

Project 2

How do I stop and start Amazon EC2 instances at regular intervals using Lambda:

Description:

In your IT real-time world, we may need to schedule the EC2 instance or RDS instance to restart at regular intervals; it all depends on the needs of your project (it could be hourly, daily, weekly, or monthly). We have multiple ways to do this job. For instance, we can use the corn jobs (Linux) or schedule tasks" (Windows) to run the script. and yet another method for performing a manual restart. However, in AWS, we have a serverless service to handle this task which is **LAMBDA**.

Synopsis:

Scenario 1: I want to reduce my Amazon Elastic Compute Cloud (Amazon EC2) usage by stopping and starting my EC2 instances automatically.

Scenario 2: Create an IAM policy and execution role for your Lambda function

Scenario 3: Create Lambda functions that stop and start your EC2 instances

Scenario 3: Create Lambda functions that stop and start your EC2 instances

Scenario 4: Check the status of your EC2 instances

Scenario 5: Create EventBridge rules that run your Lambda functions

Short description

You can use AWS Lambda and Amazon EventBridge to automatically stop and start EC2 instances.

To use Lambda to stop and start EC2 instances at regular intervals, complete the following steps:

1. Create a custom AWS Identity and Access Management (IAM) policy and execution role for your Lambda function.
2. Create Lambda functions that stop and start your EC2 instances.
3. Test your Lambda functions.
4. Create EventBridge rules that run your function on a schedule.

First, we create an EC2 instance.

The screenshot shows the AWS Console Home page. On the left, there's a sidebar with 'Recently visited' services: Lambda, Amazon EventBridge, IAM, EC2, Amazon Machine Learning, VPC, and Route 53. To the right, there's a 'Welcome to AWS' section with three cards: 'Getting started with AWS' (Learn the fundamentals), 'Training and certification' (Learn from AWS experts), and 'What's new with AWS?' (Discover new AWS services, features, and Regions). At the bottom, there's a 'Cost and usage' summary showing current month costs at \$0.00, forecasted month end costs at \$0.03, and last month costs at \$0.00. A note says 'Costs shown are unblended. Learn more'.

The screenshot shows the AWS EC2 Home page. On the left, there's a sidebar with sections like EC2 Global View, Events, Tags, Limits, Instances (with sub-options like Instances Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations), Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), and Network & Security (Security Groups, Elastic IPs, Placement Groups). The main area shows the 'Resources' section with counts for various EC2 components. Below it is the 'Launch instance' section with a 'Launch instance' button and a 'Migrate a server' option. There are also sections for 'Scheduled events' (Asia Pacific (Mumbai)) and 'Migrate a server'. On the right, there are several panels: 'Account attributes' (Supported platforms: VPC, Default VPC: vpc-0af44c8154f5cb36d, Settings, EBS encryption, Zone, EC2 Serial Console, Default credit specification, Console experiments), 'Explore AWS' (10 Things You Can Do Today to Reduce AWS Costs, Save up to 90% on EC2 with Spot Instances, Amazon GuardDuty Malware Protection), and 'Additional information'.

EC2 > Instances > Launch an instance

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS images

Quick Start

Amazon Linux  macOS Ubuntu Windows Red Hat 5 

Browse more AMIs including AMIs from AWS, Marketplace and the Community

Amazon Linux 2023 AMI

Free tier eligible 

arn-0170decf0f8aef04d5de5981e6466096 (64-bit (Arm), usf)
Virtualization: hvm ENA enabled: true Root device type: ebs

Summary

Number of instances Info

Software Image (AMI)
Amazon Linux 2023 AMI 2023.0.2... read more
ami-0571ec0eaf0770bae

Virtual server type (Instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 10 GB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 G of bandwidth to the internet. 

Launch Instance

Amazon Linux 2023 AMI
ami-0376ec8eadcf70aae (64-bit (x86), uefi-preferred) / ami-0405dec981e546698 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Description
Amazon Linux 2023.0.20230322.0 x86_64 HVM kernel-6.1

Architecture Boot mode AMI ID
64-bit (x86) uefi-preferred ami-0376ec8eadcf70aae Verified provider

Number of instances Info
1

Software Image (AMI)
Amazon Linux 2023 AMI 2023.0.2...
ami-0376ec8eadcf70aae

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable). Instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 Gb of bandwidth to the internet.

Key pair (login) Info
You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required
AWS-Mumbai Create new key pair

Network settings Info
VPC - required Info
vpc-daf4f4c154f5cb36d (default)

Cancel Launch instance Review commands

VPC - required

vpc-0af44c8154f5cb36d (default)

Subnet Info

subnet-07fe01df07b591fb7 VPC: vpc-0af44c8154f5cb36d Owner: 846780726677 Availability Zone: ap-south-1a IP addresses available: 4091 CIDR: 172.31.32.0/20

Auto-assign public IP

Enable

Firewall (security groups)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

Security group name - required

launch-wizard-1

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-zA-Z, 0-9, spaces, and _-/(){}[]^<>`~\$`

Description - required

launch-wizard-1 created 2023-04-02T07:03:20.719Z

Inbound security groups rules

Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type: ssh Protocol: TCP Port range: 22

Source type: Anywhere Description: e.g. SSH for admin desktop

0.0.0.0/0

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.0.2... read more am-057e6cecc0ff70aae

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Launch Instance

Cancel Review commands

CloudShell Feedback Language

31°C Mostly sunny

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 12:34 PM 02-04-2023

Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type: ssh Protocol: TCP Port range: 22

Source type: Anywhere Description: e.g. SSH for admin desktop

0.0.0.0/0

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.0.2... read more am-057e6cecc0ff70aae

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Launch Instance

Cancel Review commands

CloudShell Feedback Language

31°C Mostly sunny

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 12:34 PM 02-04-2023

The screenshot shows the AWS EC2 Instances Launch an Instance page. A green success message box displays "Successfully initiated launch of instance i-076b4f653a0db14c1". Below it is a "Launch log" button. The main area is titled "Next Steps" with a search bar. It lists several actions in cards:

- Create billing and free tier usage alerts
- Connect to your instance
- Connect an RDS database
- Create EBS snapshot policy
- Manage detailed monitoring
- Create Load Balancer
- Create AWS budget
- Manage CloudWatch alarms

Each card includes a "Learn more" link and a "Create" button.

The screenshot shows the AWS EC2 Instances page. On the left, a sidebar navigation includes EC2 Dashboard, EC2 Global View, Events, Tags, Limits, Instances (selected), Instances Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, Lifecycle Manager, Network & Security, Security Groups, Elastic IPs, Placement Groups, and CloudShell, Feedback, Language. The main content area shows a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
test server	i-076b4f653a0db14c1	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	ec2-65-2-144-216.ap-s...	65.2.144.216	-

Below the table is a detailed view for the instance i-076b4f653a0db14c1 (test server). The "Details" tab is selected, showing the following information:

- Instance ID: i-076b4f653a0db14c1 (test server)
- Public IPv4 address: 65.2.144.216 [open address]
- Private IPv4 addresses: 172.31.38.128
- Public IPv4 DNS: ec2-65-2-144-216.ap-south-1.compute.amazonaws.com [open address]
- Instance state: Running
- Private IP DNS name (IPv4 only): ip-172-31-38-128.ap-south-1.compute.internal
- Instance type: t2.micro
- VPC ID: vpc-0af44d8154f5cb36d
- Elastic IP addresses: -
- AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. [Learn more]

At the bottom right, there are links for CloudShell, Feedback, Language, and the AWS footer: © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 12:36 PM 02-04-2023

After the instance has been created, we will go to IAM to create a role and policy that is required for the Lambda operation.

The screenshot shows the AWS IAM service interface. On the left, there's a navigation sidebar with links like 'Identity and Access Management (IAM)', 'Access management', 'Access reports', and 'Related consoles'. The main content area is titled 'Roles' and shows a list of 11 IAM roles. Each role entry includes the role name, the service it's associated with (e.g., ops.apigateway, dynamodb.application-autoscaling), the last activity date, and a 'Delete' button. Below the role list, there's a section for 'Roles Anywhere' with a 'Manage' button. At the bottom, there are sections for 'Access AWS from your non AWS workloads' (using X.509 Standard) and 'Temporary credentials'.

Role Name	Trusted Entities	Last Activity
AWSServiceRoleForAPIGateway	AWS Service: ops.apigateway (Service-Linked Role)	-
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable	AWS Service: dynamodb.application-autoscaling (Service-Linked Role)	46 days ago
AWSServiceRoleForAutoScaling	AWS Service: autoscaling (Service-Linked Role)	27 days ago
AWSServiceRoleForCloudWatchEvents	AWS Service: events (Service-Linked Role)	68 days ago
AWSServiceRoleForElasticBeanstalkManagedUpdates	AWS Service: managedupdates.elasticbeanstalk (Service-Linked Role)	-
AWSServiceRoleForElasticLoadBalancing	AWS Service: elasticloadbalancing (Service-Linked Role)	27 days ago
AWSServiceRoleForGlobalAccelerator	AWS Service: globalaccelerator (Service-Linked Role)	-
AWSServiceRoleForRDS	AWS Service: rds (Service-Linked Role)	57 minutes ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	100 days ago
AWSServiceRoleForTrustAdvisor	AWS Service: trustadvisor (Service-Linked Role)	-
AWSServiceRoleForVPCTransitGateway	AWS Service: transitgateway (Service-Linked Role)	44 days ago

Screenshot of the AWS IAM Roles page:

The page shows a list of 11 IAM roles, each with a checkbox, role name, trusted entity, last activity, and a delete button. The roles listed are:

- AWSServiceRoleForAPIGateway
- AWSServiceRoleForApplicationAutoScaling_DynamoDBTable
- AWSServiceRoleForAutoScaling
- AWSServiceRoleForCloudWatchEvents
- AWSServiceRoleForElasticBeanstalkManagedUpdates
- AWSServiceRoleForElasticLoadBalancing
- AWSServiceRoleForGlobalAccelerator
- AWSServiceRoleForRDS
- AWSServiceRoleForSupport
- AWSServiceRoleForTrustedAdvisor
- AWSServiceRoleForVPCTransitGateway

Below the table, there's a section titled "Roles Anywhere" with a "Manage" button. It also shows "X.509 Standard" and "Temporary credentials".

Screenshot of the "Create role" wizard - Step 1: Select trusted entity:

The wizard has three steps: Step 1 (Select trusted entity), Step 2 (Add permissions), and Step 3 (Name, review, and create). The current step is Step 1.

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: Allow 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.

Use case:

Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

- EC2: Allows EC2 instances to call AWS services on your behalf.
- Lambda: Allows Lambda functions to call AWS services on your behalf.

Use cases for other AWS services:
Choose a service to view use case

Buttons: Cancel, Next



Screenshot of the AWS IAM 'Create role' wizard, Step 2: Add permissions.

The page shows a list of available permissions policies:

Policy name	Type	Description
AWSDirectConnectReadOnlyAccess	AWS managed	Provides read only access to AWS Direct Connect via the AWS Management Console.
AmazonGlacierReadOnlyAccess	AWS managed	Provides read only access to Amazon Glacier via the AWS Management Console.
AWSMarketplaceFullAccess	AWS managed	Provides the ability to subscribe and unsubscribe to AWS Marketplace software, allows users to manage Marketplace software instances from the AWS Management Console.
AWSSSODirectoryAdministrator	AWS managed	Administrator access for SSO Directory.
AWSIoT1ClickReadOnlyAccess	AWS managed	Provides read only access to AWS IoT 1-Click.
AutoScalingConsoleReadOnlyAccess	AWS managed	Provides read-only access to Auto Scaling via the AWS Management Console.
AmazonDMSRedshiftS3Role	AWS managed	Provides access to manage S3 settings for Redshift endpoints for DMS.
AWSQuickSightListIAM	AWS managed	Allow QuickSight to list IAM entities.
AWSHealthFullAccess	AWS managed	Allows full access to the AWS Health APIs and Notifications and the Personal Health Dashboard.
AlexaForBusinessGatewayExecution	AWS managed	Provide gateway execution access to AlexaForBusiness services.
AmazonElasticTranscoder_ReadOnlyAccess	AWS managed	Grants users read-only access to Elastic Transcoder and list access to related services.
AmazonRDSFullAccess	AWS managed	Provides full access to Amazon RDS via the AWS Management Console.
SupportUser	AWS managed	This policy grants permissions to troubleshoot and resolve issues in an AWS account. This policy also enables the user to contact AWS support.
AmazonEC2FullAccess	AWS managed	Provides full access to Amazon EC2 via the AWS Management Console.

Below the table, there is a search bar and a pagination control (1-42).

At the bottom of the page, there is a note: "A policy defines the AWS permissions that you can assign to a user, group, or role. You can create and edit a policy in the visual editor and using JSON. Learn more".

Documentation link: [Documentation](#)

Character count: 39 of 6,144.

Buttons: Cancel, Next: Tags.

The JSON code used for the policy is as follows:

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "logs:CreateLogGroup",  
                "logs:CreateLogStream",  
                "logs:PutLogEvents"  
            ],  
            "Resource": "arn:aws:logs:*:*"  
        },  
        {  
            "Effect": "Allow",  
            "Action": [  
                "ec2:Start*",  
                "ec2:Stop*"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

Screenshot of the AWS IAM Create policy step 1: Visual editor.

The JSON code defines a policy with two statements:

```
1 "Version": "2012-10-17",
2 "Statement": [
3     {
4         "Effect": "Allow",
5         "Action": [
6             "logs:CreateLogGroup",
7             "logs:CreateLogStream",
8             "logs:PutLogEvents"
9         ],
10        "Resource": "arn:aws:logs:*:*"
11    },
12    {
13        "Effect": "Allow",
14        "Action": [
15            "ec2:Start",
16            "ec2:Stop"
17        ],
18        "Resource": "*"
19    }
20 ]
21 }
22 }
```

Character count: 235 of 6,144

Screenshot of the AWS IAM Create policy step 2: Review policy.

Name: Lambda@EC2

Description: giving ec2 permissions for lambda to do the changes

Summary

Service	Access level	Resource	Request condition
CloudWatch Logs	Limited: Write	arn:aws:logs:*	None
EC2	Limited: Write	All resources	None

Tags

Key	Value
No tags associated with the resource.	

* Required

Create policy

The policy Lambda@EC2 has been created.

Policies (1/1063) Info
A policy is an object in AWS that defines permissions.

Policy name	Type	Used as	Description
Lambda@EC2	Customer managed	None	giving ec2 permissions for lam
AWSDirectConnectReadOnlyAccess	AWS managed	None	Provides read only access to #
AmazonGlacierReadOnlyAccess	AWS managed	None	Provides read only access to #
AWSMarketplaceFullAccess	AWS managed	None	Provides the ability to subscri
ClientPNSRolePolicy	AWS managed	None	Policy to enable AWS Client V
AWSSSOAdministrator	AWS managed	None	Administrator access for SSO
AWSIoTClickReadOnlyAccess	AWS managed	None	Provides read only access to #
AutoScalingConsoleReadOnlyAccess	AWS managed	None	Provides read-only access to /
AmazonDMSRedshiftS3Role	AWS managed	None	Provides access to manage S:
AWSQuickSightListIAM	AWS managed	None	Allow QuickSight to list IAM en
AWSHealthFullAccess	AWS managed	None	Allows full access to the AWS
AlexaForBusinessGatewayExecution	AWS managed	None	Provide gateway execution ac
AmazonElasticTranscoder_ReadOnlyAccess	AWS managed	None	Grants users read-only access
AmazonRDSFullAccess	AWS managed	None	Provides full access to Amazo
SupportUser	AWS managed - job function	None	This policy grants permissions

CloudShell Feedback Language
31C Mostly sunny

IAM > Roles > Create role

Add permissions Info

Permissions policies (Selected 1/829) Info
Choose one or more policies to attach to your new role.

Policy name	Type	Description
Lambda@EC2	Custom...	giving ec2 permissions for lambda to do the changes
AWSDirectConnectReadOnlyAccess	AWS m...	Provides read only access to AWS Direct Connect via the AWS Management Console.
AmazonGlacierReadOnlyAccess	AWS m...	Provides read only access to Amazon Glacier via the AWS Management Console.
AWSMarketplaceFullAccess	AWS m...	Provides the ability to subscribe and unsubscribe to AWS Marketplace software, allows users to manage Marketplace software instances fro...
AWSSSOAdministrator	AWS m...	Administrator access for SSO Directory
AWSIoTClickReadOnlyAccess	AWS m...	Provides read only access to AWS IoT 1-Click.
AutoScalingConsoleReadOnlyAccess	AWS m...	Provides read-only access to Auto Scaling via the AWS Management Console.
AmazonDMSRedshiftS3Role	AWS m...	Provides access to manage S3 settings for Redshift endpoints for DMS.
AWSQuickSightListIAM	AWS m...	Allow QuickSight to list IAM entities
AWSHealthFullAccess	AWS m...	Allows full access to the AWS Health APIs and Notifications and the Personal Health Dashboard
AlexaForBusinessGatewayExecution	AWS m...	Provide gateway execution access to AlexaForBusiness services
AmazonElasticTranscoder_ReadOnlyAccess	AWS m...	Grants users read-only access to Elastic Transcoder and list access to related services.
AmazonRDSFullAccess	AWS m...	Provides full access to Amazon RDS via the AWS Management Console.
SupportUser	AWS m...	This policy grants permissions to troubleshoot and resolve issues in an AWS account. This policy also enables the user to contact AWS supp...

CloudShell Feedback Language
31C Mostly sunny

Screenshot of the AWS IAM Roles creation process, Step 3: Name, review, and create.

Role details

Name: LambdaOpsEC2

Description: Allows Lambda functions to call AWS services on your behalf.

Step 1: Select trusted entities

```

1 < [
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "sts:AssumeRole"
8       ],
9       "Principal": [
10        "Service": [
11          "lambda.amazonaws.com"
12        ]
13      ]
14    }
15  ]
16 ]
```

Step 2: Add permissions

Step 3: Name, review, and create

Next Step

Screenshot of the AWS IAM Role creation process:

Step 2: Add permissions

Permissions policy summary

Policy name	Type	Attached as
Lambda@EC2	Customer managed	Permissions policy

Tags

Add tags - optional Info
Tags are key-value pairs that you can add to AWS resources to help identify, organize, or search for resources.

No tags associated with the resource.

Add tag You can add up to 50 more tags.

Cancel Previous Create role

Screenshot of the IAM Roles list:

Identity and Access Management (IAM)

Role LambdaOpsEC2 created.

IAM > Roles

Roles (Selected 1/12) Info
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.

Role name	Trusted entities	Last activity
AWSServiceRoleForAPIGateway	AWS Service: ops.apigateway (Service-Linked Role)	-
AWSServiceRoleForApplicationAutoScaling_DynamoDBTable	AWS Service: dynamodb.application-autoscaling (Service-Linked Role)	46 days ago
AWSServiceRoleForAutoScaling	AWS Service: autoscaling (Service-Linked Role)	27 days ago
AWSServiceRoleForCloudWatchEvents	AWS Service: events (Service-Linked Role)	68 days ago
AWSServiceRoleForElasticBeanstalkManagedUpdates	AWS Service: managedupdates.elasticbeanstalk (Service-Linked Role)	-
AWSServiceRoleForElasticLoadBalancing	AWS Service: elasticloadbalancing (Service-Linked Role)	27 days ago
AWSServiceRoleForGlobalAccelerator	AWS Service: globalaccelerator (Service-Linked Role)	-
AWSServiceRoleForRDS	AWS Service: rds (Service-Linked Role)	1 hour ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	100 days ago
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	-
AWSServiceRoleForVPCTransitGateway	AWS Service: transitgateway (Service-Linked Role)	44 days ago
LambdaOpsEC2	AWS Service: lambda	-

Roles Anywhere Info
Authenticate your non-AWS workloads and securely provide access to AWS services.

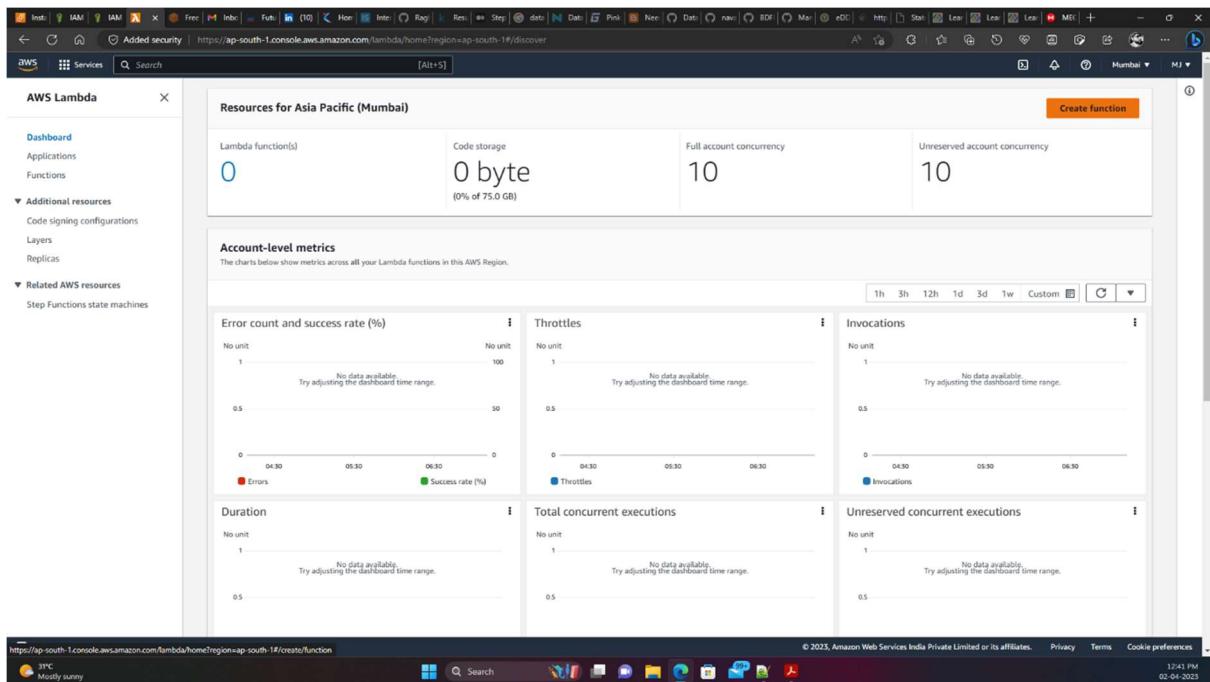
View role Create role

CloudShell Feedback Language 3°C Mostly sunny © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences 12:40 PM 02-04-2023

The screenshot shows the AWS IAM service interface. On the left, the navigation pane is open with 'Identity and Access Management (IAM)' selected. Under 'Access management', 'Roles' is expanded, showing 'LambdaOpsEC2'. The main content area displays the details for the 'LambdaOpsEC2' role. The 'Permissions' tab is active, showing one managed policy named 'Lambda@EC2'. This policy is customer-managed and grants EC2 permissions to Lambda. Below the policy list, there is a section for 'Permissions boundary - (not set)' which includes a 'Set permissions boundary' button.

After the creation of instance, role, and policy we will create 2 Lambda functions to start and stop the instance.

This screenshot shows the AWS Lambda service interface. A search bar at the top left contains the query 'lambda'. The left sidebar has a search bar and lists services like Features, Resources, and Documentation. The main content area shows search results for 'lambda', with 'Lambda' being the top result. To the right, the 'LambdaOpsEC2' role's permissions page is displayed, showing the previously created policy. The 'Description' field of the policy is visible, stating 'giving ec2 permissions for lambda to do the changes'.



Creation of StopFunc in Lambda and select the role which we had created earlier to give access to control and manage EC2 instance.

Screenshot of the AWS Lambda function creation interface.

Basic Information

- Function name:** StopFunc
- Runtime info:** Python 3.9
- Architecture:** x86_64
- Permissions:** LambdaOpEC2

Advanced settings

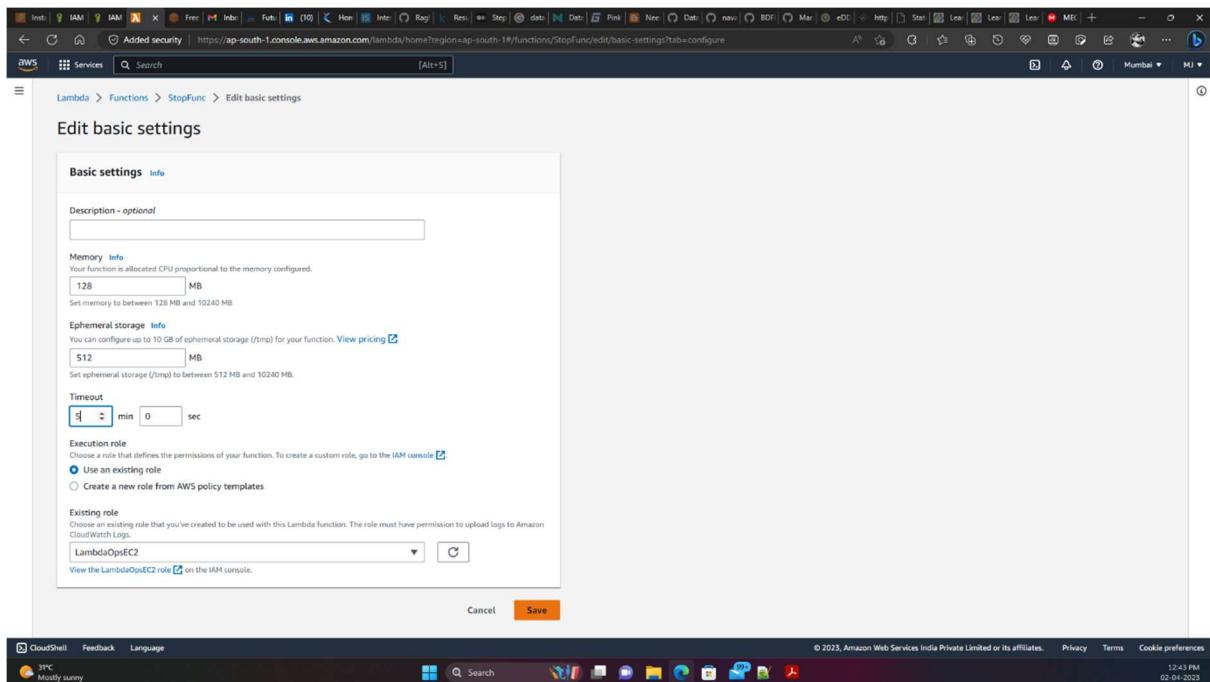
Function overview

- Name:** StopFunc
- Description:** -
- Last modified:** 42 seconds ago
- Function ARN:** arn:aws:lambda:ap-south-1:846780726677:function:StopFunc
- Function URL:** Info

Configuration

General configuration

Description	Memory	Ephemeral storage
-	128 MB	512 MB
Timeout	0 min 3 sec	



The python code used for the stopping instance is as follows:

```
import boto3

region = 'ap-south-1'

instances = ['i-076b4f653a0db14c1']

ec2 = boto3.client('ec2', region_name=region)
```

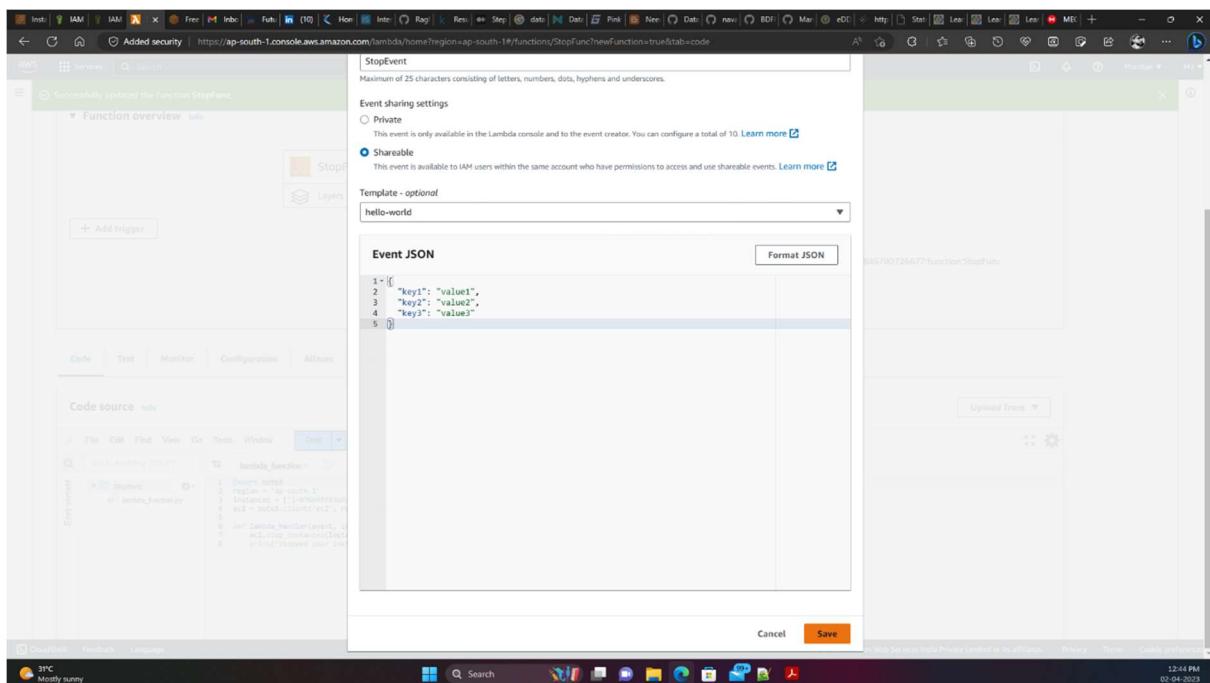
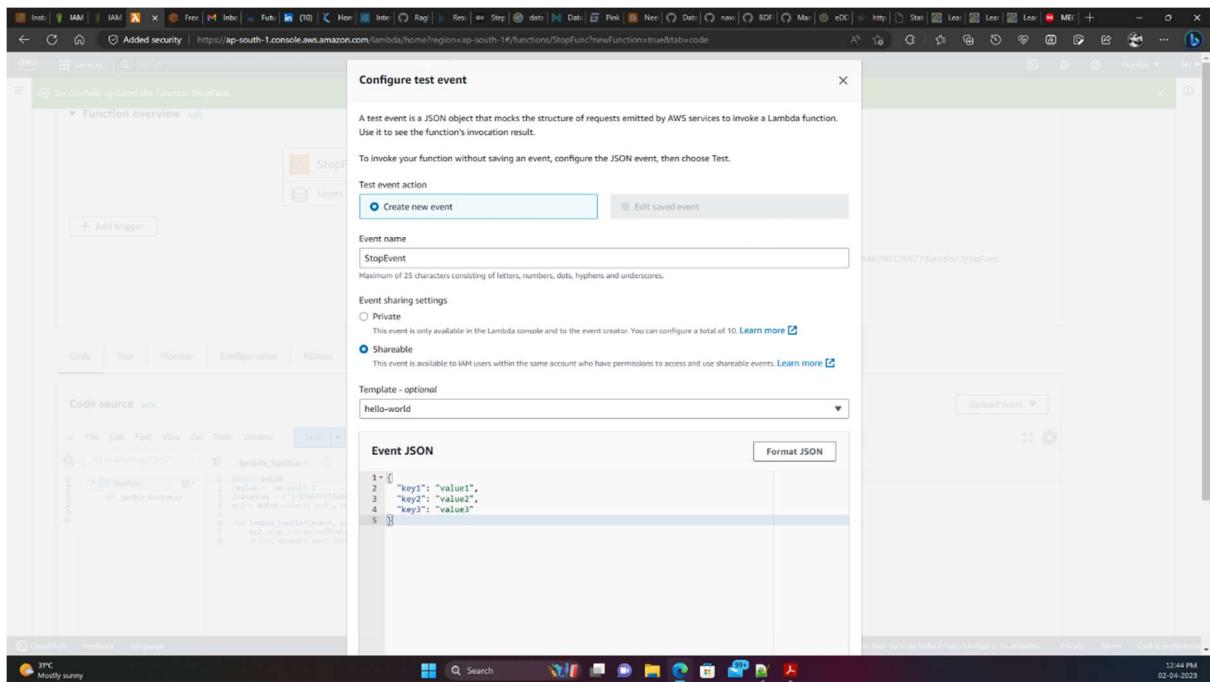
```
def lambda_handler(event, context):
    ec2.stop_instances(InstanceIds=instances)
    print('stopped your instances: ' + str(instances))
```

To test the code, we need to first deploy and create an event then we can test the code and see the result.

Screenshot of the AWS Lambda Function Overview page for 'StopFunc'. The function was successfully updated 15 seconds ago. It has no triggers or destinations. The ARN is arn:aws:lambda:ap-south-1:846780726677:function:StopFunc.

Code source

```
lambda_function
1 import boto3
2 region = 'ap-south-1'
3 instances = ['i-07604f653eb0b14c1']
4 ec2 = boto3.client('ec2', region_name=region)
5
6 def lambda_handler(event, context):
7     ec2.stop_instances(InstanceIds=instances)
8     print('stopped your instances: ' + str(instances))
```



The screenshot shows the AWS Lambda Function Overview page for the 'StopFunc' function. The function has no triggers or destinations. It was last modified 38 seconds ago. The ARN is arn:aws:lambda:ap-south-1:846780726677:function:StopFunc. The URL is available at [Function URL](#). The code source editor shows the following Python code:

```
lambda_function.py
1 import boto3
2 region = 'ap-south-1'
3 instances = ['i-07604f653ab0b14c1']
4 ec2 = boto3.client('ec2', region_name=region)
5
6 def lambda_handler(event, context):
7     ec2.stop_instances(InstanceIds = str(instances))
8     print(f'stopped your instances - {str(instances)}')
```

Here we can see the status of the instance is stopped.

The screenshot shows the AWS Lambda Function Overview page for the 'StopFunc' function. The function has no triggers or destinations. It was last modified 1 minute ago. The ARN is arn:aws:lambda:ap-south-1:846780726677:function:StopFunc. The URL is available at [Function URL](#). The code source editor shows the execution result of the 'StopEvent' test event. The response is null. The function logs show the following:

```
START RequestId: fc1601c5-3327-47d5-8002-57f134b2f8bc Version: $LATEST
STOP RequestId: fc1601c5-3327-47d5-8002-57f134b2f8bc
END RequestId: fc1601c5-3327-47d5-8002-57f134b2f8bc
REPORT RequestId: fc1601c5-3327-47d5-8002-57f134b2f8bc Duration: 512.61 ms Billed Duration: 513 ms Memory Size: 128 MB Max Memory Used: 78 MB Init Duration: 366.23 ms
Request ID: fc1601c5-3327-47d5-8002-57f134b2f80c
```

Screenshot of the AWS EC2 Instances page showing a single instance named "test server".

Instances (1/1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
test server	i-076b4f653a0db14c1	Stopping	t2.micro	-	No alarms	ap-south-1a	ec2-65-2-144-216.ap-s...	65.2.144.216	-

Instance: i-076b4f653a0db14c1 (test server)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance summary

Instance ID	i-076b4f653a0db14c1 (test server)	Public IPv4 address	Private IPv4 addresses
IPv6 address	-	Instance state	172.31.38.128
Hostname type	IP name: ip-172-31-38-128.ap-south-1.compute.internal	Private IP DNS name (IPv4 only)	Public IPv4 DNS
Answer private resource DNS name	-	ip-172-31-38-128.ap-south-1.compute.internal	-
Auto-assigned IP address	-	Instance type	t2.micro
VPC ID	vpc-0af44d8154f5cb36d	Elastic IP addresses	-

AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations.

Screenshot of the AWS EC2 Instances page showing a single instance named "test server".

Instances (1/1) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
test server	i-076b4f653a0db14c1	Stopped	t2.micro	-	No alarms	ap-south-1a	-	65.2.144.216	-

Instance: i-076b4f653a0db14c1 (test server)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance summary

Instance ID	i-076b4f653a0db14c1 (test server)	Public IPv4 address	Private IPv4 addresses
IPv6 address	-	Instance state	172.31.38.128
Hostname type	IP name: ip-172-31-38-128.ap-south-1.compute.internal	Private IP DNS name (IPv4 only)	Public IPv4 DNS
Answer private resource DNS name	-	ip-172-31-38-128.ap-south-1.compute.internal	-
Auto-assigned IP address	-	Instance type	t2.micro
VPC ID	vpc-0af44d8154f5cb36d	Elastic IP addresses	-

AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations.

Now we will be creating another function StartFunc which is used to start the stopped instance.

The screenshot shows the AWS Lambda Functions page. On the left, the navigation menu includes 'Dashboard', 'Applications', 'Functions', 'Additional resources', 'Code signing configurations', 'Layers', 'Replicas', and 'Related AWS resources'. The main area displays a table titled 'Functions (1)'. The table has columns for 'Function name', 'Description', 'Package type', 'Runtime', and 'Last modified'. A single row is shown for 'StopFunc', which was last modified 2 minutes ago. At the top right of the table, there are 'Actions' and a 'Create function' button.

Create function

AWS Serverless Application Repository applications have moved to [Create application](#).

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: StartFunc

Runtime: Python 3.9

Architecture: x86_64

Permissions: 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

Screenshot of the AWS Lambda function creation interface (Create function) in the AWS Management Console.

Basic Information

- Function name:** StartFunc
- Runtime info:** Python 3.9
- Architecture info:** x86_64
- Permissions info:** LambdaOpcEC2 role

Execution role: Use an existing role (LambdaOpcEC2)

Existing role: LambdaOpcEC2

Advanced settings: None

Create function

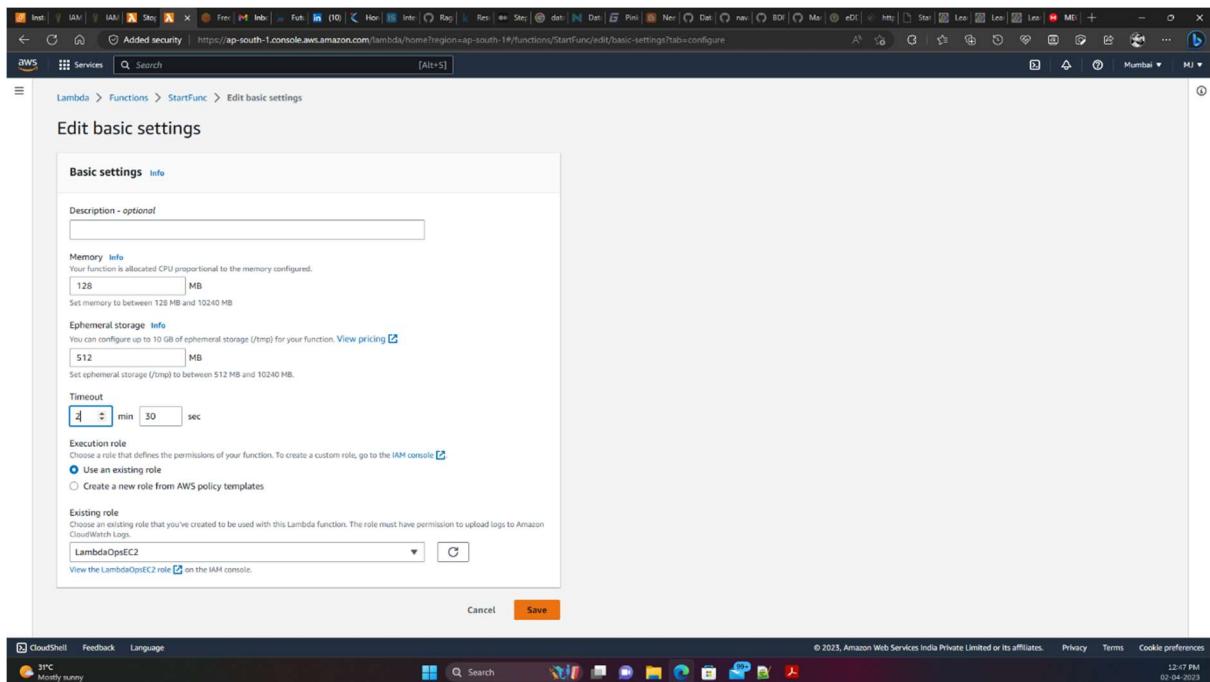
Screenshot of the AWS Lambda function configuration interface (Configuration tab) for the newly created function StartFunc.

Function overview

- Name:** StartFunc
- Description:** -
- Last modified:** 33 seconds ago
- Function ARN:** arn:aws:lambda:ap-south-1:846780726677:function:StartFunc
- Function URL:** Info

General configuration

Setting	Value
Memory	128 MB
Timeout	0 min 3 sec
Ephemeral storage	512 MB



Python code used in StartFunc is as follows:

```
import boto3

region = 'ap-south-1'

instances = ['i-076b4f653a0db14c1']

ec2 = boto3.client('ec2', region_name=region)
```

```
def lambda_handler(event, context):
    ec2.start_instances(InstanceIds=instances)
    print('started your instances: ' + str(instances))
```

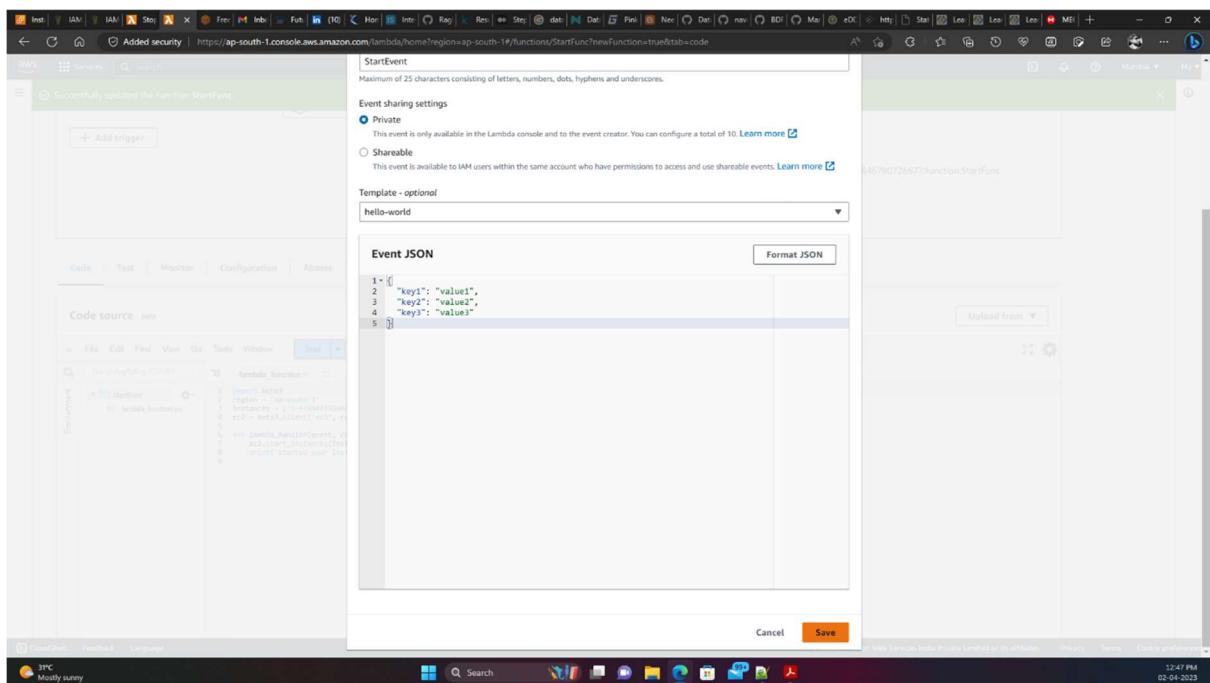
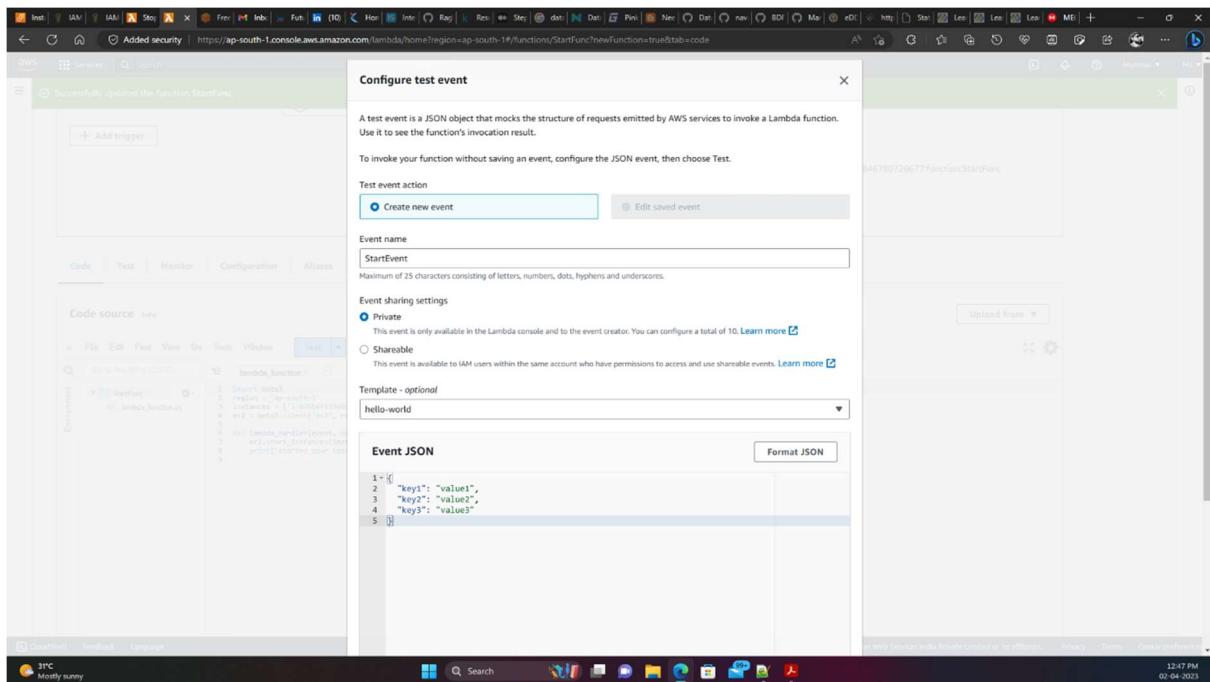
To test the code, we need to first deploy and create an event then we can test the code and see the result.

