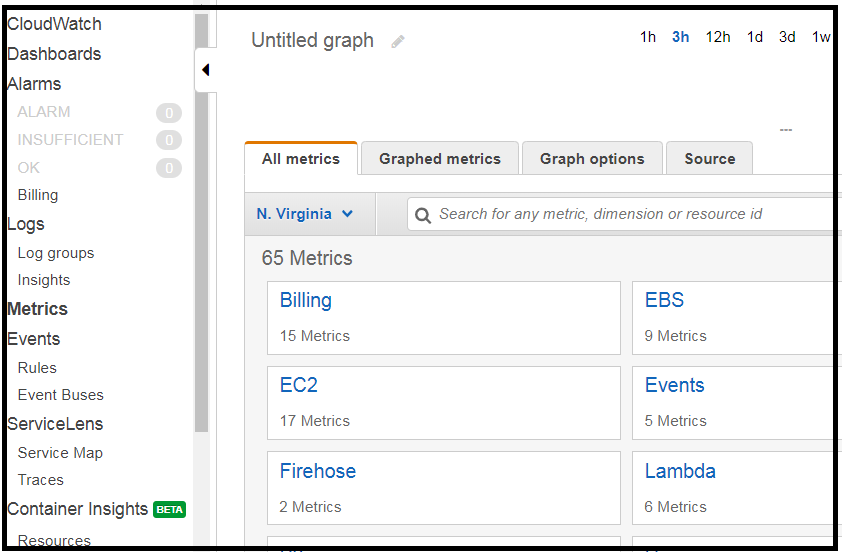
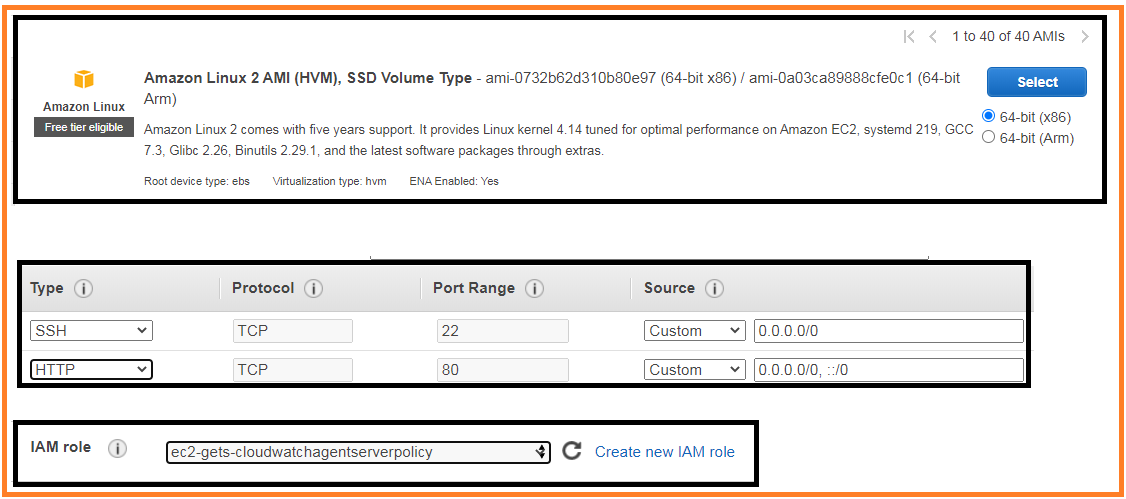
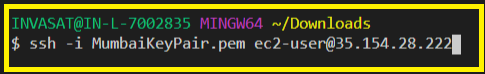
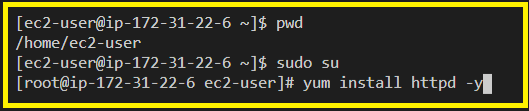
