KPR Institute of Engineering and Technology
Learn Beyond (Autonomous, NAAC "A")
Avinashi Road, Arasur, Coimbatore.

aws

Department of Computer Science and Engineering
Accredited by NBA | Permanently Affiliated to Anna University - Chennai

AWS CLOUD AND DEVOPS

by Mr. Mahendran Selvakumar

Automating EC2 Instance Management Using AWS Lambda with GitHub Copilot

Name: Sooriya N

Class: III CSE C

Organized by KPR Institute of Engineering and Technology

Department of Computer Science and Engineering

Overview

This documentation outlines the process of automating the start and stop operations for an EC2 instance using AWS Lambda. GitHub Copilot to assist in generating Python scripts for these Lambda functions, making development efficient and intuitive. Screenshots of these tasks are included for clarity.

Prerequisites

Before starting, ensure the following prerequisites are met:

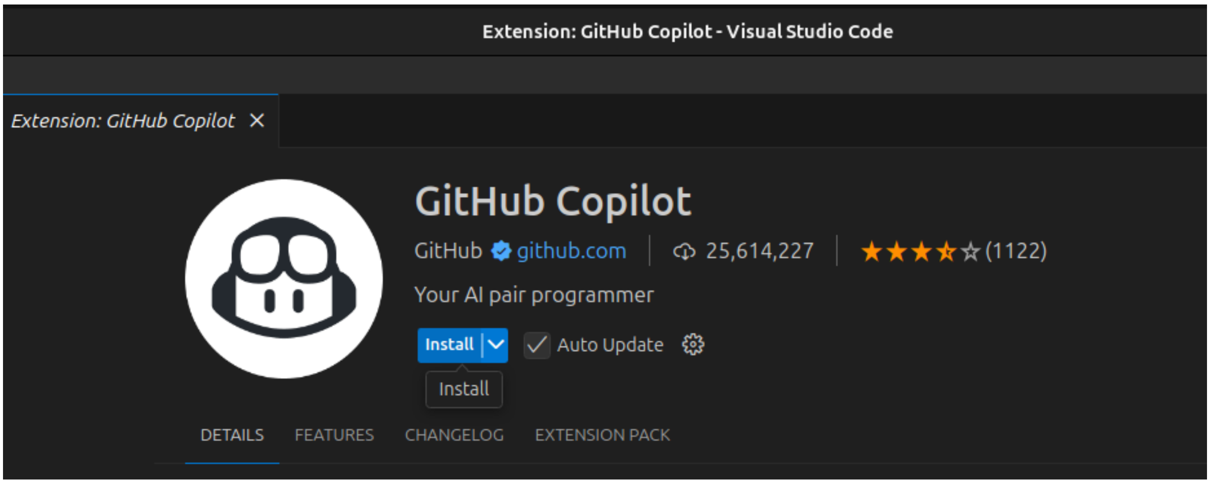
- An AWS account with access to EC2 and Lambda services.
- A configured IAM role with the necessary permissions (AmazonEC2FullAccess for simplicity).
- GitHub Copilot enabled in your development environment.

Accessing GITHUB Copilot

GitHub in VS Code:

Open VS Code and go to the Extensions view by clicking the Extensions icon in the Activity Bar.

Search for "GitHub Pull Requests and Issues" and install it.



Extension: GitHub Copilot - Visual Studio Code

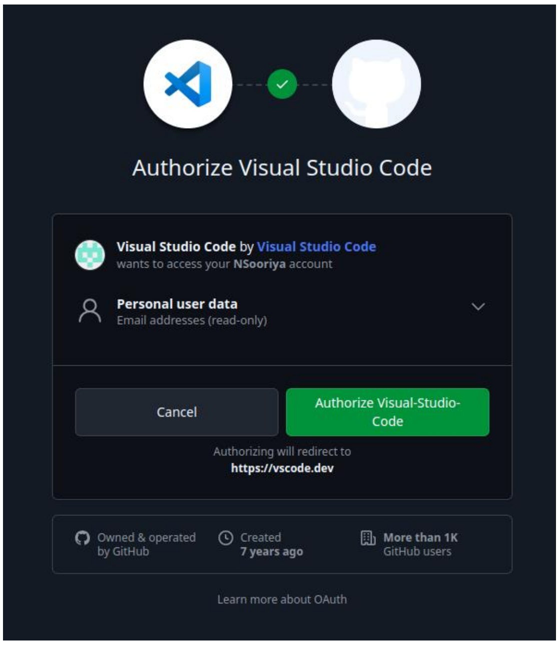
Extension: GitHub Copilot ×

GitHub Copilot

GitHub github.com | 25,614,227 | ★★★★☆ (1122)

Your AI pair programmer

Before using the Copilot, GitHub must be logged in and Visual Studio Code must be authorized. This is necessary to establish a secure and authenticated connection between your GitHub account and the Copilot service.



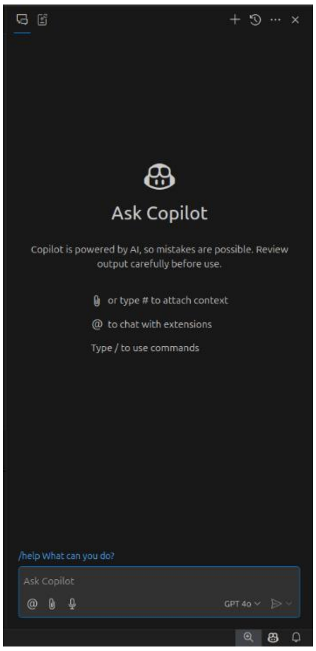
Authorize Visual Studio Code

Visual Studio Code by **Visual Studio Code** wants to access your **NSooriya** account

Personal user data Email addresses (read-only)

Authorizing will redirect to **https://vscode.dev**

Cancel
Authorize Visual-Studio-Code
Owned & operated by GitHub
Created 7 years ago
More than 1K GitHub users
Learn more about OAuth



Ask Copilot

Copilot is powered by AI, so mistakes are possible. Review output carefully before use.