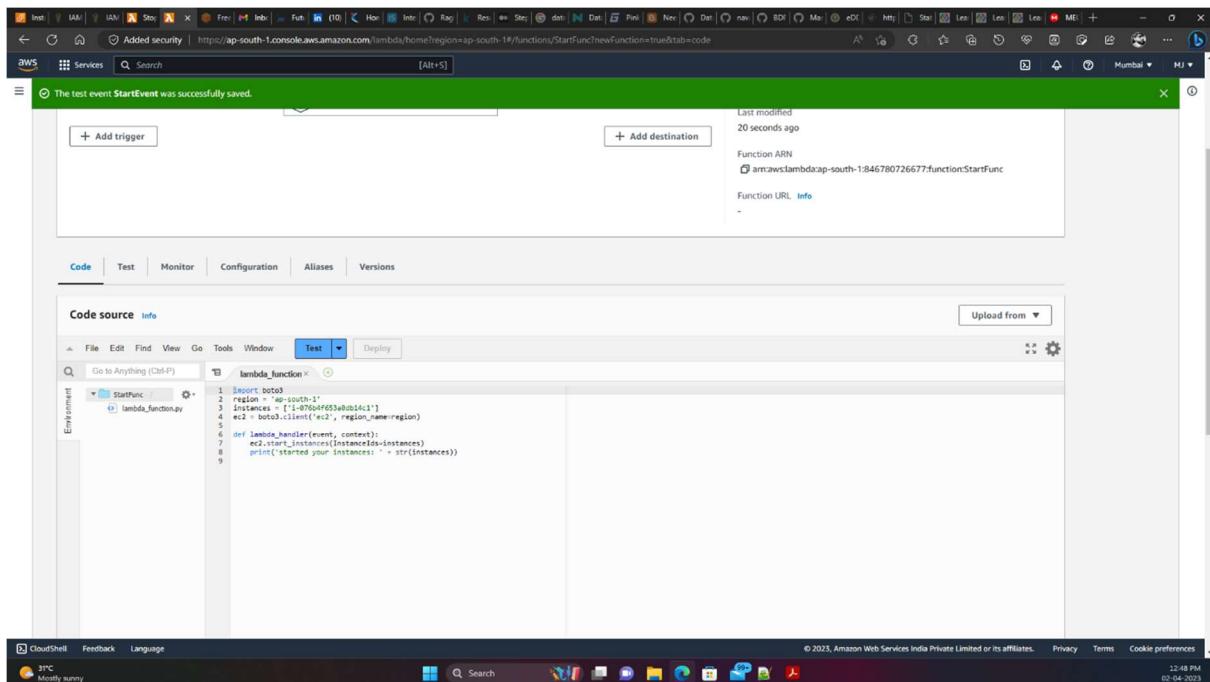
The screenshot shows the AWS Lambda console interface. A success message at the top states "Successfully updated the function StartFunc." Below this, the function details are displayed: Last modified 2 seconds ago, Function ARN arn:aws:lambda:ap-south-1:846780726677:function:StartFunc, and Function URL Info.

The "Code source" tab is selected, showing the code editor with the following Python script:

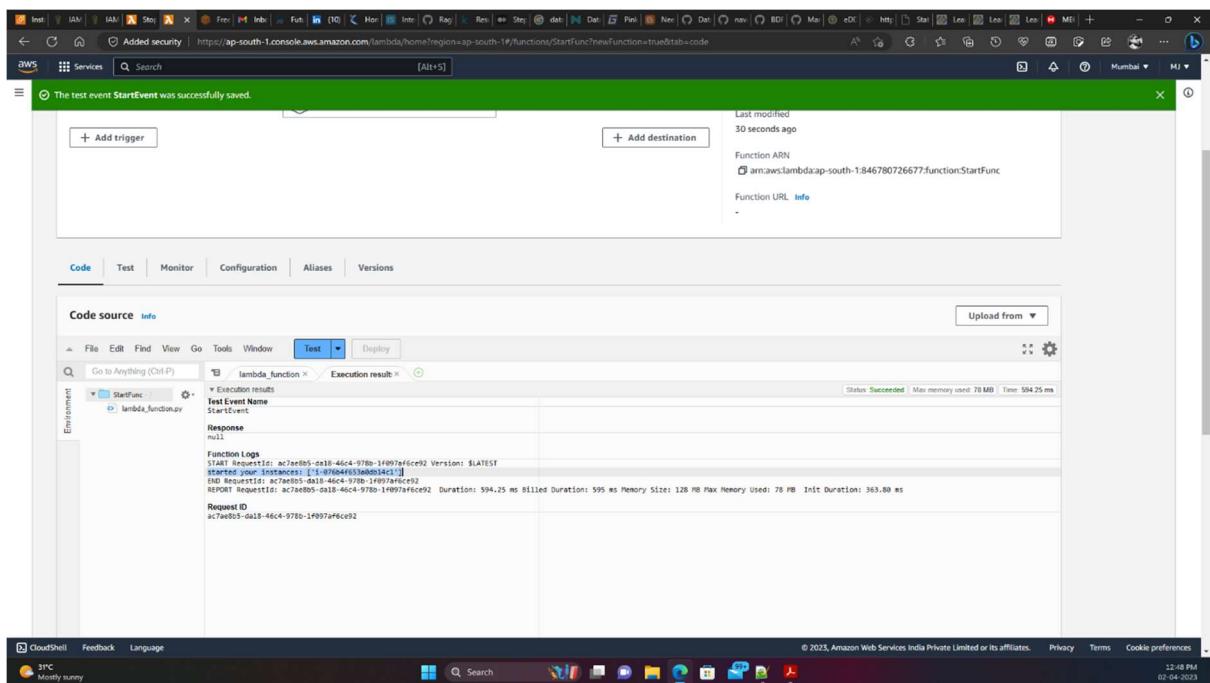
```
1 import boto3
2 region = 'ap-south-1'
3 ec2 = boto3.client('ec2', region_name=region)
4 ec2 = boto3.client('ec2', region_name=region)
5
6 def lambda_handler(event, context):
7     ec2.start_instances(Instances=instances)
8     print("started your instances: " + str(instances))
9
```

The browser status bar at the bottom indicates "CloudShell Feedback Language" and the date "02-04-2023".





Here we can see the instance has been started using lambda function.



What we have done to Start and Stop the functions using the lambda function until now is the manual procedure.

Our aim is to make the process automated.

So, to do that we use triggers in the lambda function. These procedures can be seen in the attached snapshots below.

Here we are using EventBridge to make the process completely automated.

The test event StopEvent was successfully saved.

Lambda > Functions > StopFunc

StopFunc

Function overview [Info](#)

 StopFunc
Layers (0)

+ Add trigger + Add destination

Description
Last modified 6 minutes ago
Function ARN arn:aws:lambda:ap-south-1:846780726677:function:StopFunc
Function URL [Info](#)

Code | Test | Monitor | Configuration | Aliases | Versions

Code source [Info](#)

File Edit Find View Go Tools Window **Test** Deploy

Go to Anything (Ctrl P) lambda_function Execution result

Execution results
Test Event Name StopEvent
Response null

Function Logs

https://ap-south-1.console.aws.amazon.com/lambda/home?region=ap-south-1#functions/StopFunc?newFunction=true&tab=code

3°C Mostly sunny

Added security | https://ap-south-1.console.aws.amazon.com/lambda/home?region=ap-south-1#functions/StopFunc?newFunction=true&tab=code

Services | Search [Alt+S]

Lambda > Add trigger

Add trigger

Trigger configuration [Info](#)

EventBridge (CloudWatch Events)
aws events management-tools

Rule
Pick an existing rule, or create a new one:
 Create a new rule
 Existing rules

Rule name
Enter a name to uniquely identify your rule.
LambdaStopEC2

Rule description
Provide an optional description for your rule.
eventbridge automatically stops the running ec2 instances

Rule type
Trigger your target based on an event pattern, or based on an automated schedule.
 Event pattern
 Schedule expression

Schedule expression
Self-trigger your target on an automated schedule using [Cron or rate expressions](#). Cron expressions are in UTC.
rate(7 minutes)

e.g. rate(1 day), cron(0 17 ? * MON-FRI)

Lambda will add the necessary permissions for Amazon EventBridge (CloudWatch Events) to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

Cancel **Add**

CloudShell Feedback Language

3°C Mostly sunny

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

12:50 PM 02-04-2023

Screenshot of the AWS Lambda console showing the configuration for the function "StopFunc".

The "Function overview" section shows the function name "StopFunc", a trigger named "EventBridge (CloudWatch Events)", and a status message indicating the trigger was successfully added.

The "Configuration" tab is selected, showing the "Triggers" section which lists one trigger named "EventBridge (CloudWatch Events): LambdaStopEC2". The rule state is set to "ENABLED".

Below the configuration tabs, the CloudShell, Feedback, and Language buttons are visible.

Instances (1/1) Info

The Instances page shows a single instance named "test server" with the ID "i-076b4f653a0db14c1". The instance is currently stopped. The details pane for this instance is open, showing the following information:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP
test server	i-076b4f653a0db14c1	Stopped	t2.micro	-	No alarms	ap-south-1a	-	-	-

The "Details" tab is selected in the instance details pane. The public IPv4 address listed is "Private IPv4 addresses".

The test event StartEvent was successfully saved.

Lambda > Functions > StartFunc

StartFunc

Description

Last modified 6 minutes ago

Function ARN arn:aws:lambda:ap-south-1:846780726677:function:StartFunc

Function URL Info

Code | Test | Monitor | Configuration | Aliases | Versions

Code source Info

File Edit Find View Go Tools Window Test Deploy

Execution result

Test Event Name StartEvent Response null Function Logs

https://ap-south-1.console.aws.amazon.com/lambda/home?region=ap-south-1#function:StartFunc

37°C Mostly sunny

Lambda > Add trigger

Add trigger

Trigger configuration Info

EventBridge (CloudWatch Events)

Rule

Create a new rule

Existing rules

Rule name

Enter a name to uniquely identify your rule. LambdaStartEC2

Rule description

Provide an optional description for your rule. lambda function automatically starts the ec2 instance

Rule type

Trigger your target based on an event pattern, or based on an automated schedule.

Event pattern

Schedule expression

Schedule expression Self-trigger your target on an automated schedule using [Cron or rate expressions](#). Cron expressions are in UTC.

rate(7 minutes)

e.g. rate(1 day), cron(0 17 ? * MON-FRI *)

Lambda will add the necessary permissions for Amazon EventBridge (CloudWatch Events) to invoke your Lambda function from this trigger. [Learn more](#) about the Lambda permissions model.

Cancel Add

CloudShell Feedback Language

37°C Mostly sunny

Screenshot of the AWS Lambda console showing the configuration of a function named "StartFunc".

The "Triggers" section shows one trigger named "EventBridge (CloudWatch Events)" which has been successfully added to the function. The trigger's ARN is arn:aws:lambda:ap-south-1:1846780726677:function:StartFunc.

The "Configuration" tab is selected, showing the function's ARN and URL.

Screenshot of the AWS EC2 console showing the instances page. One instance named "test server" (ID: i-076b4f653a0db14c1) is listed, currently in a "Pending" state.

The instance details page for "test server" shows the following information:

Details	Security	Networking	Storage	Status checks	Monitoring	Tags
Instance ID: i-076b4f653a0db14c1						
Public IPv4 address: 172.31.17.111						
Private IPv4 addresses: 172.31.17.111						

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Screenshot of the AWS CloudWatch Metrics console showing a log stream for the instance i-076b4f653a0db14c1. The log stream contains several entries related to the test server's configuration and startup.

```
2023-02-04T12:35:23+00:00 i-076b4f653a0db14c1 /var/log/cloud-init.log [INFO] 2023-02-04 12:35:23.000000000 UTC: cloud-init: version 23.0.1-1~23.0.1-0ubuntu1.23.0.1-0ubuntu1.1 running on i-076b4f653a0db14c1 (ap-south-1)
```

Attribute	Value
Name	test server
Instance ID	i-076b4f653a0db14c1
Instance state	Running
Instance type	t2.micro
Status check	-
Alarm status	No alarms
Availability Zone	ap-south-1a
Public IPv4 DNS	ec2-15-235-238-66.ap...
Public IPv4 IP	13.235.238.66
Elastic IP	-

The AWS CloudWatch Metrics interface shows the following metrics for the instance:

Metric	Value
aws.ec2.instance.state	running
aws.ec2.instance.type	t2.micro
aws.ec2.instance.id	i-076b4f653a0db14c1
aws.ec2.availability.zone	ap-south-1a
aws.ec2.public.ip.v4	13.235.238.66
aws.ec2.private.ip.v4	172.17.0.2