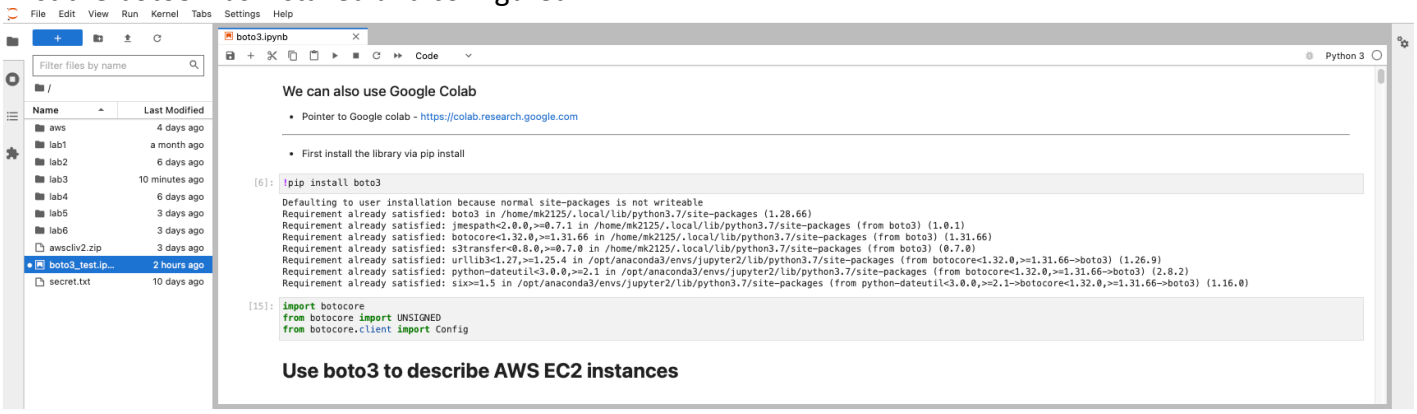
Mushran Khan
Lab3

First the boto3 was installed and configured.



Then we need to create an key ID and use the and access key to connect to my console. This was used to check all the AWS EC2 instances. The following code was used.

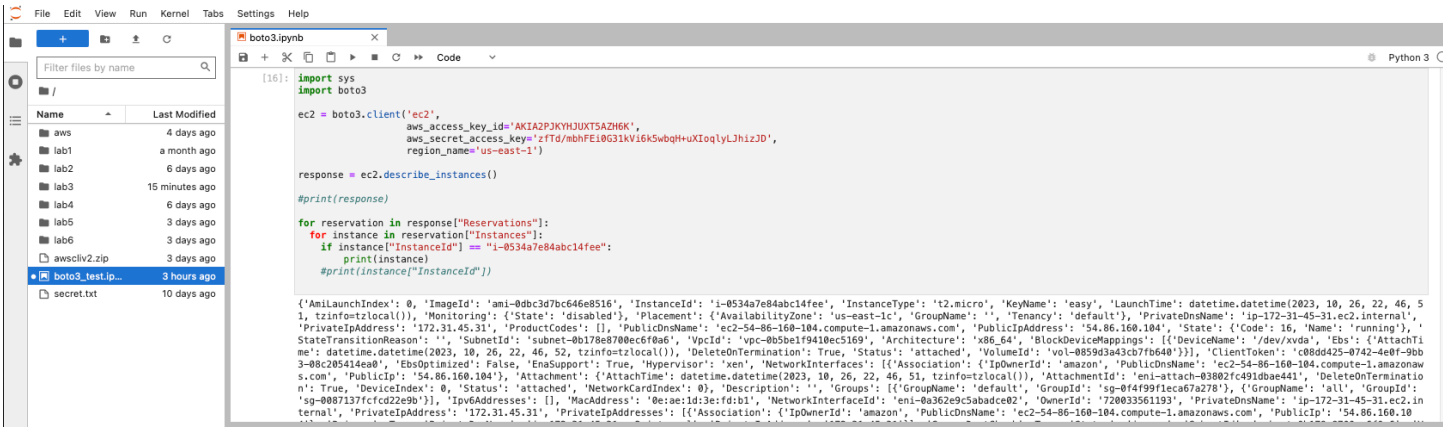
```
import sys
import boto3

ec2 = boto3.client('ec2',
            aws_access_key_id='AKIA2PJKYHJUXT5AZH6K',
            aws_secret_access_key='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',
            region_name='us-east-1')

response = ec2.describe_instances()

#print(response)

for reservation in response["Reservations"]:
  for instance in reservation["Instances"]:
    if instance["InstanceId"] == "i-0534a7e84abc14fee":
        print(instance)
    #print(instance["InstanceId"])
```