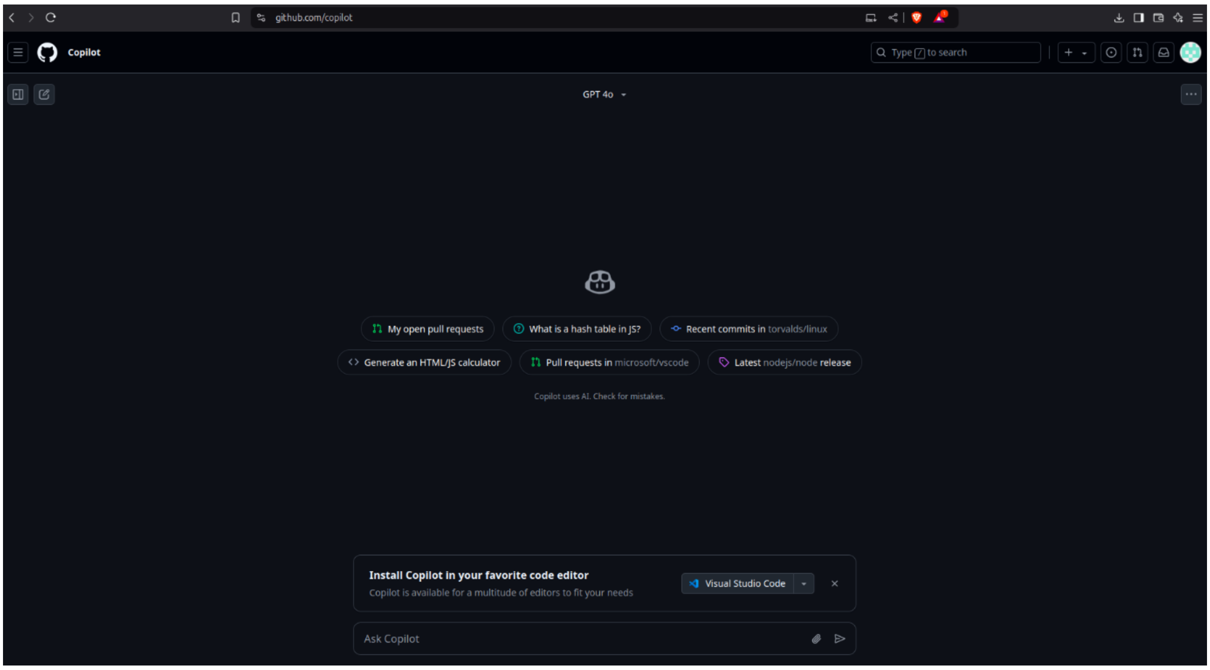
or type # to attach context

@ to chat with extensions

Type / to use commands

GitHub via the Official Website:

To access GitHub Copilot through the official website, sign in to your GitHub account, navigate to the GitHub Copilot page.



github.com/copilot

Copilot uses AI. Check for mistakes.

Install Copilot in your favorite code editor
Copilot is available for a multitude of editors to fit your needs

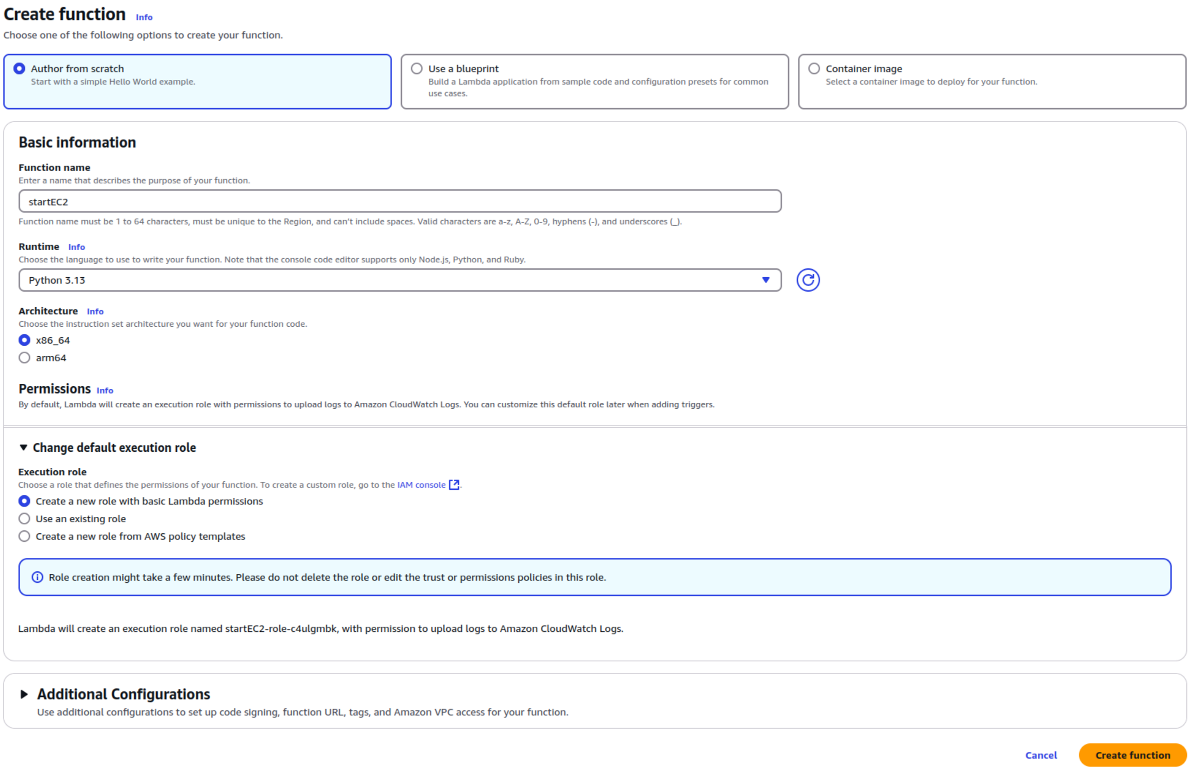
Copilot
Type / to search
GPT 4o
My open pull requests
What is a hash table in JS?
Recent commits in torvalds/linux
Generate an HTML/JS calculator
Pull requests in microsoft/vscode
Latest nodejs/node release
Visual Studio Code
Ask Copilot

Free for Students: GitHub Copilot is available for free to verified students through the GitHub Student Developer Pack, providing access to the full range of features.

Free for Open-Source Contributors: GitHub Copilot is also free for maintainers of public, open-source repositories. If you are actively contributing to open-source projects, you can apply for free access.

Creating the Lambda Function

- Go to the AWS Management Console.
- In the search bar, type **Lambda** and open the **AWS Lambda** service.
- Click **Create function**.
- Select **Author from scratch**.
    - **Function name**: Provide a name for your function.
    - **Runtime**: Choose Python 3.x or any supported runtime.
    - **Role**: Create a new role with basic Lamba permissions.
- Click **Create function**.



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.

startEC2

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 3.13

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

● x86_64
○ arm64

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
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.

● Create a new role with basic Lambda permissions
○ Use an existing role
○ Create a new role from AWS policy templates

ⓘ Role creation might take a few minutes. Please do not delete the role or edit the trust or permissions policies in this role.

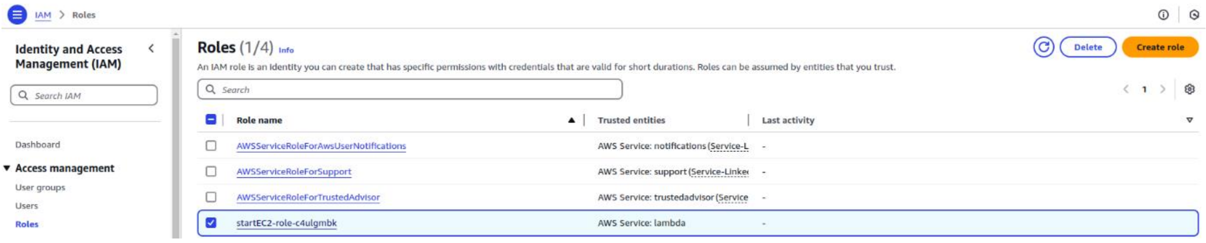
Lambda will create an execution role named startEC2-role-c4ulgmbk, with permission to upload logs to Amazon CloudWatch Logs.

▶ Additional Configurations
Use additional configurations to set up code signing, function URL, tags, and Amazon VPC access for your function.

ADDING PERMISSION FOR THE ROLE:

Roles can be created with proper permissions and used for a Lamba function, if "Create a new role with basic Lambda permissions" was selected then adding permission to the newly created role for accessing EC2 Instances must be done before testing the Lambda function.

Select the Role which has been created for the Lambda function.



