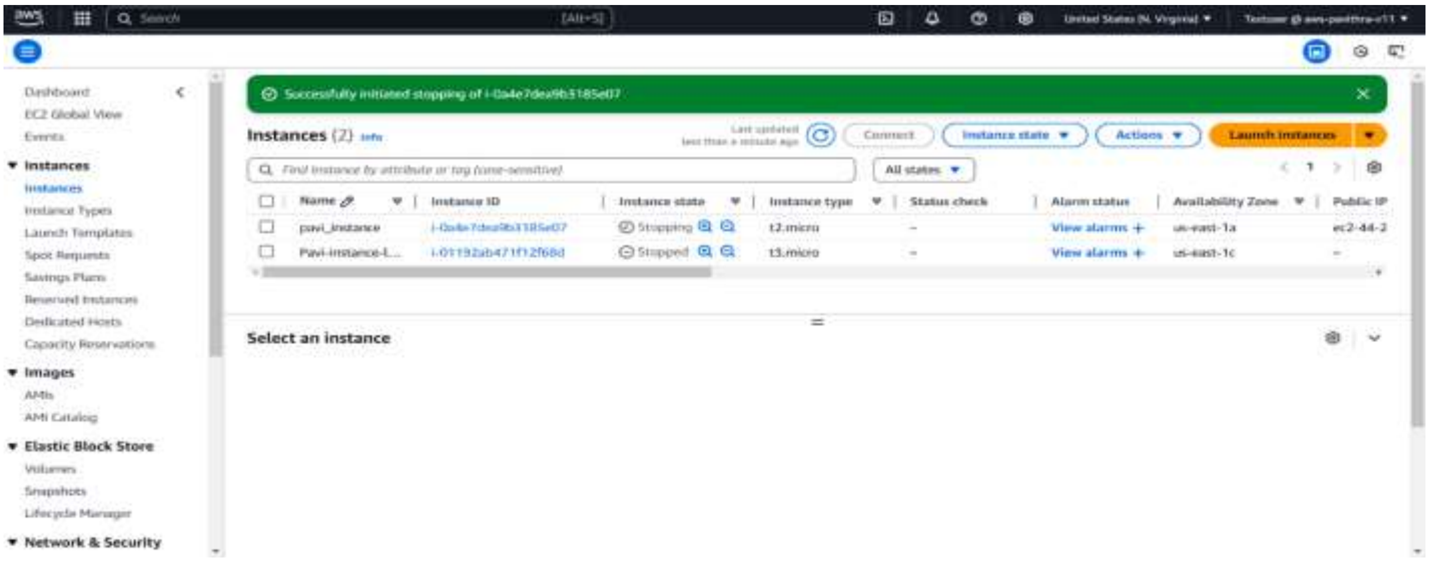
Creating and terminating an EC2 Instance Using AWS CLI and Python Boto3

Part 1: Create and Terminate EC2 Instance Using AWS CLI

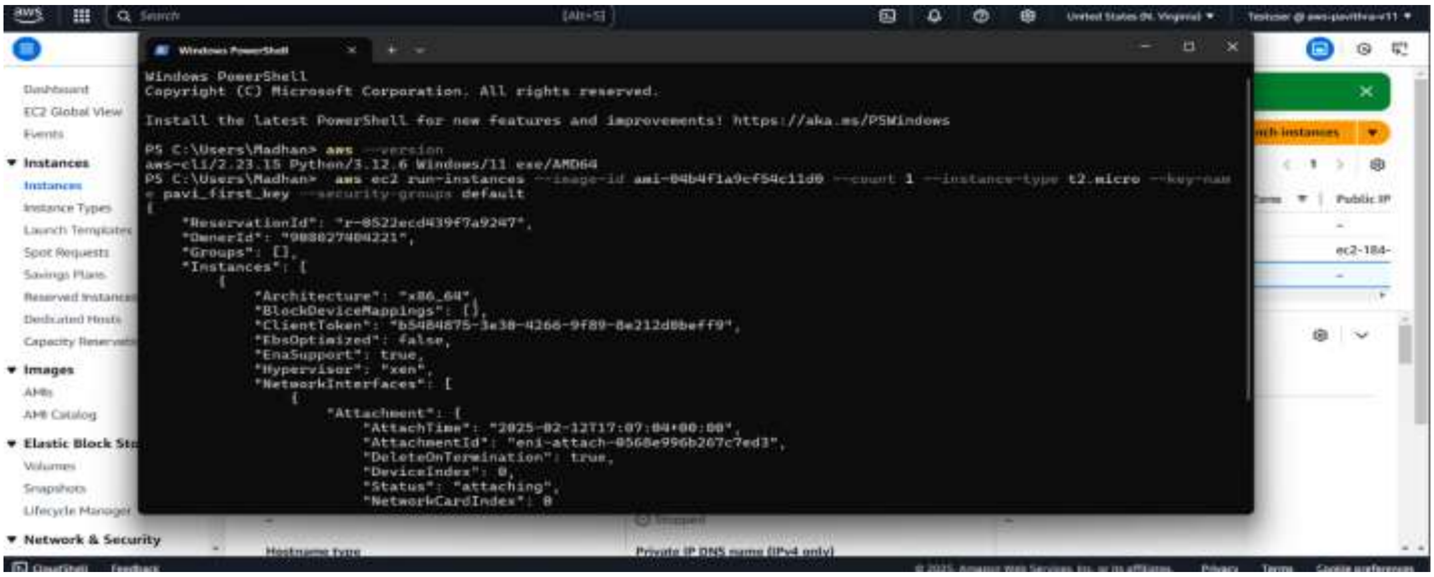
Step 1: Create an EC2 Instance Using AWS CLI

