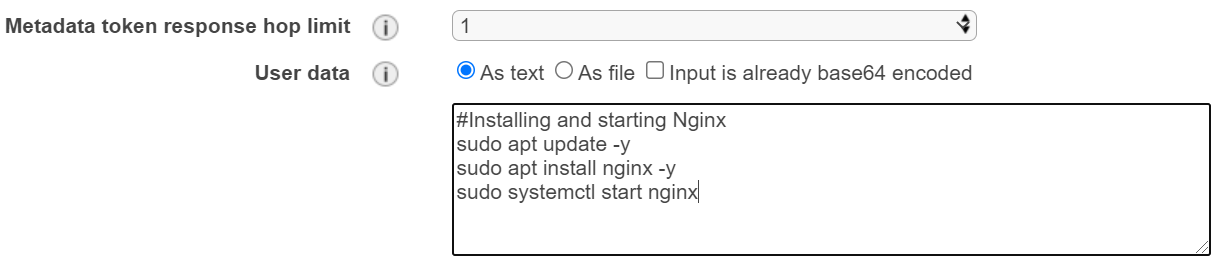
Nginx servers start and stop automation.

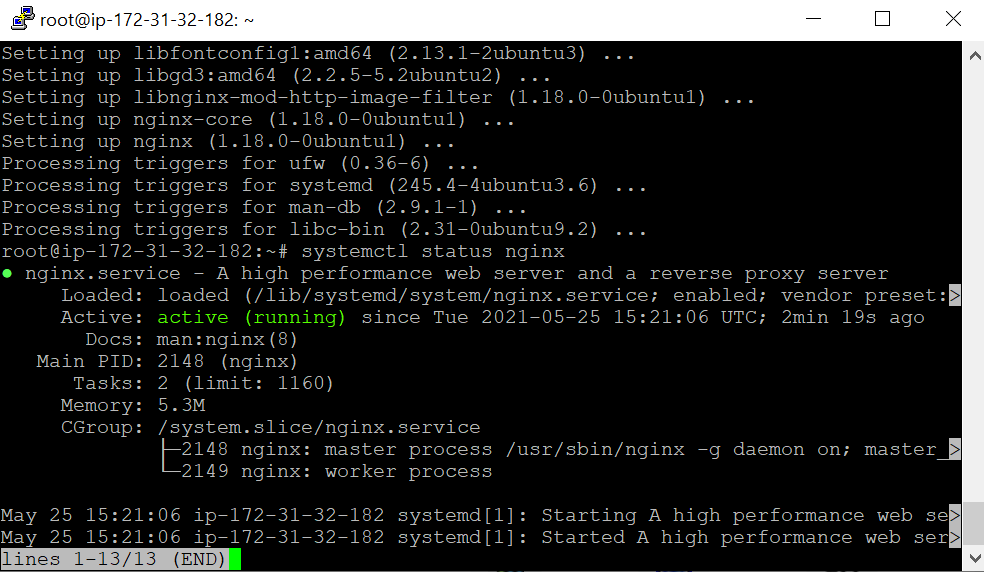
1. Launched two EC2 instance (Ubuntu) with user data given below and everything default (VPC, Subnet, Security Group) in us-east-1, us-east-2 region.

* #Installing and starting Nginx
* sudo apt update -y
* sudo apt install nginx -y
* sudo systemctl start nginx

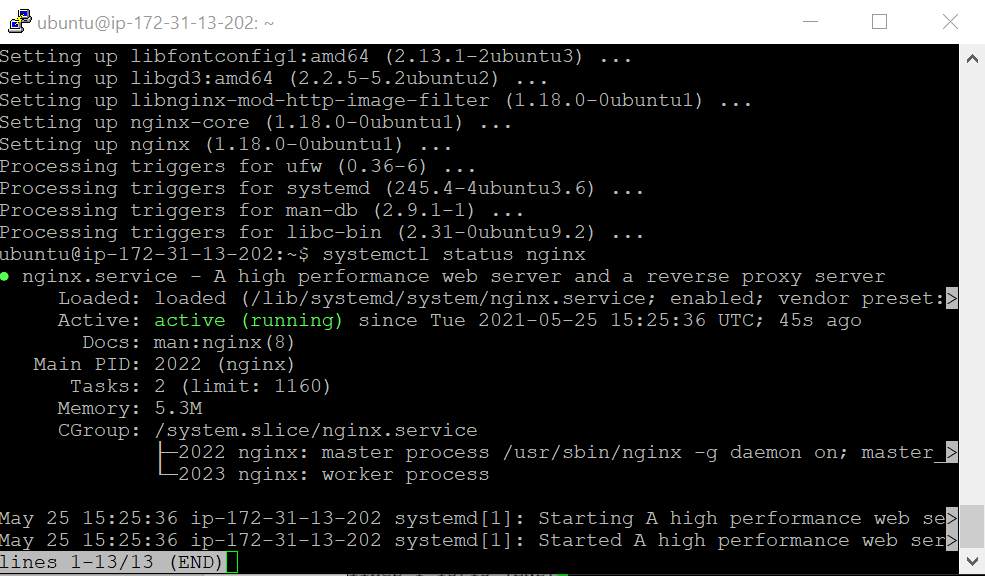
1. We can also install Nginx using same commands after deploying the servers.
   * To verify the service - sudo systemctl status nginx.
   * To configure - sudo vi /etc/nginx/nginx.conf
   * Type the server IP in browser to validate the availability.



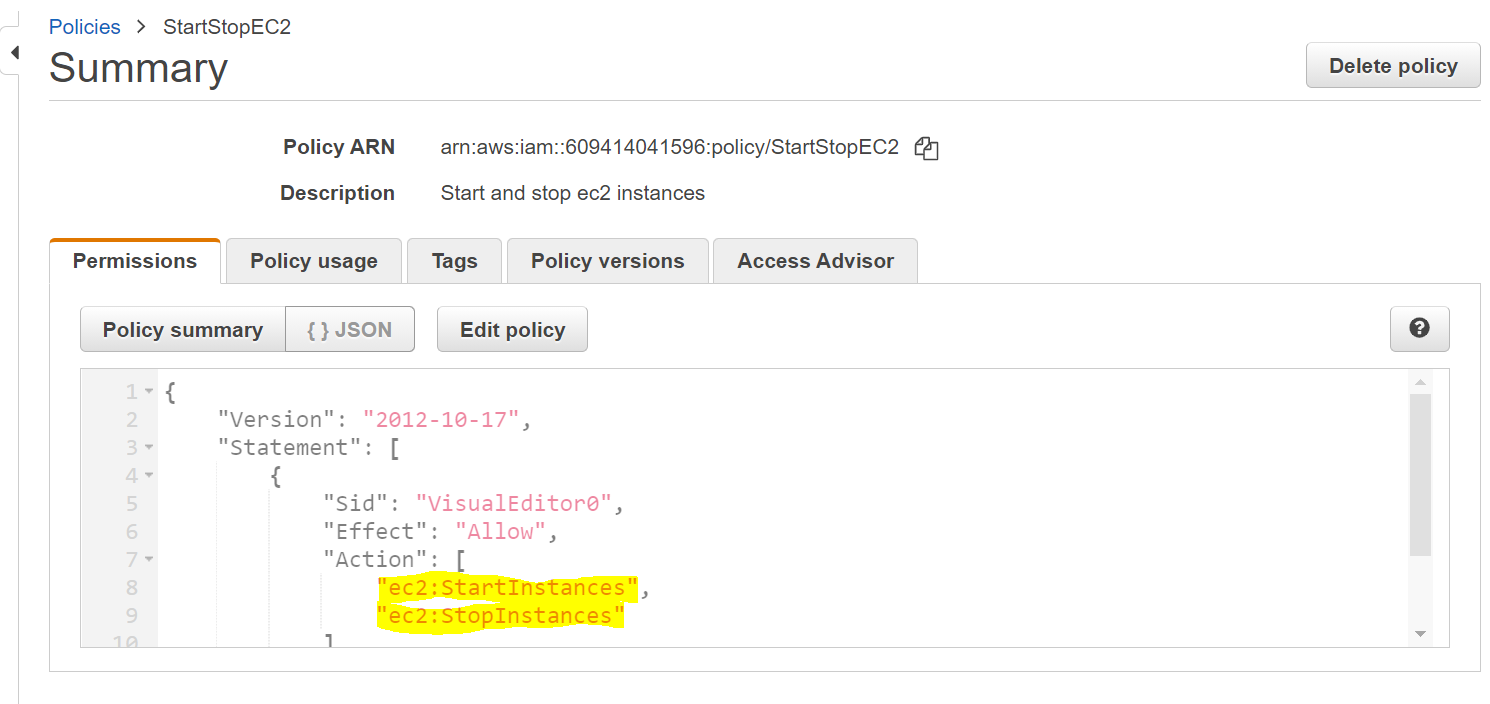
1. Server1 = SLUS1A - 18.207.165.90 (us-east-1)



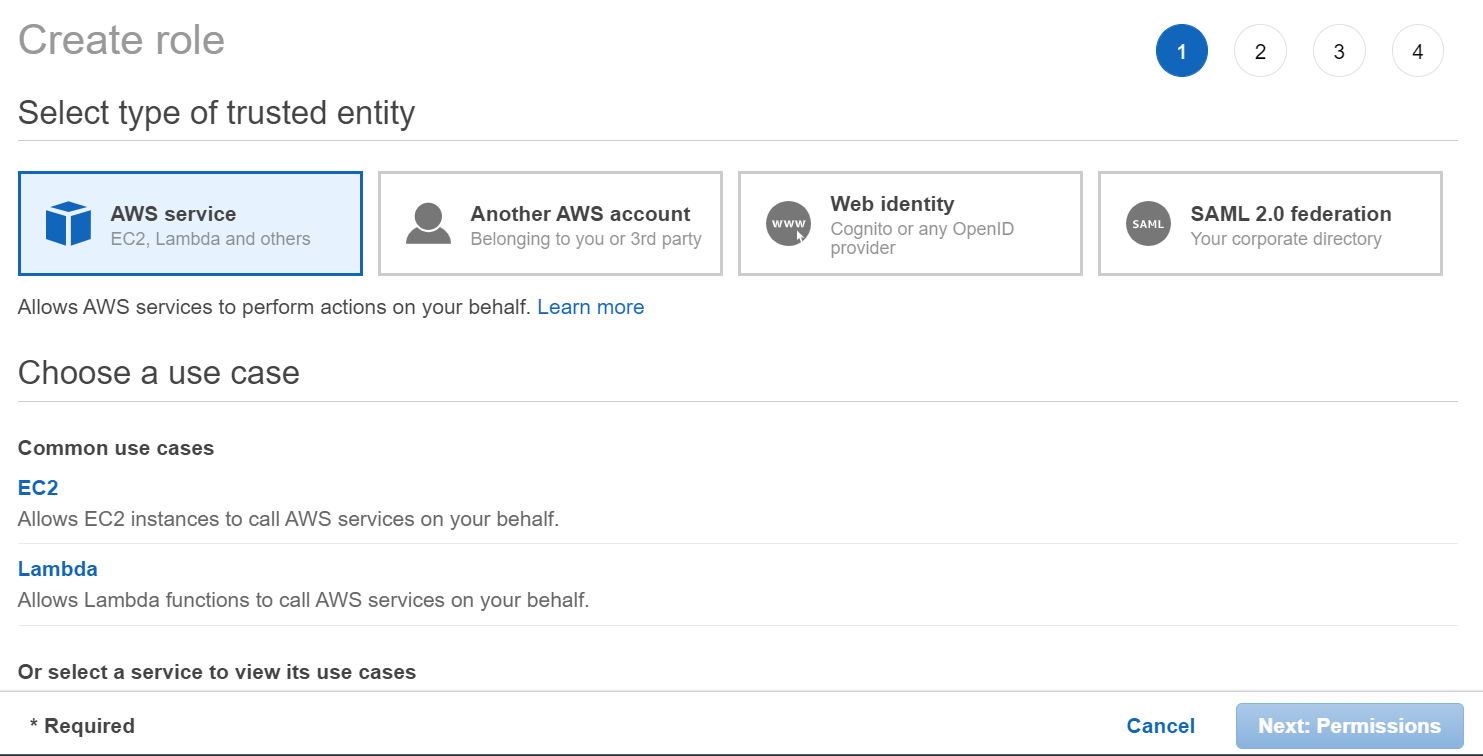
1. Server2 = SLUS1A - 18.220.245.95 (us-east-2)



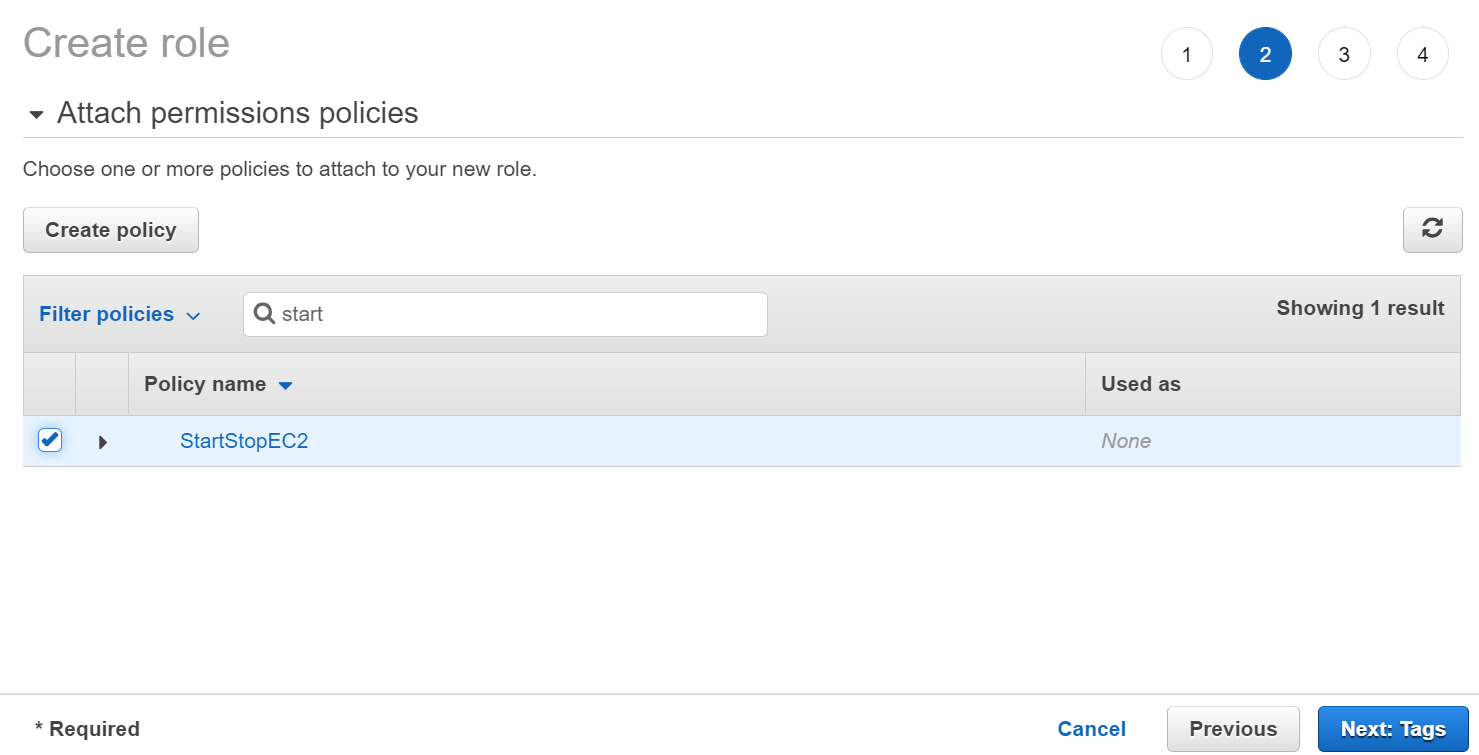
1. To start and stop the instance at the specific time we can use AWS Lambda + CloudWatch.
   * Create a policy for AWS Lambda which allows it to Stop and Start an EC2.



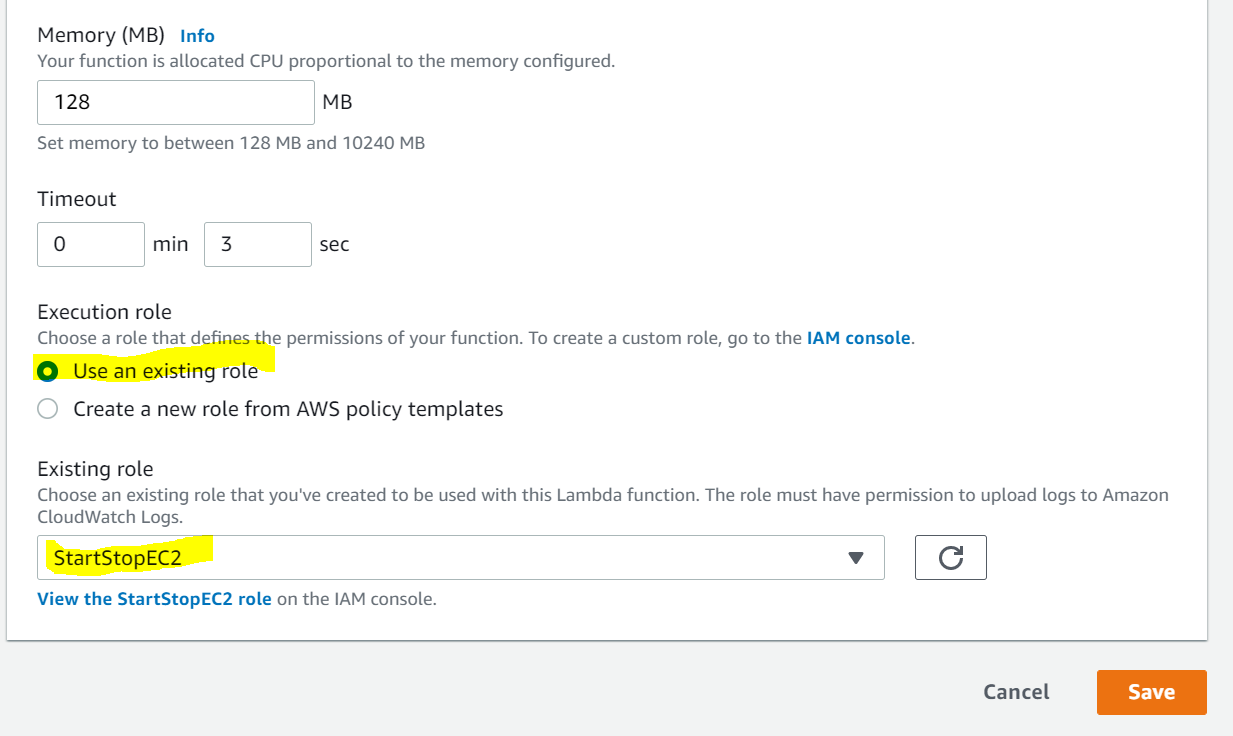
1. Create a role for AWS Lambda.



