Lambda Function to Shut Down and Terminate an EC2 Instance

Level: **Intermediate**

Amazon EC2 AWS Lambda Amazon Web Services

English ▾

Your last attempt on **23-Nov-2024** View all

Required Points	10
Lab Duration	**00:30:00**
Average Start time	Less than a minute

Start Guided Lab →

Lab Overview Lab Steps

Cloud Architect, Cloud Developer, Cloud Administrator

Compute, Serverless

Lab Steps

Task 1: Sign in to AWS Management Console

1. Click on the **Open Console** button, and you will get redirected to AWS Console in a new browser tab.
2. On the AWS sign-in page,

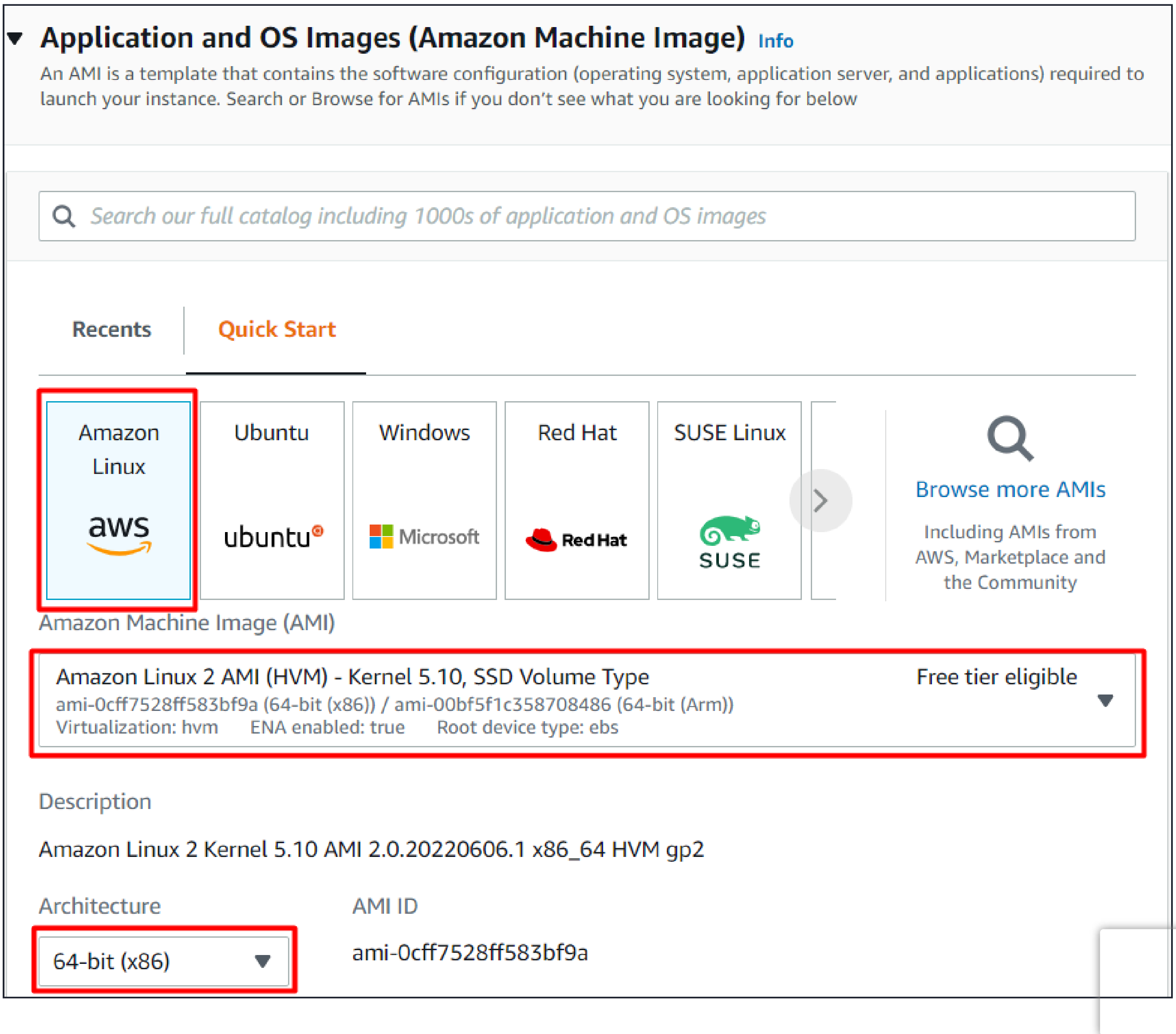
   - Leave the Account ID as default. Never edit/remove the 12-digit Account ID present in the AWS Console. Otherwise, you cannot proceed with the lab.

