1.Python boto3 sdk for creating bucket using API Credentials:

#!/usr/bin/python

From boto3.session import Session

session = Session(aws\_access\_key\_id=’……..’,aws\_secret\_access\_key=’………….’)

s3 = session.resource(‘s3’)

s3.create\_bucket(‘samplebucket’)

2.Creating bucket with IAM roles without using API Credentials:

#!/usr/bin/python

import boto3

import sys

Import botocore

Action = sys.argv[1]

bucket\_name=sys.argv[2]

s3\_client = boto3.client(‘s3’)

If action == ‘create-bucket’:

Print “Creating bucket”

s3\_client.create\_bucket(Bucket=bucket\_name)

3.Deleting bucket:

s3\_resource = boto3.client(‘s3’)

If action == ‘delete-bucket’:

Print “Deleting bucket”

Bucket = s3\_resource.Bucket(bucket\_name)

for key in bucket.objects.all():

key.delete()

Bucket.delete()

import boto3

#define the connection

ec2 = boto3.resource('ec2')

def lambda\_handler(event, context):

# Use the filter() method of the instances collection to retrieve

# all running EC2 instances.

filters = [{

'Name': 'instance-state-name',

'Values': ['running']

},

{'Name': 'tag:Role',

'Values': ['MyRoleTagValue']}

]

#filter the instances

instances = ec2.instances.filter(Filters=filters)

#locate all running instances

RunningInstances = [instance.id for instance in instances]

#print the instances for logging purposes

#print RunningInstances

#make sure there are actually instances to shut down.

if len(RunningInstances) > 0:

#perform the shutdown

shuttingDown = ec2.instances.filter(InstanceIds=RunningInstances).stop()

print shuttingDown

else:

print "Nothing to see here"

#!/usr/bin/python

from boto3.session import Session

session = Session(aws\_access\_key\_id='AKIAJX5#!/usr/bin/python

from boto3.session import Session

session = Session(aws\_access\_key\_id='AKIAJX5RVMXGSAK546HA',aws\_secret\_access\_key='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',region\_name='us-east-1')

ec2 = session.resource('ec2')

filters = [{

'Name': 'instance-state-name',

'Values': ['running']

},

{'Name': 'tag:Role',

'Values': ['MyRoleTagValue']}

]

#filtering all the instances

instances = ec2.instances.filter(Filters=filters)

RunningInstances = [instance.id for instance in instances]

#Making sure that there are actually instances to shut down.

if len(RunningInstances) > 0:

shuttingDown = ec2.instances.filter(InstanceIds=RunningInstances).stop()

RVMXGSAK546HA',aws\_secret\_access\_key='C/PJzABvgl+4KXFUfF6nyOE9WcXuTeN161CWkHMd',region\_name='us-east-1')

ec2 = session.resource('ec2')

filters = [{

'Name': 'instance-state-name',

'Values': ['running']

},

{'Name': 'tag:Role',

'Values': ['MyRoleTagValue']}

]

#filtering all the instances

instances = ec2.instances.filter(Filters=filters)

RunningInstances = [instance.id for instance in instances]

#Making sure that there are actually instances to shut down.

if len(RunningInstances) > 0:

shuttingDown = ec2.instances.filter(InstanceIds=RunningInstances).stop()