Roll No: 20BCE204

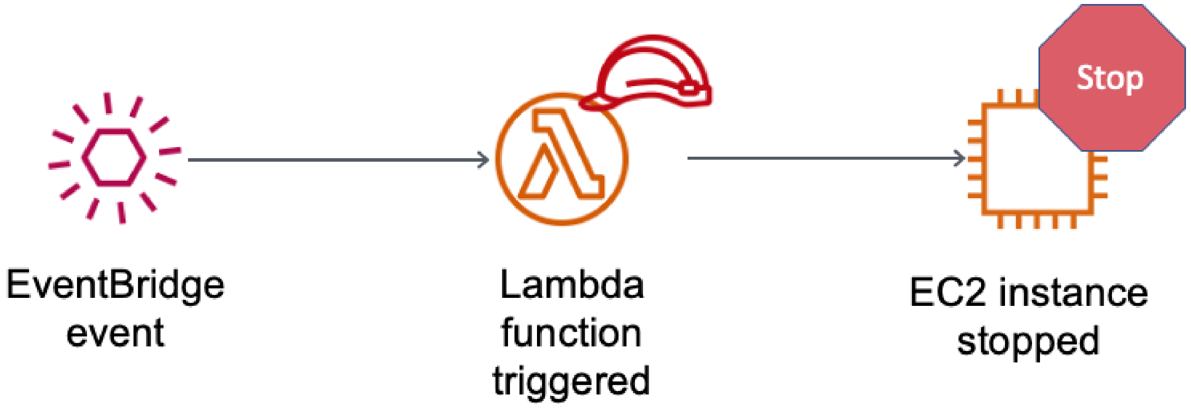
Course: 2CSDE67 Cloud Computing

Practical No: 4

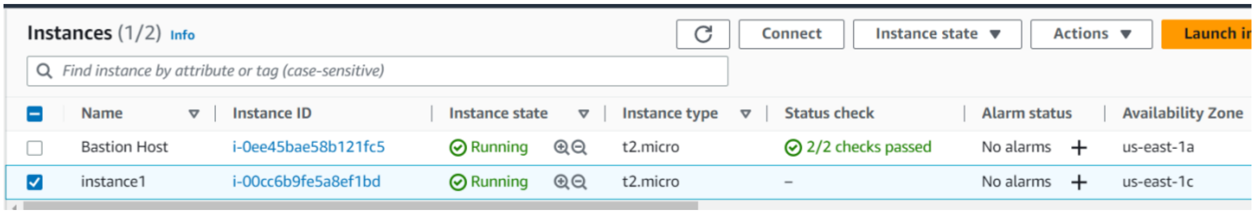
Aim: Working with an IaaS Cloud Computing: Using AWS (Amazon Web Services) to understating the following concept. Do load balancing in amazon EC2

Introduction:

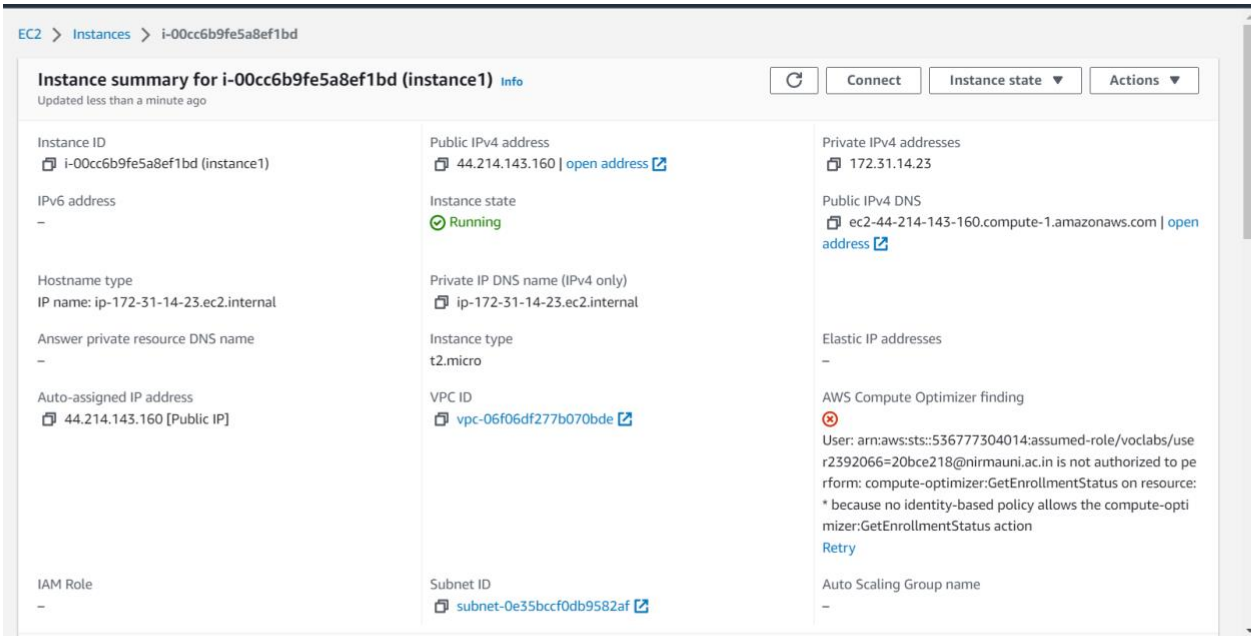
AWS Lambda is an event-driven, serverless computing platform provided by Amazon as a part of Amazon Web Services. It is a computing service that runs code in response to events and automatically manages the computing resources required by that code.



Already running instances of EC2



☐	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
☐	Bastion Host	i-0ee45bae58b121fc5	✓ Running	t2.micro	✓ 2/2 checks passed	No alarms +	us-east-1a
☑	instance1	i-00cc6b9fe5a8ef1bd	✓ Running	t2.micro	–	No alarms +	us-east-1c



Instance summary for i-00cc6b9fe5a8ef1bd (instance1)

Updated less than a minute ago

Instance ID
i-00cc6b9fe5a8ef1bd (instance1)

Public IPv4 address
44.214.143.160 | open address

Private IPv4 addresses
172.31.14.23

IPv6 address
–

Instance state
Running

Public IPv4 DNS
ec2-44-214-143-160.compute-1.amazonaws.com | open address

Hostname type
IP name: ip-172-31-14-23.ec2.internal

Private IP DNS name (IPv4 only)
ip-172-31-14-23.ec2.internal

Answer private resource DNS name
–

Instance type
t2.micro

Elastic IP addresses
–

Auto-assigned IP address
44.214.143.160 [Public IP]

VPC ID
vpc-06f06df277b070bde

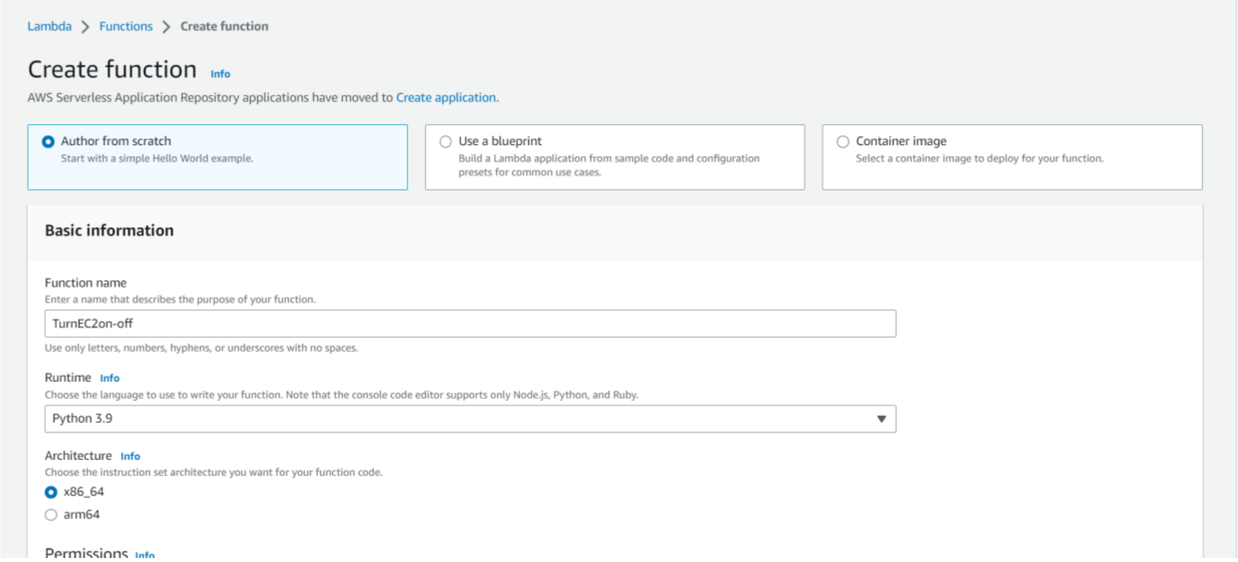
AWS Compute Optimizer finding
User: arn:aws:sts::536777304014:assumed-role/voclabs/user2392066=20bce218@nirmauni.ac.in is not authorized to perform: compute-optimizer:GetEnrollmentStatus on resource: * because no identity-based policy allows the compute-optimizer:GetEnrollmentStatus action
Retry

IAM Role
–

Subnet ID
subnet-0e35bccf0db9582af

Auto Scaling Group name
–

Creating a Lambda function in AWS console



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

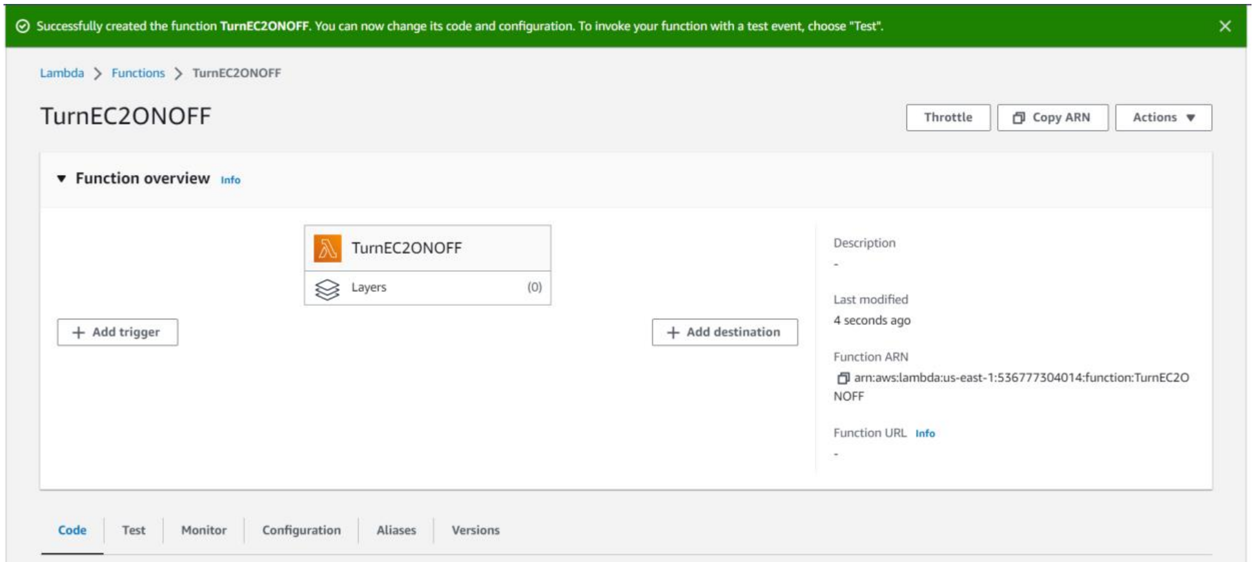
Function name
Enter a name that describes the purpose of your function.
TurnEC2on-off
Use only letters, numbers, hyphens, or underscores with no spaces.

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 3.9 ▼

Architecture Info
Choose the instruction set architecture you want for your function code.
● x86_64
○ arm64

Permissions Info

Lambda function dashboard:



Successfully created the function **TurnEC2ONOFF**. You can now change its code and configuration. To invoke your function with a test event, choose "Test".

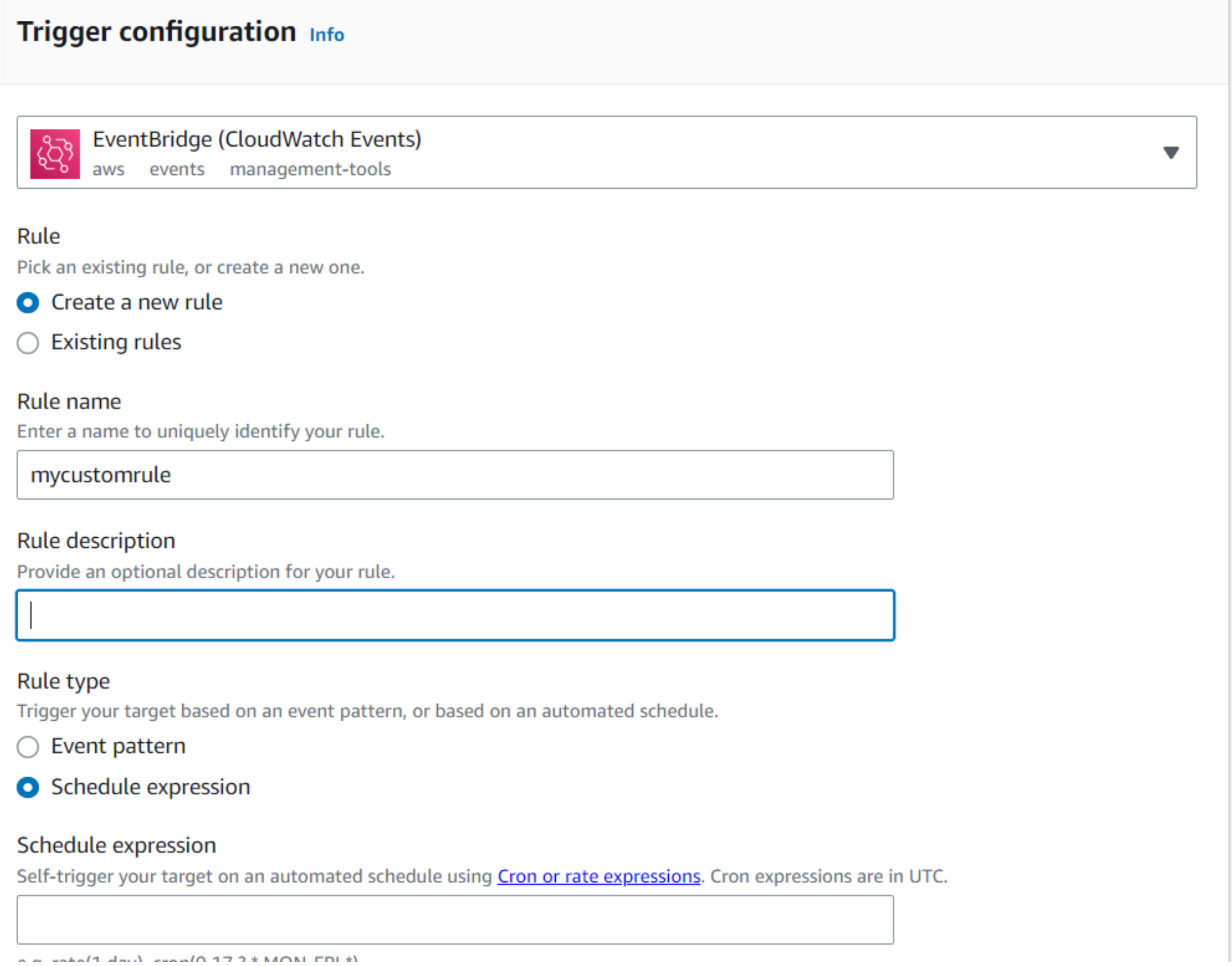
Description
-

Last modified
4 seconds ago

Function ARN
arn:aws:lambda:us-east-1:536777304014:function:TurnEC2ONOFF

Function URL Info
-

Creating a trigger for the function to run:



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.

mycustomrule

Rule description
Provide an optional description for your rule.

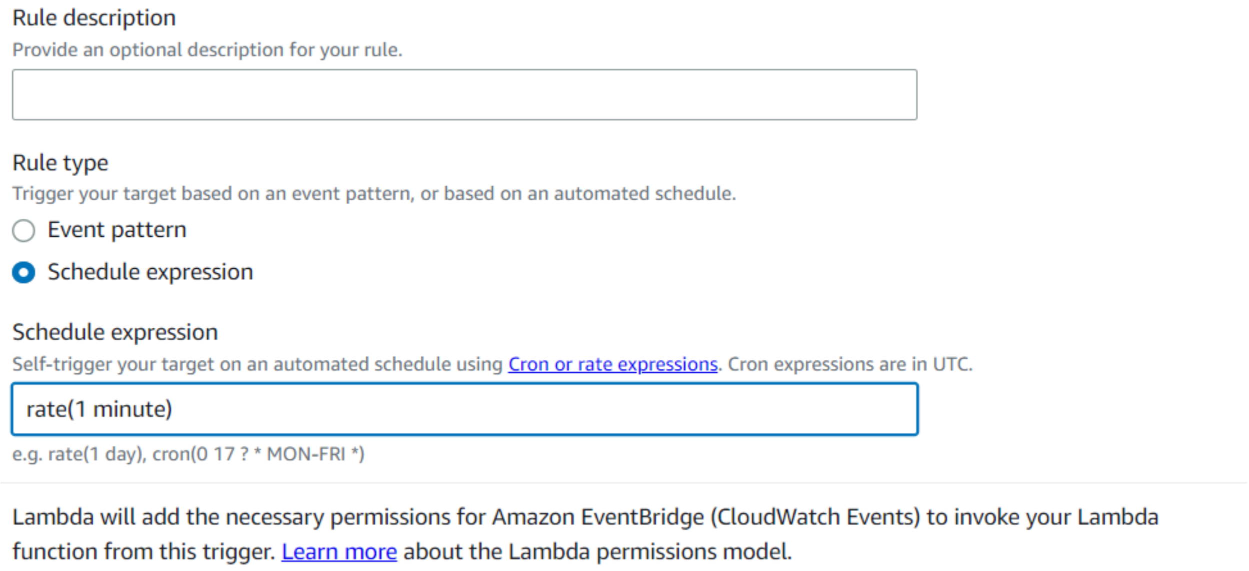
Rule type
Trigger your target based on an event pattern, or based on an automated schedule.

○ Event pattern
● Schedule expression

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

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

Providing the rate of execution



Rule description
Provide an optional description for your rule.

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

○ Event pattern
● Schedule expression

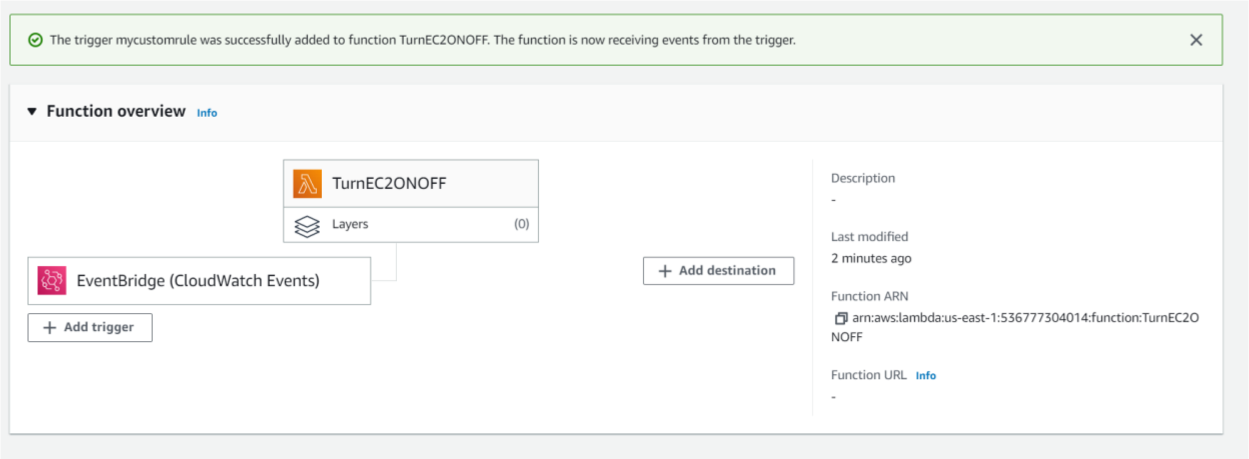
Schedule expression
Self-trigger your target on an automated schedule using Cron or rate expressions. Cron expressions are in UTC.

rate(1 minute)

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.

Configured function



The trigger mycustomrule was successfully added to function TurnEC2ONOFF. The function is now receiving events from the trigger.

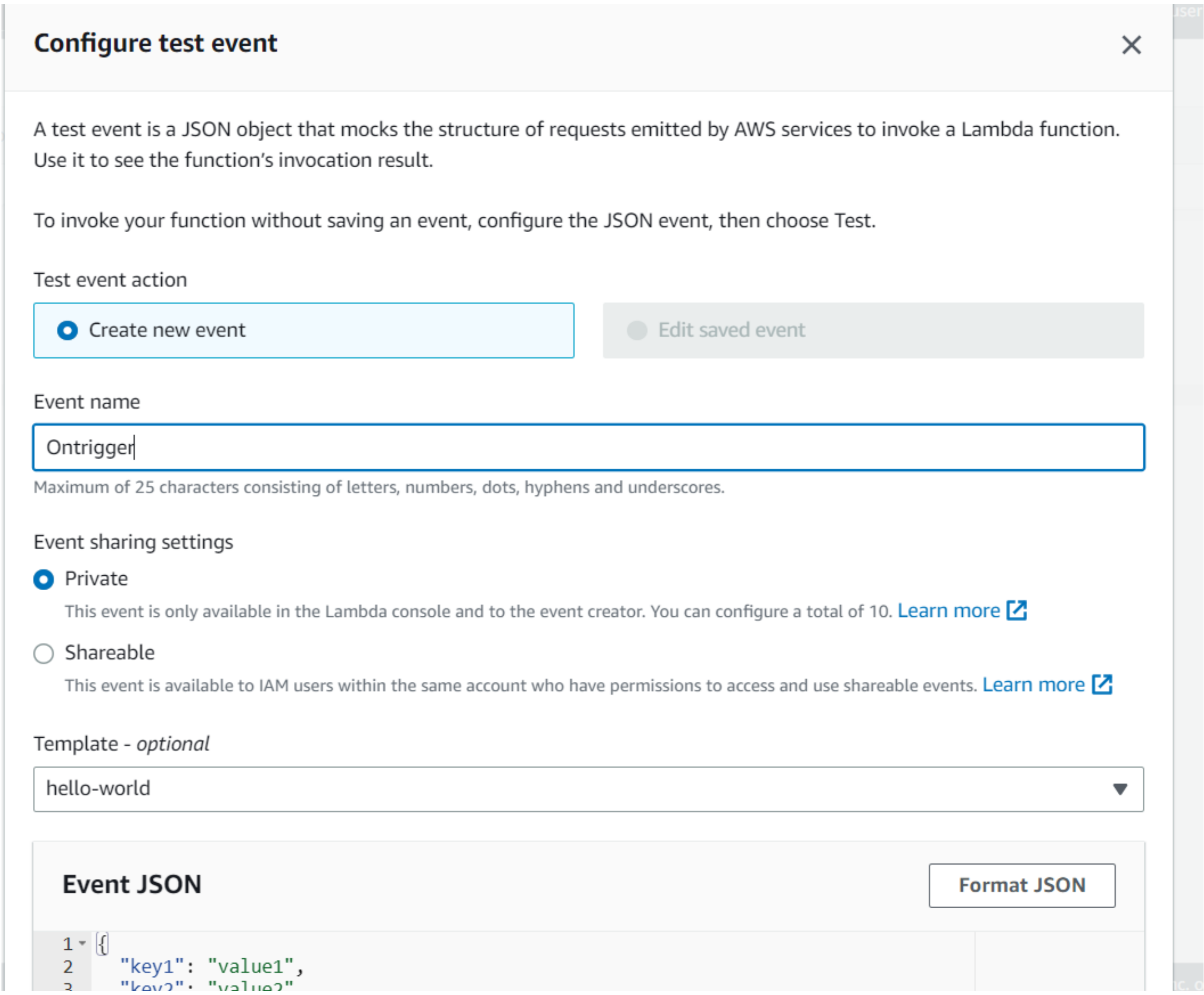
Description
-

Last modified
2 minutes ago

Function ARN
arn:aws:lambda:us-east-1:536777304014:function:TurnEC2ONOFF

Function URL Info
-

Creating a custom test event to test the function:



Configure test event ✕

A test event is a JSON object that mocks the structure of requests emitted by AWS services to invoke a Lambda function. Use it to see the function’s invocation result.

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

Ontrigger

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

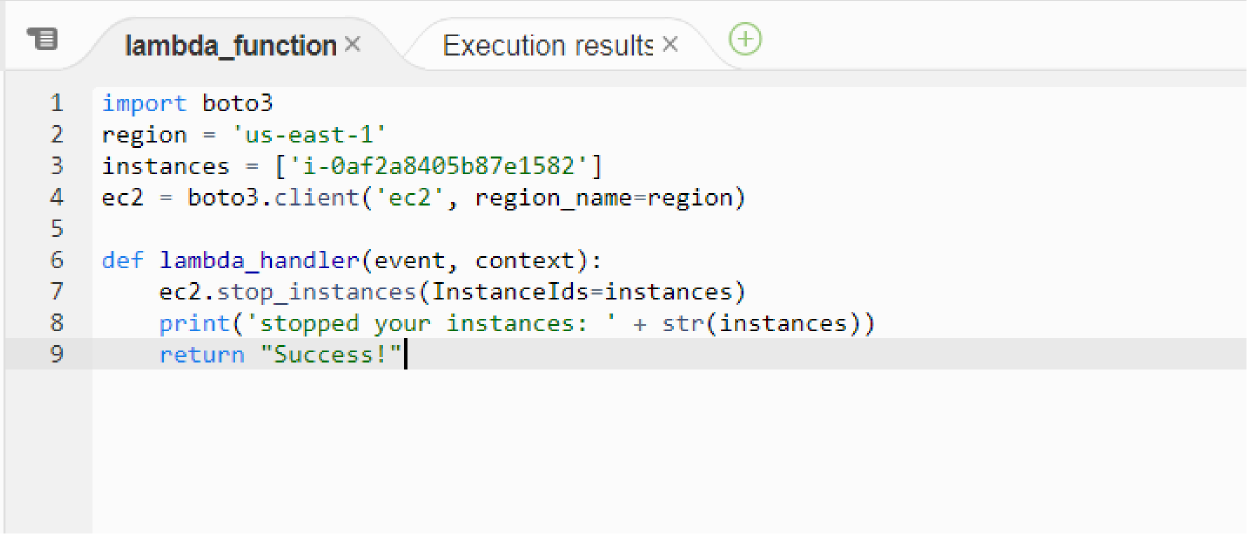
Template - *optional*

hello-world ▼

Event JSON Format JSON

```
{
  "key1": "value1",
  "key2": "value2"
```

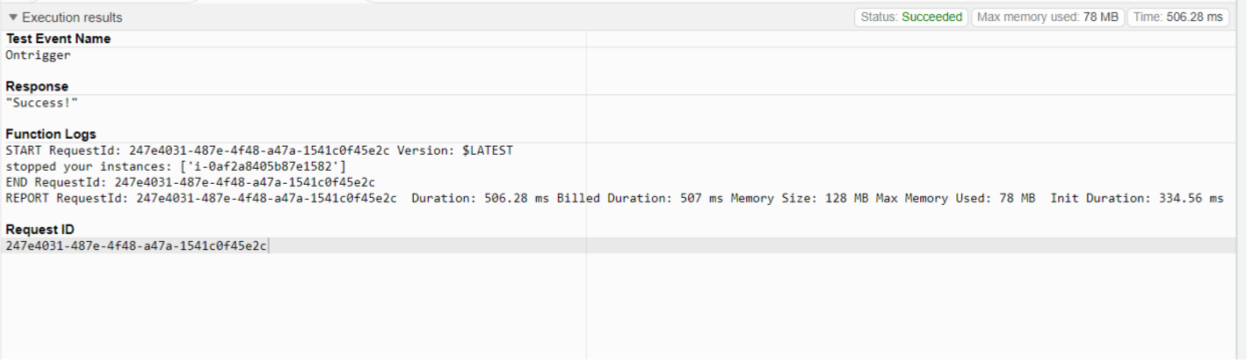
Function body:



```python
import boto3
region = 'us-east-1'
instances = ['i-0af2a8405b87e1582']
ec2 = boto3.client('ec2', region_name=region)

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

Execution result of the function:



Test Event Name

```
Ontrigger
```

Response

```
"Success!"
```

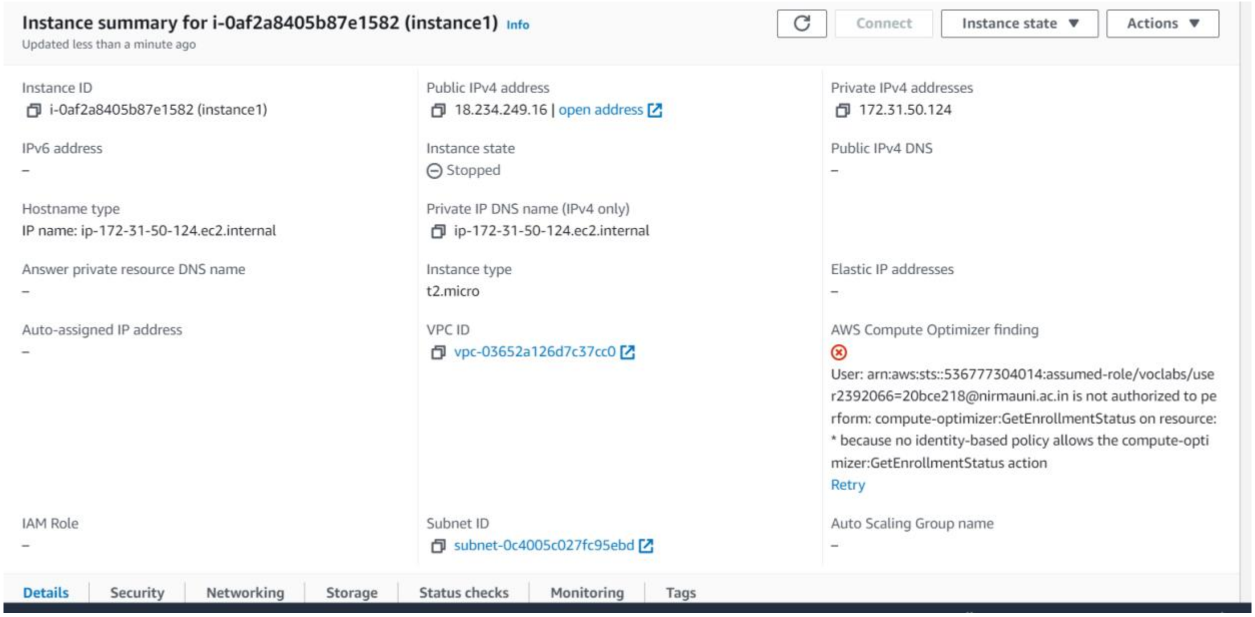
Function Logs

```
START RequestId: 247e4031-487e-4f48-a47a-1541c0f45e2c Version: $LATEST
stopped your instances: ['i-0af2a8405b87e1582']
END RequestId: 247e4031-487e-4f48-a47a-1541c0f45e2c
REPORT RequestId: 247e4031-487e-4f48-a47a-1541c0f45e2c  Duration: 506.28 ms Billed Duration: 507 ms Memory Size: 128 MB Max Memory Used: 78 MB  Init Duration: 334.56 ms
```

Request ID

```
247e4031-487e-4f48-a47a-1541c0f45e2c
```

The VM instance is stopped by executing the Function



Instance summary for i-0af2a8405b87e1582 (instance1) Info

Updated less than a minute ago

Instance ID
i-0af2a8405b87e1582 (instance1)

Public IPv4 address
18.234.249.16 | open address

Private IPv4 addresses
172.31.50.124

IPv6 address
–

Instance state
Stopped

Public IPv4 DNS
–

Hostname type
IP name: ip-172-31-50-124.ec2.internal

Private IP DNS name (IPv4 only)
ip-172-31-50-124.ec2.internal

Answer private resource DNS name
–

Instance type
t2.micro

Elastic IP addresses
–

Auto-assigned IP address
–

VPC ID
vpc-03652a126d7c37cc0

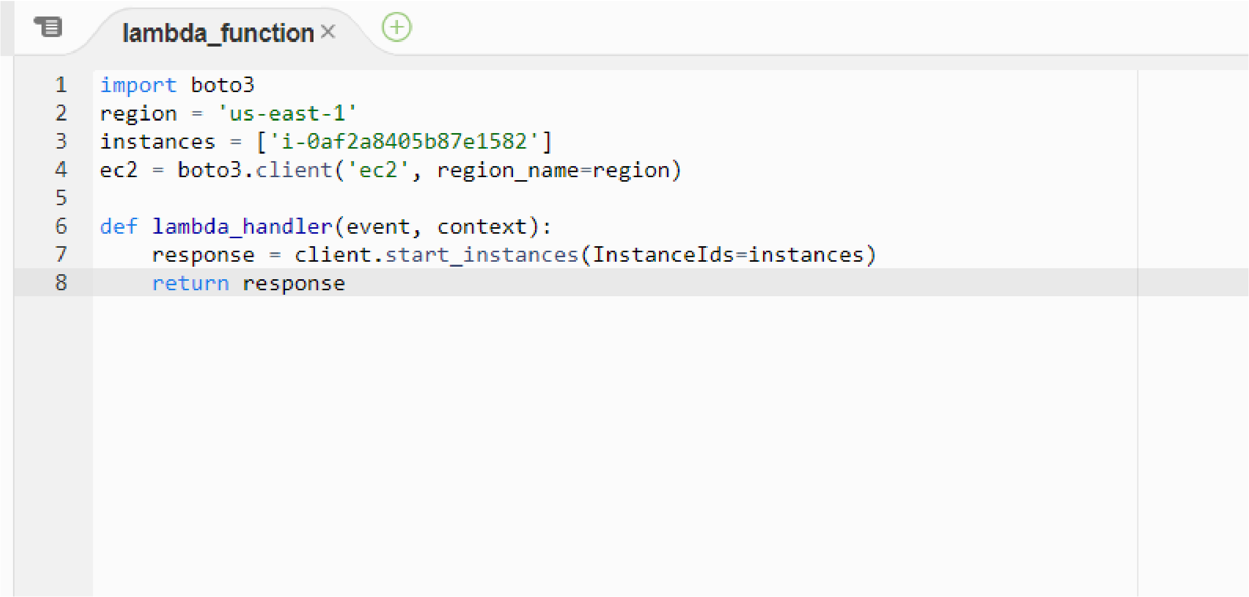
AWS Compute Optimizer finding
User: arn:aws:sts::536777304014:assumed-role/voclabs/user2392066=20bce218@nirmauni.ac.in is not authorized to perform: compute-optimizer:GetEnrollmentStatus on resource: * because no identity-based policy allows the compute-optimizer:GetEnrollmentStatus action
Retry

IAM Role
–

Subnet ID
subnet-0c4005c027fc95ebd

Auto Scaling Group name
–

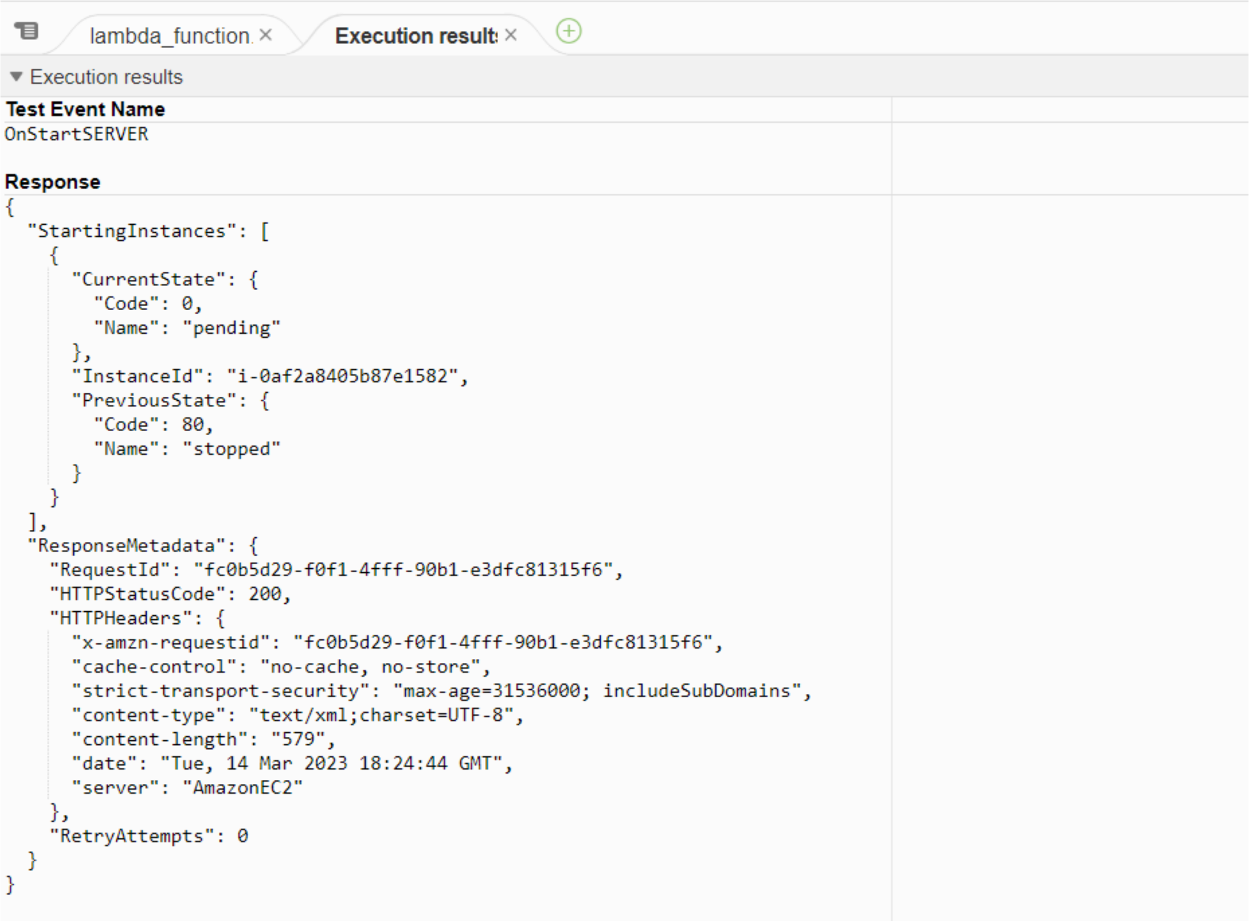
Function body to turn the instance on:



```python
import boto3
region = 'us-east-1'
instances = ['i-0af2a8405b87e1582']
ec2 = boto3.client('ec2', region_name=region)

def lambda_handler(event, context):
    response = client.start_instances(InstanceIds=instances)
    return response
```

Execution result:



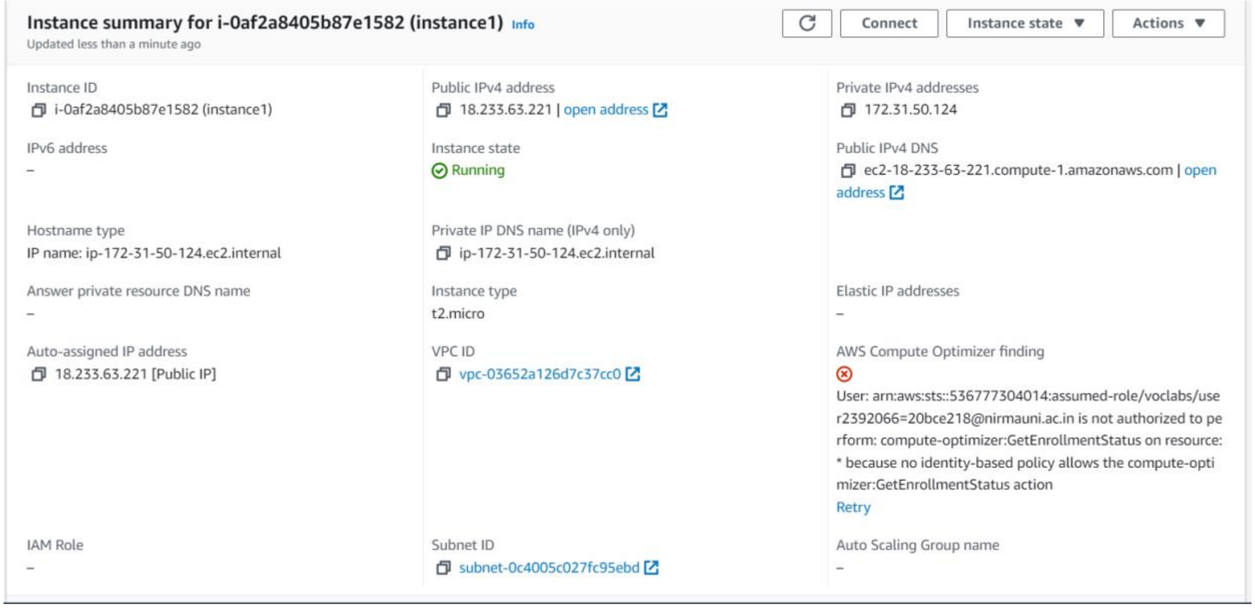
Test Event Name

OnStartSERVER

Response

```
{
  "StartingInstances": [
    {
      "CurrentState": {
        "Code": 0,
        "Name": "pending"
      },
      "InstanceId": "i-0af2a8405b87e1582",
      "PreviousState": {
        "Code": 80,
        "Name": "stopped"
      }
    }
  ],
  "ResponseMetadata": {
    "RequestId": "fc0b5d29-f0f1-4fff-90b1-e3dfc81315f6",
    "HTTPStatusCode": 200,
    "HTTPHeaders": {
      "x-amzn-requestid": "fc0b5d29-f0f1-4fff-90b1-e3dfc81315f6",
      "cache-control": "no-cache, no-store",
      "strict-transport-security": "max-age=31536000; includeSubDomains",
      "content-type": "text/xml;charset=UTF-8",
      "content-length": "579",
      "date": "Tue, 14 Mar 2023 18:24:44 GMT",
      "server": "AmazonEC2"
    },
    "RetryAttempts": 0
  }
}
```

The instance is started by executing the function:



Instance summary for i-0af2a8405b87e1582 (instance1) Info

Updated less than a minute ago

Instance ID i-0af2a8405b87e1582 (instance1)	Public IPv4 address 18.233.63.221 | open address	Private IPv4 addresses 172.31.50.124
IPv6 address –	Instance state Running	Public IPv4 DNS ec2-18-233-63-221.compute-1.amazonaws.com | open address
Hostname type IP name: ip-172-31-50-124.ec2.internal	Private IP DNS name (IPv4 only) ip-172-31-50-124.ec2.internal	
Answer private resource DNS name –	Instance type t2.micro	Elastic IP addresses –
Auto-assigned IP address 18.233.63.221 [Public IP]	VPC ID vpc-03652a126d7c37cc0	AWS Compute Optimizer finding ⊗ User: arn:aws:sts::536777304014:assumed-role/voclabs/user2392066=20bce218@nirmauni.ac.in is not authorized to perform: compute-optimizer:GetEnrollmentStatus on resource: * because no identity-based policy allows the compute-optimizer:GetEnrollmentStatus action Retry
IAM Role –	Subnet ID subnet-0c4005c027fc95ebd	Auto Scaling Group name –

Conclusion:

In this practical I have learnt about the Amazon Lambda function service. I have created a Lambda function on the AWS cloud and configured it to start and stop the VM instance provided by AWS.