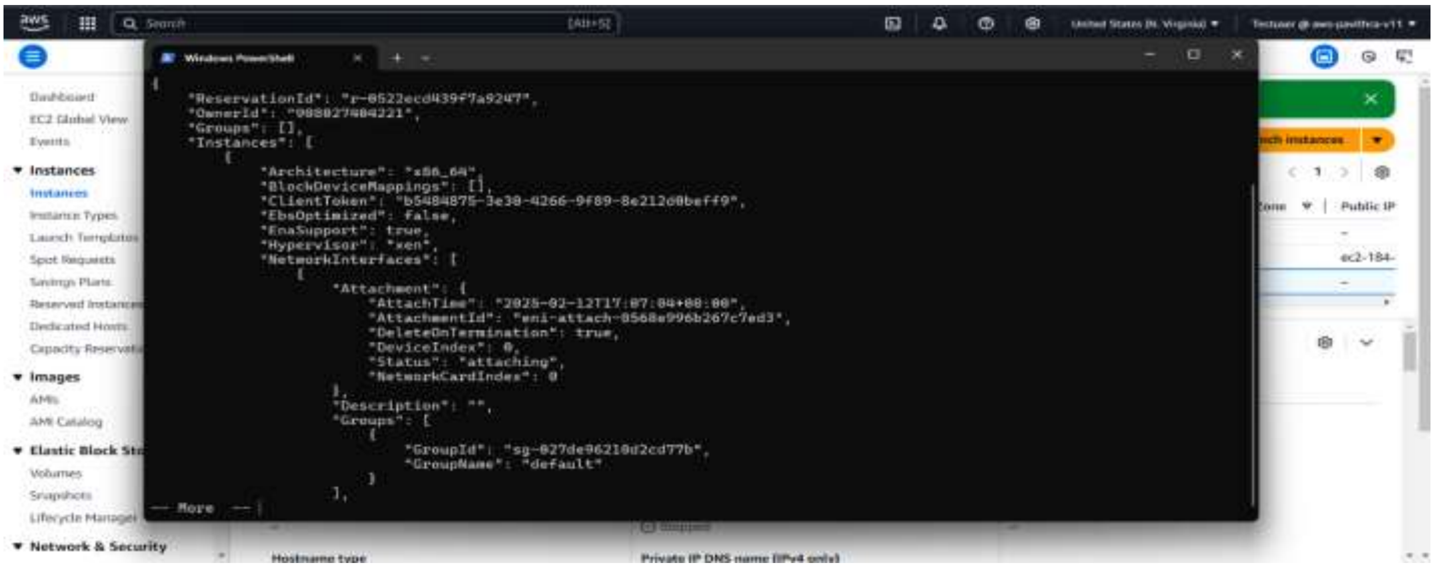

Successfully initiated stopping of i-0a4e7dea9b3185e07

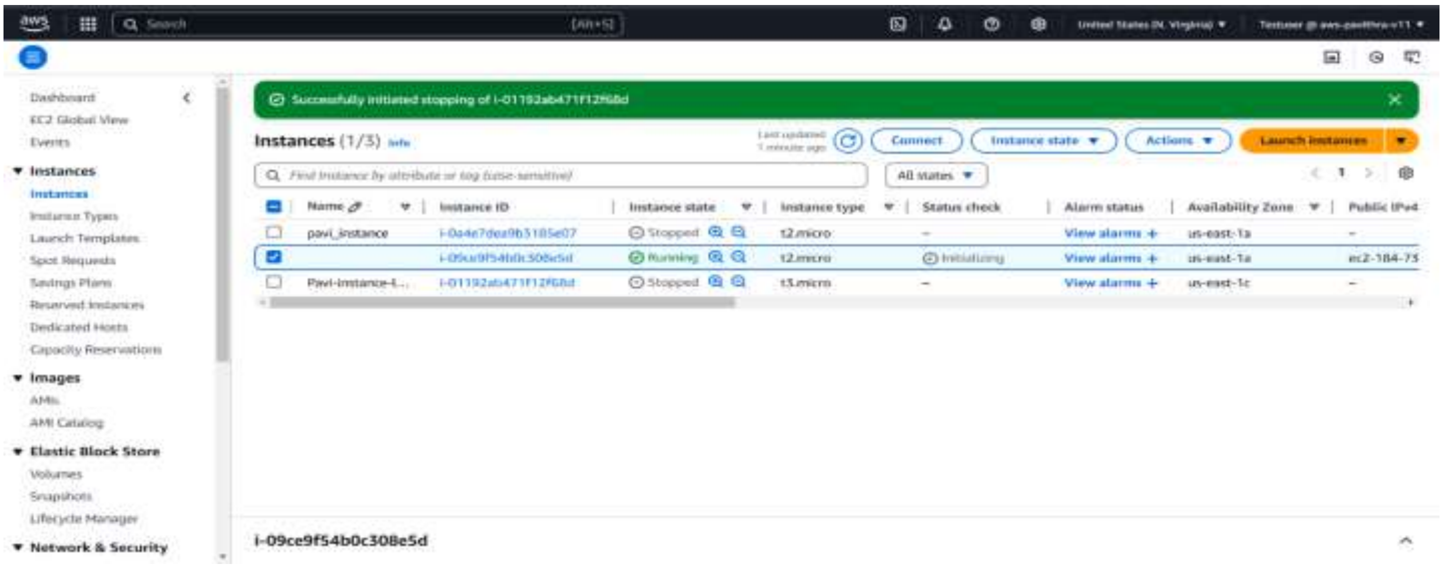
Instances (2)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
☐	pavi_instance	i-0a4e7dea9b3185e07	Stopping	t2.micro	-	View alarms +	us-east-1a	ec2-44-2
☐	Pavi-instance-L...	i-01192ab471f12f68d	Stopped	t3.micro	-	View alarms +	us-east-1c	-

Select an instance


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Madhan> aws --version
aws-cli/2.23.15 Python/3.12.6 Windows/11 exe/AMD64
PS C:\Users\Madhan>  aws ec2 run-instances --image-id ami-04b4f1a9cf54c11d0 --count 1 --instance-type t2.micro --key-nam
e pavi_first_key --security-groups default
{
    "ReservationId": "r-0522ecd439f7a9247",
    "OwnerId": "908827404221",
    "Groups": [],
    "Instances": [
        {
            "Architecture": "x86_64",
            "BlockDeviceMappings": [],
            "ClientToken": "b5484875-3e38-4266-9f89-8e212d0beff9",
            "EbsOptimized": false,
            "EnaSupport": true,
            "Hypervisor": "xen",
            "NetworkInterfaces": [
                {
                    "Attachment": {
                        "AttachTime": "2025-02-12T17:07:04+00:00",
                        "AttachmentId": "eni-attach-0568e996b267c7ed3",
                        "DeleteOnTermination": true,
                        "DeviceIndex": 0,
                        "Status": "attaching",
                        "NetworkCardIndex": 0
```


