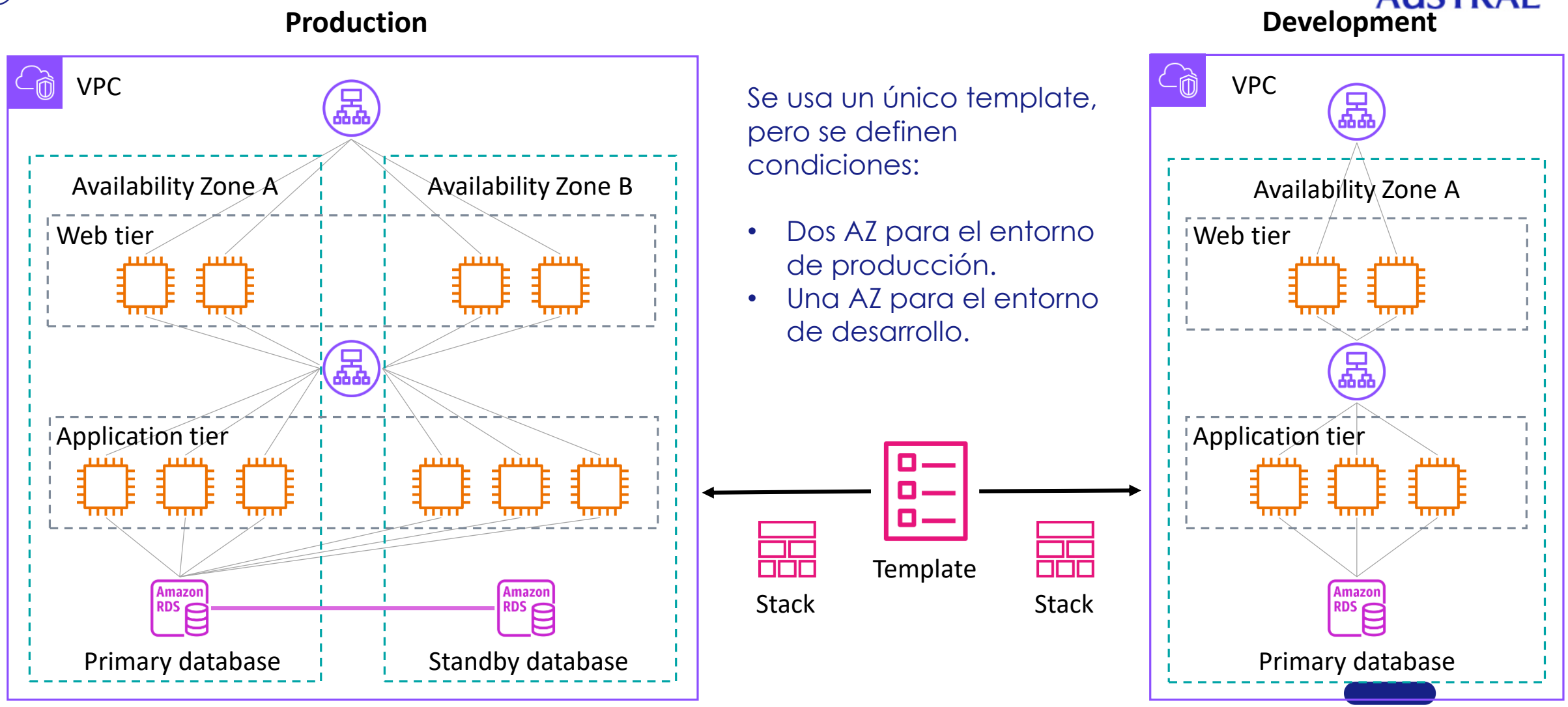
Resources define lo que necesitamos crear en una cuenta de AWS

Outputs especifica los valores que recibiremos cuando finalice la creación del *stack*.

