

Implementación de Servicios en AWS con Terraform y Python

Objetivo: [🔗](#)

Desarrollar un entorno en la nube utilizando servicios de AWS, Terraform para la infraestructura y Python con la librería boto3 para la interacción con estos servicios.

Pasos: [🔗](#)

1. Crear 2 SQS con Terraform [🔗](#)

1.1. Escribe el script Terraform:

```
1 hcl

1 # main.tf
2
3 provider "aws" {
4   region = "us-east-1"
5 }
6
7 resource "aws_sqs_queue" "queue1" {
8   name                  = "queue1"
9   message_retention_seconds = 86400
10  visibility_timeout_seconds = 30
11 }
12
13 resource "aws_sqs_queue" "queue2" {
14   name                  = "queue2"
15   message_retention_seconds = 86400
16   visibility_timeout_seconds = 30
17 }
18
```

1.2. Ejecuta Terraform:

```
1 bash

1 terraform init
2 terraform apply
3
```

2. Crear manualmente un mensaje para alguna de las colas SQS [🔗](#)

2.1. Usando AWS CLI o la consola de AWS, envía un mensaje a una de las colas creadas.

2.2. Para verificar el mensaje por consola:

```
1 bash

1 awslocal sqs receive-message --queue-url <URL_DE_TU_COLA>
2
```

(Reemplaza `<URL_DE_TU_COLA>` con la URL de la cola que has usado)

3. Crear tabla DynamoDB en Terraform [↗](#)

3.1. Escribe el script Terraform:

```
1 hcl

1 # main.tf
2
3 resource "aws_dynamodb_table" "my_table" {
4     name           = "my-table"
5     billing_mode    = "PAY_PER_REQUEST"
6     hash_key       = "id"
7
8     attribute {
9         name = "id"
10        type = "S"
11    }
12 }
13
```

3.2. Ejecuta Terraform:

```
1 bash

1 terraform init
2 terraform apply
3
```

4. Crear código de Python para levantar los 3 servicios [↗](#)

4.1. Escribe el script Python (deploy_services.py):

```
1 python

1 import boto3
2 import time
3
4 dynamodb = boto3.client('dynamodb', region_name='us-east-1')
5
6 table_name = "my-table"
7 dynamodb.create_table(
8     TableName=table_name,
9     KeySchema=[{'AttributeName': 'id', 'KeyType': 'HASH'}],
10    AttributeDefinitions=[{'AttributeName': 'id', 'AttributeType': 'S'}],
11    BillingMode='PAY_PER_REQUEST'
12 )
13
14 print(f"Tabla DynamoDB '{table_name}' creada.")
15
16 sqs = boto3.client('sqs', region_name='us-east-1')
17
18 queue_names = ["queue1", "queue2"]
19 for queue_name in queue_names:
20     response = sqs.create_queue(QueueName=queue_name)
21     print(f"Cola SQS '{queue_name}' creada. URL: {response['QueueUrl']}")
22
23 ec2 = boto3.resource('ec2', region_name='us-east-1')
24 instance = ec2.create_instances(
```

```
25     ImageId='ami-12345678',
26     MinCount=1,
27     MaxCount=1,
28     InstanceType='t2.micro',
29     KeyName='your-key-pair'
30 )[0]
31
32 print(f"Instancia EC2 creada. ID: {instance.id}")
33
34 instance.wait_until_running()
35 print("La instancia EC2 está en ejecución.")
36
```

4.2. Ejecuta el script:

```
1 bash
```

```
1 python deploy_services.py
```