IAM > Roles

Identity and Access Management (IAM)

Search IAM

Dashboard

Access management

User groups

Users

Roles

Roles (1/4) Info

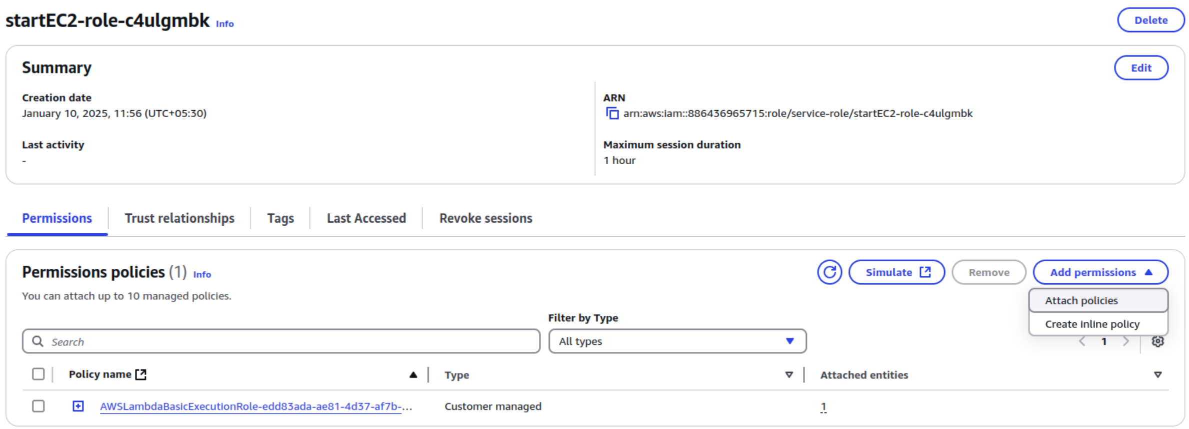
Delete | Create role

An IAM role is an identity you can create that has specific permissions with credentials that are valid for short durations. Roles can be assumed by entities that you trust.

Search

	Role name	Trusted entities	Last activity
□	AWSServiceRoleForAwsUserNotifications	AWS Service: notifications (Service-L	-
□	AWSServiceRoleForSupport	AWS Service: support (Service-Linke	-
□	AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service	-
☑	startEC2-role-c4ulgmbk	AWS Service: lambda	-

Click on “Add permissions” and search for “AmazonEC2FullAccess” and add it.



startEC2-role-c4ulgmbk Info

Delete

Summary

Edit

Creation date
January 10, 2025, 11:56 (UTC+05:30)

Last activity
-

ARN
arn:aws:iam::886436965715:role/service-role/startEC2-role-c4ulgmbk

Maximum session duration
1 hour

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

Permissions policies (1) Info

Simulate | Remove | Add permissions

Attach policies

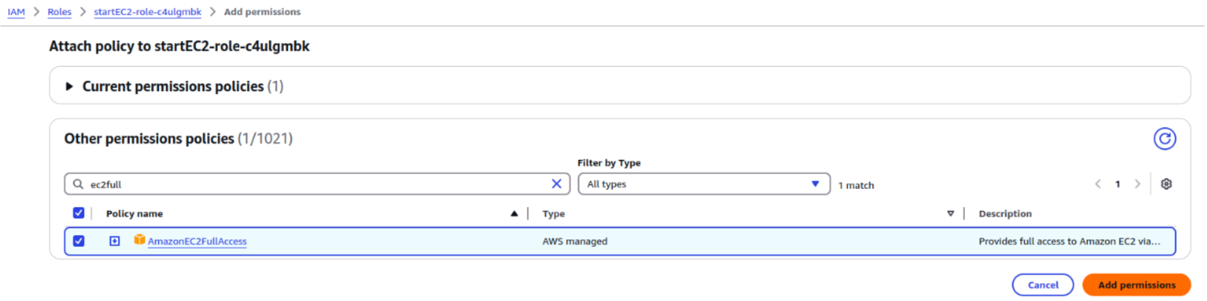
Create inline policy

You can attach up to 10 managed policies.

Filter by Type: All types

Search

		Policy name	Type	Attached entities
□	⊞	AWSLambdaBasicExecutionRole-edd83ada-ae81-4d37-af7b-...	Customer managed	1



IAM > Roles > startEC2-role-c4ulgmbk > Add permissions

Attach policy to startEC2-role-c4ulgmbk

▸ **Current permissions policies** (1)

Other permissions policies (1/1021)

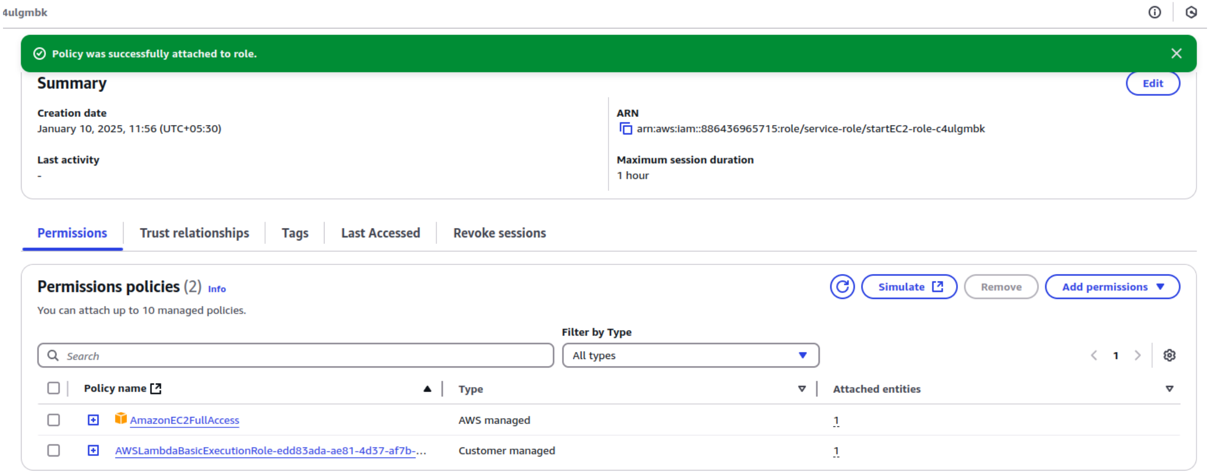
Filter by Type: All types

ec2full

1 match

		Policy name	Type	Description
☑	⊞	AmazonEC2FullAccess	AWS managed	Provides full access to Amazon EC2 via...

Cancel | Add permissions



4ulgmbk

Policy was successfully attached to role.

Summary

Edit

Creation date
January 10, 2025, 11:56 (UTC+05:30)

Last activity
-

ARN
arn:aws:iam::886436965715:role/service-role/startEC2-role-c4ulgmbk

Maximum session duration
1 hour

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

Permissions policies (2) Info

Simulate | Remove | Add permissions

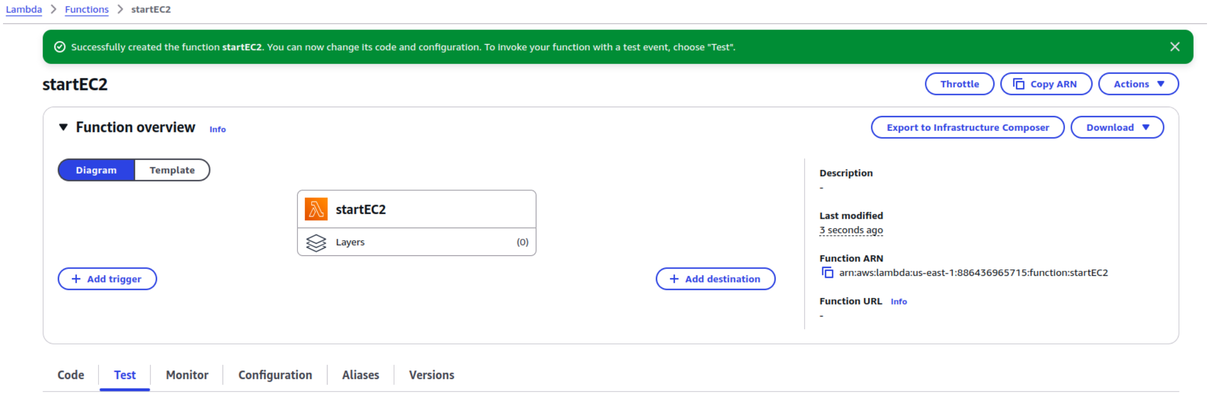
You can attach up to 10 managed policies.

