

School: CLOUDBOOSTA	Course: Cloud Computing & DevOps
Name: AKOSA SAMUEL ONYEJEKWE	Module: Serverless Computing & Database
	Class 6: Assignment 4 of 6_Serverless_1
Student No: CBALC0168	Issue Date: April 1, 2026

Tasks:

1. Launch an EC2 instance
2. Write Lambda function to stop EC2 instance
3. Create IAM role for Lambda permissions
4. Execute Lambda function to stop instance
5. Write Lambda function to start instance
6. Write Lambda function to terminate instance

Goal:

To use AWS Lambda functions to **stop, start, and terminate** an EC2 instance automatically.

Result:

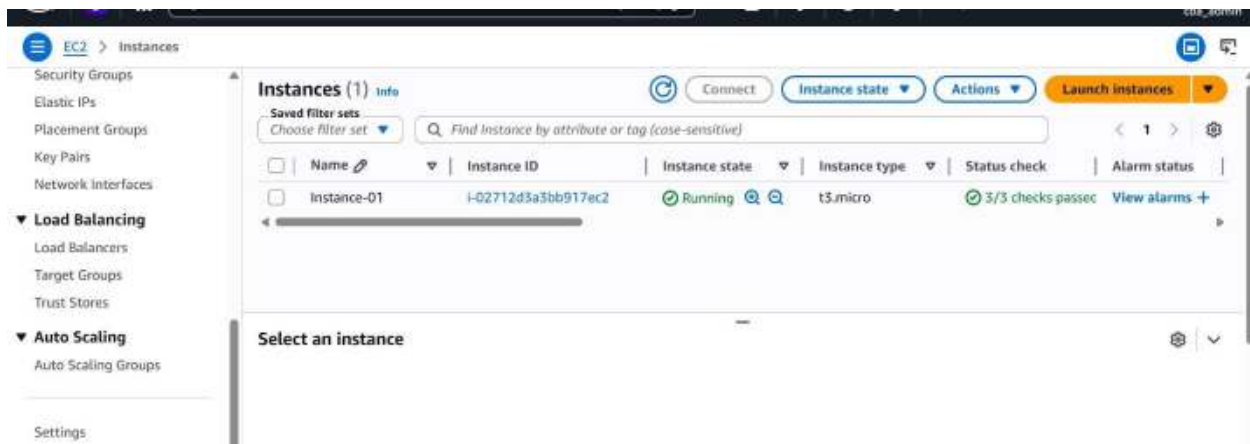
- EC2 instance successfully launched
- Lambda function successfully stopped the instance
- Lambda function successfully started the instance
- Lambda function successfully terminated the instance

```
import boto3

instances = ['i-02712d3a3bb917ec2']

ec2 = boto3.client('ec2', region_name='eu-west-3')

def lambda_handler(event, context):
    ec2.start_instances(InstanceIds=instances)
    print("started the running instance " + str(instances))
```



```
import boto3

instances = ['i-02712d3a3bb917ec2']

ec2 = boto3.client('ec2', region_name='eu-west-3')

def lambda_handler(event, context):
    ec2.stop_instances(InstanceIds=instances)
    print("stopped the running instance " + str(instances))
```



```
import boto3

instances = ['i-02712d3a3bb917ec2']

ec2 = boto3.client('ec2', region_name='eu-west-3')

def lambda_handler(event, context):
    ec2.terminate_instances(InstanceIds=instances)
    print("terminated the running instance " + str(instances))
```



Evidence from screenshots:

- EC2 instance initially running (page 6)
- Instance stopped after Lambda execution (page 16)
- Instance started again successfully (page 18)
- Instance terminated successfully (page 20)

Action:

• Step 1: Logged into AWS and Launched EC2 Instance

- Accessed AWS console
- Launched EC2 instance
- Copied **Instance ID** for later use

Evidence:

- EC2 instance details showing running state (page 6)
- Instance ID captured (page 8)