- Now copy your **Username** and **Password** in the Lab Console to the **IAM Username and Password** in AWS Console and click on the **Sign-in** button

3. Once Signed In to the AWS Management Console, make the default AWS Region as **US East (N. Virginia) us-east-1.**

Task 2: Launching two EC2 Instances

1. Make sure you are in the **US East (N. Virginia) us-east-1** Region.
2. Navigate to **EC2** by clicking on the **Services** menu at the top, then click on **EC2** in the **Compute** section.
3. Navigate to **Instances** on the left panel and click on **Launch instances** button.
4. Choose an Amazon Machine Image (AMI): Select **Amazon Linux 2 AMI** in the drop-down.
   - Choose **architecture** as **64-bit(x86)**



Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images
Recents
Quick Start
Amazon Linux
aws
Ubuntu
ubuntu
Windows
Microsoft
Red Hat
Red Hat
SUSE Linux
SUSE
Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type — Free tier eligible
ami-0cff7528ff583bf9a (64-bit (x86)) / ami-00bf5f1c358708486 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

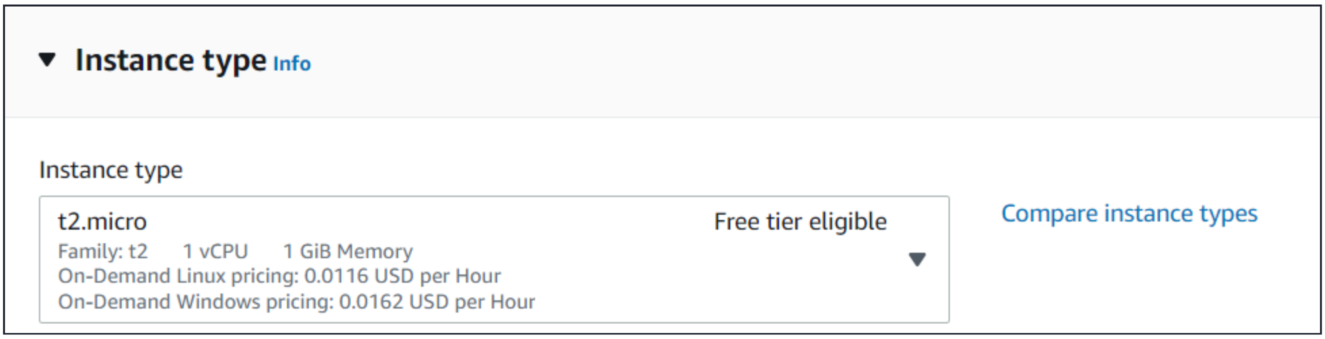
Description

Amazon Linux 2 Kernel 5.10 AMI 2.0.20220606.1 x86_64 HVM gp2

Architecture: 64-bit (x86)

AMI ID: ami-0cff7528ff583bf9a

5. Choose an **Instance Type**: Select **t2.micro.**



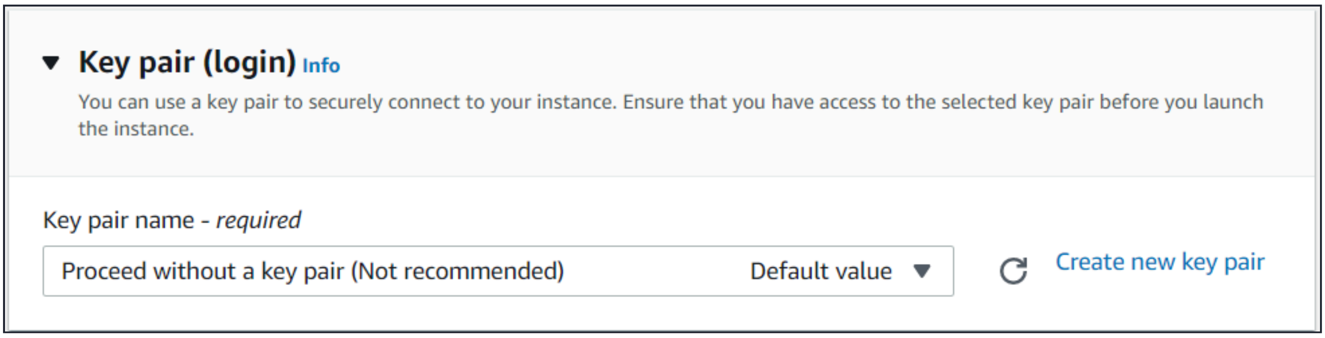
▼ Instance type Info

Instance type

t2.micro
Family: t2 1 vCPU 1 GiB Memory
On-Demand Linux pricing: 0.0116 USD per Hour
On-Demand Windows pricing: 0.0162 USD per Hour