1. Assign the policy to the AWS Lambda Role which we created in previous step to Start and Stop EC2 instances.



1. Go to AWS Lambda by searching in the unified search bar > Create function > Author from scratch > Provide name > Select the program language > Select permission (Use existing role – StartStopEC2) > Edit the configuration values to suit our needs (i.e. timeout – 10 seconds).



1. Provide the below python 3.8 code and create two functions to Stop/Start the instance (Region, Instances = Update value) > Deploy code and then click on test after setting test configuration.

import boto3

region = 'us-east-1'

instances = ['i-032a4eee0b1783f84']

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

def lambda\_handler(event, context):

ec2.stop\_instances(InstanceIds=instances)

print('stopped your instances: ' + str(instances))

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

import boto3

region = 'us-east-1'

instances = ['i-032a4eee0b1783f84']

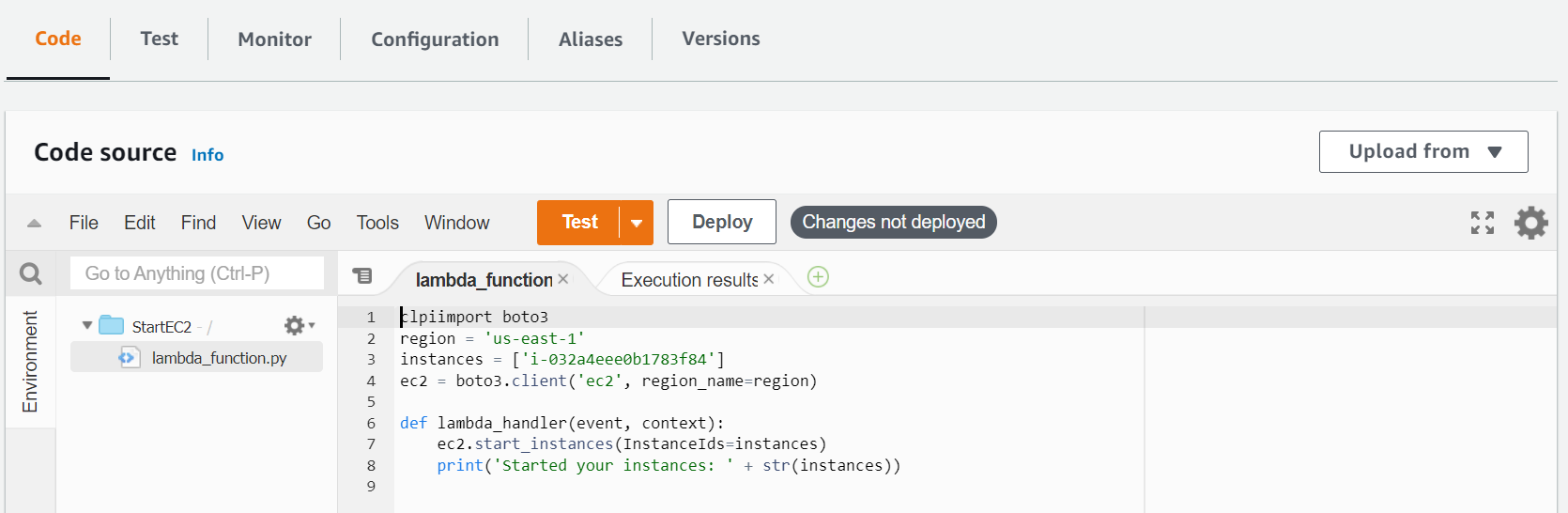
ec2 = boto3.client('ec2', region\_name=region)

def lambda\_handler(event, context):

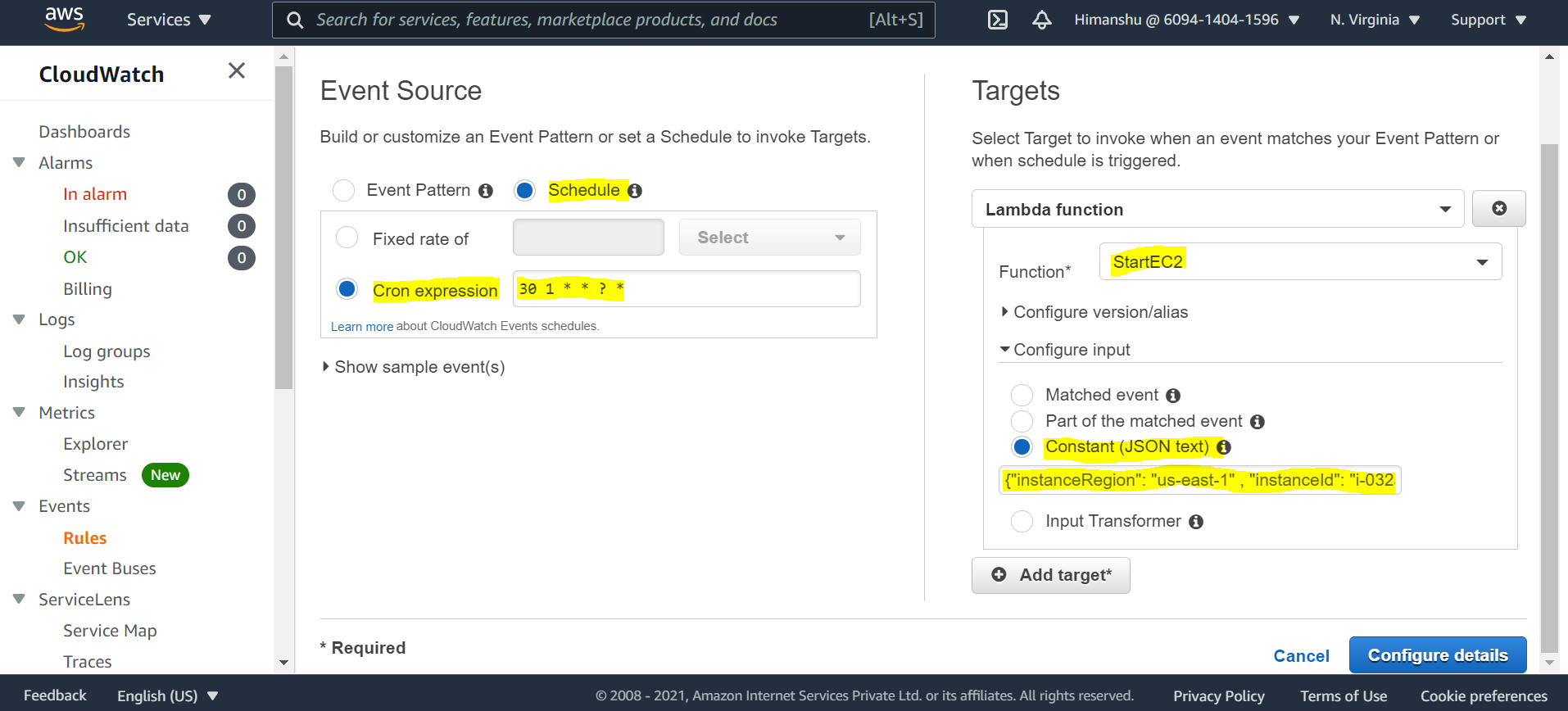
ec2.start\_instances(InstanceIds=instances)

print('started your instances: ' + str(instances))

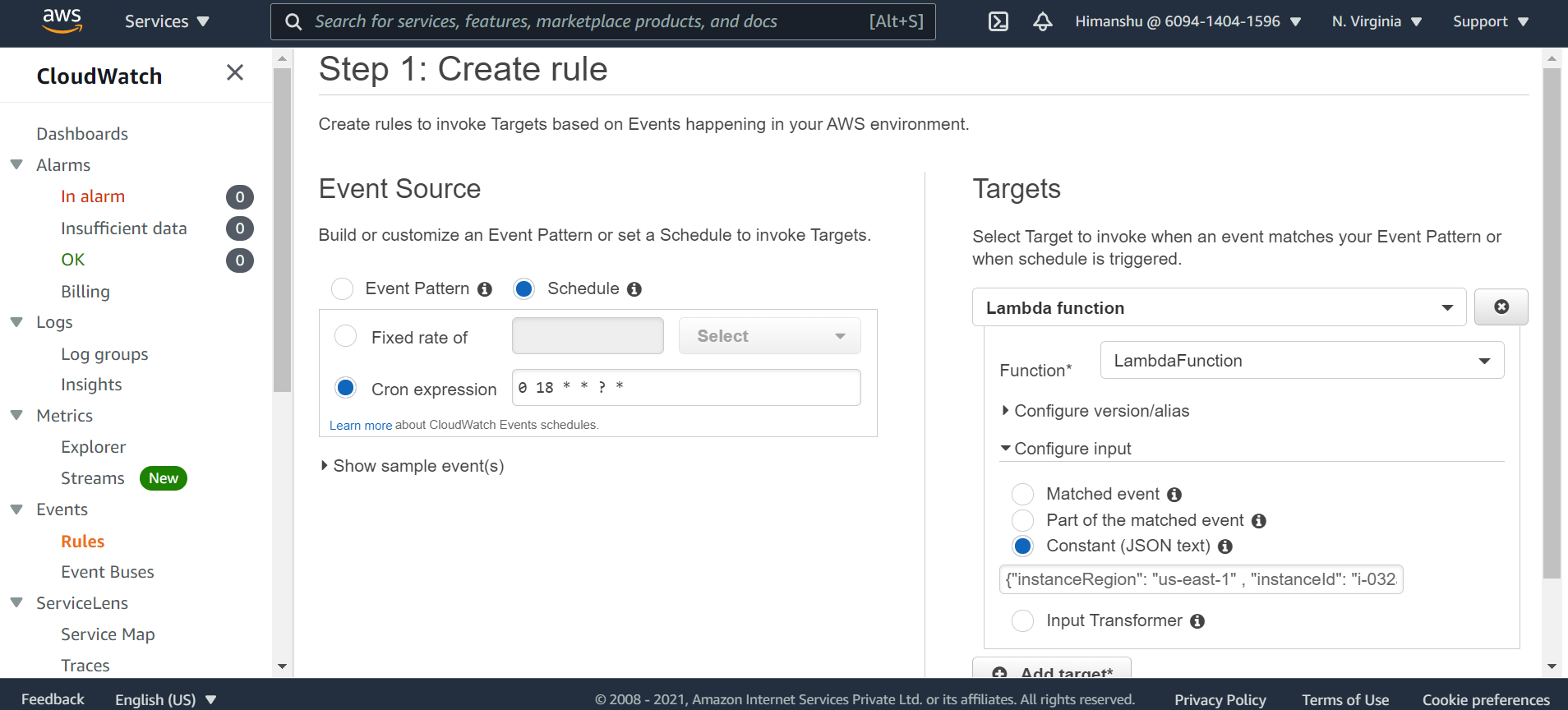
1. Provide the below python 3.8 code and create two functions to Stop/Start the instance (Region, Instances = Update value) > Deploy code and then click on test after setting test configuration.



1. Go to CloudWatch and create new rule > Create rule in CloudWatch using cron expression – 30 1 \* \* ? \* (Start Instance at 7:00 AM IST) > Pass the constant: {"instanceRegion": "us-east-1", "instanceId": "i-032a4eee0b1783f84"}



1. Create another rule > Create rule in CloudWatch using cron expression – 30 1 \* \* ? \* (Start Instance at 11:30 AM IST) > Pass the constant: {"instanceRegion": "us-east-1", "instanceId": "i-032a4eee0b1783f84"}



1. We can copy AMI from one region to another to create same server (Region peering connection is needed).
2. We can directly stop the instance using CloudWatch event rule, but we do not have an option to start from the same. Hence the Lambda approach.
3. We can deploy the same using CloudFormation.
4. <seconds> <minutes> <hours> <days of month> <months> <days of week> <years>