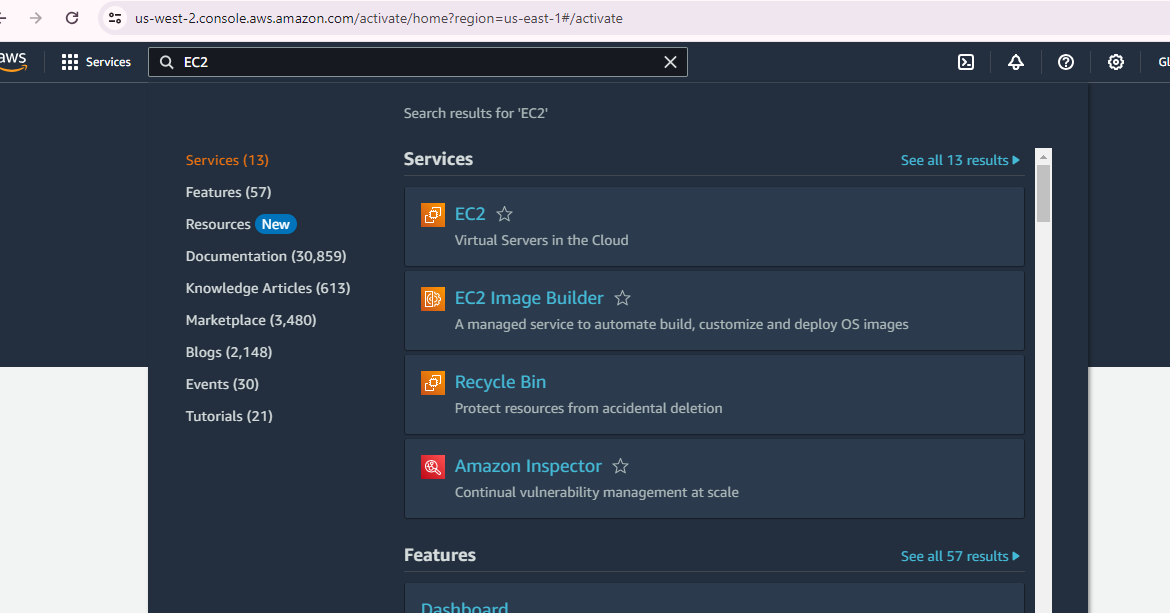
Documentation

1. Infrustructure Setup for web application  
     
   This setup consists of three AWS EC2 instances(Ubuntu) and an application load balancer.

1.1 Launching EC2 instances

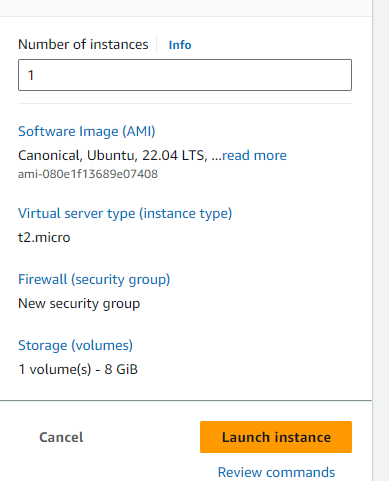
First, visit AWS free tier console.

Search and select EC2.

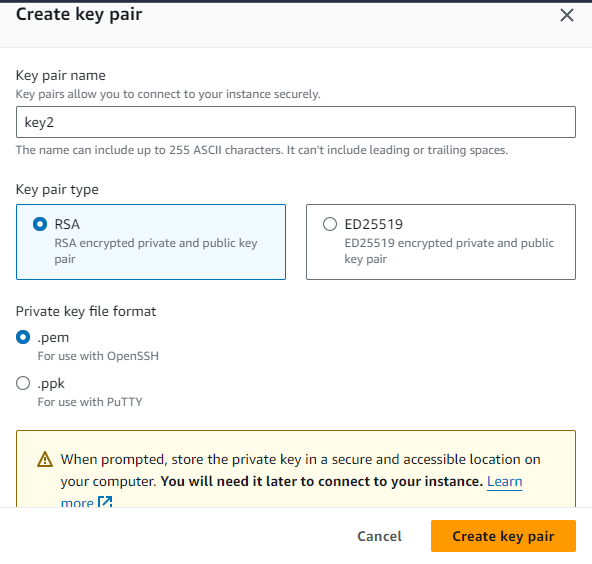


Click ‘Launch Instance’

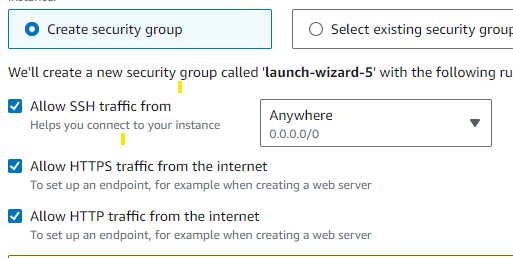
Select required details (AMI, Instance type, storage..etc)



Create a key pair in order to connect to the instance securely.



Allowing SSH, HTTP and HTTPS traffic.



1.2 Setup Apache tomcat on launched instances.

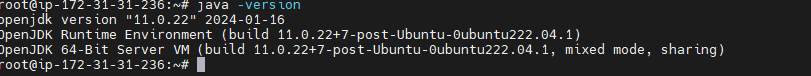
On launched EC2 instances, configure Apache Tomcat.

Install Java on instances

sudo apt-get update

sudo apt-get install openjdk-11-jdk

Verify java is installed properly by running java -version command.



Install apache-tomcat-10.1.20

mv apache-tomcat-10.1.20.tar.gz /opt/tomcat

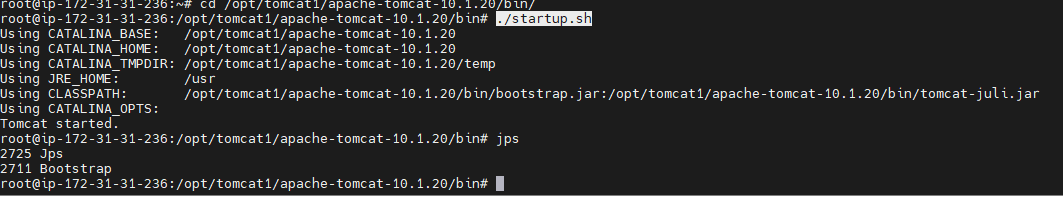
Extract tar file

tar -xvf apache-tomcat-10.1.20.tar.gz

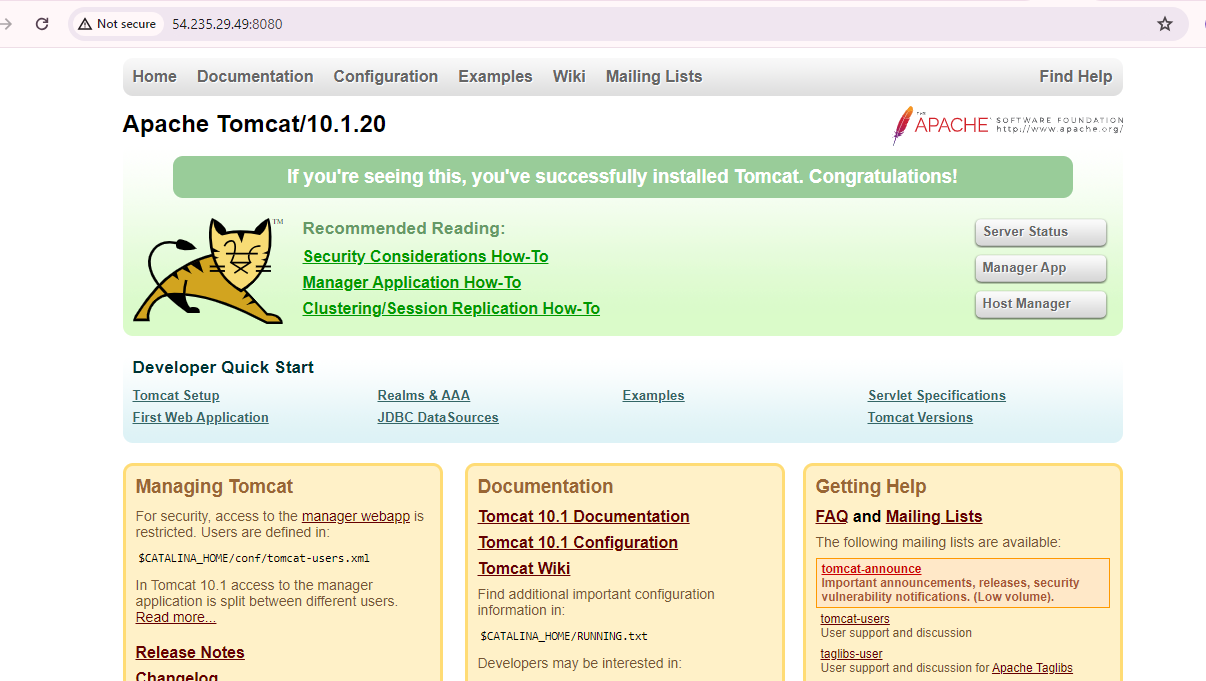
Start tomcat.

cd /opt/tomcat1/apache-tomcat-10.1.20/bin/

./startup.sh



Browse <http://instance_public_ip:8080> to check whether it is working properly.

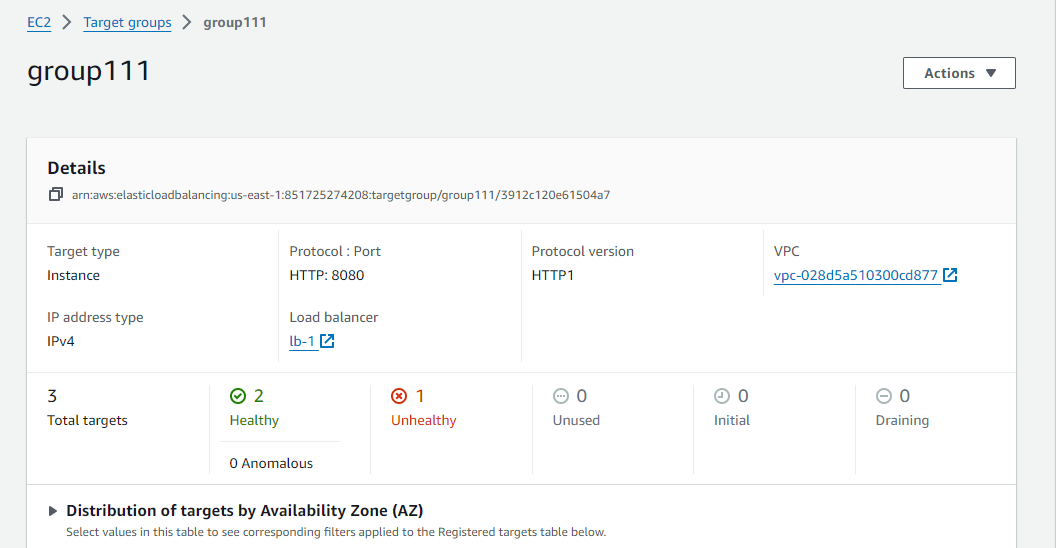


Required war files should be uploaded to following location for running a web application.

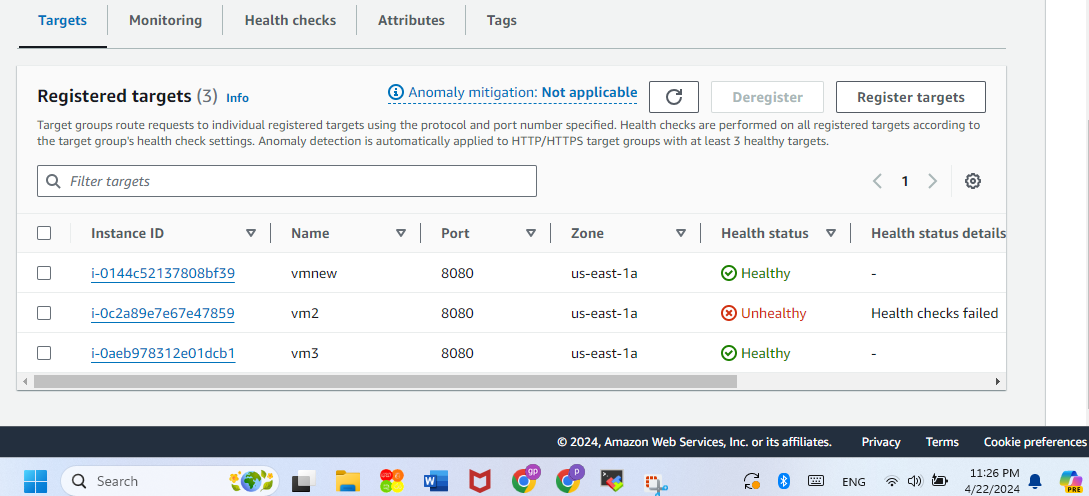


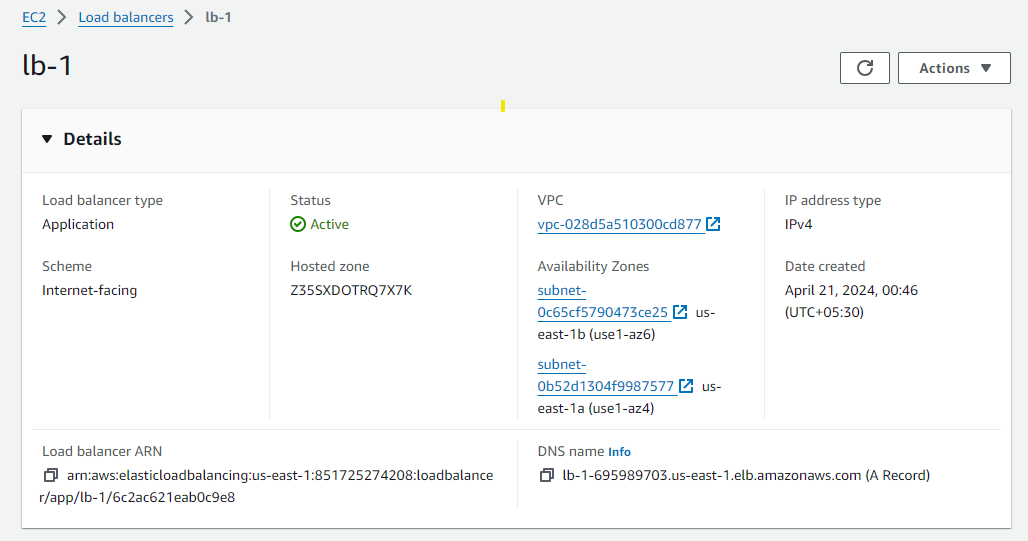
1.3 Launching an application load balancer.