- **Step 2: Created Lambda Function (Stop EC2)**

- Created Lambda function from scratch
- Used Python runtime
- Implemented code using **boto3** to stop EC2

Code logic used:

- Import boto3
- Define instance ID
- Call `stop_instances()`

Evidence:

- Lambda function creation (page 7)
- Stop function code shown (page 9)

- **Step 3: Encountered Permission Error**

- Initial execution failed due to missing IAM permissions

Evidence:

- Error shown during execution (page 11)

- **Step 4: Created IAM Role and Attached Permissions**

- Navigated to Lambda configuration
- Created execution role
- Attached policy:
 - **AmazonEC2FullAccess**

Evidence:

- IAM role configuration (page 11-12)
- Policy attached successfully (page 12)

- **Step 5: Stopped EC2 Instance Using Lambda**

- Ran Lambda test event
- Execution successful (response = null)
- EC2 instance changed from running → stopped

Evidence:

- Successful test execution (page 15)
- Instance stopped state (page 16)

- **Step 6: Started EC2 Instance Using Lambda**

- Updated Lambda code to use `start_instances()`
- Executed test
- Instance returned to running state

Evidence:

- Start function code (page 13)
- Instance running again (page 18)

- **Step 7: Terminated EC2 Instance Using Lambda**

- Updated Lambda code to use `terminate_instances()`
- Executed test
- Instance terminated successfully

Evidence:

- Terminate function code (page 14)
- Instance terminated state (page 20)

Screenshots follow.

Summary

- Lambda functions were used to control EC2 lifecycle
- IAM role was required for permissions
- Three operations were successfully implemented:
 - Stop instance
 - Start instance
 - Terminate instance
- System validated using Lambda test executions and EC2 dashboard

Lab objective achieved:

“Start, Stop and Terminate an EC2 Instance using Lambda functions.”

EC2 > Instances

Security Groups

Elastic IPs

Placement Groups

Key Pairs

Network Interfaces

▼ Load Balancing

Load Balancers

Target Groups

Trust Stores

▼ Auto Scaling

Auto Scaling Groups

Settings

Instances (1) Info

Connect

Instance state ▼

Actions ▼

Launch instances ▼

Saved filter sets

Choose filter set ▼

Find Instance by attribute or tag (case-sensitive)

< 1 >

☐	Name	Instance ID	Instance state ▼	Instance type ▼	Status check	Alarm status
☐	Instance-01	i-02712d3a3bb917ec2	Running	t3.micro	3/3 checks passed	View alarms +

Select an instance

Create function [Info](#)

Choose one of the following options to create your function.

☒ Author from scratch
Start with a simple Hello World example.

☐ Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.

☐ Container image
Select a container image to deploy for your function.

Basic information

Function name [Info](#)

Enter a name that describes the purpose of your function.

Function name must be 1 to 64 characters, must be unique to the Region, and can't include spaces. Valid characters are a-z, A-Z, 0-9, hyphens (-), and underscores (_).

Runtime [Info](#)

Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby.

Architecture [Info](#)

Choose the instruction set architecture you want for your function code.

☐ arm64

☒ x86_64

Permissions [Info](#)

By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.

▶ Change default execution role

▶ Additional configurations

Use additional configurations to set up networking, security, and governance for your function. These settings help secure and customize your Lambda function deployment.

Cancel

Create function

CloudShell

Feedback

Console Mobile App

© 2026, Amazon Web Services, Inc. or its affiliates.

[Privacy](#)

[Terms](#)

[Cookie preferences](#)

Successfully created the function "stop-ec2-function". You can now change its code and configuration. To invoke your function with a test event, choose "Test".

stop-ec2-function

Throttle

Copy ARN

Actions

Function overview

Diagram

Template



stop-ec2-function



Layers

(0)

+ Add trigger

+ Add destination

Export to Infrastructure Composer

Download

Function ARN

arn:aws:lambda:eu-west-3:009850210027:function:stop-ec2-function

Description

-

Last modified

6 minutes ago

Code Test Monitor Configuration Aliases Versions

Code source

Open in Visual Studio Code

Upload from

▼

Code editor showing the lambda_function.py file with the following code:

```
1 import boto3
2
3 instances = ['i-02712d3a3bb917ec2'] # my EC2 instance ID
4
5 ec2 = boto3.client('ec2', region_name='eu-west-3')
6
7 def lambda_handler(event, context):
8     ec2.stop_instances(InstanceIds=instances)
9     print("stopped the running instance " + str(instances))
```

Left sidebar shows the Explorer view with the following structure:

- STOP-EC2-FUNCTION
 - lambda_function.py
- DEPLOY (Undeployed)
 - Deploy (Ctrl+Shift+U)
 - Test (Ctrl+Shift+I)
- TEST EVENTS [NONE SELECTED]
 - + Create new test event

Bottom status bar shows: Undeployed Changes, 0 errors, 0 warnings, Amazon QL, Ln 9, Col 60, Spaces: 4, UTF-8, LF, Python, Lambda, Layout: US.

Code properties

Package size
305 byte

SHA256 hash

fyxzT08Fj/GWsOZujgHy45p6qwpvDvPQktarFwZ6B0s=

Last modified

6 minutes ago

Runtime settings

Edit

Edit runtime management configuration

Runtime
Python 3.14

Handler

lambda_function.lambda_handler

Architecture

x86_64

Runtime management configuration

Runtime version ARN

arn:aws:lambda:eu-west-3::runtime:027e67fdaf9cea74675a8052be97d7232acebbdfdde682b84171030a9773437f

Update runtime version

Auto

Layers (0)

Last fetched 4/7/2026, 1:22:45 AM

Edit

Search by attributes or search by keyword

< 1 > ⚙

Merge order	Name	Layer version	Compatible runtimes	Compatible architectures	Version ARN
There is no data to display.					

Successfully created the function "stop-ec2-function". You can now change its code and configuration. To invoke your function with a test event, choose "Test".



DEPLOY ✓ Current

Deploy (Ctrl+Shift+U)

Test (Ctrl+Shift+I)

TEST EVENTS [NONE SELECTED]

+ Create new test event

```
4
5  ec2 = boto3.client('ec2', region_name='eu-west-3')
6
7  def lambda_handler(event, context):
8      ec2.stop_instances(InstanceIds=instances)
9      print("stopped the running instance " + str(instances))
```

Successfully updated the function stop-ec2-function.

Attach policy to stop-ec2-function-role-08epkeik

▶

Current permissions policies (1)

Other permissions policies (1/1143)

Filter by Type

All types

1 match

<

1

>

☒

Policy name

▲

☒

Type

▼

☒

Description

☒

AmazonEC2FullAccess

AWS managed

Provides full access to Amazon EC2 via t...

Cancel

Add permissions

 IAM > Roles > stop-ec2-function-role-08epkeik

Identity and Access Management (IAM)

Dashboard

▼ Access Management

User groups

Users

Roles

Policies

Identity providers

Account settings

Root access management

Temporary delegation requests

✔ Policy was successfully attached to role.

✕

stop-ec2-function-role-08epkeik [Info](#)

[Delete](#)

Summary [Edit](#)

Creation date
April 07, 2026, 01:09 (UTC+03:00)

ARN
 arn:aws:iam::009850210027:role/service-role/stop-ec2-function-role-08epkeik

Last activity
-

Maximum session duration
1 hour

- Permissions
- Trust relationships
- Tags
- Last Accessed
- Revoke sessions

Permissions policies (2) [Info](#)

 [Simulate](#) [Remove](#) [Add permissions](#)

You can attach up to 10 managed policies.

Filter by Type
All types

< 1 > 

☐	Policy name ↗	Type	Attached entities
☐	  AmazonEC2FullAccess	AWS managed	1
☐	 AWSLambdaBasicExecutionRo...	Customer managed	1

► Permissions boundary (not set)

▼ Generate policy based on CloudTrail events

You can generate a new policy based on the access activity for this role, then customize, create, and attach it to this role. AWS uses your CloudTrail events to identify the services and actions used and generate a policy. [Learn more](#)

[Generate policy](#)

No requests to generate a policy in the past 7 days.

✓ The test event "test" was successfully saved.

stop-ec2-function

[Throttle](#)[Copy ARN](#)[Actions](#)

Function overview [Info](#)

[Diagram](#)[Template](#)**stop-ec2-function**

Layers

(0)

[+ Add trigger](#)[+ Add destination](#)[Export to Infrastructure Composer](#)[Download](#)

Function ARN

arn:aws:lambda:eu-west-3:009850210027:function:stop-ec2-function

Description

-

Last modified

43 minutes ago

[Code](#) [Test](#) [Monitor](#) [Configuration](#) [Aliases](#) [Versions](#)

Code source [Info](#)

[Open in Visual Studio Code](#)[Upload from](#)

EXPLORE

STOP-EC2-FUNCTION

lambda_function.py

DEPLOY

Deploy (Ctrl+Shift+U)

Test (Ctrl+Shift+I)

TEST EVENTS [SELECTED: TEST]

Create new test event

Private saved events

Lambda Deployed

0 0 0

Amazon Q

stop-ec2-function

lambda_function.py

Find

1 import boto3

2

3 instances = ['i-02712d3a3bb917ec2'] # my EC2

4

5 ec2 = boto3.client('ec2', region_name='eu-west

6

7 def lambda_handler(event, context):

8 ec2.stop_instances(InstanceIds=instances)

9 print("stopped the running instance " + st

Create new test event

Invoke

Save

Event Name

test

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Invocation type

Synchronous

Executes the Lambda function and blocks until receiving the function's response with a maximum timeout of 15 minutes. Returns function output or error details directly to the calling application.

Test event is saved successfully.

request ID. Function processes independently, with results optionally sent to a

No changes to deploy

Lambda

Layout: US

Code properties [Info](#)

Package size
325 byte

SHA256 hash

gxx6FEP6SXEGsNSbEsAaaeGP6WzhkFAQtSvCR18
hSPM=

Last modified

43 minutes ago

Runtime settings [Info](#)

[Edit](#)[Edit runtime management configuration](#)

Runtime
Python 3.14

Handler [Info](#)

lambda_function.lambda_handler

Architecture [Info](#)

x86_64

Runtime management configuration

Runtime version ARN [Info](#)

arn:aws:lambda:eu-west-3::runtime:027e67fdaf9cea74675a8052be97d7232acebbdf
dde682b84171030a9773437f

Update runtime version [Info](#)

Auto

Layers (0) [Info](#)

Last fetched 4/7/2026, 2:06:52 AM

[Edit](#)

< 1 >



Merge order	Name	Layer version	Compatible runtimes	Compatible architectures	Version ARN
There is no data to display.					



CloudShell

[Feedback](#)



Console Mobile App

© 2026, Amazon Web Services, Inc. or its affiliates.

[Privacy](#)

[Terms](#)

[Cookie preferences](#)

Lambda > Functions > stop-ec2-function

The test event "test" was successfully saved.

lambda_function.py

DEPLOY

Deploy (Ctrl+Shift+U)

Test (Ctrl+Shift+I)

TEST EVENTS [SELECTED: TEST]

+ Create new test event

Private saved events

1 import boto3

2

3 instances = ['i-02712d3a3bb917ec2'] # my EC2

4

5 ec2 = boto3.client('ec2', region_name='eu-west-1')

6

7 def lambda_handler(event, context):

8 ec2.stop_instances(InstanceIds=instances)

PROBLEMS

OUTPUT

CODE REFERENCE LOG

TERMINAL

Status: Succeeded

Test Event Name: test

Response:

null

Create new test event

Event Name

test

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Invocation type

Synchronous

Executes the Lambda function and blocks until receiving the function's response.