```
{
    "ReservationId": "r-0522ecd439f7a9247",
    "OwnerId": "908827404221",
    "Groups": [],
    "Instances": [
        {
            "Architecture": "x86_64",
            "BlockDeviceMappings": [],
            "ClientToken": "b5484875-3e38-4266-9f89-8e212d0beff9",
            "EbsOptimized": false,
            "EnaSupport": true,
            "Hypervisor": "xen",
            "NetworkInterfaces": [
                {
                    "Attachment": {
                        "AttachTime": "2025-02-12T17:07:04+00:00",
                        "AttachmentId": "eni-attach-0568e996b267c7ed3",
                        "DeleteOnTermination": true,
                        "DeviceIndex": 0,
                        "Status": "attaching",
                        "NetworkCardIndex": 0
                    },
                    "Description": "",
                    "Groups": [
                        {
                            "GroupId": "sg-027de96210d2cd77b",
                            "GroupName": "default"
                        }
                    ],
-- More  --
```

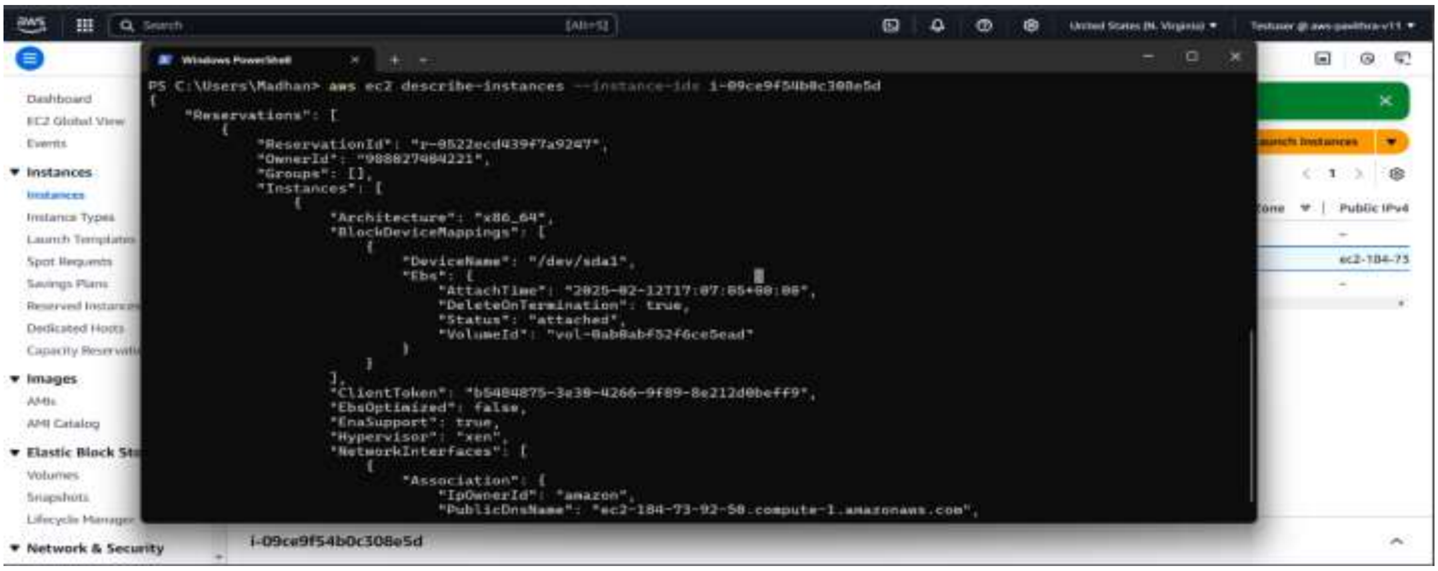

Successfully initiated stopping of i-01192ab471f12f68d

Instances (1/3)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
☐	pavi_instance	i-0a4e7dea9b3185e07	Stopped	t2.micro	–	View alarms +	us-east-1a	–
☑		i-09ce9f54b0c308e5d	Running	t2.micro	Initializing	View alarms +	us-east-1a	ec2-184-73
☐	Pavi-instance-l...	i-01192ab471f12f68d	Stopped	t3.micro	–	View alarms +	us-east-1c	–