Configuring target group.

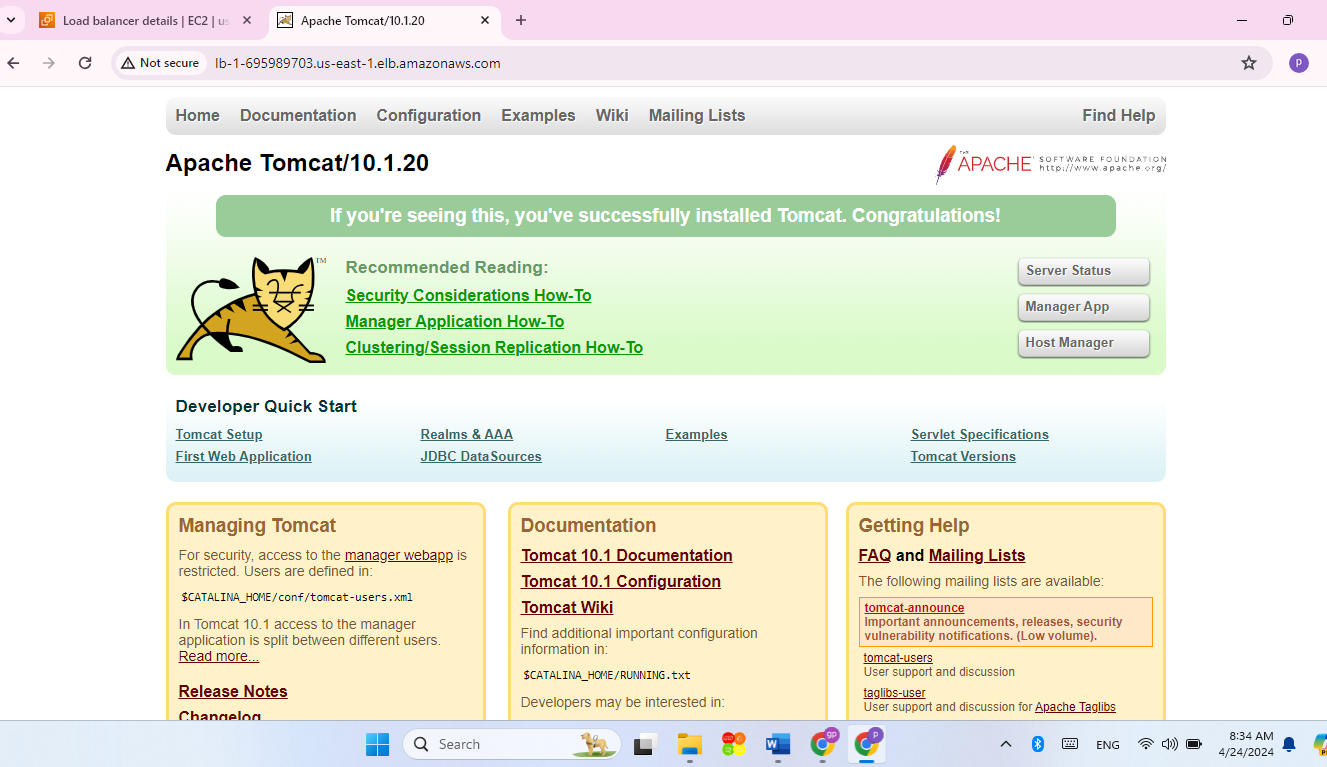


Adding all 3 tomcat servers to the group. Port should be 8080.





Configurations can be checked by browsing DNS of application load balancer. Tomcat web page should be appeared if configured correctly.



1. Cloudwatch monitoring Setup

Five critical stats that will be observed are:

**CPU Utilization**

**Disk Utilization**

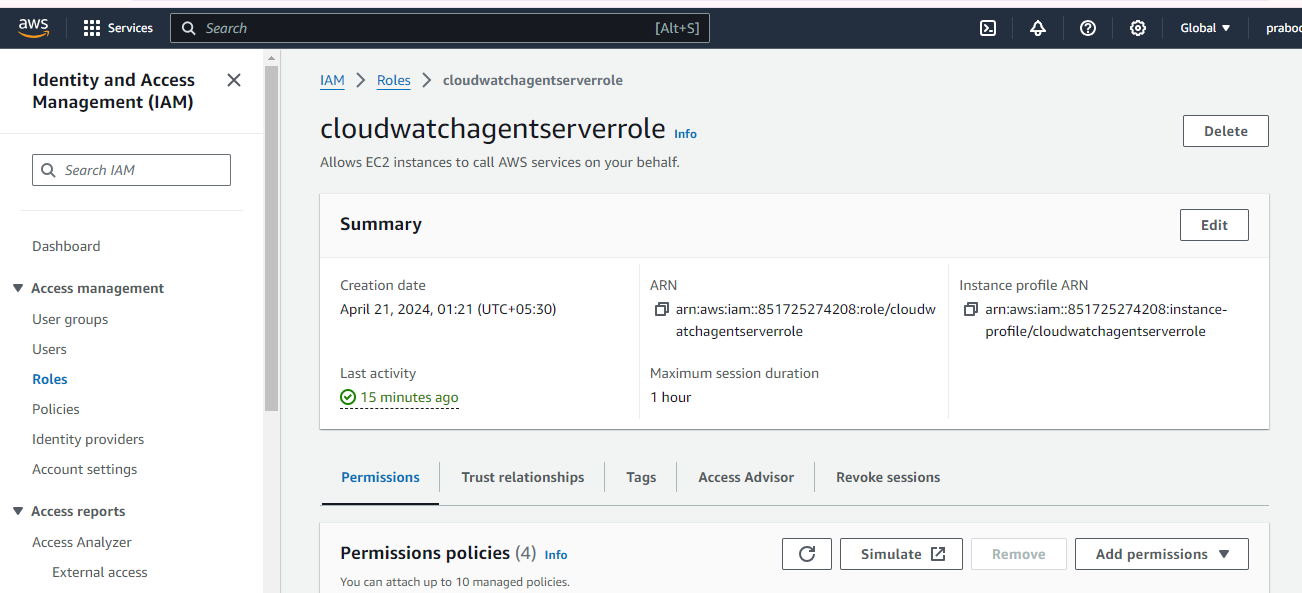
**Memory Utilization**

**Network Utilization**

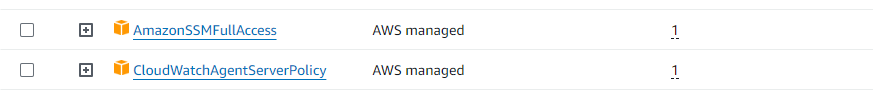
**Instance Status Check**

* 1. Install Cloudwatch agent on instances

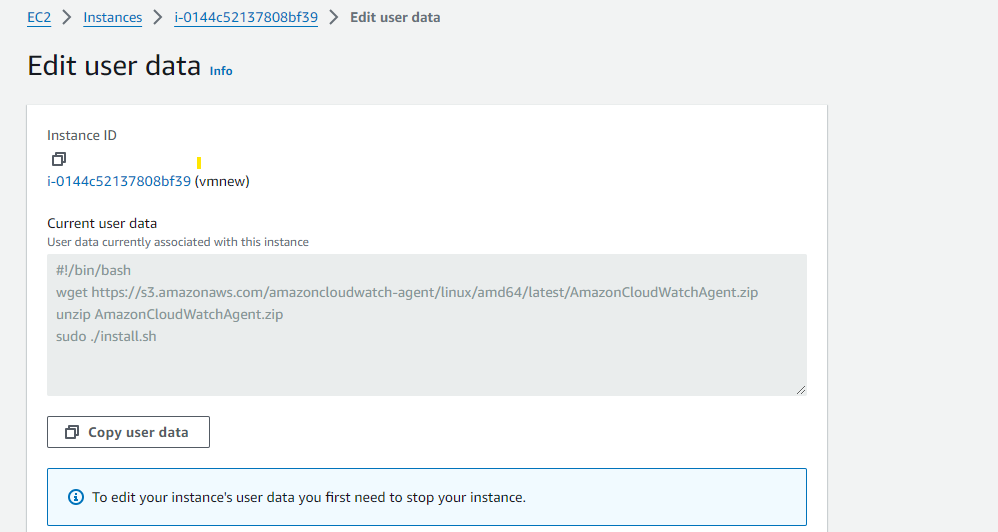
Create a IAM role and assign it to EC2 instances.



Below policies should be attached to the role.



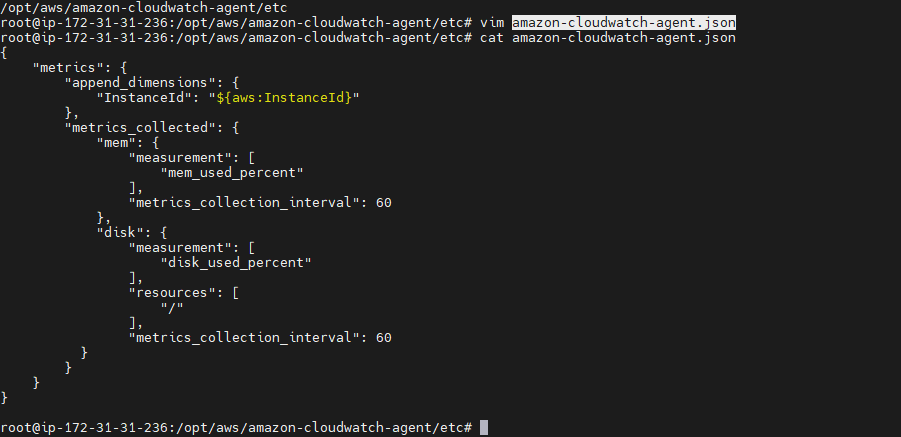
Add commands to instance user data section in order to install cloudwatch agent.



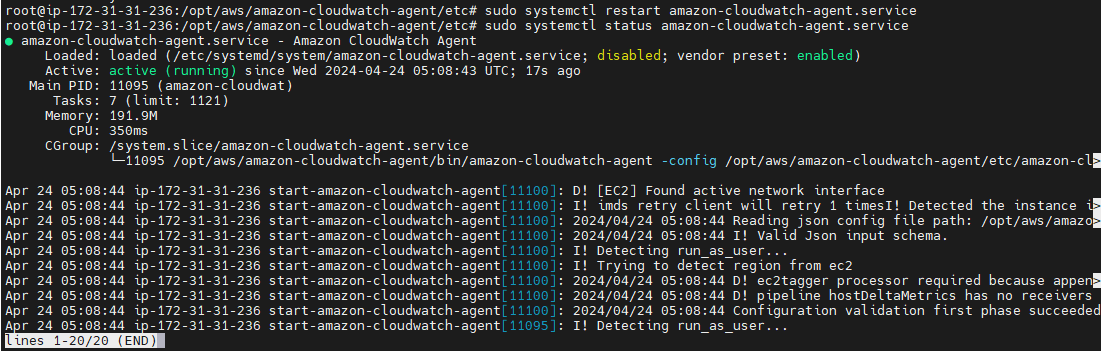
Then, SSH to servers and go to below directory.

cd /opt/aws/amazon-cloudwatch-agent/etc

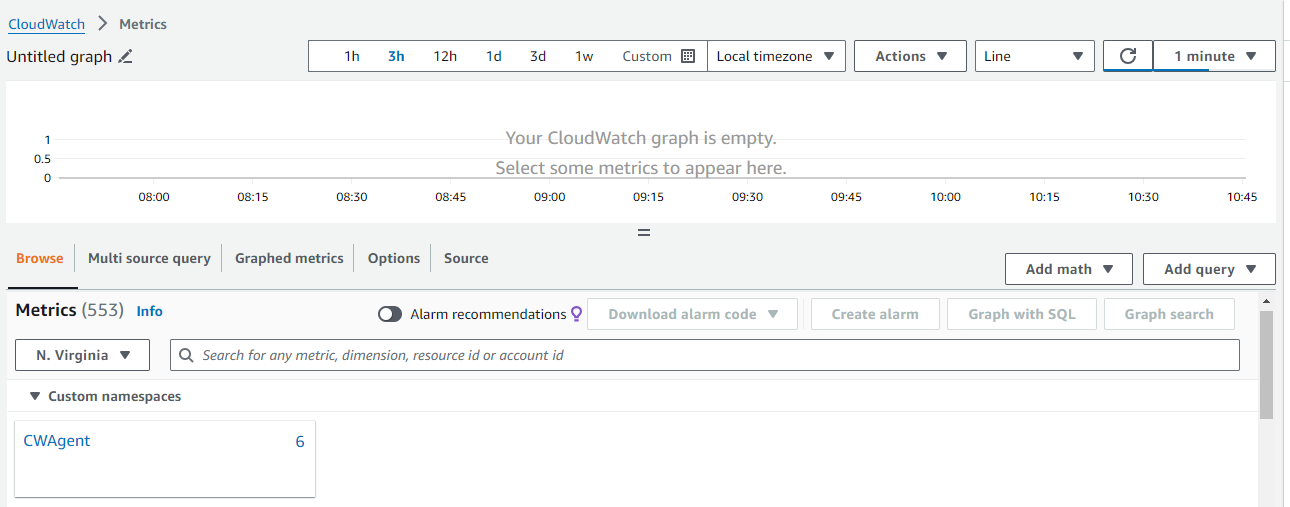
Create a json file to collect cpu and disk utilization metrics on the instance.



