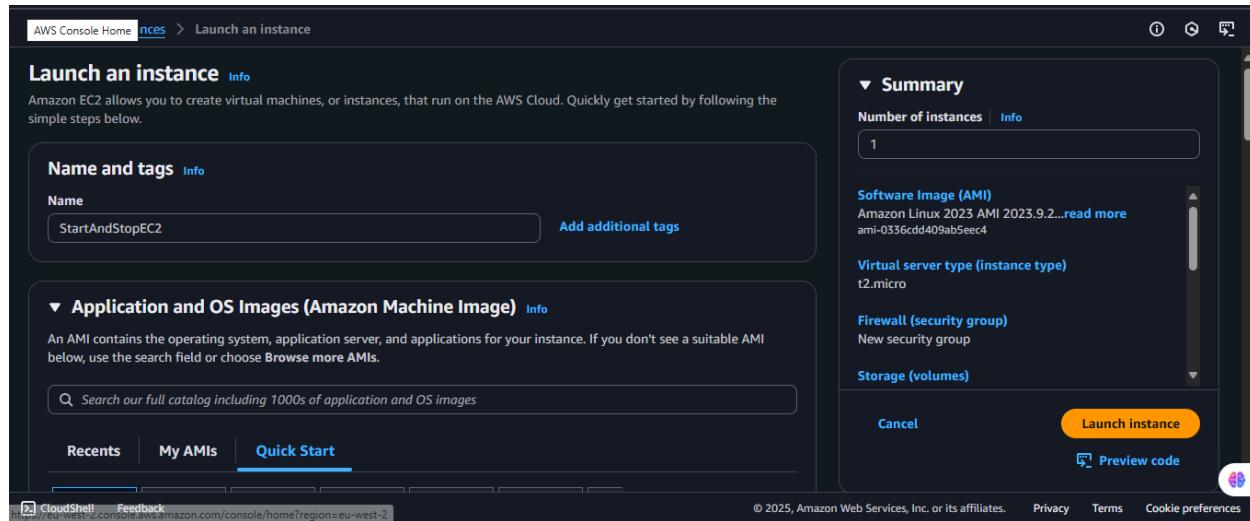


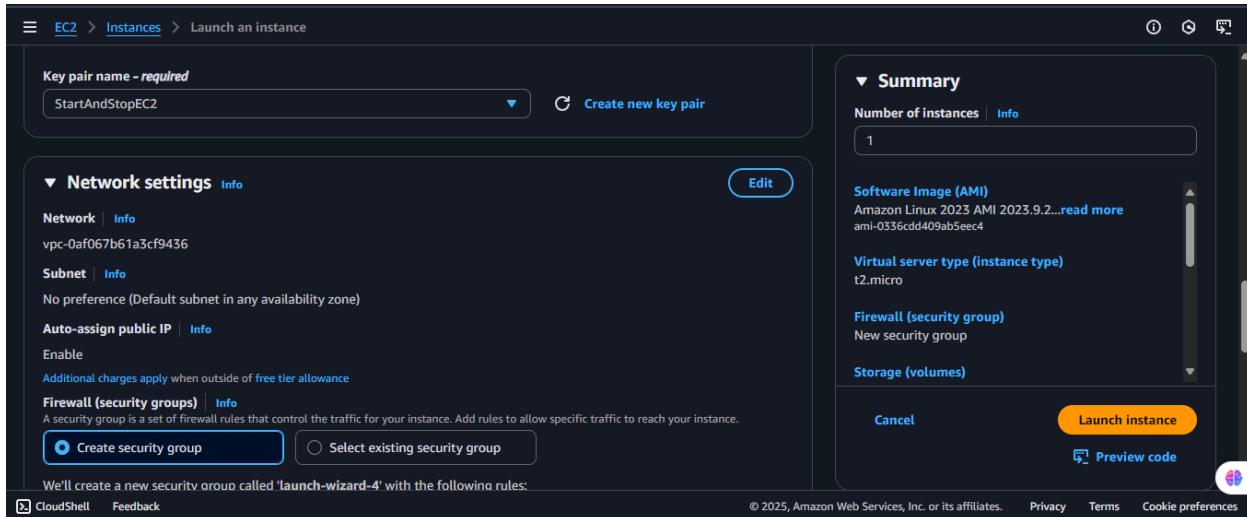
Automate Start-Stop EC2 Instances Using Lambda - Step-by-Step Guide

1. Launch the instance
2. Create the IAM Policy
3. Create IAM role
4. Create the 2 Lambda Function for start and stop
5. Cloud Watch
 1. Create a role for start and stop.