Filter by Type: All types

Search

		Policy name	Type	Attached entities
□	⊞	AmazonEC2FullAccess	AWS managed	1
□	⊞	AWSLambdaBasicExecutionRole-edd83ada-ae81-4d37-af7b-...	Customer managed	1

WRITE LAMBDA FUNCTION CODE:



Lambda > Functions > startEC2

Successfully created the function **startEC2**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

startEC2

Description
-

Last modified
3 seconds ago

Function ARN
arn:aws:lambda:us-east-1:886436965715:function:startEC2

Function URL
-

Once your function is created, you will see the function editor.

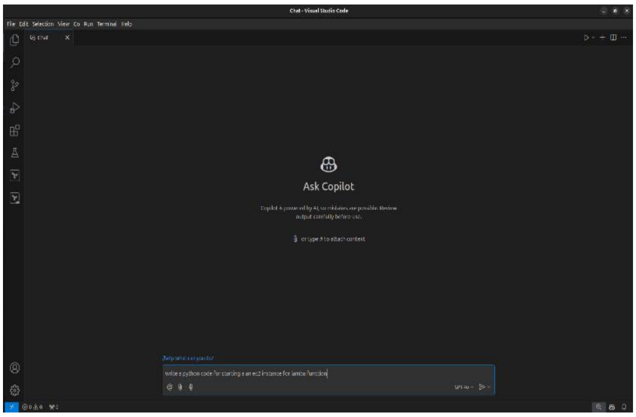
Write or paste the code in the Code Source section for Starting an EC2 instance obtained from interacting with the GitHub Copilot.

CODE FROM GITHUB COPILOT:

PROMPTING GITHUB COPILOT

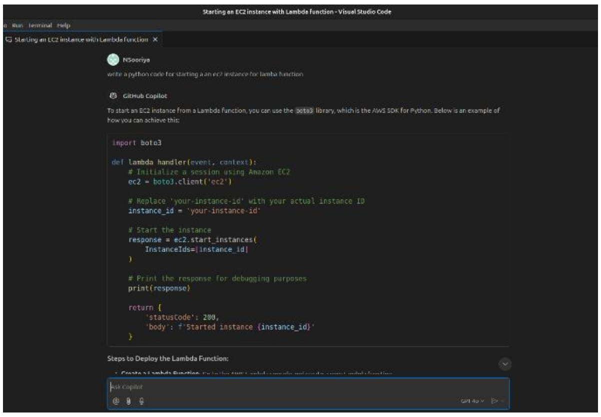
- Opened Visual Studio Code with the GitHub Copilot extension installed and activated.
- GITHUB Copilot can be found in the bottom right in Visual Studio Code.
- Started the interaction with a comment describing the objective:

# Write a python code for starting an EC2 instance for lambda function



write a python code for starting a an ec2 instance for lambe function

Chat - Visual Studio Code
File Edit Selection View Go Run Terminal Help
Chat
Ask Copilot
Copilot is powered by AI, so mistakes are possible. Review output carefully before use.
or type # to attach context
/help What can you do?



NSooriya

write a python code for starting a an ec2 instance for lamba function

GitHub Copilot

To start an EC2 instance from a Lambda function, you can use the `boto3` library, which is the AWS SDK for Python. Below is an example of how you can achieve this:

```
import boto3

def lambda_handler(event, context):
    # Initialize a session using Amazon EC2
    ec2 = boto3.client('ec2')

    # Replace 'your-instance-id' with your actual instance ID
    instance_id = 'your-instance-id'

    # Start the instance
    response = ec2.start_instances(
        InstanceIds=[instance_id]
    )

    # Print the response for debugging purposes
    print(response)

    return {
        'statusCode': 200,
        'body': f'Started instance {instance_id}'
    }
```

Steps to Deploy the Lambda Function:

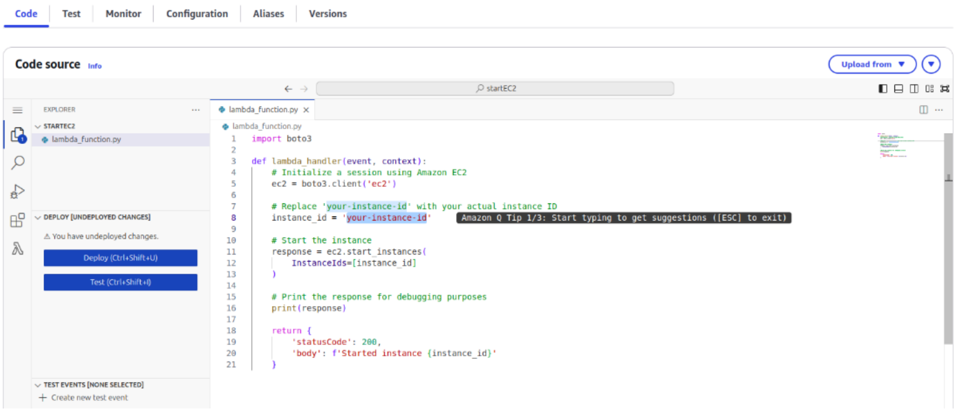
Starting an EC2 instance with Lambda function - Visual Studio Code
Run Terminal Help
Starting an EC2 instance with Lambda function
Ask Copilot

CODE GENERATED BY COPILOT

1. Importing the boto3 library to interact with AWS services.
2. Defining the lambda_handler function, which serves as the entry point for the Lambda function.
3. Initializing an EC2 client with boto3.client('ec2').

4. Starting the EC2 instance by calling the start_instances method and providing the instance ID.
5. Printing the response for debugging purposes.
6. Returning an HTTP response with a 200 status code and a success message.

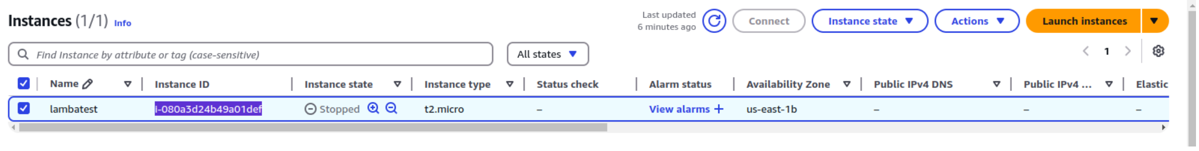
Paste the code in Code Source section in the Lambda function.