AWS CloudFormation Designer



Condiciones en CloudFormation

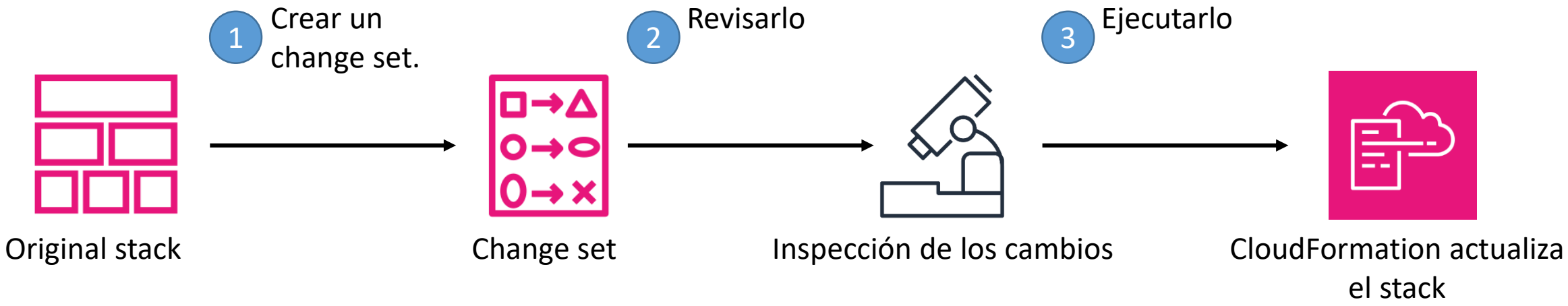


Se usa un único template, pero se definen condiciones:

- Dos AZ para el entorno de producción.
- Una AZ para el entorno de desarrollo.