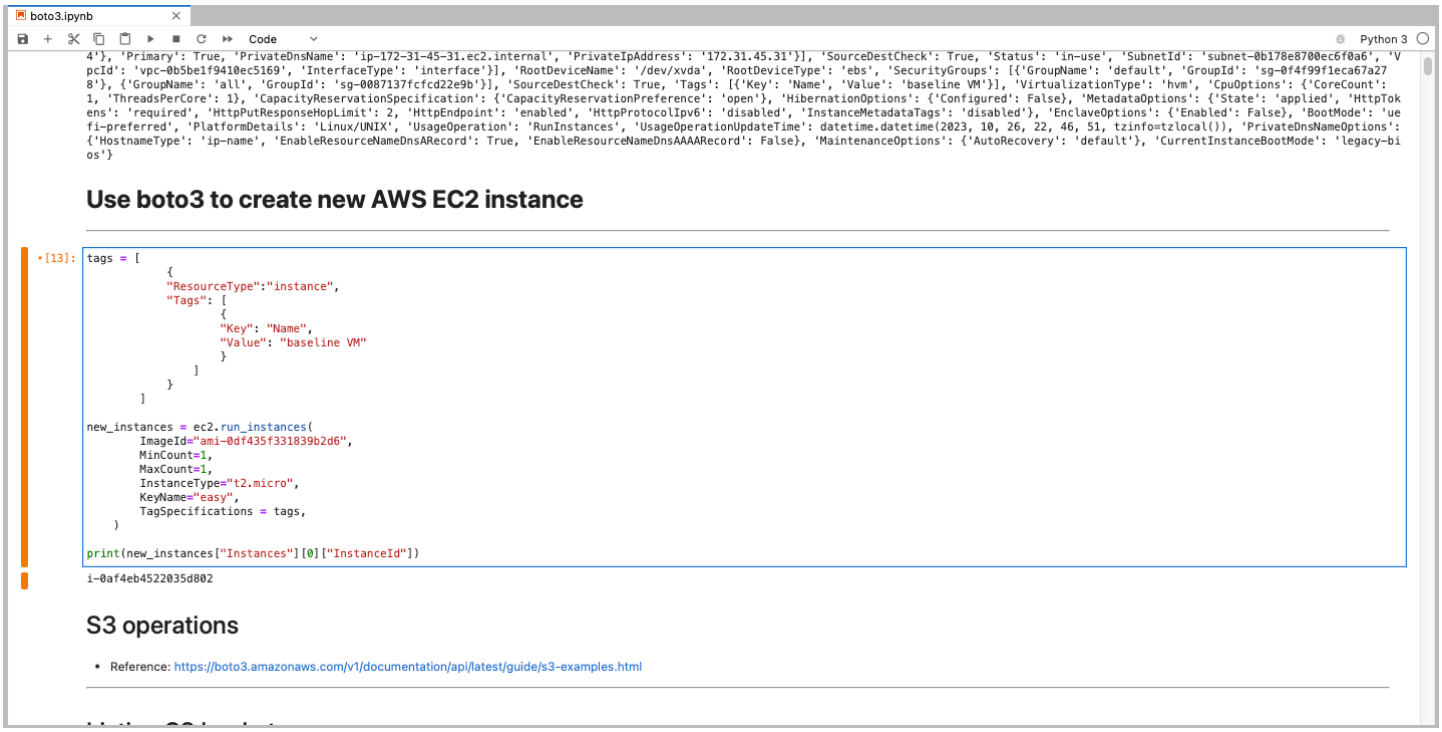


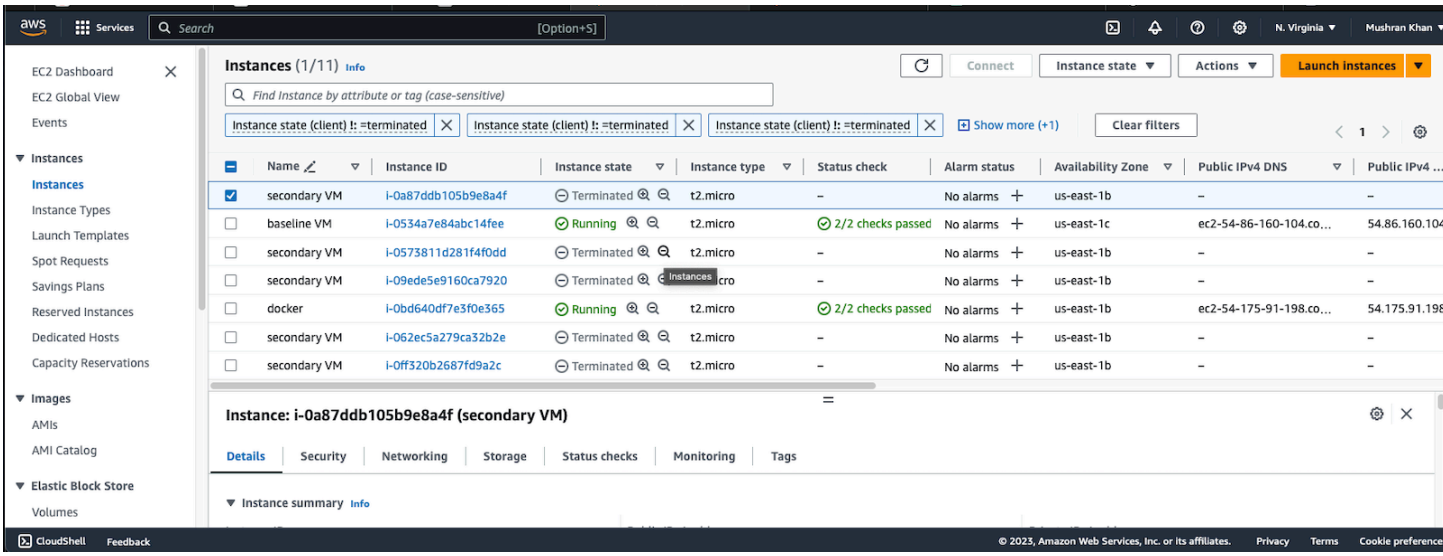
Using the boto3, I create a new VM in Amazon EC2 (“micro” type) and named it “baseline VM”. The follow code was used.

```
tags = [
        {
        "ResourceType":"instance",
        "Tags": [
            {
            "Key": "Name",
            "Value": "baseline VM"
            }
          ]
        }
      ]

new_instances = ec2.run_instances(
      ImageId="ami-0df435f331839b2d6",
      MinCount=1,
      MaxCount=1,
      InstanceType="t2.micro",
      KeyName="easy",
      TagSpecifications = tags,
   )

print(new_instances["Instances"][0]["InstanceId"])
```



