### **Lab Guide: Creating and Terminating an EC2 Instance Using AWS CLI and Python Boto3**

## **Objective**

This lab provides step-by-step instructions on how to:

1. Create an EC2 instance using AWS CLI.
2. Terminate an EC2 instance using AWS CLI.
3. Create an EC2 instance using Python Boto3.
4. Terminate an EC2 instance using Python Boto3.

By the end of this lab, you will be able to provision and delete EC2 instances programmatically.

## **Prerequisites**

1. **AWS CLI Installed**: Download and install the AWS CLI from [AWS CLI Download](https://aws.amazon.com/cli/).
2. **AWS CLI Configured**: Run aws configure and set up your AWS credentials.
3. **Python Installed**: Install Python 3.x.

**Boto3 Installed**: Install Boto3 using:  
bash  
  
pip install boto3

1. **IAM User with EC2 Full Access**: Ensure the IAM user has permissions to create and terminate EC2 instances.

# **Part 1: Create and Terminate EC2 Instance Using AWS CLI**

## **Step 1: Create an EC2 Instance Using AWS CLI**

1. **Open Terminal or Command Prompt.**

**Run the following command to create an EC2 instance**:  
bash  
Copy  
aws ec2 run-instances --image-id ami-0c02fb55956c7d316 --count 1 --instance-type t2.micro --key-name MyKeyPair --security-groups default

A screen shot of a computer

Description automatically generated

1. Replace MyKeyPair with your actual key pair name.

**Output Example:**json  
Copy  
{

"Instances": [

{

"InstanceId": "i-0123456789abcdef0",

"ImageId": "ami-0c02fb55956c7d316",

"InstanceType": "t2.micro",

"State": {

"Code": 0,

"Name": "pending"

}

}

]

}

1. Note the **Instance ID** (i-0123456789abcdef0).

## **Step 2: Verify the Running Instance**

Run the following command to verify the instance is running:

bash

Copy

aws ec2 describe-instances --instance-ids i-0123456789abcdef0

Replace i-0123456789abcdef0 with your instance ID.

A screen shot of a computer screen

Description automatically generated

## **Step 3: Terminate the EC2 Instance Using AWS CLI**

Run the following command:  
bash  
Copy  
aws ec2 terminate-instances --instance-ids i-012345678

Copy

Replace `i-0123456789abcdef0` with your instance ID.

A computer screen shot of a computer code

Description automatically generated

2. Verify that the instance is terminating by running:

```bash

aws ec2 describe-instances --instance-ids i-0123456789abcdef0

The **State** should show "Shutting-down" followed by "Terminated".

A computer screen with white text

Description automatically generated

# **Part 2: Create and Terminate EC2 Instance Using Python Boto3**

## **Step 1: Create an EC2 Instance Using Python Boto3**

**Open a terminal and create a Python script named create\_ec2.py**:  
python  
  
import boto3

# Initialize a session using Boto3

ec2 = boto3.resource('ec2', region\_name='us-east-1')

# Create an EC2 instance

instance = ec2.create\_instances(

ImageId='ami-0c02fb55956c7d316', # Amazon Linux 2 AMI (Free Tier Eligible)

MinCount=1,

MaxCount=1,

InstanceType='t2.micro',

KeyName='MyKeyPair', # Ensure you have created this key pair

TagSpecifications=[

{

'ResourceType': 'instance',

'Tags': [

{'Key': 'Name', 'Value': 'MyPythonEC2Instance'}

]

}

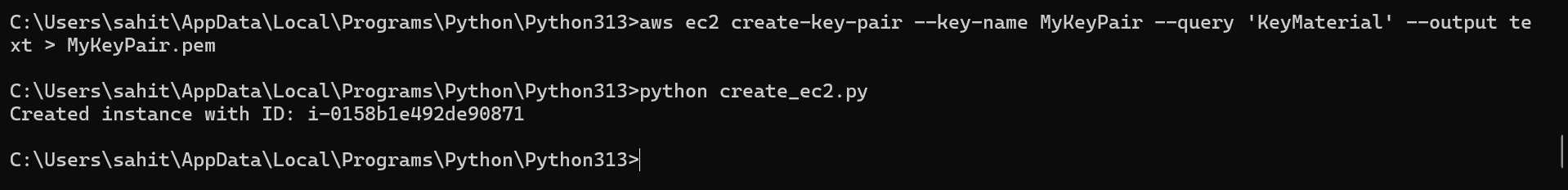
]

)

print(f'Created instance with ID: {instance[0].id}')

**Run the script to create an instance**:  
bash  
  
python create\_ec2.py

1. **Note the instance ID displayed in the output**.



## **Step 2: Terminate EC2 Instance Using Python Boto3**

**Create another Python script named terminate\_ec2.py**:  
python  
  
import boto3

# Initialize EC2 client

ec2\_client = boto3.client('ec2', region\_name='us-east-1')

# Replace with your instance ID

instance\_id = 'i-0123456789abcdef0'

# Stop and terminate the instance

print(f"Stopping instance: {instance\_id}")

ec2\_client.stop\_instances(InstanceIds=[instance\_id])

waiter = ec2\_client.get\_waiter('instance\_stopped')

waiter.wait(InstanceIds=[instance\_id])

print(f"Instance {instance\_id} stopped.")

print(f"Terminating instance: {instance\_id}")

ec2\_client.terminate\_instances(InstanceIds=[instance\_id])

waiter = ec2\_client.get\_waiter('instance\_terminated')

waiter.wait(InstanceIds=[instance\_id])

print(f"Instance {instance\_id} terminated.")

**Run the script to terminate the instance**:  
bash  
Copy  
python terminate\_ec2.py

A computer screen shot of a black background

Description automatically generated

# **Verification and Cleanup**

**Check running instances using AWS CLI**:  
bash  
Copy  
aws ec2 describe-instances

1. Ensure there are no running instances.
2. **Check via AWS Console**:
   * Navigate to **EC2 Dashboard**.
   * Verify that no instances are running.

A computer screen shot of white text

Description automatically generated

# **Deliverables**

1. **Screenshots**:
   * AWS CLI output for instance creation and termination.
   * AWS Console showing instance state changes.
   * Python script execution output.
2. **Summary Report**:
   * Describe steps followed.
   * List any challenges faced and how they were resolved.