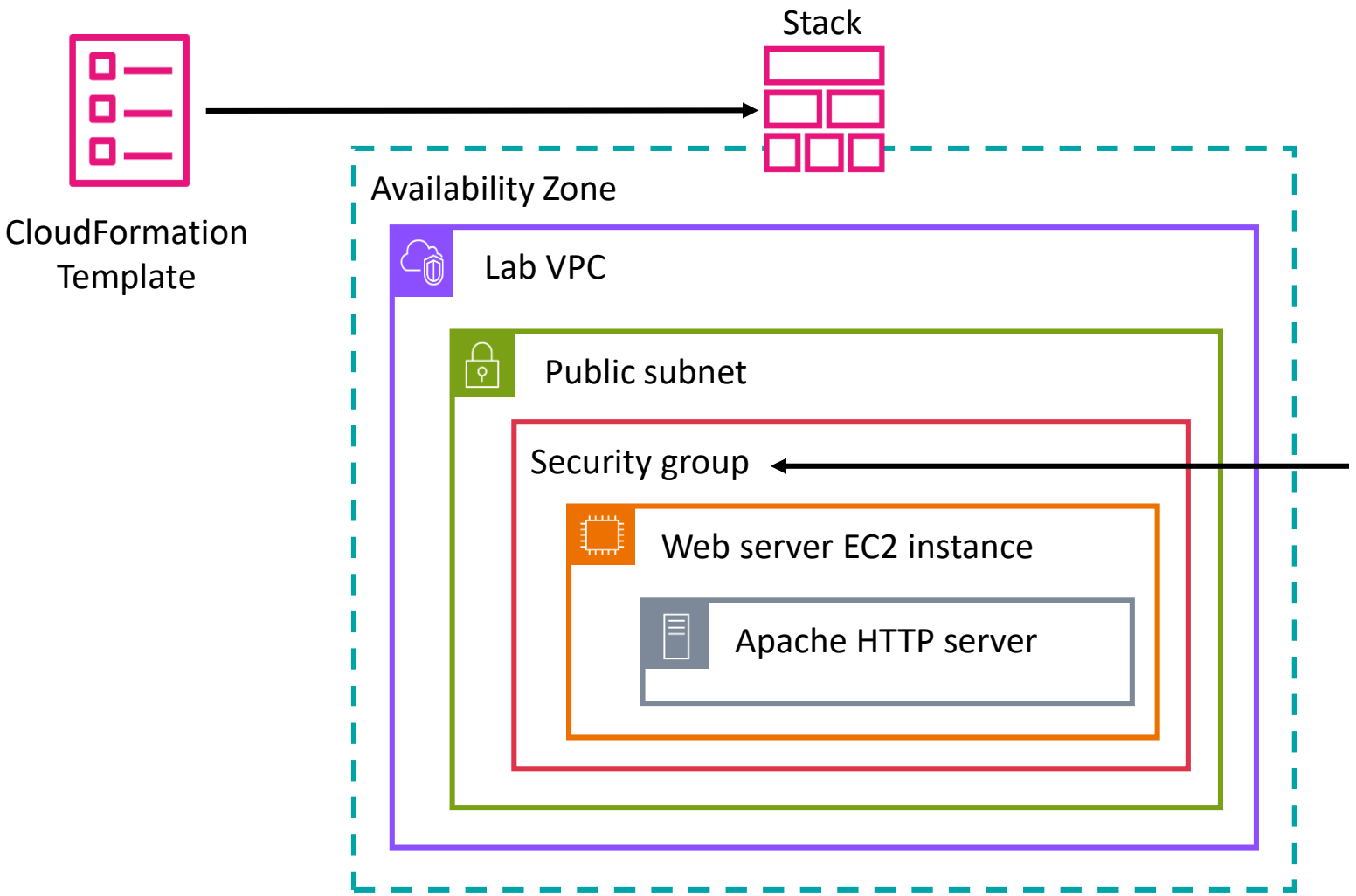
Conjuntos de cambios en CloudFormation

Se pueden usar *change sets* para hacer una vista previa antes de la implementación de cambios.



El atributo **DeletionPolicy** se puede usar para preservar o resguardar un recurso cuando se borra o actualiza su *stack*

Detección de desvíos en CloudFormation



Escenario

1. Un entorno de aplicación se crea con un *stack* de CloudFormation.
2. Luego, alguien **modifica manualmente el security group** *y abre un puerto entrante*.
3. Se ejecuta la detección de desvíos en el *stack*.
4. Todos los recursos, excepto el *security group* muestran el estado IN_SYNC, pero el *security group* muestra un estado MODIFIED con detalles.

Alcance y organización de *templates*

Categoría	Tipo de template
Frontend services	Interfaces web, acceso mobile y dashboard de análitica.
Backend services	Búsqueda, pagos, revisions y recomendaciones
Shared services	Bases de datos CRM, monitoreo, alarmas, subredes y *security groups*
Network	VPCs, internet gateways, virtual private networks (VPNs), and NAT devices
Security	AWS Identity and Access Management (IAM) policies, users, groups, and roles

CloudFormation

Resumen

Es un servicio de IaC que se puede usar para modelar, crear y gestionar un conjunto de recursos de AWS

En CloudFormation, la IaC se define en *templates* creados en JSON o YAML.

Cuando se crean recursos de AWS a partir de un *template,* forman un *stack*

Las acciones permitidas sobre un *stack* existente incluyen: actualización, detección de desvíos y eliminación del *stack*.