i-09ce9f54b0c308e5d

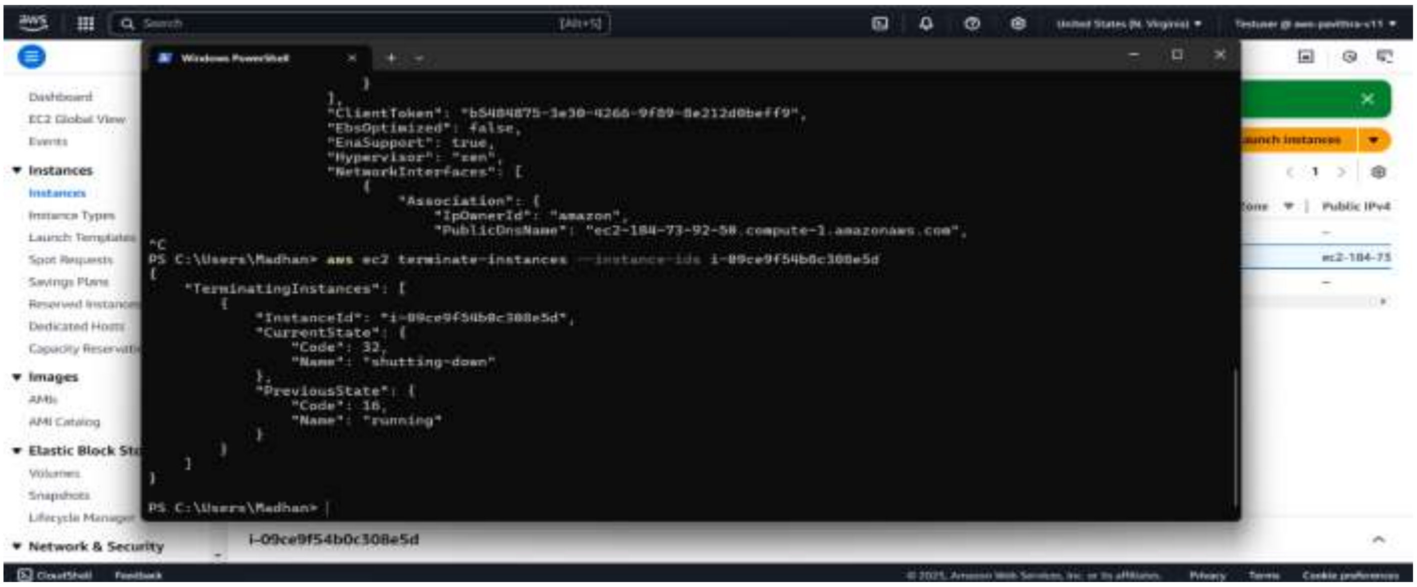
Step 2: Verify the Running Instance



```
PS C:\Users\Madhan> aws ec2 describe-instances --instance-ids i-09ce9f54b0c308e5d
{
    "Reservations": [
        {
            "ReservationId": "r-0522ecd439f7a9247",
            "OwnerId": "988827484221",
            "Groups": [],
            "Instances": [
                {
                    "Architecture": "x86_64",
                    "BlockDeviceMappings": [
                        {
                            "DeviceName": "/dev/sda1",
                            "Ebs": {
                                "AttachTime": "2025-02-12T17:07:05+00:00",
                                "DeleteOnTermination": true,
                                "Status": "attached",
                                "VolumeId": "vol-0ab0abf52f6ce5ead"
                            }
                        }
                    ],
                    "ClientToken": "b5484875-3e38-4266-9f89-8e212d0beff9",
                    "EbsOptimized": false,
                    "EnaSupport": true,
                    "Hypervisor": "xen",
                    "NetworkInterfaces": [
                        {
                            "Association": {
                                "IpOwnerId": "amazon",
                                "PublicDnsName": "ec2-184-73-92-58.compute-1.amazonaws.com",
```

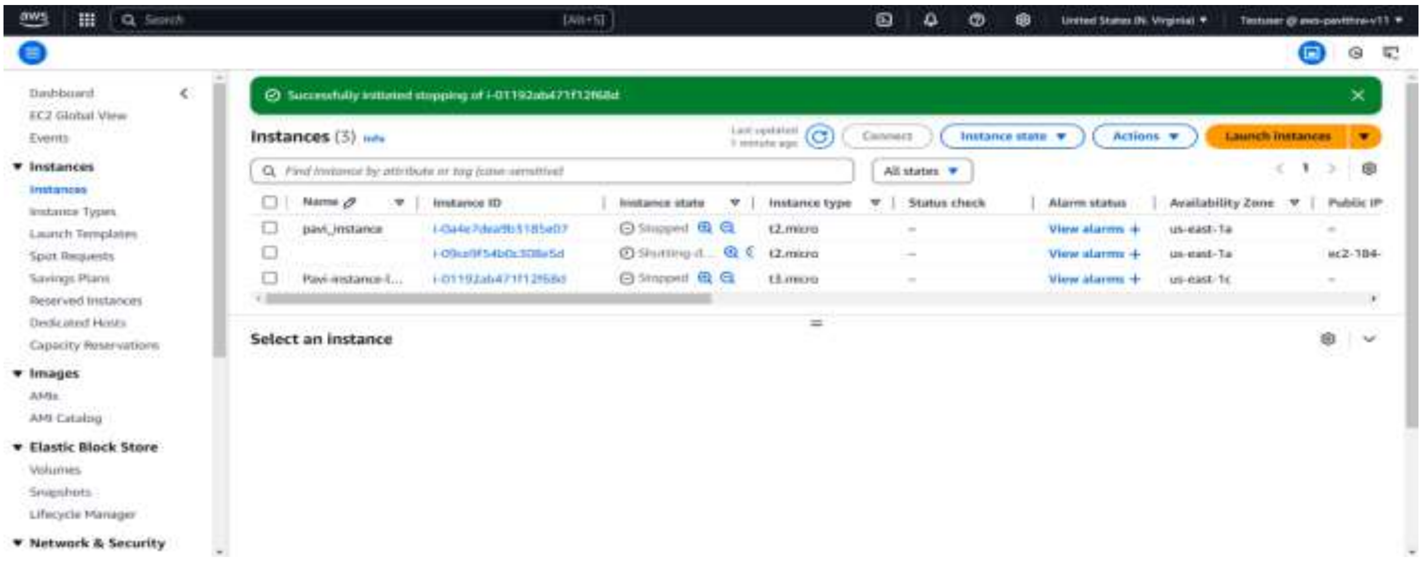
i-09ce9f54b0c308e5d

Step 3: Terminate the EC2 Instance Using AWS CLI



