## **AWS Assignment 9: Configure AWS CLI, SDK and Boto**

### **Configure AWS CLI**

I have attached the scripts below which you can test

import boto3

import botocore

BUCKET\_NAME = 'my-bucket' # replace with your bucket name

KEY = 'my\_image\_in\_s3.jpg' # replace with your object key

s3 = boto3.resource('s3')

try:

s3.Bucket(BUCKET\_NAME).download\_file(KEY, 'my\_local\_image.jpg')

except botocore.exceptions.ClientError as e:

if e.response['Error']['Code'] == "404":

print("The object does not exist.")

else:

raise

import sys

import boto3

ec2 = boto3.client('ec2')

INSTANCE\_ID = sys.argv[2]

if sys.argv[1] == 'ON':

response = ec2.start\_instances(InstanceIds=[INSTANCE\_ID])

else:

response = ec2.reboot\_instances(InstanceIds=[INSTANCE\_ID])

print(response)

import boto3

import logging

import datetime

from dateutil import parser

REGION = 'ap-south-1'

ec2client = boto3.client('ec2', region\_name=REGION)

def lambda\_handler(event, context):

# TODO implement

instancelist = []

filters=[

{

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

'Values': ['running']

},

{

'Name': 'image-id',

'Values': ['ami-f3e5aa9c']

},

{

'Name': 'tag:Name',

'Values': ['DevServer']

}

]

response = ec2client.describe\_instances(Filters=filters)

for reservation in response["Reservations"]:

for instance in reservation["Instances"]:

print(instance["LaunchTime"])

launchtime = instance["LaunchTime"]

launchtime\_naive = launchtime.replace(tzinfo=None)

then = datetime.datetime.utcnow().replace(tzinfo=None)

lt\_delta = then - launchtime\_naive

if lt\_delta.days > 1:

print(instance["InstanceId"])

instancelist.append(instance["InstanceId"])

return instancelist