AWS Quick Start

Automatización de arquitectura

UNIVERSIDAD AUSTRAL

AWS Quick Starts

AWS Quick Starts

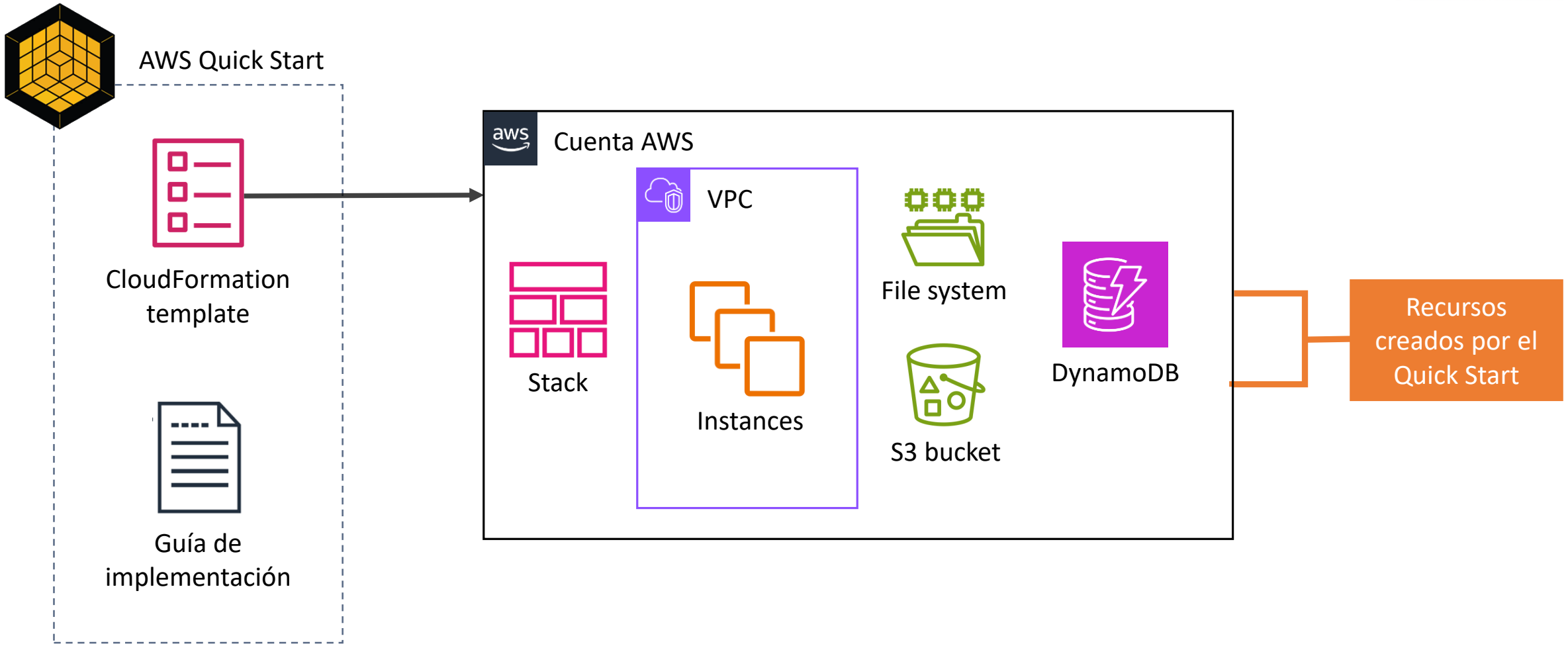
Implementaciones *gold-standard*

Se basan en las buenas prácticas de AWS sobre seguridad y alta disponibilidad.