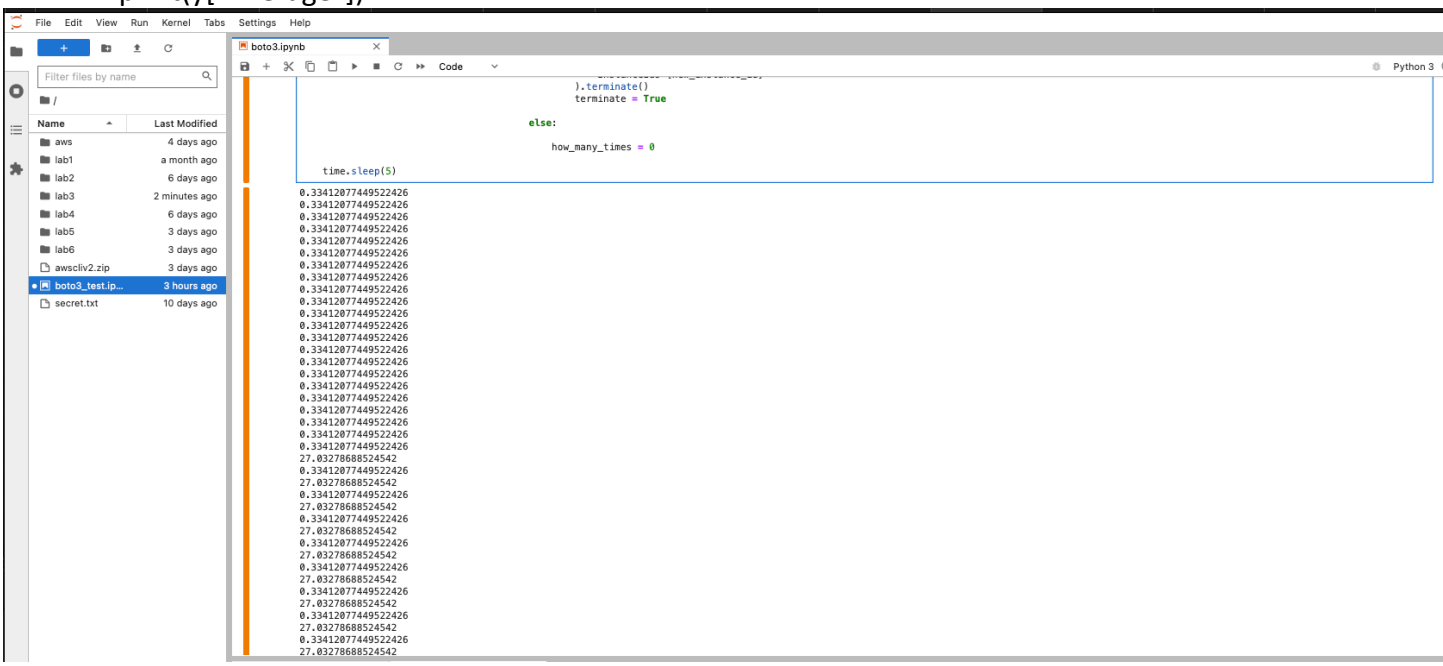


Using the Instance ID and the follow code, to monitor the CPUUtlization of the baseline VM in an endless loop.

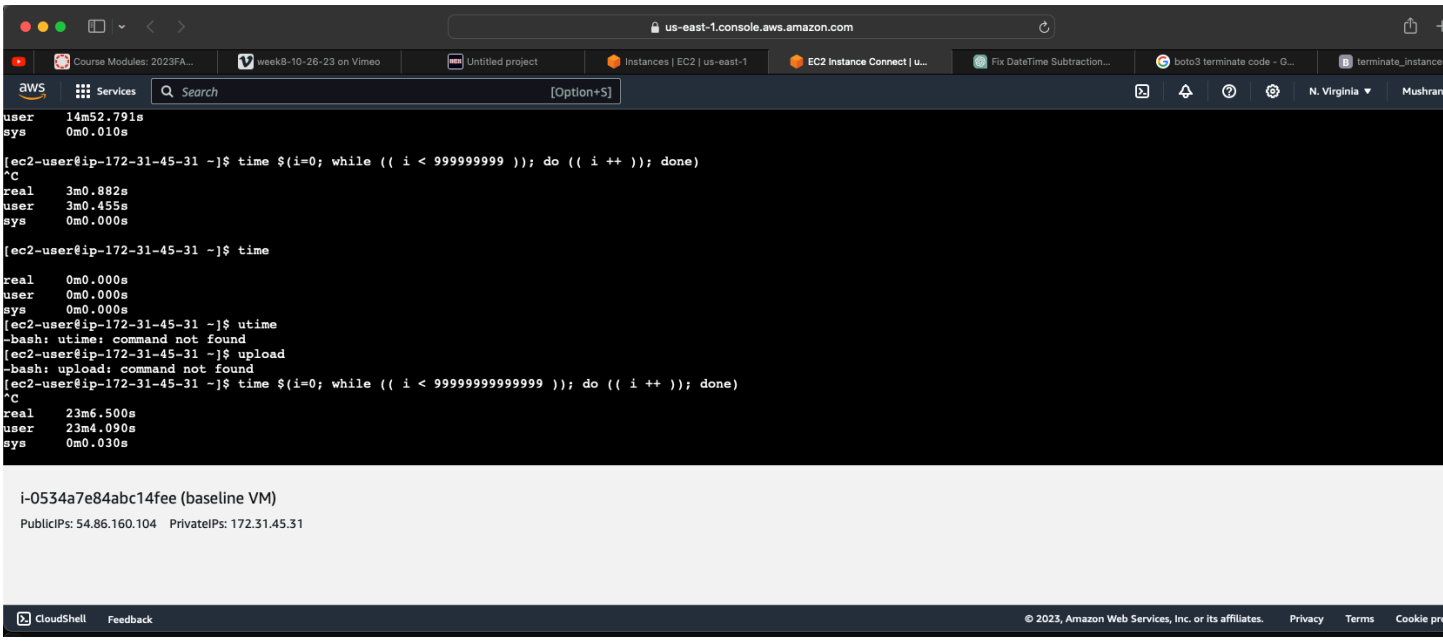
```
now = datetime.now()
   client = boto3.client("cloudwatch", region_name="us-east-1")
   response = client.get_metric_statistics(
      Namespace="AWS/EC2",
      MetricName="CPUUtilization",
```

```
    Dimensions=[
        {"Name": "InstanceId", "Value": "i-0534a7e84abc14fee"},
    ],
    StartTime=now - timedelta(seconds=600),
    EndTime=now,
    Period=60,
    Statistics=[
        "Average",
    ],
    Unit="Percent",
)

for k, v in response.items():
    if k == "Datapoints":
        for y in v:
            print(y["Average"])
```