```
            }
        ],
        "ClientToken": "b5484875-3e30-4266-9f89-8e212d0beff9",
        "EbsOptimized": false,
        "EnaSupport": true,
        "Hypervisor": "xen",
        "NetworkInterfaces": [
            {
                "Association": {
                    "IpOwnerId": "amazon",
                    "PublicDnsName": "ec2-184-73-92-58.compute-1.amazonaws.com",
^C
PS C:\Users\Madhan> aws ec2 terminate-instances --instance-ids i-09ce9f54b0c308e5d
{
    "TerminatingInstances": [
        {
            "InstanceId": "i-09ce9f54b0c308e5d",
            "CurrentState": {
                "Code": 32,
                "Name": "shutting-down"
            },
            "PreviousState": {
                "Code": 16,
                "Name": "running"
            }
        }
    ]
}
PS C:\Users\Madhan> |
```

i-09ce9f54b0c308e5d



Successfully initiated stopping of i-01192ab471f12f68d

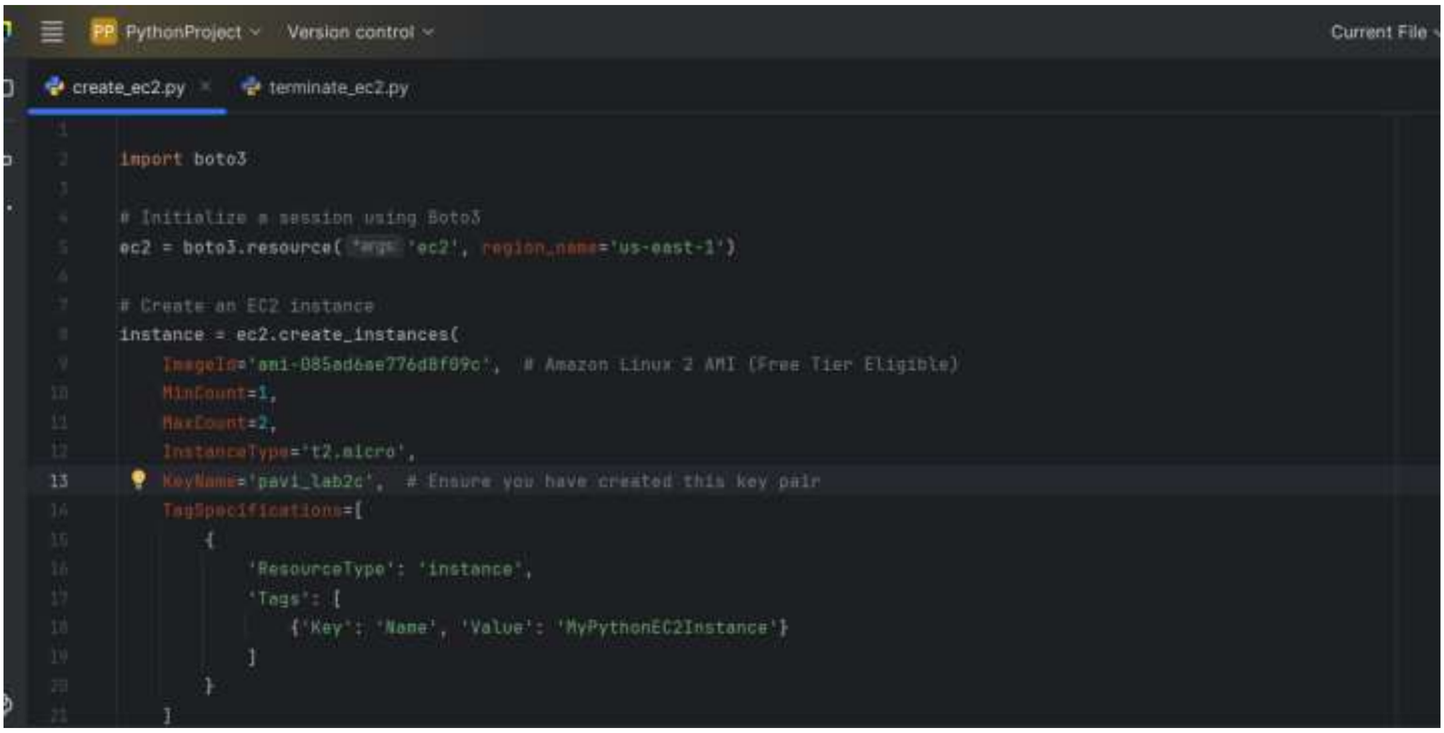
Instances (3)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
pavi_instance	i-0a4e7dea9b3185e07	Stopped	t2.micro	–	View alarms +	us-east-1a	–
	i-09ce9f54b0c308e5d	Shutting-d...	t2.micro	–	View alarms +	us-east-1a	ec2-184-
Pavi-instance-l...	i-01192ab471f12f68d	Stopped	t3.micro	–	View alarms +	us-east-1c	–

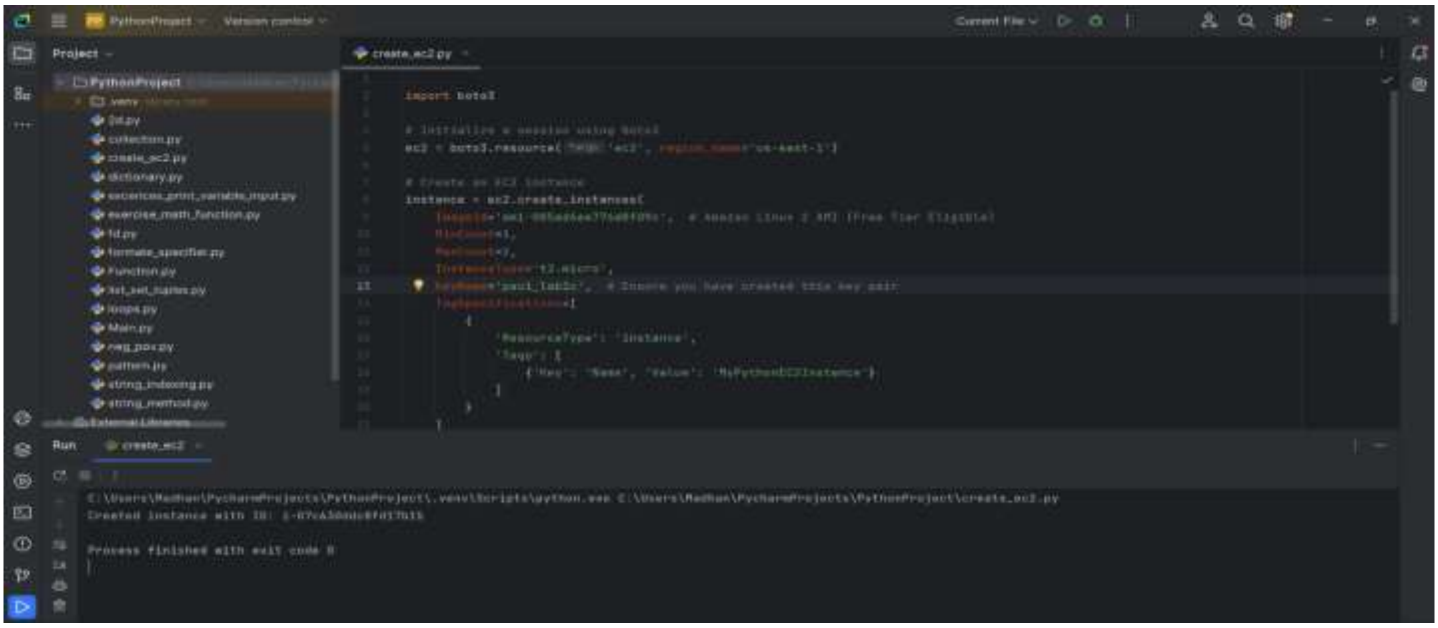
Select an instance

Part 2: Create and Terminate EC2 Instance Using Python Boto3

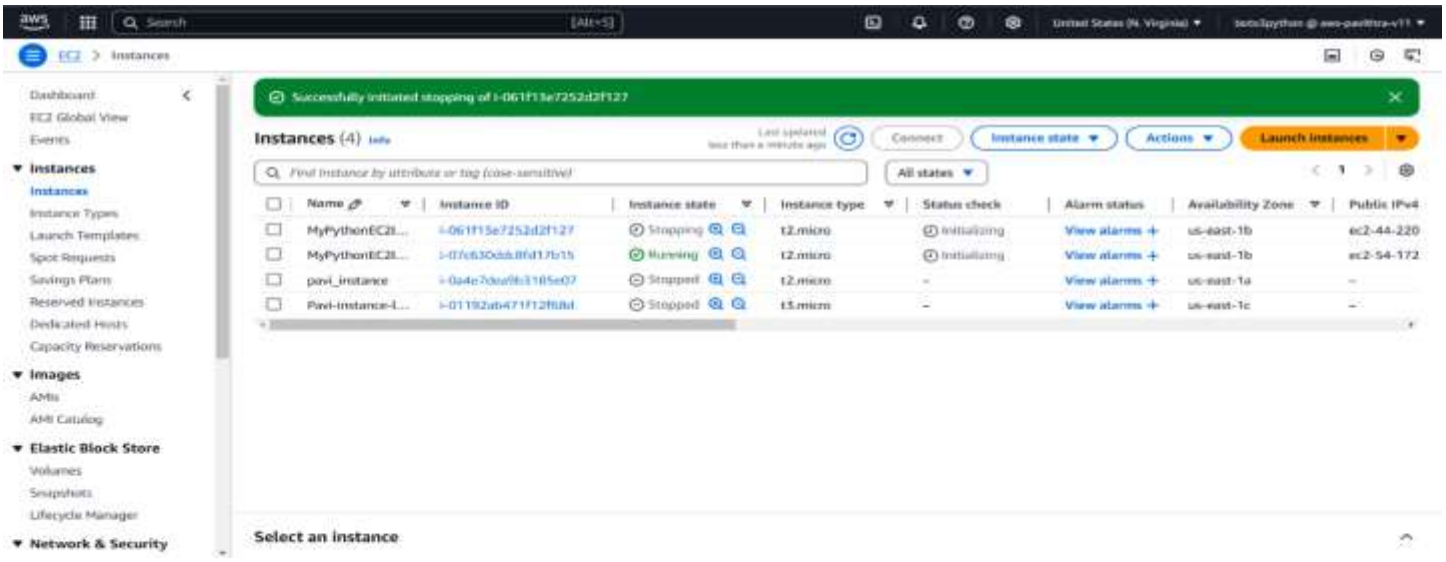
Step 1: Create an EC2 Instance Using Python Boto3