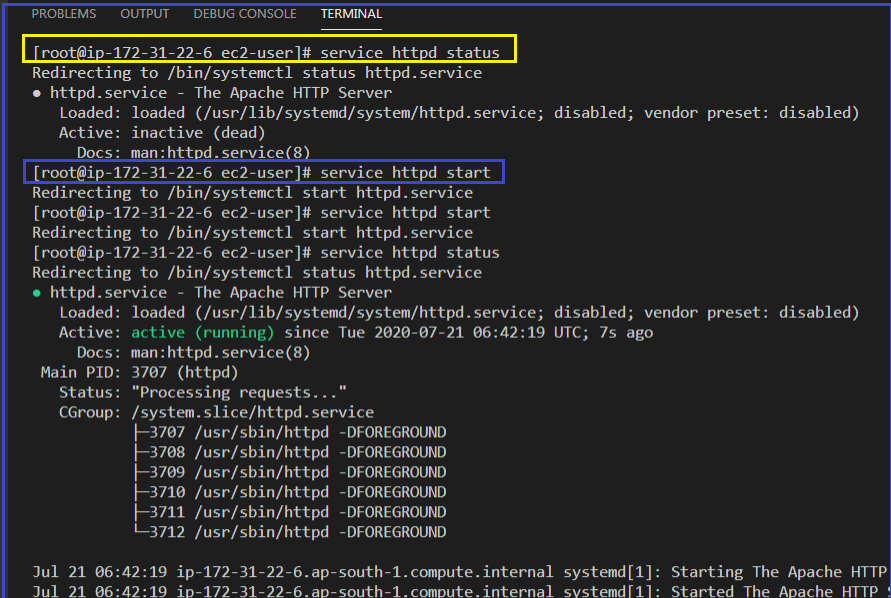
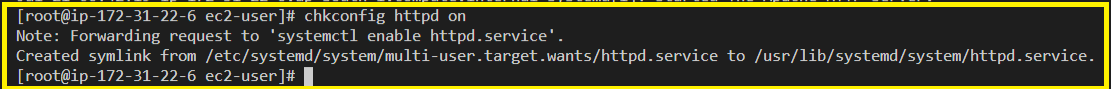
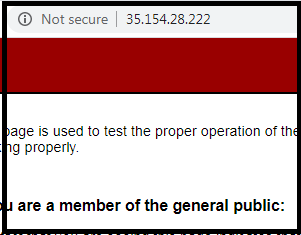
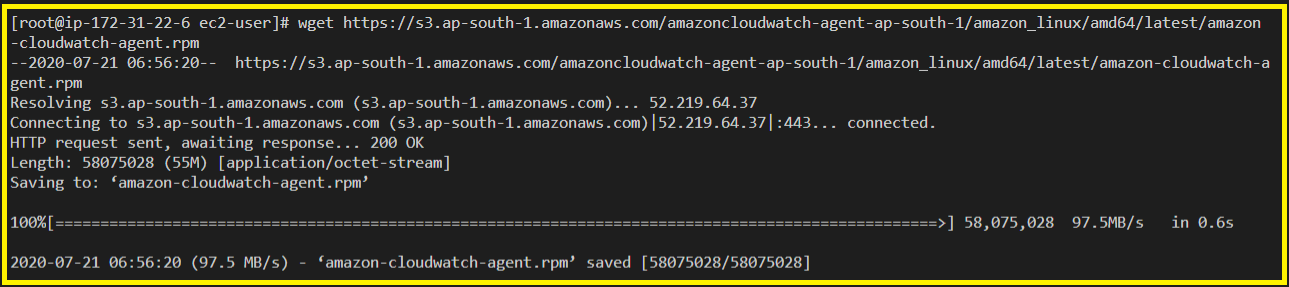
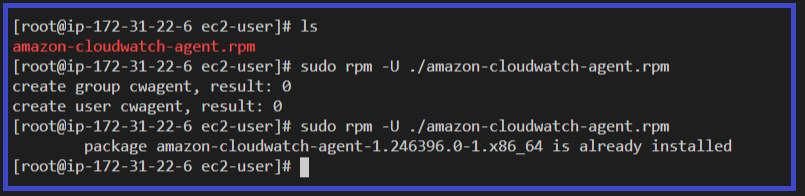
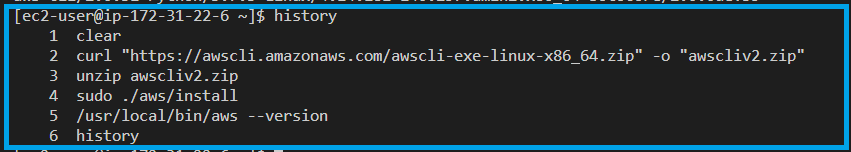
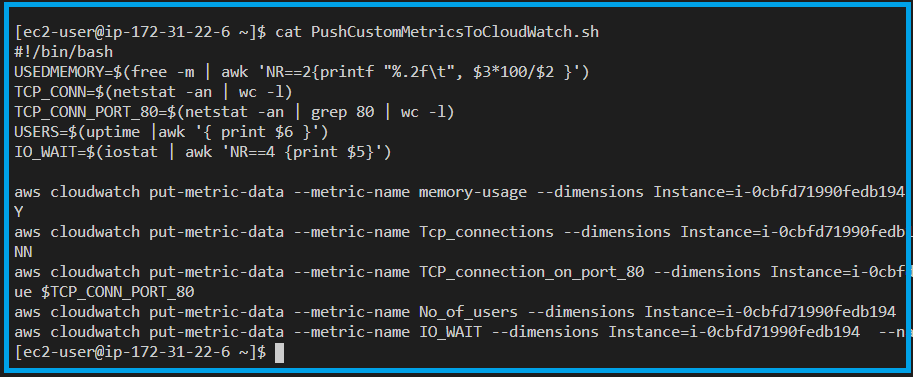
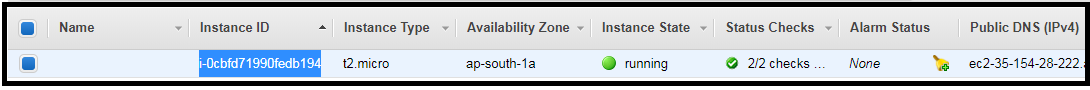
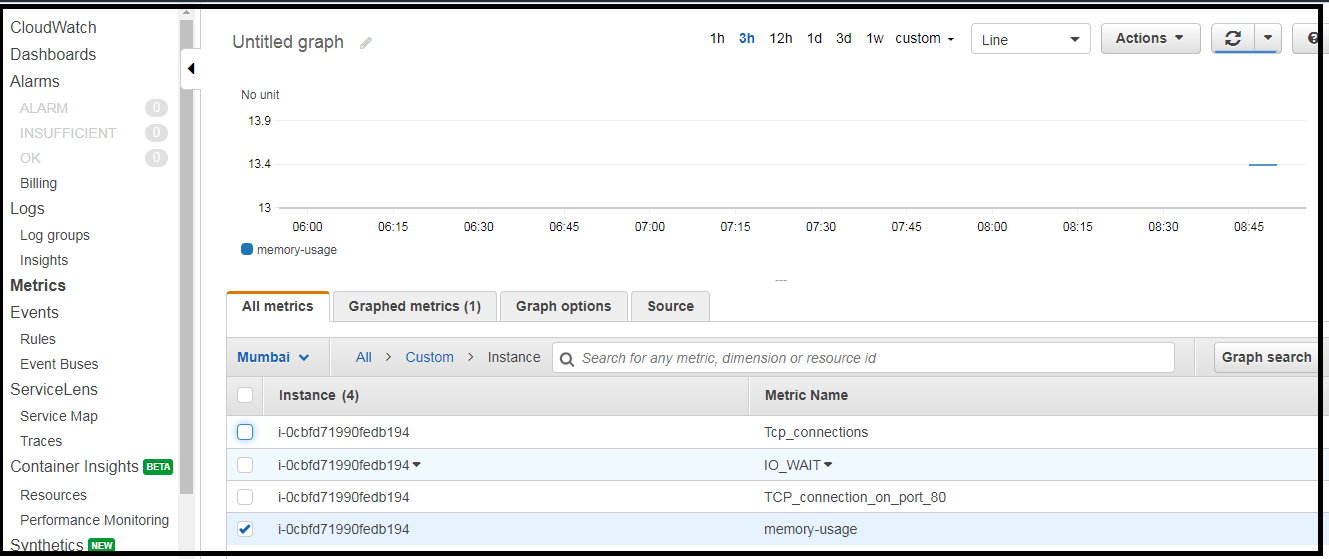
 CLOUD WATCH.  
  
Cloud watch is a monitoring service to monitor your AWS resources.  
It can be used on premise as well and its not restricted to AWS by installing the ssm agent and cloudwatch agent.  
  
It also monitors the applications that you run on AWS.  
  
Ex : Cloud watch can monitor   
Compute : Auto scaling groups, Elastic Load Balancers, Route 53 Health Checks,   
Storage and Content delivery : EBS volumes ,Storage Gateways, Cloud front  
Database & Analytics : Dynamo DB, Elasticache Nodes, RDS Instances, Elastic Map Reduce Job flows, Redshift.  
Other : cloud watch can monitor other things like sns topics, sqs queues, Opsworks, Cloud watch logs, estimated charges on your aws bills.  
  
1. What can Cloud Watch monitor on EC2 instances and why you need custom metrics for ?  
By default cloud watch can monitor Host level metrics.  
They are CPU – DISK – NETWORK – STATUS CHECK. [3 different things from a physical point of view]  
  
Cloud watch can not see how much storage space is left in the virtual disk behind the ‘disk’.  
Here Disk refers to – overall disk activity.  
It can not see things like RAM utilization.  
  
What is a custom metric : Anything that falls out of cpu/disk/network is a custom metric. RAM Utilization is a custom metric.

2. How long cloud watch metrics are stored ?  
indefinite storage by default.  
You can also retrieve data from any terminated EC2 instance.  
  
3. Metric Granularity  
It depends on the service.  
Many default metrics for many default services are 1 minute, but it can be even 3 minutes or 5 minutes depending on the service.  
  
4. Normal vs Detailed monitoring.  
Normal/standard Monitoring : Every 5 minutes.  
Detailed Monitoring : Every 1 minute.  
  
5. For custom metrics the minimum granularity that you can have is 1 minute.  
  
6. If you have your default cloud watch setting on for say , EC2 : you are only ever going to get 5 minute data points, even if you have a script that is pushing data every 1 minute. Cloud watch will actually aggregate that over 5 minutes.  
  
  
7. ALARMS   
7.1 You can create an alarm to monitor any cloud watch metrics in your account.  
This can include EC2 cpu utilization, elastic load balancer latency, or even the charges on you AWS bill.  
7.2 You can set the appropriate thresholds based on which the alarms would be triggered, and you can also set what actions should be taken if an alarm state is reached.  
Ex : You could send yourself a sns notification when your bill reaches a certain amount. Or you could even trigger a lambda function which start deleting your infrastructure.   
7.3 They can trigger actions.

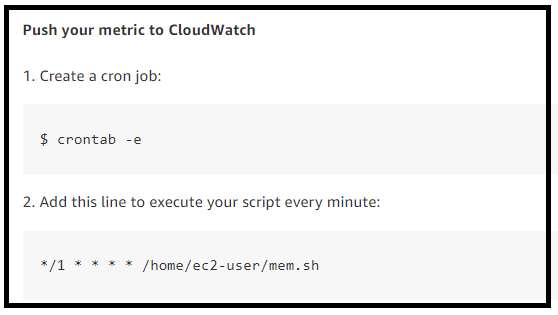
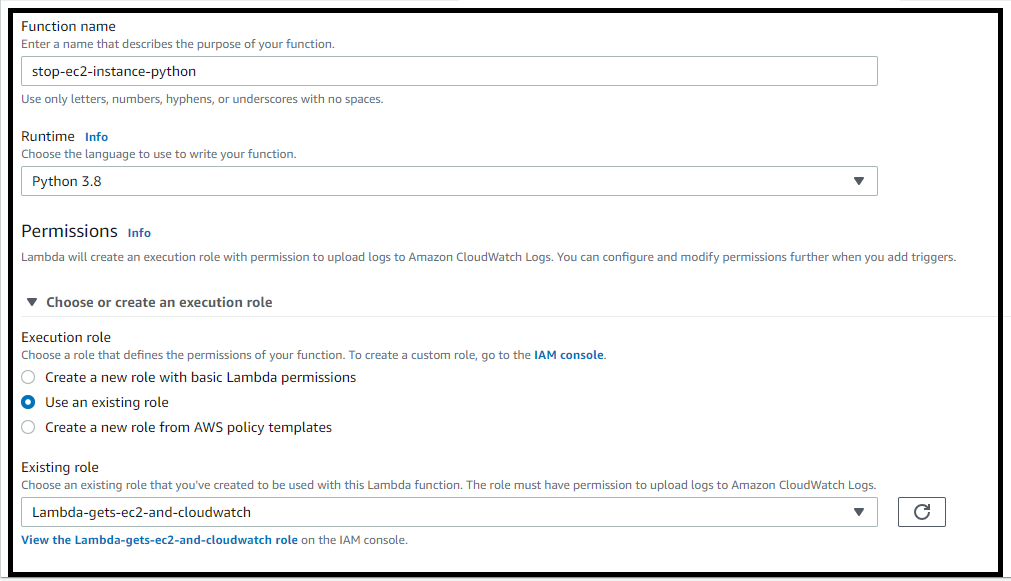
8. LAB : Monitor ‘Application logs’ as well ‘EC2 custom metrics’ using cloud watch unified agent. [Not Working]  
  
8.1 Create a role for EC2 instance , with the below policy attached.  
  
8.2 See the metrics available.  
  
  
8.3 Launch an Amazon linux 2 ami instance with the above role + use an existing regional key pair.  
  
  
8.4 Login in to the machine  
  
  
8.5 Configure the cloud watch agent as a root user. Along with Memory utilization we can also get our custom application logs to cloud watch logs.

So lets install httpd package and send the logs generated by httpd to cloud watch.  
  
  
  
  
  
  
8.6 The service is serving requests.  


8.7 Download the cloud watch package + Install the package + Use it to push data into cloud watch logs.  
  
  
  
  
  
Lets start configuring the cloud watch agent with the help of the setup wizard.  
  
  
This seems to not be working.  
  
9 : LAB : using the CLI push metrics to CLOUD WATCH  
Step 1 : Install the CLI  
  
Step 2 : run aws configure.  
  
Step 3 : Create a bash file which pushes custom metric data to cloud watch.  
  
  
  
  
  
  
Change the instance id and mention the flag –region “ap-south-1” in the above 4 lines.

Step 4 : Watch if the metrics are reflected in the Cloud watch ‘metrics’ tab.  


Step 5 : Lets create a cron Job.

  
  
10 . START AND STOP INSTANCES USING CLOUD WATCH AND LAMBDA  
  
We will learn instance start and stop using a lambda function.  
We can trigger that periodically at a particular point of time using cloud watch.  
  
10.1 : Lambda needs permissions to control EC2 instances using IAM Roles + it needs cloud watch access also to write the logs.  
So create a role associated to lambda (not EC2) with administrator permissions of lambda and cloud watch  
  
