
# Launch an EC2 Instance with Lambda

## Advance AWS Cloud Computing With DevOps Fundamentals

Trainer : Mrs. Vinolin Jermiah

Microsoft account [15 Dec 2020]

**Project Question: Provision EC2 instance with Lambda**

## Tasks

1. Log into the AWS Management Console.

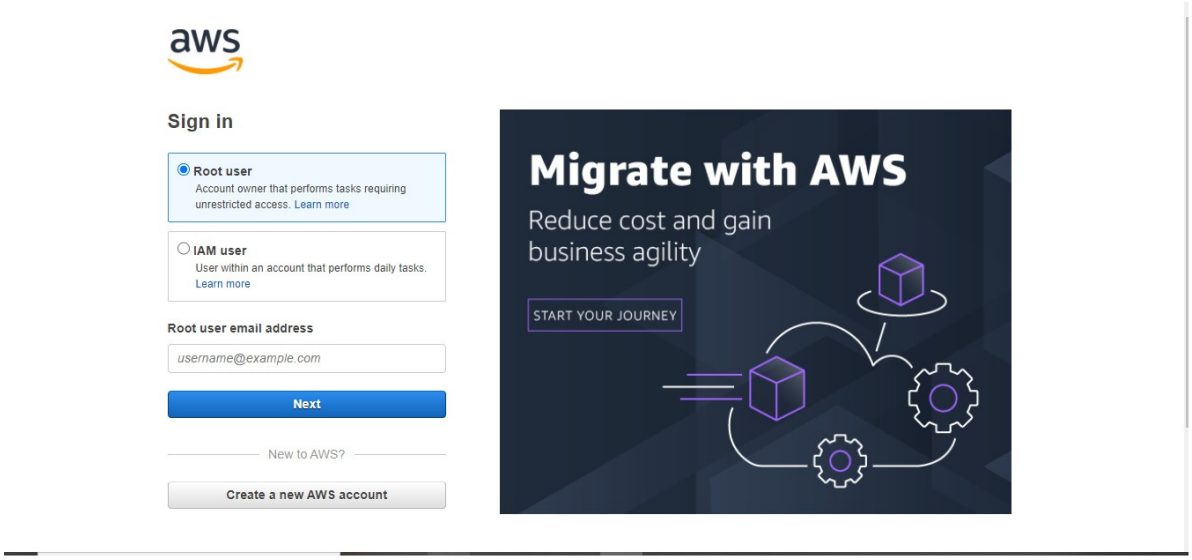
2. Create an IAM Policy and an IAM Role.

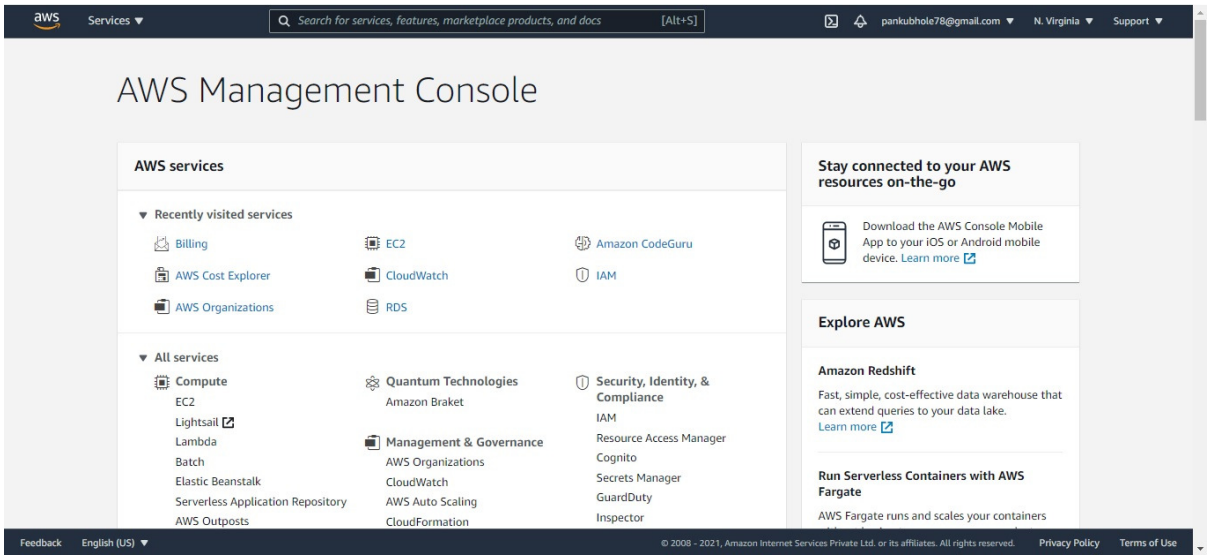
3. Create a lambda function.

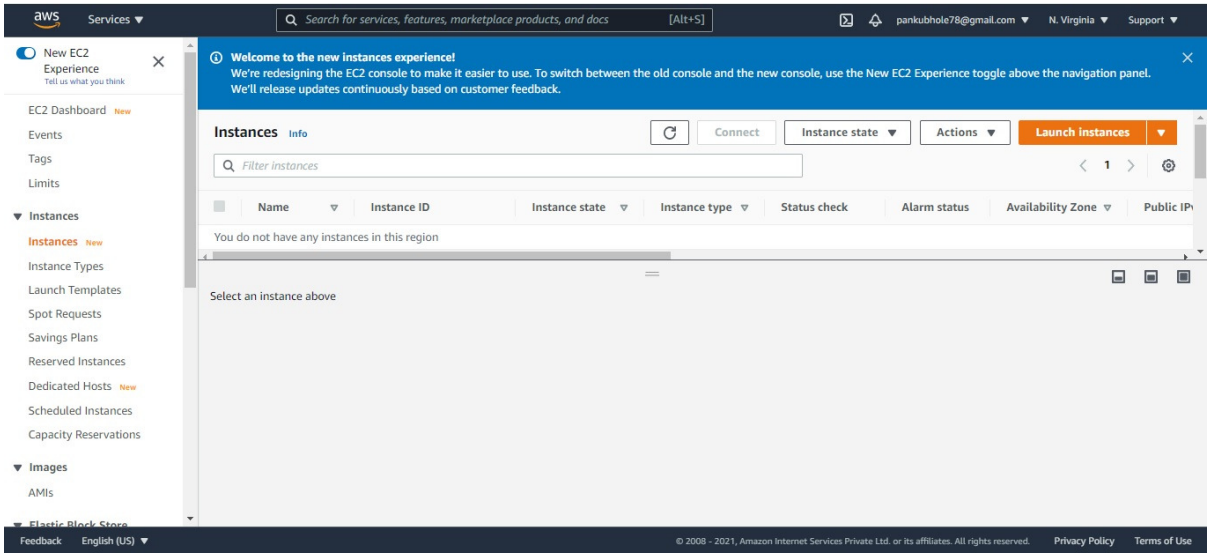
4. Configure a test event.

5. Trigger the lambda function manually using the test event. Test the new EC2 instance.

## Login to EC2 Console :





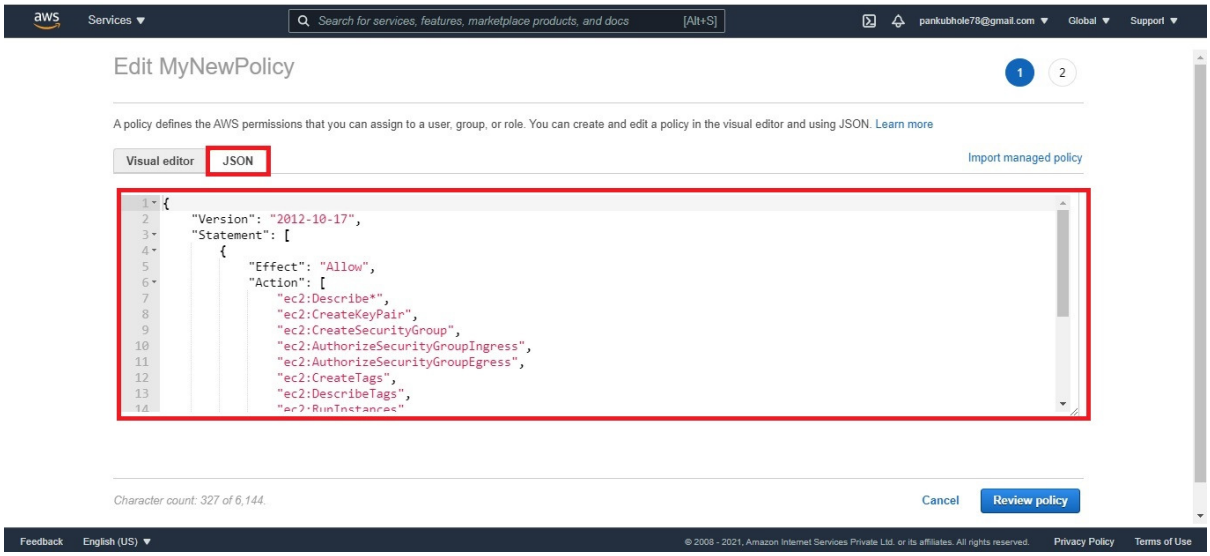

## Create an IAM Policy

As a prerequisite for creating Lambda function, we need to create a user role with a custom policy.

Go to services and select IAM.

Click on policies and click create policy.

Click on the JSON tab and copy / paste the below policy statement into the editor:



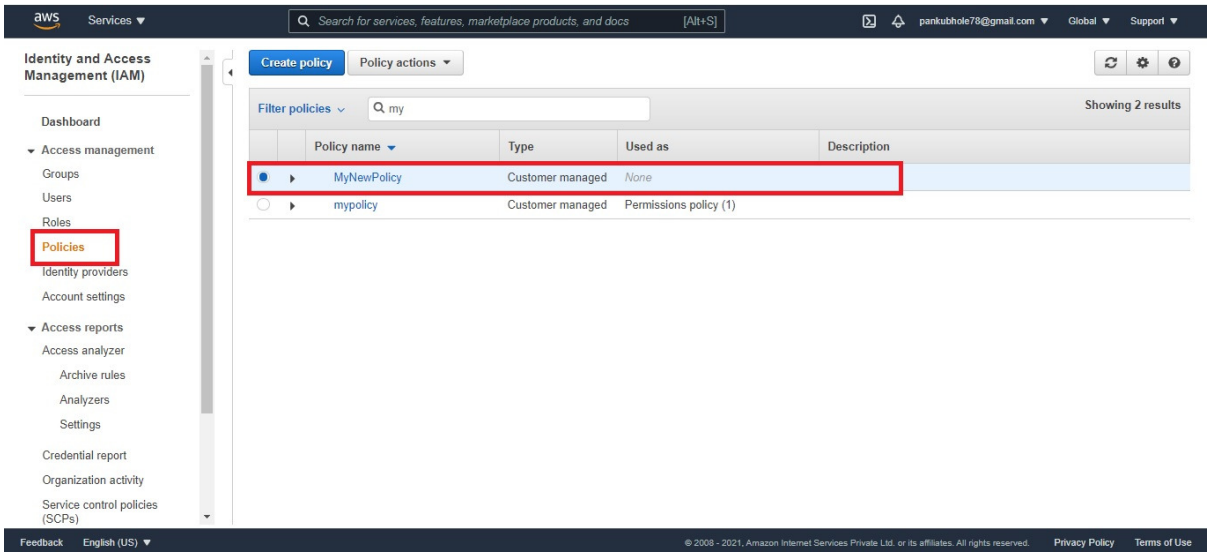
Policy JSON:

```
{
"Version": "2012-10-17",
"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"
}
}
}
]
}
```

Click on Review policy.

On Create Policy page:

Policy Name: **MyNewpolicy.**
Click on the **create policy button.**




An IAM Policy named **MyNewpolicy** is now created.

**Create an IAM Role**

In the left menu, click on Roles. Click on the create role button.

Select Lambda from the AWS Services list.

Click on Next: permissions.

Filter Policies: Search for your policy by name (**MyNewpolicy**).

Select your policy and click on the Next: tags.

Add Tags: Provide a key-value pair for the role:

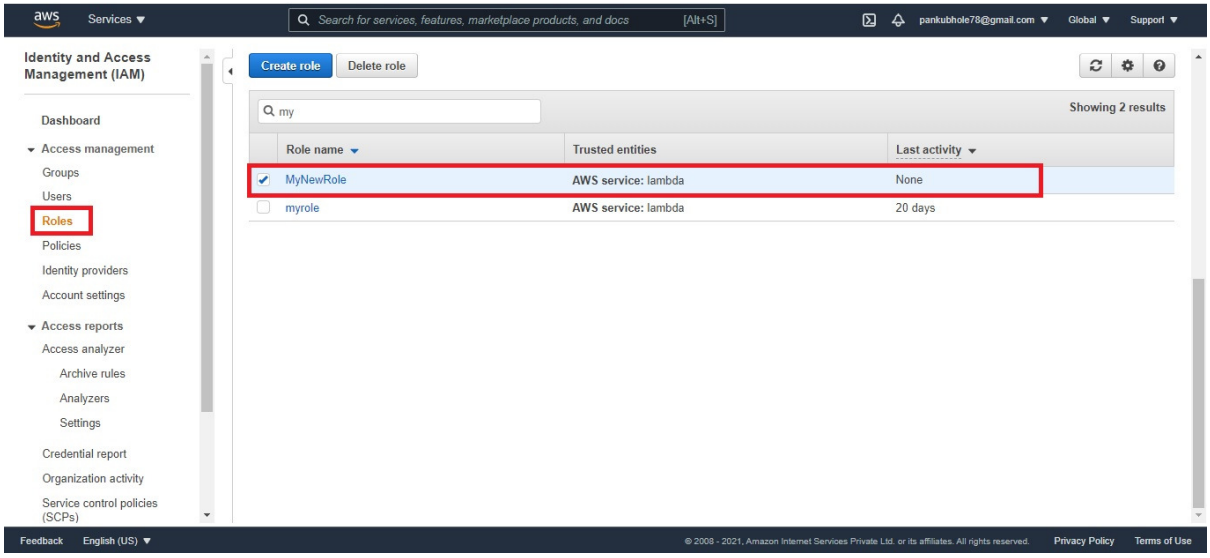
Key: name

Value: **MyNewRole**

Click on the Next: review button.

Role name: **MyNewRole**

Click on the create role button.




You have successfully created an IAM role by the name **MyNewRole**.

**Create a Lambda Function**

Go to the services menu and click on lambda.

Make sure you are in the **US East (N. Virginia) us-east-1 region.**

Click on the create function button.

Choose an author from scratch.

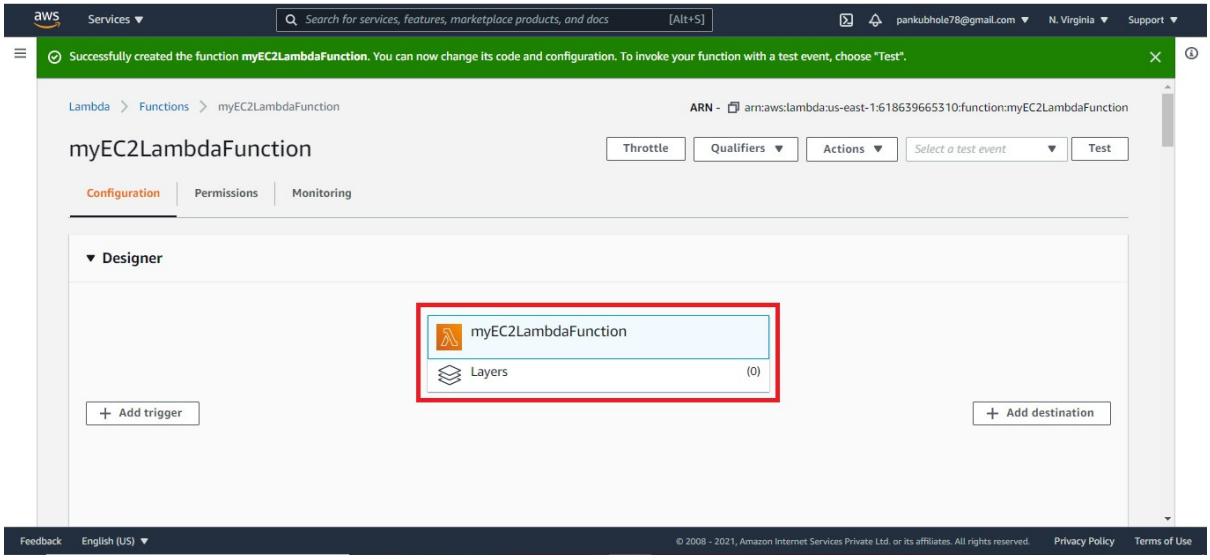
**Function name: myEC2LambdaFunction**

Runtime: Select Python 3.8

Role: In the permissions section, click on to use an existing role.

**Existing role: Select MyNewRole**

Click on **create function.**

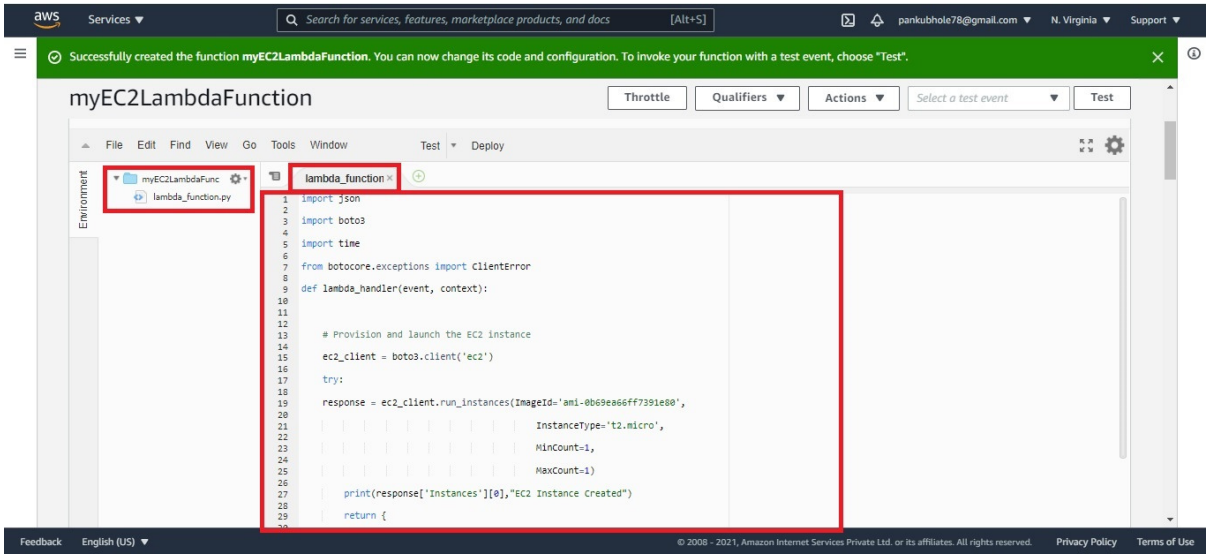



Configuration Page: Here we need to configure our lambda function. If you scroll down, you can see the

Function code section. Here we need to write some Python code to provision an EC2 instance.

You will be using boto3 SDK for AWS to write the python code.

Remove the existing code in the **lambda_function.py file**. **Copy the below code and paste it into your lambda_function.py file.**




**Note: We will not get into the details of the python code in this lab, but at a high level, it will provision an EC2 instance when it is triggered.**

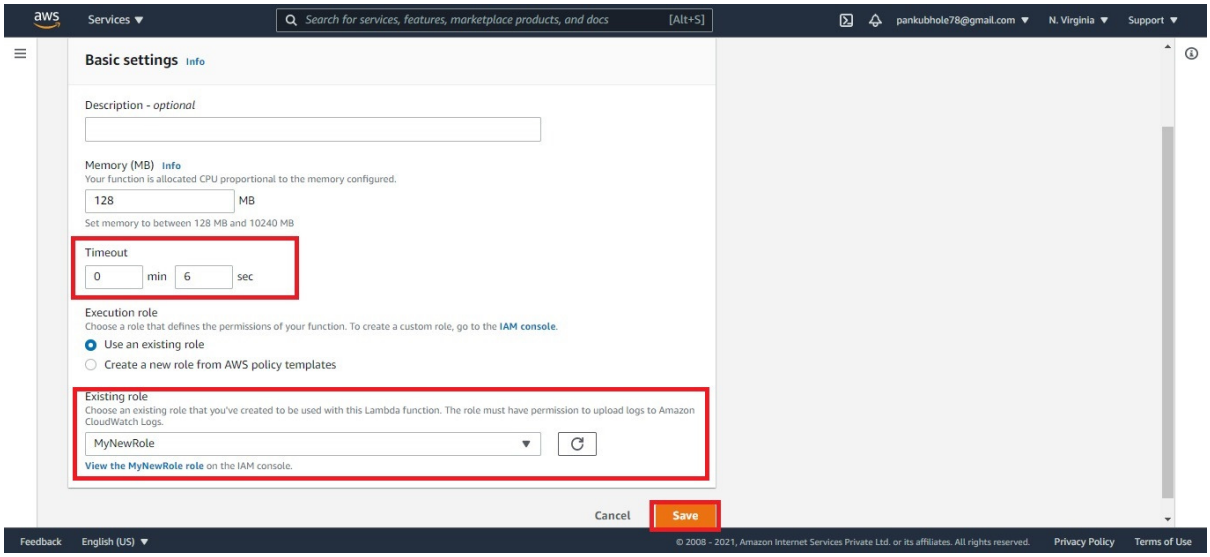
```
import json
import boto3
import time
from botocore.exceptions import ClientError
def lambda_handler(event, context):
    # Provision and launch the EC2 instance
    ec2_client = boto3.client('ec2')
    try:
 response = ec2_client.run_instances(ImageId='ami
0b69ea66ff7391e80',
InstanceType='t2.micro',
                                    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")
    }
```

7. Save the function by clicking on deploy in the top right corner.

8. Now scroll down the page and you will be able to see Basic settings click on the button.

9. In the **Timeout set seconds as 6 and then** click on the **save** button.




Configure a Test Event

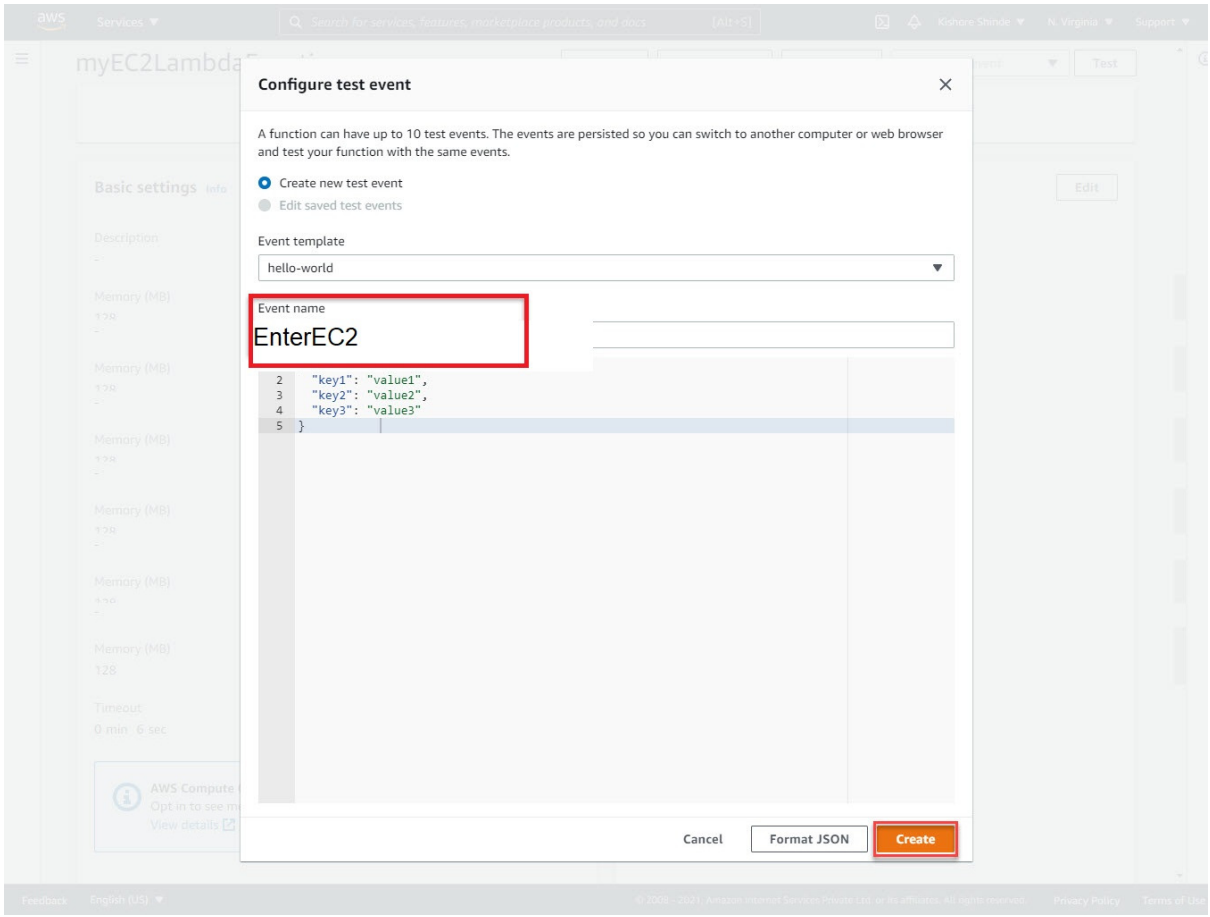
Click on the Test button at the top right corner of the page (under the ARN).

**In Configure test event page**,

**Event Name: EnterEc2**

Leave other fields as default.

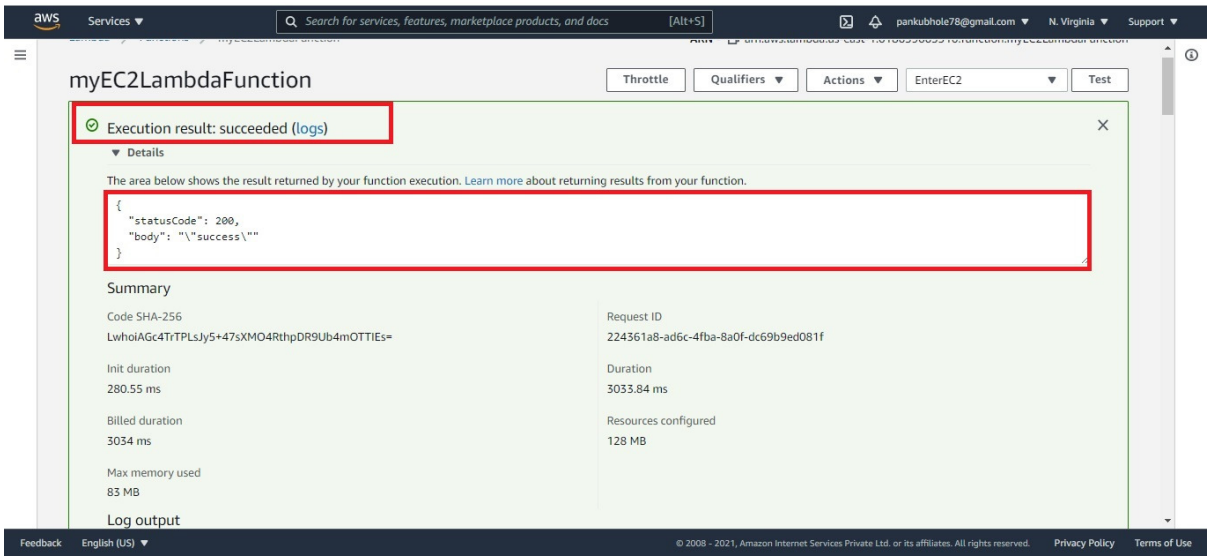
Click on create.




Provision an EC2 Instance using a Lambda Function

Once the ec2 is configured, we can trigger the lambda manually using this simple test.

Click on the Test button.

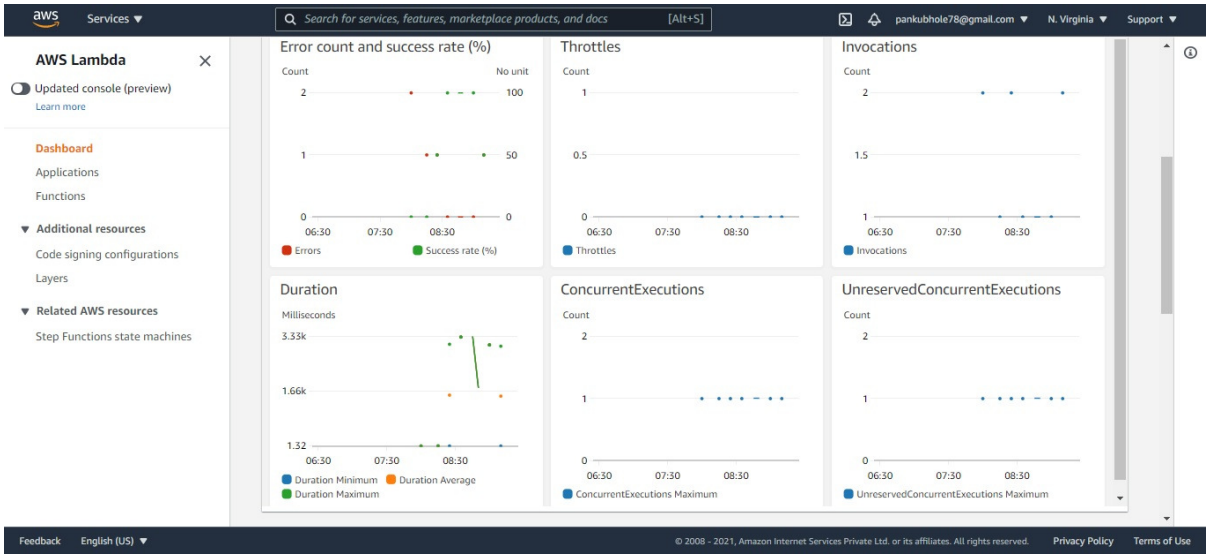



The lambda function now gets executed and an EC2 instance will be provisioned.

Once it's completed, you will be seeing a success message similar to the example shown below. It will

Display details such as:

Duration: Lambda execution time.

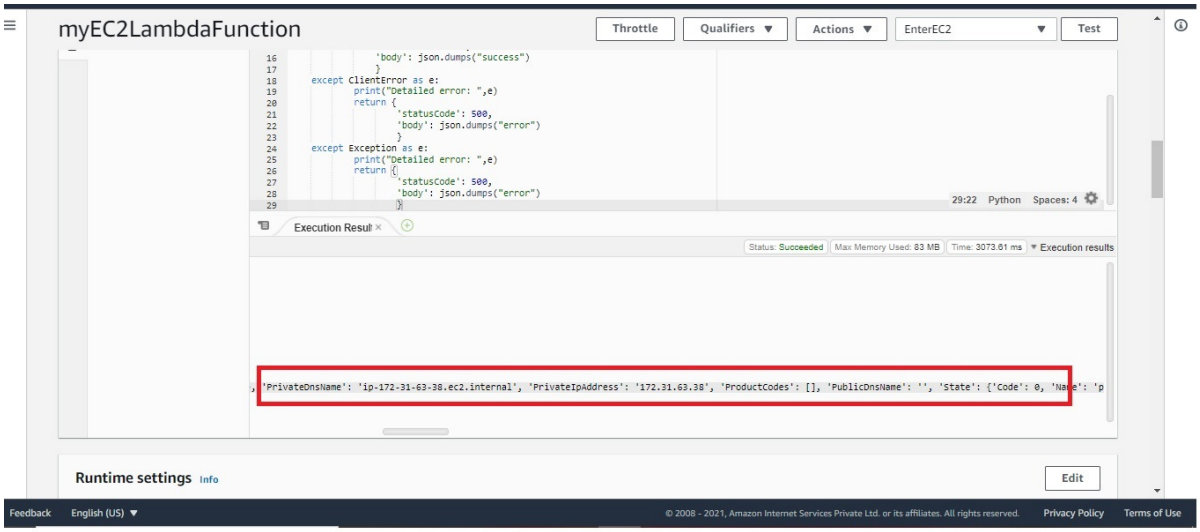



Log Output: Details of the EC2 instance provisioned.

etc...

Check that the EC2 instance launched successfully.

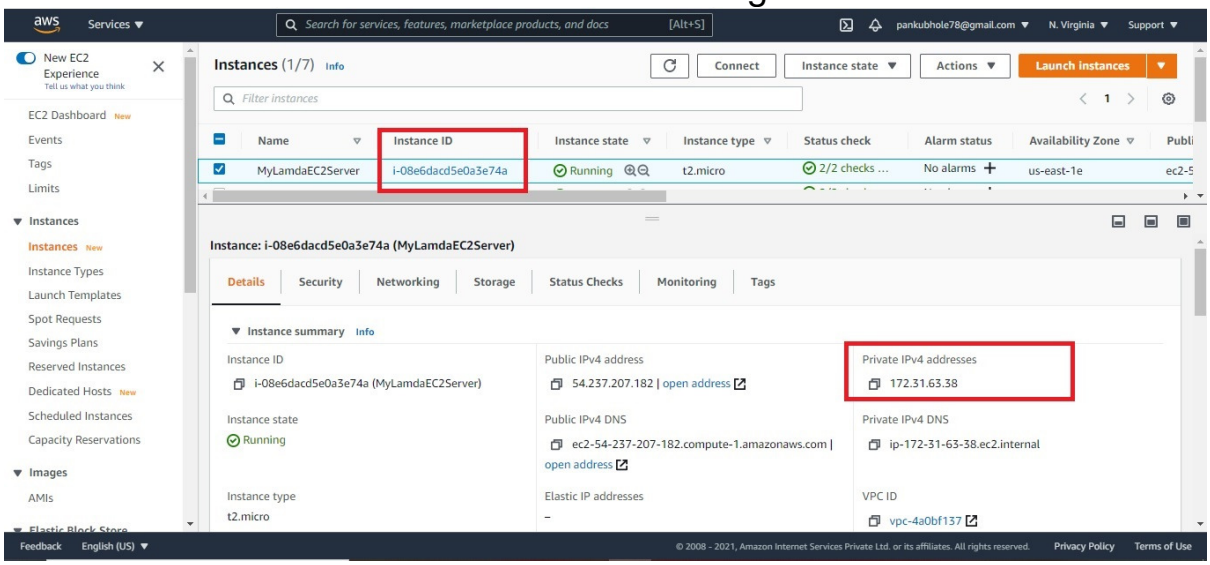
Navigate to the EC2 page from the services menu.




**(IP Address: 172.31.63.38)**

Switch off the New EC2 experience. Edit the feedback message and select yes for the experience. Click On.

This will allow us to use the old console.
Go to Instances in the left menu.
Please wait until the ec2 instance is in running state




**IP Address: 172.31.63.38**

You can see the EC2 instance that has been provisioned by the Lambda function