Se pueden utilizar para crear arquitecturas completas en menos de una hora

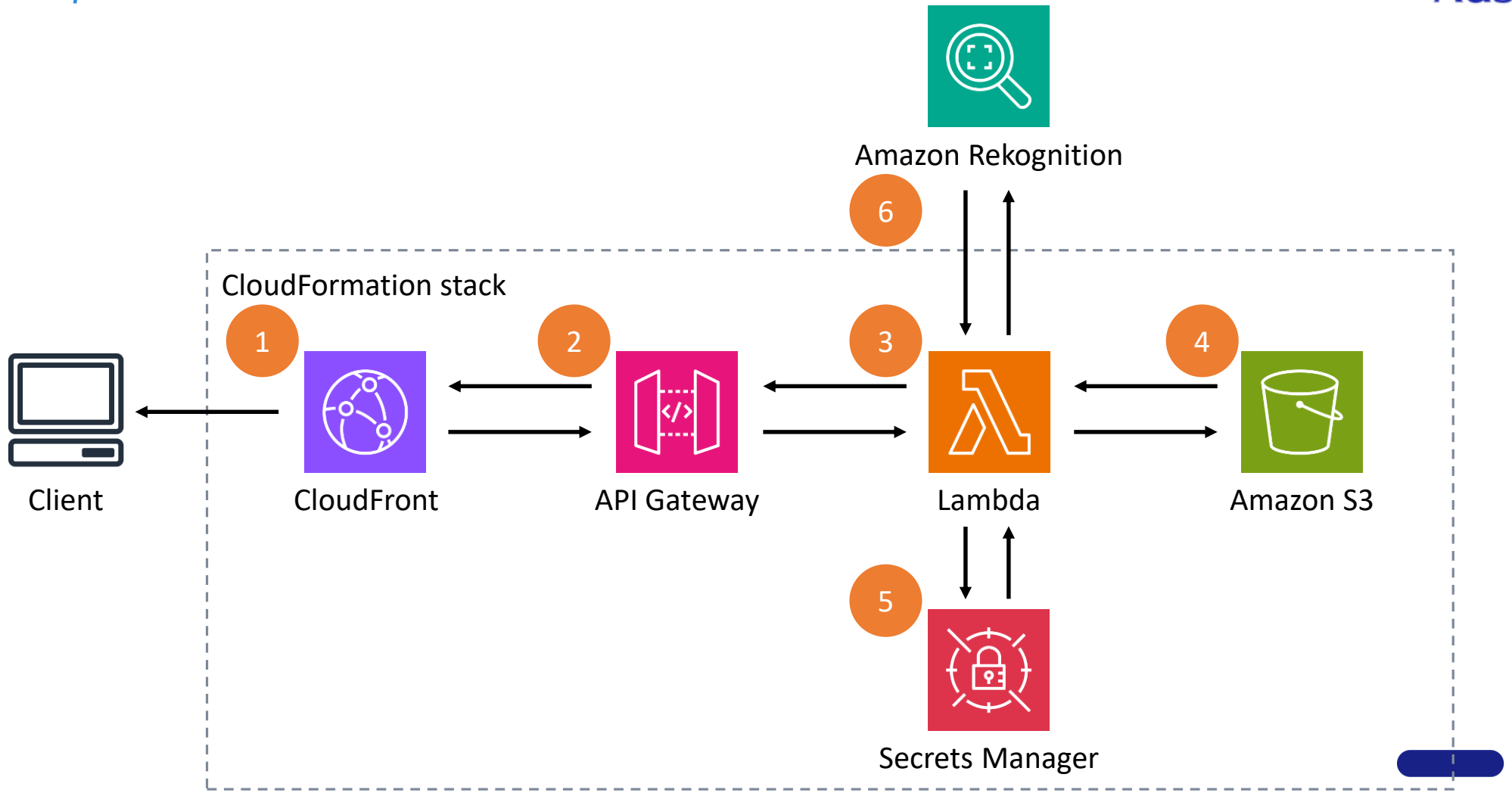
Se pueden usar para experimentar, como base para nuestra propia arquitectura.

AWS Quick Starts



AWS Quick Starts

Ejemplo



AWS QuickStarts

Resumen

AWS Quick Starts contiene templates de CloudFormation construidos por arquitectos de solución y partners que reflejan las buenas prácticas de AWS.

AWS Quick Starts está conformado por un template y una guía de implementación, que provee detalles sobre las opciones de despliegue y las configuraciones más adecuadas.

También sirve para ver patrones y prácticas que sirvan para acelerar el desarrollo de templates propios.

Amazon Q Developer

Personalización

UNIVERSIDAD AUSTRAL

Infraestructura como código

Desafíos

Error humano

Habilidades dispares

Tamaño y complejidad de los templates

Vulnerabilidades de seguridad

Amazon Q Developer

Amazon Q Developer

- Asistente de código asistido por IA generativa
- Diseñado para desarrolladores y profesionales de IT
- Genera código y ayuda a entender, construir, extender y operar aplicaciones en AWS
- Escanea el código para detector vulnerabilidades de seguridad
- Seguro y privado desde el diseño

Amazon Q Developer

Apoyo al desarrollo de aplicaciones





Plan	Create	Test and secure	Operate	Maintain and modernize
Formular preguntas para obtener una guía referenciable y contextualizada. Explicar el código con codificación conversacional	Recibir recomendaciones en varios idiomas. Implementar funciones a través de comentarios o prompts. Conversar con el IDE.	Generar tests unitarios Escanear el código para detector vulnerabilidades y recibir sugerencias de remediación	Detectar y corregir errores Verificar la conectividad de red con VPC Reachability Analyzer.	Modernizar el código con Amazon Q Developer Agent.

