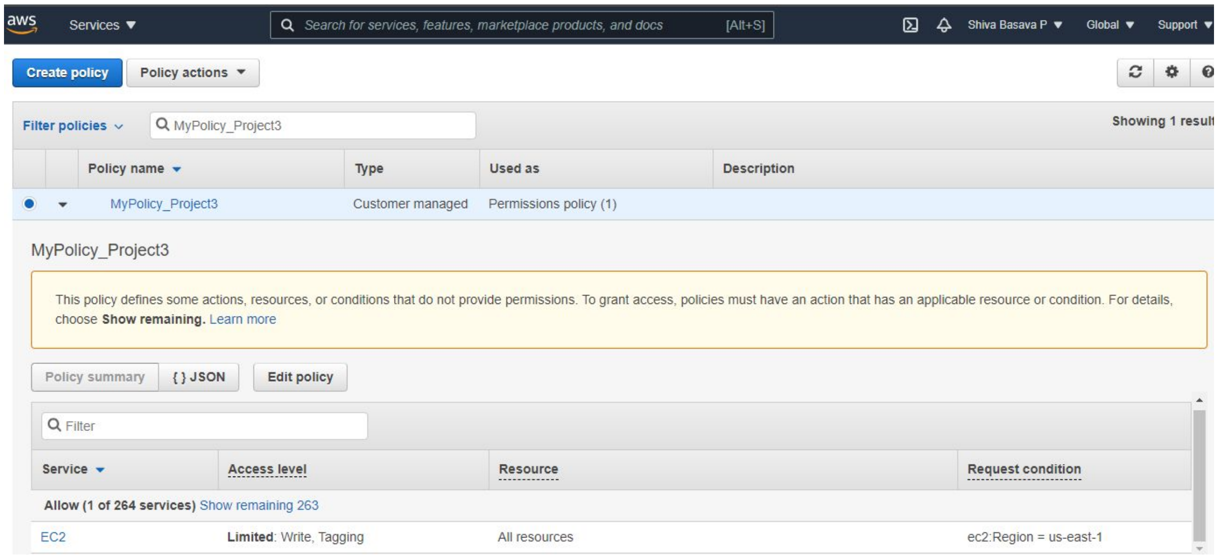
Project - 3

Project Title: Provision EC2 instance with Lambda

Following are the sequence of steps & screenshots for the solution,

I. Log into the AWS Management Console.
II. Create an IAM Policy and an IAM Role.

1. IAM Policy - **MyPolicy_Project3**



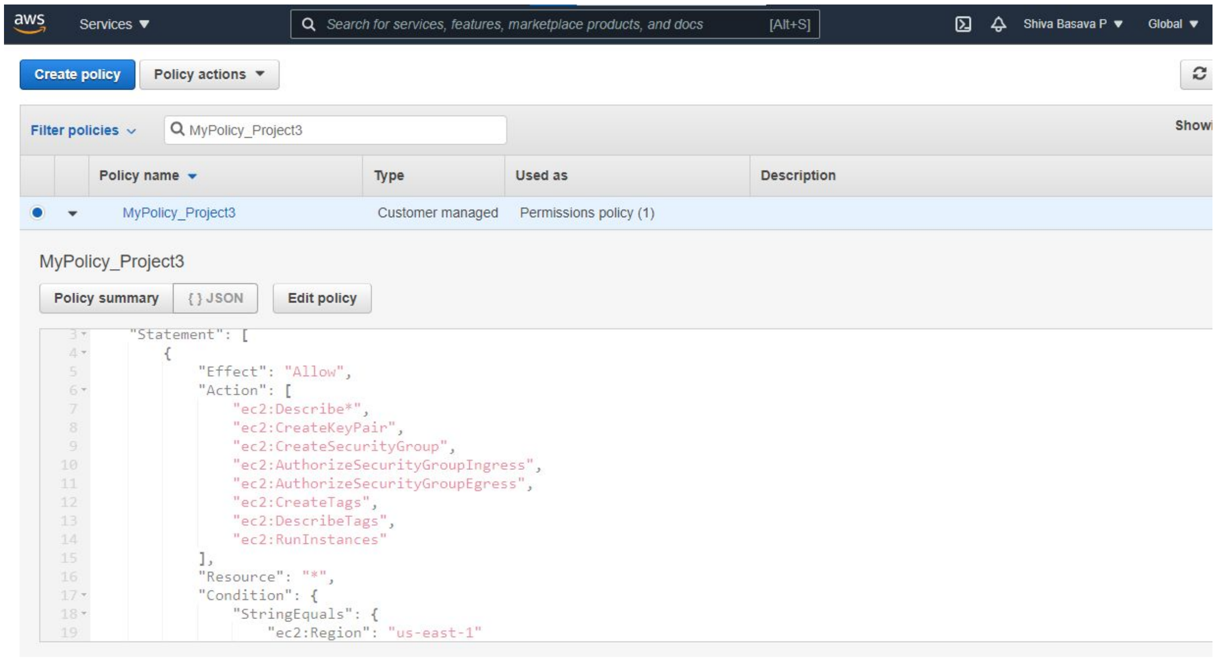
Policy name	Type	Used as	Description
MyPolicy_Project3	Customer managed	Permissions policy (1)	

MyPolicy_Project3

This policy defines some actions, resources, or conditions that do not provide permissions. To grant access, policies must have an action that has an applicable resource or condition. For details, choose **Show remaining.** Learn more

Service	Access level	Resource	Request condition
Allow (1 of 264 services) Show remaining 263			
EC2	**Limited**: Write, Tagging	All resources	ec2:Region = us-east-1

JSON file

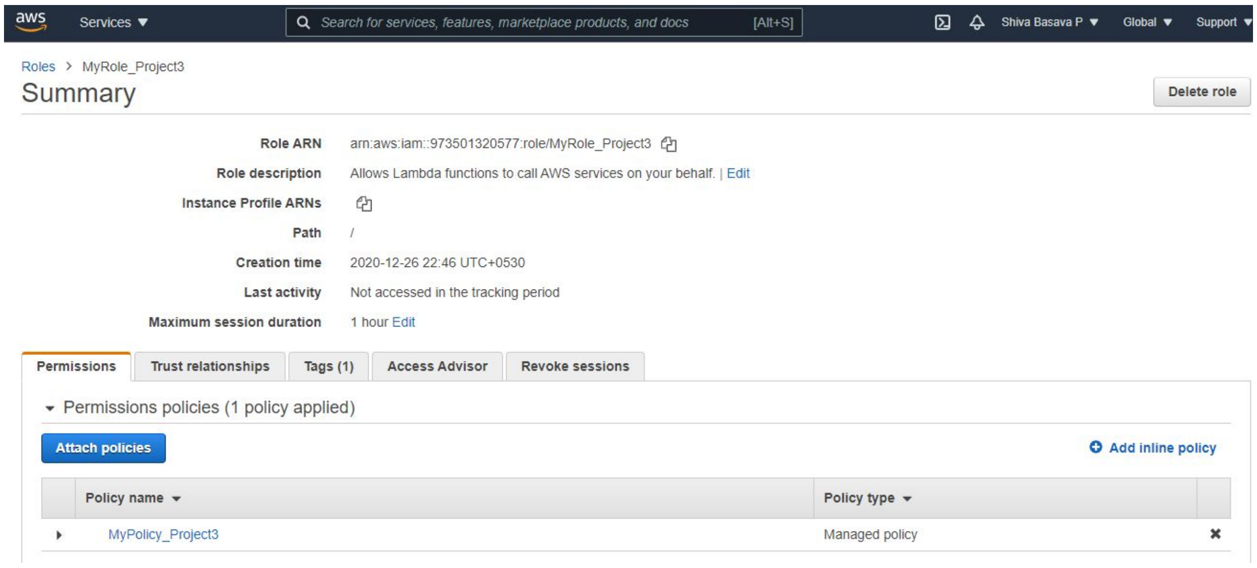


Policy name	Type	Used as	Description
MyPolicy_Project3	Customer managed	Permissions policy (1)	

MyPolicy_Project3

```
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "ec2:Describe*",
                "ec2:CreateKeyPair",
                "ec2:CreateSecurityGroup",
                "ec2:AuthorizeSecurityGroupIngress",
                "ec2:AuthorizeSecurityGroupEgress",
                "ec2:CreateTags",
                "ec2:DescribeTags",
                "ec2:RunInstances"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "ec2:Region": "us-east-1"
```

2. IAM Role - **MyRole_Project3**



Roles > MyRole_Project3

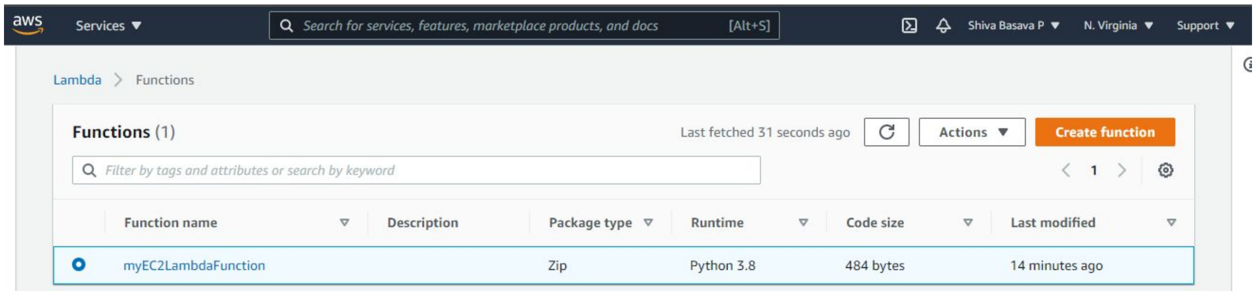
Summary

Role ARN	arn:aws:iam::973501320577:role/MyRole_Project3
Role description	Allows Lambda functions to call AWS services on your behalf. | Edit
Instance Profile ARNs	
Path	/
Creation time	2020-12-26 22:46 UTC+0530
Last activity	Not accessed in the tracking period
Maximum session duration	1 hour Edit

Permissions policies (1 policy applied)

Policy name	Policy type
MyPolicy_Project3	Managed policy

III. Created a lambda function - **myEC2LambdaFucntion**



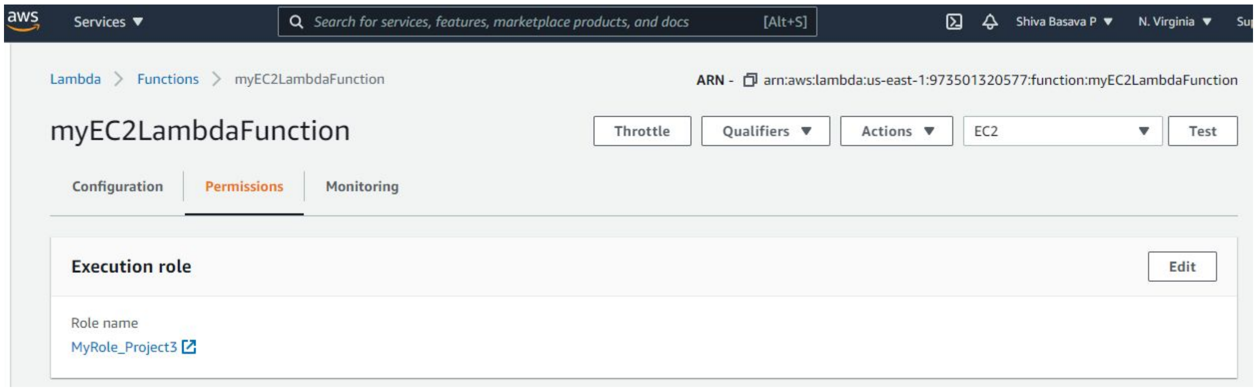
Lambda > Functions

Functions (1)

Last fetched 31 seconds ago

Function name	Description	Package type	Runtime	Code size	Last modified
myEC2LambdaFunction		Zip	Python 3.8	484 bytes	14 minutes ago

And other details of the same, Execution Role - **MyRole_Project3**



Lambda > Functions > myEC2LambdaFunction

ARN - arn:aws:lambda:us-east-1:973501320577:function:myEC2LambdaFunction

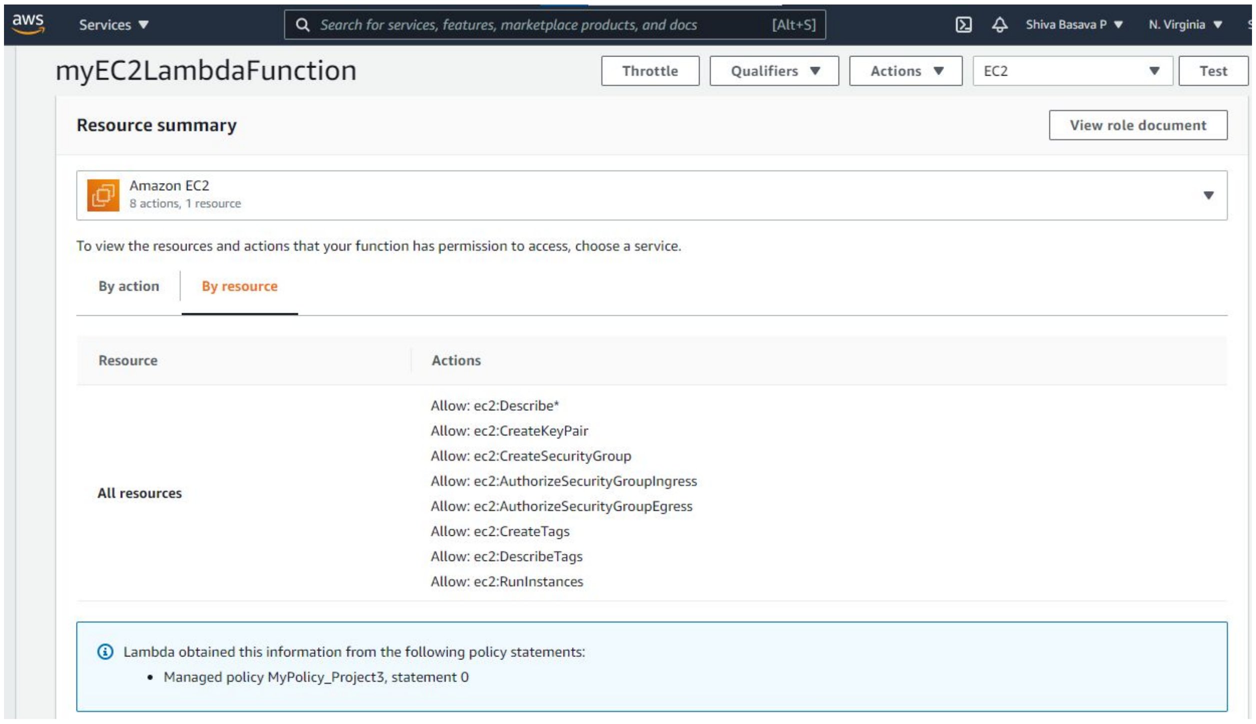
myEC2LambdaFunction

Execution role

Role name

MyRole_Project3

Execution Role - **MyPolicy_Project3**



myEC2LambdaFunction

Resource summary

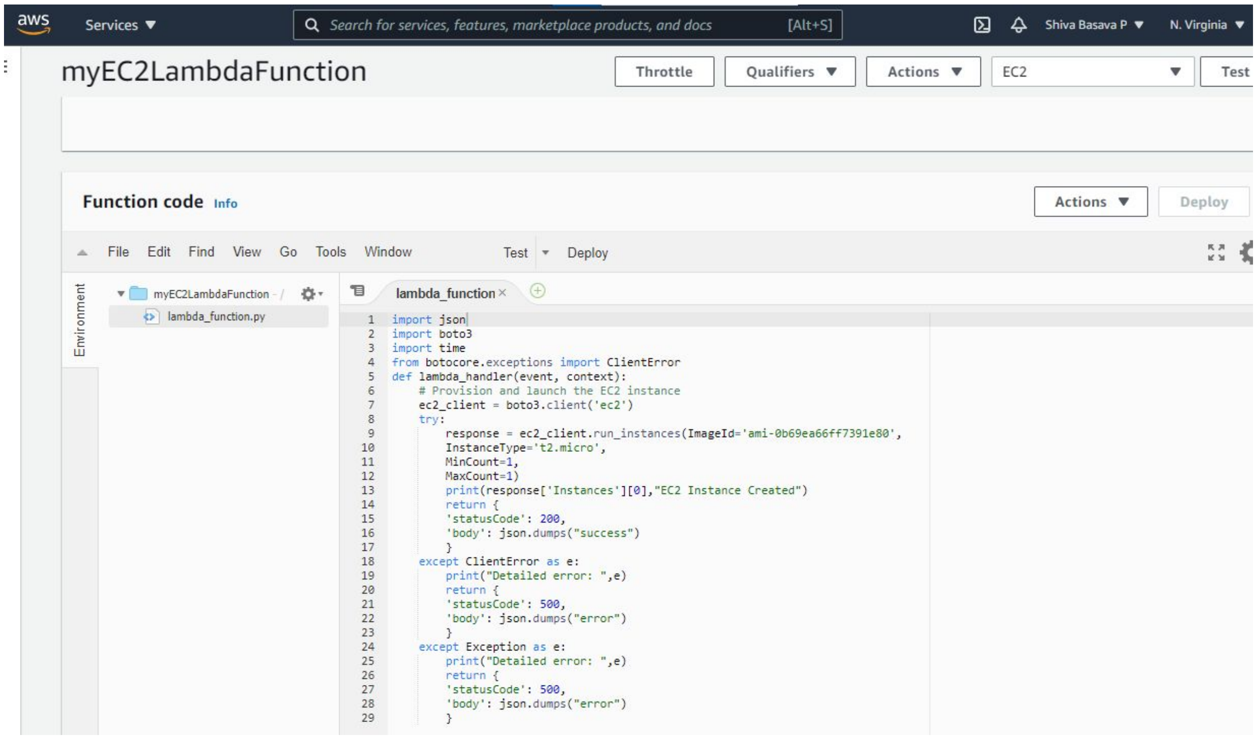
Amazon EC2
8 actions, 1 resource

To view the resources and actions that your function has permission to access, choose a service.

Resource	Actions
All resources	Allow: ec2:Describe* Allow: ec2:CreateKeyPair Allow: ec2:CreateSecurityGroup Allow: ec2:AuthorizeSecurityGroupIngress Allow: ec2:AuthorizeSecurityGroupEgress Allow: ec2:CreateTags Allow: ec2:DescribeTags Allow: ec2:RunInstances

Lambda obtained this information from the following policy statements:
- Managed policy MyPolicy_Project3, statement 0

Function Code

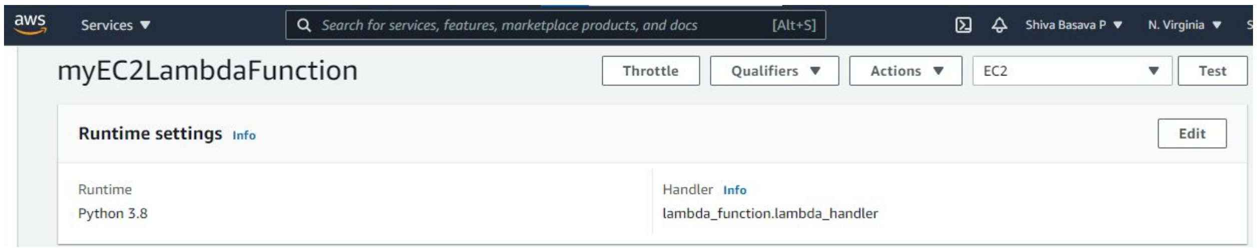


myEC2LambdaFunction

Function code

```python
import json
import boto3
import time
from botocore.exceptions import ClientError
def lambda_handler(event, context):
    # Provision and launch the EC2 instance
    ec2_client = boto3.client('ec2')
    try:
        response = ec2_client.run_instances(ImageId='ami-0b69ea66ff7391e80',
        InstanceType='t2.micro',
        MinCount=1,
        MaxCount=1)
        print(response['Instances'][0],"EC2 Instance Created")
        return {
        'statusCode': 200,
        'body': json.dumps("success")
        }
    except ClientError as e:
        print("Detailed error: ",e)
        return {
        'statusCode': 500,
        'body': json.dumps("error")
        }
    except Exception as e:
        print("Detailed error: ",e)
        return {
        'statusCode': 500,
        'body': json.dumps("error")
        }
```

Runtime Settings



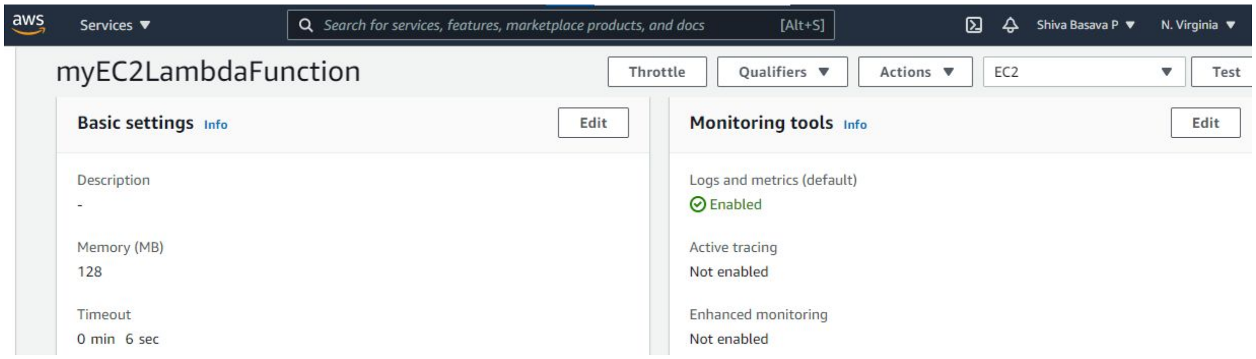
myEC2LambdaFunction

Runtime settings Info

Runtime
Python 3.8

Handler Info
lambda_function.lambda_handler

Basic Settings, **Timeout - 0 min 6 Sec**



myEC2LambdaFunction

Basic settings Info

Description
-

Memory (MB)
128

Timeout
0 min 6 sec

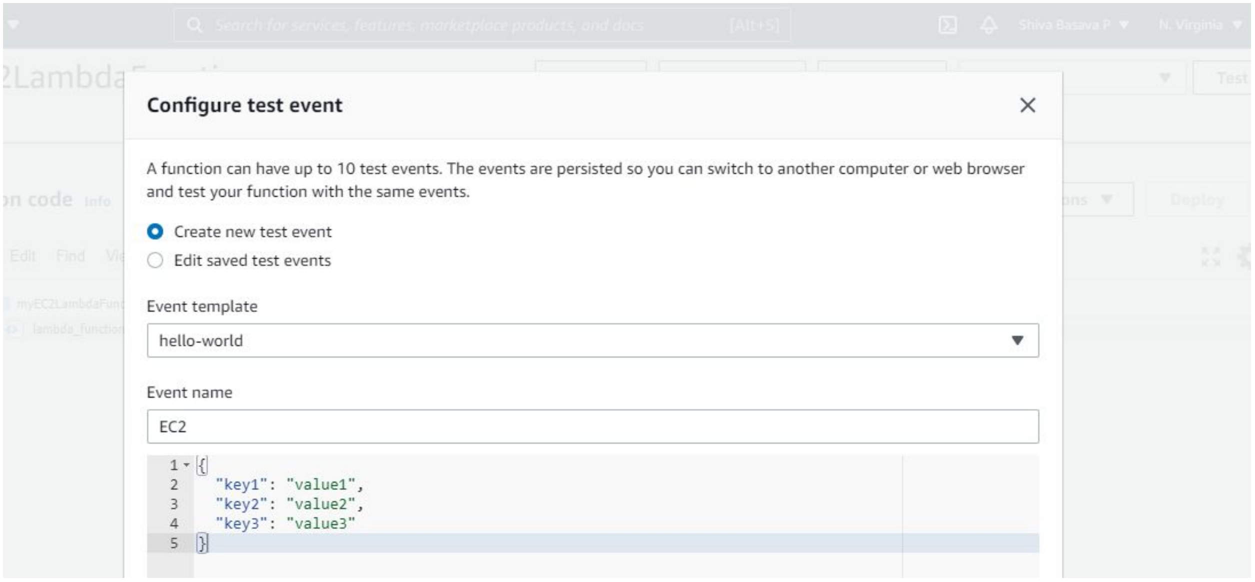
Monitoring tools Info

Logs and metrics (default)
Enabled

Active tracing
Not enabled

Enhanced monitoring
Not enabled

IV. Configuring a test event for Lambda function.



Configure test event

A function can have up to 10 test events. The events are persisted so you can switch to another computer or web browser and test your function with the same events.

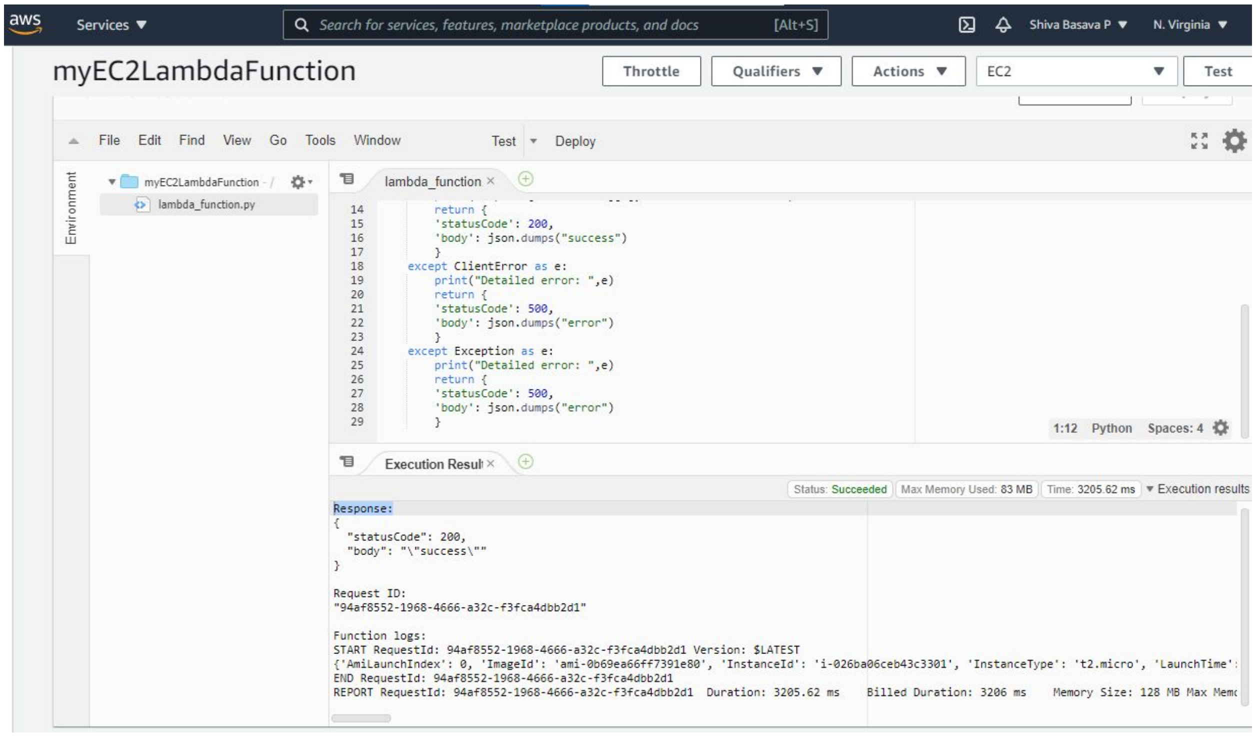
[✓ Create new test event]
Edit saved test events

Event template
hello-world

Event name
EC2

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

V. Trigger the lambda function manually using the test event.
After clicking **Test** the event - **EC2**, Response : 200



myEC2LambdaFunction

```python
        return {
        'statusCode': 200,
        'body': json.dumps("success")
        }
    except ClientError as e:
        print("Detailed error: ",e)
        return {
        'statusCode': 500,
        'body': json.dumps("error")
        }
    except Exception as e:
        print("Detailed error: ",e)
        return {
        'statusCode': 500,
        'body': json.dumps("error")
        }
```

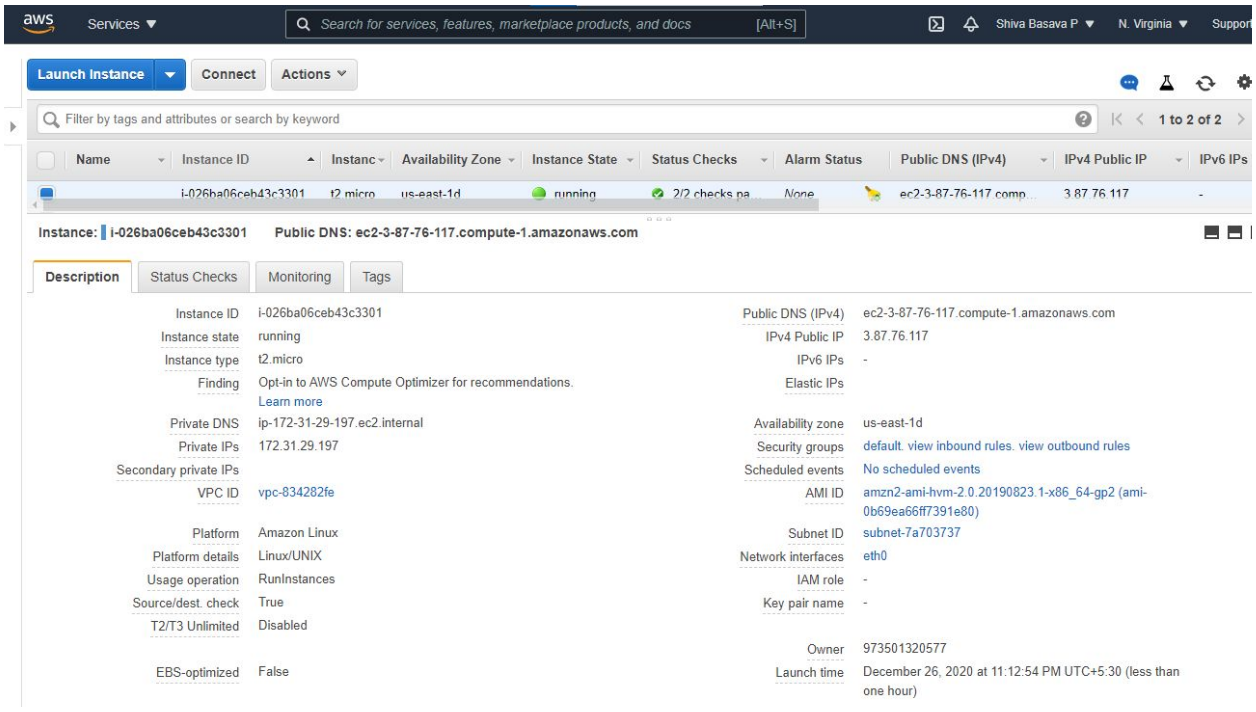
Status: Succeeded | Max Memory Used: 83 MB | Time: 3205.62 ms | Execution results

```
Response:
{
  "statusCode": 200,
  "body": "\"success\""
}

Request ID:
"94af8552-1968-4666-a32c-f3fca4dbb2d1"

Function logs:
START RequestId: 94af8552-1968-4666-a32c-f3fca4dbb2d1 Version: $LATEST
{'AmiLaunchIndex': 0, 'ImageId': 'ami-0b69ea66ff7391e80', 'InstanceId': 'i-026ba06ceb43c3301', 'InstanceType': 't2.micro', 'LaunchTime':
END RequestId: 94af8552-1968-4666-a32c-f3fca4dbb2d1
REPORT RequestId: 94af8552-1968-4666-a32c-f3fca4dbb2d1  Duration: 3205.62 ms    Billed Duration: 3206 ms    Memory Size: 128 MB Max Memc
```

EC2 is running on the console.



Name	Instance ID	Instanc	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs
	i-026ba06ceb43c3301	t2.micro	us-east-1d	● running	✓ 2/2 checks pa...	None	ec2-3-87-76-117.comp...	3.87.76.117	-

Instance: i-026ba06ceb43c3301 Public DNS: ec2-3-87-76-117.compute-1.amazonaws.com

Instance ID	i-026ba06ceb43c3301	Public DNS (IPv4)	ec2-3-87-76-117.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	3.87.76.117
Instance type	t2.micro	IPv6 IPs	-
Finding	Opt-in to AWS Compute Optimizer for recommendations. Learn more	Elastic IPs	
Private DNS	ip-172-31-29-197.ec2.internal	Availability zone	us-east-1d
Private IPs	172.31.29.197	Security groups	default. view inbound rules. view outbound rules
Secondary private IPs		Scheduled events	No scheduled events
VPC ID	vpc-834282fe	AMI ID	amzn2-ami-hvm-2.0.20190823.1-x86_64-gp2 (ami-0b69ea66ff7391e80)
Platform	Amazon Linux	Subnet ID	subnet-7a703737
Platform details	Linux/UNIX	Network interfaces	eth0
Usage operation	RunInstances	IAM role	-
Source/dest. check	True	Key pair name	-
T2/T3 Unlimited	Disabled		
		Owner	973501320577
EBS-optimized	False	Launch time	December 26, 2020 at 11:12:54 PM UTC+5:30 (less than one hour)