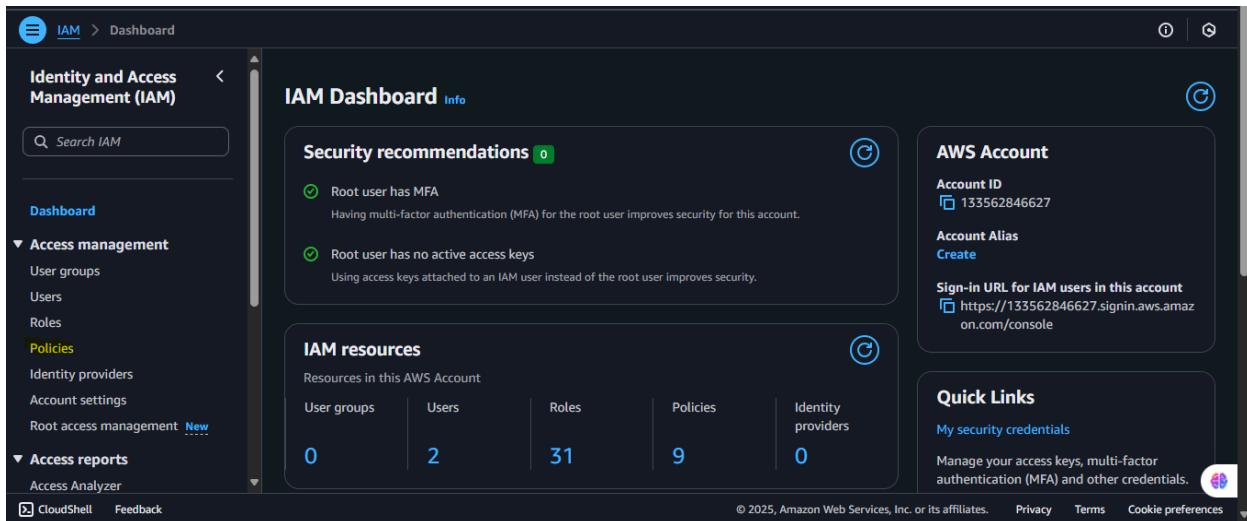
Step 1: Launch the EC2 Instance

- Navigate to EC2 Console.
- Launch a new EC2 instance with the desired configuration.
- Note the **Instance ID** and **Region**.





1. Go to IAM > Policies > Create policy
2. Choose Service: EC2
3. Choose Actions: Start Instances, Stop Instances



Policies (1402) Info

A policy is an object in AWS that defines permissions.

Policy name	Type	Used as	Description
AccessAnalyzerServiceRole	AWS managed	None	Allow Access Analyzer to analyze resources
AdministratorAccess	AWS managed - job function	Permissions policy (2), Bou...	Provides full access to AWS services an...
AdministratorAccess	AWS managed	Permissions policy (1)	Grants account administrative permis...
AdministratorAccess	AWS managed	Permissions policy (1)	Grants account administrative permis...
AIOpsAssistantPolicy	AWS managed	Permissions policy (1)	Provides ReadOnly permissions requir...
AIOpsConsoleAdministrator	AWS managed	Permissions policy (1)	Grants full access to Amazon AI Opera...
AIOpsOperatorAccess	AWS managed	Permissions policy (1)	Grants access to the Amazon AI Opera...

Specify permissions Info

Add permissions by selecting services, actions, resources, and conditions. Build permission statements using the JSON editor.

Policy editor

EC2 Allow 2 Actions

Specify what actions can be performed on specific resources in EC2.

Actions allowed

Specify actions from the service to be allowed.