10.2 Create a lambda function with the appropriate role.  
  
  
10.3 Code to Stop an instance.  
Code :   
import boto3

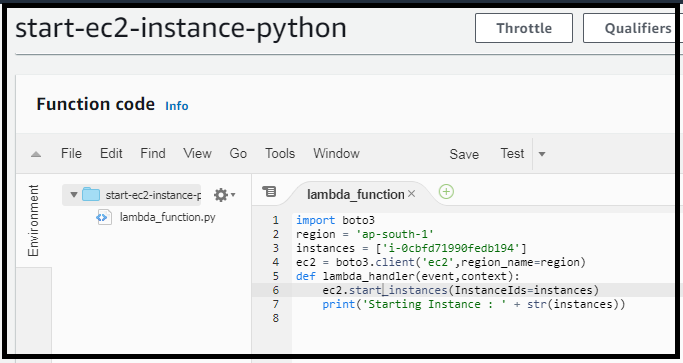
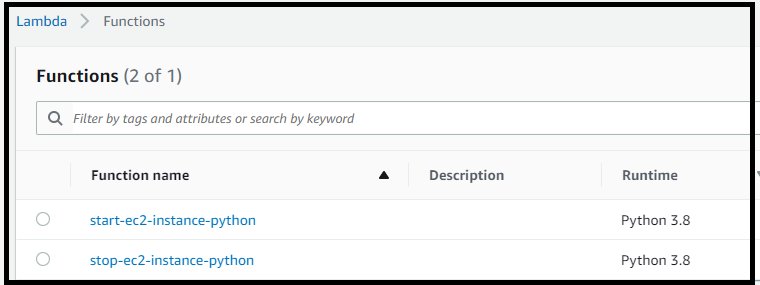
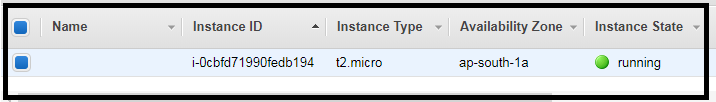
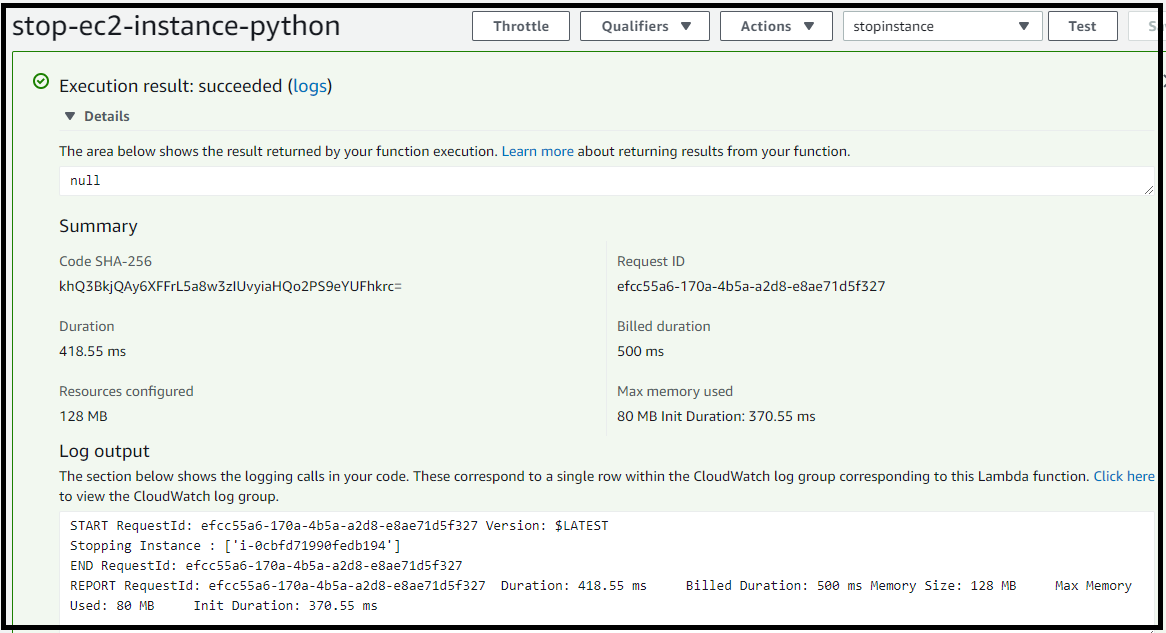
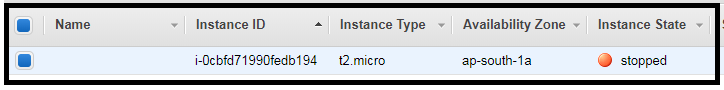
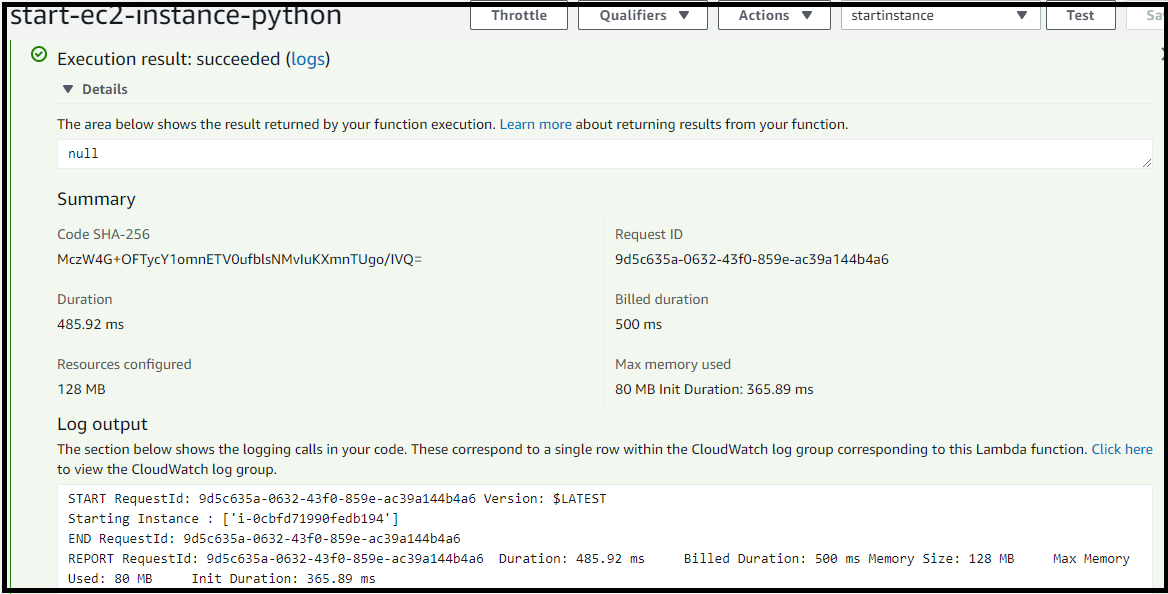
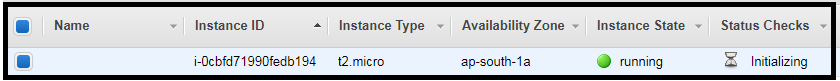
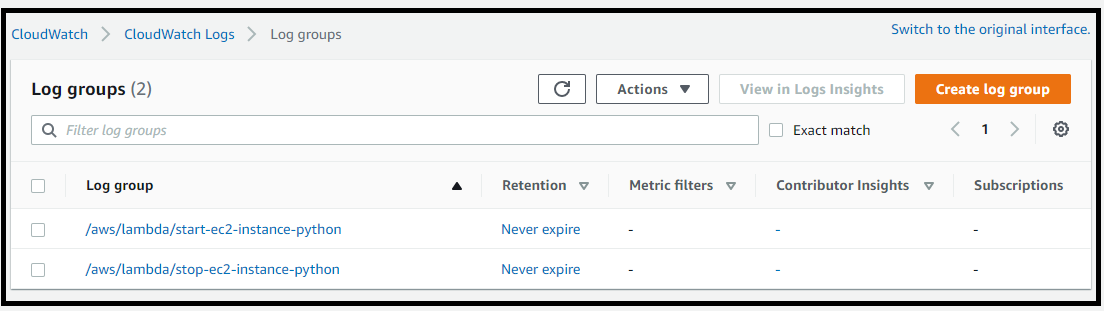
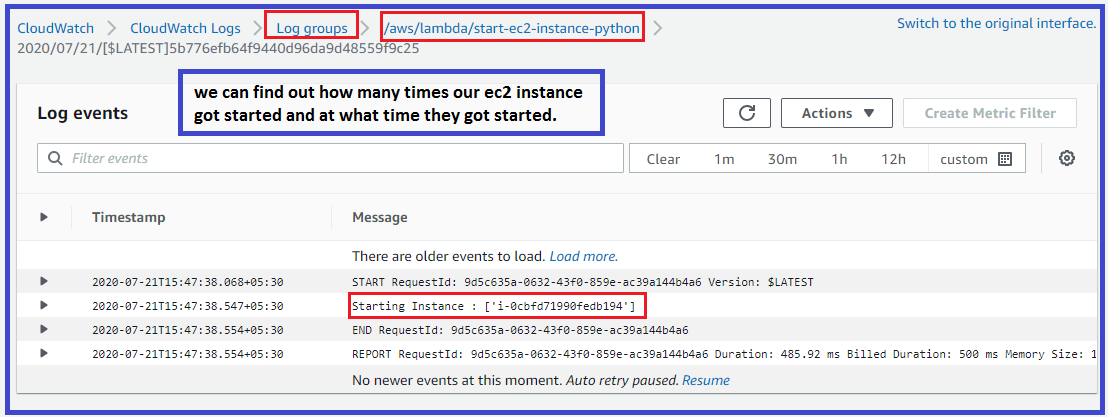
region = 'ap-south-1'

instances = ['i-0cbfd71990fedb194']

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

def lambda\_handler(event,context):

ec2.stop\_instances(InstanceIds=instances)

print('Stopping Instance : ' + str(instances))  
  
  
  
10.4 Create an other python function to start an ec2 instance.  
  
  
10.5 Test the Python functions.  
  
  
Now the EC2 instance is running.  
  
  
  
Test the Stop function.  
  
  
  
Test the Start function.  
  
  
  
  
10.6 Check for cloudwatch logs.  
  
  
  
  
  
10.6 We will use Cloud Watch Events to perform these operations periodically,Instead of doing this manually.  
We can do this by Events / Rules.  