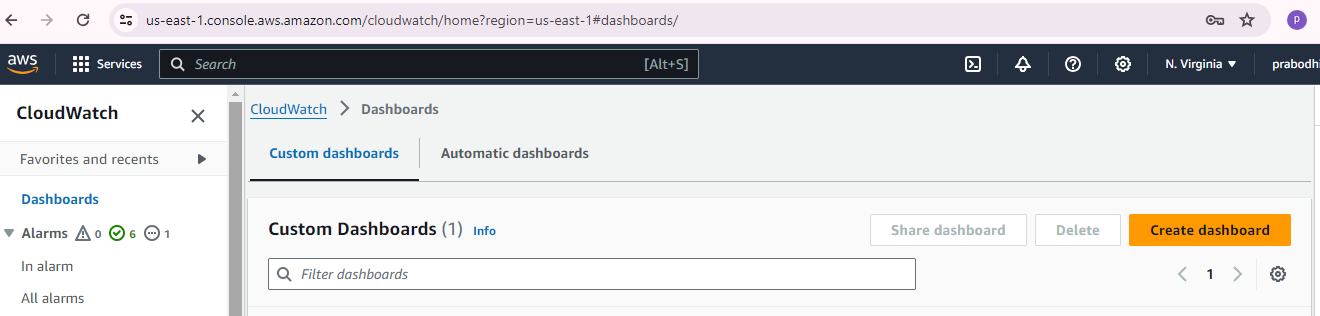
Save the file and restart cloudwatch Agent.

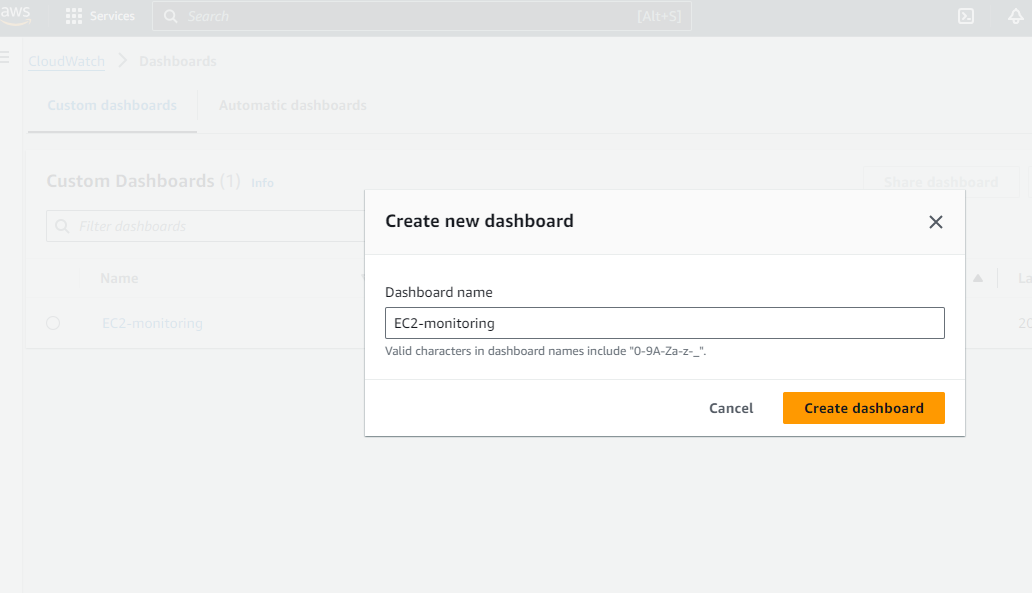


If Cloudwatch agent is installed properly , it will be appeared under custom namespaces in Cloudwatch > Metrics



* 1. Create a cloudwatch monitoring dashboard





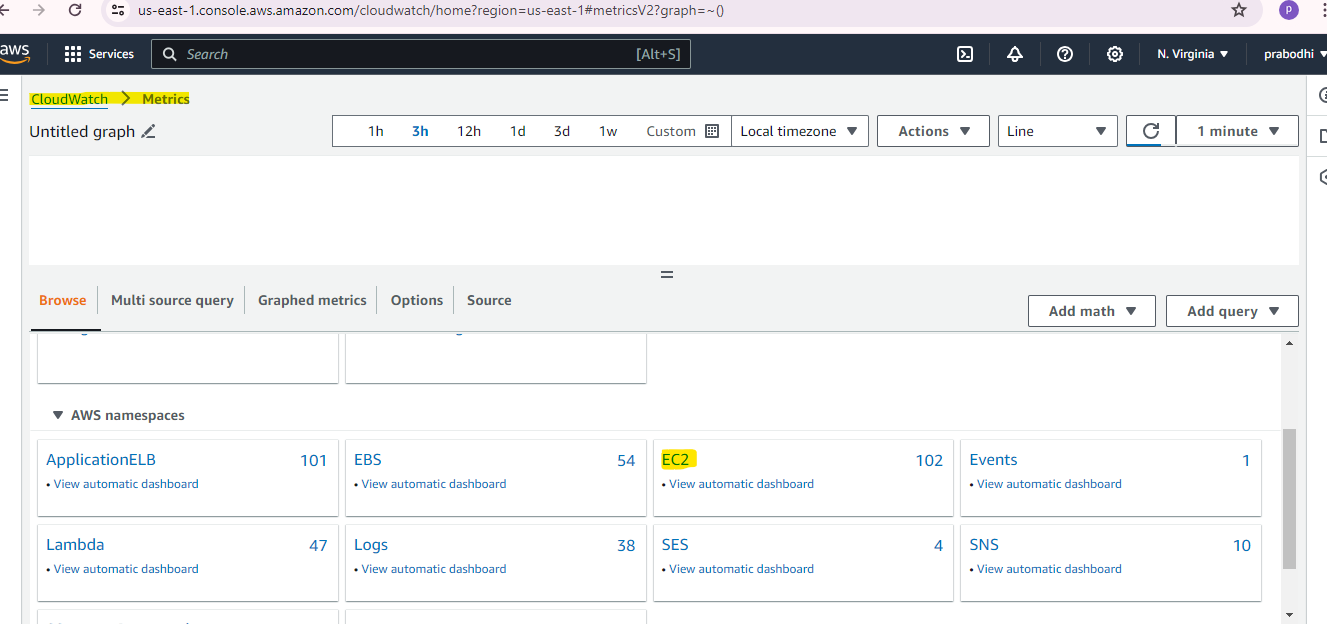
* 1. Setup Cloudwatch Monitoring

CPU Utilization

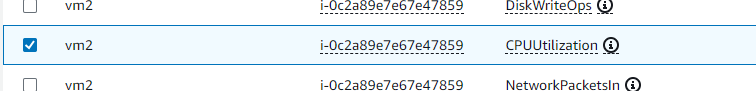
CPU Utilization can be monitored by CloudWatch service directly.(without Cloudwatch Agent)

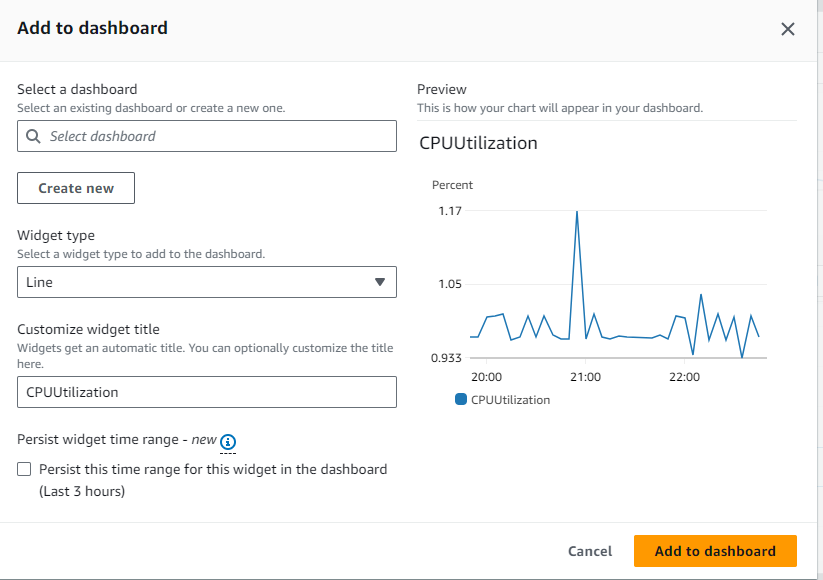
Go to Cloudwatch > Metrics

Select EC2 > ‘Per-Instance Metrics’ under AWS namespaces

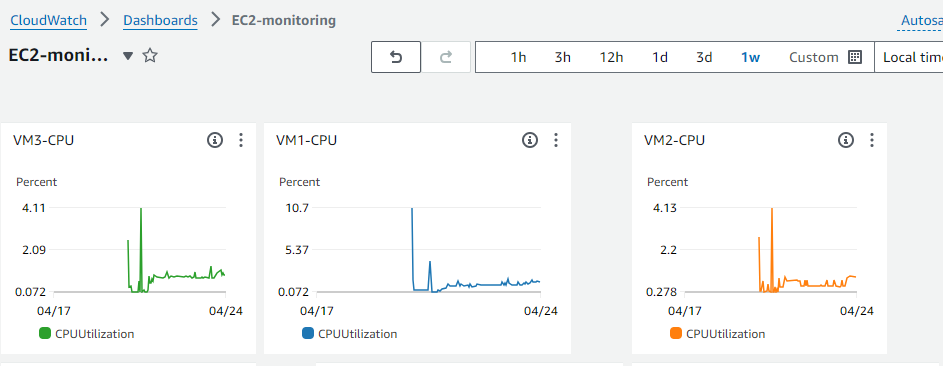


Select Metrics name: CPU Utilization of required instance. Then, add metrics to created dashboard.



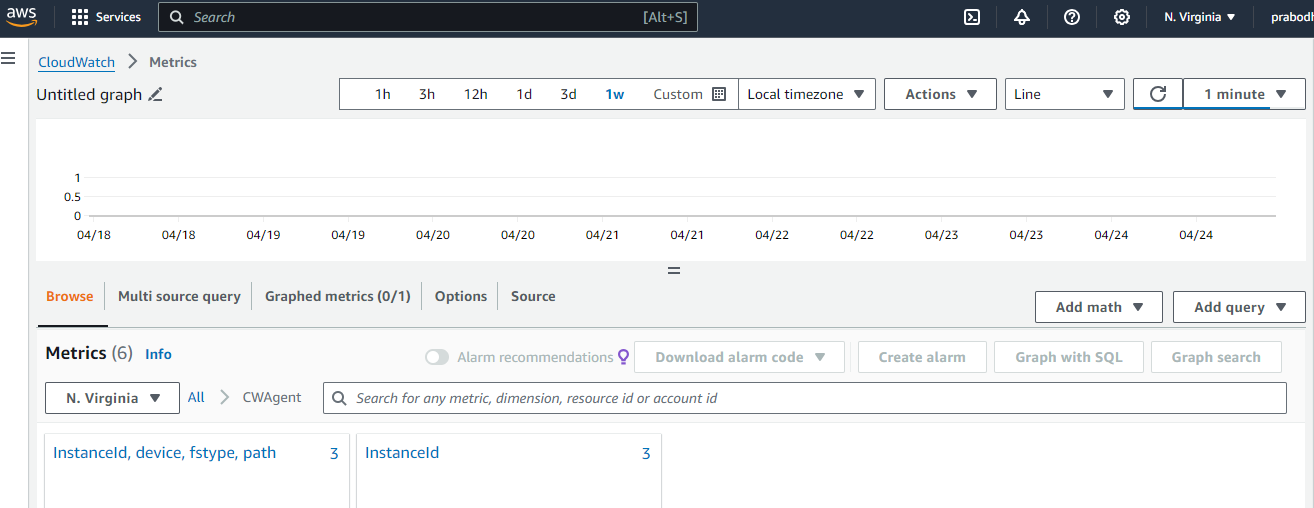


After that, CPU Utilization graph will be visible on dashboard.



Memory and Disk Utilization

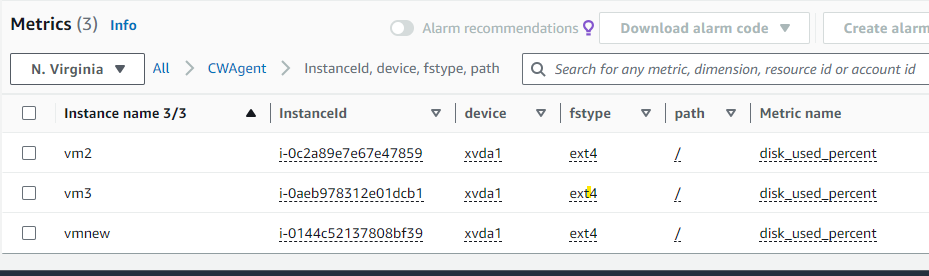
Memory and disk utilization stats are collected and transmitted to cloudwatch through cloudwatch agent.