Free tier eligible ▼

6. Key Pair: Select **Proceed without a Key Pair**



▼ Key pair (login) Info

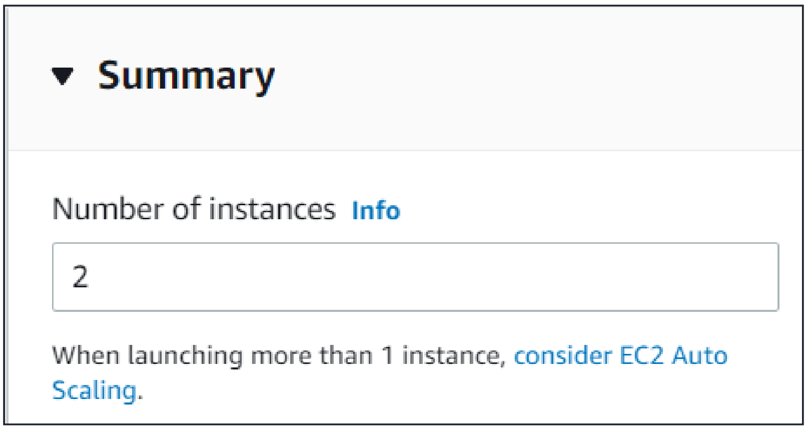
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*

Proceed without a key pair (Not recommended) Default value ▼

7. On the right side under the **Summary** section,

- Number of Instances: **2**



▼ Summary

Number of instances Info

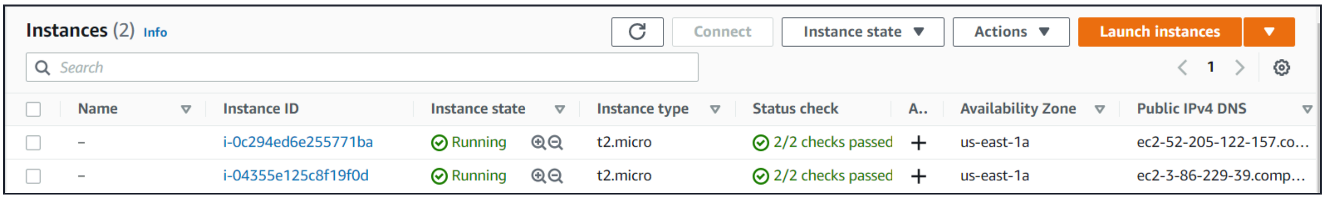
2

When launching more than 1 instance, consider EC2 Auto Scaling.

8. **Leave** the rest of the things as default.

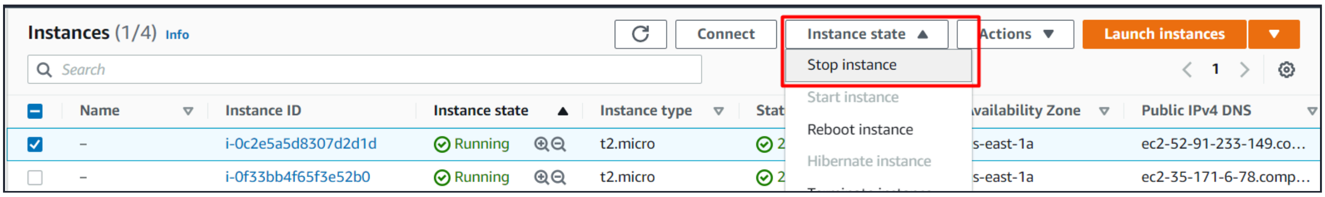
9. Click on the **Launch instance**

10. You can see all the running instances in the instances panel.



□	Name	Instance ID	Instance state	Instance type	Status check	A..	Availability Zone	Public IPv4 DNS
□	–	i-0c294ed6e255771ba	✓ Running	t2.micro	✓ 2/2 checks passed	+	us-east-1a	ec2-52-205-122-157.co...
□	–	i-04355e125c8f19f0d	✓ Running	t2.micro	✓ 2/2 checks passed	+	us-east-1a	ec2-3-86-229-39.comp...

11. Stop any one of the instance by clicking on the instance and then click on the **Instance State** button.



☐	Name	Instance ID	Instance state	Instance type	Stat	Availability Zone	Public IPv4 DNS
☑	–	i-0c2e5a5d8307d2d1d	✓ Running	t2.micro	✓ 2	s-east-1a	ec2-52-91-233-149.co...
□	–	i-0f33bb4f65f3e52b0	✓ Running	t2.micro	✓ 2	s-east-1a	ec2-35-171-6-78.comp...

Task 3: Create a Lambda Function

1. Go to **Services** menu, click on **Lambda** under the **Compute** section.
2. Make sure you are in the **US East (N. Virginia)** region.
3. Click on the **Create a function** button.
    - Choose **Author from Scratch** button.
    - Function name: Enter **myEC2LambdaFunction**
    - Runtime: Select **Python 3.8**
    - Permissions: click on the **Change default execution role** and choose **Use an existing role**
    - **Existing role:** Select **task102_role_<RANDOM_NUMBER>** from the dropdown list.
    - Click on **Create function** button.
4. **Configuration Page:** Here we need to configure our lambda function. If you scroll down, you can see the **Code source** section. Here, we need to write some **Python code** that will shut down and terminate an EC2 instance.
5. You will be using boto3 SDK for AWS to write the python code.
6. Remove the existing code in AWS lambda **lambda_function.py**. Copy the below code and paste it into your **lambda_function.py** file.

```
import json

import boto3
```

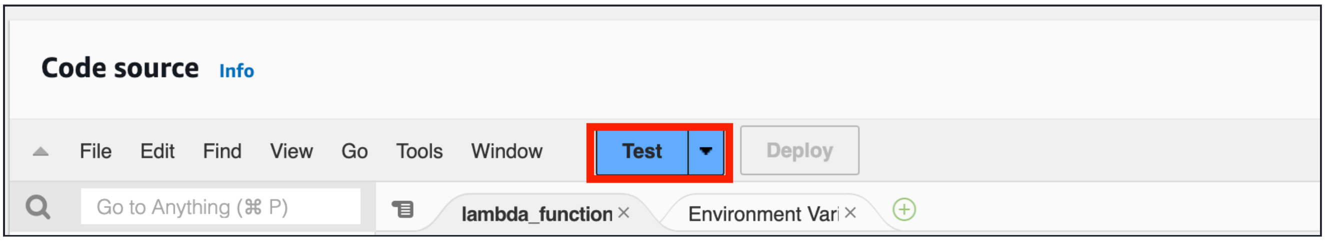
```
def lambda_handler(event, context):
    region = 'us-east-1'
    client = boto3.client("ec2", region_name=region)
    status = client.describe_instance_status(IncludeAllInstances = True)

    for i in status["InstanceStatuses"]:

        instaId = list(i["InstanceId"].split(" "))
        if i["InstanceState"]["Name"] == "running":
            print("Instances status : ", i["InstanceState"]["Name"])
            client.stop_instances(InstanceIds=instaId)
            print("Stopping the instance",i["InstanceId"])

        elif i["InstanceState"]["Name"] == "stopped":
            print("Instances status : ", i["InstanceState"]["Name"])
            client.terminate_instances(InstanceIds=instaId)
            print("Terminating the instance",i["InstanceId"])

        elif i["InstanceState"]["Name"] == "terminated":
            print("Terminated the instance",i["InstanceId"])

        else:
            print("Please wait for the instance to be stopped or running state")
        print("\n")
    return {
        'statusCode': 200,
    }
```

7. Save the function by clicking on **Deploy** button.

Task 4: Configure Test Event

In this task, we are going to configure a test event that allows users to manually trigger the Lambda function for testing purposes.

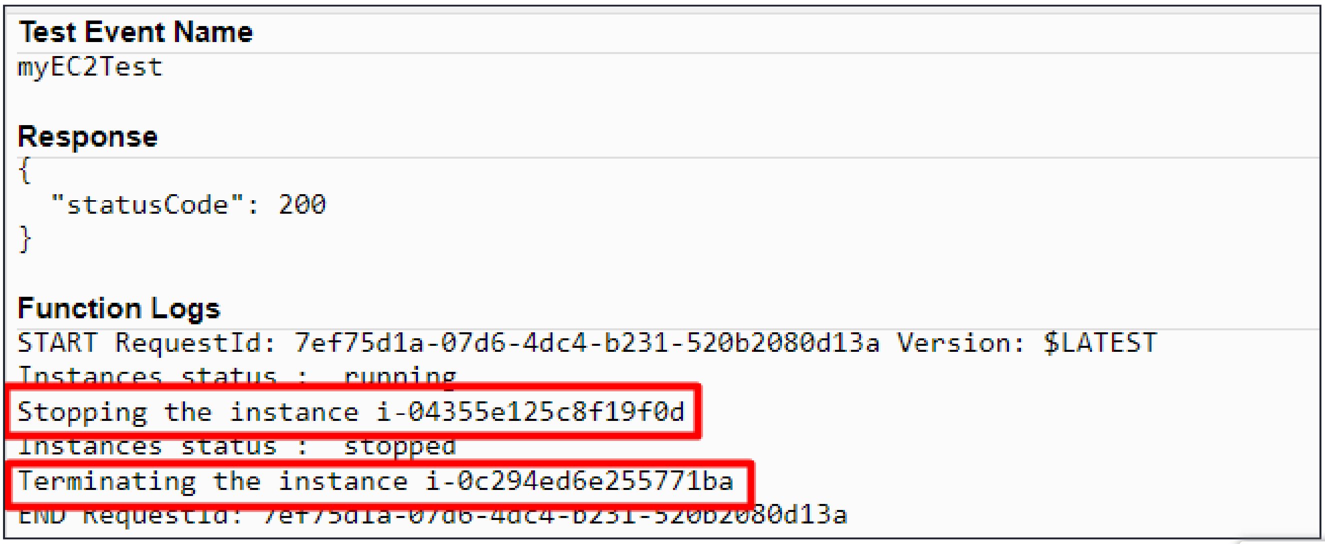
1. Click on the **Test** button.



2. On the **Configure test event** page,
   - Event Name: Enter **myEC2Test**
   - Leave other fields as default.
   - Click on **Save** button.

Task 5: Performing Stop and Terminate actions on EC2 Instances

1. Once the **EC2Test** is configured, we can trigger the lambda using a simple manual test.
2. Click on the **Test** button again.
3. The lambda function will be executed, the running EC2 instance will be stopped, and the stopped instance will be terminated.
4. Once it completes, you will be seeing a success message (as shown below).



Test Event Name

```
myEC2Test
```

Response

```
{
  "statusCode": 200
}
```

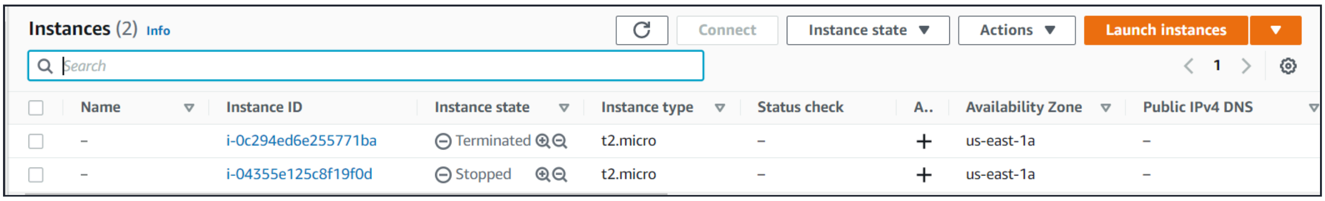
Function Logs

```
START RequestId: 7ef75d1a-07d6-4dc4-b231-520b2080d13a Version: $LATEST
Instances status :  running
Stopping the instance i-04355e125c8f19f0d
Instances status :  stopped
Terminating the instance i-0c294ed6e255771ba
END RequestId: 7ef75d1a-07d6-4dc4-b231-520b2080d13a
```

Task 6: Check the EC2 instance's status

In this task, we will check the status of the EC2 instances to verify that the Lambda function has correctly stopped and terminated the instances.

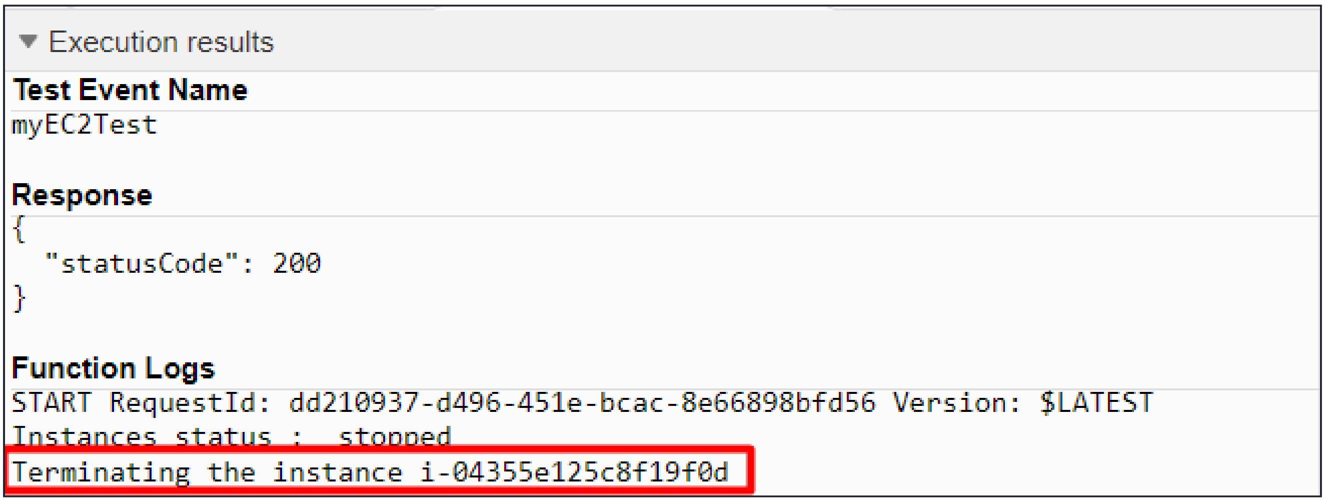
1. Navigate to the **EC2** page from the **services** menu.
2. Go to **Instances** in the left menu.
3. You can see that the running instance is stopped, and the stopped instance is terminated.



☐	Name	Instance ID	Instance state	Instance type	Status check	A..	Availability Zone	Public IPv4 DNS
☐	–	i-0c294ed6e255771ba	Terminated	t2.micro	–	+	us-east-1a	–
☐	–	i-04355e125c8f19f0d	Stopped	t2.micro	–	+	us-east-1a	–

Task 7: Performing Stop and Terminate action again

1. Navigate to the **Lambda** service again and select **myEC2LambdaFunction.**
2. Click on the **Test** button again.
3. The lambda function will be executed again and the stopped EC2 instance will be terminated.
4. Once it completes, you will be seeing a success message (as shown below).



▼ Execution results

Test Event Name

myEC2Test

Response

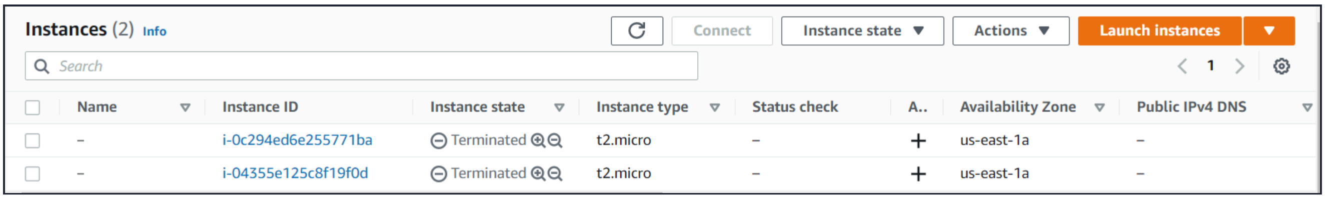
```
{
  "statusCode": 200
}
```

Function Logs

```
START RequestId: dd210937-d496-451e-bcac-8e66898bfd56 Version: $LATEST
Instances status :  stopped
Terminating the instance i-04355e125c8f19f0d
```

Task 8: Check the EC2 instance's status again

1. Navigate to the **EC2** page from the **services** menu.
2. Go to **Instances** in the left menu.
3. You can see that the stopped instance is terminated.



