import boto3

import sys

from dateutil import parser

region = 'us-east-1'

def newest\_image(list\_of\_images):

latest = None

for image in list\_of\_images:

if not latest:

latest = image

continue

if parser.parse(image['CreationDate']) > parser.parse(latest['CreationDate']):

latest = image

return latest

def lambda\_handler(event, context):

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

filters = [ {

'Name': 'name',

'Values': ['amzn-ami-hvm-\*']

},{

'Name': 'description',

'Values': ['Amazon Linux AMI\*']

},{

'Name': 'architecture',

'Values': ['x86\_64']

},{

'Name': 'owner-alias',

'Values': ['amazon']

},{

'Name': 'owner-id',

'Values': ['137112412989']

},{

'Name': 'state',

'Values': ['available']

},{

'Name': 'root-device-type',

'Values': ['ebs']

},{

'Name': 'virtualization-type',

'Values': ['hvm']

},{

'Name': 'hypervisor',

'Values': ['xen']

},{

'Name': 'image-type',

'Values': ['machine']

} ]

response = client.describe\_images(Owners=['amazon'], Filters=filters)

source\_image = newest\_image(response['Images'])

return source\_image['ImageId']