```python

import boto3

# Initialize a session using Boto3
ec2 = boto3.resource( *args: 'ec2', region_name='us-east-1')

# Create an EC2 instance
instance = ec2.create_instances(
    ImageId='ami-085ad6ae776d8f09c',  # Amazon Linux 2 AMI (Free Tier Eligible)
    MinCount=1,
    MaxCount=2,
    InstanceType='t2.micro',
    KeyName='pavi_lab2c',  # Ensure you have created this key pair
    TagSpecifications=[
        {
            'ResourceType': 'instance',
            'Tags': [
                {'Key': 'Name', 'Value': 'MyPythonEC2Instance'}
            ]
        }
    ]
```



```
C:\Users\Madhan\PycharmProjects\PythonProject\.venv\Scripts\python.exe C:\Users\Madhan\PycharmProjects\PythonProject\create_ec2.py
Created instance with ID: [illegible]

Process finished with exit code 0
```

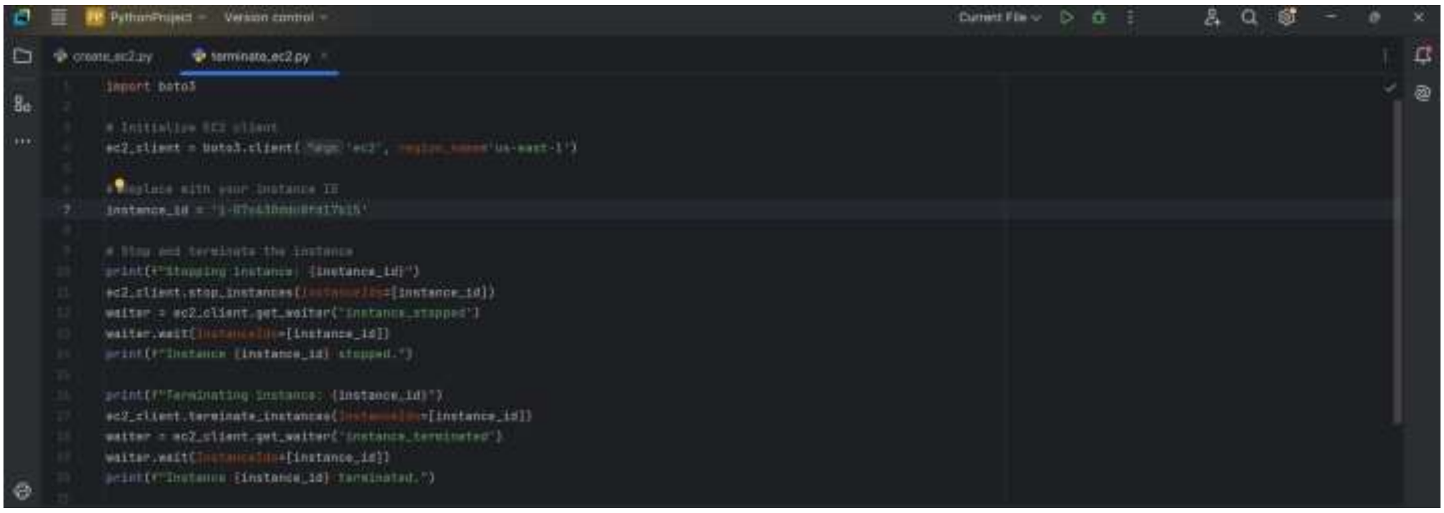

Successfully initiated stopping of i-061f15e7252d2f127

Instances (4)

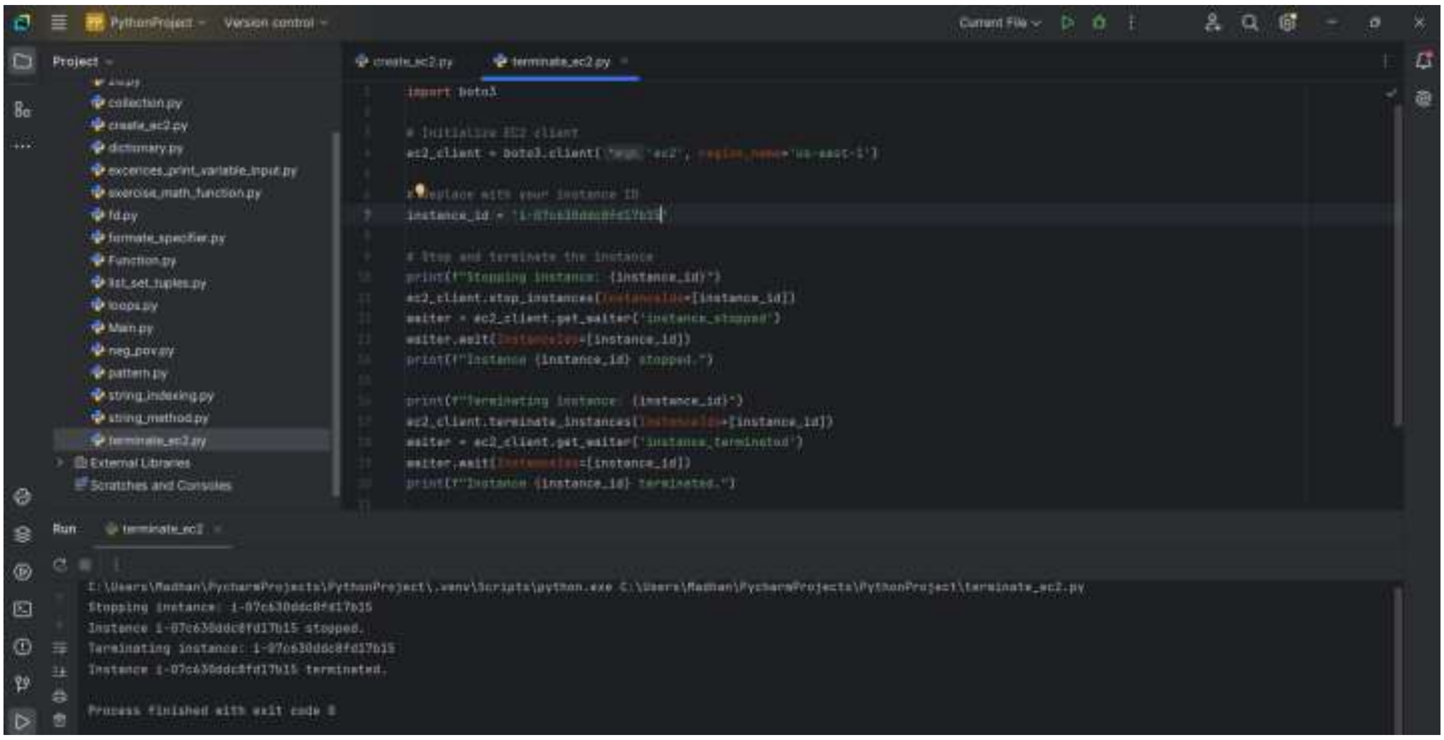
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
MyPythonEC2I...	i-061f15e7252d2f127	Stopping	t2.micro	Initializing	View alarms +	us-east-1b	ec2-44-220
MyPythonEC2I...	i-07c630ddc8fd17b15	Running	t2.micro	Initializing	View alarms +	us-east-1b	ec2-54-172
pavi_instance	i-0a4e7dea9b3185e07	Stopped	t2.micro	-	View alarms +	us-east-1a	-
Pavi-instance-I...	i-01192ab471f12f68d	Stopped	t3.micro	-	View alarms +	us-east-1c	-

---

Step 2: Terminate EC2 Instance Using Python Boto3

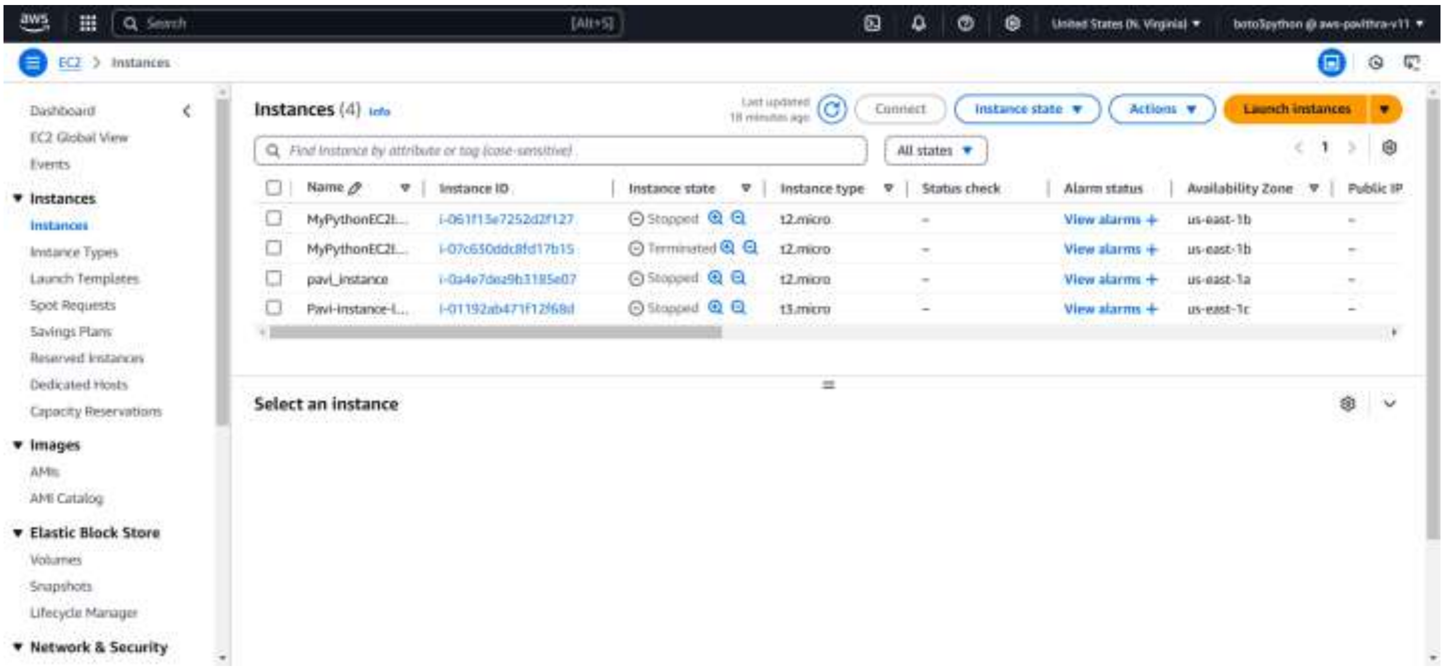


