

Gestión de Instancias EC2 con Python

Objetivos de la Práctica

1. Lanzar varias instancias EC2
2. Listar y describir las instancias
3. Detener y borrar las instancias

```
In [ ]: import boto3
import os
```

```
In [ ]: session = boto3.Session(
    aws_access_key_id = os.getenv('AWS_ACCESS_KEY_ID'),
    aws_secret_access_key = os.getenv("AWS_SECRET_ACCESS_KEY"),
    region_name = 'us-east-1'
)

ec2_client = session.client('ec2')
```

1. Lanzar Varias Instancias EC2

```
In [ ]: # Ejecutar la solicitud para lanzar una instancia EC2
response = ec2_client.run_instances(
    ImageId = 'ami-08a0d1e16fc3f61ea',
    InstanceType = 't2.micro',
    MinCount = 1,
    MaxCount = 1
)
```

```
In [ ]: # Imprimir el ID de la instancia lanzada
instance_id = response['Instances'][0]['InstanceId']
print(f'La instancia ha sido lanzada con ID: {instance_id}')
```

La instancia ha sido lanzada con ID: i-01642a030325a32b5

```
In [ ]: # Ejecutar la solicitud para lanzar una instancia EC2
response2 = ec2_client.run_instances(
    ImageId = 'ami-0eaf7c3456e7b5b68',
    InstanceType = 't2.micro',
    MinCount = 1,
    MaxCount = 1
)
```

```
In [ ]: # Imprimir el ID de la instancia lanzada
instance_id = response2['Instances'][0]['InstanceId']
print(f'La instancia ha sido lanzada con ID: {instance_id}')
```

La instancia ha sido lanzada con ID: i-06471e09a295bfff8

Anexo - Como Encontrar IDS de las AMIs

Puedes encontrar los IDs de las AMIs (Amazon Machine Images) de varias maneras. Aquí te muestro algunos métodos comunes:

1. AWS Management Console:

- Iniciar sesión en la consola de AWS: Ve a <https://aws.amazon.com/> y haz clic en "Sign In to the Console" en la esquina superior derecha.
- Ir al servicio EC2: En el menú de servicios, selecciona EC2.

- Seleccionar "AMIs" en el menú de la izquierda: En la sección "Images", haz clic en "AMIs".
- Filtrar las AMIs: Puedes filtrar por propietario (como "Owned by me" o "Public images") y por otras características como el sistema operativo, la arquitectura y la región.
- Ver el ID de la AMI: El ID de la AMI se mostrará en la columna "AMI ID".

2. AWS CLI (Command Line Interface):

Si tienes la AWS CLI instalada y configurada, puedes usar el siguiente comando para listar las AMIs disponibles:

```
aws ec2 describe-images --owners amazon --query 'Images[*].[ImageId,Name,Description]' --output table
```

Este comando muestra una lista de AMIs públicas propiedad de Amazon. Puedes ajustar los parámetros para buscar AMIs específicas.

3. Boto3 (Python)

```
import boto3
```

```
# Crear una sesión de Boto3
```

```
session = boto3.Session(
    aws_access_key_id='YOUR_ACCESS_KEY',
    aws_secret_access_key='YOUR_SECRET_KEY',
    region_name='us-west-2' # Cambia esto según tu región
)
```

```
# Crear un cliente EC2
```

```
ec2_client = session.client('ec2')
```

```
# Describir las imágenes
```

```
response = ec2_client.describe_images(Owners=['amazon'])
```

```
# Imprimir los IDs y nombres de las AMIs
```

```
for image in response['Images']:
    print(f"AMI ID: {image['ImageId']}, Name: {image['Name']}, Description: {image.get('Description', 'No description')}")
```

Notas Importantes

- Región: Asegúrate de que la AMI que seleccionas esté disponible en la región en la que planeas lanzar la instancia.
- Propósito: Elige una AMI que cumpla con tus requisitos específicos, como el sistema operativo, las aplicaciones preinstaladas, etc.
- Propietario: Verifica si la AMI es pública, privada, o proporcionada por AWS Marketplace.