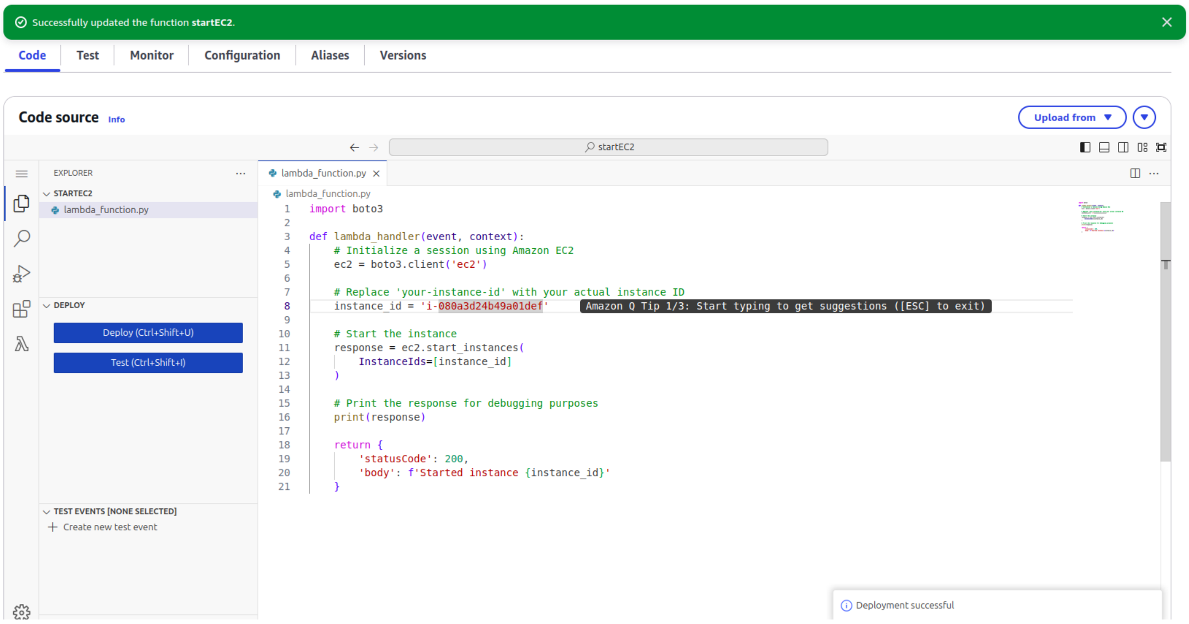
```
import boto3

def lambda_handler(event, context):
    # Initialize a session using Amazon EC2
    ec2 = boto3.client('ec2')

    # Replace 'your-instance-id' with your actual instance ID
    instance_id = 'your-instance-id'

    # Start the instance
    response = ec2.start_instances(
        InstanceIds=[instance_id]
    )

    # Print the response for debugging purposes
    print(response)

    return {
        'statusCode': 200,
        'body': f'Started instance {instance_id}'
    }
```

Amazon Q Tip 1/3: Start typing to get suggestions ([ESC] to exit)

Change the “your_instance_id” to the target Instance ID from the Instances page in the AWS Console.



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic
☑	lambatest	i-080a3d24b49a01def	Stopped	t2.micro	–	View alarms +	us-east-1b	–	–	–



Successfully updated the function startEC2.

```
import boto3

def lambda_handler(event, context):
    # Initialize a session using Amazon EC2
    ec2 = boto3.client('ec2')

    # Replace 'your-instance-id' with your actual instance ID
    instance_id = 'i-080a3d24b49a01def'

    # Start the instance
    response = ec2.start_instances(
        InstanceIds=[instance_id]
    )

    # Print the response for debugging purposes
    print(response)

    return {
        'statusCode': 200,
        'body': f'Started instance {instance_id}'
    }
```

Amazon Q Tip 1/3: Start typing to get suggestions ([ESC] to exit)

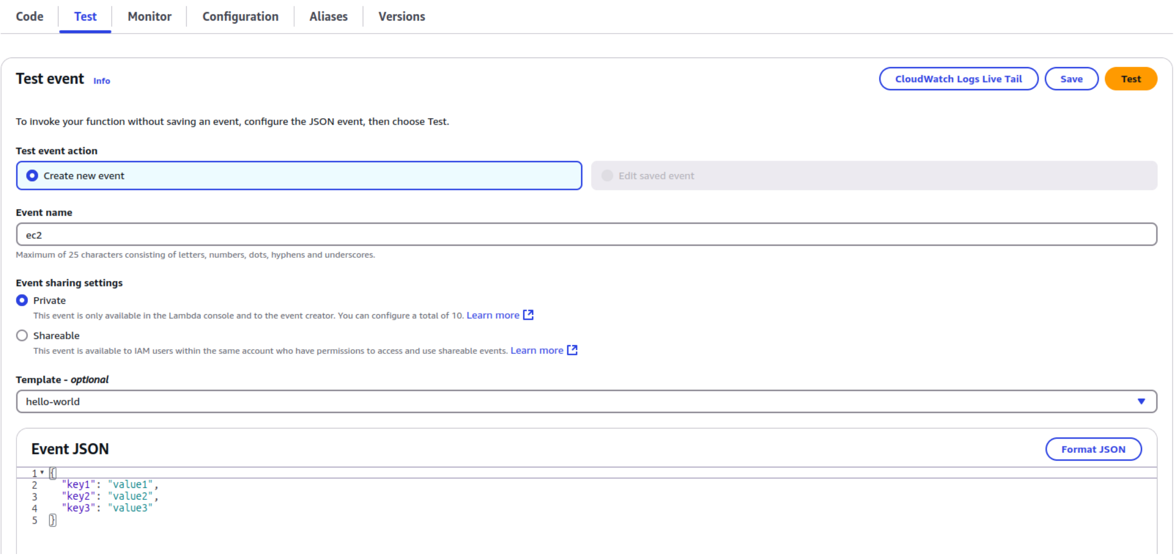
Deployment successful

Click the Deploy button to update the Lambda function. When you make updates to your Lambda function (such as adding or modifying the code, configuration, or environment variables), you need to deploy those changes for them to take effect.

Testing the Lambda Function

Click **Test** in the Lambda console to trigger the function manually.

Provide test event data if necessary, or leave it blank if not required for API call.



Test event

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action

● Create new event ○ Edit saved event

Event name

ec2

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

Event sharing settings

● Private

This event is only available in the Lambda console and to the event creator. You can configure a total of 10. Learn more

○ Shareable

This event is available to IAM users within the same account who have permissions to access and use shareable events. Learn more

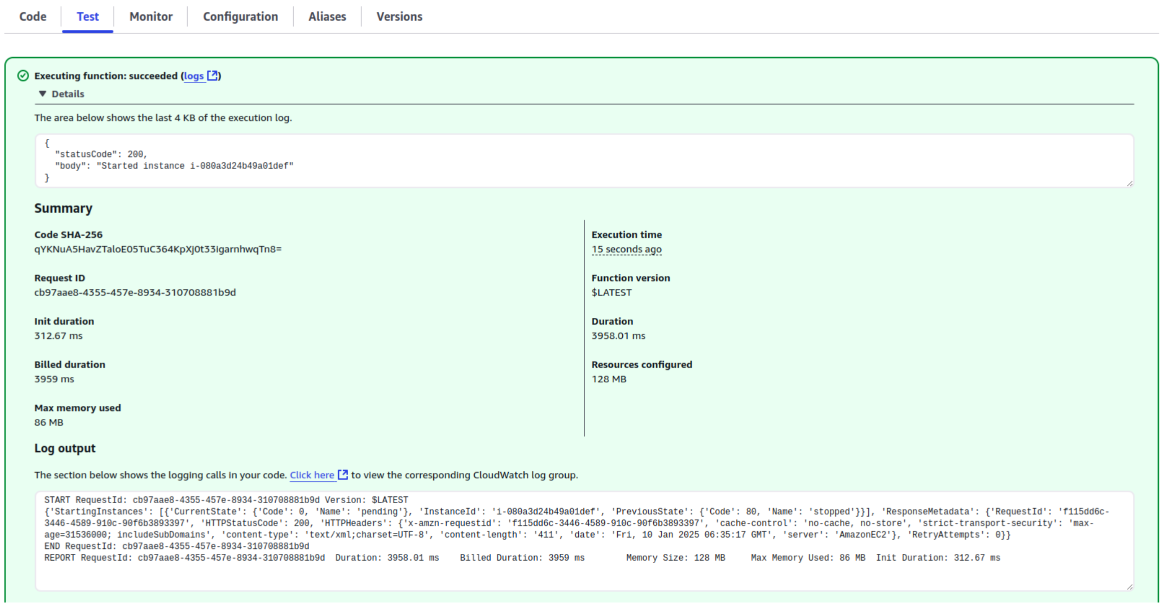
Template - *optional*

hello-world

Event JSON

```
{
  "key1": "value1",
  "key2": "value2",
  "key3": "value3"
}
```

After testing, you'll see the output in the **Execution result** section.



Executing function: succeeded (logs)

▼ Details

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

```
{
  "statusCode": 200,
  "body": "Started instance i-080a3d24b49a01def"
}
```

Summary

Code SHA-256
qYKNuA5HavZTaloE05TuC364KpXj0t33igarnhwqTn8=

Request ID
cb97aae8-4355-457e-8934-310708881b9d

Init duration
312.67 ms

Billed duration
3959 ms

Max memory used
86 MB

Execution time
15 seconds ago

Function version
$LATEST

Duration
3958.01 ms

Resources configured
128 MB

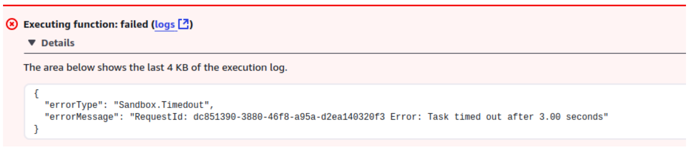
Log output

The section below shows the logging calls in your code. Click here to view the corresponding CloudWatch log group.

```
START RequestId: cb97aae8-4355-457e-8934-310708881b9d Version: $LATEST
{'StartingInstances': [{'CurrentState': {'Code': 0, 'Name': 'pending'}, 'InstanceId': 'i-080a3d24b49a01def', 'PreviousState': {'Code': 80, 'Name': 'stopped'}}], 'ResponseMetadata': {'RequestId': 'f115dd6c-3446-4589-910c-90f6b3893397', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-requestid': 'f115dd6c-3446-4589-910c-90f6b3893397', 'cache-control': 'no-cache, no-store', 'strict-transport-security': 'max-age=31536000; includeSubDomains', 'content-type': 'text/xml;charset=UTF-8', 'content-length': '411', 'date': 'Fri, 10 Jan 2025 06:35:17 GMT', 'server': 'AmazonEC2'}, 'RetryAttempts': 0}}
END RequestId: cb97aae8-4355-457e-8934-310708881b9d
REPORT RequestId: cb97aae8-4355-457e-8934-310708881b9d  Duration: 3958.01 ms    Billed Duration: 3959 ms        Memory Size: 128 MB     Max Memory Used: 86 MB  Init Duration: 312.67 ms
```

NOTE:

If the Function’s execution failed because of Timeout, go to the Configuration section and increase the Timeout value in general configuration.

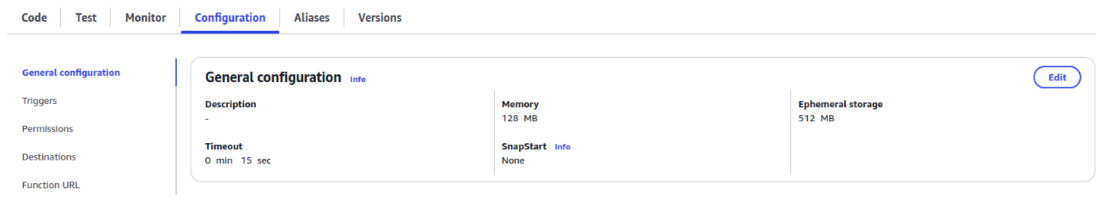


Executing function: failed (logs)

▼ Details

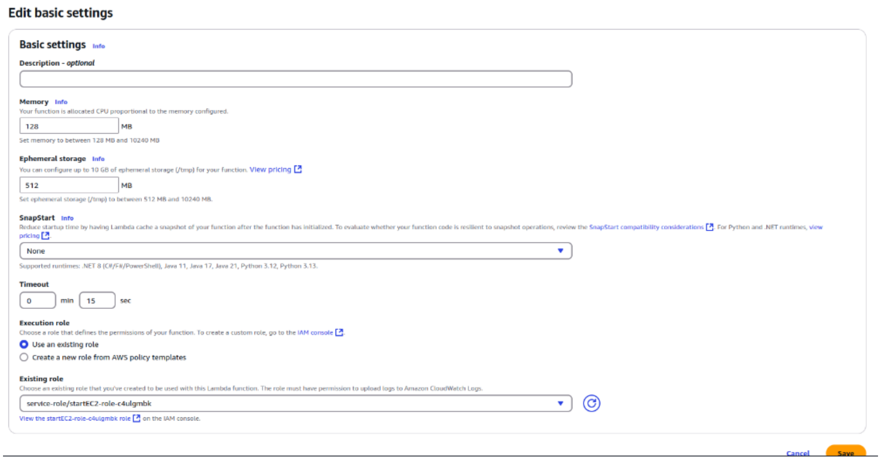
The area below shows the last 4 KB of the execution log.

```
{
  "errorType": "Sandbox.Timedout",
  "errorMessage": "RequestId: dc851390-3880-46f8-a95a-d2ea140320f3 Error: Task timed out after 3.00 seconds"
}
```



General configuration Info

Description -	Memory 128 MB	Ephemeral storage 512 MB
Timeout 0 min 15 sec	SnapStart Info None	



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 8 (C#/F#/PowerShell), Java 11, Java 17, Java 21, Python 3.12, Python 3.13.

Timeout
0 min 15 sec

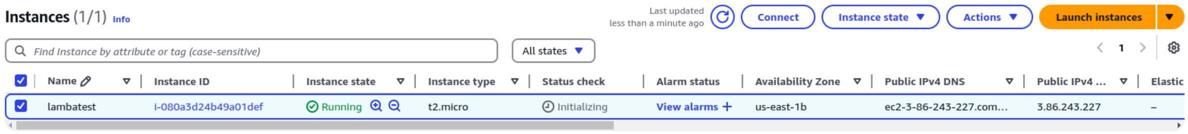
Execution role
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.
● Use an existing role
○ Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.
service-role/startEC2-role-c4ulgmbk
View the startEC2-role-c4ulgmbk role on the IAM console.

Click on “Save” and test the function again.

CHECK THE EXECUTION RESULT:

If the execution of the Lambda function is successful, go to the Instances dashboard in the console and check if the target instance have started.

