Infrastructure as code (IaC) is the process of managing and provisioning IT resources through machine-readable definition files, rather than physical hardware configuration or interactive configuration tools. This includes bare-metal servers as well as VMs and the associated configuration resources.

In any given enterprise which uses IaC for VM deployment and management, the process consists of the following 2 steps:

1) Create an OS image pre-configured with the required software and applications
2) Use the IaC system to create VMs based on the configured OS and with the required permissions

In this DIY exercise, we will be using custom Python code in an AWS Lambda function to perform the second step, i.e. using IaC to deploy VMs.

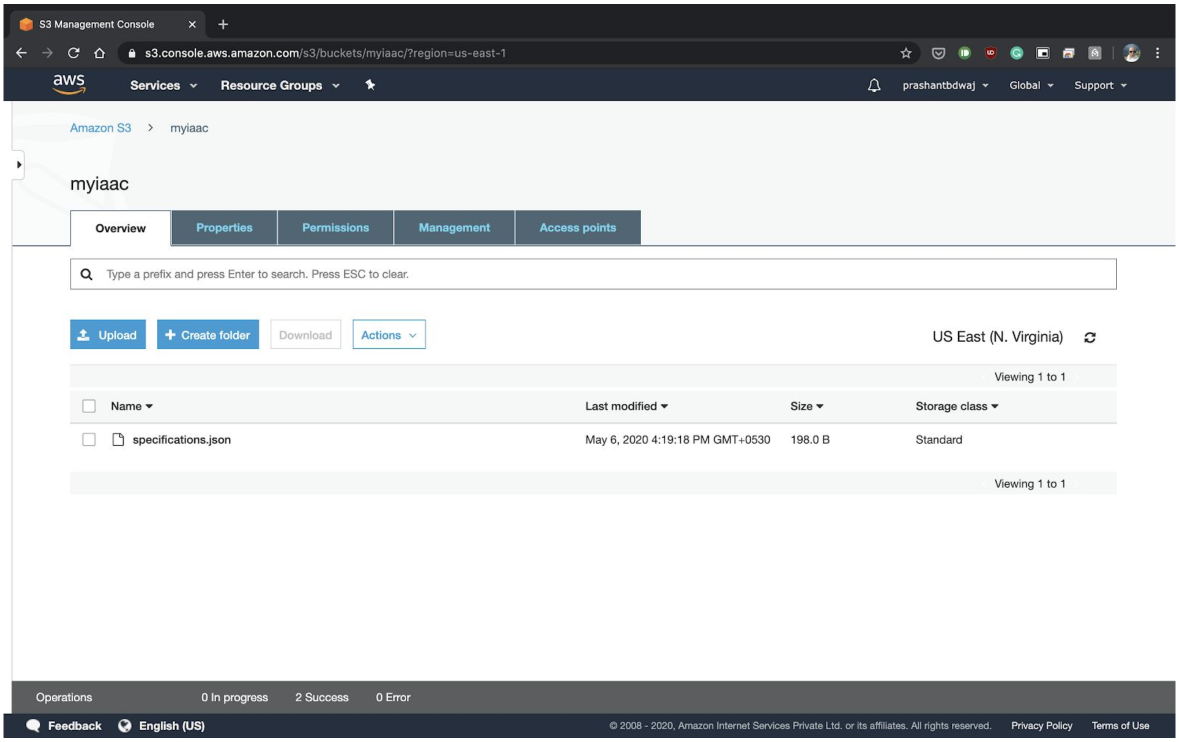
Requirements:

1) AWS account: This exercise should not be performed on an Educate account as some features may not be available
2) Configuration files:
    a) Lambda program: *myIaaC.py*
    b) JSON file: *specification.json*
    c) Test event JSON file: test.json

Steps to be taken:

1) Ensure that the selected region is N.Virginia (us-east-1)
2) Navigate to S3 and create a bucket called *myiaac-[SOMENUMBERS]*
    a) Eg myiaac-55898412 to make the bucket name unique

3) Upload the provided specification.json file to the bucket



myiaac

US East (N. Virginia)

