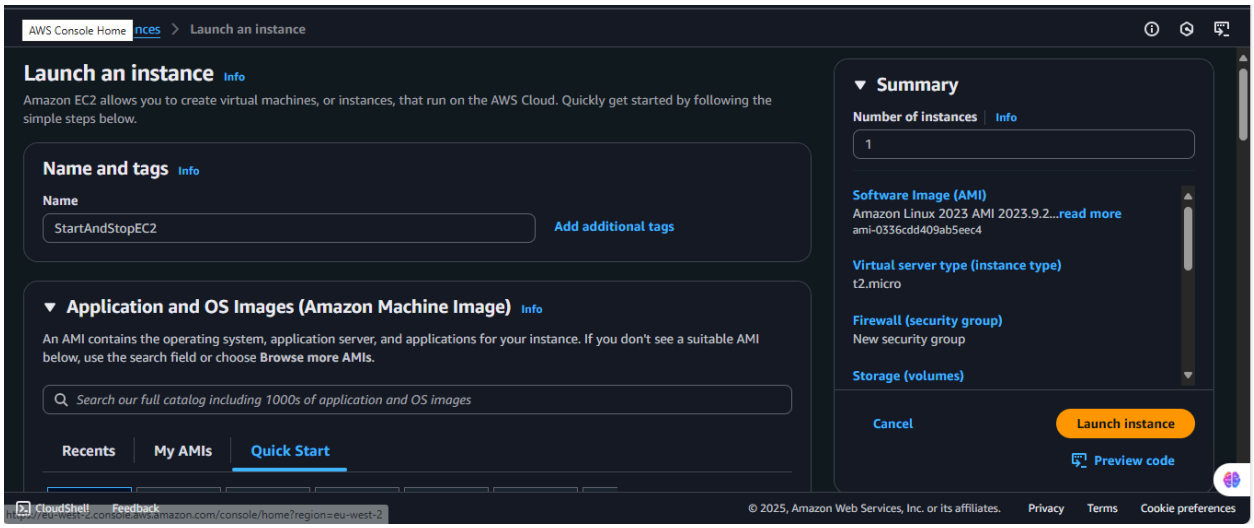
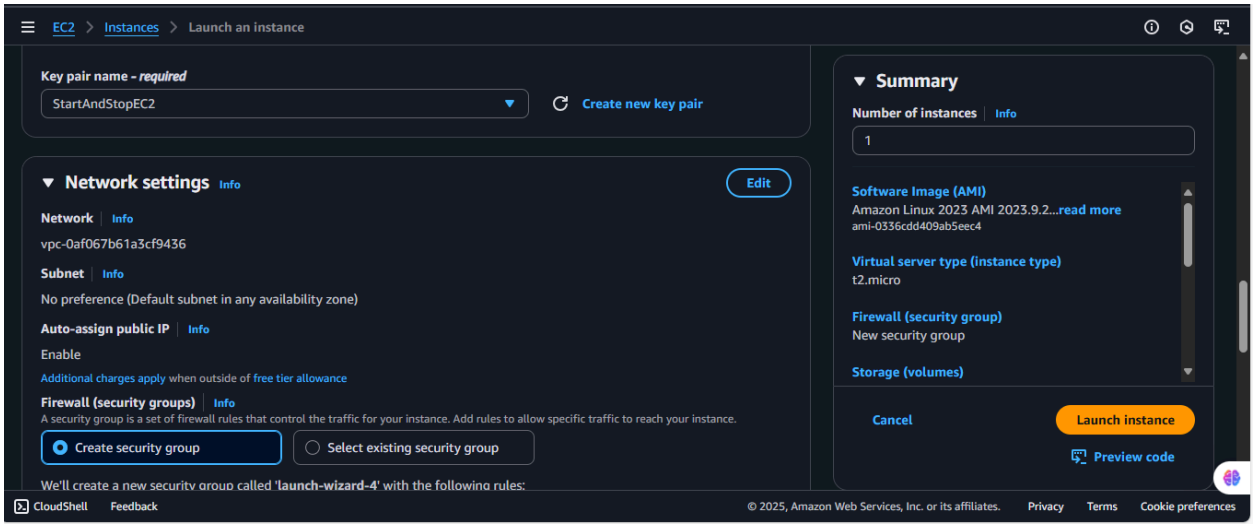
Automate Start-Stop EC2 Instances Using Lambda - Step-by-Step Guide