```python
import boto3

# Initialize EC2 client
ec2_client = boto3.client('ec2', region_name='us-east-1')

# Replace with your instance ID
instance_id = 'i-07c630ddc8fd17b15'

# 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.")
```



```python
import boto3

# Initialize EC2 client
ec2_client = boto3.client('ec2', region_name='us-east-1')

# Replace with your instance ID
instance_id = 'i-07c630ddc8fd17b15'

# 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.")
```

```
C:\Users\Madhan\PycharmProjects\PythonProject\.venv\Scripts\python.exe C:\Users\Madhan\PycharmProjects\PythonProject\terminate_ec2.py
Stopping instance: i-07c630ddc8fd17b15
Instance i-07c630ddc8fd17b15 stopped.
Terminating instance: i-07c630ddc8fd17b15
Instance i-07c630ddc8fd17b15 terminated.

Process finished with exit code 0
```

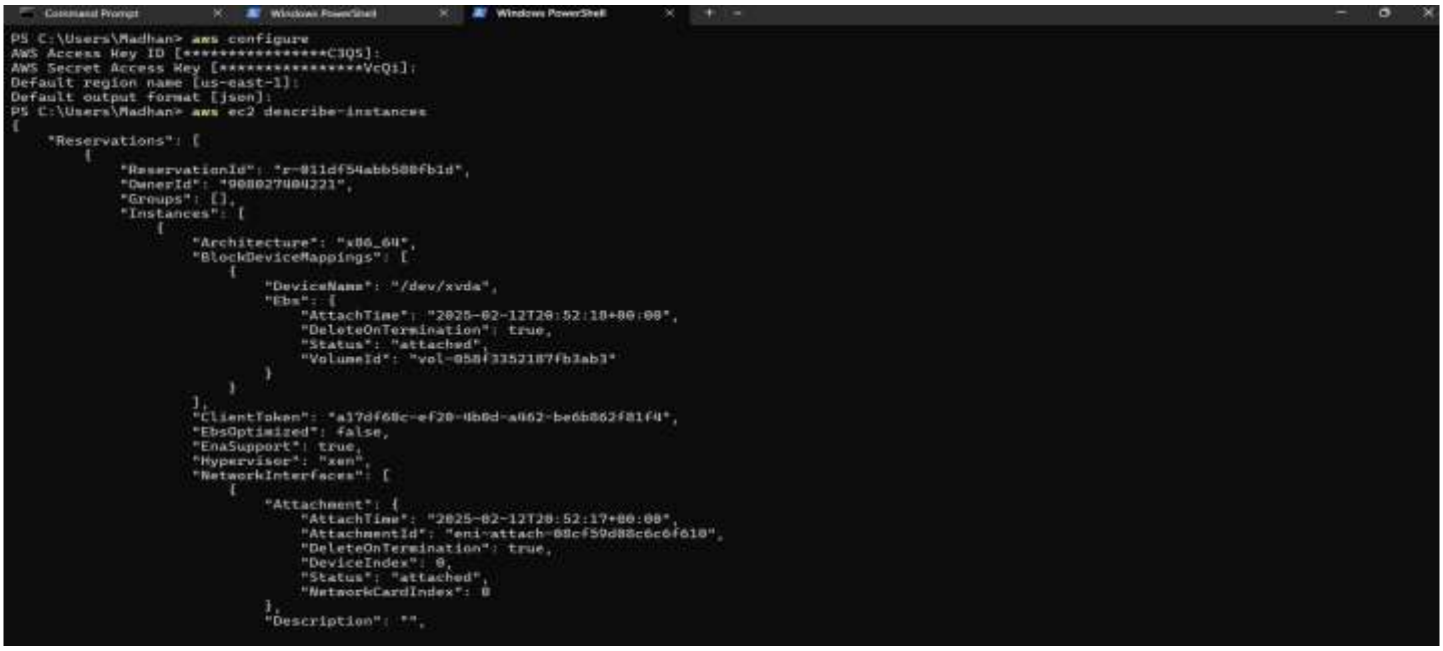


Instances (4) Info

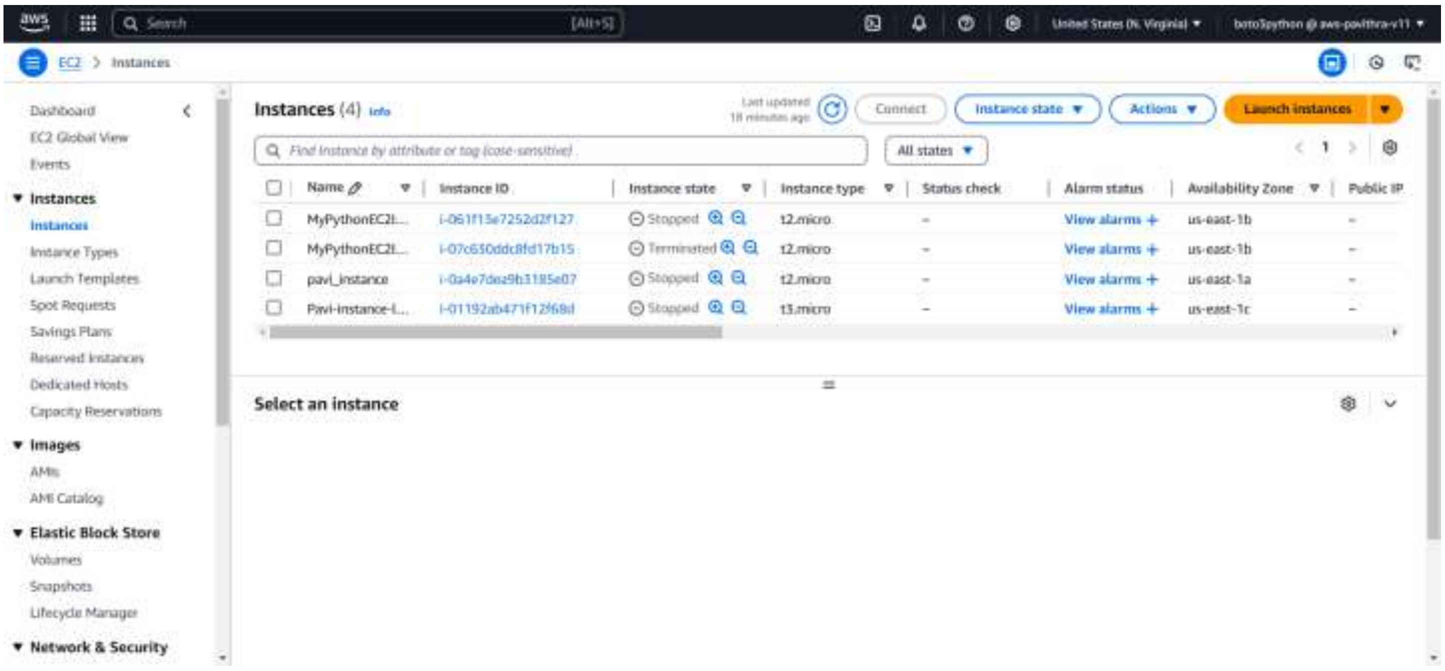
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
MyPythonEC2I...	i-061f13e7252d2f127	Stopped	t2.micro	–	View alarms +	us-east-1b	–
MyPythonEC2I...	i-07c630ddc8fd17b15	Terminated	t2.micro	–	View alarms +	us-east-1b	–
pavi_instance	i-0a4e7dea9b3185e07	Stopped	t2.micro	–	View alarms +	us-east-1a	–
Pavi-instance-l...	i-01192ab471f12f68d	Stopped	t3.micro	–	View alarms +	us-east-1c	–

Select an instance

Verification and Cleanup