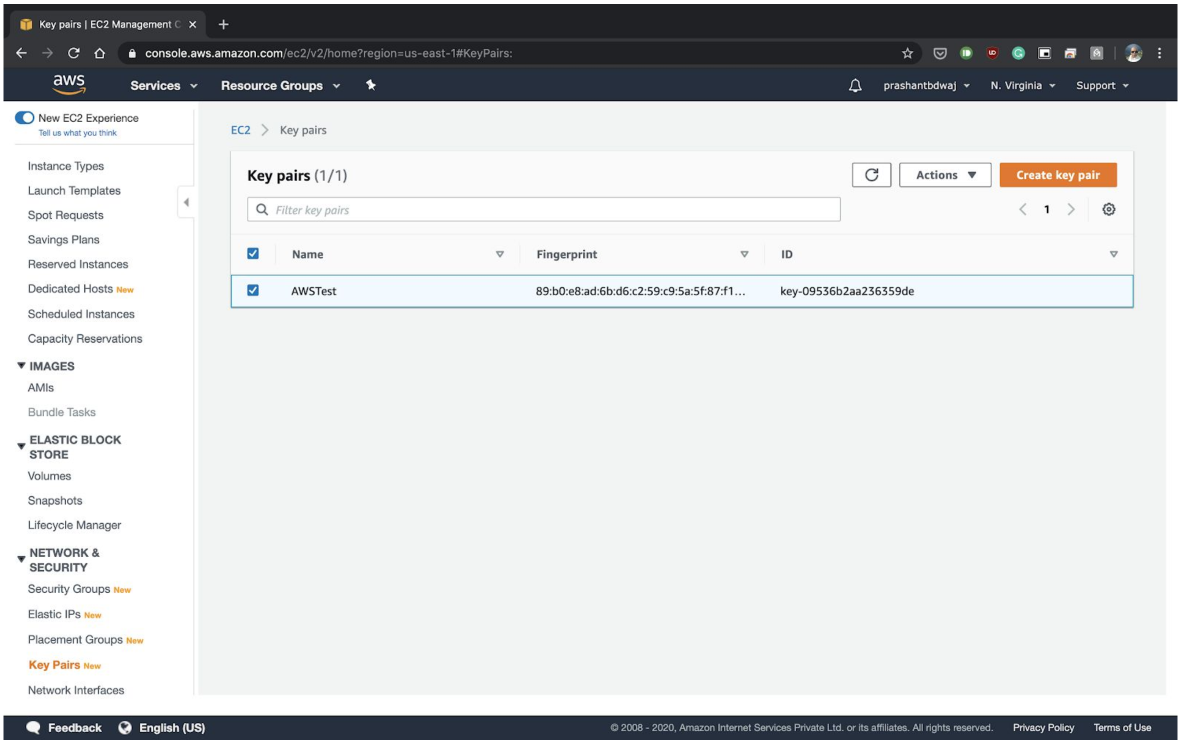
Viewing 1 to 1

	Name	Last modified	Size	Storage class
□	specifications.json	May 6, 2020 4:19:18 PM GMT+0530	198.0 B	Standard

Viewing 1 to 1

Operations 0 In progress 2 Success 0 Error

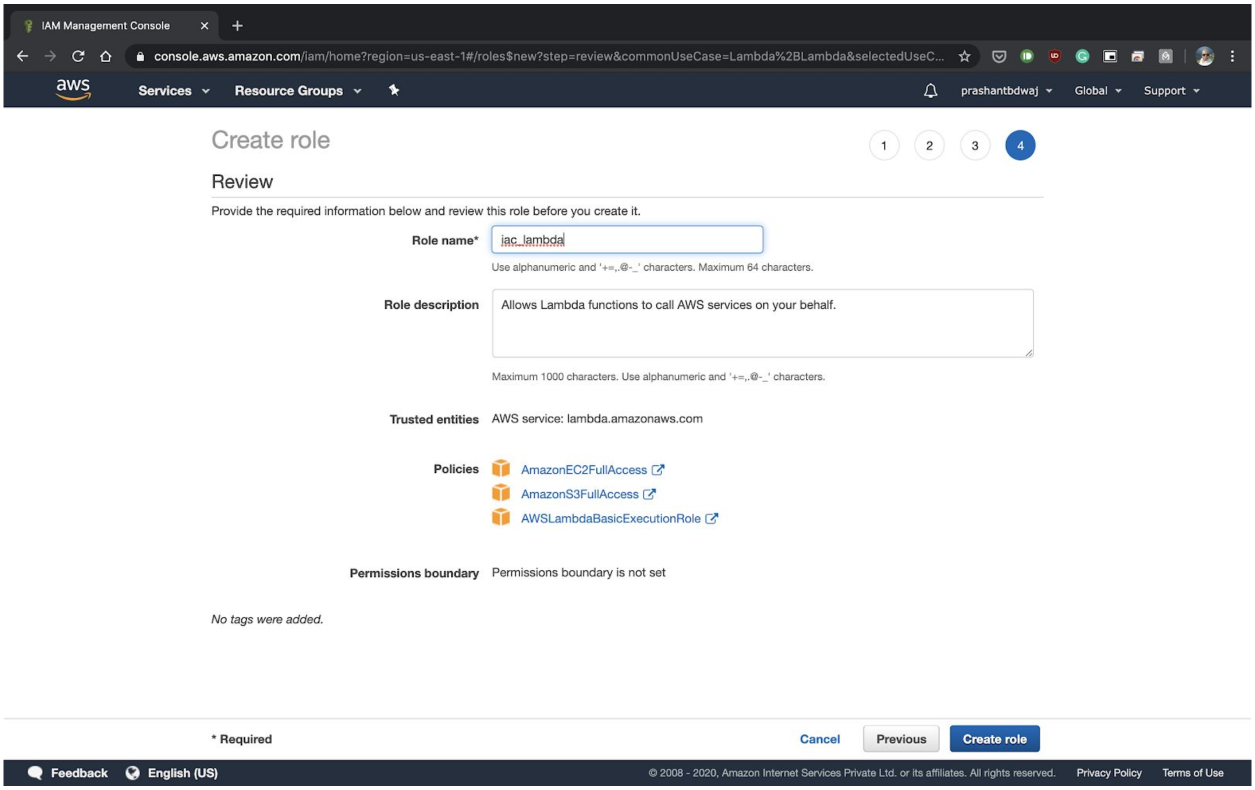
4) Navigate to EC2, then go to Key Pairs and create a new keypair called *AWSTest.*



Key pairs (1/1)

☑	Name	Fingerprint	ID
☑	AWSTest	89:b0:e8:ad:6b:d6:c2:59:c9:5a:5f:87:f1...	key-09536b2aa236359de

5) Verify that there is no security group called *serversg* by navigating to Security groups under EC2. If there is, then delete it.
6) Navigate to IAM->Roles
7) Create a new role called *iac_lambda* with AWS Lambda as the use case and attach the following policies to it
    a) AWSLambdaBasicExecutionRole
    b) AmazonS3FullAccess
    c) AmazonEC2FullAccess



Create role

Review

Provide the required information below and review this role before you create it.

Role name* iac_lambda

Use alphanumeric and '+=,.@-_' characters. Maximum 64 characters.

Role description Allows Lambda functions to call AWS services on your behalf.

Maximum 1000 characters. Use alphanumeric and '+=,.@-_' characters.

Trusted entities AWS service: lambda.amazonaws.com

Policies AmazonEC2FullAccess
AmazonS3FullAccess
AWSLambdaBasicExecutionRole

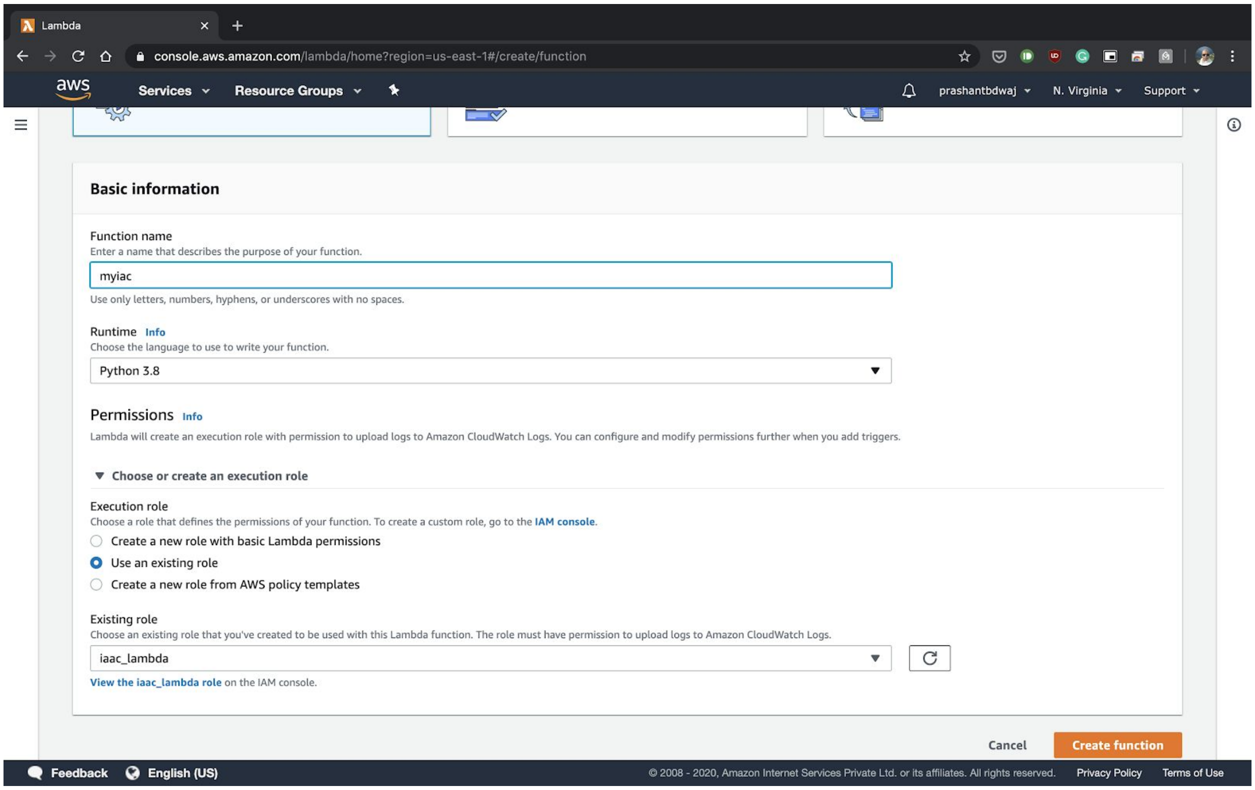
Permissions boundary Permissions boundary is not set

No tags were added.

* Required

© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.

8) Navigate to AWS Lambda
9) Create a Lambda function from scratch with the name *myiac* and runtime set to Python 3.8
10) Select the above-created role *iac_lambda* as an execution role by selecting “Use an existing role” and click on Create


Lambda
console.aws.amazon.com/lambda/home?region=us-east-1#/create/function
aws
Services
Resource Groups
prashantbdwaj
N. Virginia
Support

Basic information

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

Runtime Info
Choose the language to use to write your function.
Python 3.8

Permissions Info
Lambda will create an execution role with permission to upload logs to Amazon CloudWatch Logs. You can configure and modify permissions further when you add triggers.

▼ Choose or create an 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.
iaac_lambda
View the iaac_lambda role on the IAM console.

Cancel
Create function
Feedback
English (US)
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

11) Under Configuration, select Add Trigger and enter the following details

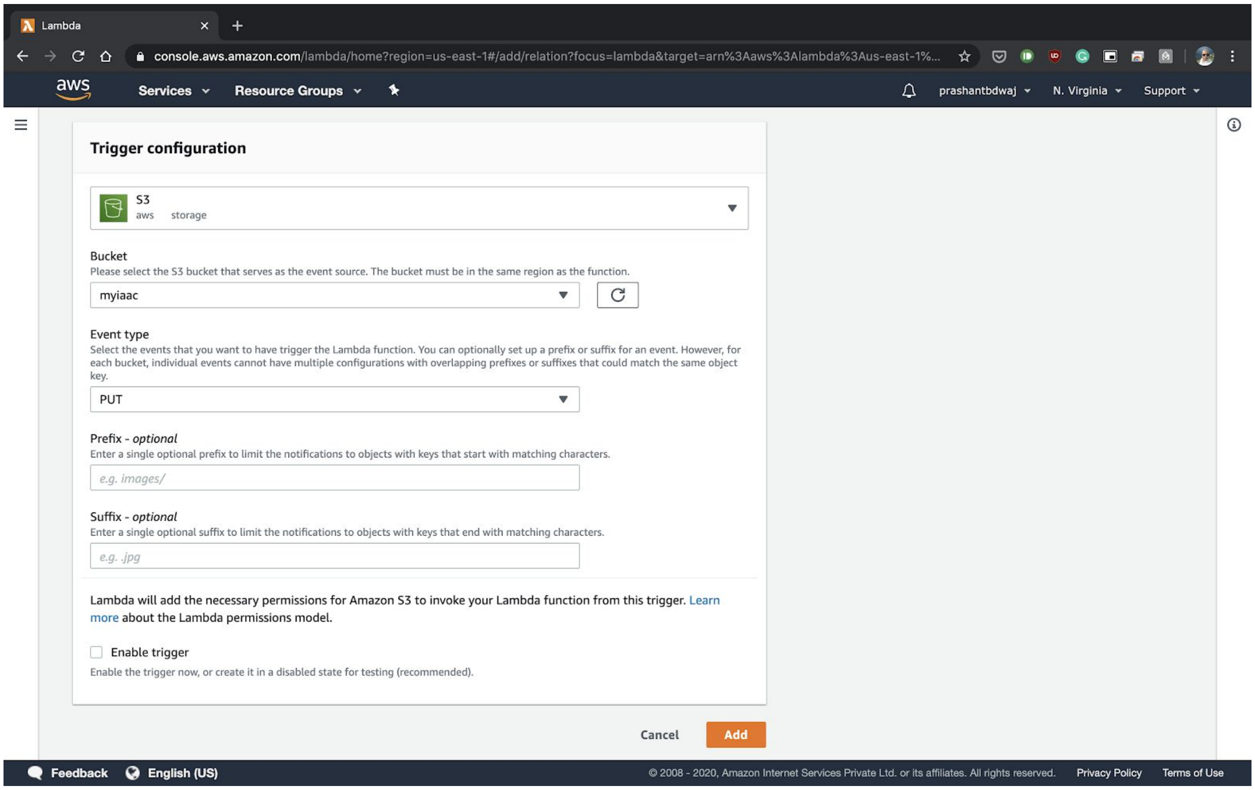
   a) Trigger: S3

   b) Bucket name: myiaac

   c) Event type : PUT

   d) Uncheck “Enable now”

   Rest of the fields need not be filled


Lambda
console.aws.amazon.com/lambda/home?region=us-east-1#/add/relation?focus=lambda&target=arn%3Aaws%3Alambda%3Aus-east-1%...
aws
Services
Resource Groups
prashantbdwaj
N. Virginia
Support

Trigger configuration

S3
aws storage

Bucket
Please select the S3 bucket that serves as the event source. The bucket must be in the same region as the function.
myiaac

Event type
Select the events that you want to have trigger the Lambda function. You can optionally set up a prefix or suffix for an event. However, for each bucket, individual events cannot have multiple configurations with overlapping prefixes or suffixes that could match the same object key.
PUT

Prefix - *optional*
Enter a single optional prefix to limit the notifications to objects with keys that start with matching characters.
e.g. images/

Suffix - *optional*
Enter a single optional suffix to limit the notifications to objects with keys that end with matching characters.
e.g. .jpg

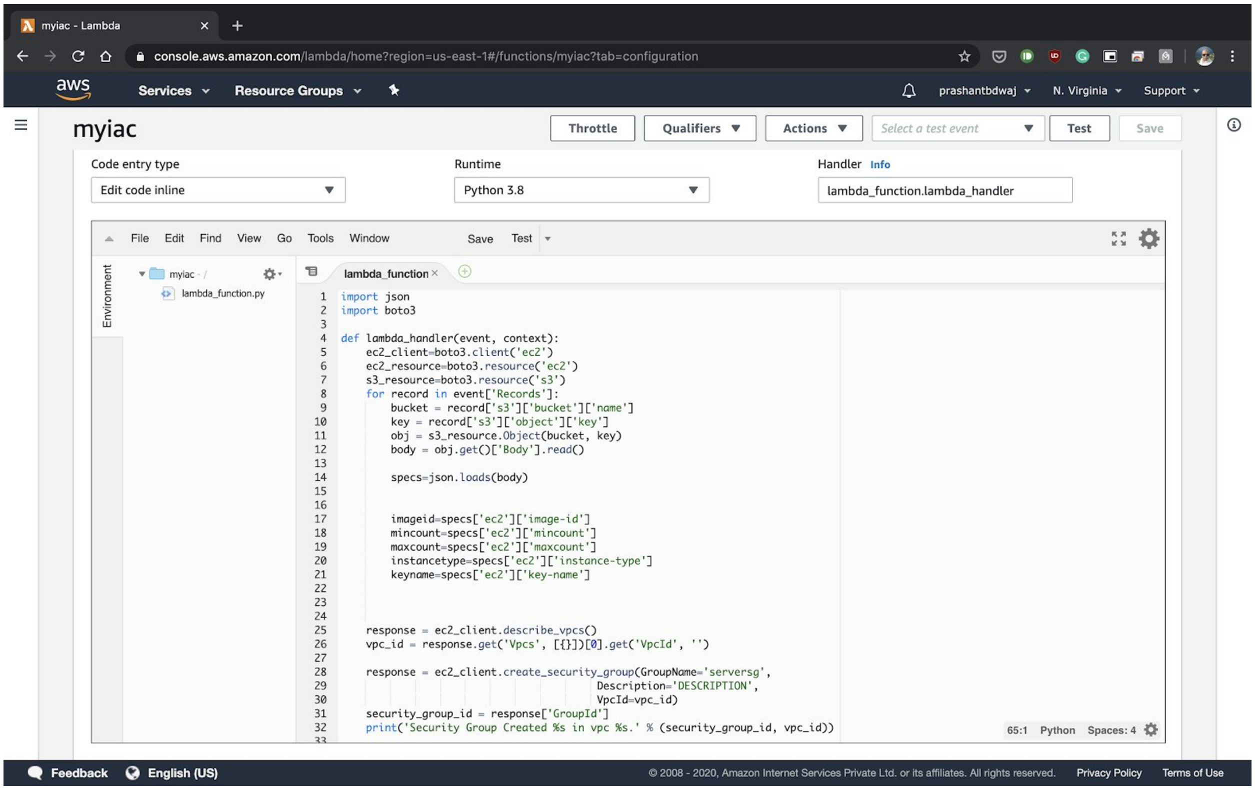
Lambda will add the necessary permissions for Amazon S3 to invoke your Lambda function from this trigger. Learn more about the Lambda permissions model.

□ Enable trigger
Enable the trigger now, or create it in a disabled state for testing (recommended).

Cancel
Add
Feedback
English (US)
© 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved.
Privacy Policy
Terms of Use

12) Under Configuration, scroll down to Basic Settings and set the timeout to 2 minutes

13) Under Function Code, ensure that the runtime is set to Python 3.8 and replace the text with the code given in the file myIaaC.py and press Save on the top right corner



Code entry type: Edit code inline

Runtime: Python 3.8

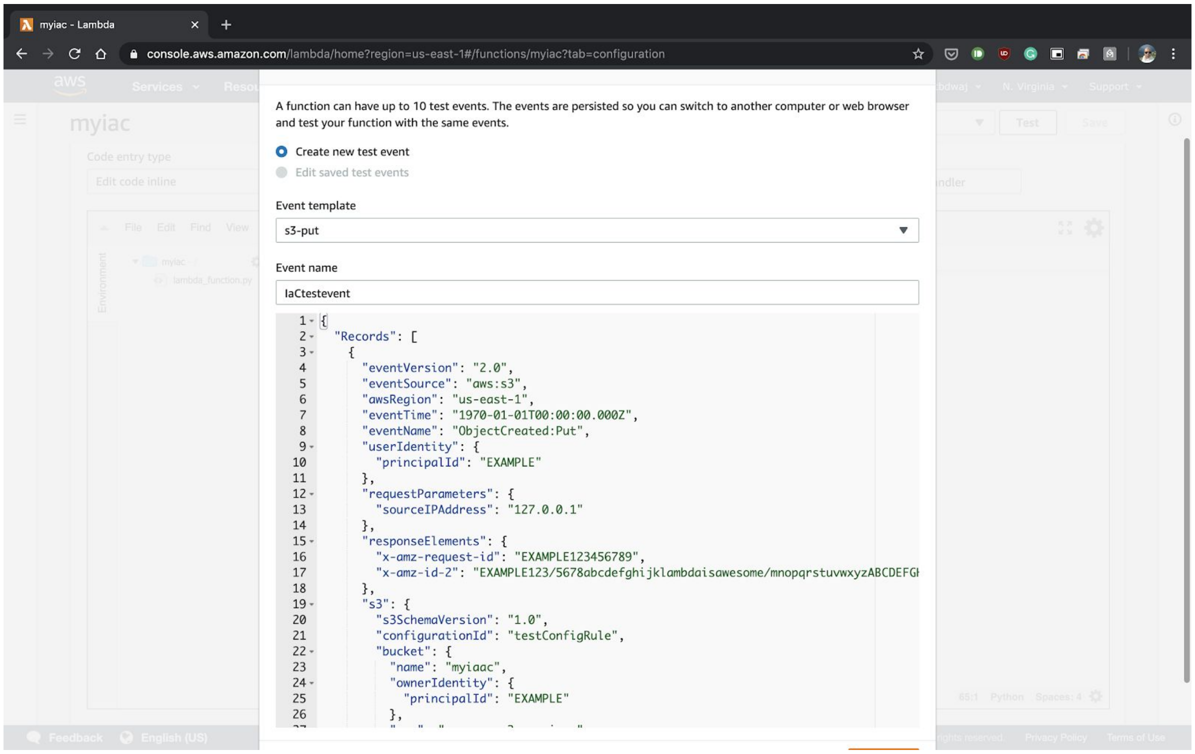
Handler: lambda_function.lambda_handler

```python
import json
import boto3

def lambda_handler(event, context):
    ec2_client=boto3.client('ec2')
    ec2_resource=boto3.resource('ec2')
    s3_resource=boto3.resource('s3')
    for record in event['Records']:
        bucket = record['s3']['bucket']['name']
        key = record['s3']['object']['key']
        obj = s3_resource.Object(bucket, key)
        body = obj.get()['Body'].read()

        specs=json.loads(body)


        imageid=specs['ec2']['image-id']
        mincount=specs['ec2']['mincount']
        maxcount=specs['ec2']['maxcount']
        instancetype=specs['ec2']['instance-type']
        keyname=specs['ec2']['key-name']



    response = ec2_client.describe_vpcs()
    vpc_id = response.get('Vpcs', [{}])[0].get('VpcId', '')

    response = ec2_client.create_security_group(GroupName='serversg',
                                         Description='DESCRIPTION',
                                         VpcId=vpc_id)
    security_group_id = response['GroupId']
    print('Security Group Created %s in vpc %s.' % (security_group_id, vpc_id))
```

14) On the top right corner, select “Select a test event” and select then “Configure test events”.

15) Under event template, select s3-put, and enter the event name as “IaCtestevent”

16) Replace the test event template with the code in the file test.json and click on Create



myiac - Lambda
console.aws.amazon.com/lambda/home?region=us-east-1#/functions/myiac?tab=configuration

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.

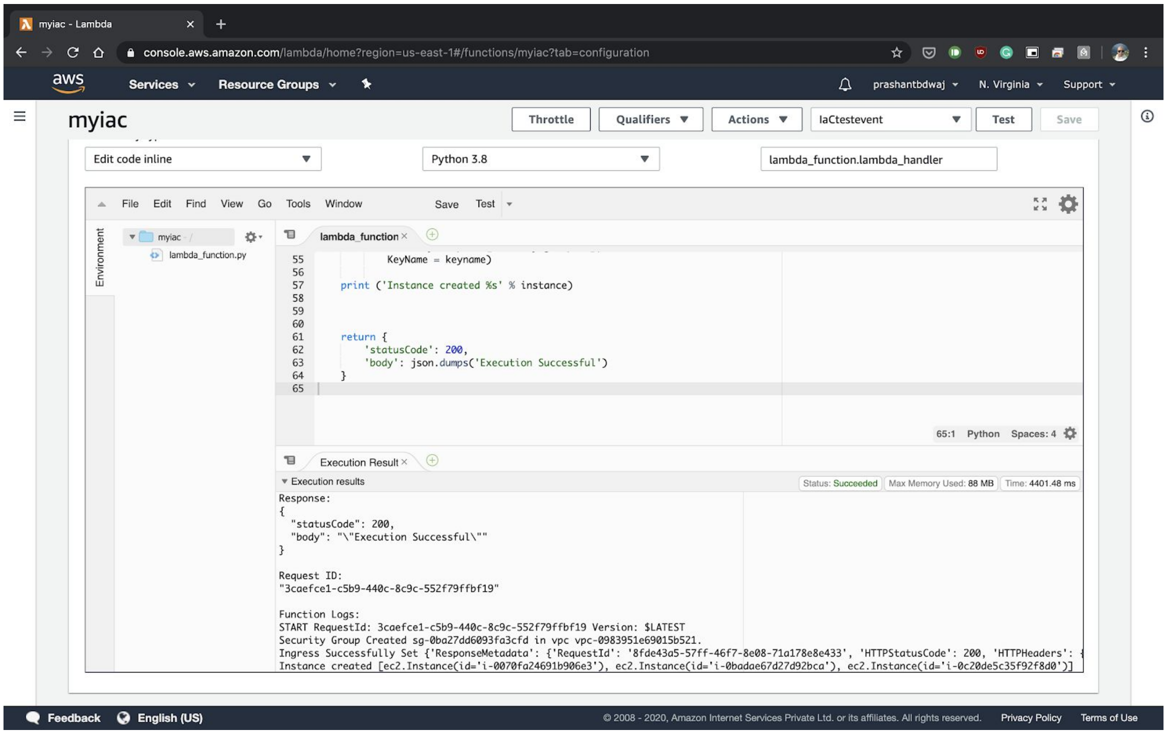
● Create new test event
○ Edit saved test events

Event template: s3-put

Event name: IaCtestevent

```
{
  "Records": [
    {
      "eventVersion": "2.0",
      "eventSource": "aws:s3",
      "awsRegion": "us-east-1",
      "eventTime": "1970-01-01T00:00:00.000Z",
      "eventName": "ObjectCreated:Put",
      "userIdentity": {
        "principalId": "EXAMPLE"
      },
      "requestParameters": {
        "sourceIPAddress": "127.0.0.1"
      },
      "responseElements": {
        "x-amz-request-id": "EXAMPLE123456789",
        "x-amz-id-2": "EXAMPLE123/5678abcdefghijklambdaisawesome/mnopqrstuvwxyzABCDEFGH
      },
      "s3": {
        "s3SchemaVersion": "1.0",
        "configurationId": "testConfigRule",
        "bucket": {
          "name": "myiaac",
          "ownerIdentity": {
            "principalId": "EXAMPLE"
          },
```

17) Click on Test in the top right corner and wait for the function to finish executing.



Edit code inline | Python 3.8 | lambda_function.lambda_handler

```
                KeyName = keyname)

    print ('Instance created %s' % instance)



    return {
        'statusCode': 200,
        'body': json.dumps('Execution Successful')
    }
```

Execution results — Status: Succeeded | Max Memory Used: 88 MB | Time: 4401.48 ms

```
Response:
{
  "statusCode": 200,
  "body": "\"Execution Successful\""
}

Request ID:
"3caefce1-c5b9-440c-8c9c-552f79ffbf19"

Function Logs:
START RequestId: 3caefce1-c5b9-440c-8c9c-552f79ffbf19 Version: $LATEST
Security Group Created sg-0ba27dd6093fa3cfd in vpc vpc-0983951e69015b521.
Ingress Successfully Set {'ResponseMetadata': {'RequestId': '8fde43a5-57ff-46f7-8e08-71a178e8e433', 'HTTPStatusCode': 200, 'HTTPHeaders': {
Instance created [ec2.Instance(id='i-0070fa24691b906e3'), ec2.Instance(id='i-0badae67d27d92bca'), ec2.Instance(id='i-0c20de5c35f92f8d0')]
```

18) Navigate to EC2 and verify that 3 new instances have been created and they all use the security group *serversg* which has ports open for SSH(22) and HTTP(80).

Bonus Objectives:

1) Open the file specifications.json and try the following tasks
    a) Modify the AMI ID to launch an Amazon Linux 2 AMI instead of Ubuntu
    b) Modify the value maxcount to launch 2 instead of 3 instances (Note: If you attempt to launch a large number of instances, say 10, the Lambda function may timeout.If this happens, simply increase the timeout value)
    c) Try using a different key-pair. The keypair should be already created before running the Lambda functions
2) Trigger the Lambda function in realtime instead of through a test event