Effect Allow Deny

Filter Actions

Manual actions | Add actions

All EC2 actions (ec2:*)

Access level

List (197)

Create policy

<input type="checkbox"/> ResetNetworkInterfaceAttribute Info	<input type="checkbox"/> RestoreAddressToClassic Info	<input type="checkbox"/> RestoreImageFromRecycleBin Info
<input type="checkbox"/> RestoreManagedPrefixListVersion Info	<input type="checkbox"/> RestoreSnapshotFromRecycleBin Info	<input type="checkbox"/> RestoreSnapshotTier Info
<input type="checkbox"/> RevokeClientVpnIngress Info	<input type="checkbox"/> RevokeSecurityGroupEgress Info	<input type="checkbox"/> RevokeSecurityGroupIngress Info
<input type="checkbox"/> RunInstances Info	<input type="checkbox"/> RunScheduledInstances Info	<input type="checkbox"/> SendDiagnosticInterrupt Info
<input type="checkbox"/> SendSpotInstanceInterruptions Info	<input checked="" type="checkbox"/> StartInstances Info	<input type="checkbox"/> StartNetworkInsightsAccessScopeAnalysis Info
<input type="checkbox"/> StartNetworkInsightsAnalysis Info	<input type="checkbox"/> StartVpcEndpointServicePrivateDnsVerification Info	<input checked="" type="checkbox"/> StopInstances Info
<input type="checkbox"/> TerminateClientVpnConnections Info	<input type="checkbox"/> TerminateInstances Info	<input type="checkbox"/> UnassignIpv6Addresses Info
<input type="checkbox"/> UnassignPrivateIpAddresses Info	<input type="checkbox"/> UnassignPrivateNatGatewayAddresses Info	<input type="checkbox"/> UnlockSnapshot Info
<input type="checkbox"/> UnmonitorInstances Info	<input type="checkbox"/> UpdateSecurityGroupRuleDescriptionsEgress Info	<input type="checkbox"/> UpdateSecurityGroupRuleDescriptionsIngress Info
<input type="checkbox"/> WithdrawByoipCidr Info		

Permissions management (16)

Tagging (2)

Step 3: Create IAM Role for Lambda

1. Navigate to **IAM > Roles > Create Role**.
2. **Trusted entity type: AWS Service**
3. **Use case à Service or use case è Lambda**
4. **Permissions policies à Attach the policy created in Step 2.**
5. Name the role (e.g., LambdaEC2StartStopRole) and create it.
6. Role à Trust policy à Edit the **trust policy** and modify it to allow scheduler.amazonaws.com.

Select trusted entity Info

Trusted entity type

- AWS service**
Allow AWS services like EC2, Lambda, or others to perform actions in this account.
- AWS account**
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.
- Web identity**
Allows users federated by the specified external web identity provider to assume this role to perform actions in this account.
- SAML 2.0 federation**
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.
- Custom trust policy**
Create a custom trust policy to enable others to perform actions in this account.

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Add permissions Info

Permissions policies (1) Info
The service-linked role that you selected requires the following policy.

Policy name	Type
<input checked="" type="checkbox"/> AWSServiceRoleForEC2ScheduledInstances	AWS managed

Cancel Previous Next

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

The screenshot shows the 'Name, review, and create' step of the IAM role creation wizard. On the left, a navigation sidebar lists 'Step 1 Select trusted entity', 'Step 2 Add permissions', and 'Step 3 Name, review, and create' (which is highlighted). The main area is titled 'Role details' and contains fields for 'Role name' (set to 'AWSServiceRoleForEC2ScheduledInstances') and 'Description' (set to 'Allows EC2 Scheduled Instances to manage instances on your behalf'). Below these fields is a note about character restrictions. At the bottom of the main area, there's a 'Step 1: Select trusted entities' section with an 'Edit' button and a 'Trust policy' link.

Step 4: Create Lambda Functions (Start and Stop EC2)

1. Create a function
2. Choose the option **Author from scratch**.
3. **Basic Information à Function name à Give the name**
4. **Runtime à** Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby. à **Python 3.13**

The screenshot shows the 'Create function' wizard. At the top, it says 'Choose one of the following options to create your function.' Three options are available: 'Author from scratch' (selected), 'Use a blueprint', and 'Container image'. The 'Basic information' section is expanded, showing a 'Function name' field with the value 'Adish-Instance-Management-Start'. Below this, the 'Runtime' section is expanded, showing a dropdown menu set to 'Python 3.13'. The 'Architecture' section is partially visible at the bottom.

Lambda > Functions > Create function

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/Lambda_function-role-9i5yg9oh [View the Lambda_function-role-9i5yg9oh role](#) on the IAM console.

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.