2. Listar y Describir las Instancias

```
In [ ]: # Describir las instancias
response = ec2_client.describe_instances()

# Iterar sobre las reservas y las instancias
for reservation in response['Reservations']:
    for instance in reservation['Instances']:
        print(f'ID de la instancia: {instance["InstanceId"]}')
        print(f'Estado de la instancia: {instance["State"]["Name"]}')
        print(f'Tipo de instancia: {instance["InstanceType"]}')
        print(f'Zona de disponibilidad: {instance["Placement"]["AvailabilityZone"]}')
        print(f'AMI ID: {instance["ImageId"]}')
        print('---')
```

```
ID de la instancia: i-01642a030325a32b5
Estado de la instancia: running
Tipo de instancia: t2.micro
Zona de disponibilidad: us-east-1a
AMI ID: ami-08a0d1e16fc3f61ea
---
ID de la instancia: i-06471e09a295bfff8
Estado de la instancia: running
Tipo de instancia: t2.micro
Zona de disponibilidad: us-east-1a
AMI ID: ami-0eaf7c3456e7b5b68
---
```

3. Detener y Borrar las Instancias

```
In [ ]: # Detener una instancia
response = ec2_client.stop_instances(
    InstanceIds=[
        'i-01642a030325a32b5',
    ]
)

print(f'La instancia {instance_id} se está deteniendo.')
```

La instancia i-06471e09a295bfff8 se está deteniendo.

Volvemos a listar y describir las instancias para comprobar los cambios

```
In [ ]: # Describir las instancias
response = ec2_client.describe_instances()

# Iterar sobre las reservas y las instancias
for reservation in response['Reservations']:
    for instance in reservation['Instances']:
        print(f'ID de la instancia: {instance["InstanceId"]}')
        print(f'Estado de la instancia: {instance["State"]["Name"]}')
        print(f'Tipo de instancia: {instance["InstanceType"]}')
        print(f'Zona de disponibilidad: {instance["Placement"]["AvailabilityZone"]}')
        print(f'AMI ID: {instance["ImageId"]}')
        print('---')
```

```
ID de la instancia: i-01642a030325a32b5
Estado de la instancia: stopped
Tipo de instancia: t2.micro
Zona de disponibilidad: us-east-1a
AMI ID: ami-08a0d1e16fc3f61ea
---
ID de la instancia: i-06471e09a295bfff8
Estado de la instancia: running
Tipo de instancia: t2.micro
Zona de disponibilidad: us-east-1a
AMI ID: ami-0eaf7c3456e7b5b68
---
```

Volvemos a ponerla en marcha

```
In [ ]: # Ejecutar una instancia
response = ec2_client.start_instances(
    InstanceIds = [
        "i-01642a030325a32b5"
    ]
)
```

```
In [ ]: # Describir las instancias
response = ec2_client.describe_instances()
```

```
# Iterar sobre Las reservas y Las instancias
for reservation in response['Reservations']:
    for instance in reservation['Instances']:
        print(f'ID de la instancia: {instance["InstanceId"]}')
        print(f'Estado de la instancia: {instance["State"]["Name"]}')
        print(f'Tipo de instancia: {instance["InstanceType"]}')
        print(f'Zona de disponibilidad: {instance["Placement"]["AvailabilityZone"]}')
        print(f'AMI ID: {instance["ImageId"]}')
        print('---')
```

ID de la instancia: i-01642a030325a32b5

Estado de la instancia: running

Tipo de instancia: t2.micro

Zona de disponibilidad: us-east-1a

AMI ID: ami-08a0d1e16fc3f61ea

ID de la instancia: i-06471e09a295bfff8

Estado de la instancia: running

Tipo de instancia: t2.micro

Zona de disponibilidad: us-east-1a

AMI ID: ami-0eaf7c3456e7b5b68

```
In [ ]: # Crear una lista para almacenar los IDs de las instancias
instance_ids = []

# Iterar sobre las reservas y las instancias para recopilar los IDs de las instancias
for reservation in response['Reservations']:
    for instance in reservation['Instances']:
        instance_ids.append(instance['InstanceId'])

# Detener todas las instancias recopiladas
if instance_ids:
    response = ec2_client.stop_instances(
        InstanceIds=instance_ids
    )

    print(f'Las siguientes instancias se están deteniendo: {instance_ids}')
else:
    print('No se encontraron instancias para detener.')
```

Las siguientes instancias se están deteniendo: ['i-01642a030325a32b5', 'i-06471e09a295bfff8']

```
In [ ]: # Borraremos todas las instancias

# Describir las instancias
response = ec2_client.describe_instances()

# Crear una lista para almacenar los IDs de las instancias detenidas
stopped_instance_ids = []

# Iterar sobre las reservas y las instancias
for reservation in response['Reservations']:
    for instance in reservation['Instances']:
        if instance['State']['Name'] == 'stopped':
            stopped_instance_ids.append(instance['InstanceId'])

# Terminar todas las instancias detenidas
if stopped_instance_ids:
    response = ec2_client.terminate_instances(
        InstanceIds=stopped_instance_ids
    )
    print(f'Las siguientes instancias detenidas están siendo terminadas: {stopped_instance_ids}')
else:
    print('No se encontraron instancias detenidas para terminar.')
```

Las siguientes instancias detenidas están siendo terminadas: ['i-01642a030325a32b5', 'i-06471e09a295bfff8']