1. Lunch 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.

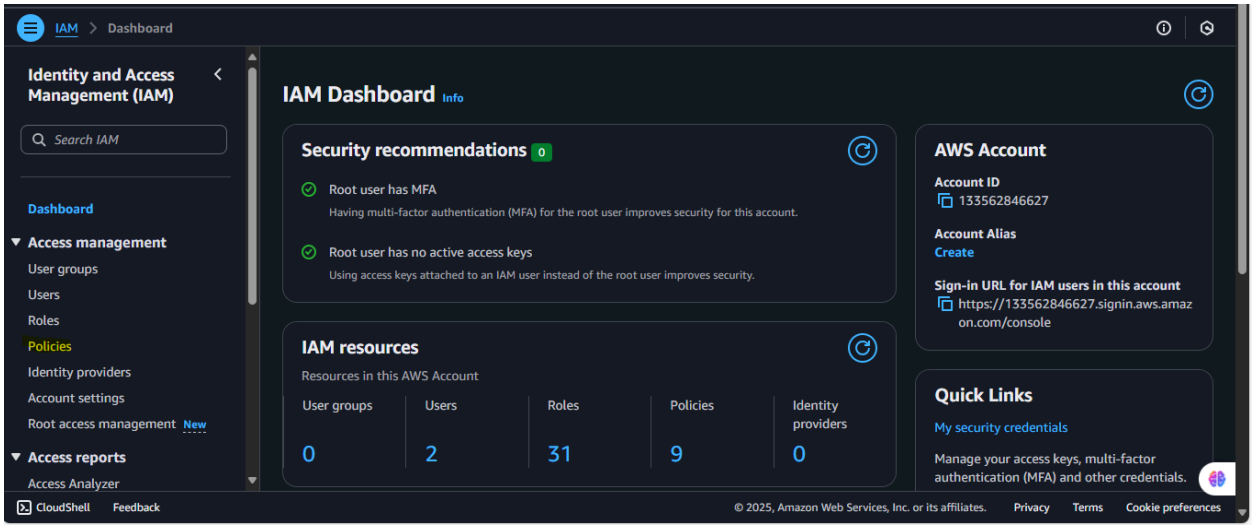
Step 1: Launch the EC2 Instance

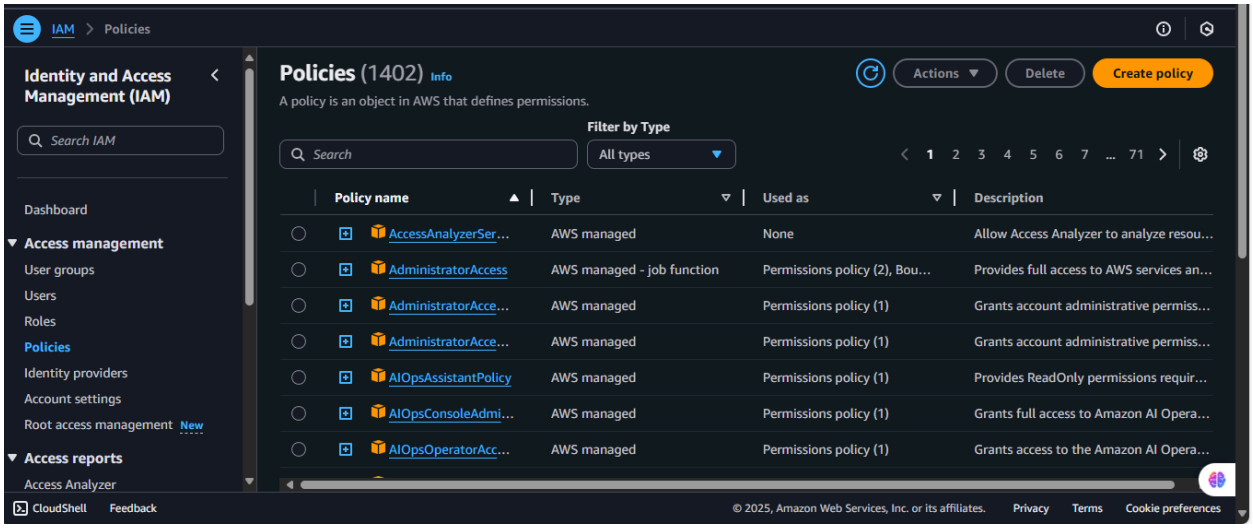
- Navigate to EC2 Console.
- Launch a new EC2 instance with the desired configuration.
- Note the **Instance ID** and **Region**.

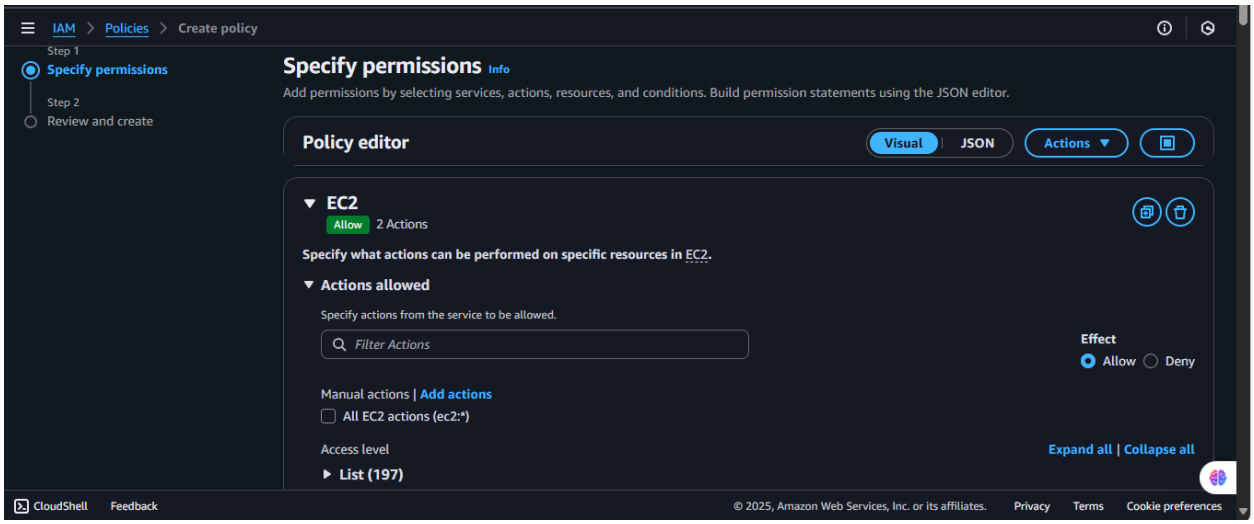


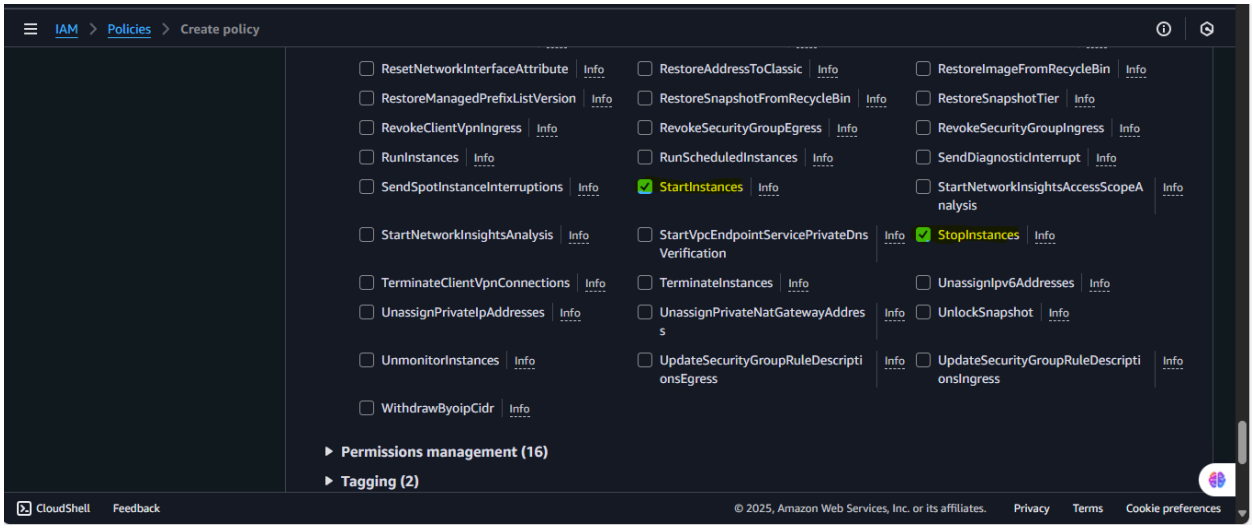


1. **Go to IAM > Policies > Create policy**
2. **Choose Service: EC2**
3. **Choose Actions: Start Instances, Stop Instances**



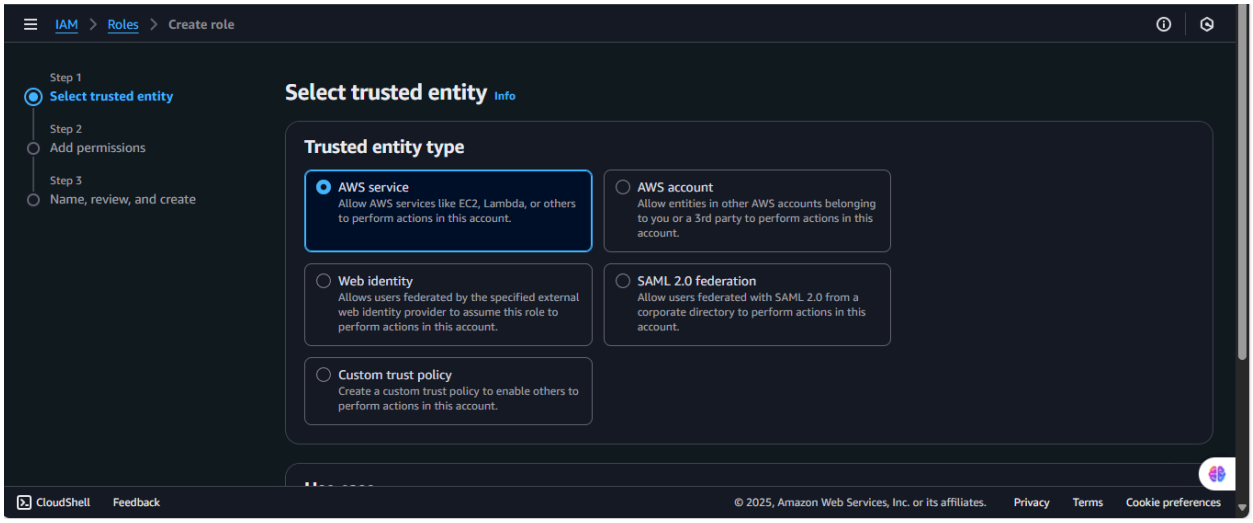


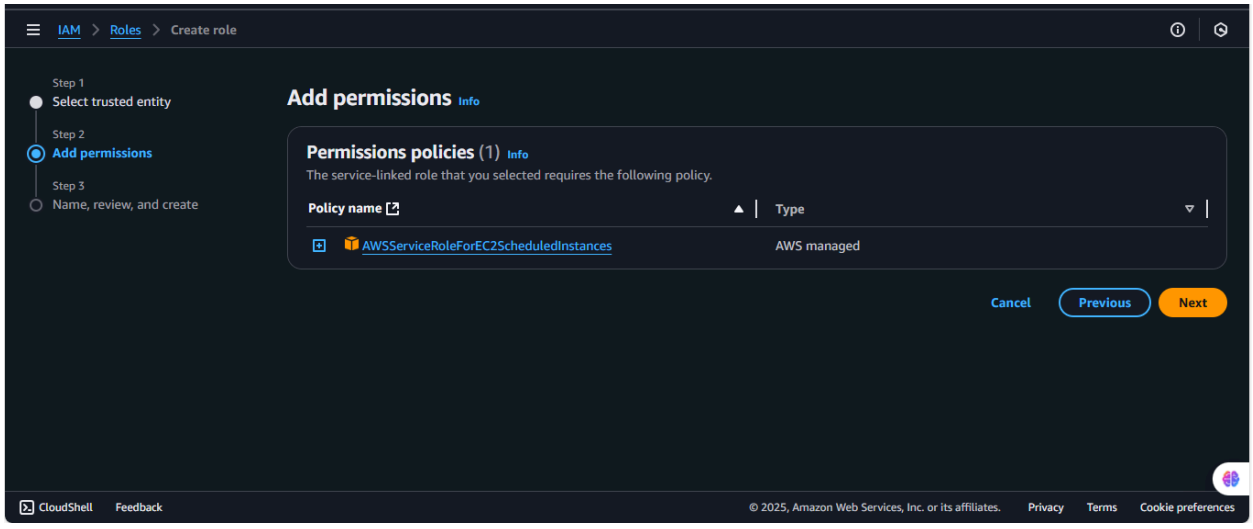


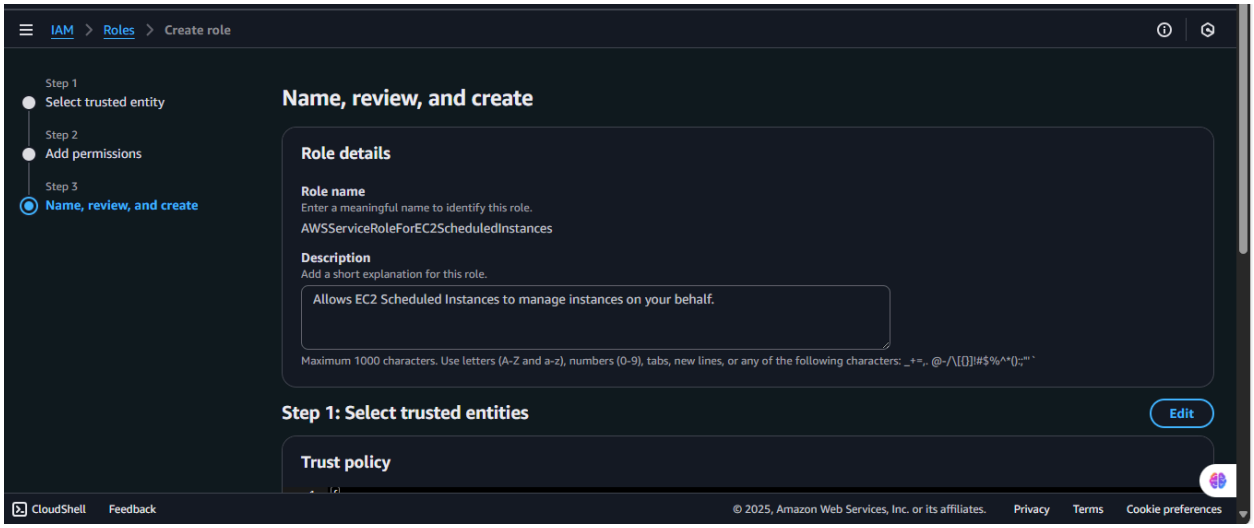


Step 3: Create IAM Role for Lambda

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.

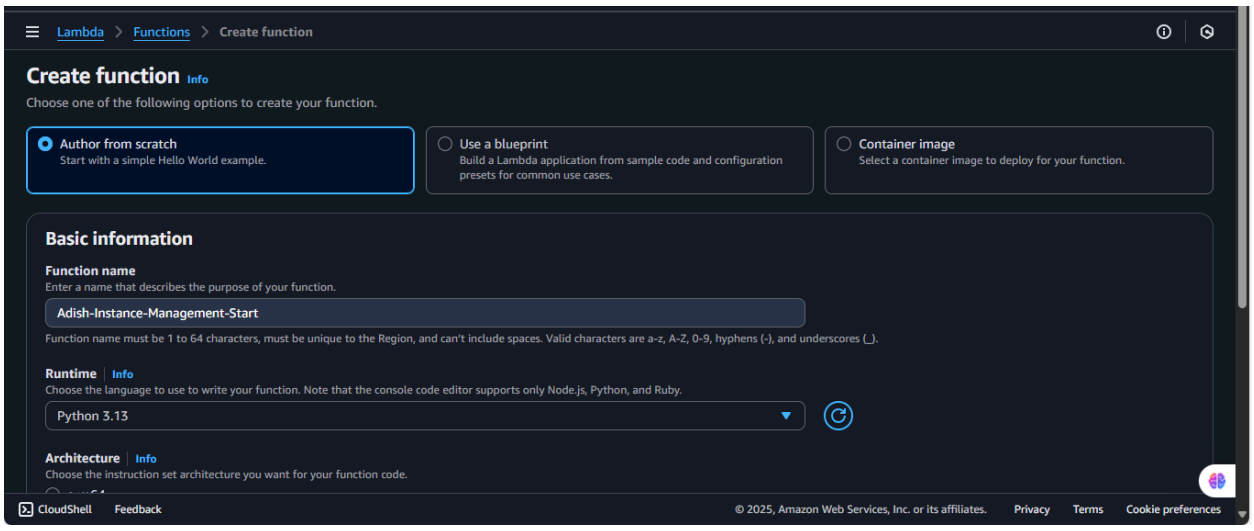


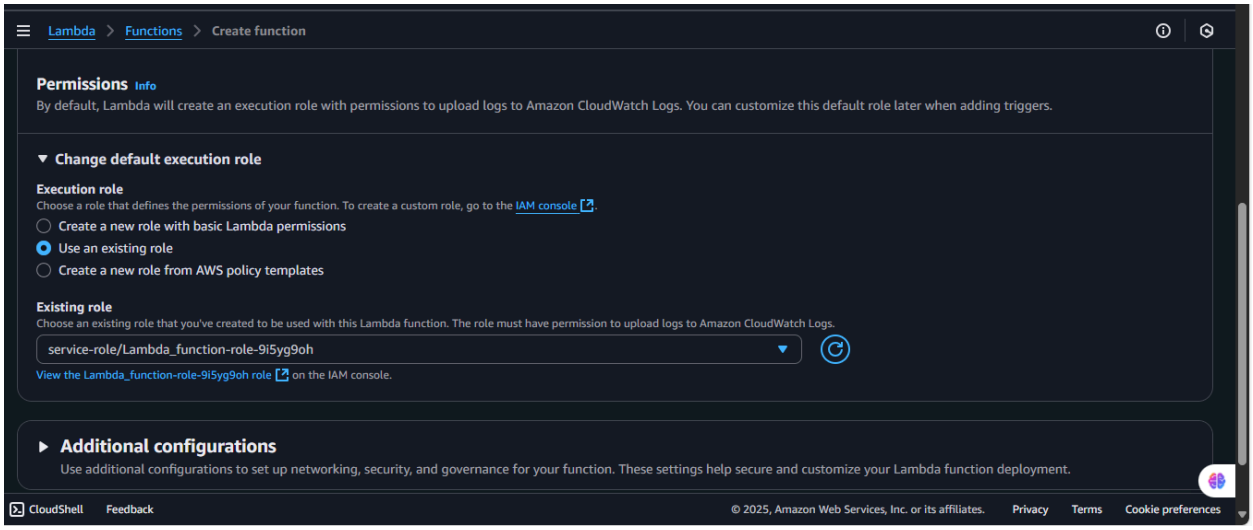


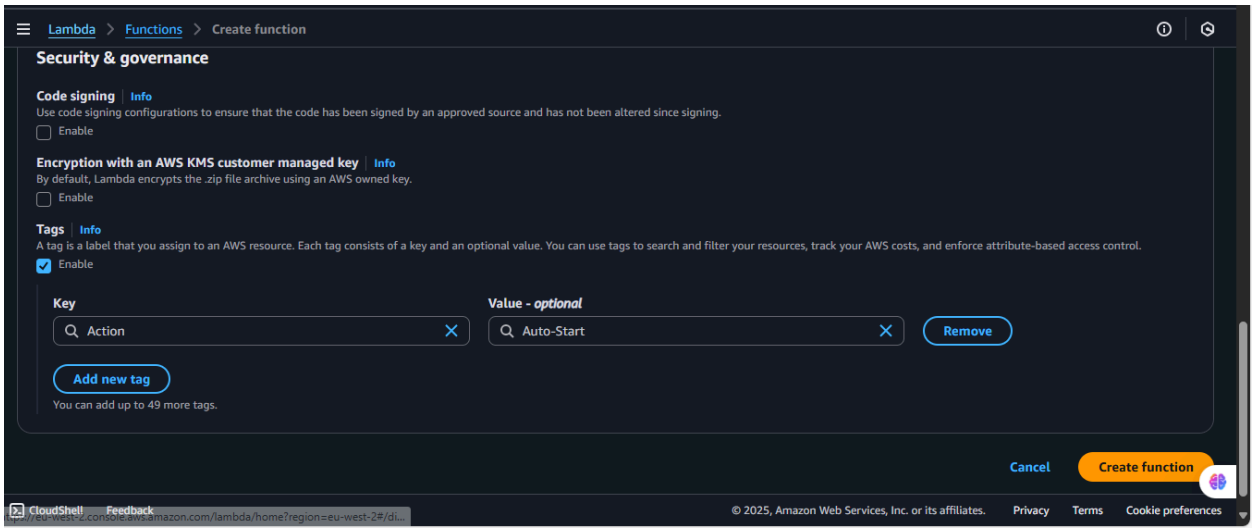


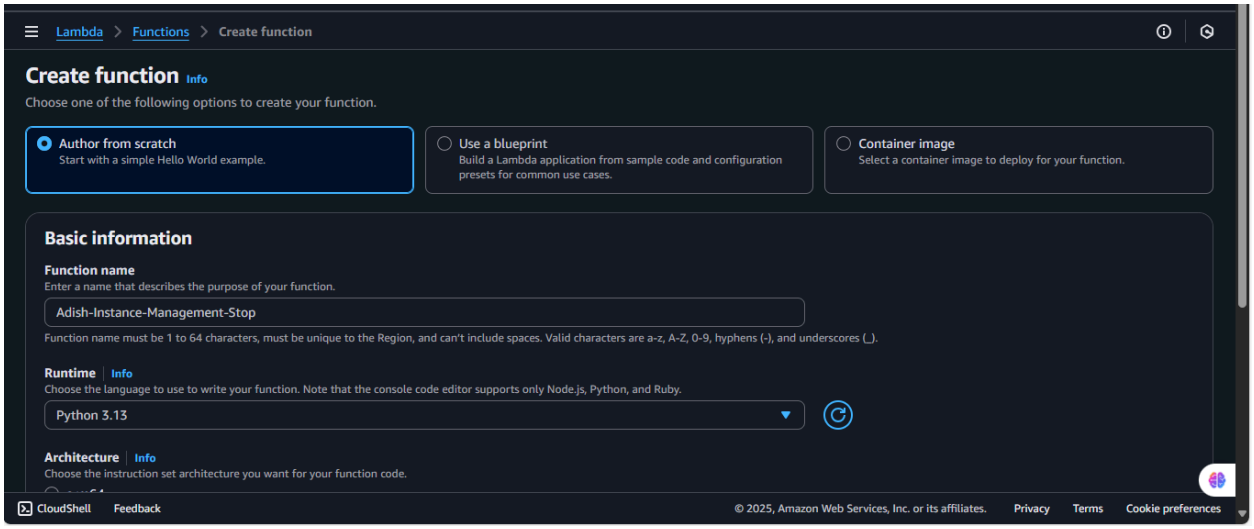
Step 4: Create Lambda Functions (Start and Stop EC2)

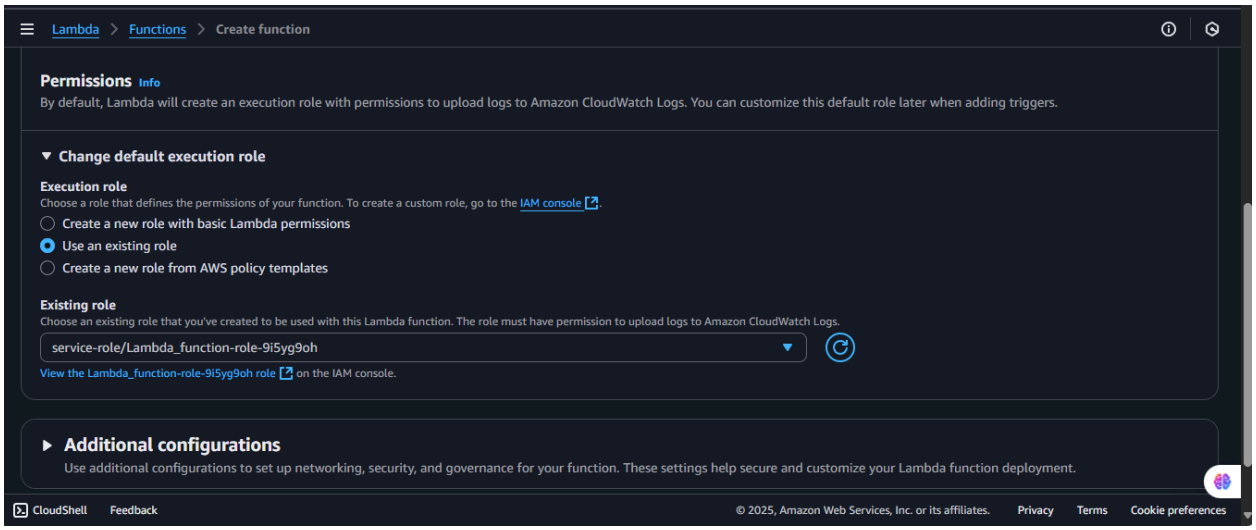
1. Create a function
2. Choose the option **Author from scratch.**
3. **Basic Information à Function name à Give the name**
4. **Runtime à** Choose the language to use to write your function. Note that the console code editor supports only Node.js, Python, and Ruby. à **Python 3.13**

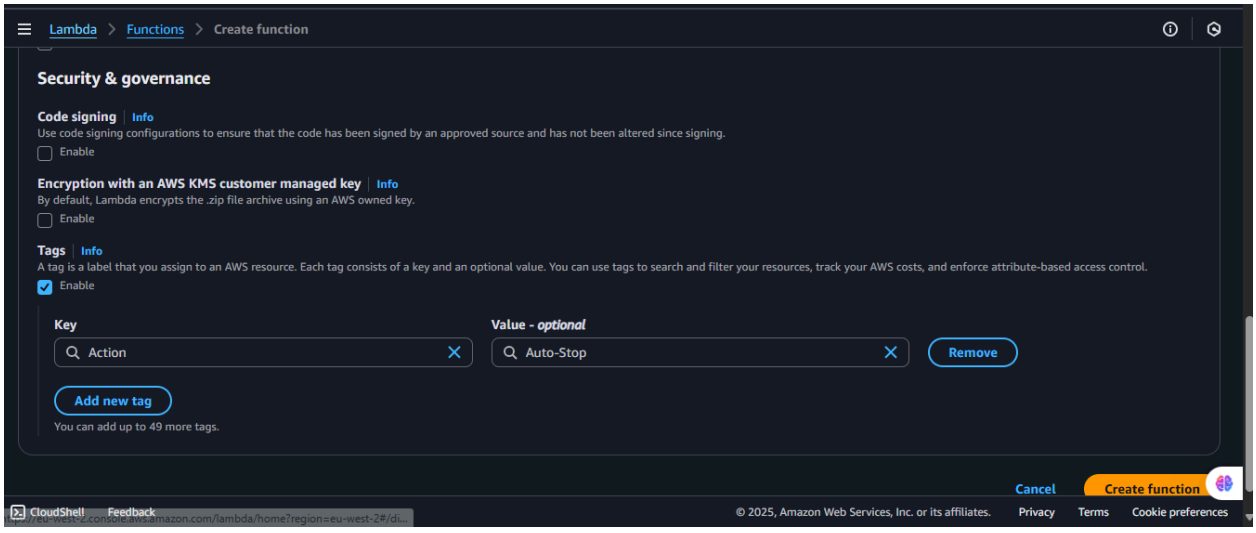


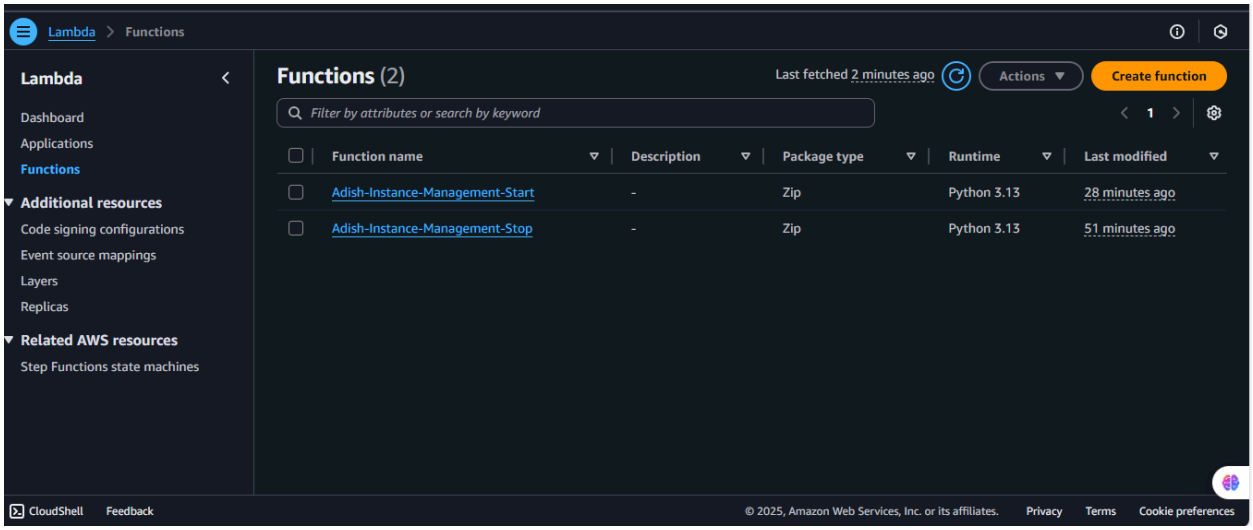

Lambda > Functions > Create function
Permissions Info
By default, Lambda will create an execution role with permissions to upload logs to Amazon CloudWatch Logs. You can customize this default role later when adding triggers.
Change default execution role
Execution role
Choose a role that defines the 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

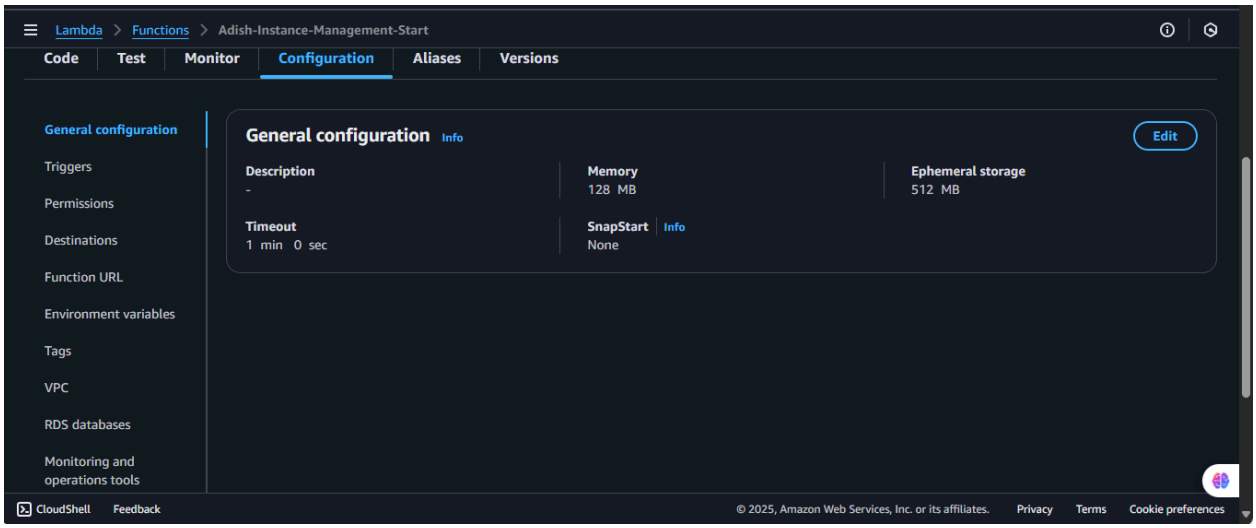

Lambda > Functions > Create function
Security & governance
Code signing Info
Use code signing configurations to ensure that the code has been signed by an approved source and has not been altered since signing.
Enable
Encryption with an AWS KMS customer managed key Info
By default, Lambda encrypts the .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
Action
Value - optional
Auto-Start
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

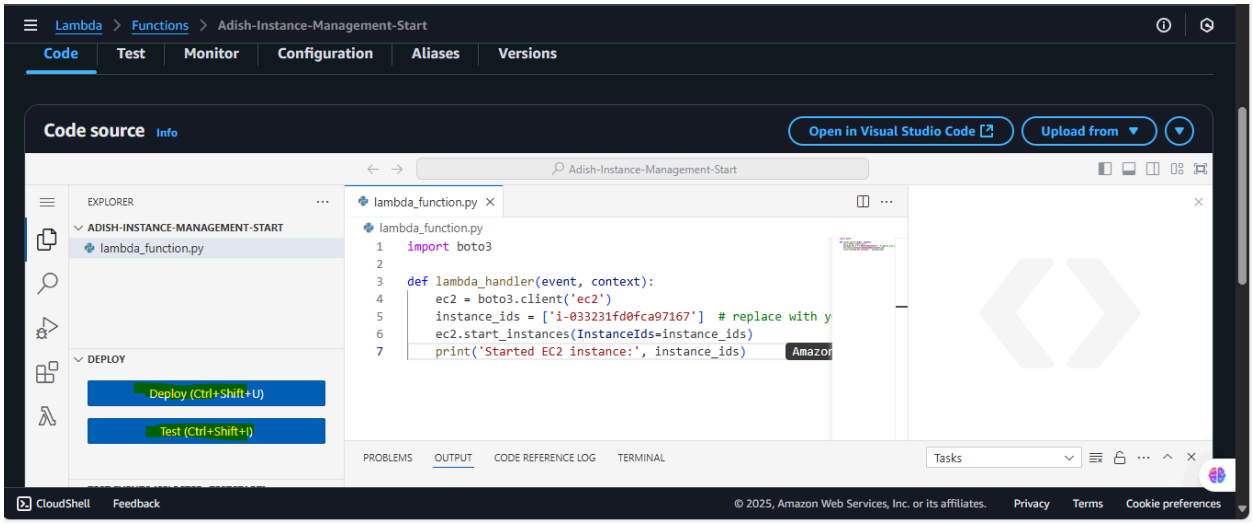

Lambda > Functions > Create function
Create function Info
Choose one of the following options to create your function.
Author from scratch
Start with a simple Hello World example.
Use a blueprint
Build a Lambda application from sample code and configuration presets for common use cases.
Container image
Select a container image to deploy for your function.
Basic information
Function name
Enter a name that describes the purpose of your function.
Adish-Instance-Management-Stop
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.
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.
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

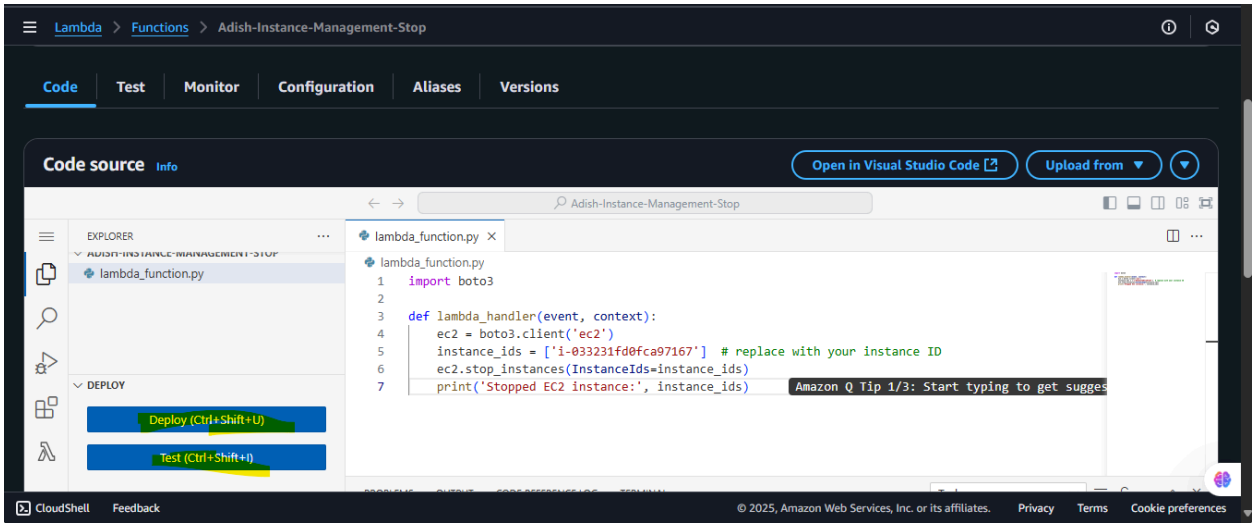

Lambda > Functions > Create function
Security & governance
Code signing Info
Use code signing configurations to ensure that the code has been signed by an approved source and has not been altered since signing.
Enable
Encryption with an AWS KMS customer managed key Info
By default, Lambda encrypts the .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
Action
Value - optional
Auto-Stop
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
Lambda
Dashboard
Applications
Functions
Additional resources
Code signing configurations
Event source mappings
Layers
Replicas
Related AWS resources
Step Functions state machines
Functions (2)
Last fetched 2 minutes ago
Actions
Create function
Filter by attributes or search by keyword
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
Code
Test
Monitor
Configuration
Aliases
Versions
General configuration
Triggers
Permissions
Destinations
Function URL
Environment variables
Tags
VPC
RDS databases
Monitoring and operations tools
General configuration Info
Edit
Description
-
Memory
128 MB
Ephemeral storage
512 MB
Timeout
1 min 0 sec
SnapStart Info
None
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Lambda > Functions > Adish-Instance-Management-Start
Code
Test
Monitor
Configuration
Aliases
Versions
Code source Info
Open in Visual Studio Code
Upload from
EXPLORER
ADISH-INSTANCE-MANAGEMENT-START
lambda_function.py
DEPLOY
Deploy (Ctrl+Shift+U)
Test (Ctrl+Shift+I)
import boto3
def lambda_handler(event, context):
ec2 = boto3.client('ec2')
instance_ids = ['i-033231fd0fca97167'] # replace with y
ec2.start_instances(InstanceIds=instance_ids)
print('Started EC2 instance:', instance_ids)
PROBLEMS
OUTPUT
CODE REFERENCE LOG
TERMINAL
Tasks
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

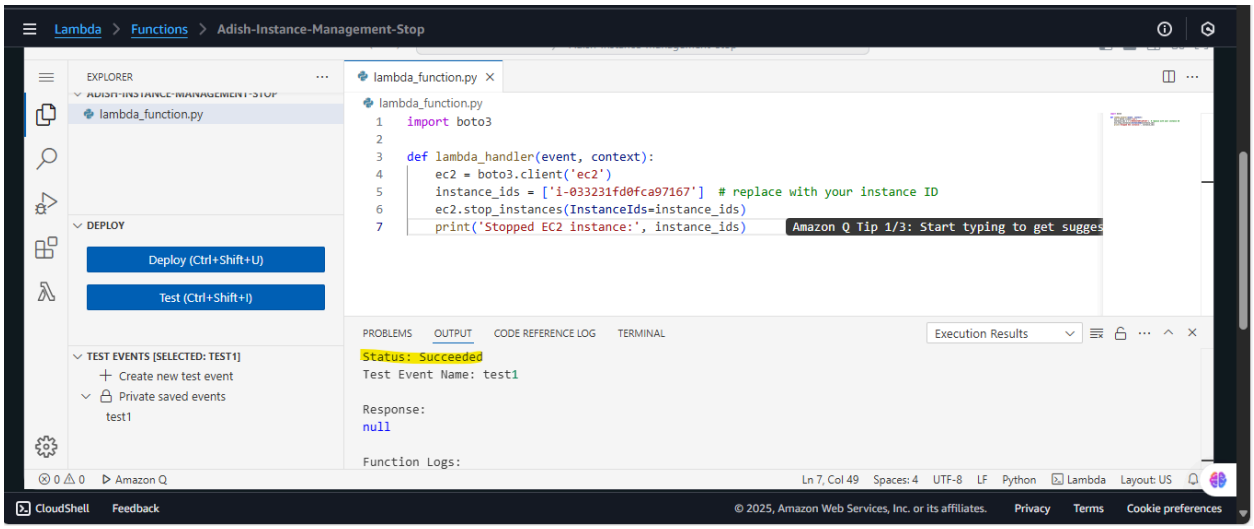
```python
import boto3


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)
```

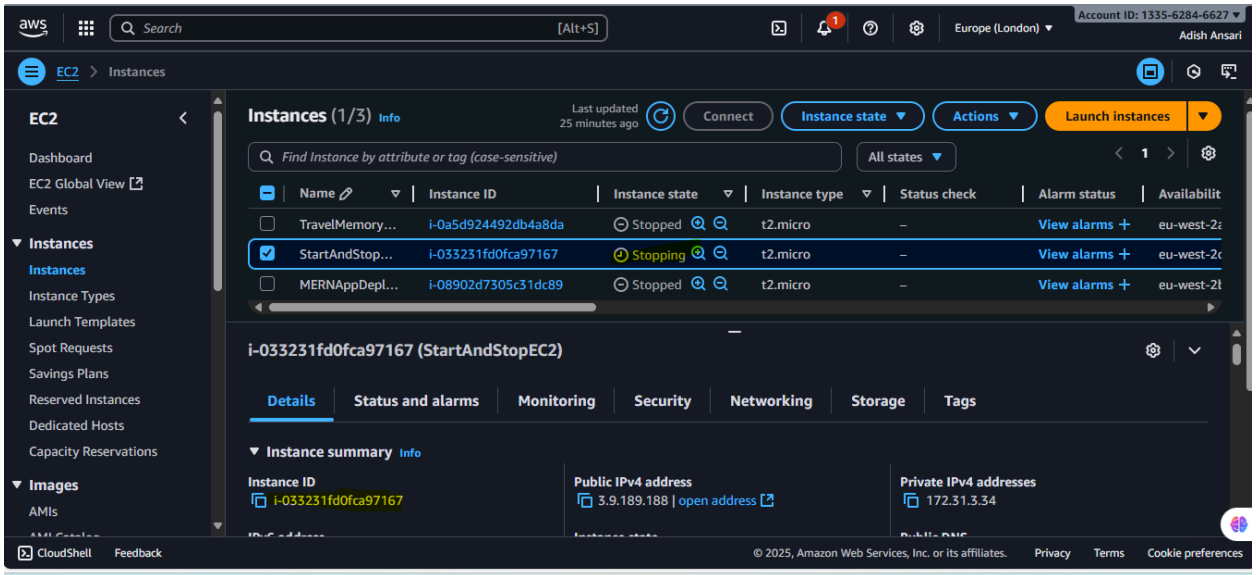


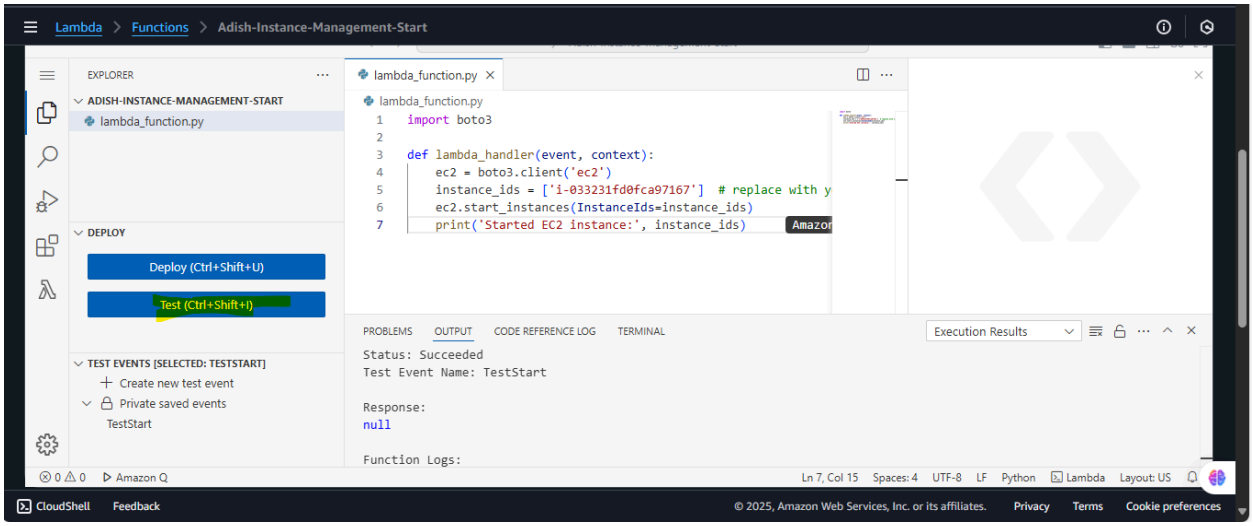

```python
import boto3


def lambda_handler(event, context):
    ec2 = boto3.client('ec2')
    instance_ids = ['i-033231fd0fca97167']  # replace with your instance ID
    ec2.stop_instances(InstanceIds=instance_ids)
```

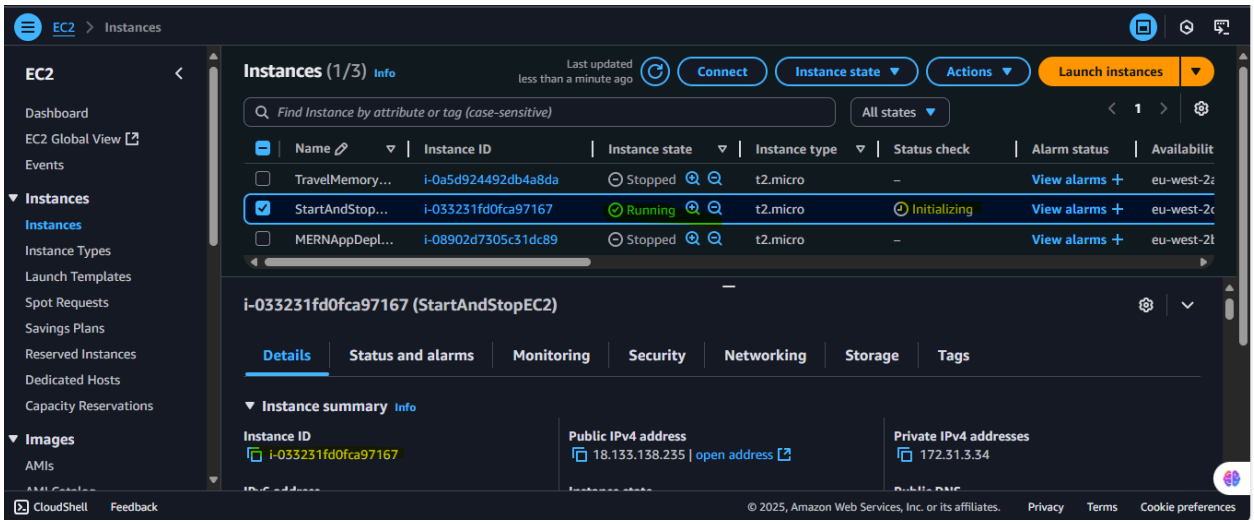
```
print('Stopped EC2 instance:', instance_ids)
```

Testing






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 y
6 ec2.start_instances(InstanceIds=instance_ids)
7 print('Started EC2 instance:', instance_ids)
PROBLEMS OUTPUT CODE REFERENCE LOG TERMINAL
Execution Results
Status: Succeeded
Test Event Name: TestStart
Response:
null
Function Logs:
Ln 7, Col 15 Spaces: 4 UTF-8 LF Python Lambda Layout: US
CloudShell Feedback
© 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
Last updated less than a minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
TravelMemory... i-0a5d924492db4a8da Stopped t2.micro – View alarms
StartAndStop... i-033231fd0fca97167 Running t2.micro Initializing View alarms
MERNAppDepl... i-08902d7305c31dc89 Stopped t2.micro – View alarms
i-033231fd0fca97167 (StartAndStopEC2)
Details
Status and alarms
Monitoring
Security
Networking
Storage
Tags
Instance summary Info
Instance ID
i-033231fd0fca97167
Public IPv4 address
18.133.138.235 | open address
Private IPv4 addresses
172.31.3.34
CloudShell Feedback
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences