**Assessment 27– Lambda**

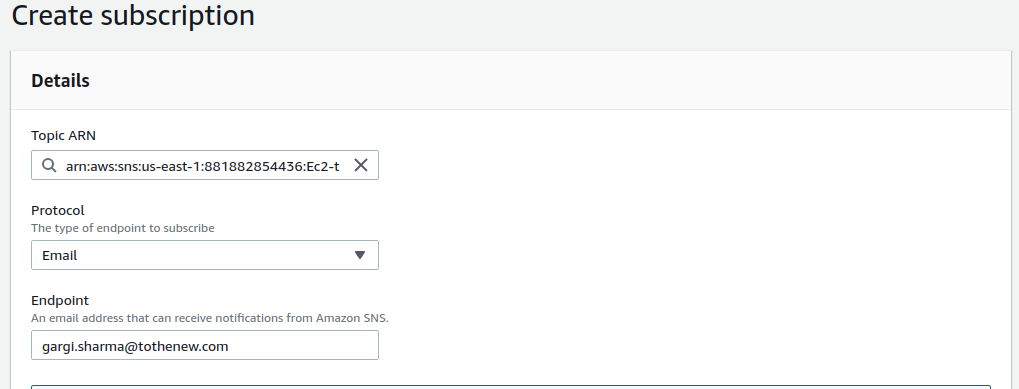
**Trainee Name : Gargi Sharma**

**Mentor Name : Mr. Akansh Gupta**

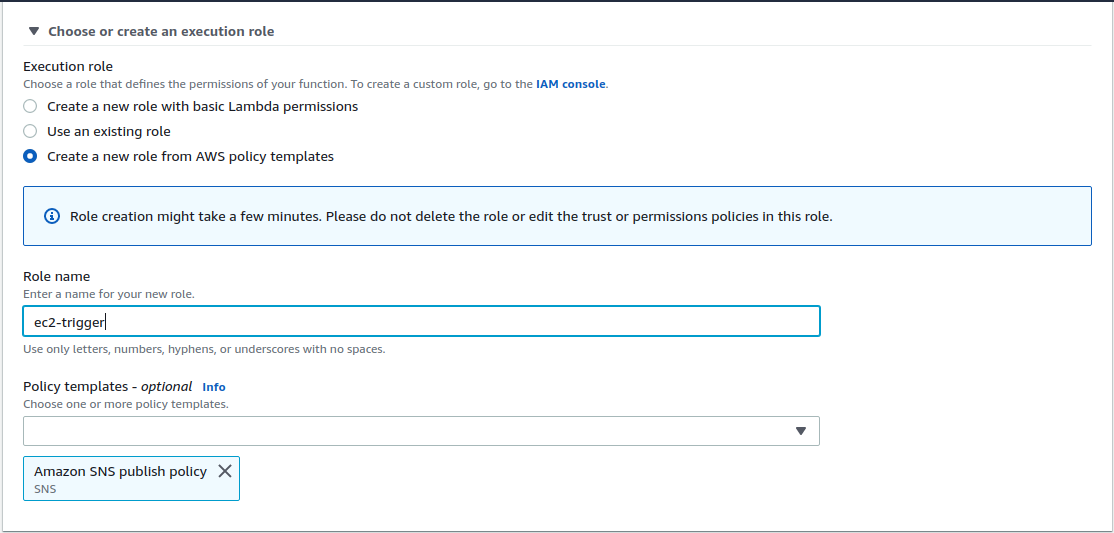
**College Name : UPES**

1. **Create a Lambda function which gets triggered from EC2 Action and Notify about changes via SNS Topic.**

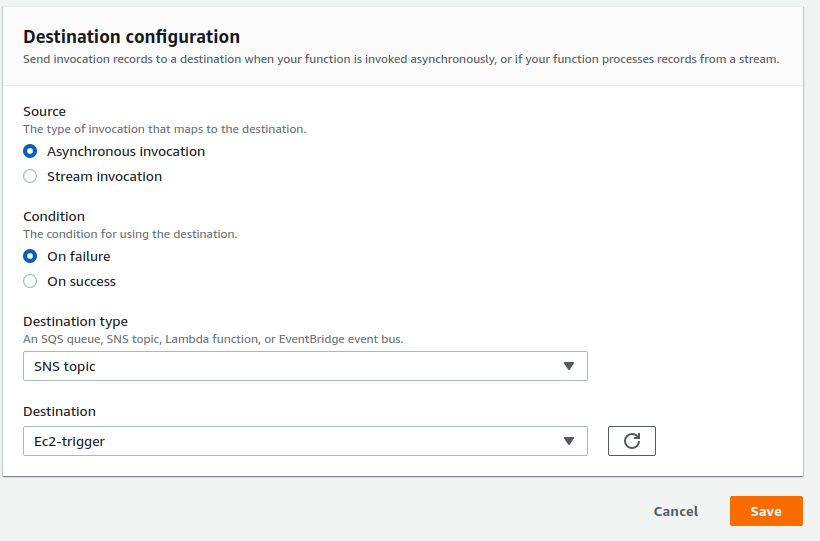
Create an SNS topic and subscribe to it:



Create a Lambda function and grant permissions to SNS. Grant SNS publish policy:

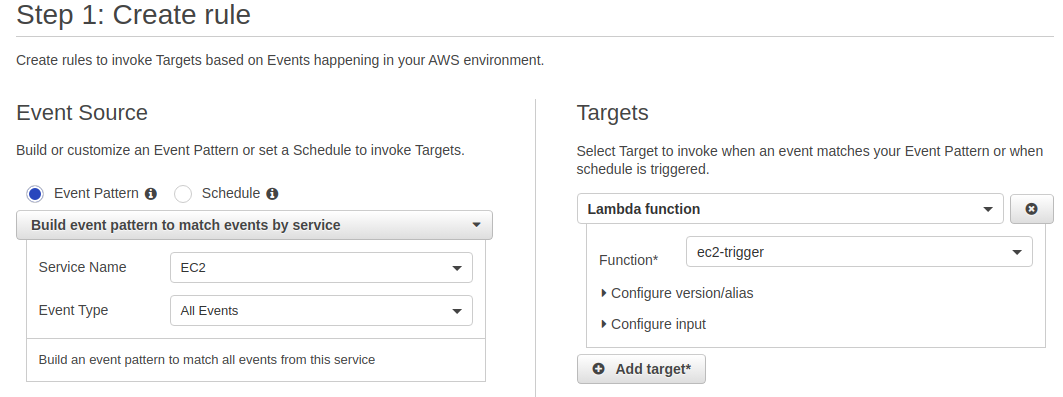


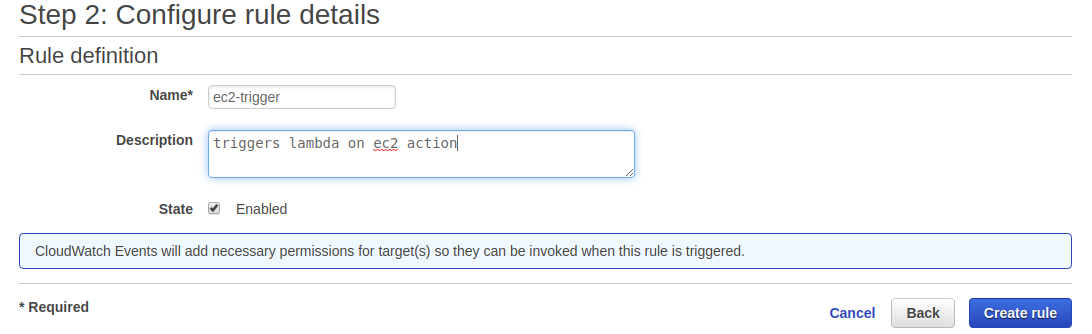
After creating lambda , go to add destination and and select your SNS topic, add 2 destinations, for failure and for success.



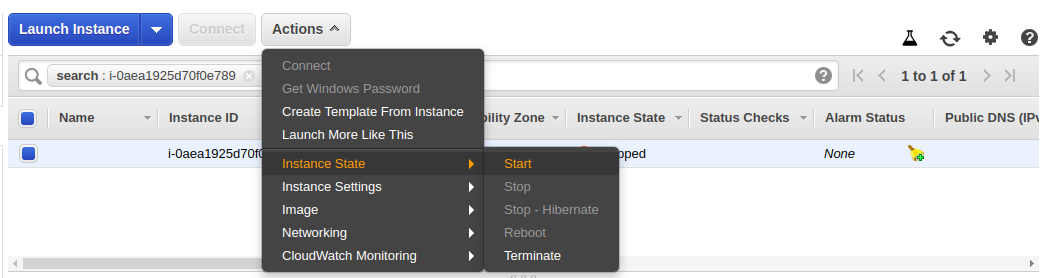


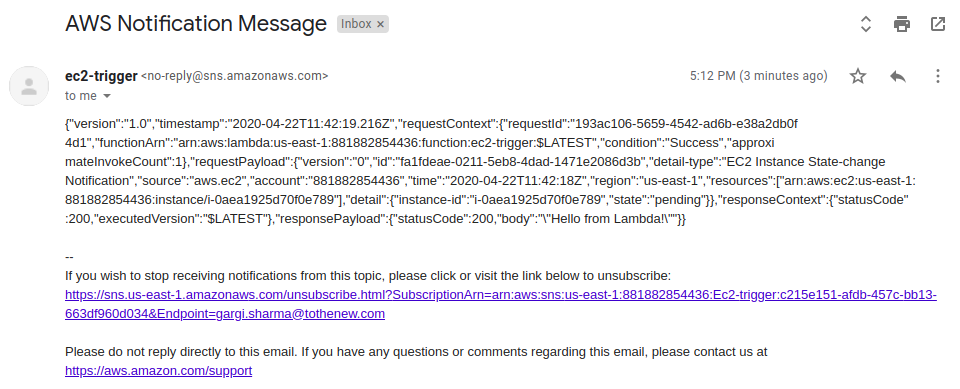
Create a cloudwatch event and add SNS notification in target:





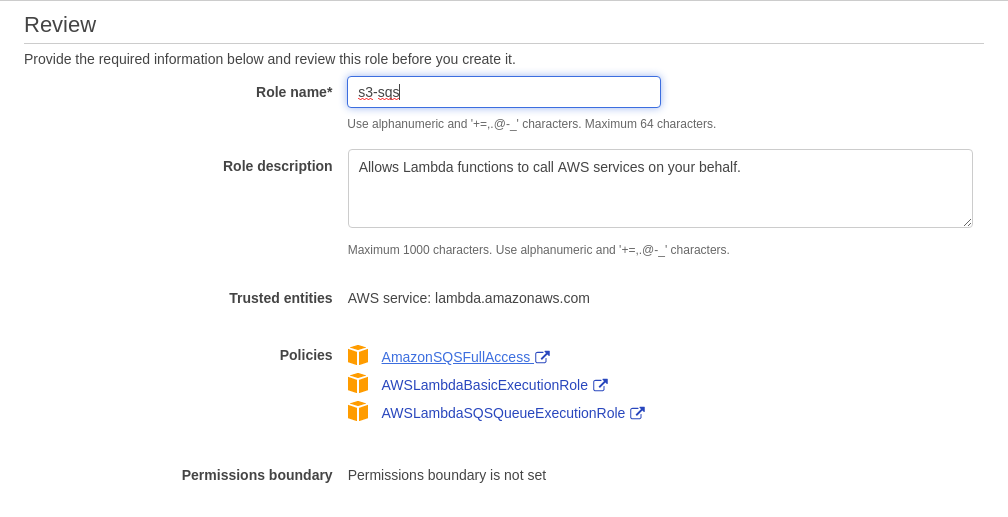
Now when we start/stop/terminate an instance , we receive a notification:



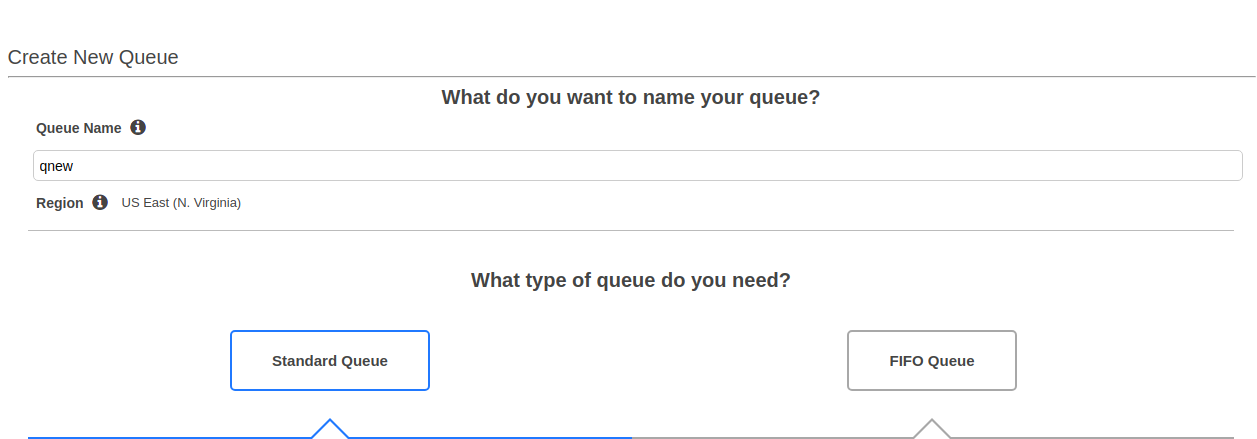


1. **Create a Lambda function which gets invoked whenever an image is added to a s3 bucket and push the key to SQS.**

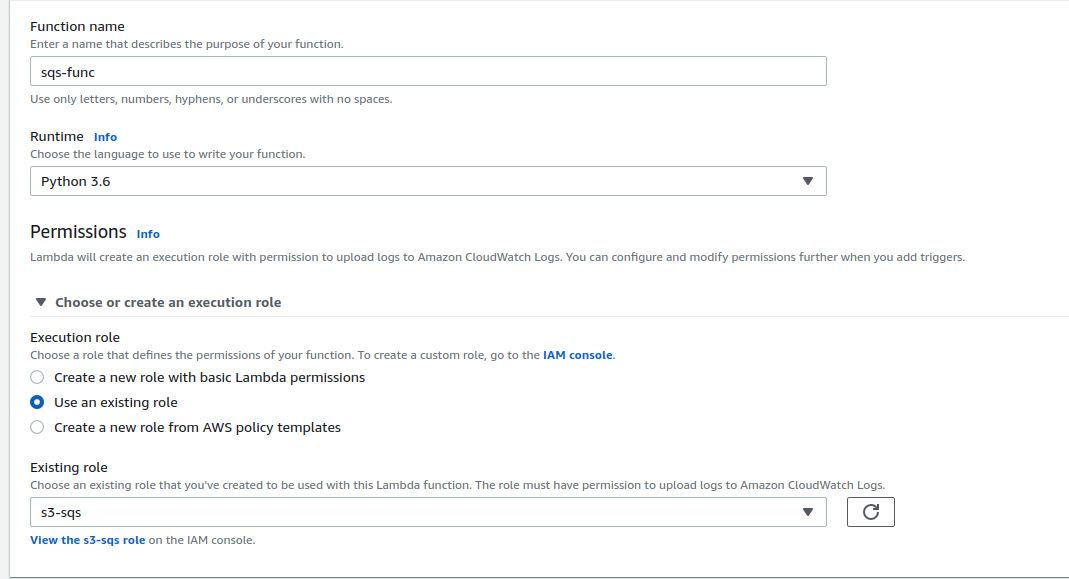
Create a role and attach the following policies:

****

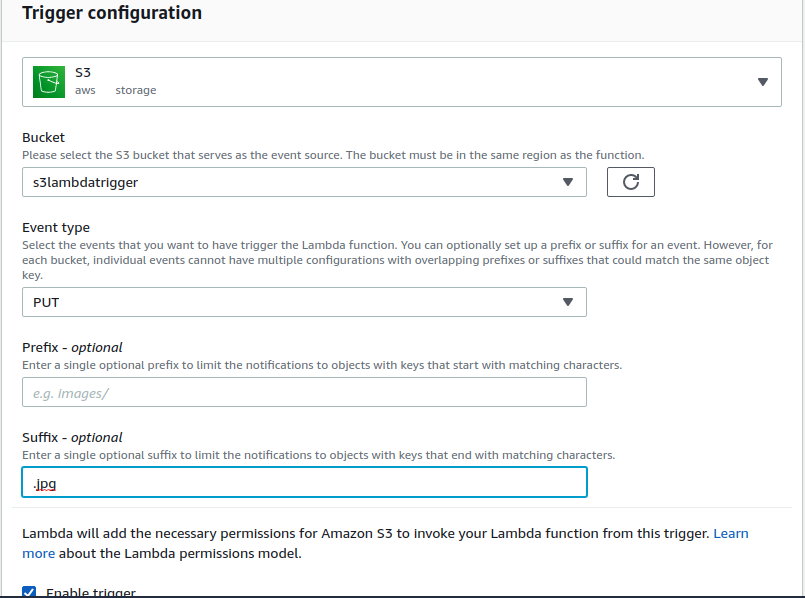
Create SQS Queue and select standard queue:

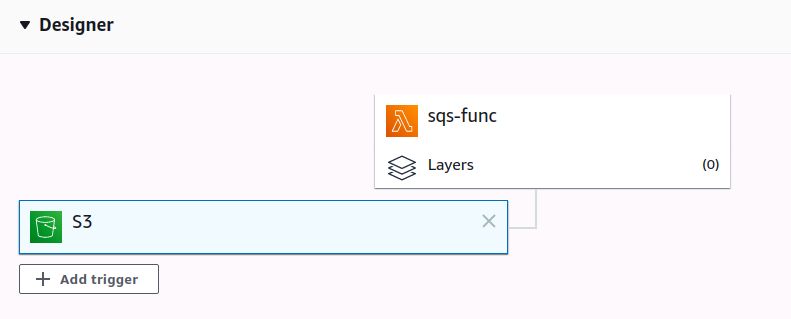
****

Now create a lambda function and attach the role:

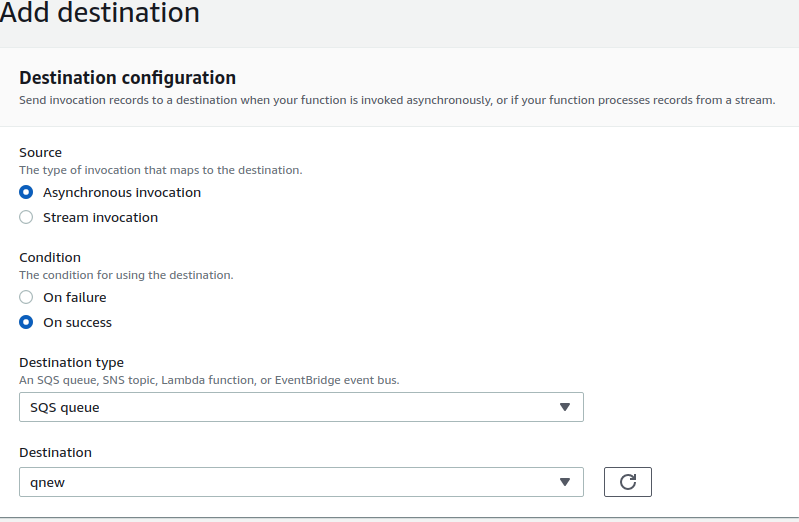
****

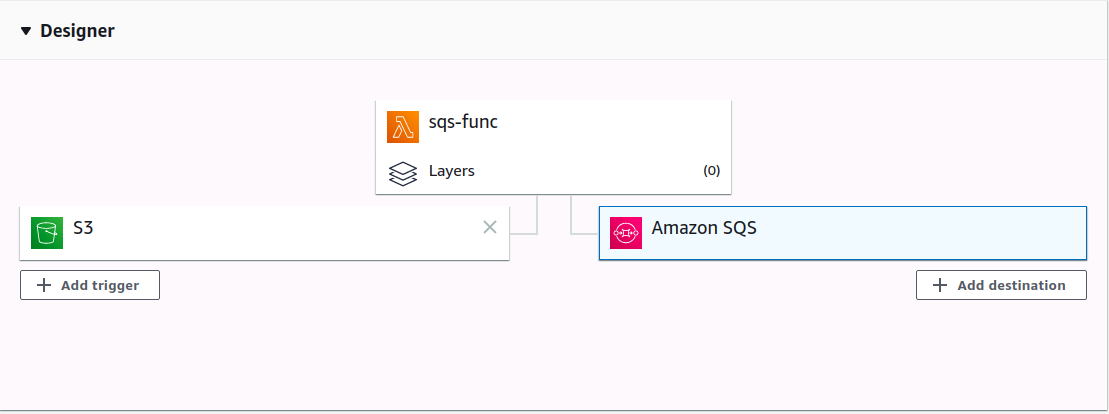
Now add trigger and select s3. Choose the s3 bucket and and select PUT operation:

****

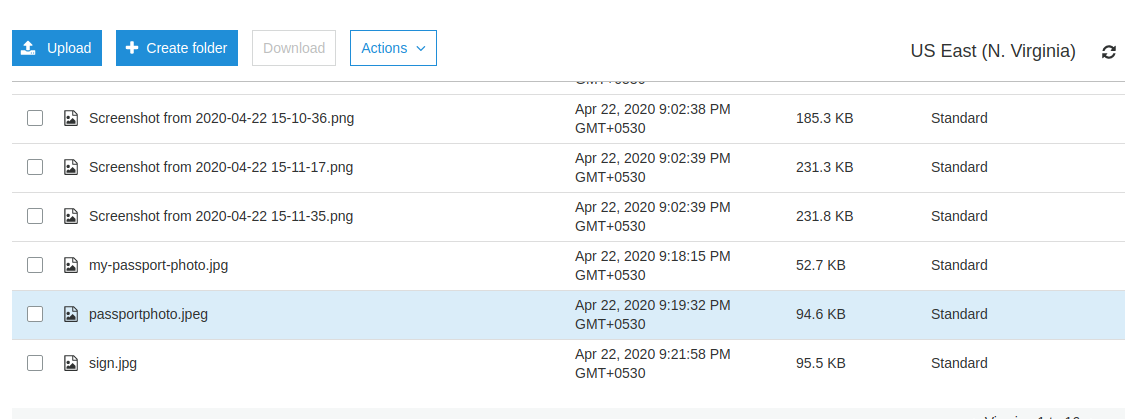
****

Add destination and select the newly made queue:

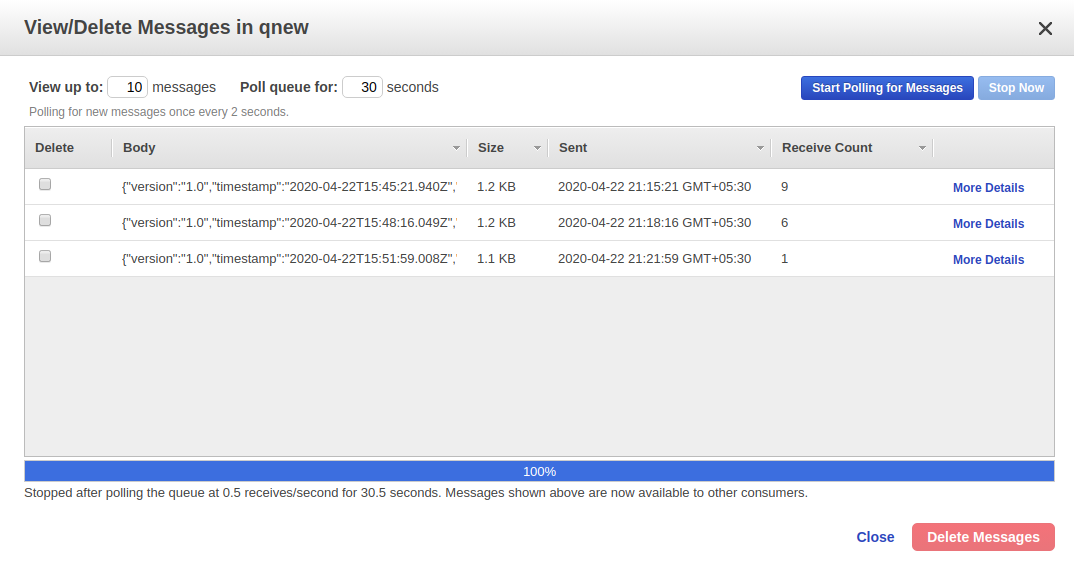
****

****

Once the setup is done, add the jpg file to s3 bucket. Then go and view messages in the queue.

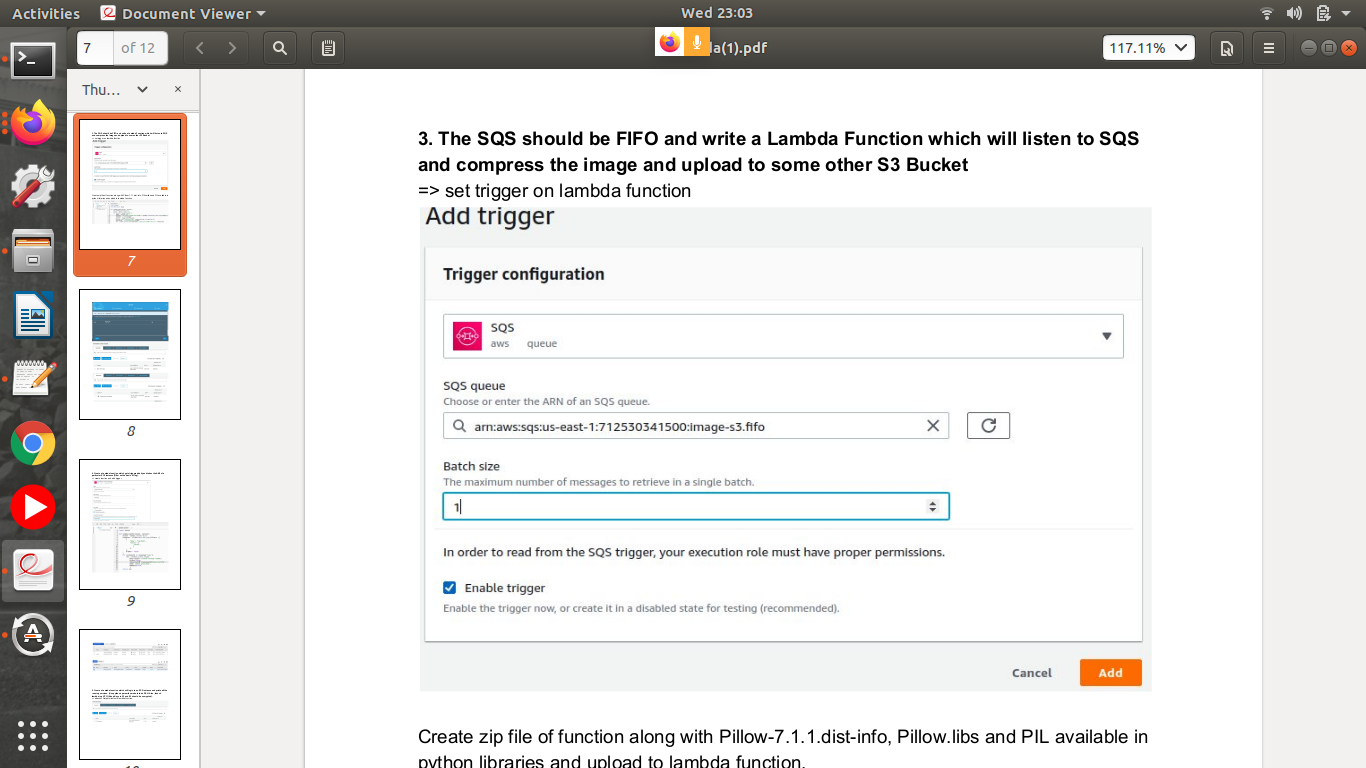
****

Start polling the messages to see the SQS Queue messages:

****

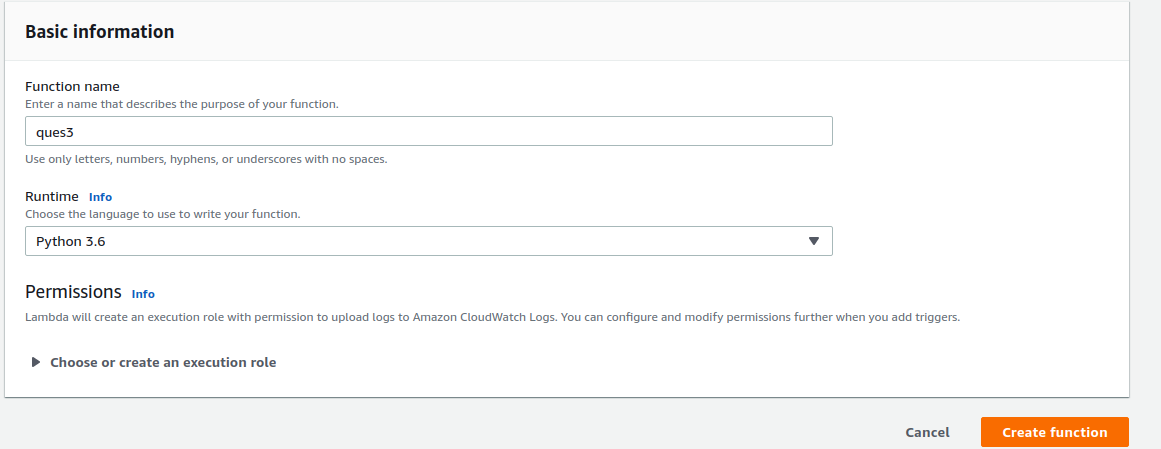
1. **The SQS should be FIFO and write a Lambda Function which will listen to SQS and compress the image and upload to some other S3 Bucket**

Set the trigger on lambda function:



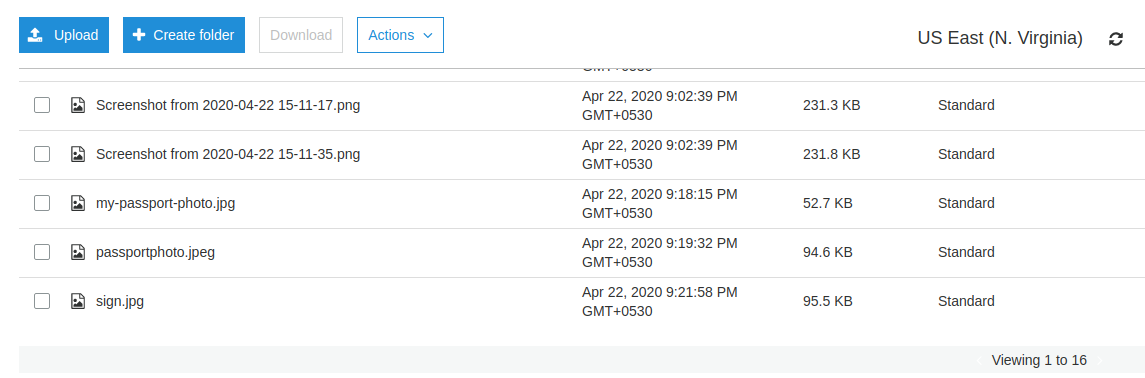
Create zip file of function along with Pillow-7.1.1.dist-info, Pillow.libs and PIL available in

python libraries and upload to lambda function.

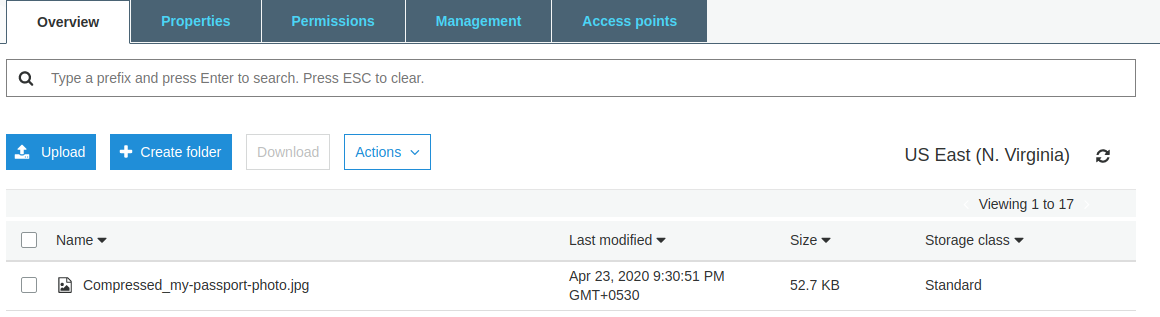




Upload a file to s3 bucket:

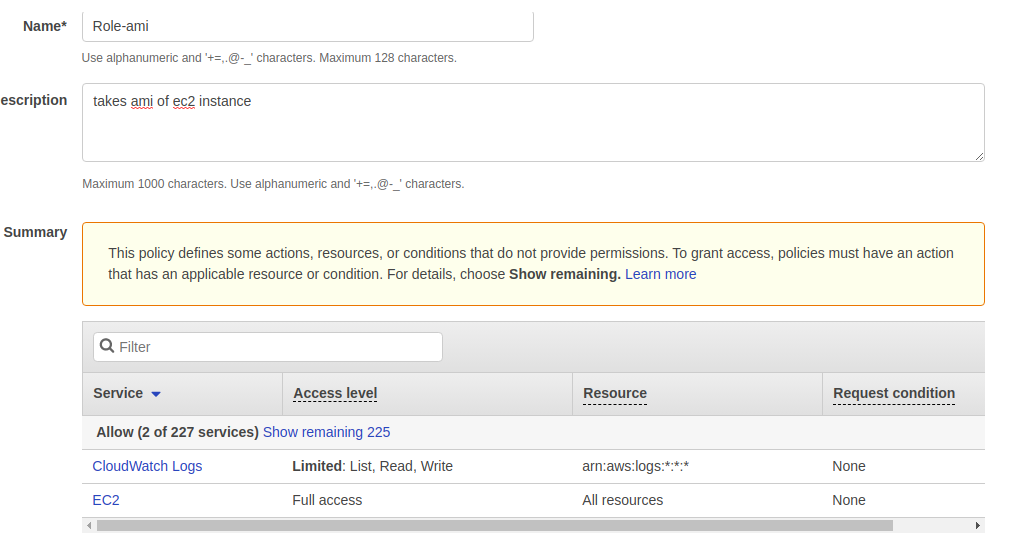


As soon as we upload an image, SQS will be triggered and file will be compressed:

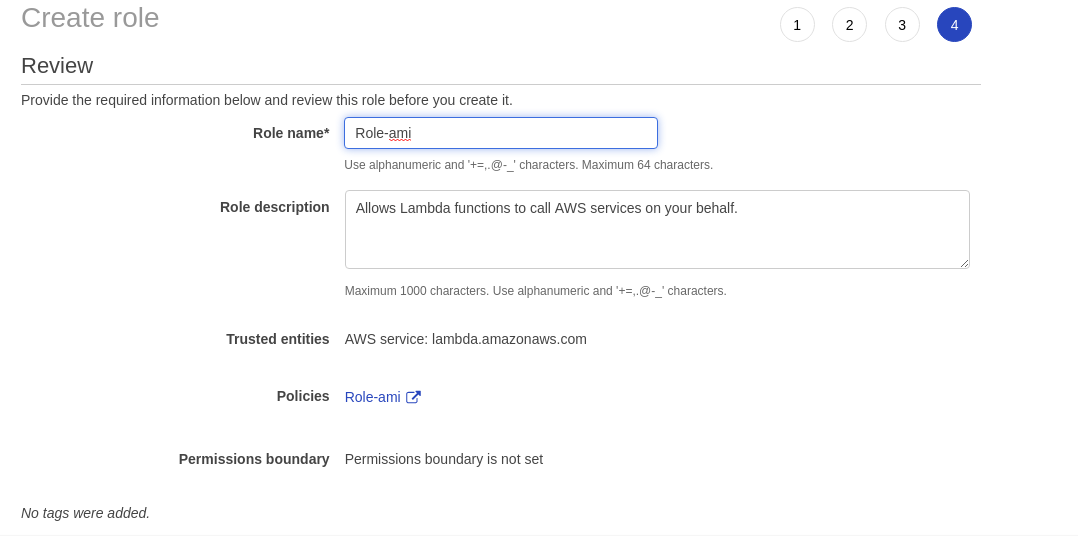


1. **Create a Lambda function which gets triggered daily and takes the AMI of a particular EC2 instance(Filter on the basis of Tag).**

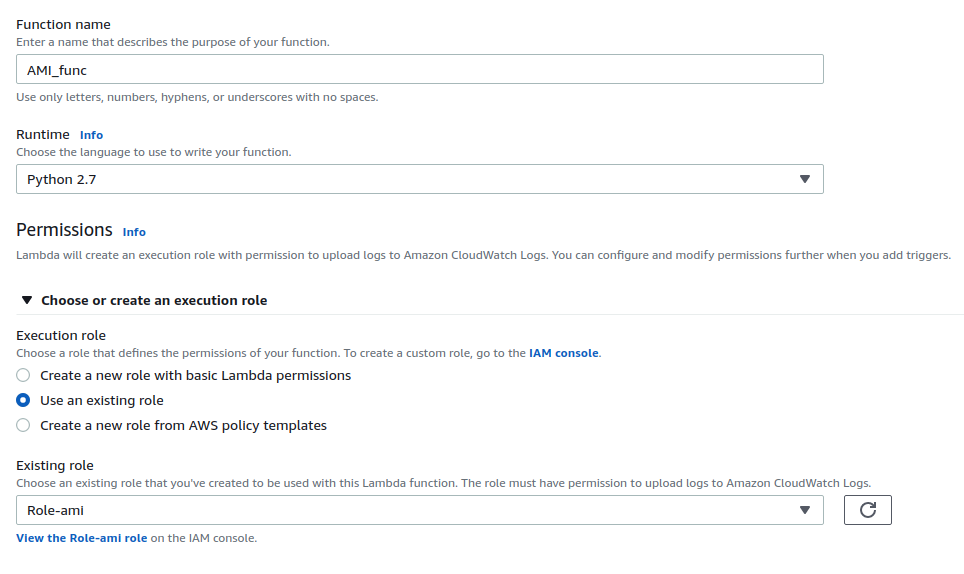
Create a policy and attach to the role:

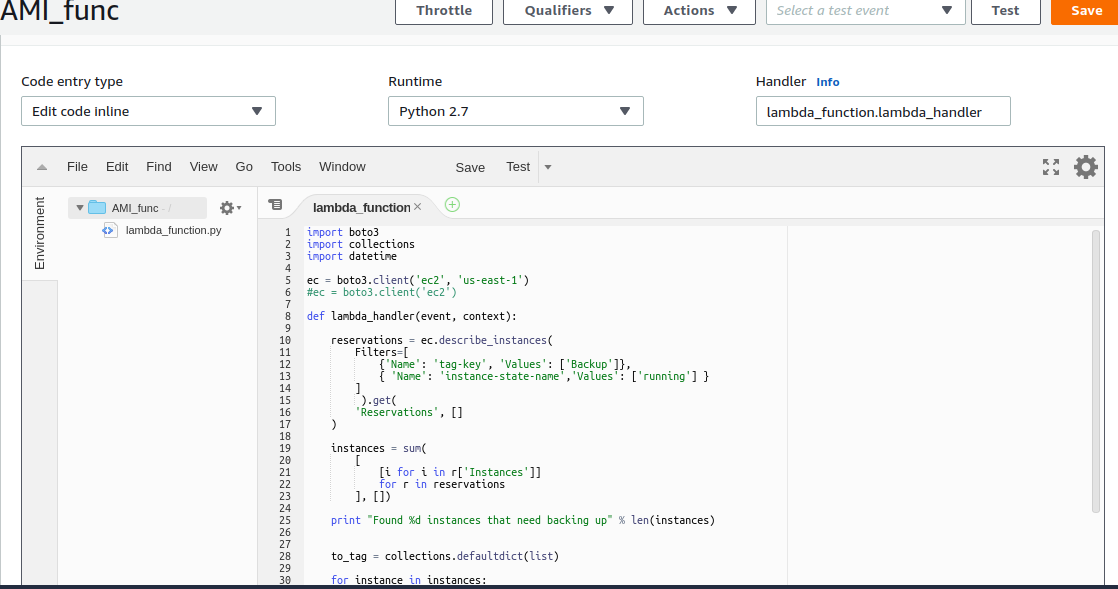


Role:

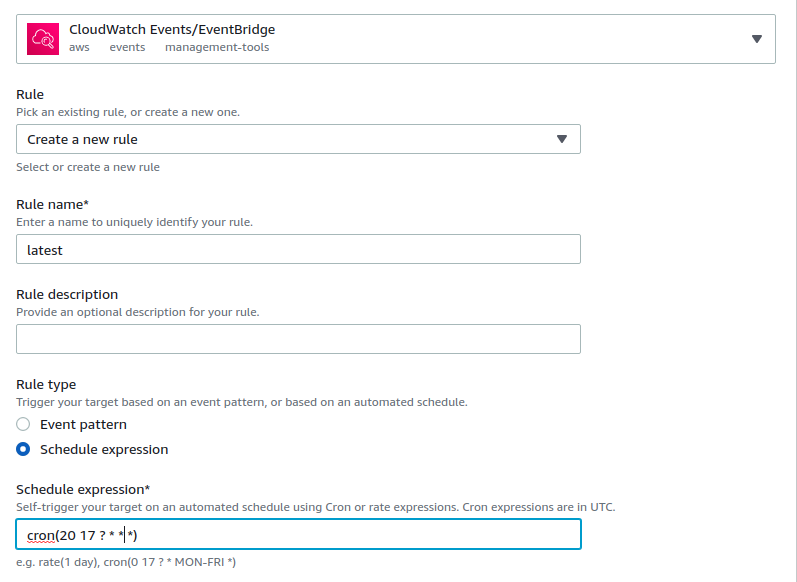


Create a lambda function:

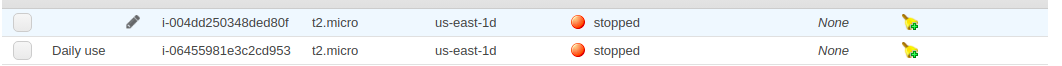




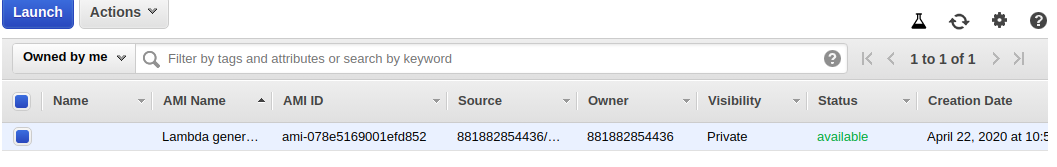
Now add trigger. Choose cloudwatch events and event bridge. Set a cron job for everyday day 8.30pm:



Edit tags of the instance and name the instance ”master”:

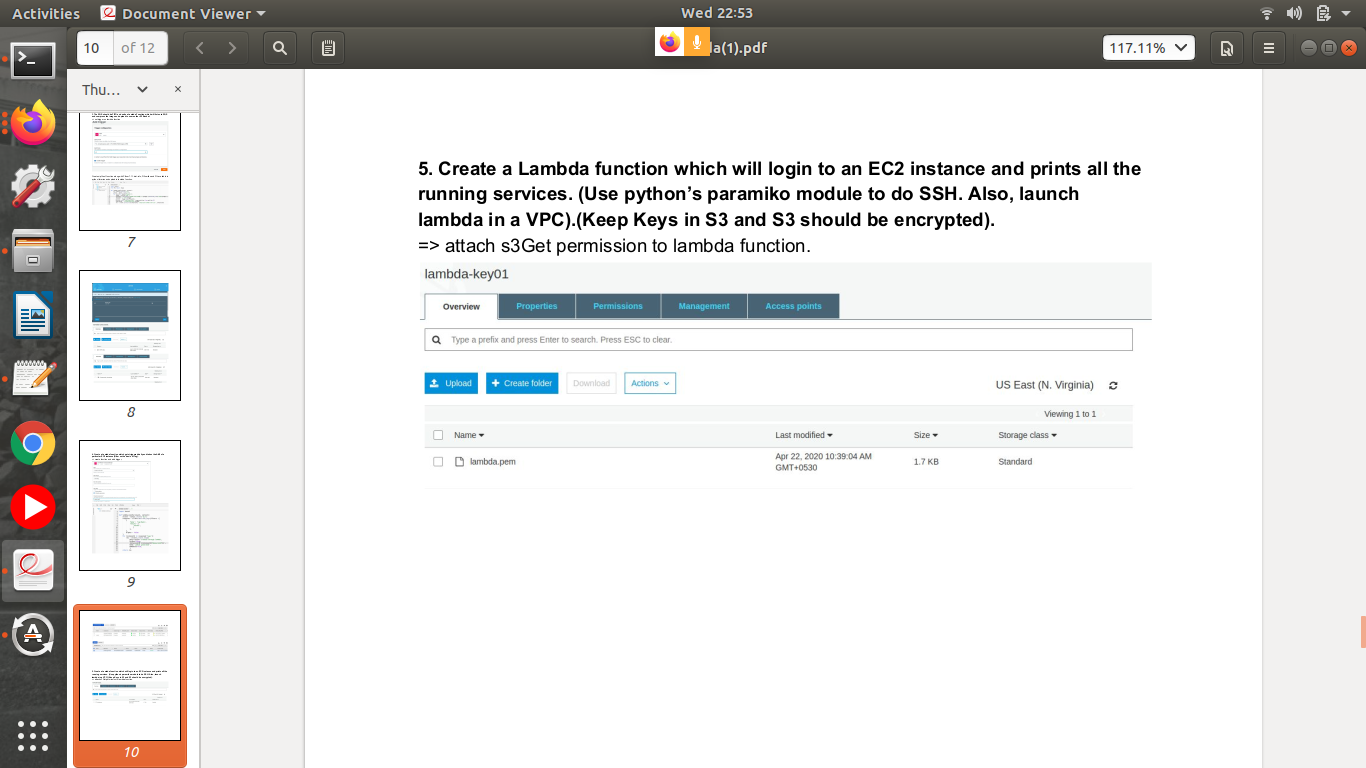


Ami created:

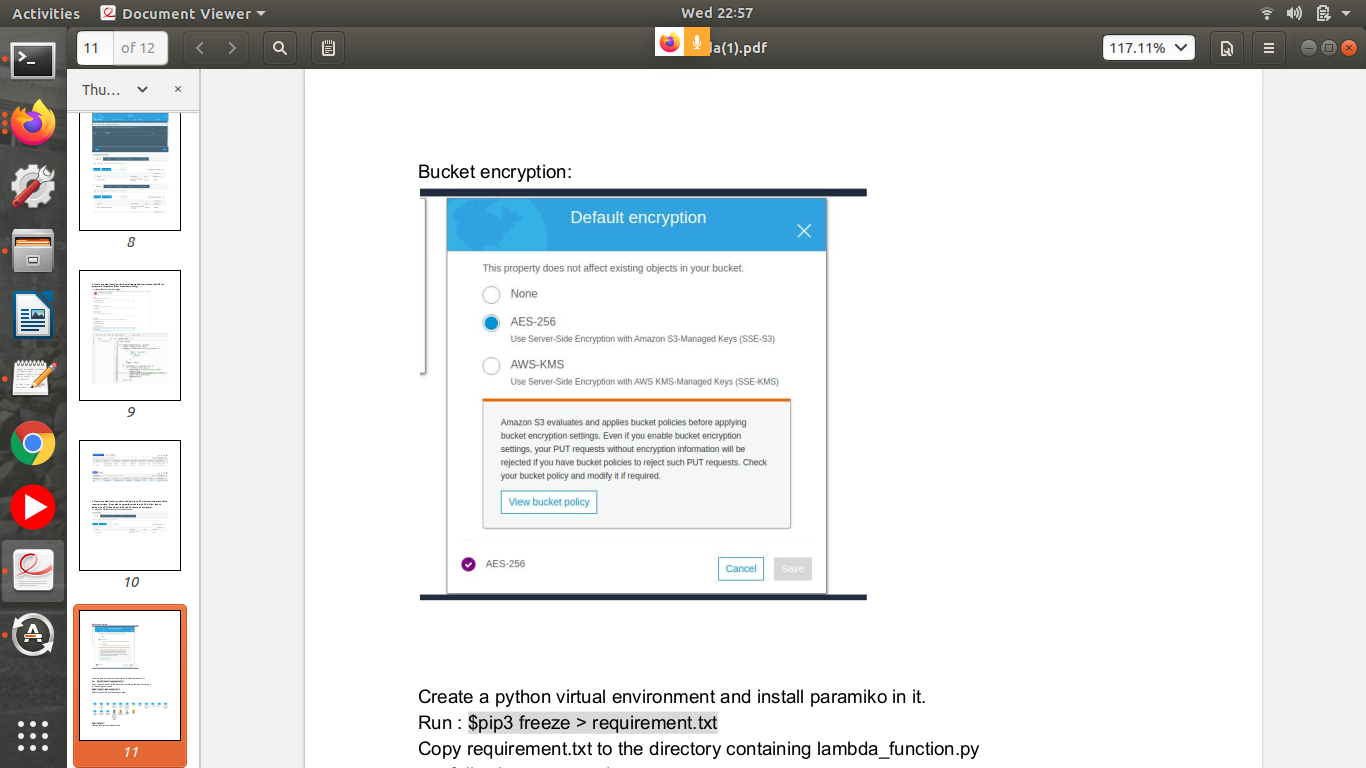


1. **Create a Lambda function which will login to an EC2 instance and prints all the running services. (Use python’s paramiko module to do SSH. Also, launch lambda in a VPC).(Keep Keys in S3 and S3 should be encrypted)**

Create a s3 bucket and give s3 permissions to lambda so that lambda can access s3. The ssh key will be kept in the bucket and retrieved by lambda to ssh into the instance.



Set bucket default encryption to AES-256:



Now create a python virtual environment and install a paramiko module on it. Python version should be 3.8

Run the following commands:

$pip3 freeze > requirement.txt

Copy requirement.txt to the directory containing lambda\_function.py

run following command.

$pip3 install -r requirement.txt -t .

$zip -r new.zip \*

Upload new.zip to aws lambda function :