```
PS C:\Users\Madhan> aws configure
AWS Access Key ID [****************C3Q5]:
AWS Secret Access Key [****************VcQ1]:
Default region name [us-east-1]:
Default output format [json]:
PS C:\Users\Madhan> aws ec2 describe-instances
{
    "Reservations": [
        {
            "ReservationId": "r-011df54abb580fb1d",
            "OwnerId": "908027404221",
            "Groups": [],
            "Instances": [
                {
                    "Architecture": "x86_64",
                    "BlockDeviceMappings": [
                        {
                            "DeviceName": "/dev/xvda",
                            "Ebs": {
                                "AttachTime": "2025-02-12T20:52:18+00:00",
                                "DeleteOnTermination": true,
                                "Status": "attached",
                                "VolumeId": "vol-058f3352187fb3ab3"
                            }
                        }
                    ],
                    "ClientToken": "a17df68c-ef20-4b0d-a462-be6b862f81f4",
                    "EbsOptimized": false,
                    "EnaSupport": true,
                    "Hypervisor": "xen",
                    "NetworkInterfaces": [
                        {
                            "Attachment": {
                                "AttachTime": "2025-02-12T20:52:17+00:00",
                                "AttachmentId": "eni-attach-08cf59d88c6c6f610",
                                "DeleteOnTermination": true,
                                "DeviceIndex": 0,
                                "Status": "attached",
                                "NetworkCardIndex": 0
                            },
                            "Description": "",
```



Instances (4) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
MyPythonEC2I...	i-061f13e7252d2f127	Stopped	t2.micro	-	View alarms +	us-east-1b	-
MyPythonEC2I...	i-07c630ddc8fd17b15	Terminated	t2.micro	-	View alarms +	us-east-1b	-
pavi_instance	i-0a4e7dea9b3185e07	Stopped	t2.micro	-	View alarms +	us-east-1a	-
Pavi-instance-I...	i-01192ab471f12f68d	Stopped	t3.micro	-	View alarms +	us-east-1c	-

1)Summary:

The assignment was divided into two parts:

1. **Using AWS CLI to create and terminate an EC2 instance**
2. **Using Python Boto3 to create and terminate an EC2 instance**

Part 1: Create and Terminate an EC2 Instance Using AWS CLI

This part involved executing AWS CLI commands to create, verify, and terminate an EC2 instance.

Step 1: Create an EC2 Instance Using AWS CLI

1. **Open a terminal or command prompt with AWS CLI installed.**
2. **Use the following command to create an EC2 instance:**
3. **aws ec2 run-instances --image-id <AMI-ID> --count 1 --instance-type <Instance-Type> --key-name <Key-Pair-Name> --security-groups <Security-Group>**
   - **<AMI-ID>: Amazon Machine Image ID for the instance**
   - **<Instance-Type>: Type of EC2 instance (e.g., t2.micro)**
   - **<Key-Pair-Name>: Name of the key pair for SSH access**
   - **<Security-Group>: Security group name to allow network access**

Step 2: Verify the Running Instance

1. **Check the status of the instance using:**
2. **aws ec2 describe-instances --instance-ids <Instance-ID>**
3. **The output provides details such as the instance state, public IP, and configuration.**

Step 3: Terminate the EC2 Instance Using AWS CLI

1. **To terminate the instance, run:**
2. **aws ec2 terminate-instances --instance-ids <Instance-ID>**

3. **Verify that the instance has been successfully terminated using:**
4. **aws ec2 describe-instances --instance-ids <Instance-ID>**
    - **The instance should show as terminated.**

Part 2: Create and Terminate an EC2 Instance Using Python Boto3

This section involved writing a Python script using the Boto3 library to create and terminate an EC2 instance.

Step 1: Create an EC2 Instance Using Python Boto3

Install the boto3 library (if not already installed):

```
pip install boto3
```

Write a Python script to create an EC2 instance:

```
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-085ad6ae776d8f09c',  # Amazon Linux 2 AMI (Free Tier Eligible)
  MinCount=1,
  MaxCount=2,
  InstanceType='t2.micro',
  KeyName='pavi_lab2c',  # Ensure you have created this key pair
  TagSpecifications=[
    {
      'ResourceType': 'instance',
      'Tags': [
        {'Key': 'Name', 'Value': 'MyPythonEC2Instance'}
      ]
    }
  ]

)

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

Execute the script, and it will return the newly created instance ID.

Step 2: Terminate EC2 Instance Using Python Boto3

```
    import boto3

# Initialize EC2 client
ec2_client = boto3.client('ec2', region_name='us-east-1')

# Replace with your instance ID
```

```
instance_id = 'i-07c630ddc8fd17b15'

# 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.")
```

Execute the script, and it will terminate the specified EC2 instance.

Step 3: Verification and Cleanup

- After termination, check the status of the instance using AWS CLI:
- aws ec2 describe-instances --instance-ids <Instance-ID>
- Ensure the instance status is terminated.

2)Issue Faced and Resolution

While working with **Boto3** to create and terminate an **EC2 instance**, I encountered an issue where my user did not have the **EC2FullAccess** permission. Due to this, I was unable to execute the necessary API calls for EC2 instance management.

To resolve this, I updated the **IAM policy** by attaching the AmazonEC2FullAccess policy to my user. After applying the new permissions, I retried the Boto3 commands, and the issue was successfully resolved, allowing the script to function as expected