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Lambda > Functions > Create function

Security & governance

Code signing [Info](#)
Use code signing configurations to ensure that the code has been signed by an approved source and has not been altered since signing.
 Enable

Encryption with an AWS KMS customer managed key [Info](#)
By default, Lambda encrypts the .zip file archive using an AWS owned key.
 Enable

Tags [Info](#)
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, track your AWS costs, and enforce attribute-based access control.
 Enable

Key Value - optional
Action Auto-Start [Remove](#)

Add new tag You can add up to 49 more tags.

Cancel Create function © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

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.

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.
 ↻

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#) ▼

Lambda > Functions > Create function

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.
 ↻ View the Lambda_function-role-9i5yg9oh role ↗ on the IAM console.

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.

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#) ▼

Lambda > Functions > Create function

Security & governance

Code signing Info
Use code signing configurations to ensure that the code has been signed by an approved source and has not been altered since signing.
 Enable

Encryption with an AWS KMS customer managed key Info
By default, Lambda encrypts the .zip file archive using an AWS owned key.
 Enable

Tags Info
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, track your AWS costs, and enforce attribute-based access control.
 Enable

Key	Value - optional
<input type="text" value="Action"/>	<input type="text" value="Auto-Stop"/> X Remove

Add new tag

You can add up to 49 more tags.

Cancel Create function ↗

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#) ▼

Lambda > Functions

Functions (2)

Function name	Description	Package type	Runtime	Last modified
Adish-Instance-Management-Start	-	Zip	Python 3.13	28 minutes ago
Adish-Instance-Management-Stop	-	Zip	Python 3.13	51 minutes ago

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Lambda > Functions > Adish-Instance-Management-Start

Configuration

General configuration

Description	Memory	Ephemeral storage
-	128 MB	512 MB

Triggers

Permissions

Destinations

Function URL

Environment variables

Tags

VPC

RDS databases

Monitoring and operations tools

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Lambda > Functions > Adish-Instance-Management-Start

Code source

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

lambda_function.py

```
1 import boto3
2
3 def lambda_handler(event, context):
4     ec2 = boto3.client('ec2')
5     instance_ids = ['i-033231fd0fcfa97167'] # replace with your instance IDs
6     ec2.start_instances(InstanceIds=instance_ids)
7     print('Started EC2 instance:', instance_ids)
```

EXPLORER

ADISH-INSTANCE-MANAGEMENT-START

lambda_function.py

DEPLOY

Deploy (Ctrl+Shift+U)

Test (Ctrl+Shift+I)

PROBLEMS OUTPUT CODE REFERENCE LOG TERMINAL Tasks

CloudShell Feedback

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

```

import boto3

def lambda_handler(event, context):

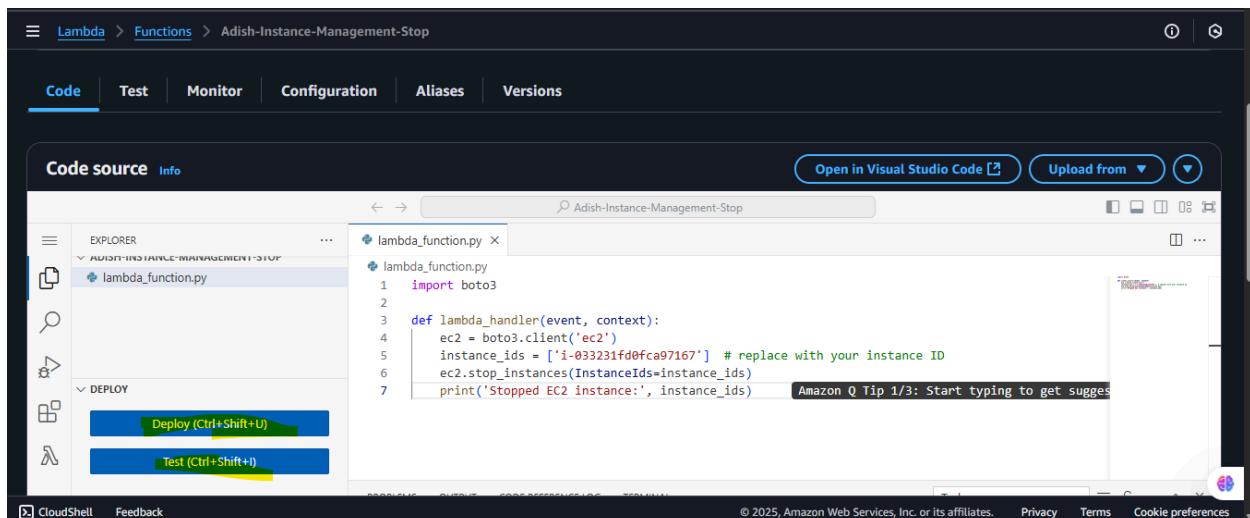
    ec2 = boto3.client('ec2')

    instance_ids = ['i-033231fd0fca97167'] # replace with your instance ID

    ec2.start_instances(InstanceIds=instance_ids)

    print('Started EC2 instance:', instance_ids)

