

School: CLOUDBOOSTA	Course: Cloud Computing & DevOps
Name: AKOSA SAMUEL ONYEJEKWE	Module: Serverless Computing & Database
	Class 6: Assignment 5 of 6_ Serverless_2
Student No: CBALC0168	Issue Date: April 1, 2026

Tasks:

1. Provision an EC2 instance using AWS Lambda
2. Create IAM role for Lambda with EC2 permissions
3. Write and deploy Lambda function code
4. Test Lambda function and verify EC2 instance creation
5. Create Lambda function to stop/start EC2 dynamically
6. Schedule automation using Amazon EventBridge

Goal:

To automate the provisioning, management, and scheduling of EC2 instances using AWS Lambda and EventBridge.

Result:

- EC2 instance successfully provisioned using Lambda
- Instance tagged with **AutoSchedule=True**
- Instance state dynamically controlled (start/stop)
- EventBridge scheduling configured using cron expressions
- Automation verified through Lambda logs and EC2 status

```

import boto3

def lambda_handler(event, context):
    ec2 = boto3.client("ec2", region_name="eu-west-3") # Updated region

    response = ec2.run_instances(
        ImageId="ami-0d8c303e6b2e300dd", # Your AMI
        InstanceType="t3.micro",
        KeyName="cba_keypair", # Your key pair
        MinCount=1,
        MaxCount=1,
        TagSpecifications=[
            {
                "ResourceType": "instance",
                "Tags": [
                    {"Key": "AutoSchedule", "Value": "True"},
                    {"Key": "Name", "Value": "lambda-launched-instance"}
                ]
            }
        ]
    )

    instance_id = response["Instances"][0]["InstanceId"]

    return {
        "statusCode": 200,
        "message": f"EC2 Instance {instance_id} launched successfully in eu-west-3!"
    }

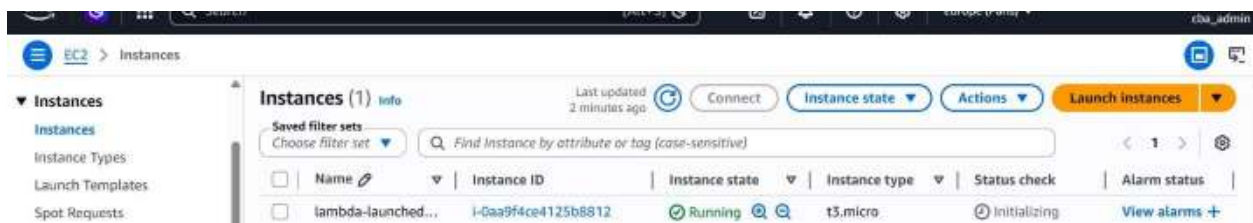
```

```

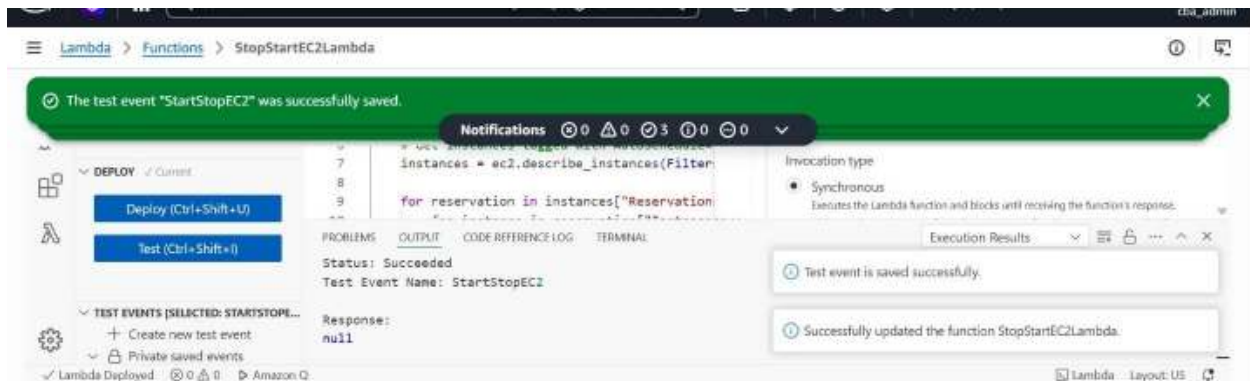
    "message": "EC2 Instance i-0aa9f4ce4125b8812 launched successfully in eu-west-3!"
}

```

The area below shows the last 4 KB of the execution log.



Name	Instance ID	Instance state	Instance type	Status check	Alarm status
lambda-launched...	i-0aa9f4ce4125b8812	Running	t3.micro	Initializing	View alarms +



Evidence from the screenshots:

- EC2 instance running after Lambda execution (page 18)
- Instance tagged with AutoSchedule=True (page 19)
- Instance stopped successfully after second Lambda execution (page 27)
- EventBridge schedule configured (page 28)

Action:

- **Step 1: Created IAM Role for Lambda**
 - Created IAM role: **LambdaEC2ProvisionRole**
 - Attached policies:
 - AmazonEC2FullAccess
 - AWSLambdaBasicExecutionRole

As shown in:

- IAM role creation steps (page 8-10)
- Role successfully created (page 5)

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

- Function Name: **ProvisionEC2Lambda**
- Runtime: Python 3.14
- Execution role: LambdaEC2ProvisionRole

Evidence:

- Function creation screen (page 6)
- Configuration details shown (page 12)

• Step 3: Implemented Lambda Code

- Used **boto3** to launch EC2 instance
- Configured:
 - AMI ID
 - Instance type (t3.micro)
 - Key pair
 - Region
- Added tag:
 - **AutoSchedule = True**

Evidence:

- Code implementation (page 10)
- Code instructions (page 13-14)

• Step 4: Tested Lambda Function

- Created test event: **TestProvisionEC2**
- Executed function successfully
- EC2 instance launched and verified

Evidence:

- Successful test execution (page 16)
- Instance running in EC2 dashboard (page 18)

• Step 5: Created Stop/Start Lambda Function

- Function Name: **StopStartEC2Lambda**
- Logic:
 - Identify instances with tag **AutoSchedule=True**
 - Stop running instances
 - Start stopped instances

Evidence:

- Function creation (page 22)
- Successful execution output (page 26)

• Step 6: Scheduled Automation Using EventBridge

- Created rule: **EC2ScheduleRule**
- Schedule type: Cron expression

Examples used:

- `0 8 * * ? * → Start instance daily`
- `0 18 * * ? * → Stop instance daily`

Evidence:

- EventBridge schedule configuration (page 28)
- Cron explanation (page 22-23)

• Step 7: Verification

- Verified Lambda execution via logs
- Verified EC2 instance state changes

Evidence:

- Instance stopped state (page 27)
- Verification steps described (page 24)

Screenshots follow.

Summary

- Lambda was used to **provision EC2 instances automatically**
- EC2 instances were tagged for dynamic management
- A second Lambda function handled **start/stop operations**
- EventBridge enabled **scheduled automation using cron expressions**
- System was verified using:
 - EC2 dashboard
 - Lambda execution results
 - CloudWatch logs

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

Enter a name that describes the purpose of your function.

ProvisionEC2Lambda

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.

Python 3.14 ▾

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

Execution role

To access other AWS services and resources your function needs an IAM role. Choose a role with the right permissions for your function.

☐ Create default role

☒ Use another role

LambdaEC2ProvisionRole ▾



Create new role

[View role details in IAM](#)

► 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

Quick Start

 [Browse more AMIs](#)

Including AMIs from
AWS, Marketplace and
the Community

Number of instances [Info](#)

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.10....[read more](#)
ami-0d8c303e6b2e300dd

Software Image (AMI)

Amazon Linux 2023 AMI 2023.10....[read more](#)
ami-0d8c303e6b2e300dd

aws

Search

[Alt+S]

Europe (Paris)

cba_admin

EC2

> Key pairs

▼ Instances

Instances

Instance Types

Launch Templates

Spot Requests

Key pairs (1/1) Info

Find Key Pair by attribute or tag

Actions

Create key pair

☒	Name	Type	Created	Fingerprint	ID
☒	cba_keypair	rsa	2026/03/28 16:32 GMT+2	df:8c:7c:c3:b4:15:fe:0e:fa:0...	key-022b138b028686eb4

```
import boto3
```

```
def lambda_handler(event, context):
```

```
    ec2 = boto3.client("ec2", region_name="eu-west-3") # Updated region
```

```
    response = ec2.run_instances(
```

```
        ImageId="ami-0d8c303e6b2e300dd", # Your AMI
```

```
        InstanceType="t3.micro",
```

```
        KeyName="cba_keypair", # Your key pair
```

```
        MinCount=1,
```

```
        MaxCount=1,
```

```
        TagSpecifications=[
```

```
            {
```

```
                "ResourceType": "instance",
```

```
                "Tags": [
```

```
                    {"Key": "AutoSchedule", "Value": "True"},
```

```
                    {"Key": "Name", "Value": "lambda-launched-instance"}]
```

```
            ]
```

```
        }
```

```
    ]
```

```
)
```

```
    instance_id = response["Instances"][0]["InstanceId"]
```

```
    return {
```

```
        "statusCode": 200,
```

```
        "message": f"EC2 Instance {instance_id} launched successfully in eu-west-3!"
```

```
    }
```