Load was generated by connecting to the baseline VM and using the benchmark code.
time $(i=0; while ((i < 9999999999)); do ((i ++)); done)



We used the following code to automatically create a new secondary VM when the CPU utilization crossed the 50% usage.

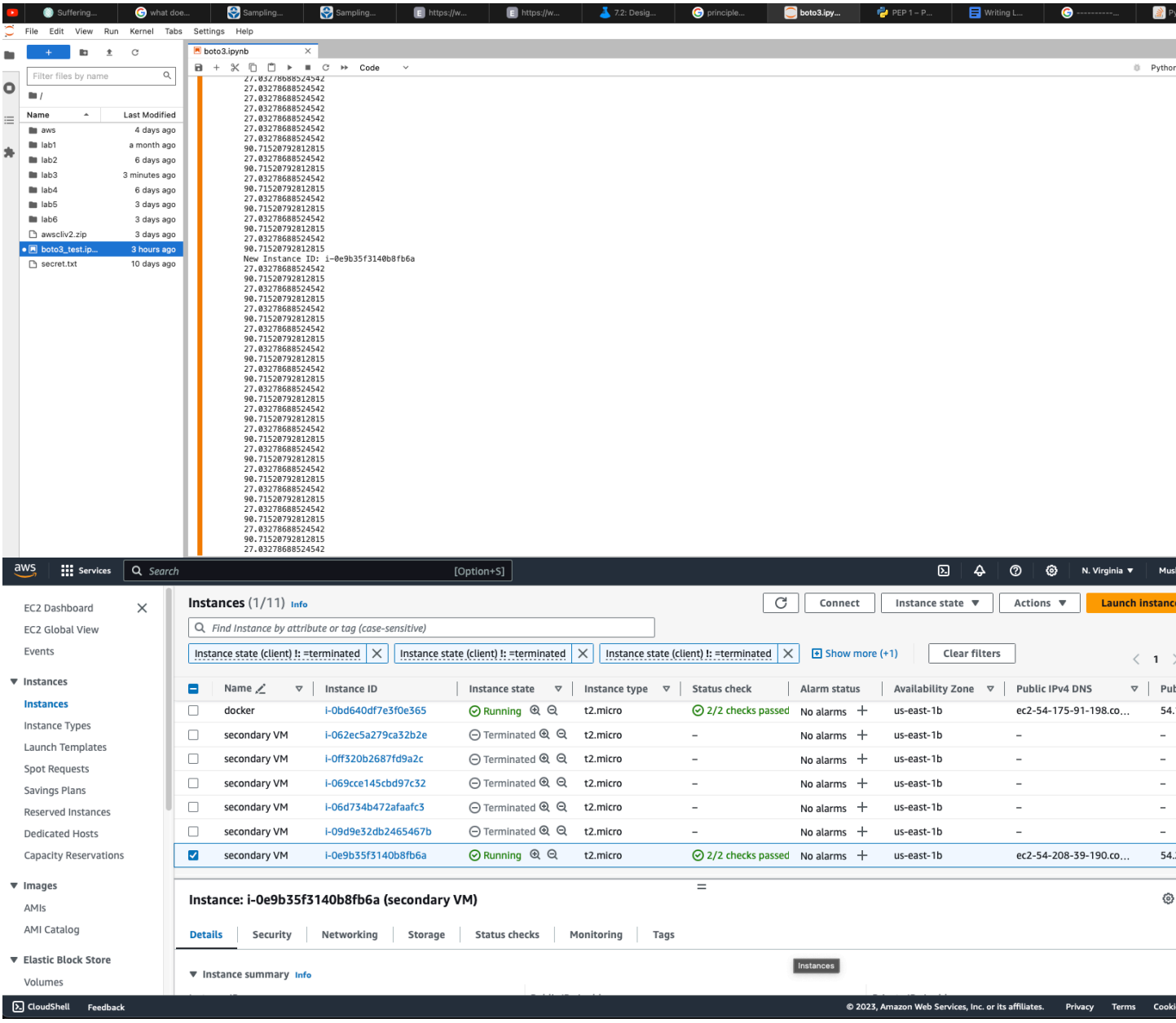
```
if y["Average"] > 50 and not secondary:
        if how_many_times < 5:
            how_many_times += 1

        else:
            how_many_times = 0

            # Create secondary VM
            tags = [
                {
                    "ResourceType": "instance",
                    "Tags": [{"Key": "Name", "Value": "secondary VM"}],
                }
            ]

            new_instances = ec2.create_instances(
                ImageId="ami-0df435f331839b2d6",
                MinCount=1,
                MaxCount=1,
                InstanceType="t2.micro",
                KeyName="easy",
                TagSpecifications=tags,
            )
```

```
new_instance_id = new_instances[0].id
print(f"New Instance ID: {new_instance_id}")
```