```



```

import boto3

def lambda_handler(event, context):

    ec2 = boto3.client('ec2')

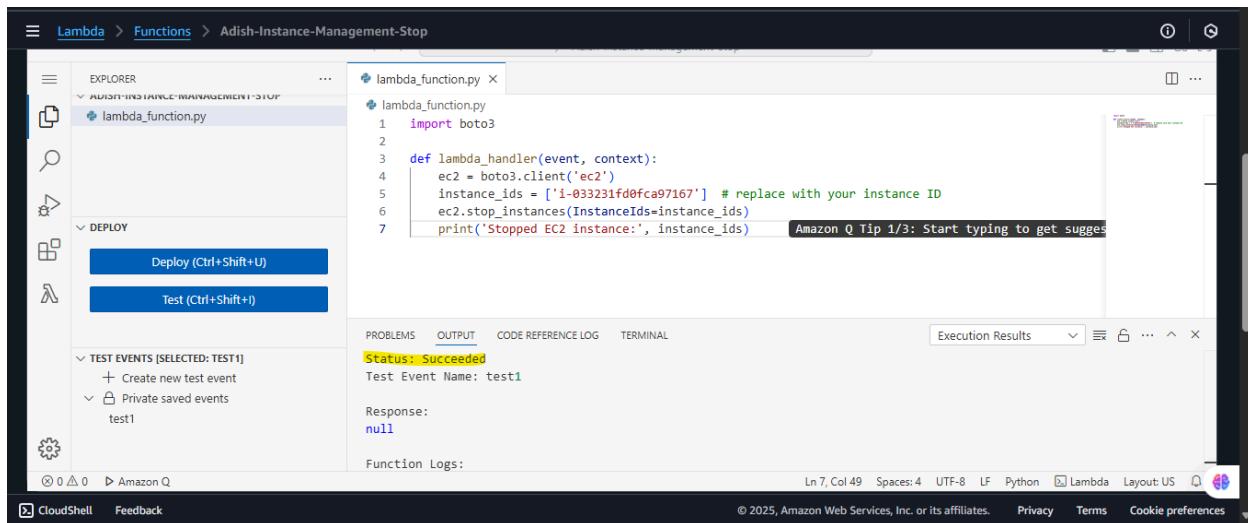
    instance_ids = ['i-033231fd0fca97167'] # replace with your instance ID

    ec2.stop_instances(InstanceIds=instance_ids)

```

```
print('Stopped EC2 instance:', instance_ids)
```

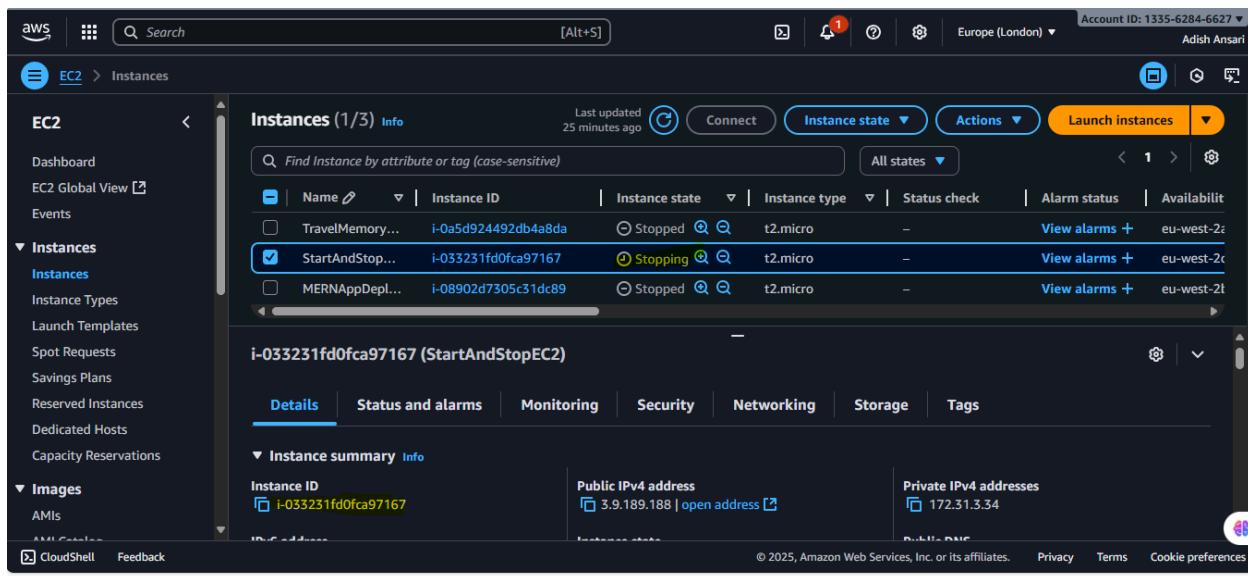
Testing



The screenshot shows the AWS Lambda Function Editor interface. The left sidebar shows the function name "Adish-Instance-Management-Stop". The main area displays the code in "lambda_function.py":

```
lambda_function.py
1 import boto3
2
3 def lambda_handler(event, context):
4     ec2 = boto3.client('ec2')
5     instance_ids = ['i-033231fd0fca97167'] # replace with your instance ID
6     ec2.stop_instances(InstanceIds=instance_ids)
7     print('Stopped EC2 instance:', instance_ids)
```

The "TEST EVENTS" section shows a selected event named "test1". The "OUTPUT" tab of the "PROBLEMS" panel indicates "Status: Succeeded". The "Response" field shows "null". The "Function Logs" section is empty.



The screenshot shows the AWS EC2 Instances page. The left sidebar lists various EC2-related services like Dashboard, Global View, Events, Instances, Launch Templates, and more. The main table shows three instances:

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability zone
TravelMemory...	i-0a5d924492db4a8da	Stopped	t2.micro	-	View alarms +	eu-west-2a	
StartAndStop...	i-033231fd0fca97167	Stopping	t2.micro	-	View alarms +	eu-west-2c	
MERNAppDepl...	i-08902d7305c31dc89	Stopped	t2.micro	-	View alarms +	eu-west-2l	

A modal window for instance "i-033231fd0fca97167 (StartAndStopEC2)" is open, showing the "Details" tab. It displays the instance ID "i-033231fd0fca97167", its public IPv4 address "3.9.189.188", and its private IPv4 address "172.31.3.34".

Lambda > Functions > Adish-Instance-Management-Start

EXPLORER
ADISH-INSTANCE-MANAGEMENT-START
lambda_function.py

DEPLOY
Deploy (Ctrl+Shift+U)
Test (Ctrl+Shift+I)

TEST EVENTS [SELECTED: TESTSTART]
Create new test event
Private saved events
TestStart

lambda_function.py

```
1 import boto3
2
3 def lambda_handler(event, context):
4     ec2 = boto3.client('ec2')
5     instance_ids = ['i-033231fd0fca97167'] # replace with your instance IDs
6     ec2.start_instances(InstanceIds=instance_ids)
7     print("Started EC2 instance:", instance_ids)
```

PROBLEMS OUTPUT CODE REFERENCE LOG TERMINAL

Status: Succeeded
Test Event Name: TestStart
Response:
null

Function Logs:

Ln 7, Col 15 Spaces: 4 UTF-8 LF Python Lambda Layout: US

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

EC2 > Instances

EC2

Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
AMIs

Instances (1/3) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability zone
TravelMemory...	i-0a5d924492db4a8da	Stopped	t2.micro	-	View alarms +	eu-west-2a
StartAndStop...	i-033231fd0fca97167	Running	t2.micro	Initializing	View alarms +	eu-west-2c
MERNAppDepl...	i-08902d7305c31dc89	Stopped	t2.micro	-	View alarms +	eu-west-2l

Last updated less than a minute ago

Connect Instance state Actions Launch instances

i-033231fd0fca97167 (StartAndStopEC2)

Details Status and alarms Monitoring Security Networking Storage Tags

Instance summary

Instance ID i-033231fd0fca97167	Public IPv4 address 18.133.138.235 open address	Private IPv4 addresses 172.31.3.34
------------------------------------	------------------------------------------------------	---------------------------------------

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences