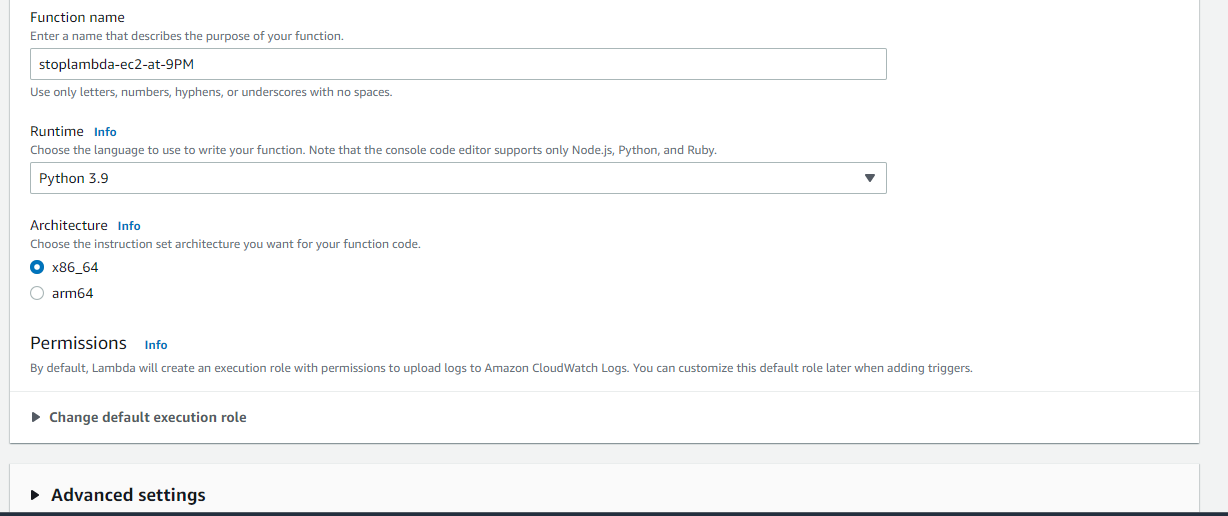
1. Create IAM role > lambda > ec2 full access > create

2. lambda > create function(STOP) > author from scratch > runtime(python 3.9 latest) >x86-64 > existing role select > create > paste scripts



import boto3

region = 'us-west-1a'

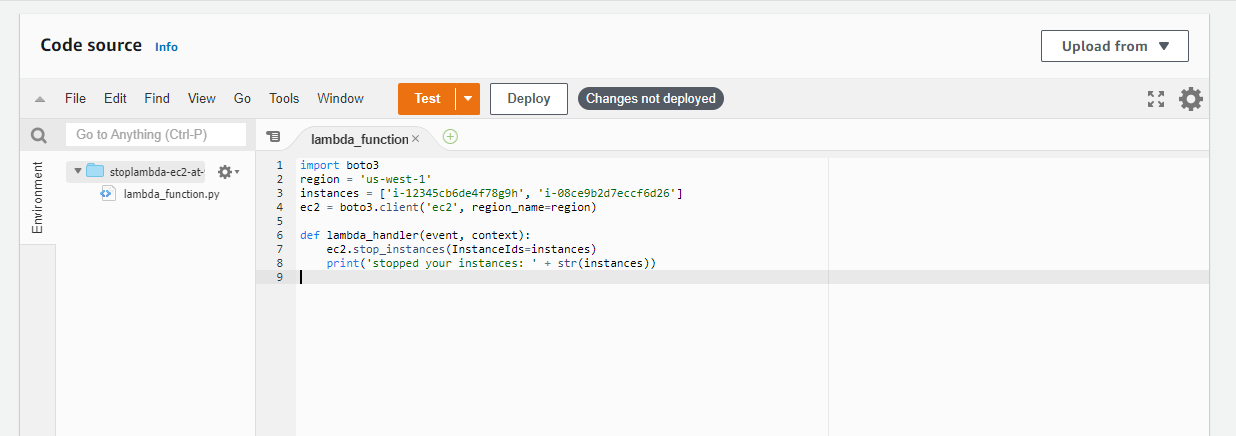
instances = ['i-12345cb6de4f78g9h', 'i-08ce9b2d7eccf6d26']

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

def lambda\_handler(event, context):