Once the CPU utilization came down below 50%, the code automatically notices it and terminate the secondary VM machine.

```
while not terminate:
  while not terminate:
        now = datetime.now()
        client = boto3.client("cloudwatch", region_name="us-east-1")
        response = client.get_metric_statistics(
```

```
        Namespace="AWS/EC2",
        MetricName="CPUUtilization",
        Dimensions=[
            {
                "Name": "InstanceId",
                "Value": "i-0534a7e84abc14fee",
            },
        ],
        StartTime=now - timedelta(seconds=600),
        EndTime=now,
        Period=60,
        Statistics=[
            "Average",
        ],
        Unit="Percent",
    )
    for k, v in response.items():
        if k == "Datapoints":
            for y in v:
                print(y["Average"])

                if y["Average"] > 50 and not secondary:
                    if how_many_times < 5:
                        how_many_times += 1

                    else:

                        # Terminate secondary VM
                        ec2.instances.filter(InstanceIds=[new_instance_id]).terminate()
                        terminate = True

                else:

                    how_many_times = 0
```

The entire code is shown below.

```
import boto3
import sys
import time
```

```
from datetime import datetime, timedelta

ec2 = boto3.resource("ec2", region_name="us-east-1")


secondary = False
how_many_times = 0


while True:

    now = datetime.now()
    client = boto3.client("cloudwatch", region_name="us-east-1")
    response = client.get_metric_statistics(
        Namespace="AWS/EC2",
        MetricName="CPUUtilization",
        Dimensions=[
            {"Name": "InstanceId", "Value": "i-0534a7e84abc14fee"},
        ],
        StartTime=now - timedelta(seconds=600),
        EndTime=now,
        Period=60,
        Statistics=[
            "Average",
        ],
        Unit="Percent",
    )

    for k, v in response.items():
        if k == "Datapoints":
            for y in v:
                print(y["Average"])

                new_instance_id = ""

                if y["Average"] > 50 and not secondary:
                    if how_many_times < 5:
                        how_many_times += 1

                    else:
                        how_many_times = 0

                        # Create secondary VM
                        tags = [
```

```
    {
        "ResourceType": "instance",
        "Tags": [{"Key": "Name", "Value": "secondary VM"}],
    }
]

new_instances = ec2.create_instances(
    ImageId="ami-0df435f331839b2d6",
    MinCount=1,
    MaxCount=1,
    InstanceType="t2.micro",
    KeyName="easy",
    TagSpecifications=tags,
)

new_instance_id = new_instances[0].id
print(f"New Instance ID: {new_instance_id}")

secondary = True
terminate = False

while not terminate:
    now = datetime.now()
    client = boto3.client("cloudwatch", region_name="us-east-1")
    response = client.get_metric_statistics(
        Namespace="AWS/EC2",
        MetricName="CPUUtilization",
        Dimensions=[
            {
                "Name": "InstanceId",
                "Value": "i-0534a7e84abc14fee",
            },
        ],
        StartTime=now - timedelta(seconds=600),
        EndTime=now,
        Period=60,
        Statistics=[
            "Average",
        ],
        Unit="Percent",
    )
    for k, v in response.items():
        if k == "Datapoints":
            for y in v:
```

```
            print(y["Average"])

            if y["Average"] > 50 and not secondary:
                if how_many_times < 5:
                    how_many_times += 1

                else:

                    # Terminate secondary VM
                    ec2.instances.filter(InstanceIds=[new_instance_id]).terminate()
                    terminate = True

            else:

                how_many_times = 0

time.sleep(5)
```