Execution Results

Test event is saved successfully.

No changes to deploy

Lambda Deployed

0 0 0

Amazon Q

Lambda

Layout: US

Instances (1) Info



Connect

Instance state

Actions

Launch instances

Saved filter sets

Choose filter set

Find Instance by attribute or tag (case-sensitive)

< 1 > Settings

☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
☐	Instance-01	i-02712d3a3bb917ec2	Stopped	t3.micro	-	View alarms

Successfully updated the function "stop-ec2-function".

Notifications 0 0 2 0 0

```
6  
7 lambda_handler(event, context):  
8     ec2.start_instances(InstanceIds=instances)  
9     print("started the running instance " + str(
```

Event Name

test

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Invocation type

☒ Synchronous

Executes the Lambda function and blocks until receiving the function's response.

PROBLEMS OUTPUT CODE REFERENCE LOG TERMINAL

Status: Succeeded
Test Event Name: test

Response:
null

Execution Results

Successfully updated the function stop-ec2-function.

aws

Search

[Alt+S]

Europe (Paris)

cba_admin

EC2 > Instances

Instances (1) Info

Connect

Instance state

Actions

Launch instances

Saved filter sets

Choose filter set

Find Instance by attribute or tag (case-sensitive)

Name

Instance ID

Instance state

Instance type

Status check

Alarm status

Instance-01

i-02712d3a3bb917ec2

Running

t3.micro

Initializing

View alarms

Lambda > Functions > stop-ec2-function

Successfully updated the function "stop-ec2-function".

Notifications 0 0 2 0 0

lambda_function.py

6
7 def lambda_handler(event, context):
8 ec2.terminate_instances(InstanceIds=instan
9 print("terminated the running instance " .

DEPLOY Current

Deploy (Ctrl+Shift+U)

Test (Ctrl+Shift+I)

TEST EVENTS [SELECTED: TEST]

+ Create new test event

Private saved events

Lambda Deployed 0 0 Amazon O

Find

6
7 def lambda_handler(event, context):
8 ec2.terminate_instances(InstanceIds=instan
9 print("terminated the running instance " .

PROBLEMS OUTPUT CODE REFERENCE LOG TERMINAL

Status: Succeeded

Test Event Name: test

Response:
null

Event Name

test

Maximum of 25 characters consisting of letters, numbers, dots, hyphens and underscores.

Invocation type

Synchronous

Executes the Lambda function and blocks until receiving the function's response.

Execution Results

Successfully updated the function stop-ec2-function.

Ln 9, Col 63 Spaces: 4 UTF-8 LF Python Lambda Layout: US

- Dashboard
- AWS Global View
- Events

Connect

Instance state ▾

Actions ▾

Launch instances ▾

Choose filter set ▼

< 1 > 

[View alarms +](#)

```
import boto3
```

```
instances = ['i-02712d3a3bb917ec2']
```

```
ec2 = boto3.client('ec2', region_name='eu-west-3')
```

```
def lambda_handler(event, context):  
    ec2.start_instances(InstanceIds=instances)  
    print("started the running instance " + str(instances))
```

```
import boto3

instances = ['i-02712d3a3bb917ec2']

ec2 = boto3.client('ec2', region_name='eu-west-3')

def lambda_handler(event, context):
    ec2.stop_instances(InstanceIds=instances)
    print("stopped the running instance " + str(instances))
```

```
import boto3

instances = ['i-02712d3a3bb917ec2']

ec2 = boto3.client('ec2', region_name='eu-west-3')

def lambda_handler(event, context):
    ec2.terminate_instances(InstanceIds=instances)
    print("terminated the running instance " + str(instances))
```