Ejemplo

Uso de Amazon Q Developer con CloudFormation

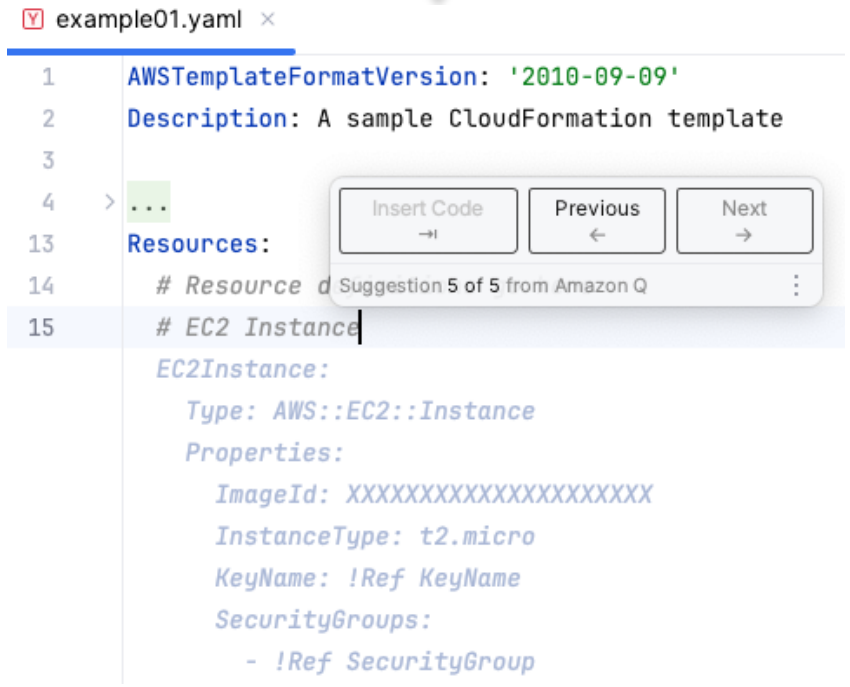
example01.yaml

```yaml
AWSTemplateFormatVersion: '2010-09-09'
Description: A sample CloudFormation template

...
Resources:
  # Resource definitions go here
  # EC2 Instance
Outputs:
  # Output values go here
```

El desarrollador empieza a escribir el nombre del recurso

Amazon Q Developer hace sugerencias que el desarrollador puede aceptar



El código se agrega al template

example01.yaml

```yaml
AWSTemplateFormatVersion: '2010-09-09'
Description: A sample CloudFormation template

...
Resources:
  # Resource definitions go here
  # EC2 Instance
  EC2Instance:
    Type: AWS::EC2::Instance
    Properties:
      ImageId: XXXXXXXXXXXXXXXXXXXX
      InstanceType: t2.micro
      KeyName: !Ref KeyName
      SecurityGroups:
        - !Ref SecurityGroup
```

Automatización

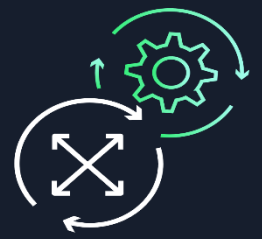
Pilares del Well-Architected Framework

UNIVERSIDAD AUSTRAL

Well-Architected Framework

Buenas prácticas para la automatización

UNIVERSIDAD AUSTRAL



Operational Excellence

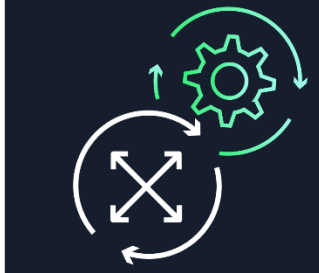
Security

Reliability

Cost Optimization

Well-Architected Framework

Prepararse y diseñar para la operación



Operational Excellence

Buenas prácticas

Realizar las operaciones como código

Hacer cambios frecuentes, pequeños y reversibles.