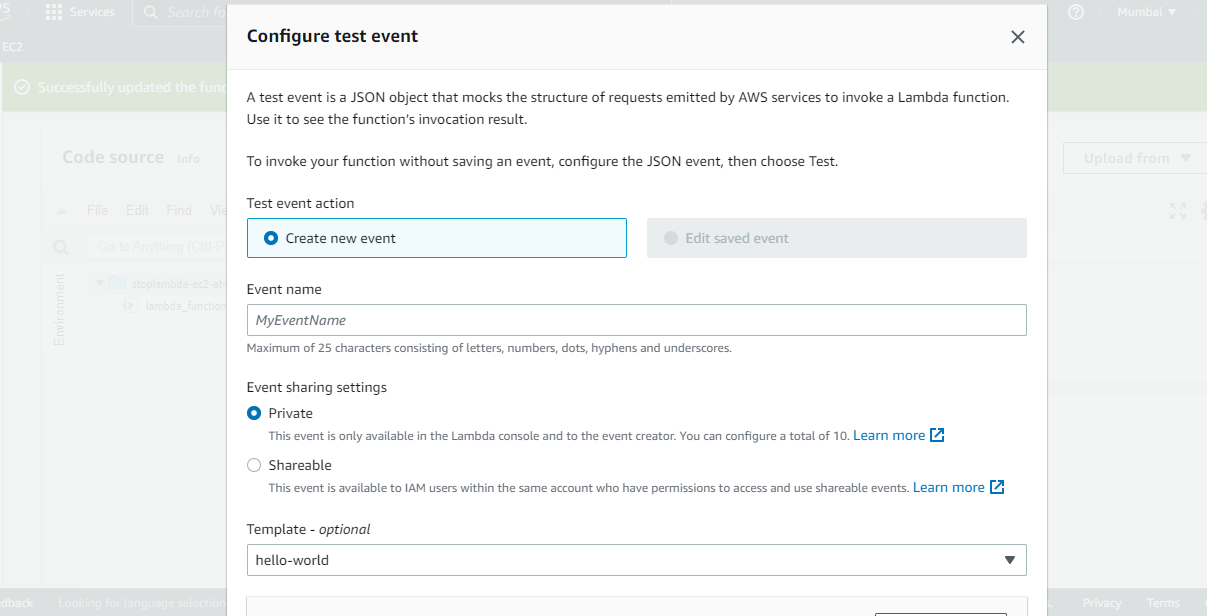
ec2.stop\_instances(InstanceIds=instances)

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

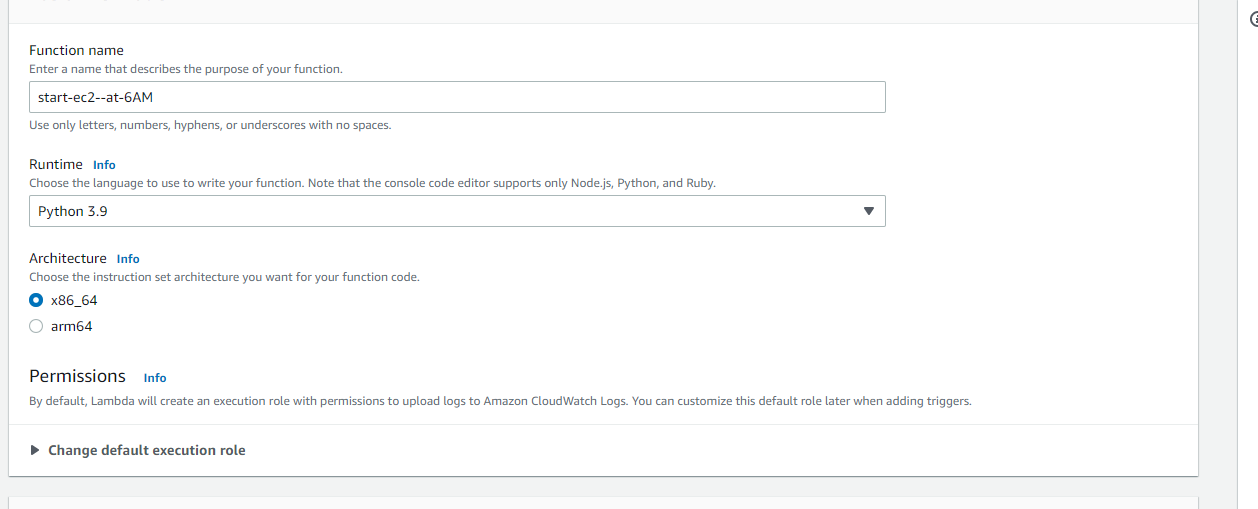


//change region & instance id only//

Then deploy & test > create new rule > private > save > test again (instance will be stopped)



lambda > create function (START)> author from scratch > runtime(python 3.9 latest) >x86-64 > existing role select > create > paste scripts



import boto3

region = 'us-west-1a'

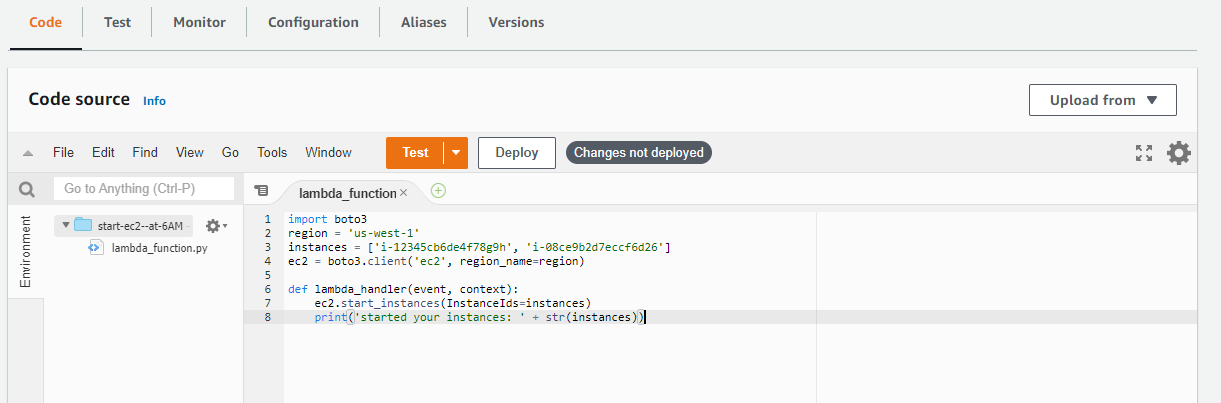
instances = ['i-12345cb6de4f78g9h', 'i-08ce9b2d7eccf6d26']

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

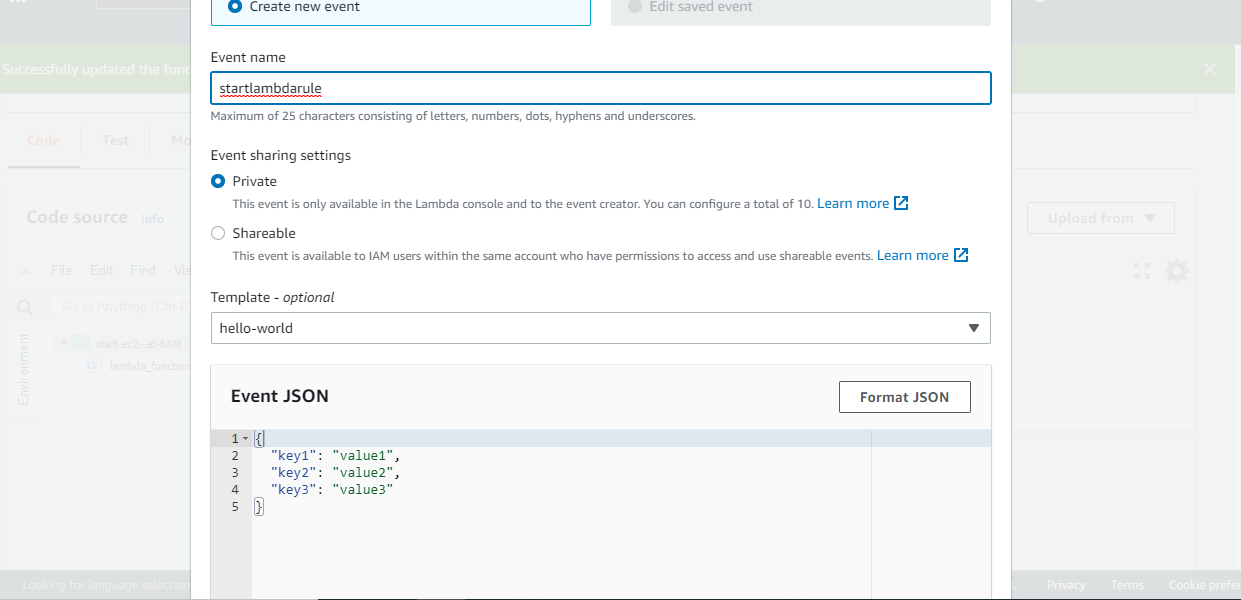
def lambda\_handler(event, context):

ec2.start\_instances(InstanceIds=instances)

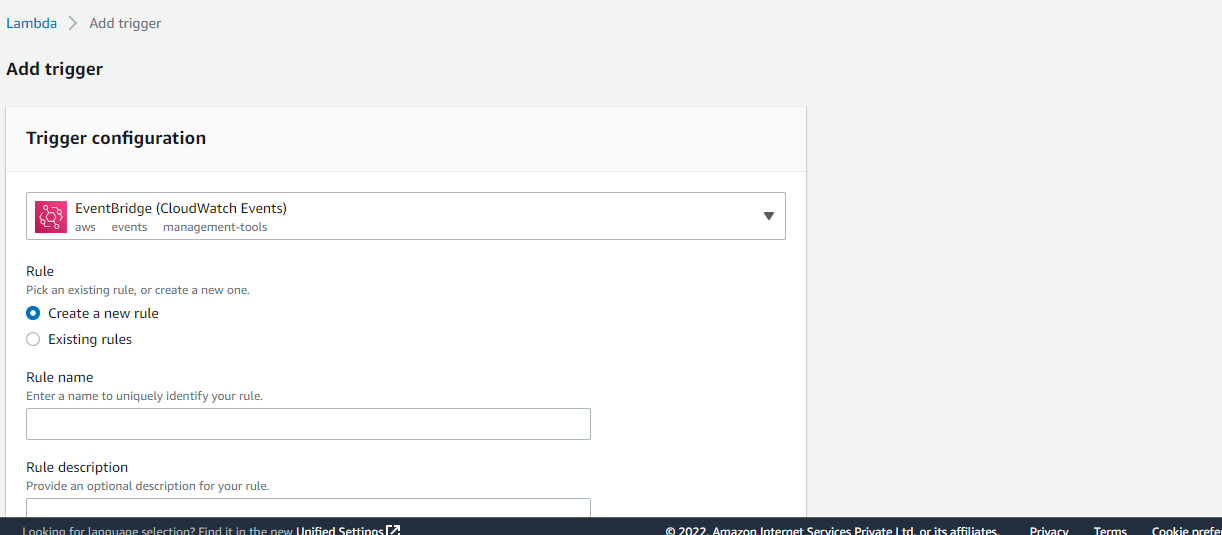
print('started your instances: ' + str(instances))

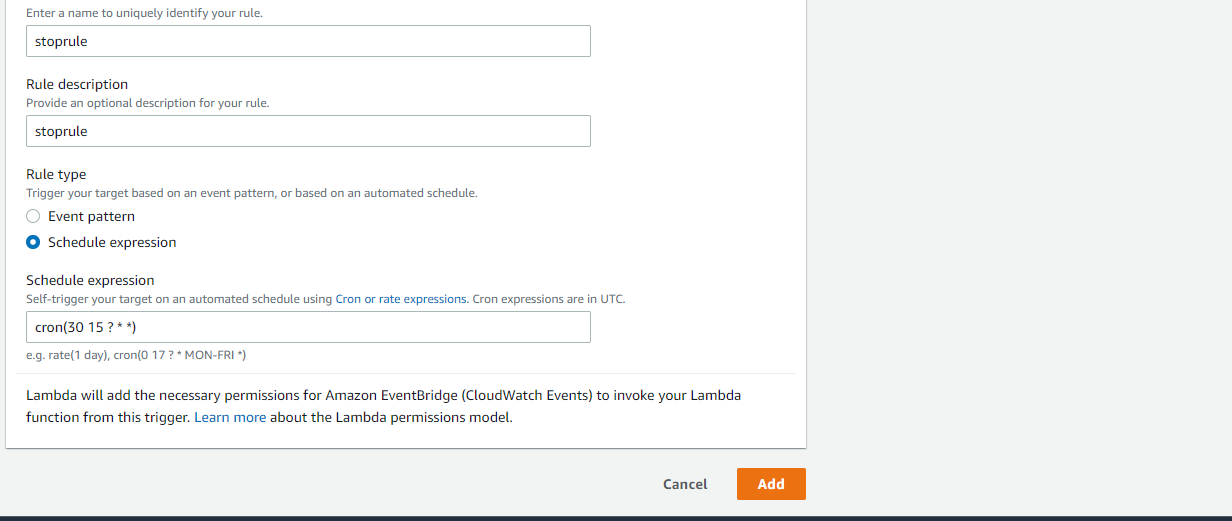


Then deploy & test > create new rule > private > save > test again (instance will be started)



1. Lambda > add trigger > select (eventbridge) > create new rule > shedule expression > use crontab ( UTC time when you want to start or stop) > add





EX: Cron(30 15 ? \* \* \*) // 30-minute, 15- o’clock, UTC-3:30 PM, IST-9PM

For pm 24 hours window for ex 11:30 PM – 30 23 ? \* \* \* -utc 11:30 PM and 5 am ist

12pm ist - 6:30 pm utc – 30 6 ? \* \* \*

Same as for start also:

Firstly change IST to UTC time after that check expression in <https://crontab.cronhub.io/>

<https://crontab.cronhub.io/> - for time confirmation

\*\*\* Check start and stop next day(AMI)

\*\*\* Check backup also next day(AMI)