

Documentación de IaaS con AWS CDK

Objetivo

Implementar una pila (*stack*) en AWS utilizando AWS CDK que configure los recursos necesarios para un backend con Django, un frontend alojado en S3 y una API Gateway para enrutar solicitudes al backend.

Descripción General

Este archivo define la infraestructura mediante AWS CDK en Python. Los principales recursos configurados son:

1. Una instancia EC2 para alojar el backend (Django).
2. Un bucket S3 para alojar el frontend estático.
3. Una API Gateway que enruta solicitudes al backend en EC2.

El código se encuentra estructurado en la clase `CdkStack`, que hereda de `Stack`, y organiza los recursos de AWS en un único entorno lógico.

Recursos Definidos

1. Instancia EC2 para el Backend

- **Propósito:** Hospedar el servidor Django.
 - **Componentes Configurados:**
 - **VPC:** Se define una VPC utilizando la clase `ec2.Vpc` con un máximo de 3 zonas de disponibilidad (`max_azs=3`), asegurando alta disponibilidad.
 - **Role de IAM:** Se crea un rol (`iam.Role`) que permite que la instancia EC2 utilice servicios de AWS.
 - **Instancia EC2:**
 - Tipo de instancia: `t2.micro` (adecuada para entornos pequeños o pruebas).
 - Imagen AMI: `AmazonLinuxImage()` para obtener un sistema base compatible.
 - Conexión a la VPC: La instancia utiliza la VPC definida.
-

2. Bucket S3 para el Frontend

- **Propósito:** Almacenar y servir los archivos estáticos del frontend.
 - **Configuraciones:**
 - Habilitado como sitio web estático (`website_index_document` y `website_error_document`).
 - Los documentos principales se llaman `index.html` y `error.html` .
-

3. API Gateway

- **Propósito:** Exponer el backend Django mediante un endpoint HTTP.
- **Configuraciones:**
 - Nombre: `Django Service` .
 - Endpoint principal: `/hello` .
 - Método GET: Configurado con una integración HTTP que apunta a una URL fija (`http://3.130.67.167:8000`).

Flujo de Configuración

1. Preparación de la Infraestructura:

- Se define una VPC con 3 zonas de disponibilidad para la alta disponibilidad de recursos.
- Se crea una instancia EC2 con un sistema operativo Amazon Linux y configuraciones básicas de tipo de instancia y rol IAM.

2. Configuración del Frontend:

- Un bucket S3 actúa como servidor de archivos estáticos para el frontend, utilizando `index.html` y `error.html`.

3. Enrutamiento de la API:

- Se configura una API Gateway con un endpoint `/hello`.
- Este endpoint enruta las solicitudes al backend Django, alojado en la instancia EC2 mediante una integración HTTP directa.

Comandos de Ejecución

1. Instalar Dependencias

```
bash

pip install -r requirements.txt
```

2. Sintetizar la Pila

Genera el archivo CloudFormation para la infraestructura.

```
bash

cdk synth
```

3. Desplegar la Pila

Despliega los recursos en AWS.

```
bash

cdk deploy
```

4. Destruir la Pila

Elimina los recursos creados.

```
bash

cdk destroy
```