Automatizar completamente la integración y la implementación.

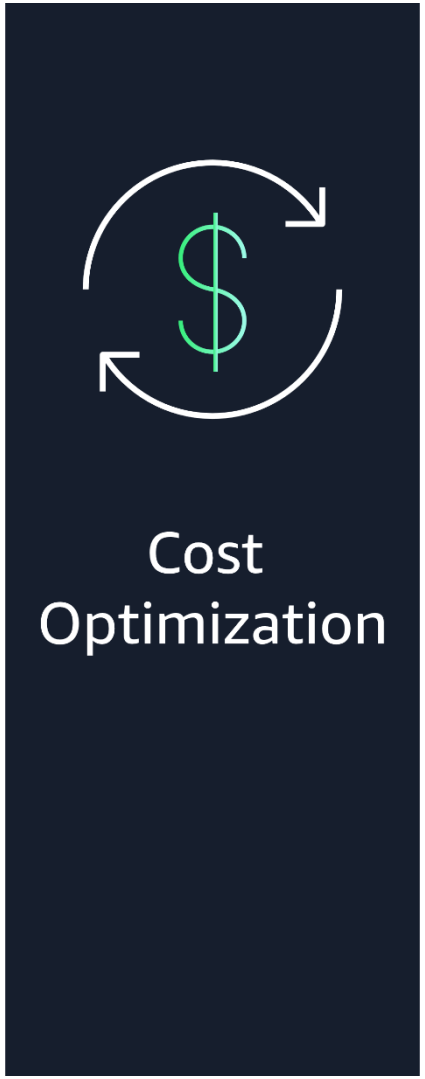
Well-Architected Framework

Seguridad

Automatizar buenas prácticas de seguridad

Well-Architected Framework

Optimización en el tiempo



Buena práctica

Aplicar automatización a las operaciones

Well-Architected Framework

Resumen

Las buenas prácticas relacionadas con la automatización de la arquitectura incluyen:

- Ejecutar operaciones como código
- Hacer cambios frecuentes, pequeños y reversibles
- Automatizar completamente la integración y la implementación
- Automatizar las buenas prácticas de seguridad
- Implementar los cambios mediante automatizaciones
- Usar automatización para crear y escalar recursos

Módulo 11

Pregunta de práctica

Consider a situation where you want to create a single AWS CloudFormation template that is capable of creating both a production environment that spans two Availability Zones and a development environment that exists in a single Availability Zone. Which optional section of the CloudFormation template will you want to make use of to configure the logic that will support this?

Identifiquemos las palabras o frases clave:

The following are the key words and phrases:

- A single AWS CloudFormation template
- An environment with two Availability Zones and an environment with a single Availability Zone
- Optional section

Módulo 11

Pregunta de práctica

Consider a situation where you want to create a single AWS CloudFormation template that is capable of creating both a production environment that spans two Availability Zones and a development environment that exists in a single Availability Zone. Which optional section of the CloudFormation template will you want to make use of to configure the logic that will support this?

Choice	Response
A	Conditions
B	Outputs
C	Resources
D	Description

Módulo 11

Pregunta de práctica

Consider a situation where you want to create a single AWS CloudFormation template that is capable of creating both a production environment that spans two Availability Zones and a development environment that exists in a single Availability Zone. Which optional section of the CloudFormation template will you want to make use of to configure the logic that will support this?

Choice	Response
A	Conditions

Módulo 11

Resumen

Reconocer cuándo automatizar la arquitectura y por qué.

Identificar cómo podemos usar IaC como estrategia para administrar recursos de nube.

Entender cómo modelar, crear y administrar una colección de recursos de AWS con AWS CloudFormation.

Identificar cómo usar AWS Quick Start para definir una arquitectura.

Reconocer los usos de Amazon Q Developer.

Aplicar los principios del AWS Well-Architected Framework a la automatización.

Muchas gracias.

www.austral.edu.ar

UNIVERSIDAD AUSTRAL