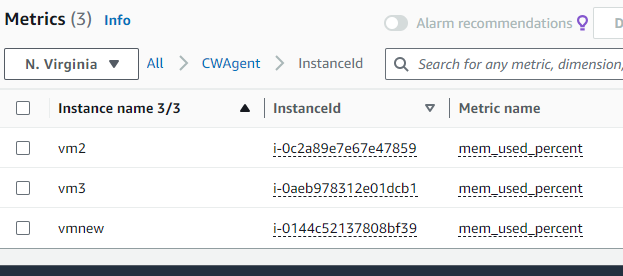


Go to Metrics > CWAgent

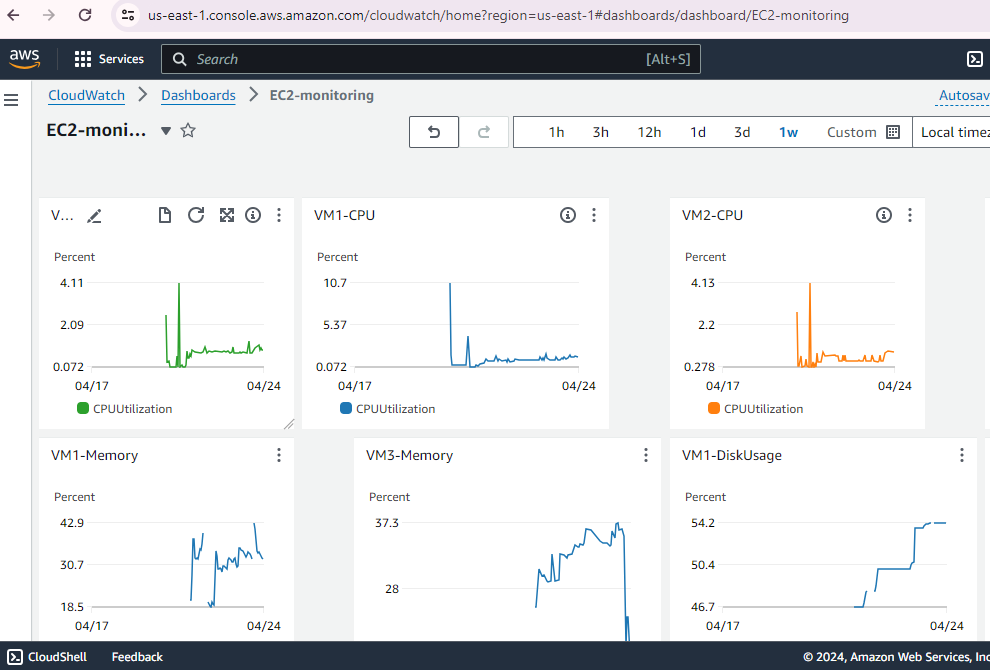
Select ‘InstanceID, device , fstype, path‘ for disk utilization

Select InstanceID for memory utilization





Add these metrics to cloudwatch dashboard similar to cpu utilization.

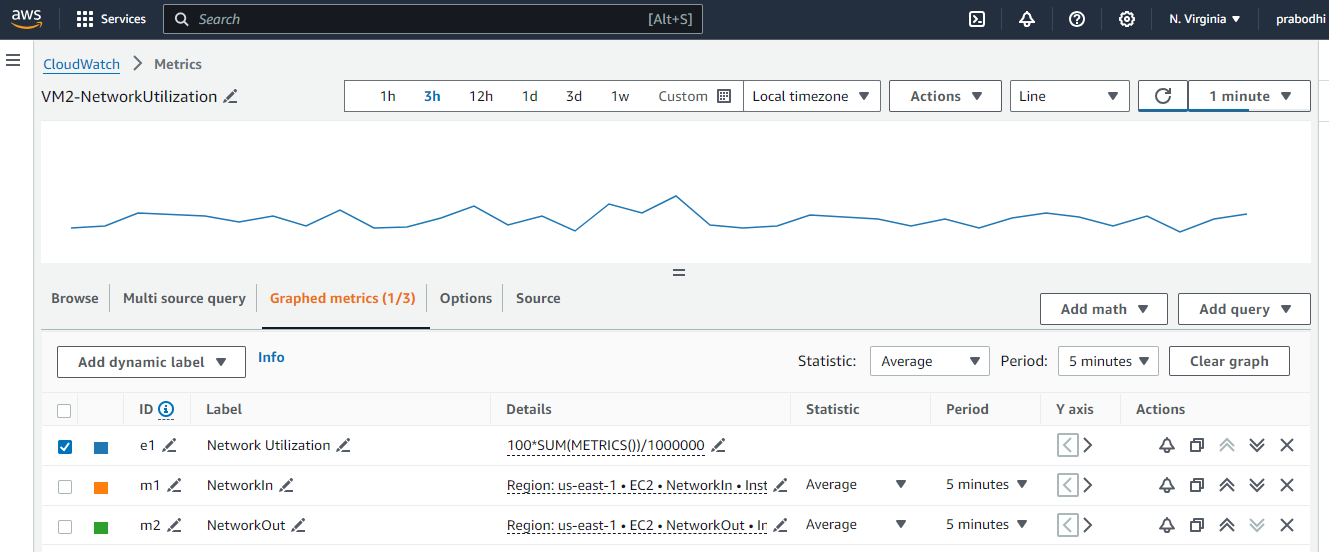


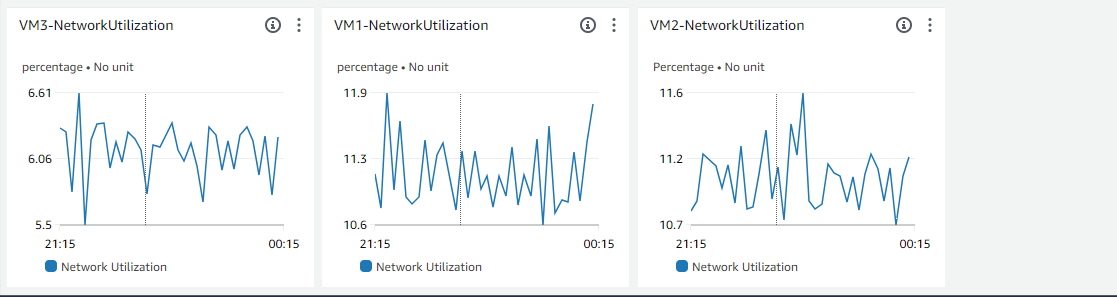
Network Utilization

For monitoring network utilization , a math expression is used with an assumption.

*Network Utilization = [NetworkIn + NetworkOut (current network traffic) / total bandwith] \* 100*

Assumption : total bandwidth = 8 mbps ( 1000 000 bytes per second)





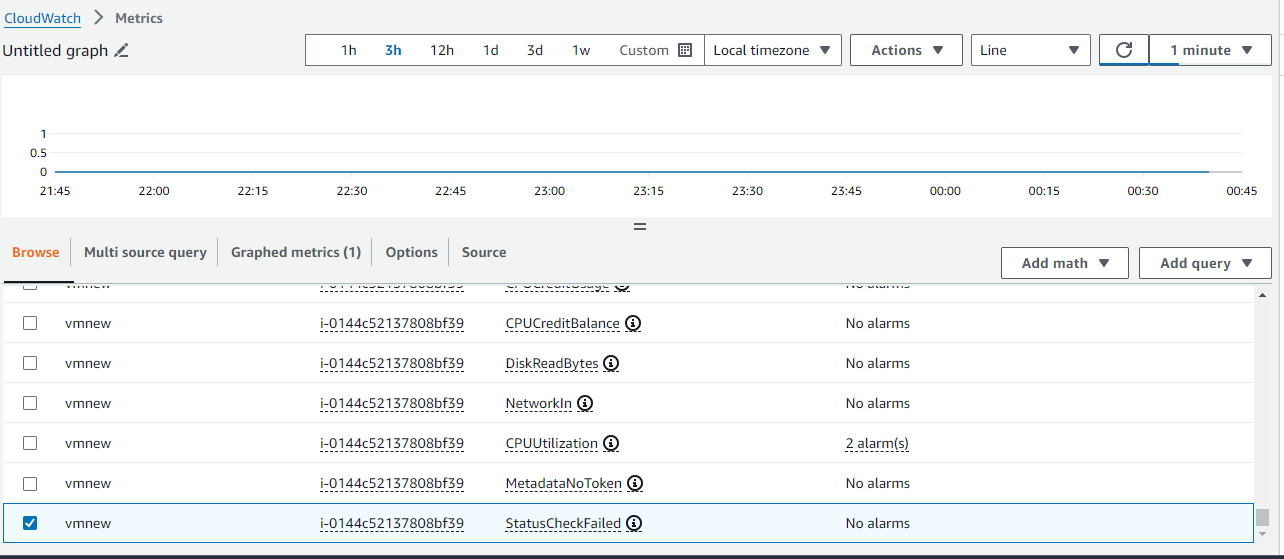
Instance Status Check Stats

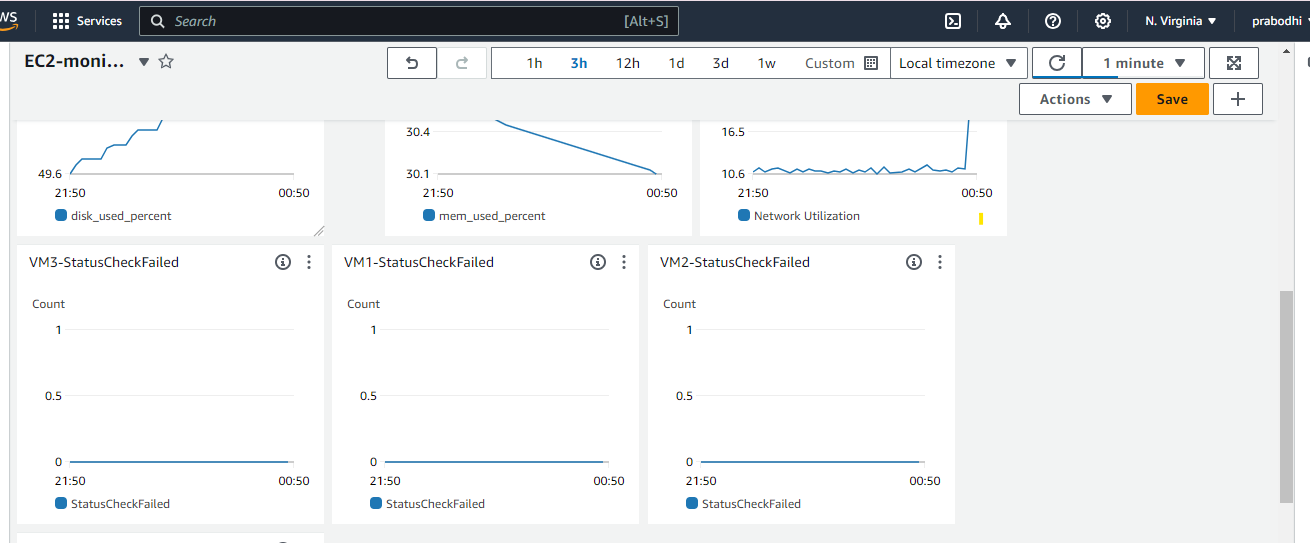
Cloudwatch Instance Status Check metrics can be used to monitor both sytem status and instance status.

System status check : monitor the AWS systems required to run your instances

Instance status check: monitor the software and configuration of instances (checks related to the operating system, instance reachability..etc)

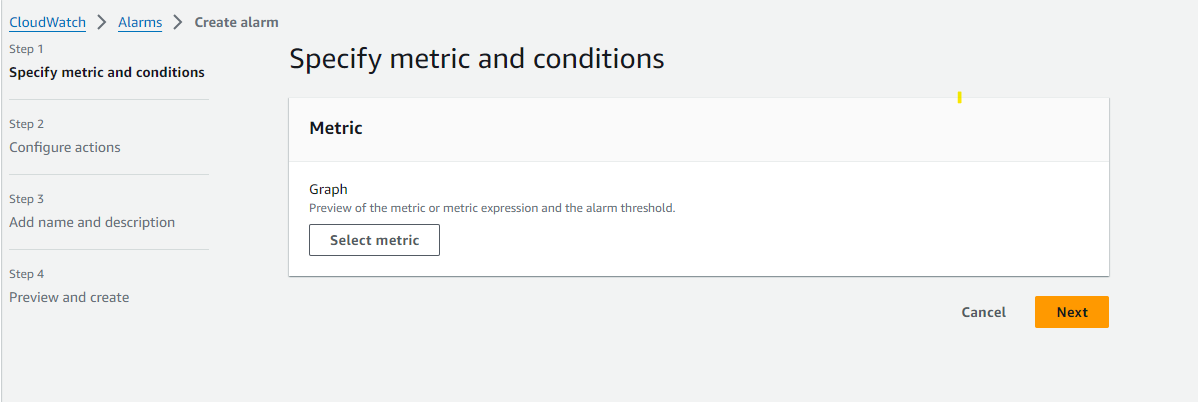
This metric can be either 0 (passed) or 1 (failed). Failures could be due to various reasons such as hardware failures, network connectivity issues, operating system problems, or misconfigurations.

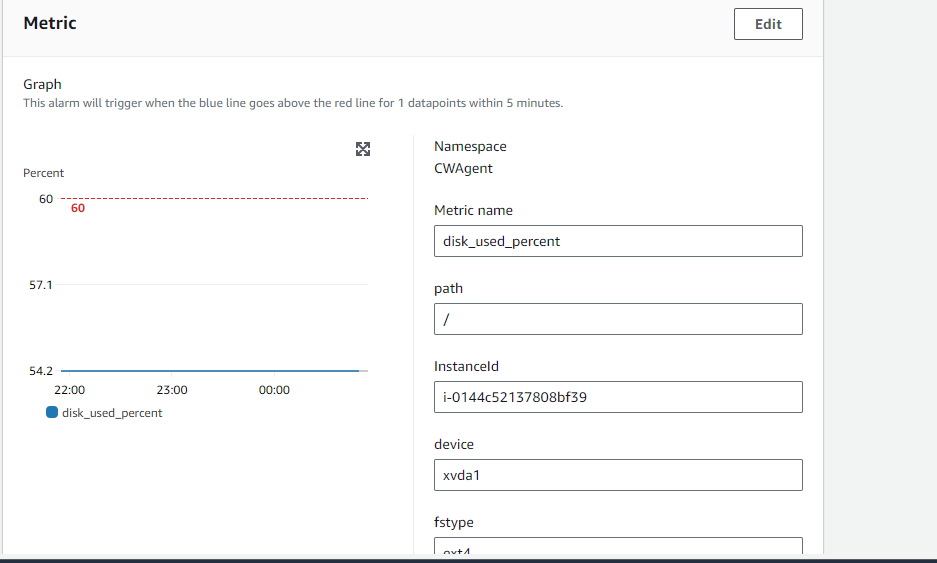


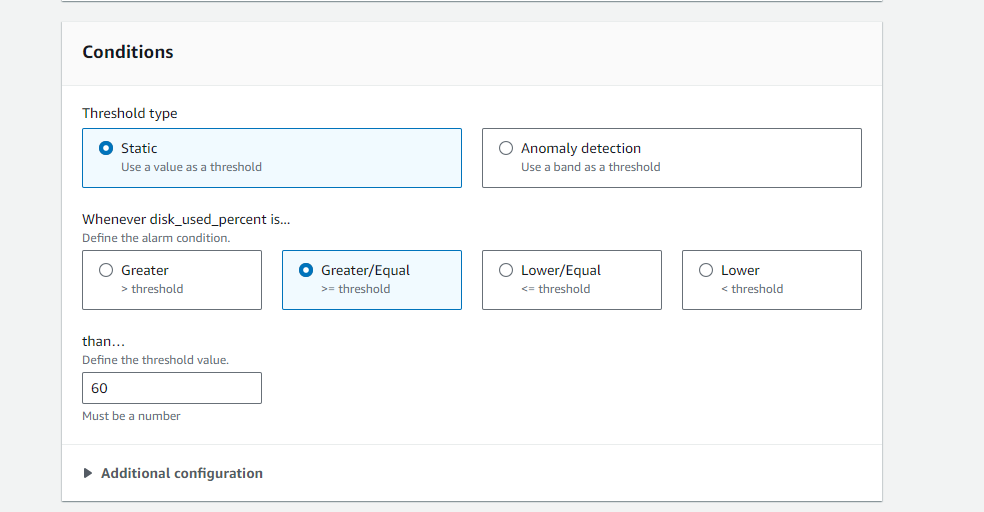


Setup thresholds and alarming via email notifications

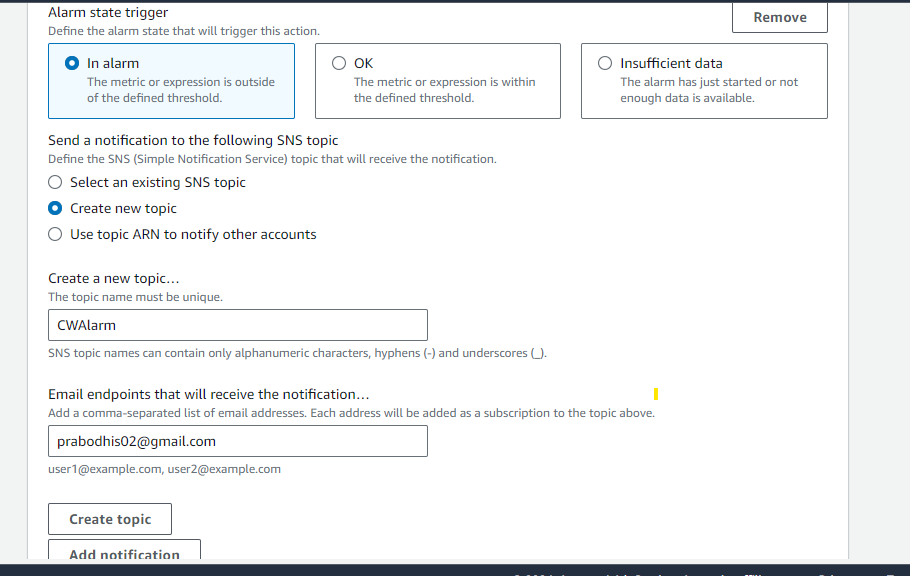
Follow below steps to create alarms



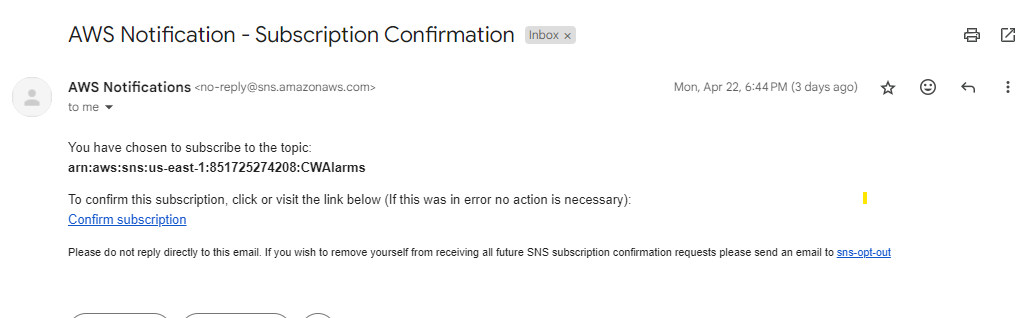


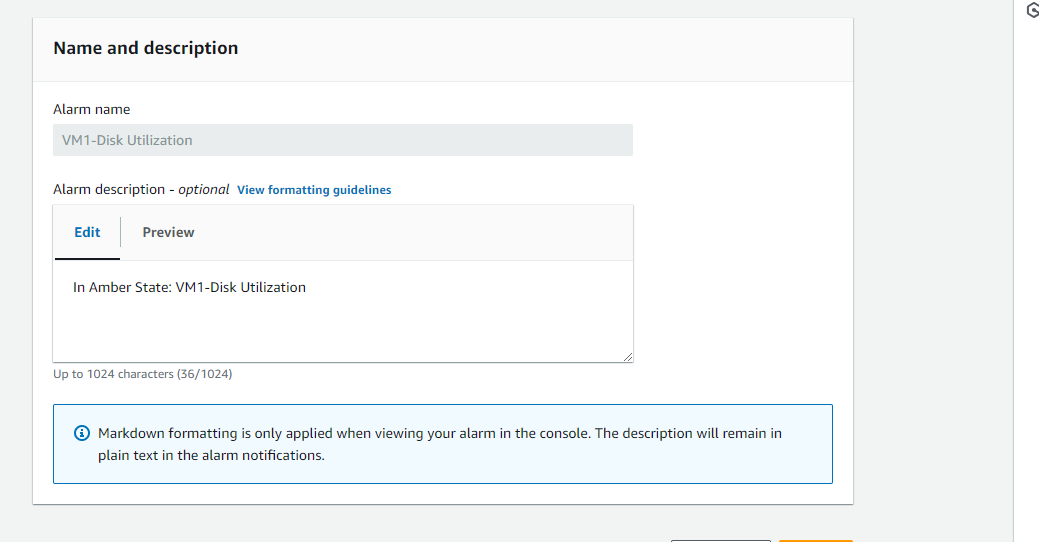


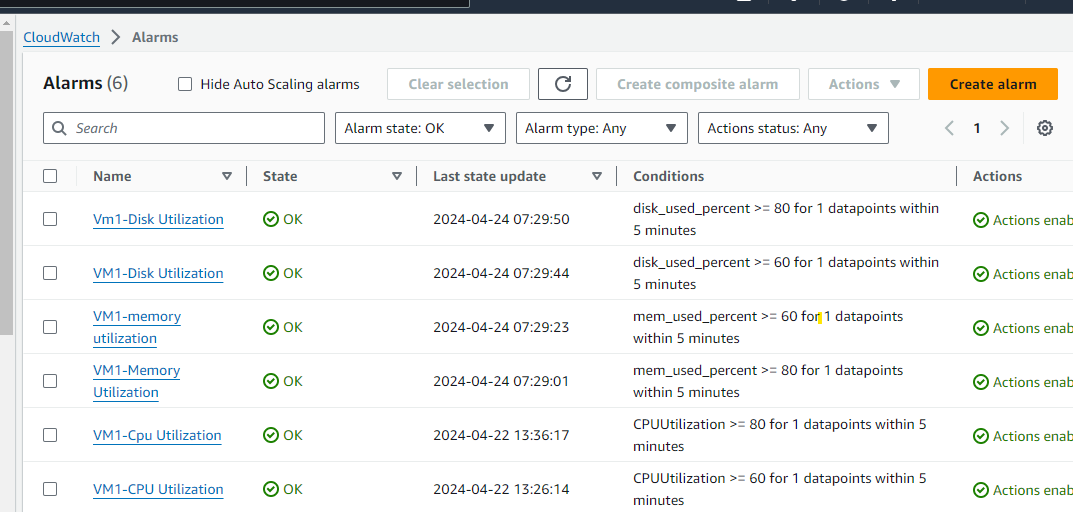
Create SNS topic to send alarms to email addresses regarding threshold violations.



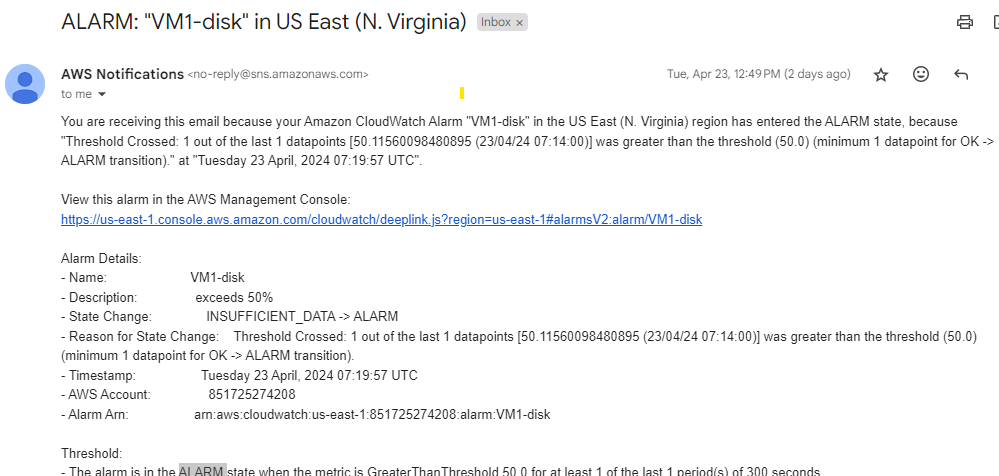
After creating the topic, it should be verified by clicking a link that received to email address.







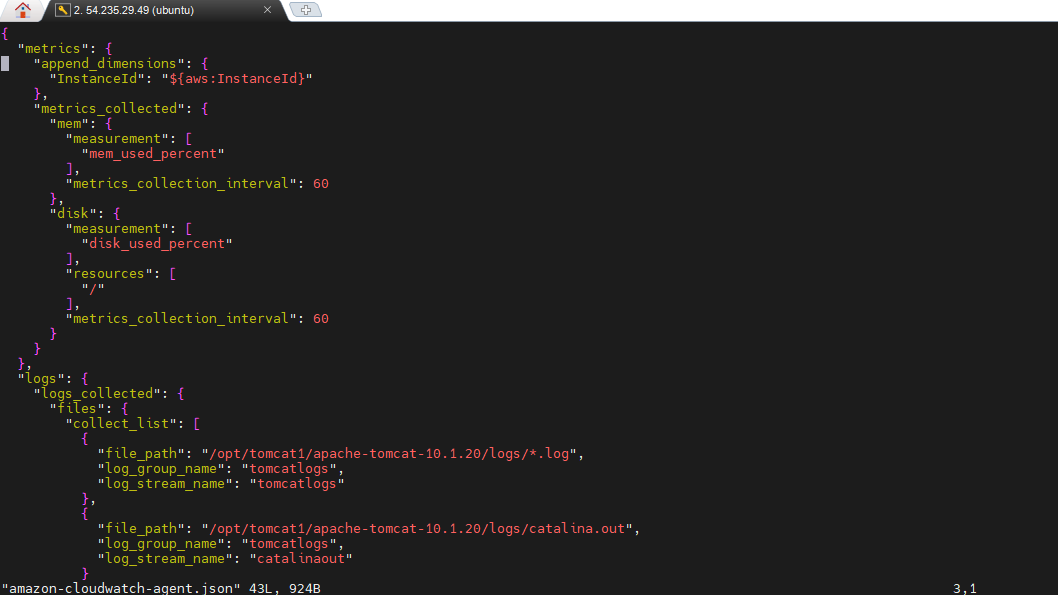
Test Scenario : VM1 Disk Utilization threshold is set to 50% to check whether the email alert is received or not.



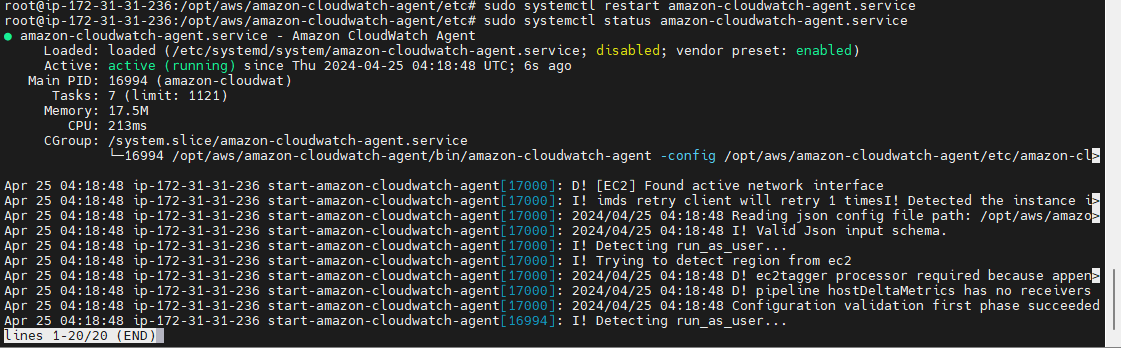
Implement error logging

First, Cloudwatch agent config file should be changed to collect necessary logs from the servers.

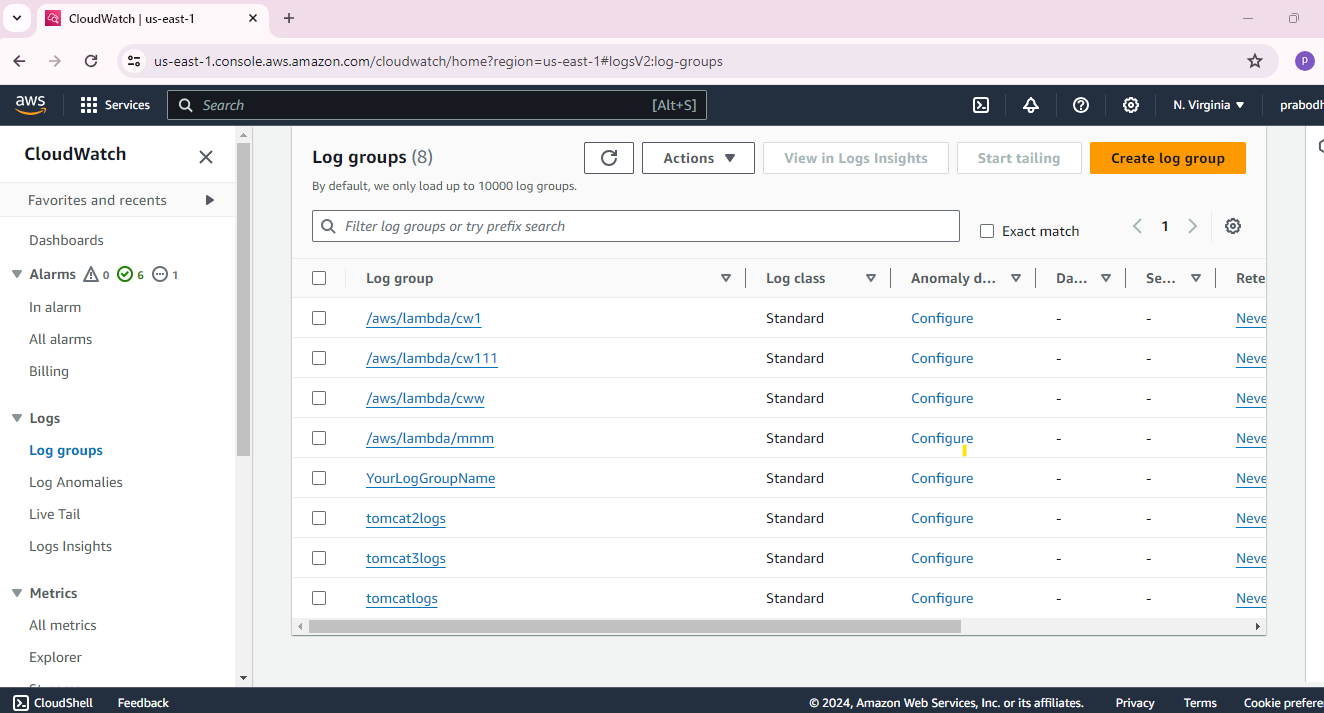
To collect tomcat logs,



Save the file and restart cloudwatch agent.

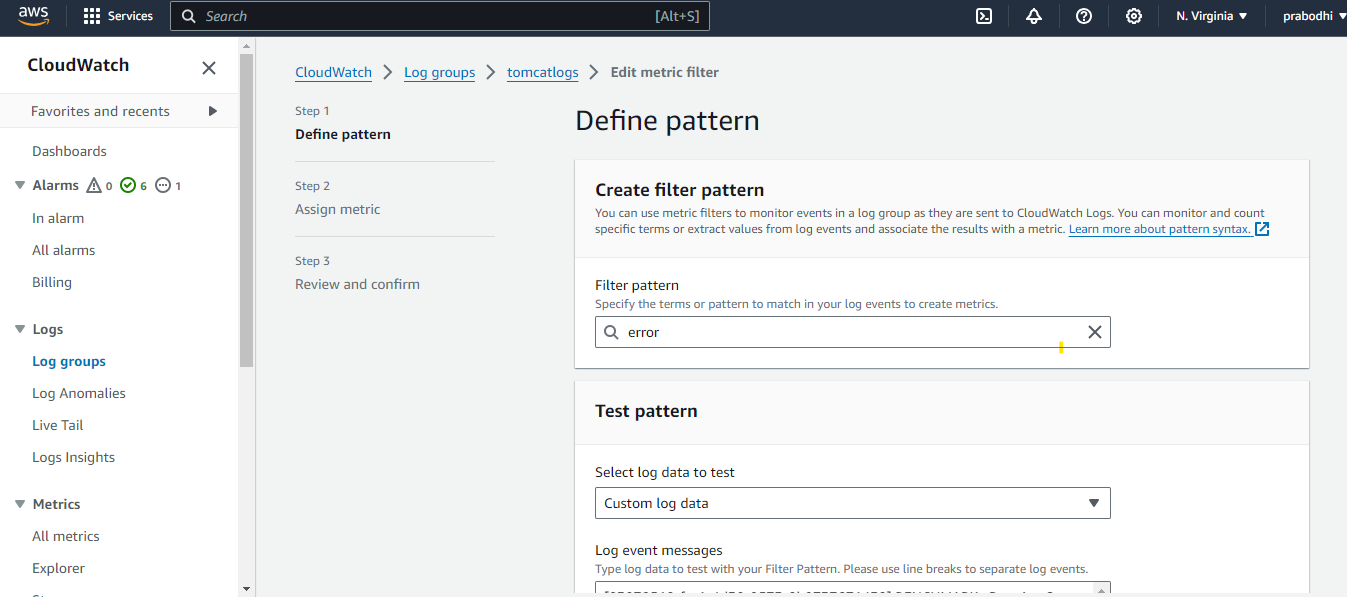


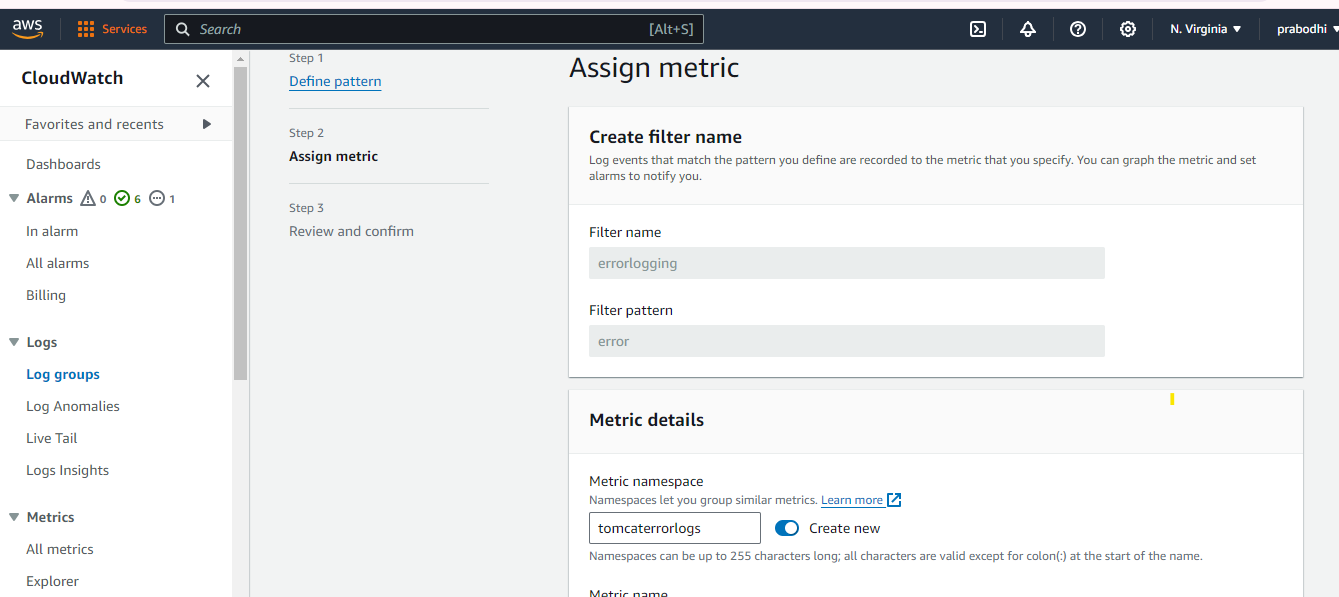
Log group ‘tomcatlogs’ should be appeared in aws console , if the configuration is correct.

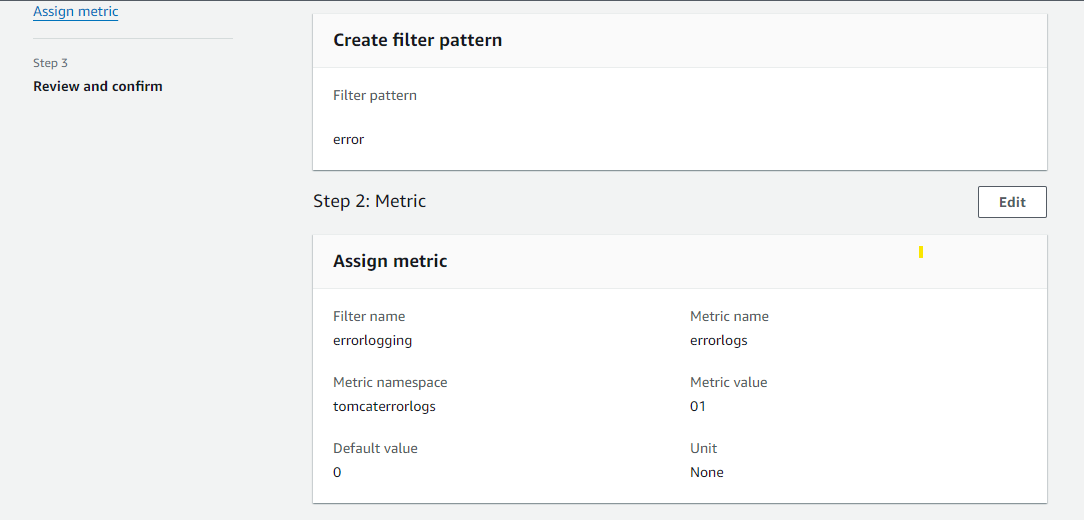


Click created log group and select a log stream inside it.

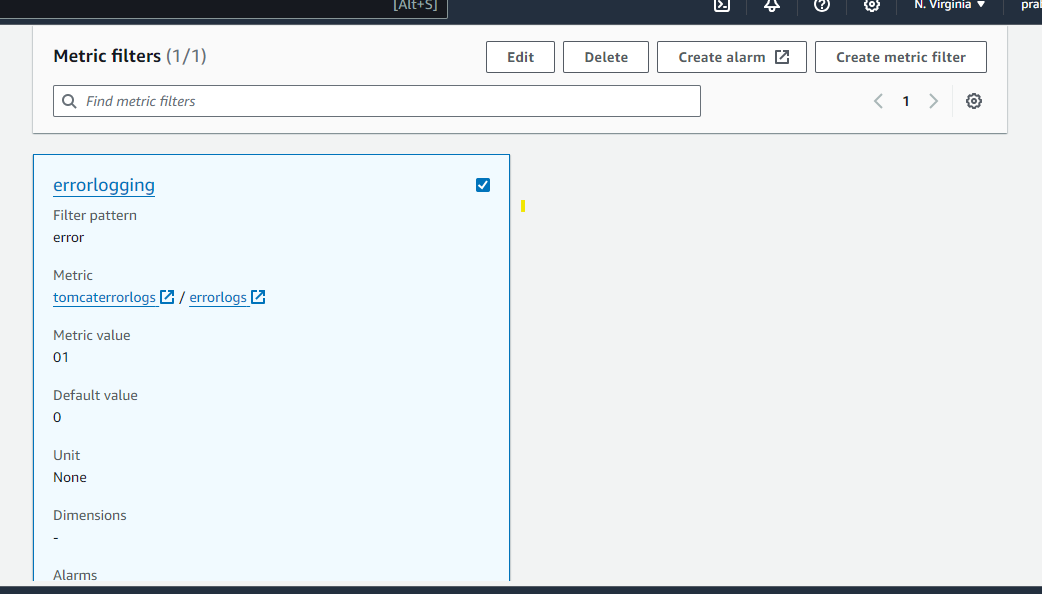
Create a metric filter.