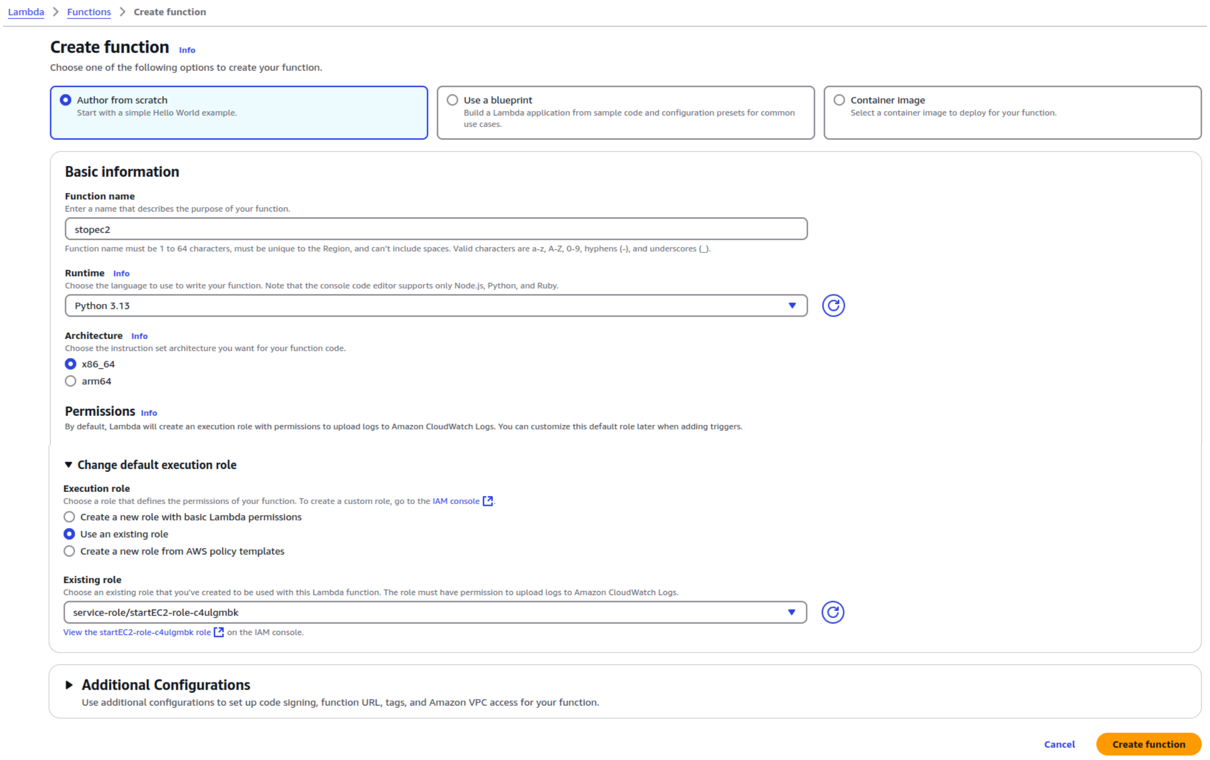


☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic
☑	lambatest	i-080a3d24b49a01def	Running	t2.micro	Initializing	View alarms +	us-east-1b	ec2-3-86-243-227.com...	3.86.243.227	–

The same process can be followed for a creating a Lambda function to Stop an EC2 instance.

Creating a Lambda Function to Stop an EC2 Instance

- Go to the **Lambda console**.
- Click **Create function**.
- Choose **Author from scratch**.
    1. Give your function a name (e.g., StopEC2).
    2. Select a runtime (e.g., Python 3.9 or later).
    3. Under Permissions, choose **Use an existing role** and select the IAM role created during the previous steps.
- Click **Create function**.



Lambda > Functions > Create function

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.

stopec2

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 3.13

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

● x86_64
○ arm64

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
Choose a role that defines the permissions of your function. To create a custom role, go to the IAM console.

○ Create a new role with basic Lambda permissions
● Use an existing role
○ Create a new role from AWS policy templates

Existing role
Choose an existing role that you've created to be used with this Lambda function. The role must have permission to upload logs to Amazon CloudWatch Logs.

service-role/startEC2-role-c4ulgmbk

View the startEC2-role-c4ulgmbk role on the IAM console.

▶ Additional Configurations
Use additional configurations to set up code signing, function URL, tags, and Amazon VPC access for your function.

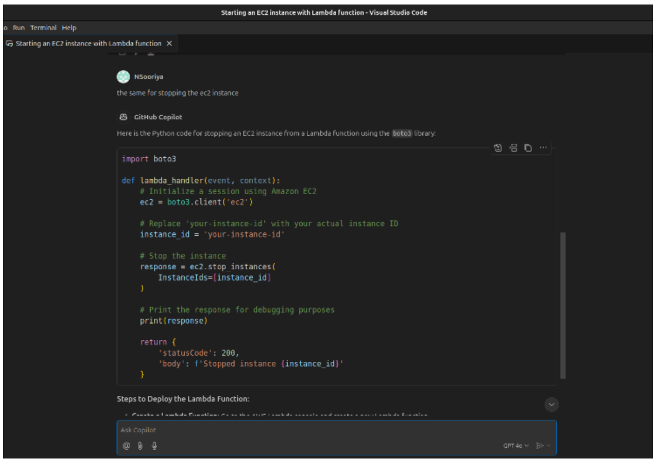
LAMBDA FUNCTION CODE:

Once the Lambda function is created, the function editor will appear. In the code editor section, the code designed to stop an EC2 instance, obtained through interaction with GitHub Copilot, should be inserted.

CODE GENERATED BY COPILOT (Stopping an EC2 Instance)

1. Importing the boto3 library to interact with AWS services. Defining the lambda_handler function, which serves as the entry point for the Lambda function.

2. Initializing an EC2 client with boto3.client('ec2').
3. Stopping the EC2 instance by calling the stop_instances method and providing the instance ID.
4. Printing the response for debugging purposes.
5. Returning an HTTP response with a 200 status code and a success message.


Starting an EC2 instance with Lambda function - Visual Studio Code
Run Terminal Help
Starting an EC2 instance with Lambda function

NSooriya

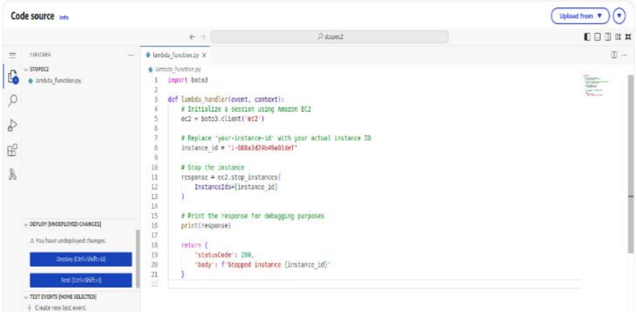
the same for stopping the ec2 instance

GitHub Copilot

Here is the Python code for stopping an EC2 instance from a Lambda function using the boto3 library:

```python
import boto3

def lambda_handler(event, context):
    # Initialize a session using Amazon EC2
    ec2 = boto3.client('ec2')

    # Replace 'your-instance-id' with your actual instance ID
    instance_id = 'your-instance-id'

    # Stop the instance
    response = ec2.stop_instances(
        InstanceIds=[instance_id]
    )

    # Print the response for debugging purposes
    print(response)

    return {
        'statusCode': 200,
        'body': f'Stopped instance {instance_id}'
    }
```

Steps to Deploy the Lambda Function:



```python
import boto3

def lambda_handler(event, context):
    # Initialize a session using Amazon EC2
    ec2 = boto3.client('ec2')

    # Replace 'your-instance-id' with your actual instance ID
    instance_id = 'i-088a3d24b49a01def'

    # Stop the instance
    response = ec2.stop_instances(
        InstanceIds=[instance_id]
    )

    # Print the response for debugging purposes
    print(response)

    return {
        'statusCode': 200,
        'body': f'Stopped instance {instance_id}'
    }
```

DEPLOYING AND TESTING THE FUNCTION:

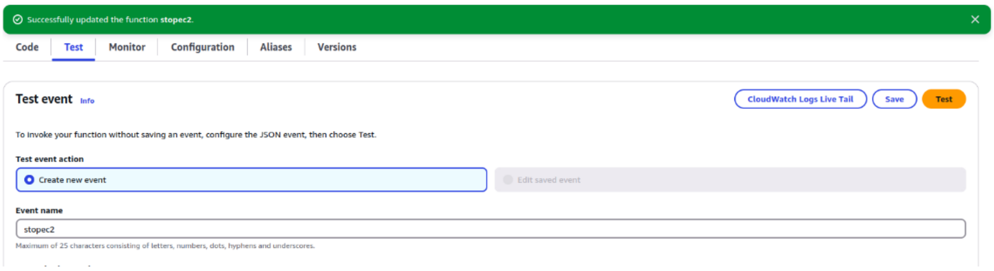
Deploying the Function:

After pasting the code (provided by GitHub Copilot or written manually) into the Lambda function's code editor:

Save the Code: Click the "Deploy" button. This action saves the code and creates a new version of your Lambda function. It's crucial to deploy after every code change.

Testing the Function :

Click **Test** in the Lambda console to trigger the function manually.



Successfully updated the function **stopec2**.

Test event

To invoke your function without saving an event, configure the JSON event, then choose Test.

Test event action

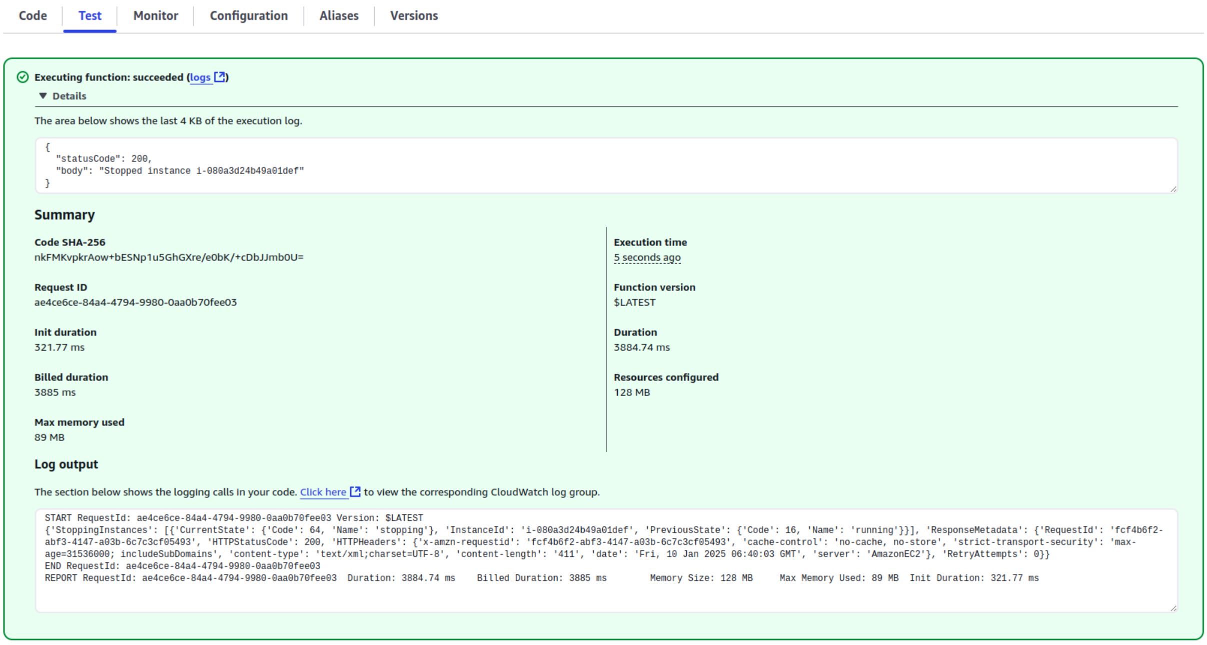
● Create new event ○ Edit saved event

Event name

stopec2

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

After testing, you’ll see the output in the **Execution result** section.



Executing function: succeeded (logs)

▼ Details

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

```
{
  "statusCode": 200,
  "body": "Stopped instance i-080a3d24b49a01def"
}
```

Summary

Code SHA-256 nkFMKvpkrAow+bESNp1u5GhGXre/e0bK/+cDbJJmb0U=	**Execution time** 5 seconds ago
Request ID ae4ce6ce-84a4-4794-9980-0aa0b70fee03	**Function version** $LATEST
Init duration 321.77 ms	**Duration** 3884.74 ms
Billed duration 3885 ms	**Resources configured** 128 MB
Max memory used 89 MB	

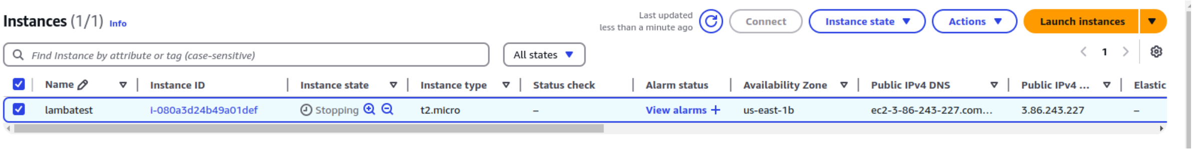
Log output

The section below shows the logging calls in your code. Click here to view the corresponding CloudWatch log group.

```
START RequestId: ae4ce6ce-84a4-4794-9980-0aa0b70fee03 Version: $LATEST
{'StoppingInstances': [{'CurrentState': {'Code': 64, 'Name': 'stopping'}, 'InstanceId': 'i-080a3d24b49a01def', 'PreviousState': {'Code': 16, 'Name': 'running'}}], 'ResponseMetadata': {'RequestId': 'fcf4b6f2-abf3-4147-a03b-6c7c3cf05493', 'HTTPStatusCode': 200, 'HTTPHeaders': {'x-amzn-requestid': 'fcf4b6f2-abf3-4147-a03b-6c7c3cf05493', 'cache-control': 'no-cache, no-store', 'strict-transport-security': 'max-age=31536000; includeSubDomains', 'content-type': 'text/xml;charset=UTF-8', 'content-length': '411', 'date': 'Fri, 10 Jan 2025 06:40:03 GMT', 'server': 'AmazonEC2'}, 'RetryAttempts': 0}}
END RequestId: ae4ce6ce-84a4-4794-9980-0aa0b70fee03
REPORT RequestId: ae4ce6ce-84a4-4794-9980-0aa0b70fee03  Duration: 3884.74 ms    Billed Duration: 3885 ms       Memory Size: 128 MB     Max Memory Used: 89 MB  Init Duration: 321.77 ms
```

The console will display the execution results, including logs, execution time, and any returned values.

Check the EC2 console to confirm that the instance has stopped.



☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic
☑	lambatest	i-080a3d24b49a01def	Stopping	t2.micro	–	View alarms +	us-east-1b	ec2-3-86-243-227.com...	3.86.243.227	–

NOTE: If you are using an AWS Free Tier account, remember to delete all resources created as part of this tutorial. Leaving resources running will result in charges once Free Tier limits are surpassed.