Successfully updated the function "ProvisionEC2Lambda".

ProvisionEC2Lambda

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

Function overview

[Diagram](#)[Template](#)**ProvisionEC2Lambda**

Layers

(0)

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

arn:aws:lambda:eu-west-3:009850210027:function:ProvisionEC2Lambda

Description

-

Last modified

15 seconds ago

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

Code source

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

EXPLORE

PROVISIONEC2LAMBDA

lambda_function.py

DEPLOY

Deploy (Ctrl+Shift+U)

Test (Ctrl+Shift+I)

TEST EVENTS [NONE SELECTED]

Create new test event

Lambda Deployed

0 0 0

Amazon Q

lambda_function.py X

lambda_function.py

```
3 def lambda_handler(event, context):
6     response = ec2.run_instances(
12         TagSpecifications=[
13             {
19             }
20         ]
21     )
22
23     instance_id = response["Instances"][0]["InstanceId"]
24
25     return {
26         "statusCode": 200,
27         "message": f"EC2 Instance {instance_id} launched successfully in eu-west-3!"
28     }
```

Successfully updated the function ProvisionEC2Lambda.

Ln 28, Col 6 Spaces: 4 UTF-8 LF Python Lambda Layout: US

Code properties

Package size
546 byte

SHA256 hash

7c8+gA5ftSU0MO0KzDt9jjwvrnhtNtc0QU3wyg8j
fSo=

Last modified

16 seconds 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:027e67fdaf9cea74675a8052be97d7232acebbdf
dde682b84171030a9773437f

Update runtime version

Auto

Layers (0)

Last fetched 4/7/2026, 5:58:43 PM

[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](#)

Edit basic settings

Basic settings [Info](#)

Description - optional

Memory [Info](#)

Your function is allocated CPU proportional to the memory configured.

128 MB

Set memory to between 128 MB and 10240 MB.

Ephemeral storage [Info](#)

You can configure up to 10 GB of ephemeral storage (/tmp) for your function. [View pricing](#)

512 MB

Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.

SnapStart [Info](#)

Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the [SnapStart compatibility considerations](#). For Python and .NET runtimes, [view pricing](#).

None

Supported runtimes: .NET 10 (C#/F#/PowerShell), .NET 8 (C#/F#/PowerShell), Java 11, Java 17, Java 21, Java 25, Python 3.12, Python 3.13, Python 3.14.

Timeout

1 min 0 sec

Execution role

To access other AWS services and resources your function needs an IAM role. Choose a role with the right permissions for your function.

LambdaEC2ProvisionRole



Create new role

[View role details in IAM](#)

Cancel

Save

Lambda > Functions > ProvisionEC2Lambda

cba_admin

Successfully updated the function "ProvisionEC2Lambda".

+ Add trigger

+ Add destination

Last modified
50 seconds ago

CodeTestMonitorConfigurationAliasesVersions

Configuration

General configuration

Triggers

Permissions

Destinations

General configuration

Description

Timeout

Memory

SnapStart

Ephemeral storage

Edit

Successfully updated the function "ProvisionEC2Lambda".

Code source [Info](#)

[Open in Visual Studio Code](#)

[Upload from](#)

ProvisionEC2Lambda

EXPLORER

PROVISIONEC2LAMBDA

- lambda_function.py

DEPLOY Undeployed

Deploy (Ctrl+Shift+U)

Test (Ctrl+Shift+I)

lambda_function.py

```
3 def lambda_handler(event, context):
6     response = ec2.run_instances(
12         TagSpecifications=[
13             {
19             }
20         ]
21     )
22
23     instance_id = response["Instances"][0]["I
24
25     return {
```

Create new test event

Invoke Save

Event Name

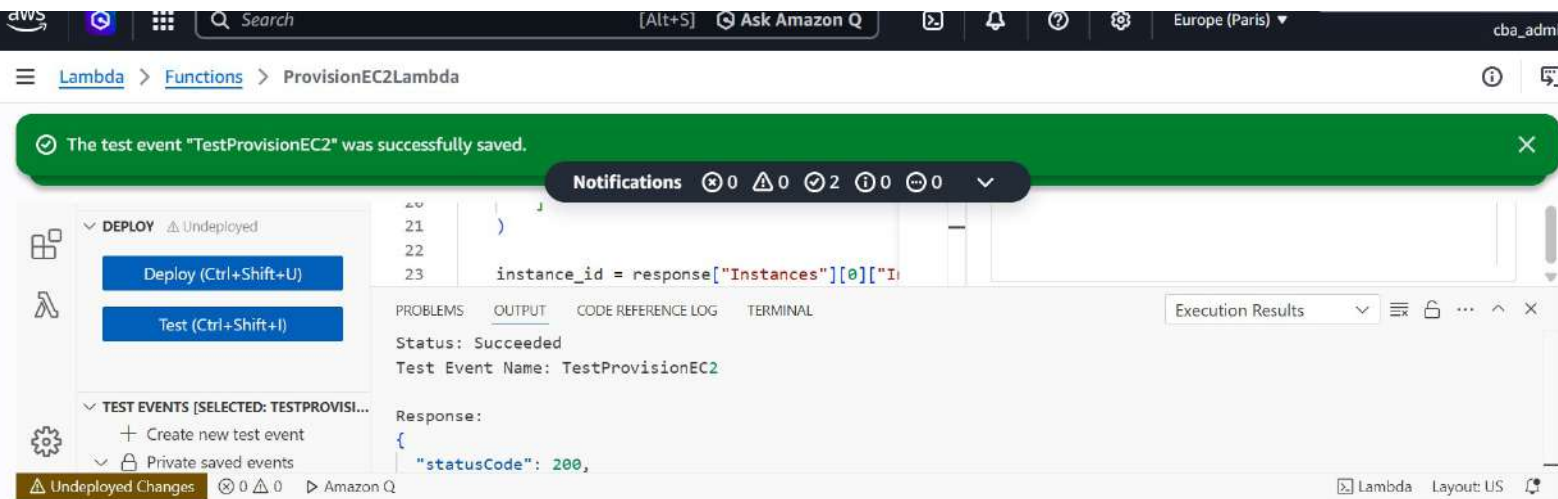
TestProvisionEC2

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.



```
"message": "EC2 Instance i-0aa9f4ce4125b8812 launched successfully in eu-west-3!"  
}
```

The area below shows the last 4 KB of the execution log.

EC2 > Instances

Instances

Instance Types

Launch Templates

Spot Requests

Instances (1) Info

Last updated 2 minutes ago

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
	lambda-launched...	i-0aa9f4ce4125b8812	Running	t3.micro	Initializing	View alarms

- Instances
 - Instance Types
 - Launch Templates
 - Spot Requests
 - Savings Plans
 - Reserved Instances
 - Dedicated Hosts
 - Capacity Reservations
 - Capacity Manager
- Images
 - AMIs
 - AMI Catalog

Instances (1/1) Info

Last updated 5 minutes ago

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
☒	lambda-launched...	i-0aa9f4ce4125b8812	Running	t3.micro	Initializing	View alarms +

i-0aa9f4ce4125b8812 (lambda-launched-instance)

Key	Value
AutoSchedule	True
Name	lambda-launched-instance

 The test event "TestProvisionEC2" was successfully saved.

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

Enter a name that describes the purpose of your function.

StopStartEC2Lambda

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.

Python 3.14 

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

Execution role

To access other AWS services and resources your function needs an IAM role. Choose a role with the right permissions for your function.

☐ Create default role

☒ Use another role

LambdaEC2ProvisionRole 



Create new role

[View role details in IAM](#) 

► 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

 The test event "TestProvisionEC2" was successfully saved.



Edit basic settings

Basic settings [Info](#)

Description - optional

Memory [Info](#)

Your function is allocated CPU proportional to the memory configured.

128

 MB

Set memory to between 128 MB and 10240 MB.

Ephemeral storage [Info](#)

You can configure up to 10 GB of ephemeral storage (/tmp) for your function. [View pricing](#)

512

 MB

Set ephemeral storage (/tmp) to between 512 MB and 10240 MB.

SnapStart [Info](#)

Reduce startup time by having Lambda cache a snapshot of your function after the function has initialized. To evaluate whether your function code is resilient to snapshot operations, review the [SnapStart compatibility considerations](#). For Python and .NET runtimes, [view pricing](#).

None

Supported runtimes: .NET 10 (C#/F#/PowerShell), .NET 8 (C#/F#/PowerShell), Java 11, Java 17, Java 21, Java 25, Python 5.12, Python 3.13, Python 3.14.

Timeout

1

 min

0

 sec

Execution role

To access other AWS services and resources your function needs an IAM role. Choose a role with the right permissions for your function.

LambdaEC2ProvisionRole

 [Create new role](#)

[View role details in IAM](#)

[Cancel](#)[Save](#)

Successfully updated the function "StopStartEC2Lambda".

Notifications 0 0 2 0 0 0

StopStartEC2Lambda

Throttle

Copy ARN

Actions

Function overview

Diagram

Template

 **StopStartEC2Lambda**

 Layers (0)

+ Add trigger

+ Add destination

Export to Infrastructure Composer

Download

Function ARN

arn:aws:lambda:eu-west-3:009850210027:function:StopStartEC2Lambda

Description

-

Last modified

11 seconds ago

Code Test Monitor Configuration Aliases Versions

Code source

Open in Visual Studio Code

Upload from

EXPLORE

- STOPSTARTEC2LAMBDA
 - lambda_function.py

DEPLOY ☒ Current

Deploy (Ctrl+Shift+U)

Test (Ctrl+Shift+I)

TEST EVENTS [NONE SELECTED]

+ Create new test event

Lambda Deployed 0 0 0 Amazon Q

lambda_function.py

```
1 import boto3
2
3 def lambda_handler(event, context):
4     ec2 = boto3.client("ec2", region_name="eu-west-3")
5
6     # Get instances tagged with AutoSchedule=True
7     instances = ec2.describe_instances(Filters=[{"Name": "tag:AutoSchedule", "Values": ["True"]}])
8
9     for reservation in instances["Reservations"]:
10         for instance in reservation["Instances"]:
11             instance_id = instance["InstanceId"]
12             state = instance["State"]["Name"]
13
14             if state == "running":
15                 ec2.stop_instances(InstanceIds=[instance_id])
16                 print(f"Stopped instance: {instance_id}")
17
```

Ln 4, Col 53 Spaces: 4 UTF-8 LF Python Lambda Layout: US

Code properties

Package size
482 byte

SHA256 hash

zRrItJ26xBjys1vKzNo2LJeNJB0tT2+SxwIVVBkvV7c=

Last modified

11 seconds ago

Runtime settings

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:6198fb9c708e48010232459840699b2264ba8c-c36621c703d5cd31b2245e02d1

Update runtime version
Auto

Layers (0)

Last fetched 4/7/2026, 6:21:21 PM

Edit

Search by attributes or search by keyword

Merge order	Name	Layer version	Compatible runtimes	Compatible architectures	Version ARN
-------------	------	---------------	---------------------	--------------------------	-------------

There is no data to display.

 The test event "StartStopEC2" was successfully saved.



Notifications  0  0  3  0  0 

StopStartEC2Lambda

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

Function overview [Info](#)

[Diagram](#)[Template](#)

 **StopStartEC2Lambda**

 Layers (0)

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

Function ARN

 arn:aws:lambda:eu-west-3:009850210027:function:StopStartEC2Lambda

Description

-

Last modified

2 minutes ago

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

Code source [Info](#)

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

EXPLORE

STOPSTARTEC2LAMBDA

lambda_function.py

DEPLOY

Deploy (Ctrl+Shift+U)

Test (Ctrl+Shift+I)

TEST EVENTS [SELECTED: STARTSTOPE...]

Create new test event

Private saved events

Lambda Deployed

0 0 0

Amazon Q

lambda_function.py

```
1 import boto3
2
3 def lambda_handler(event, context):
4     ec2 = boto3.client("ec2", region_name="eu-
5
6     # Get instances tagged with AutoSchedule=T
7     instances = ec2.describe_instances(Filters:
8
9     for reservation in instances["Reservations
10     for instance in reservation["Instances
11         instance_id = instance["InstanceId
12         state = instance["State"]["Name"]
13
14     if state == "running":
15         ec2.stop_instances(InstanceIds:
16         print(f"Stopped instance: {ins
17
```

Create new test event

Invoke

Save

Event Name

StartStopEC2

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 directly to the calling application.

test event is saved successfully.

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

Successfully updated the function StopStartEC2Lambda.

Code properties [Info](#)

Package size
482 byte

SHA256 hash

 zRfTJ26xBjys1vKzNo2LJeNJB0tt2+SxwIVVBkvV7c=

Last modified

2 minutes ago

Runtime settings [Info](#)

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:6198fb9c708e48010232459840699b2264ba8c-c36621c703d5cd31b2245e02d1

Update runtime version [Info](#)

Auto

Layers (0) [Info](#)

Last fetched 4/7/2026, 6:21:21 PM 

[Edit](#)

< 1 >



Merge order	Name	Layer version	Compatible runtimes	Compatible architectures	Version ARN
-------------	------	---------------	---------------------	--------------------------	-------------

There is no data to display.

Lambda > Functions > StopStartEC2Lambda

Deploy (Ctrl+Shift+U)

Test (Ctrl+Shift+I)

TEST EVENTS [SELECTED: STARTSTOPE...]

Create new test event

Private saved events

PROBLEMS

OUTPUT

CODE REFERENCE LOG

TERMINAL

Status: Succeeded

Test Event Name: StartStopEC2

Response:
null

Invocation type

Synchronous

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

Execution Results

Test event is saved successfully.

Successfully updated the function StopStartEC2Lambda.

The test event "StartStopEC2" was successfully saved.

Notifications 0 0 3 0 0

7

8

9

Get instances tagged with AutoSchedule

instances = ec2.describe_instances(Filter

for reservation in instances["Reservations"]

Lambda Deployed 0 0 0

Amazon Q

Lambda

Layout: US

EC2 > Instances

EC2

Dashboard

AWS Global View

Events

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

lambda-launched...

i-0aa9f4ce4125b8812

Stopped

t3.micro

-

View alarms

Amazon EventBridge

Scheduled rules

Create rule

Global endpoints

Archives

Replays

Pipes

Scheduler

Schedules

Schedule groups

Scheduled rules (legacy) Updated

Integration

Partner event sources

API destinations

Connections

Step 1

Define rule detail

Step 2

Define schedule

Step 3

Select target(s)

Step 4 - optional

Configure tags

Step 5

Review and create

Define schedule

Info

Schedule pattern

Schedule pattern

Choose the schedule type that best meets your needs.

☒ A fine-grained schedule that runs at a specific time, such as 8:00 a.m. PST on the first Monday of every month.

☐ A schedule that runs at a regular rate, such as every 10 minutes.

Cron expression

Info

Define the cron expression for the schedule

☒ cron

n (

0

8

*

*

?

*

)

Minutes

Hours

Day of month

Month

Day of week

Year

Next 10 trigger date(s)

UTC

Wed, 08 Apr 2026 08:00:00 UTC

Thu, 09 Apr 2026 08:00:00 UTC

Fri, 10 Apr 2026 08:00:00 UTC

Sat, 11 Apr 2026 08:00:00 UTC

Sun, 12 Apr 2026 08:00:00 UTC

Mon, 13 Apr 2026 08:00:00 UTC

Tue, 14 Apr 2026 08:00:00 UTC

Wed, 15 Apr 2026 08:00:00 UTC

Thu, 16 Apr 2026 08:00:00 UTC

Fri, 17 Apr 2026 08:00:00 UTC

Cancel

Previous

Next

Amazon EventBridge

Scheduled rules

Create rule

Global endpoints

Archives

Replays

Pipes

Scheduler

Schedules

Schedule groups

Scheduled rules (legacy) Updated

Integration

Partner event sources

API destinations

Connections

Step 1

Define rule detail

Step 2

Define schedule

Step 3

Select target(s)

Step 4 - optional

Configure tags

Step 5

Review and create

Select target(s)

Permissions

Note: When using the EventBridge console, EventBridge will automatically configure the proper permissions for the selected targets. If you're using the AWS CLI, SDK, or CloudFormation, you'll need to configure the proper permissions.

Target 1

Target types

Select an EventBridge event bus, EventBridge API destination (SaaS partner), or another AWS service as a target.

EventBridge event bus

EventBridge API destination

AWS service

Select a target Info

Select target(s) to invoke when an event matches your event pattern or when schedule is triggered (limit of 5 targets per rule)

Lambda function

Target location

Target in this account

Target in another AWS account

Function

StopStartEC2Lambda

Configure version/alias

Permissions

Use execution role (recommended)

Execution role

EventBridge needs permission to send events to the target specified above. By continuing, you are allowing us to do so: EventBridge and AWS Identity and Access Management

Create a new role for this specific resource

Use existing role

Role name

Amazon_EventBridge_Invoke_Lambda_3938227

Additional settings

Cancel

Skip to Review and create

Previous

Next

Add another target

CloudShell

Feedback

Console Mobile App

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

Privacy

Terms

Cookie preferences

Amazon EventBridge > Scheduled rules > Create rule

Global endpoints

Archives

Replays

Pipes

Scheduler

Schedules

Schedule groups

Scheduled rules (legacy) Updated

Integration

Partner event sources

API destinations

Connections

Step 1

Define rule detail

Step 2

Define schedule

Step 3

Select target(s)

Step 4 - optional

Configure tags

Step 5

Review and create

Review and create

Step 1: Define rule detail

Define rule detail

Rule name

EC2ScheduleRule

Description

Event bus

default

Status

Enabled

Rule type

Scheduled rule

Step 2: Build schedule

Event schedule Info

Cron expression

0 8 * * ? *

Next 10 trigger date(s)

UTC

Wed, 08 Apr 2026 08:00:00 UTC
Thu, 09 Apr 2026 08:00:00 UTC
Fri, 10 Apr 2026 08:00:00 UTC
Sat, 11 Apr 2026 08:00:00 UTC
Sun, 12 Apr 2026 08:00:00 UTC
Mon, 13 Apr 2026 08:00:00 UTC
Tue, 14 Apr 2026 08:00:00 UTC
Wed, 15 Apr 2026 08:00:00 UTC
Thu, 16 Apr 2026 08:00:00 UTC
Fri, 17 Apr 2026 08:00:00 UTC

Step 3: Select target(s)

Targets

Details	Target Name	Type	ARN	Input	Role
▼	StopStartEC2Lambda	Lambda function	arn:aws:lambda:eu-west-3:009850210027:function:StopStartEC2Lambda	Matched event	Amazon_EventBridge_Invoke_Lambda_3938227
Input to target:		Matched event			
Additional parameters:		--			
Dead-letter queue (DLQ):		-			

Step 4: Configure tag(s)

Tags (0)

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key

Value

No tags associated with this resource.

Cancel

Previous

Create rule

Amazon EventBridge

Scheduled rules

Create rule

Global endpoints

Archives

Replays

Pipes

Scheduler

Schedules

Schedule groups

Scheduled rules (legacy) Updated

Integration

Partner event sources

API destinations

Connections

Step 1

Define rule detail

Step 2

Define schedule

Step 3

Select target(s)

Step 4 - optional

Configure tags

Step 5

Review and create

Define rule detail

Scheduled rule detail

Name

EC2ScheduleRule2

Description - optional

Enter description

Event bus

default

Enable the rule on the selected event bus

EventBridge Scheduler - A new AWS scheduling capability!

A new EventBridge scheduling functionality that provides one-time and recurring scheduling functionality independent of Event buses and rules. You can create a schedule to invoke targets such as a Lambda function. [Learn More](#)

Continue in EventBridge Scheduler

Cancel

Next

Amazon EventBridge

Scheduled rules

Create rule

Global endpoints

Archives

Replays

Pipes

Scheduler

Schedules

Schedule groups

Scheduled rules (legacy) Updated

Integration

Partner event sources

API destinations

Connections

Step 1

Define rule detail

Step 2

Define schedule

Step 3

Select target(s)

Step 4 - optional

Configure tags

Step 5

Review and create

Define schedule

Info

Schedule pattern

Schedule pattern

Choose the schedule type that best meets your needs.

☒ A fine-grained schedule that runs at a specific time, such as 8:00 a.m. PST on the first Monday of every month.

☐ A schedule that runs at a regular rate, such as every 10 minutes.

Cron expression Info

Define the cron expression for the schedule

☒ cron

n (

:

)

Minutes

Hours

Day of month

Month

Day of week

Year

Next 10 trigger date(s)

UTC

Tue, 07 Apr 2026 18:00:00 UTC

Wed, 08 Apr 2026 18:00:00 UTC

Thu, 09 Apr 2026 18:00:00 UTC

Fri, 10 Apr 2026 18:00:00 UTC

Sat, 11 Apr 2026 18:00:00 UTC

Sun, 12 Apr 2026 18:00:00 UTC

Mon, 13 Apr 2026 18:00:00 UTC

Tue, 14 Apr 2026 18:00:00 UTC

Wed, 15 Apr 2026 18:00:00 UTC

Thu, 16 Apr 2026 18:00:00 UTC

Cancel

Previous

Next

Amazon EventBridge

Scheduled rules

Create rule

Global endpoints

Archives

Replays

Pipes

Scheduler

Schedules

Schedule groups

Scheduled rules (legacy) Updated

Integration

Partner event sources

API destinations

Connections

Step 1

Define rule detail

Step 2

Define schedule

Step 3

Select target(s)

Step 4 - optional

Configure tags

Step 5

Review and create

Select target(s)

Permissions

Note: When using the EventBridge console, EventBridge will automatically configure the proper permissions for the selected targets. If you're using the AWS CLI, SDK, or CloudFormation, you'll need to configure the proper permissions.

Target 1

Target types

Select an EventBridge event bus, EventBridge API destination (SaaS partner), or another AWS service as a target.

EventBridge event bus

EventBridge API destination

AWS service

Select a target Info

Select target(s) to invoke when an event matches your event pattern or when schedule is triggered (limit of 5 targets per rule)

Lambda function

Target location

Target in this account

Target in another AWS account

Function

StopStartEC2Lambda

Configure version/alias

Permissions

Use execution role (recommended)

Execution role

EventBridge needs permission to send events to the target specified above. By continuing, you are allowing us to do so: [EventBridge and AWS Identity and Access Management](#)

Create a new role for this specific resource

Use existing role

Role name

Amazon_EventBridge_Invoke_Lambda_1536613217

Additional settings

Cancel

Skip to Review and create

Previous

Next

Add another target

CloudShell

Feedback

Console Mobile App

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

Privacy

Terms

Cookie preferences

Amazon EventBridge

Scheduled rules

Create rule

Global endpoints

Archives

Replays

Pipes

Scheduler

Schedules

Schedule groups

Scheduled rules (legacy) Updated

Integration

Partner event sources

API destinations

Connections

Step 1

Define rule detail

Step 2

Define schedule

Step 3

Select target(s)

Step 4 - optional

Configure tags

Step 5

Review and create

Review and create

Step 1: Define rule detail

Define rule detail

Rule name

EC2ScheduleRule2

Description

Event bus

default

Status

Enabled

Rule type

Scheduled rule

Step 2: Build schedule

Event schedule

Cron expression

0 18 * * ? *

Next 10 trigger date(s)

UTC

Tue, 07 Apr 2026 18:00:00 UTC
Wed, 08 Apr 2026 18:00:00 UTC
Thu, 09 Apr 2026 18:00:00 UTC
Fri, 10 Apr 2026 18:00:00 UTC
Sat, 11 Apr 2026 18:00:00 UTC
Sun, 12 Apr 2026 18:00:00 UTC
Mon, 13 Apr 2026 18:00:00 UTC
Tue, 14 Apr 2026 18:00:00 UTC
Wed, 15 Apr 2026 18:00:00 UTC
Thu, 16 Apr 2026 18:00:00 UTC

Step 3: Select target(s)

Targets

Details	Target Name	Type	ARN	Input	Role
	StopStartEC2Lambda	Lambda function	arn:aws:lambda:eu-west-3:009850210027:function:StopStartEC2Lambda	Matched event	Amazon_EventBridge_Invoke_Lambda_1536613217
Input to target:		Matched event			
Additional parameters:		--			
Dead-letter queue (DLQ):		-			

Step 4: Configure tag(s)

Tags (0)

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key

Value

No tags associated with this resource.

Cancel

Previous

Create rule

CloudWatch

Favorites and recents

Ingestion

Dashboards

Alarms 0 0 0 0

GenAI Observability

Application Signals (APM) New

Infrastructure Monitoring

Logs

Log Management

CloudWatch > Log management > /aws/lambda/StopStartEC2Lambda

Stored bytes
-

Sensitive data count
-

Metric filters
0

Log streams

Tags

Data protection

Anomaly detection

Metric filters

Subscription filters

Log streams (1)

By default, we only load the most recent log streams.

Filter log streams or try prefix search

☐ Exact match ☐ Show expired [Info](#) < 1 >

☐	Log stream	Last event time
☐	2026/04/07/[\$LATEST]e0f25b327eee4f3b9a8e4a2f2394e44d	2026-04-07 15:23:57 (UTC)