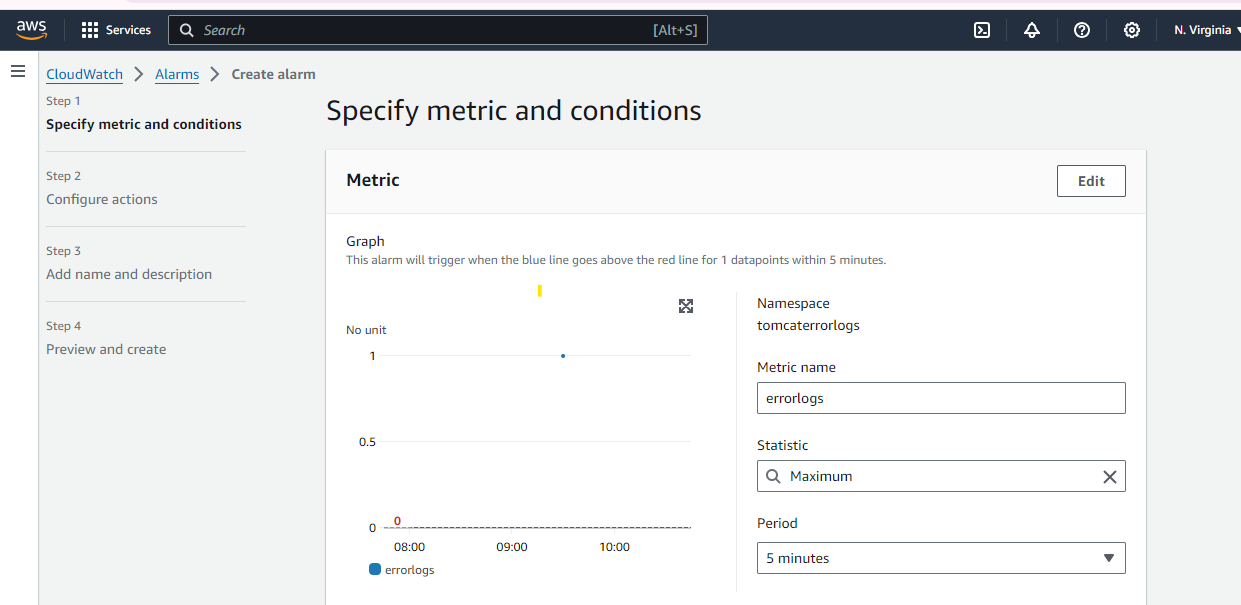




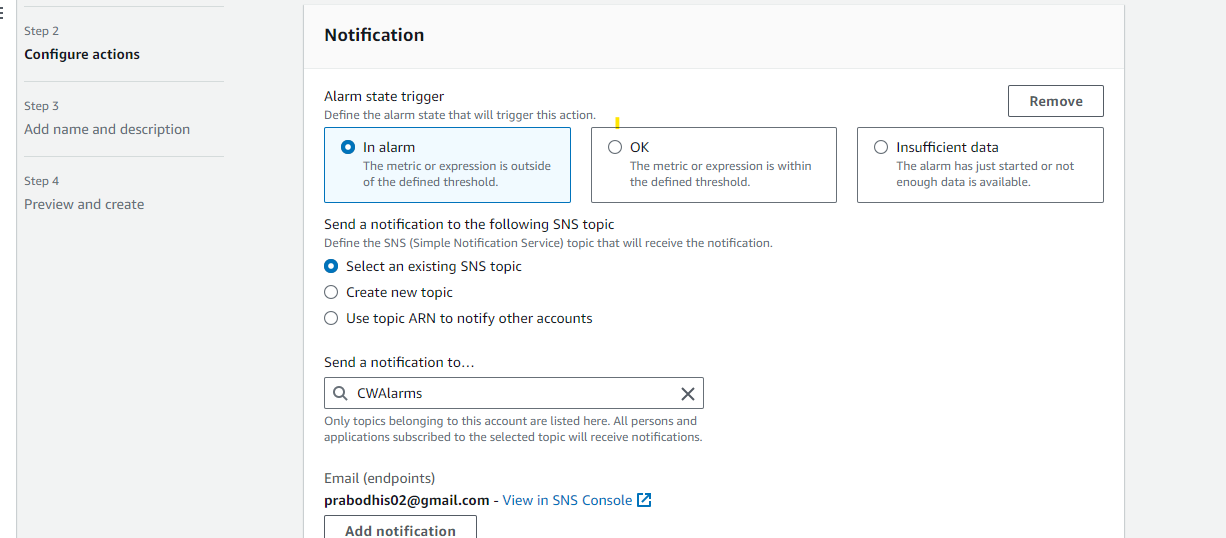


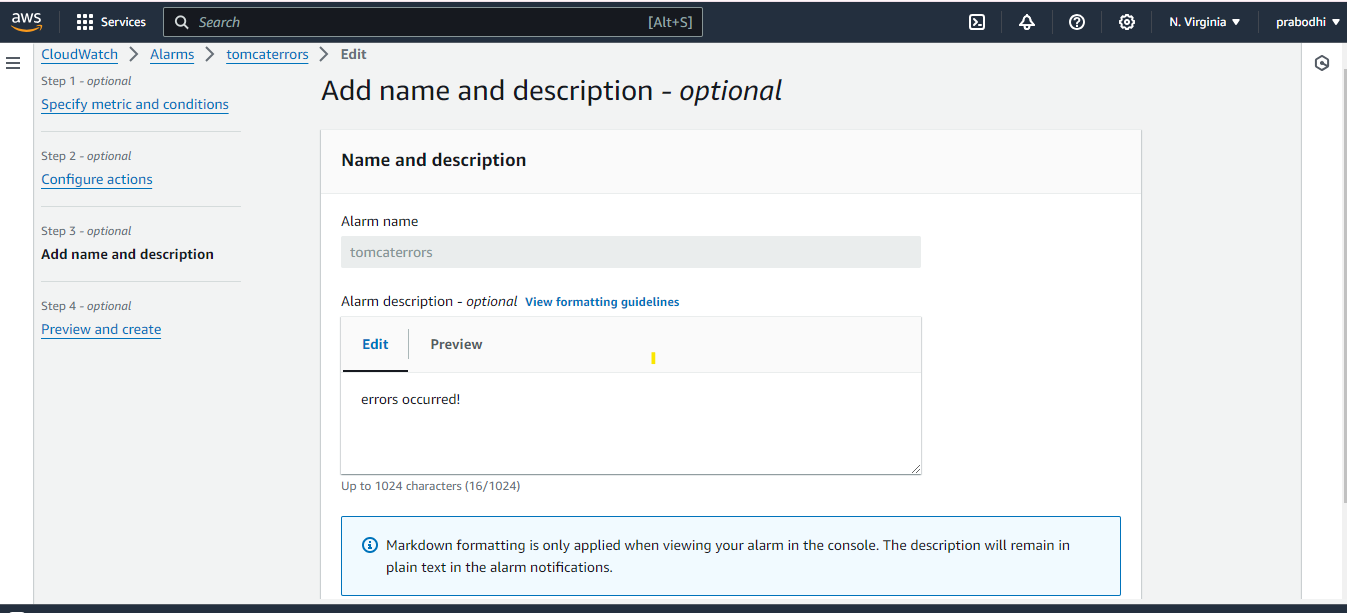
Select the created metric filter and create an alarm.

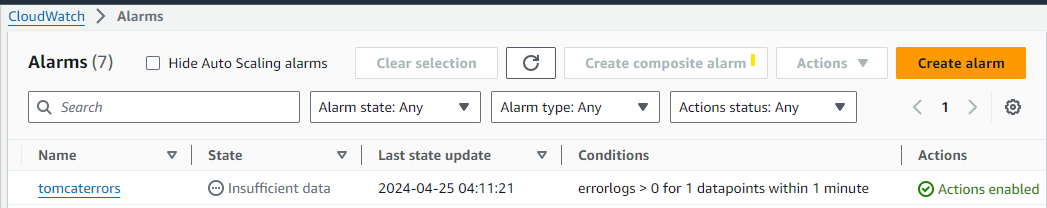




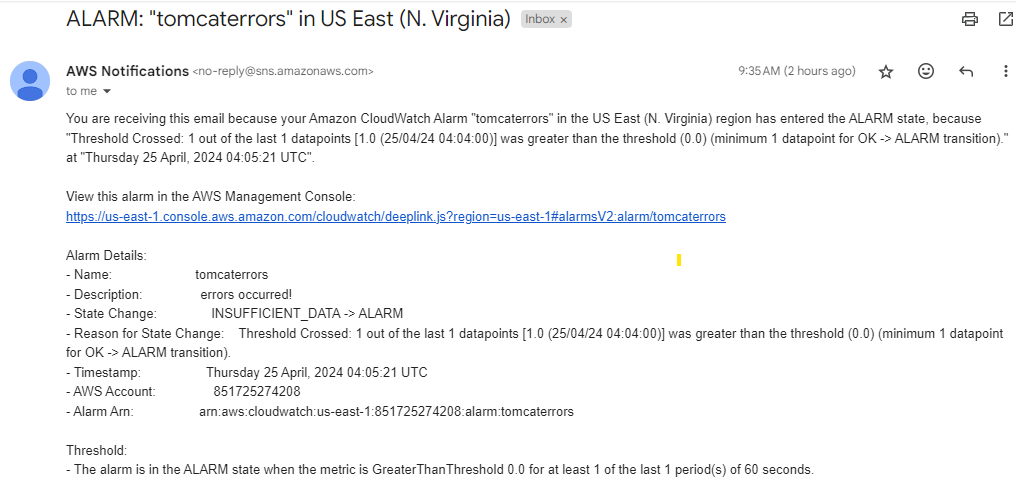
Send alerting notifications when errors are occurred in logs





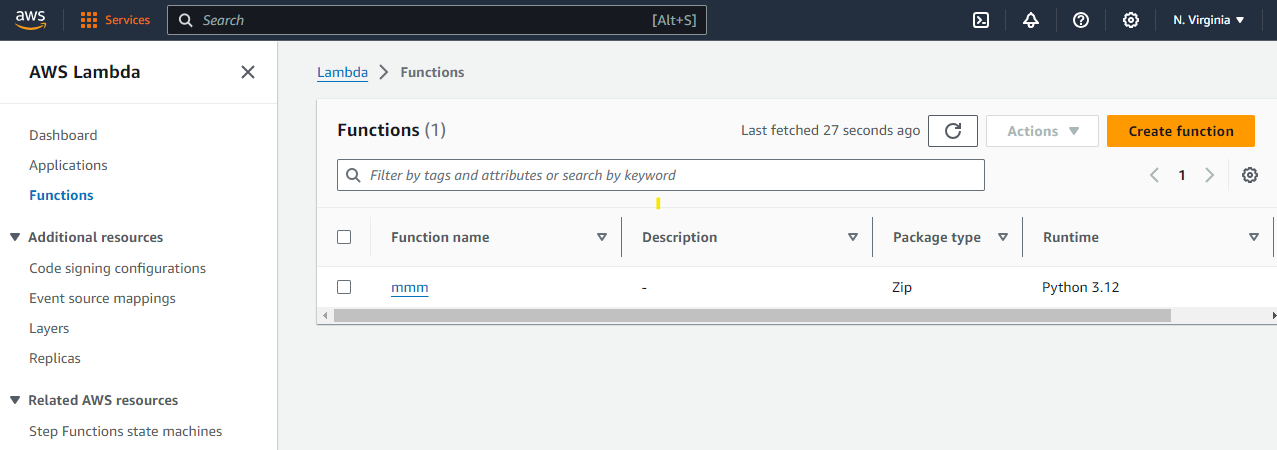


Alarm notifications are sent out when errors occurred.

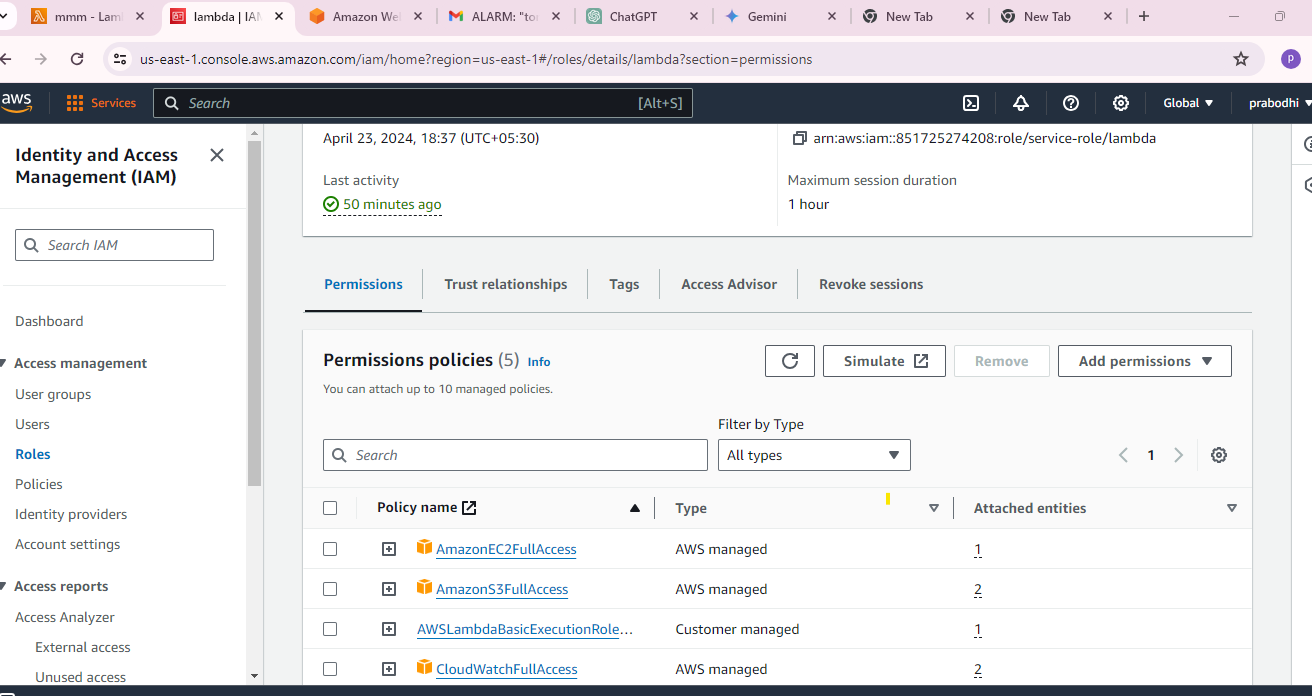


Creating a lambda function to retrieve instance stats from cloudwatch and store them as a json file inside S3 bucket.

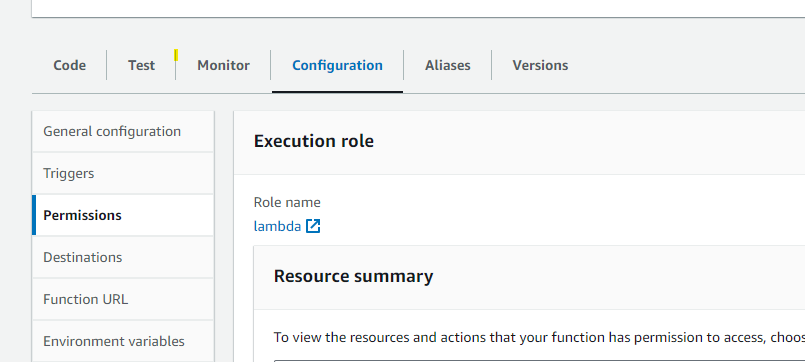
Create a lambda function. (Choose Python as runtime)



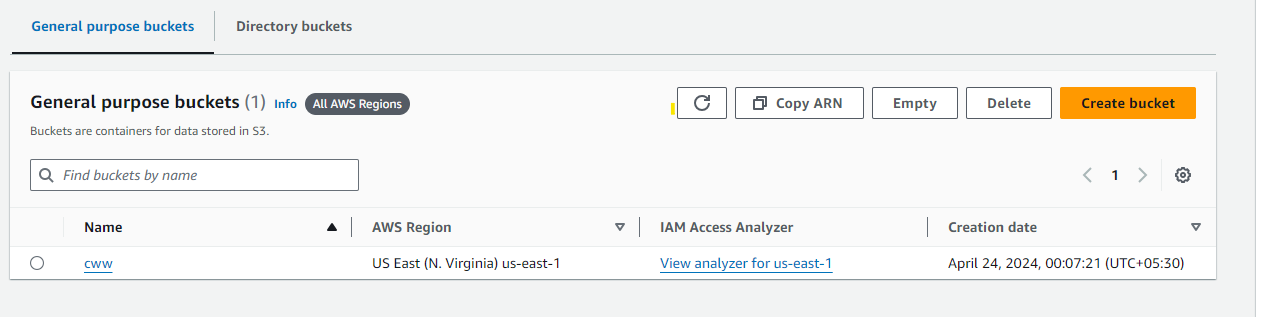
This lambda function wants to access Cloudwatch and S3 services. Create a IAM role for it.



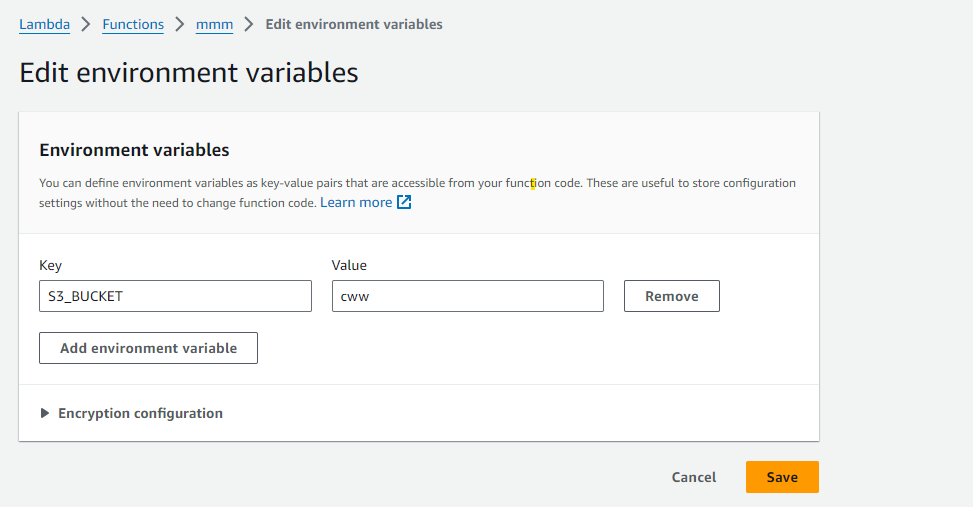
Assign created IAM role to lambda function.



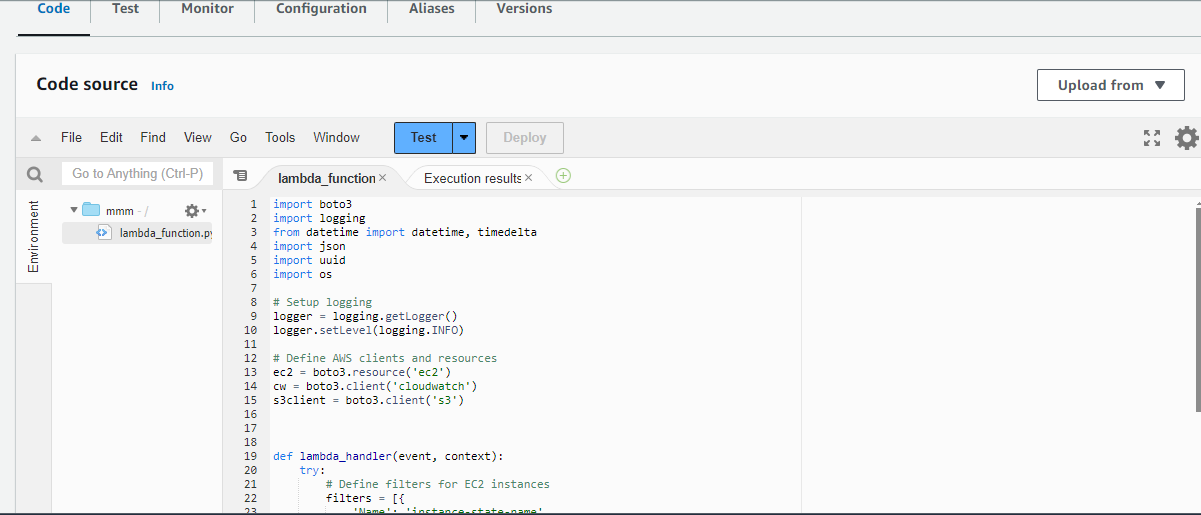
Create a S3 Bucket

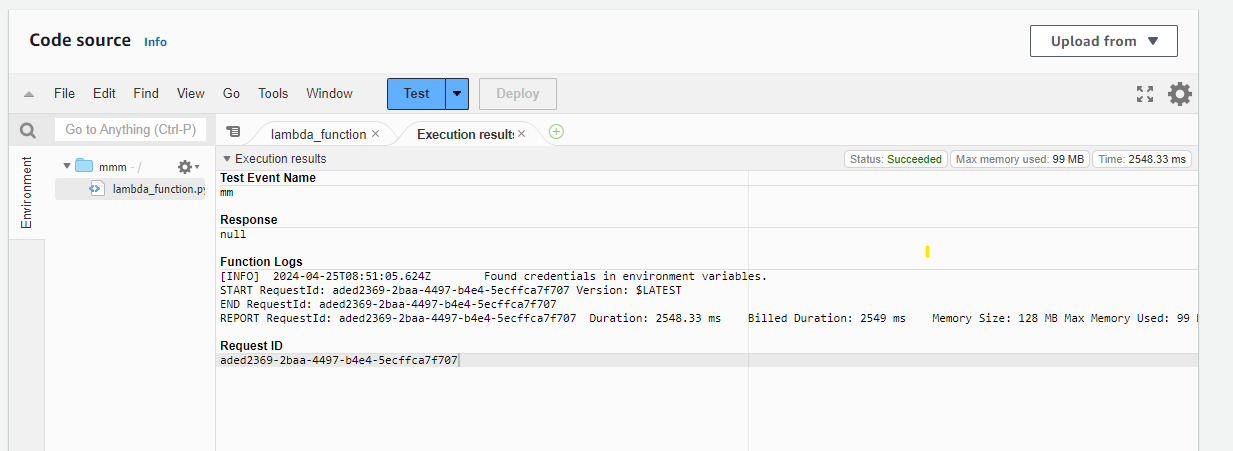


Add created s3 bucket to environment variables of lambda function.

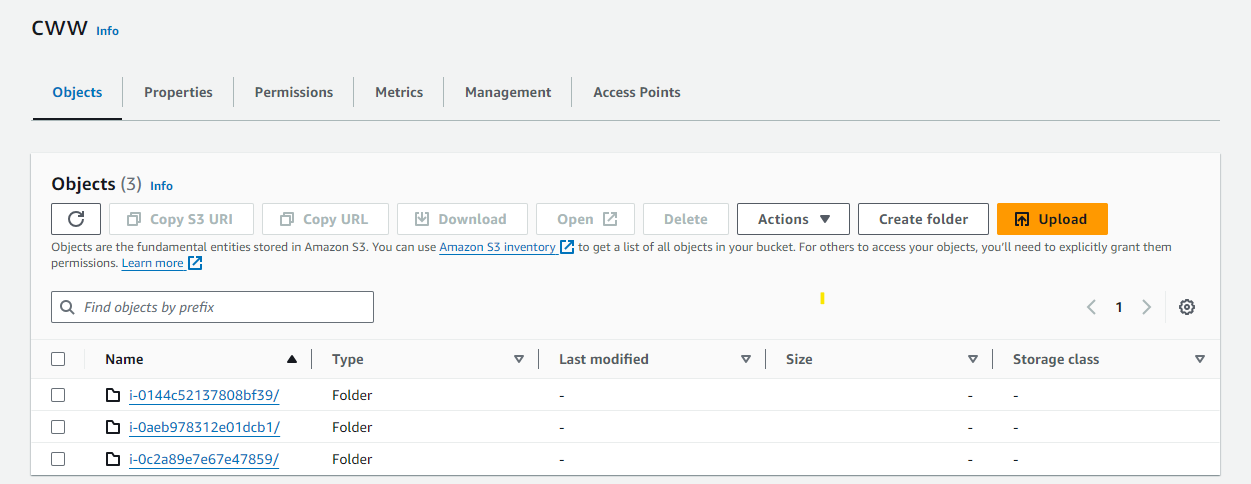


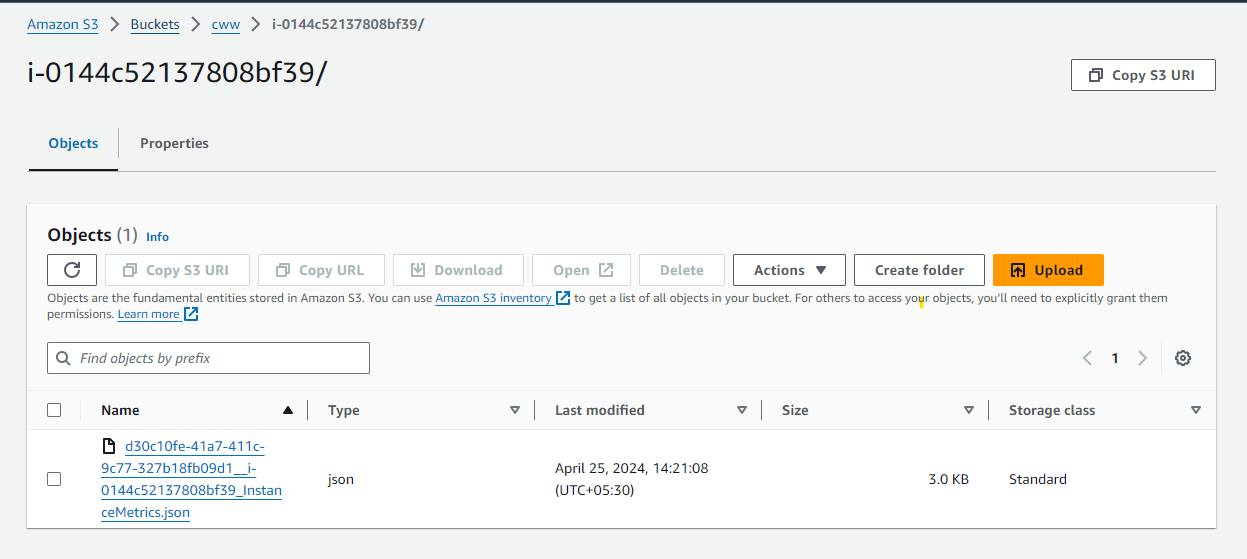
Then, deploy and test the code.





After successfully testing the code, there will be three folders holding JSON files within the assigned S3 bucket.





Optional Requirements

1. Logging and metrics standard for key stats

**Alarming levels for each metric**:

Warn/ Amber : 60% threshold

Critical : 80% threshold

**Use consistent unit for each metric** : Percentage (for CPU, network, memory and disk utilization)

**Log-retention policy:**

Store logs for 60-90 days

Archive older logs (4months – 10 months ) in a separate storage location.

**Secure Logging** : Avoiding sensitive information in logs

**Log Group Structure : Setup log groups by considering application, environment ..etc.**