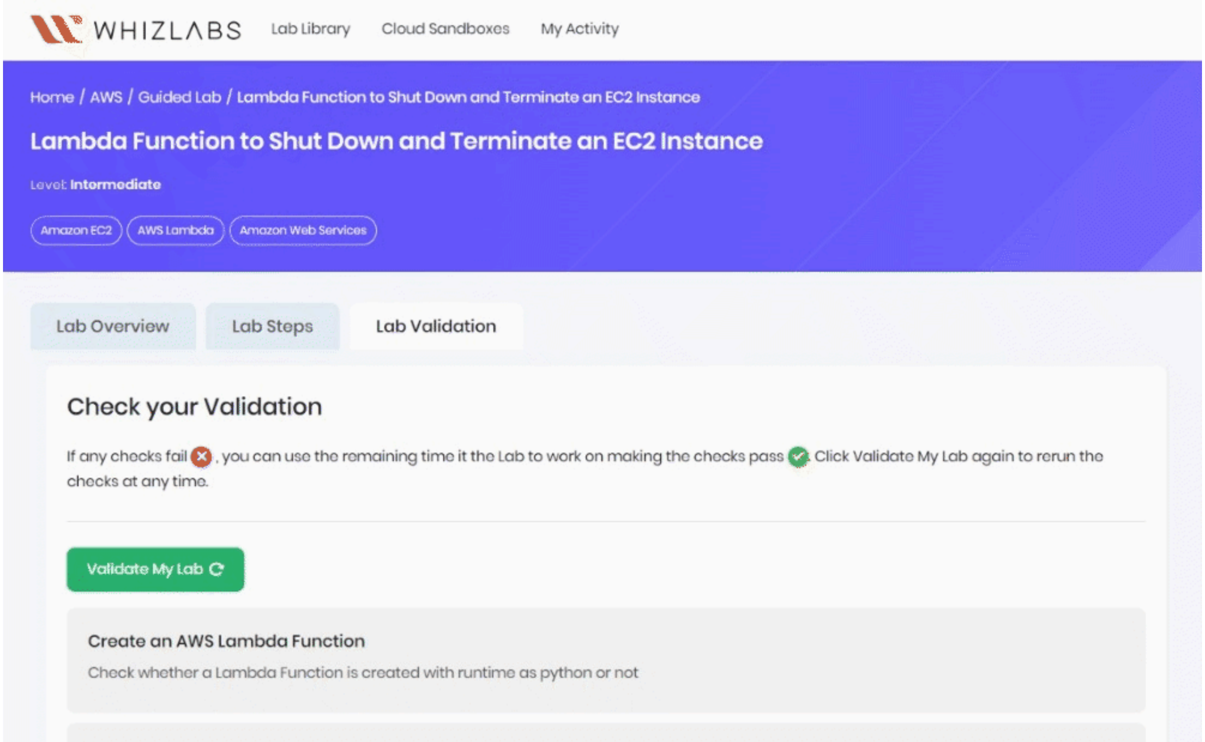
□	Name	Instance ID	Instance state	Instance type	Status check	A..	Availability Zone	Public IPv4 DNS
□	–	i-0c294ed6e255771ba	⊖ Terminated	t2.micro	–	+	us-east-1a	–
□	–	i-04355e125c8f19f0d	⊖ Terminated	t2.micro	–	+	us-east-1a	–

Do You Know?

Testing Lambda functions using test events is a valuable practice to ensure their correctness and efficiency before deploying them in production. It allows developers to identify and fix issues early in the development process, reducing the risk of errors in a live environment.

Task 9: Validation of the Lab

1. Once the lab steps are completed, please click on the **Validation** button on the left side panel.
2. This will validate the resources in the AWS account and displays whether you have completed this lab successfully or not.
3. Sample output :



Home / AWS / Guided Lab / Lambda Function to Shut Down and Terminate an EC2 Instance

Lambda Function to Shut Down and Terminate an EC2 Instance

Level: **Intermediate**

Check your Validation

If any checks fail ☒ , you can use the remaining time it the Lab to work on making the checks pass ✓ Click Validate My Lab again to rerun the checks at any time.

Create an AWS Lambda Function

Check whether a Lambda Function is created with runtime as python or not

Completion and Conclusion

1. You have created two EC2 Instances.

2. You have created a Lambda function with boto3 python code.
3. You have configured a test event and triggered it manually.
4. You have successfully shut down and terminated the EC2 instance.

End Lab

1. Sign out of the AWS Account.
2. You have successfully completed the lab.
3. Click on **End Lab** button from Whizlabs Labs console and wait till the process gets completed.