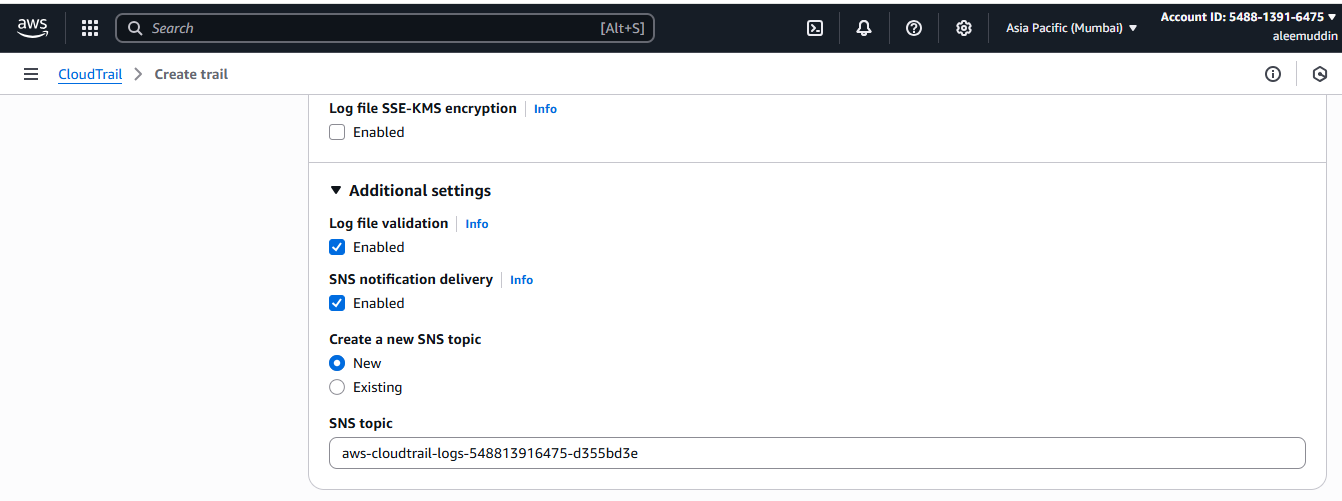
1. **Enable cloudtrail monitoring and store the events in s3 and cloudwatch log events.**

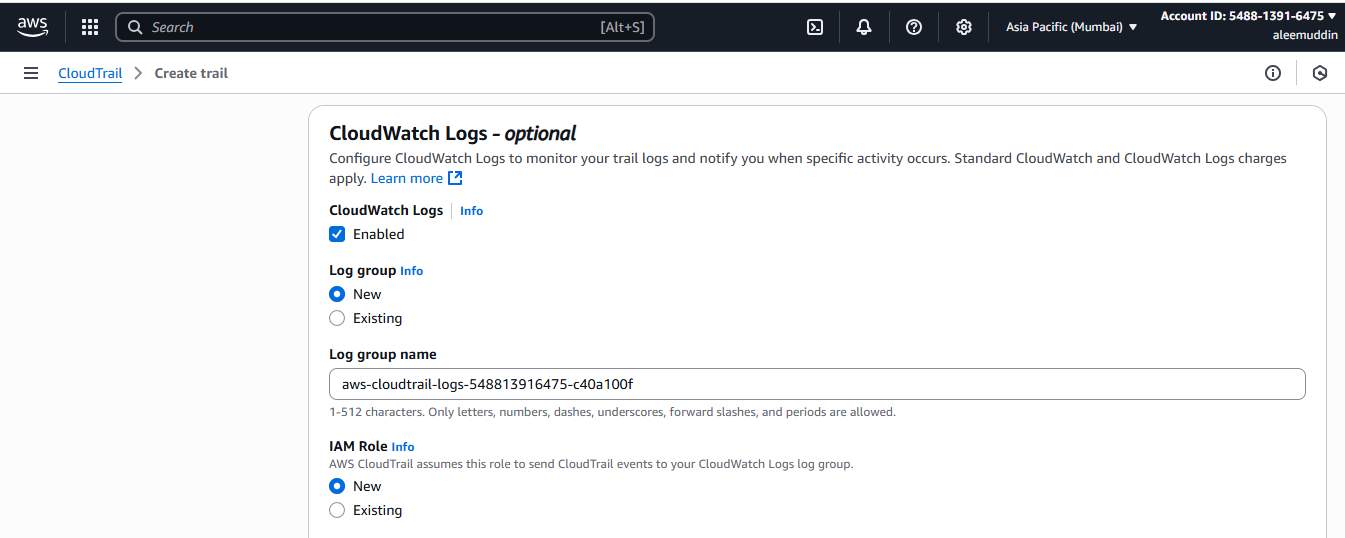
**Go to cloudtrail to create the Management/Data & Insight events, s3 bucket and enabling cloudwatch log events.**

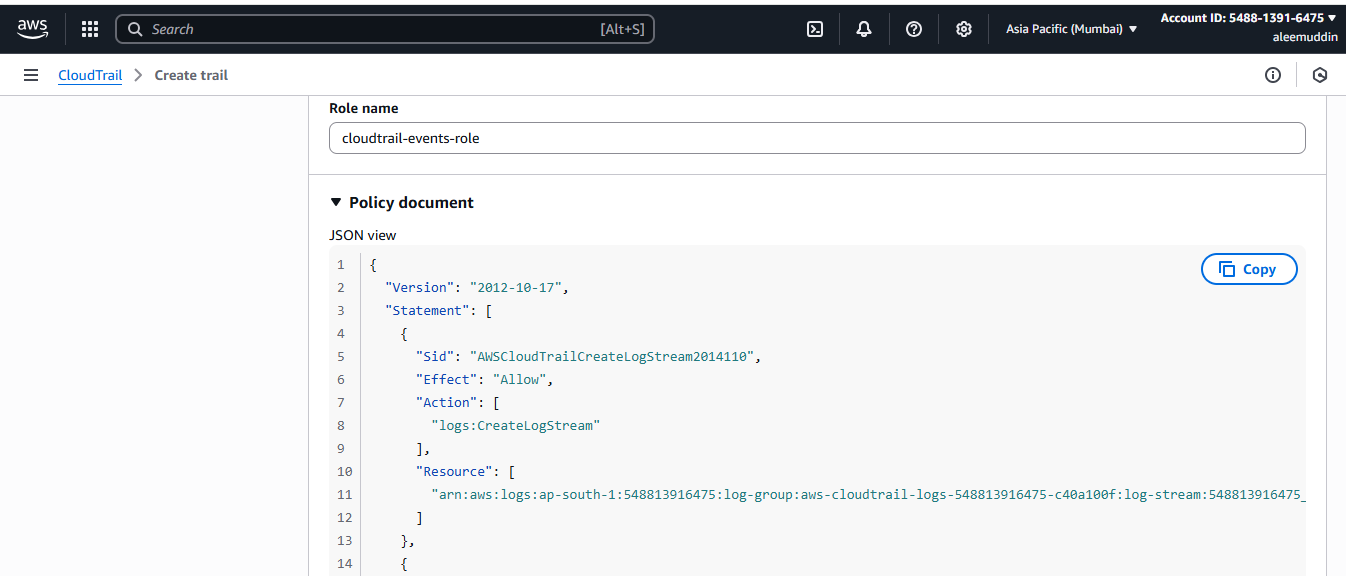
1. **Create trail from cloud trail**
2. **Select new s3 bucket, enable log validation, SNS Notification, select SNS topic and new cloudwatch.**
3. **Go to SNS service create topic creat subscriptions select Protocol as mail and give mail id and create.**
4. **Check mail and confirm then SNS created.**
5. **Now came to CT tab and select SNS existing wich is created by us.**
6. **Disable Log file SSE-KMS encryption.**
7. **Select any one or select Management events.**

**Create trail.**

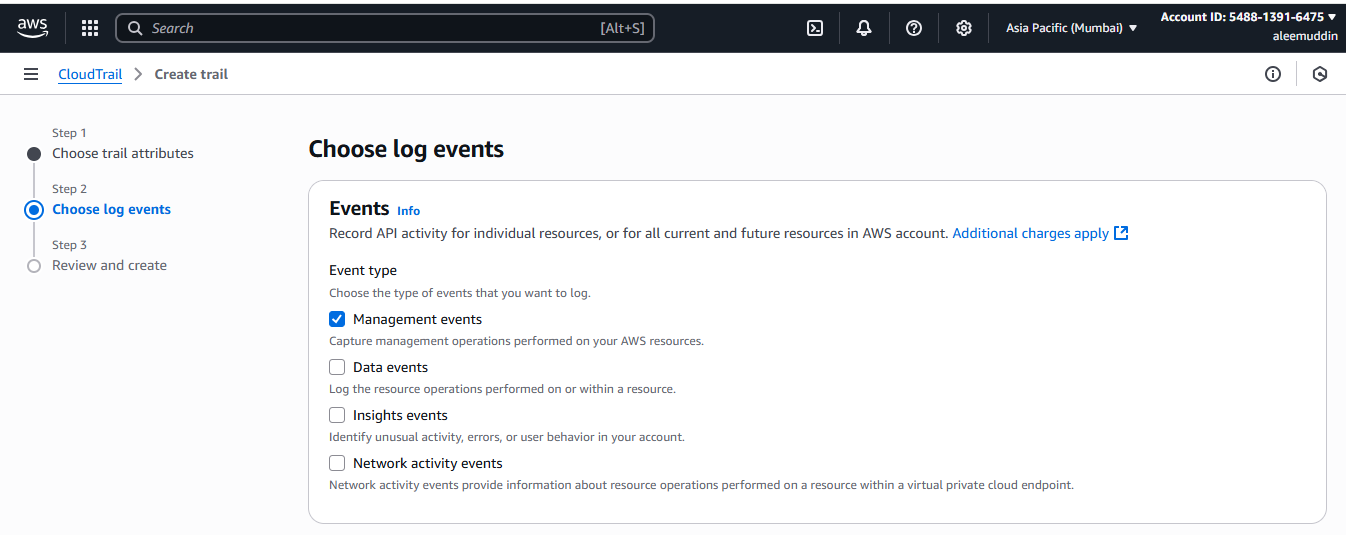


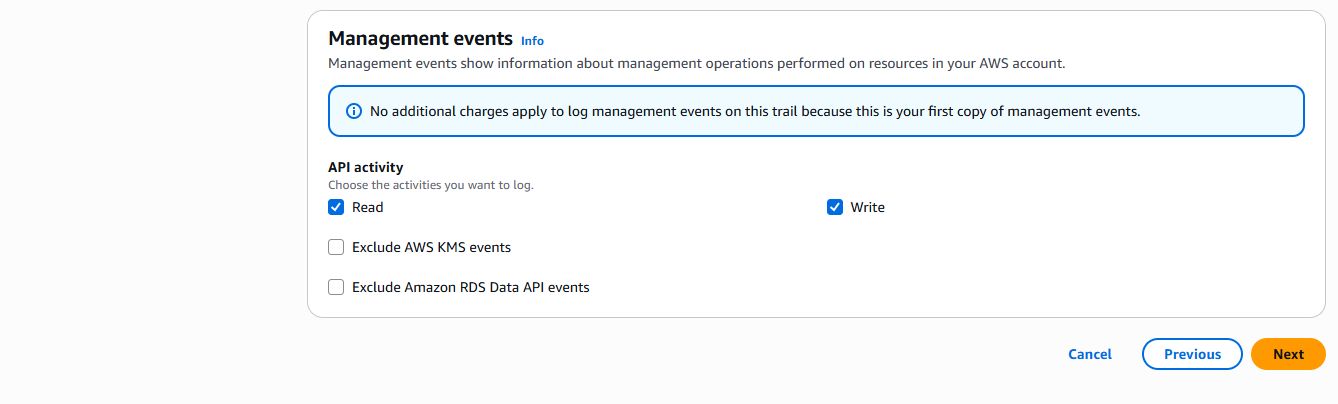


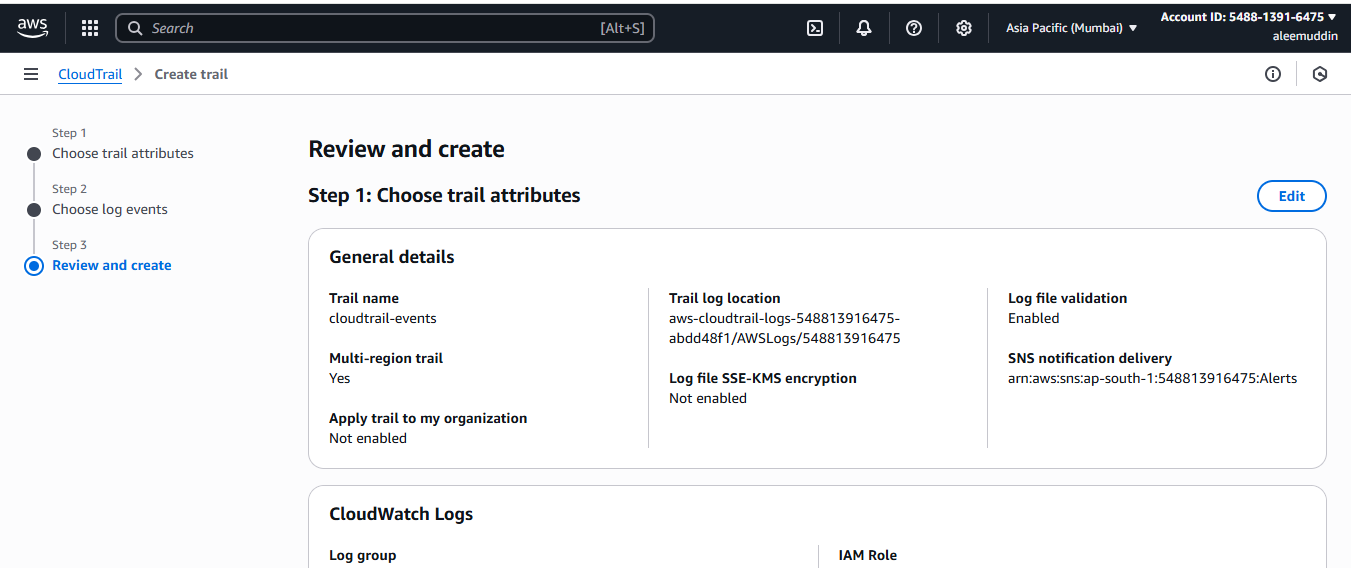




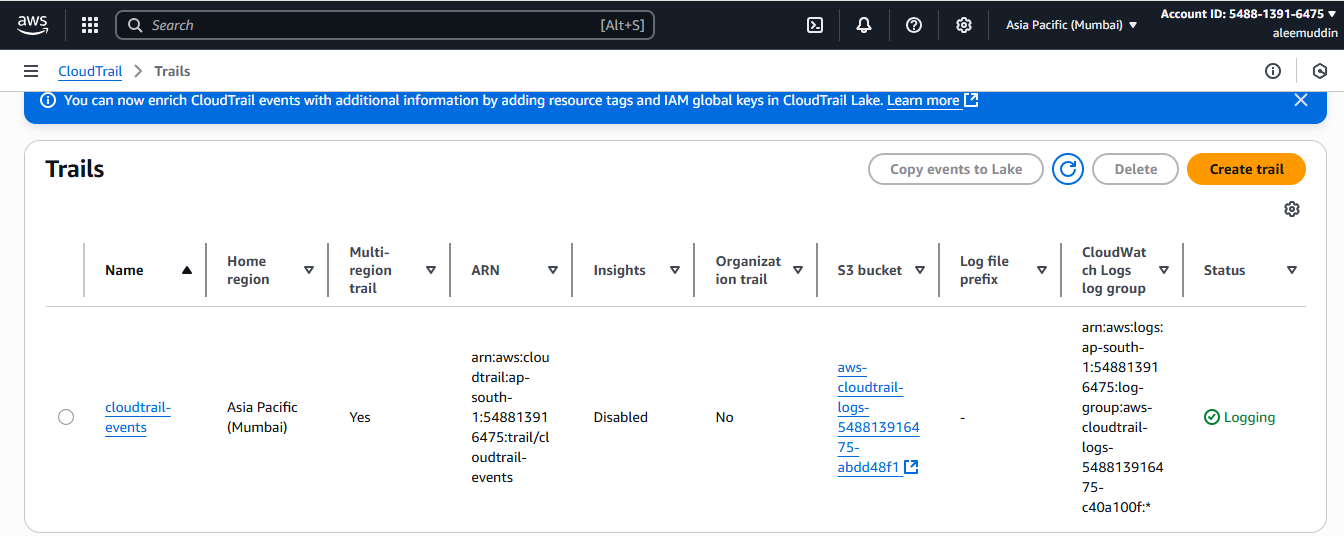
**Click to Next**



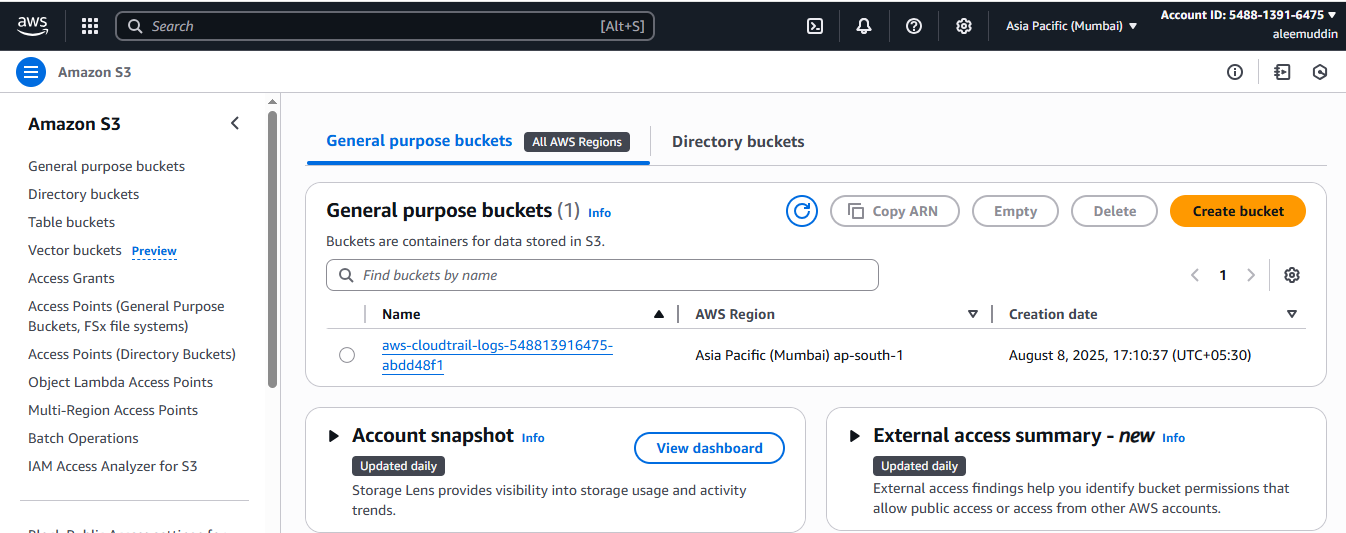


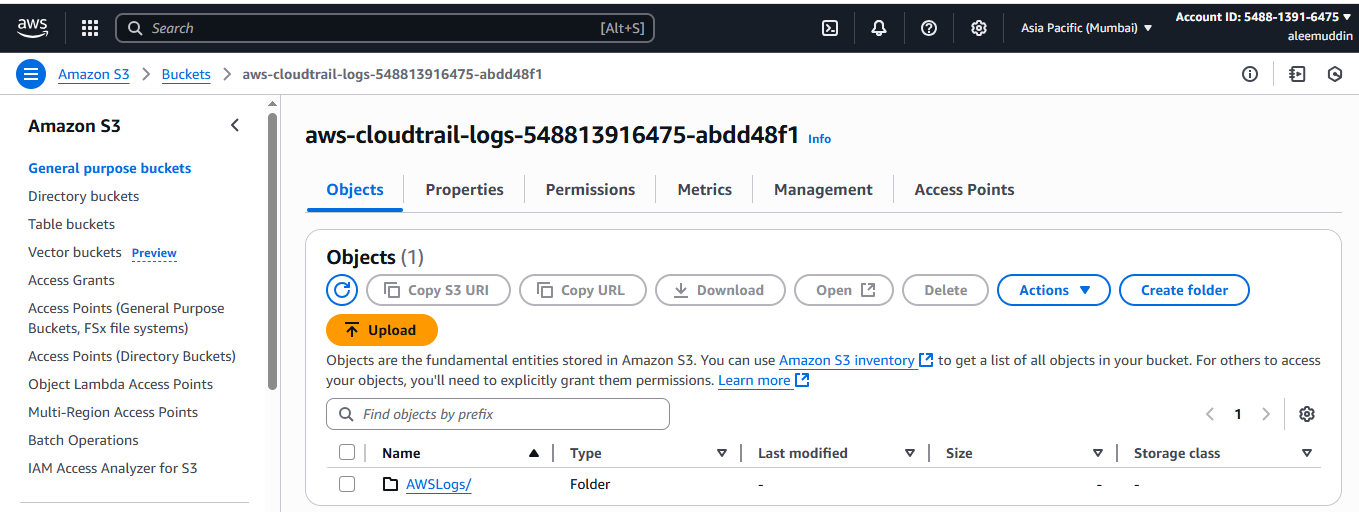


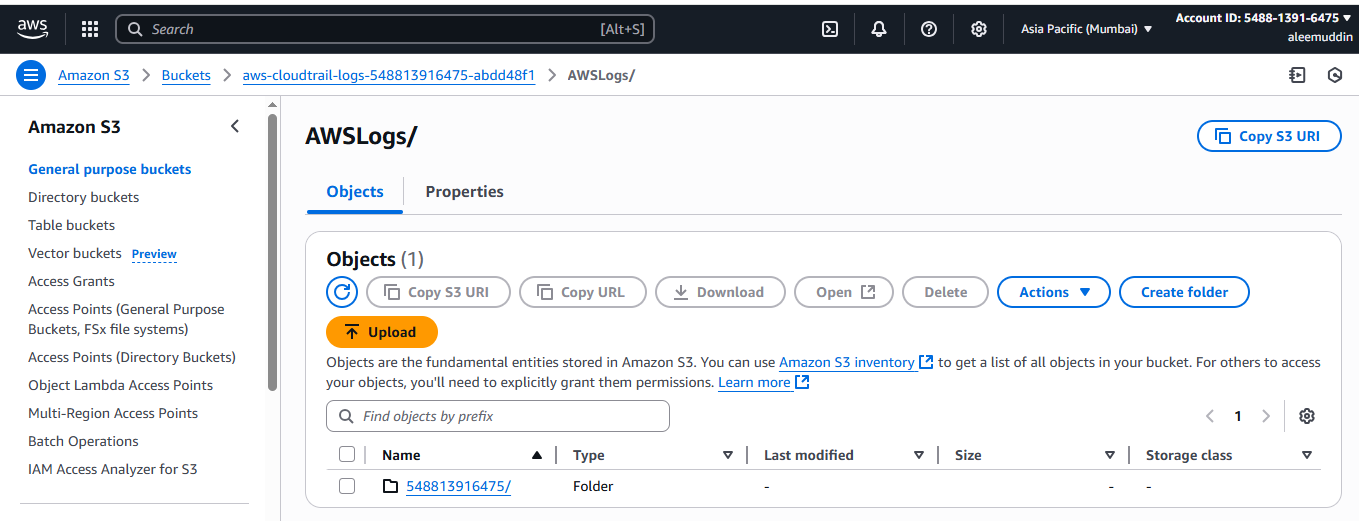
**Click to create trail**

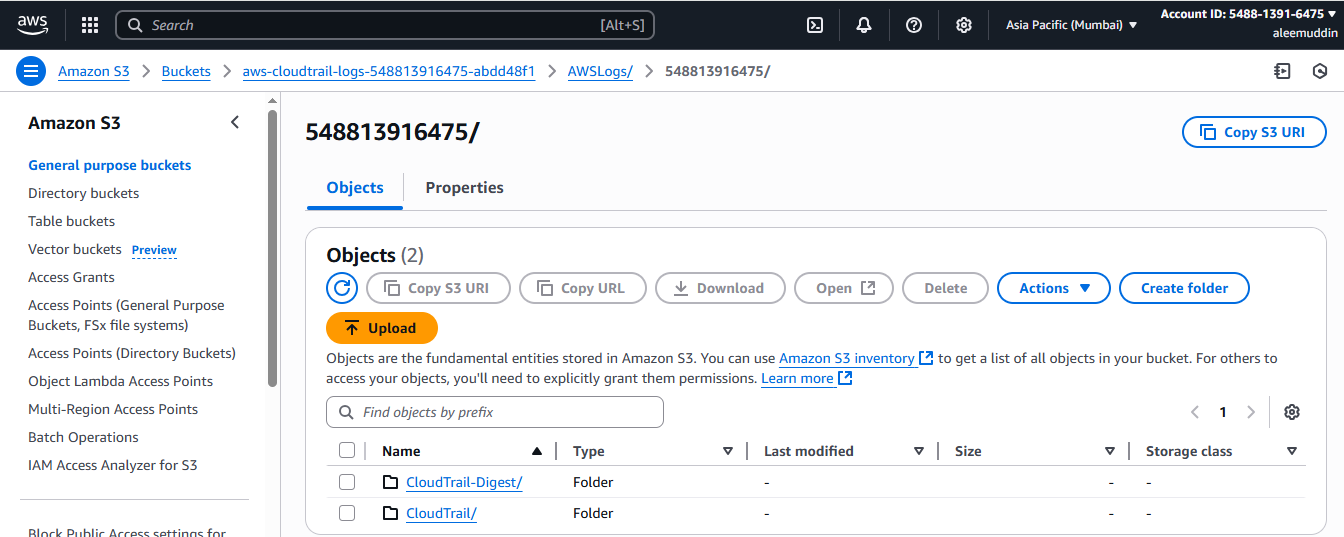


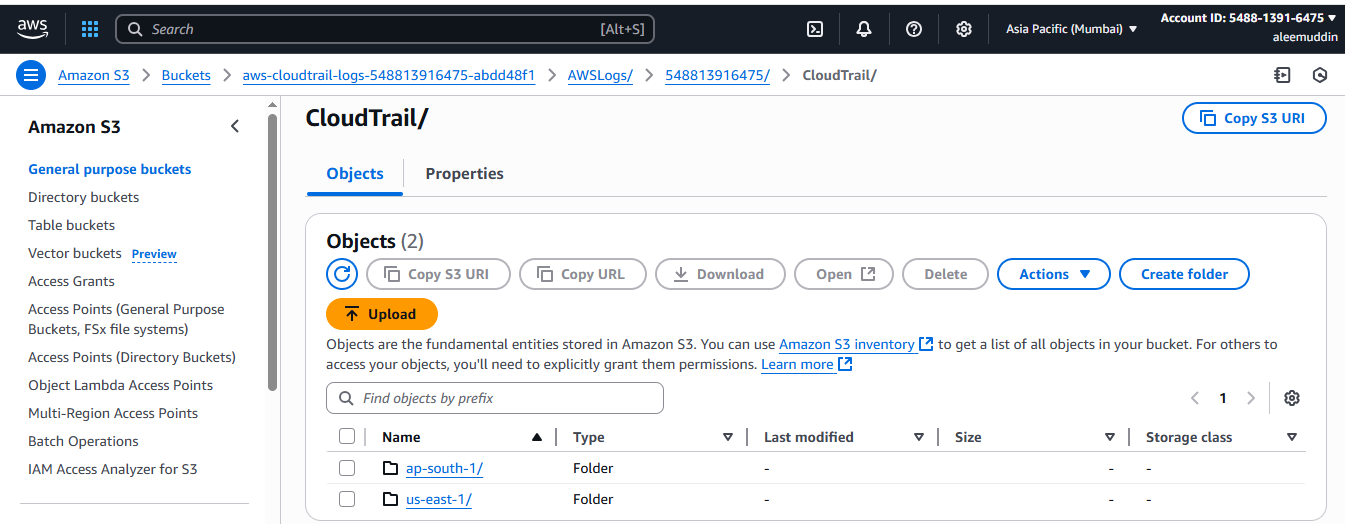
**Go to s3 to check created bucket and it objects**

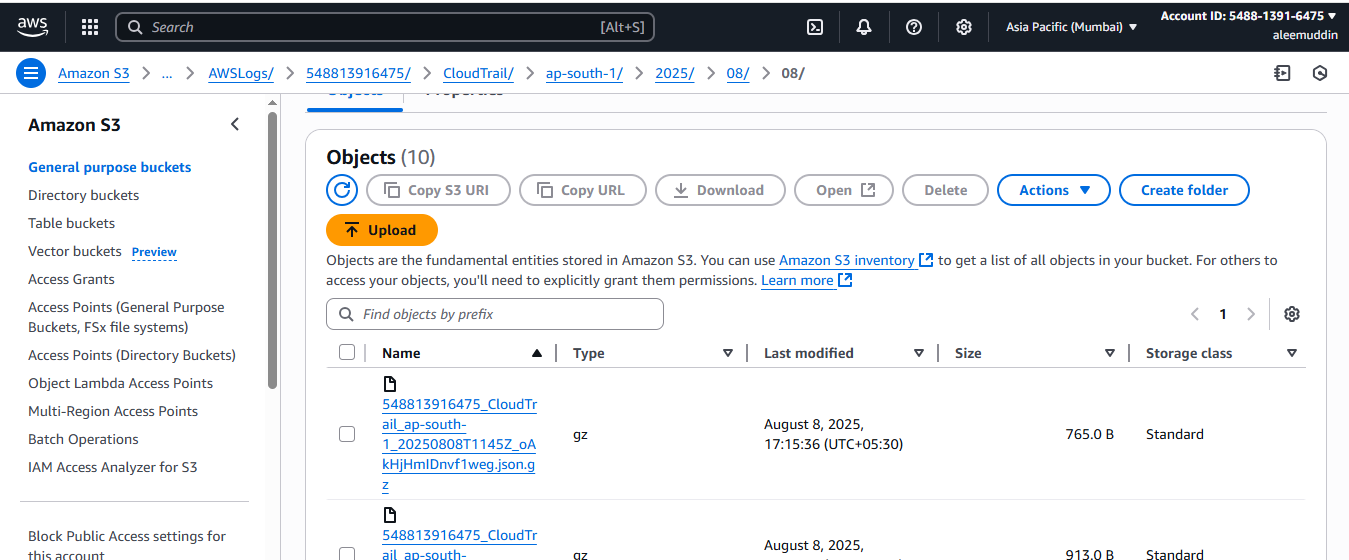




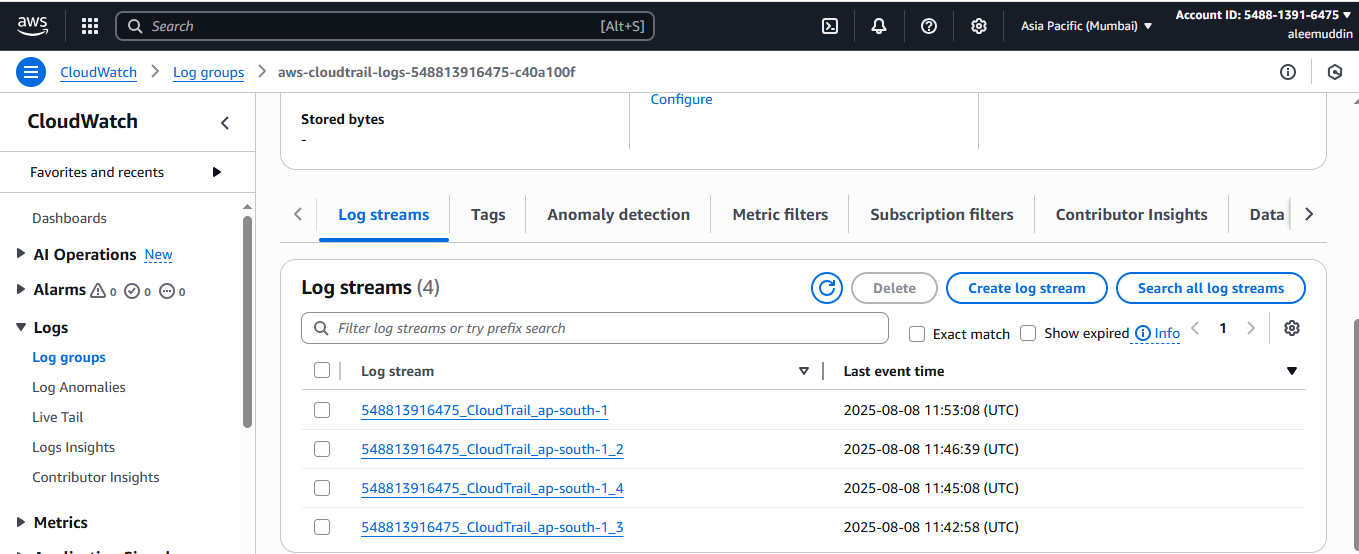


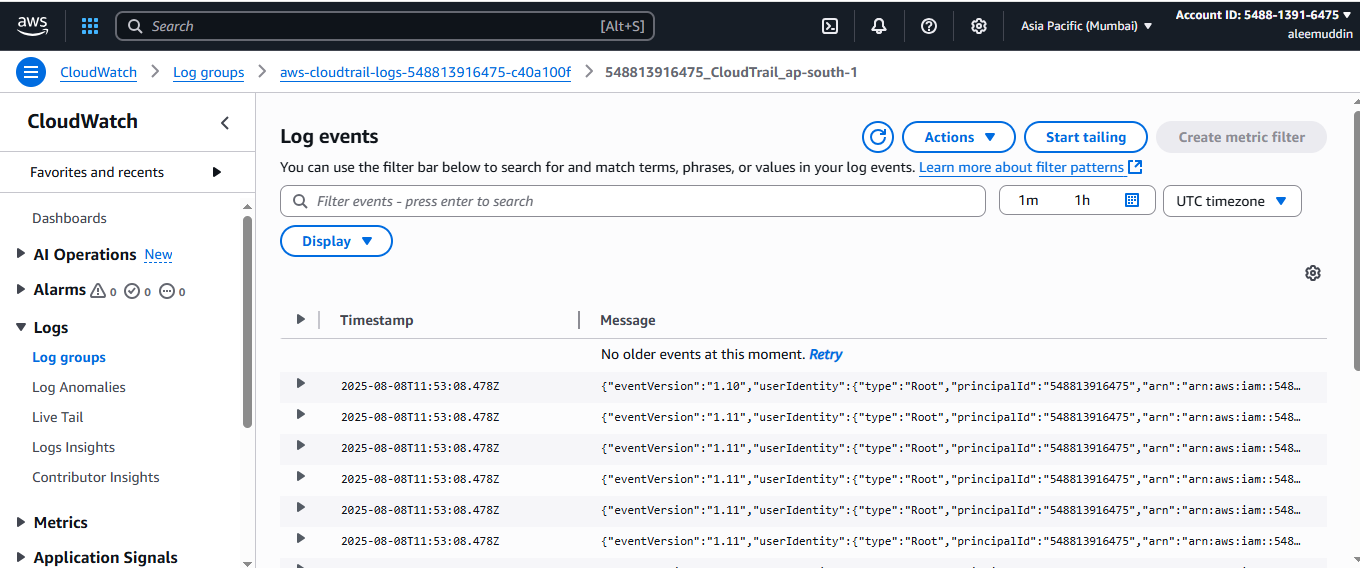


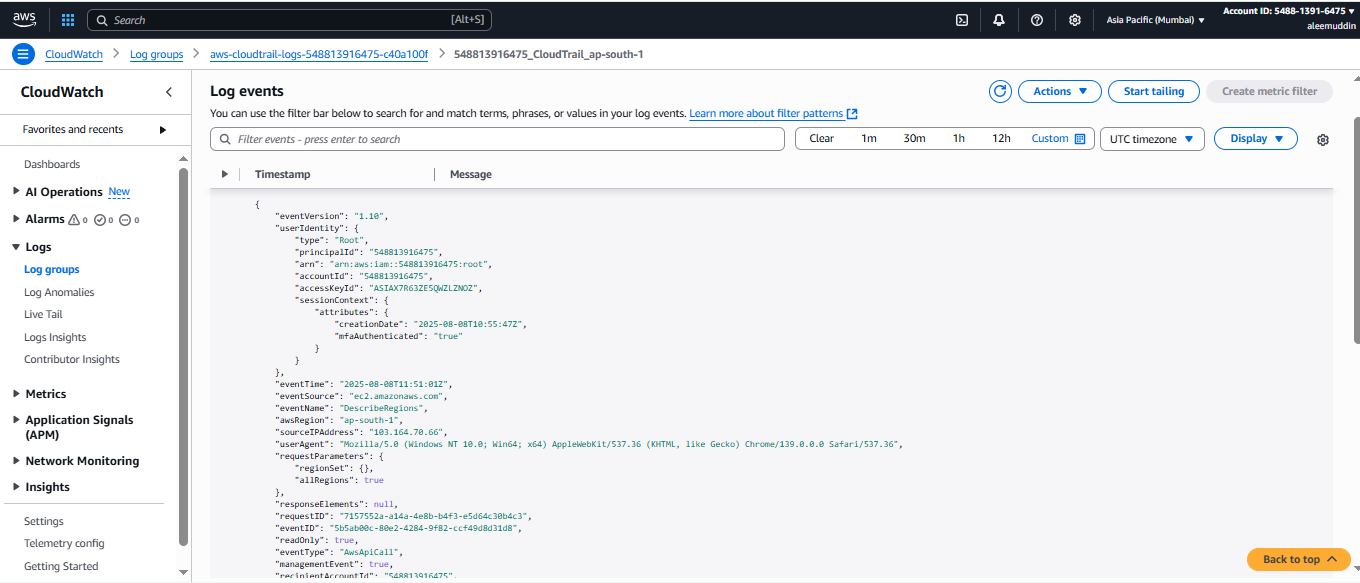




**Go to cloudwatch to check the cloudtrail log stream**

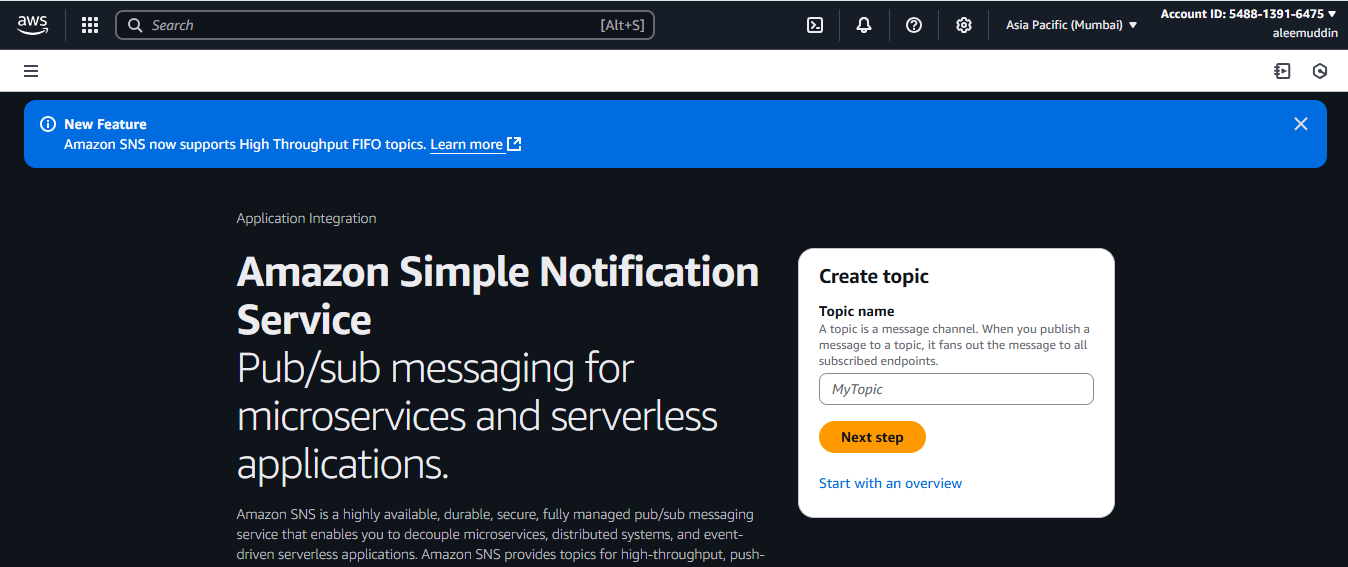




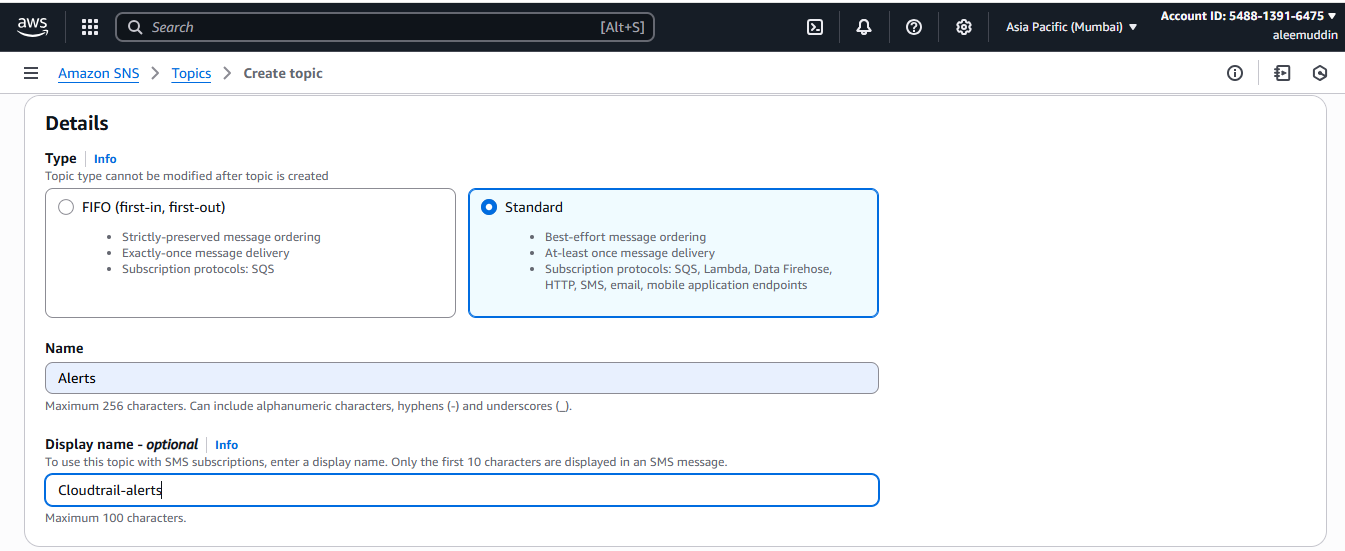


1. **Enable SNS for cloudtrial to send alert on email.**

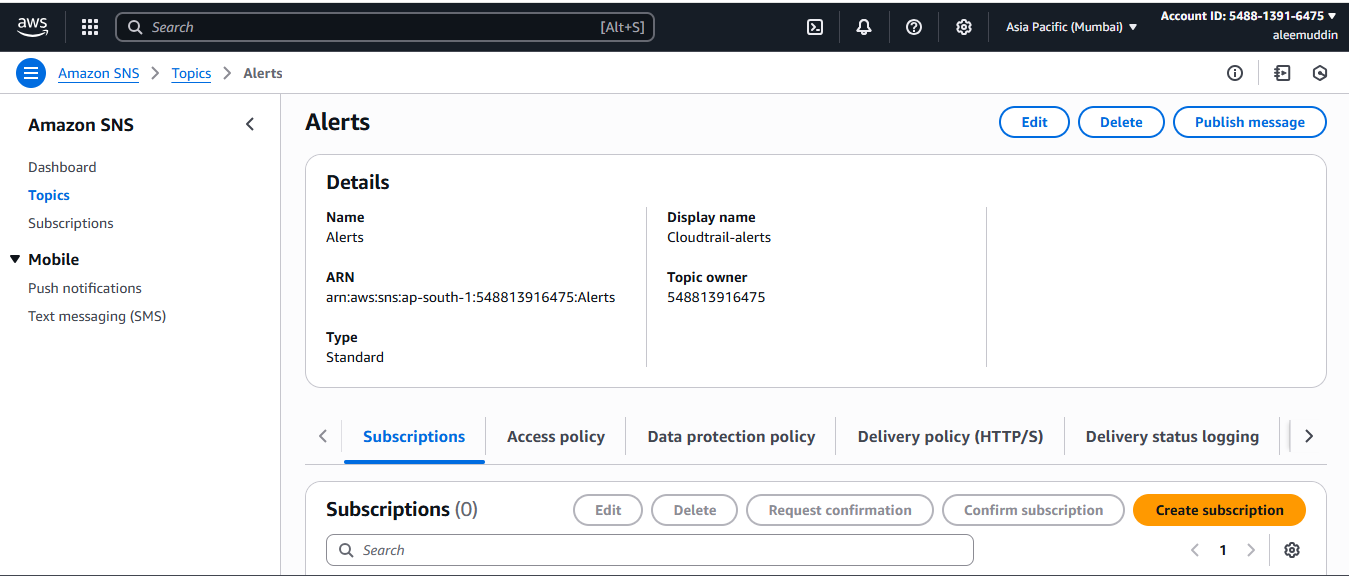
**GO to AWS search bar and type SNS**



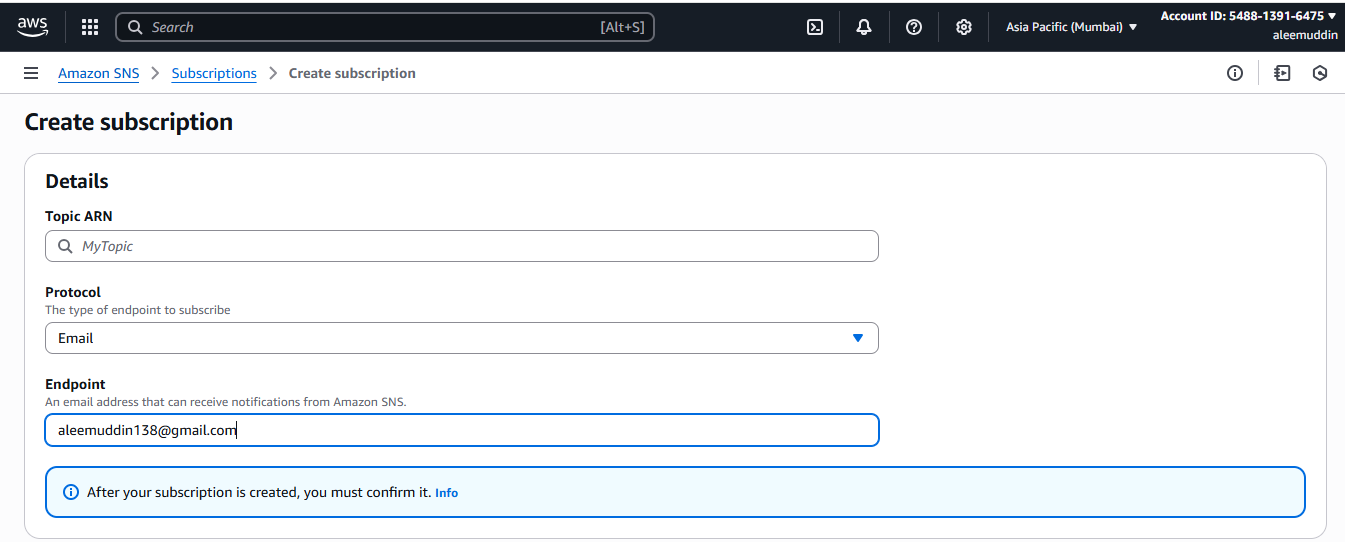
**Click on next step to create topic name**



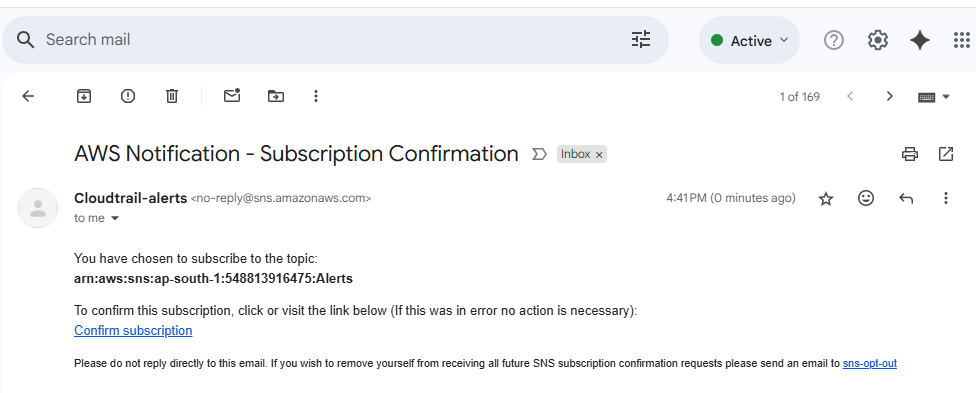
**Click Create topic**



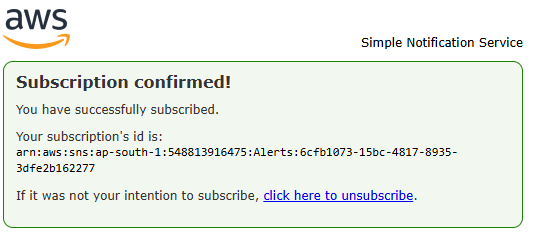
**Click on Create Subscription**



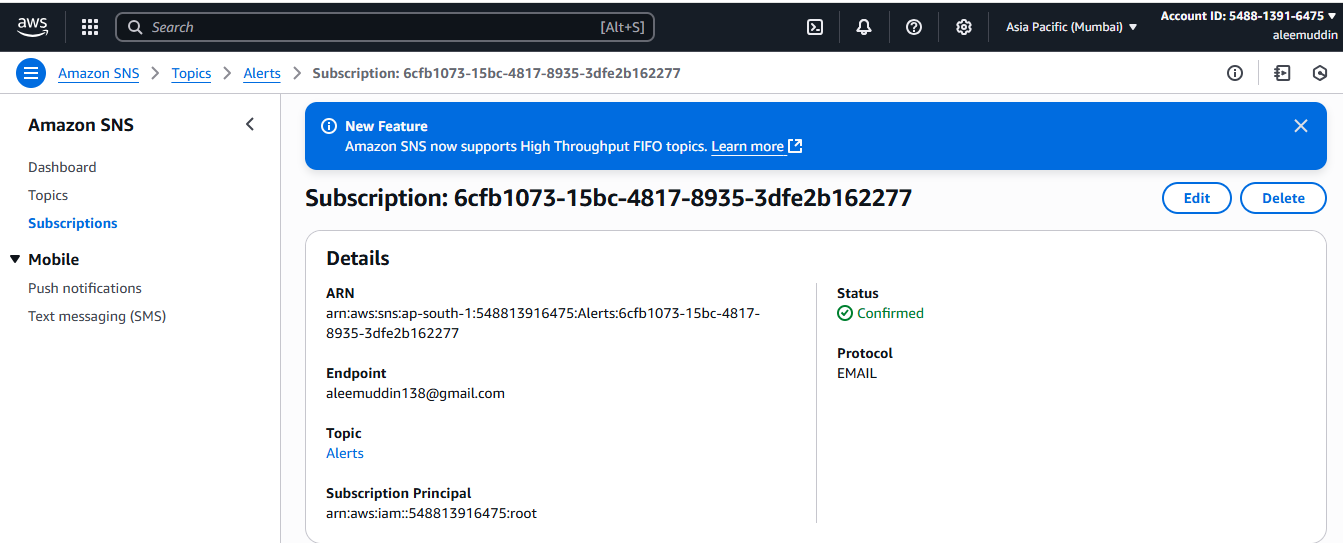
**Click on create subscription and will get the conformation Email.**



**Click on confirm subscription to confirm**



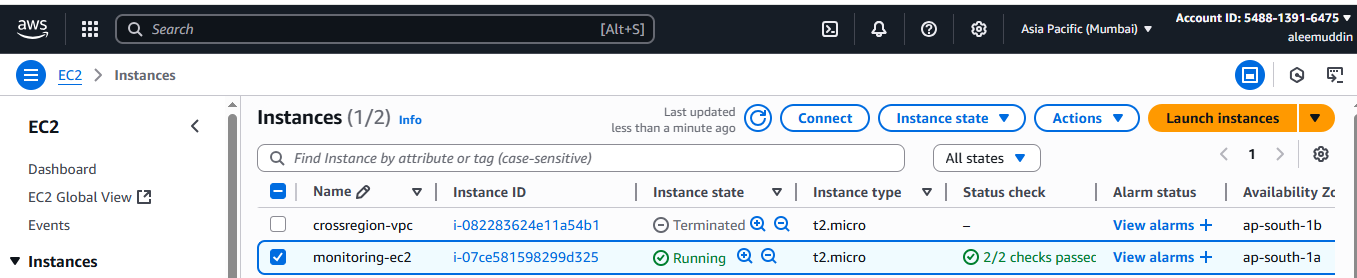
**Refresh the SNS page to change the status confirm**



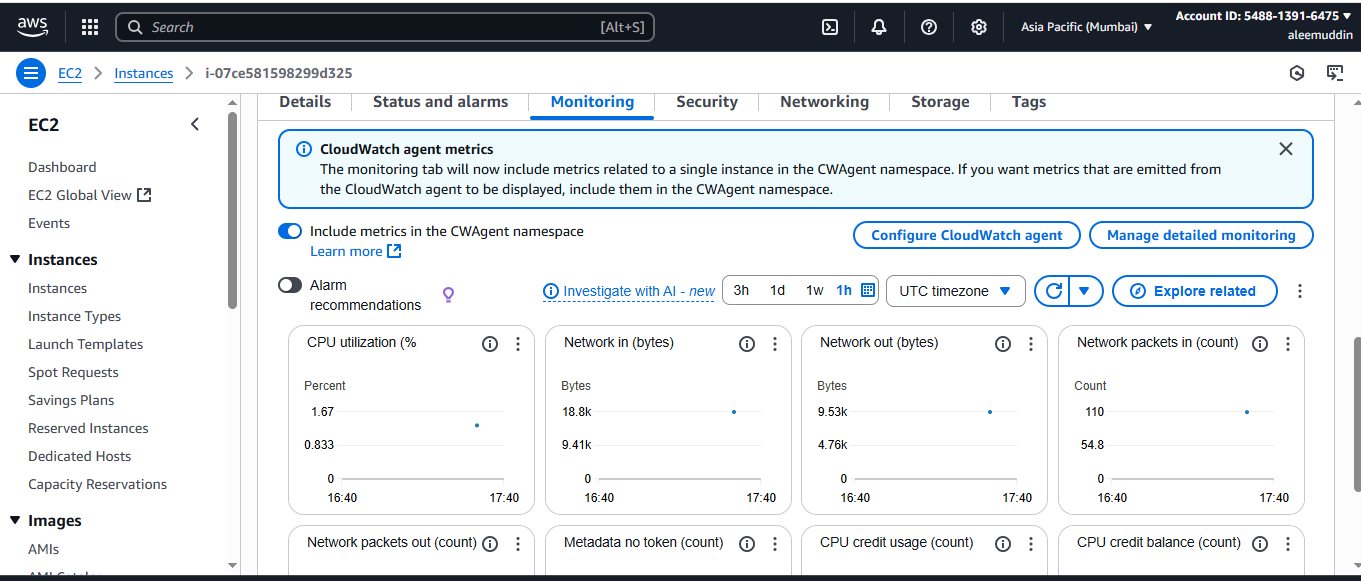
**Task is completed**

1. **Configure cloud watch monitoring and record the cpu utilization and other metrics of ec2.**

**GO to ec2 & Launch one instance name as a monitoring-ec2 metrics**

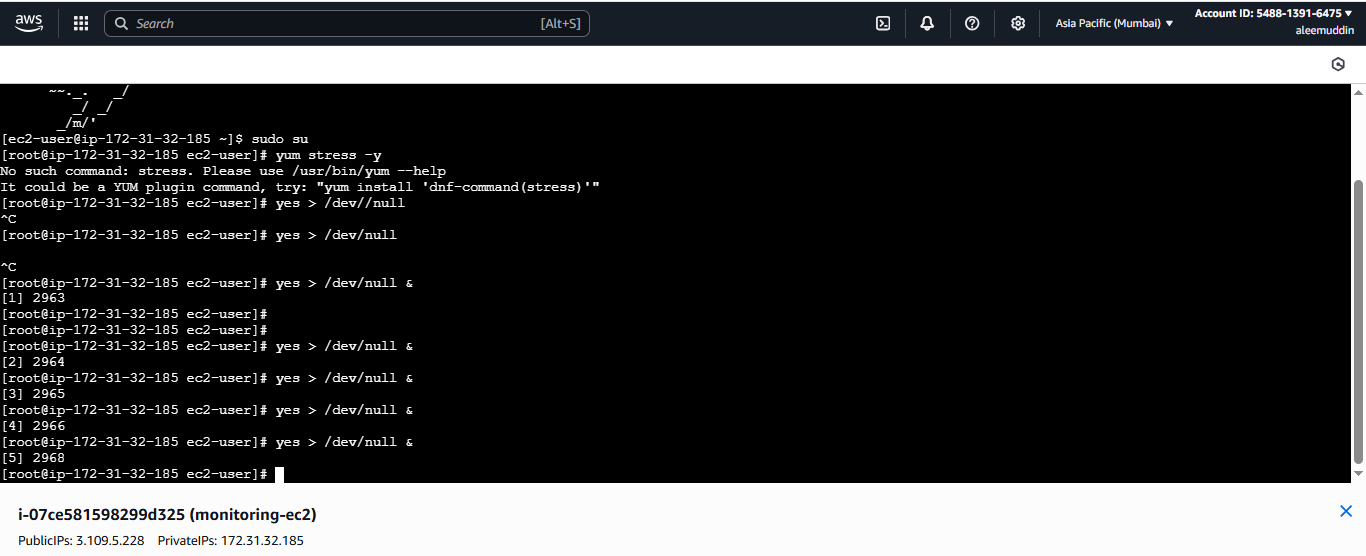


**Go to instance> Click on monitoring to get the metrics without CPU load**

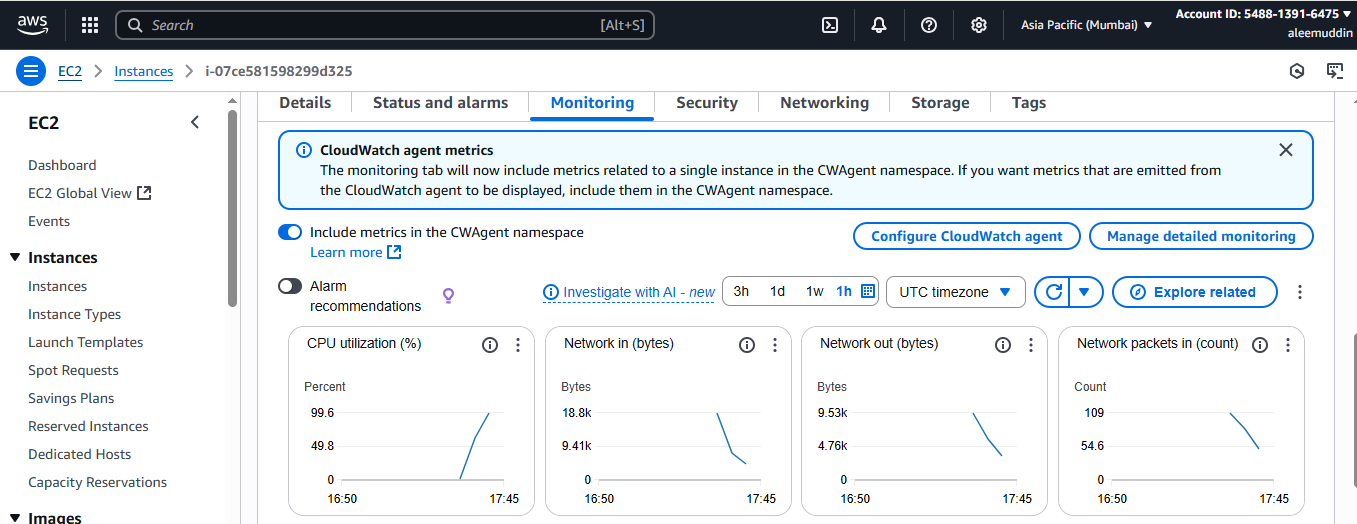


**Connect to the instance>Switch to root user>Increase the CPU load by using Linux command**

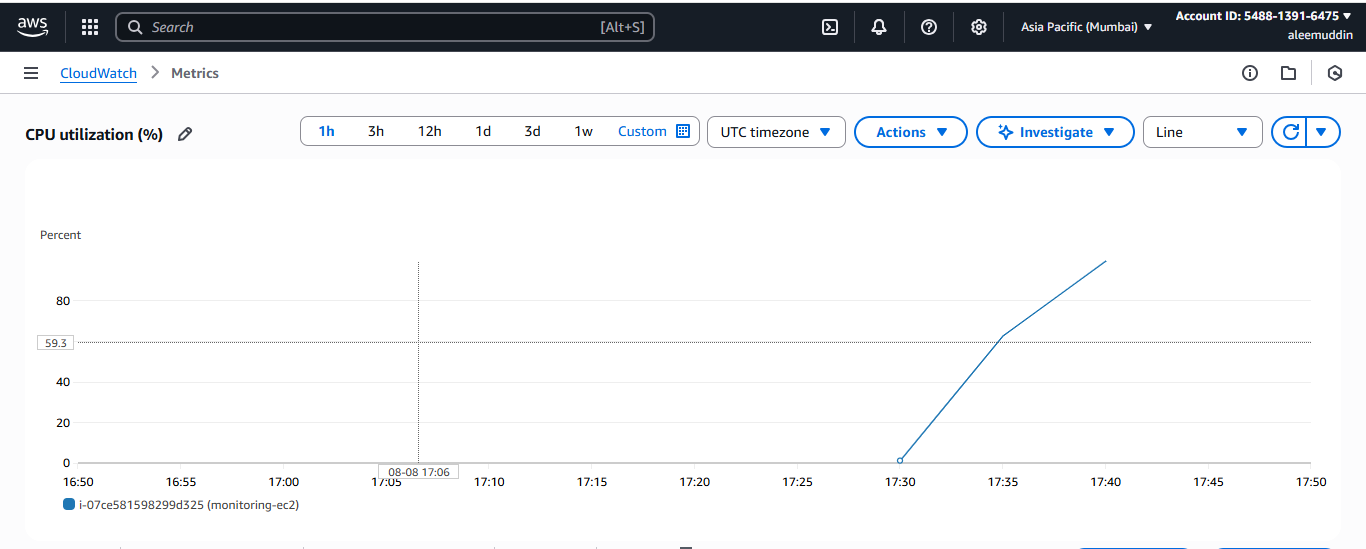
**Yum install stress –y or yes > /dev/null &**

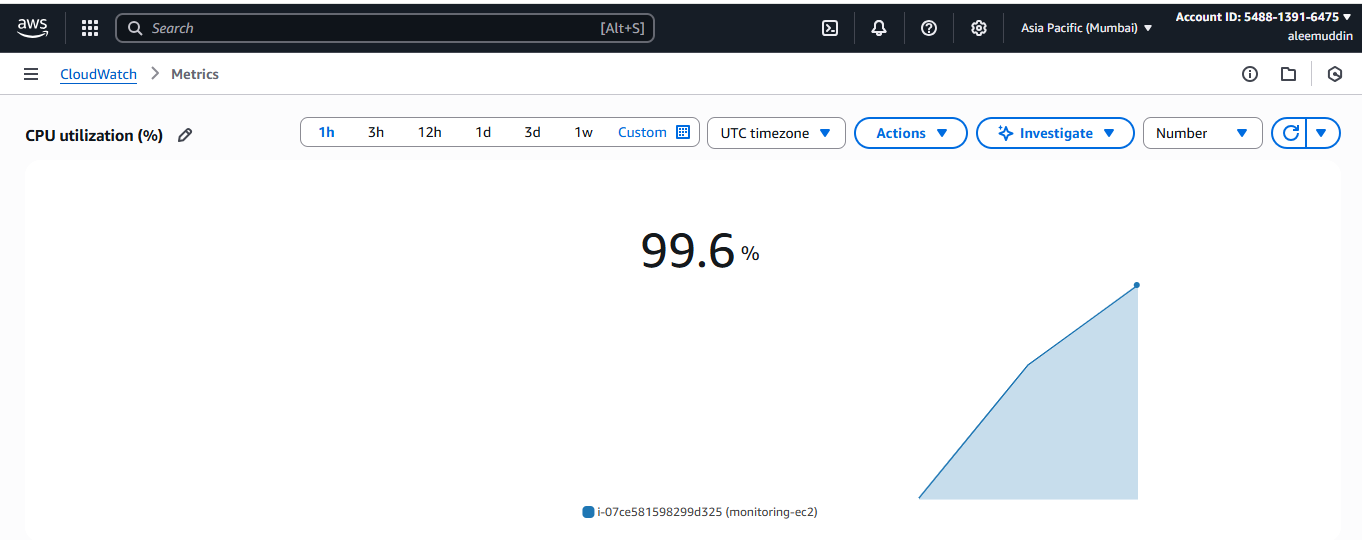


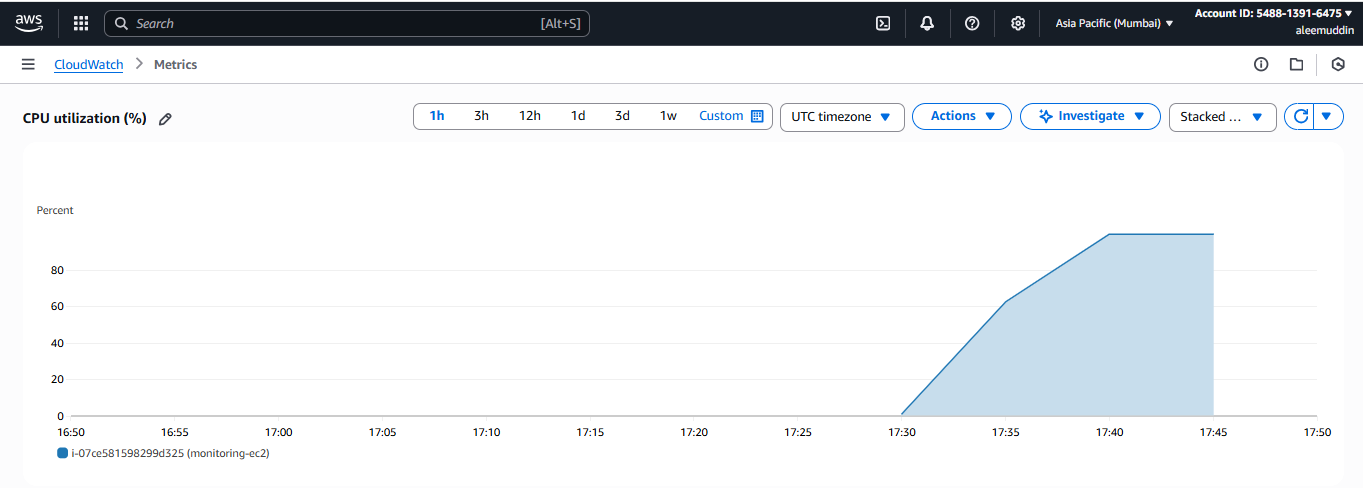
**After increasing the load we can get graphed matrics**

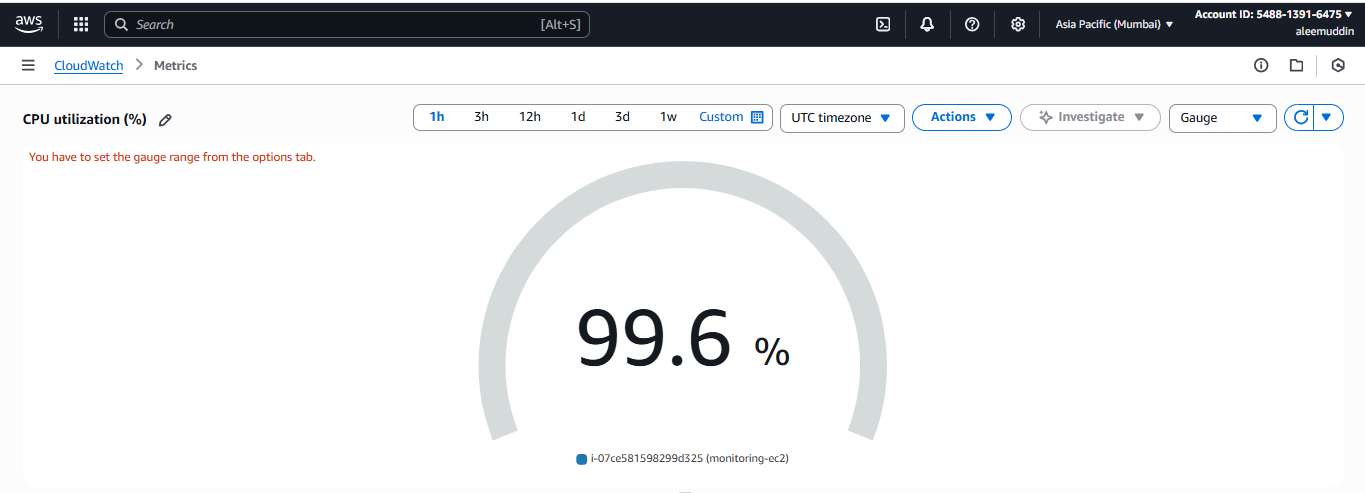


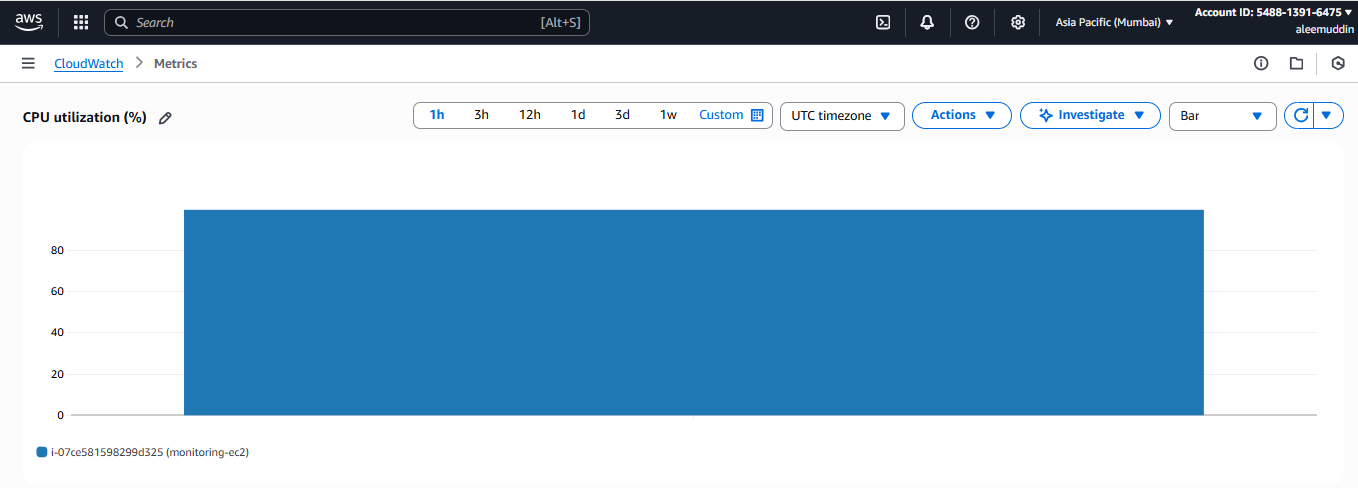
**Select CPU utilization metrics and view different graphs**

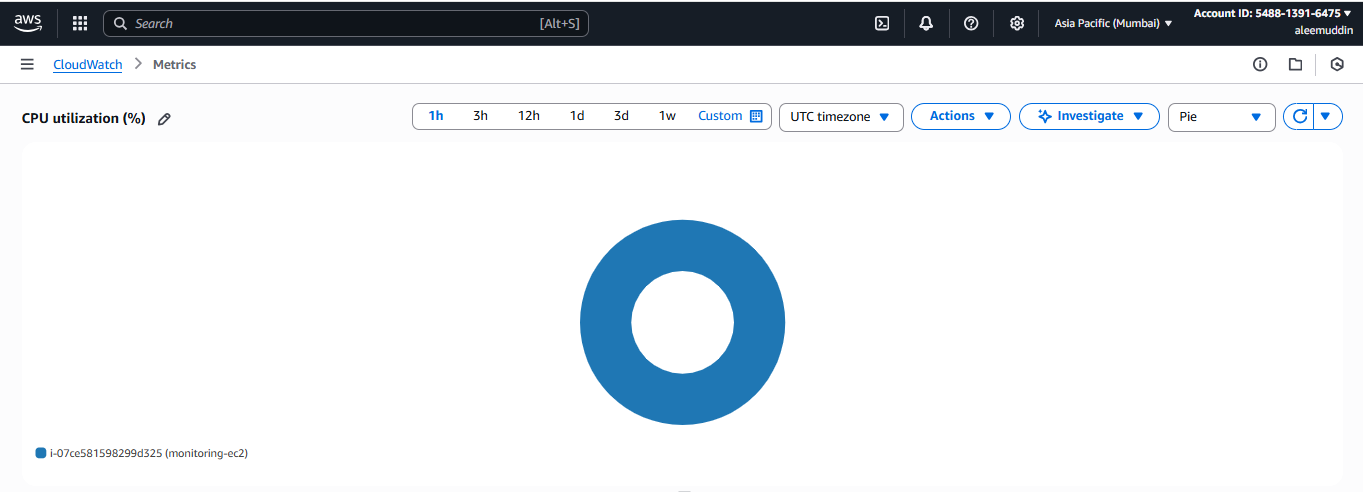




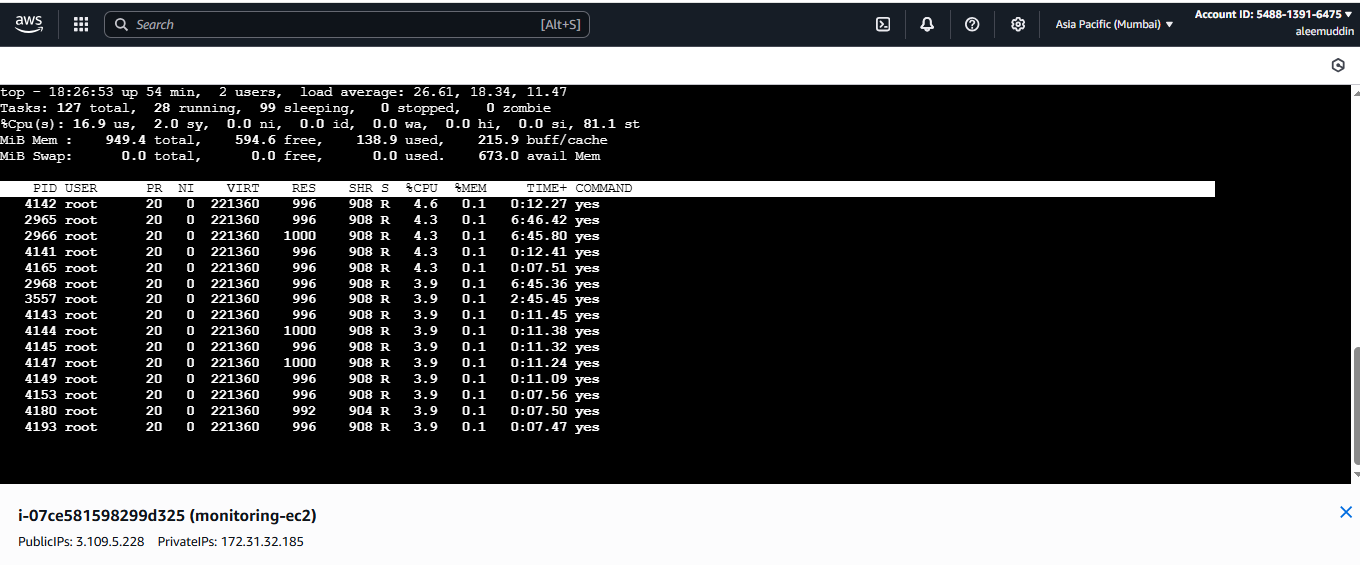




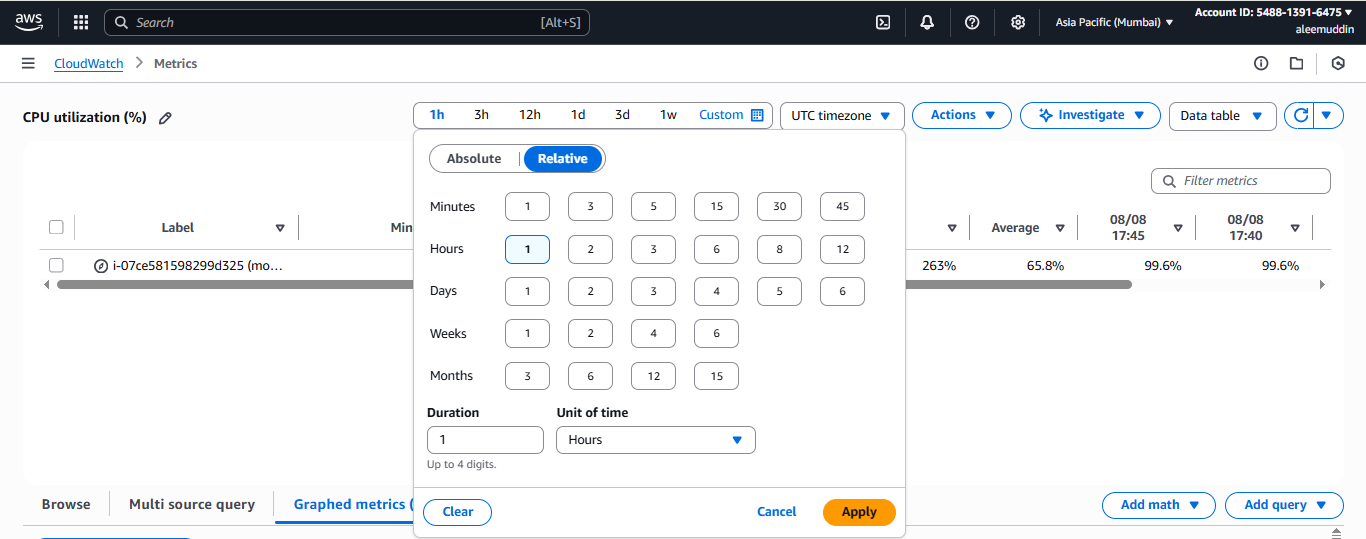




**Top command to check CPU utilization.**

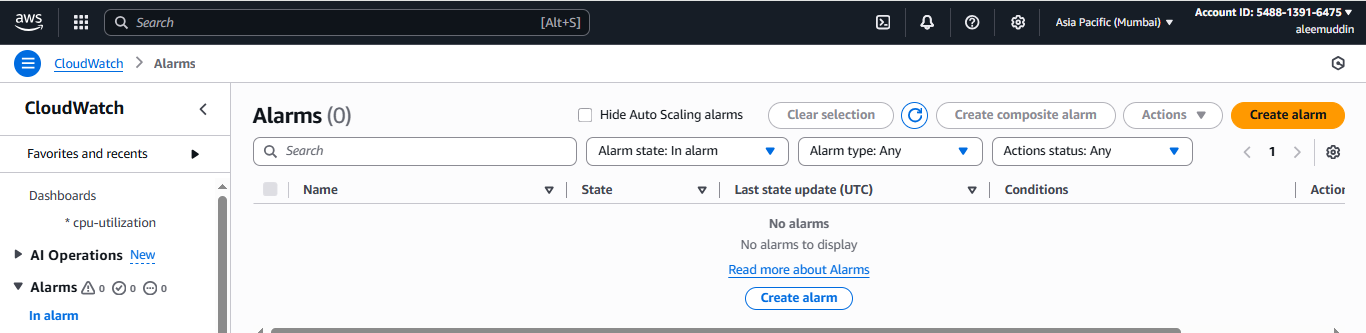


**Setting the graph time to monitor in every minutes or hours.**

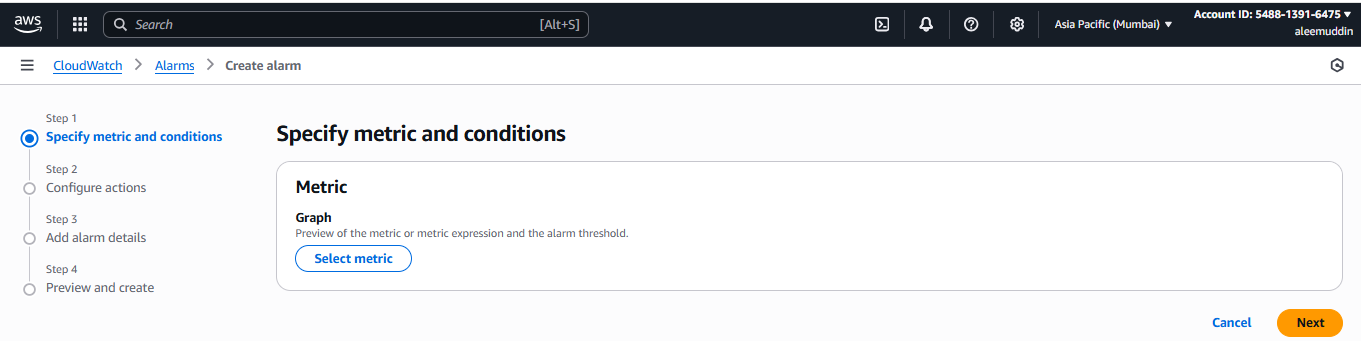


1. **Create one alarm to send alert to email if the cpu utilization is more than 70 percent.**

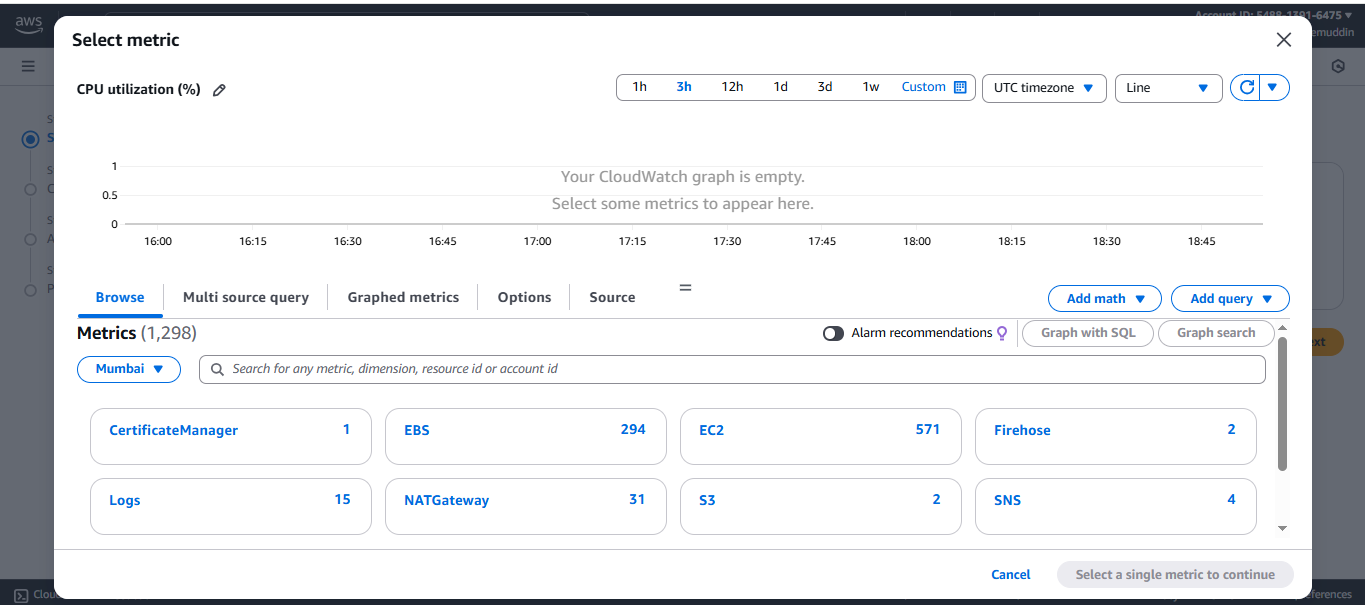
**Go to cloudwatch>Create alarm>**



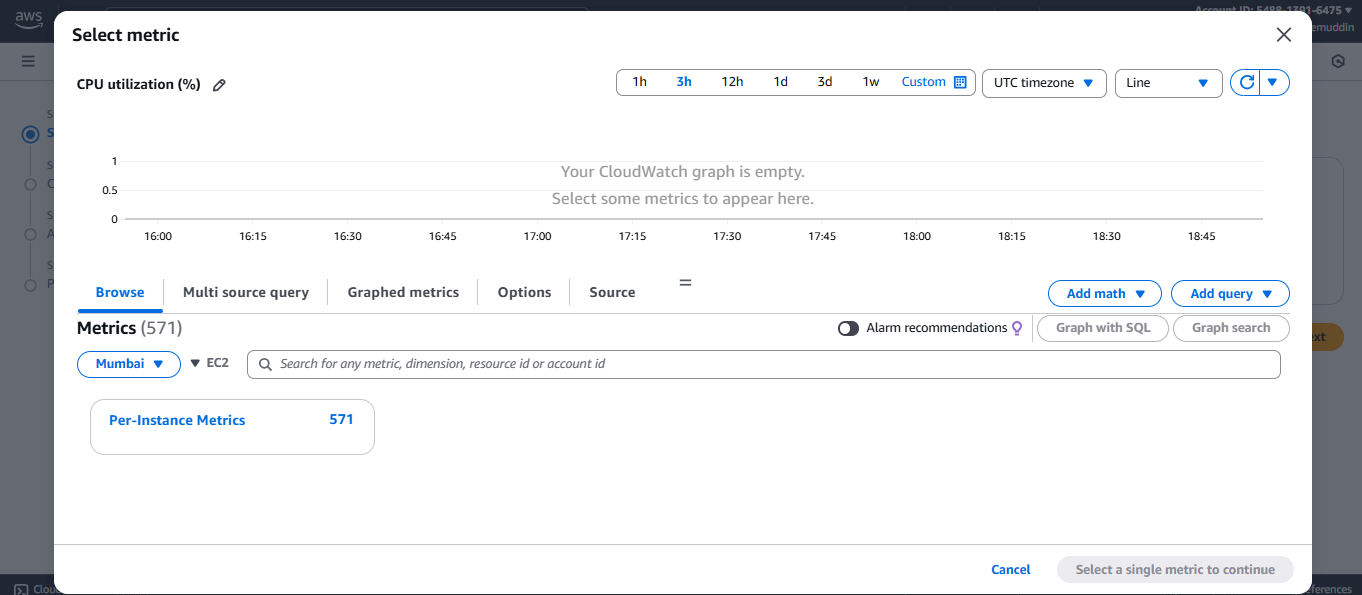
**Select metric**



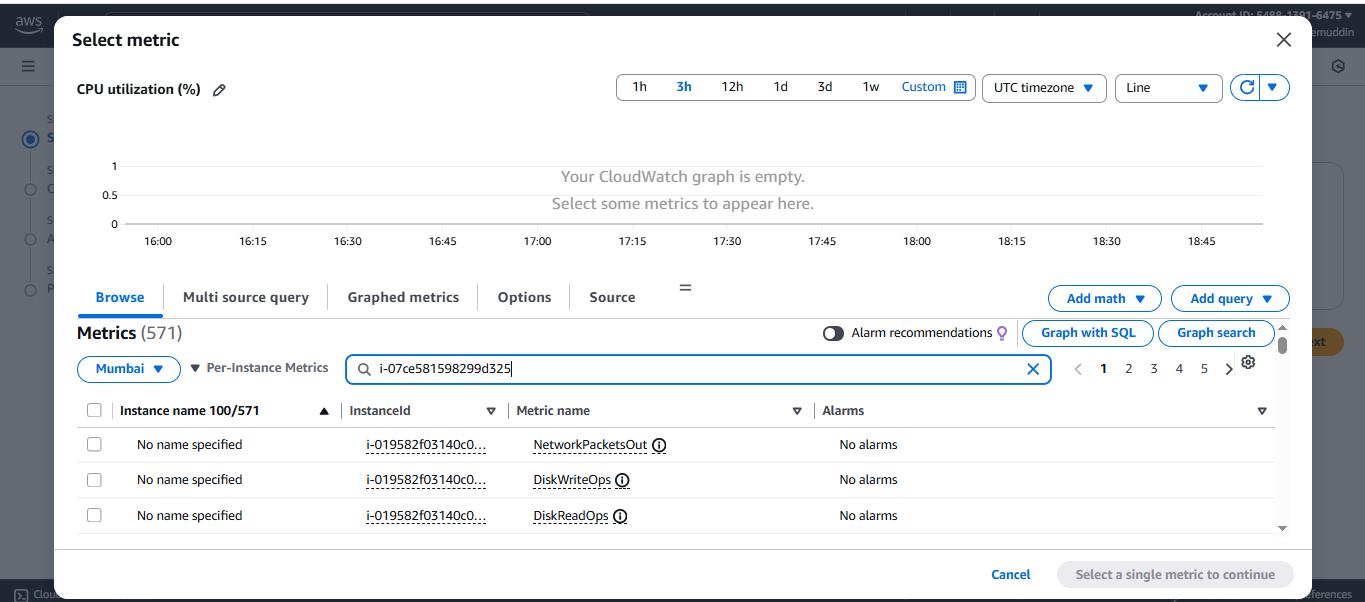
**Click on ec2 EC2**

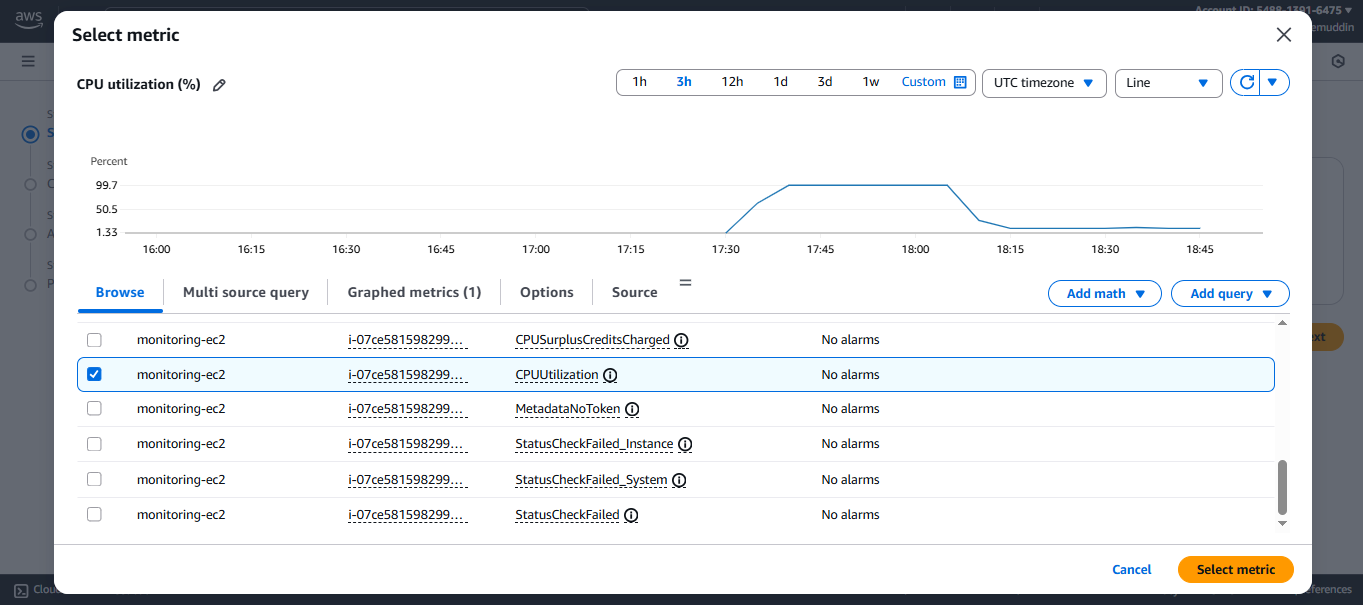


**Click on pre-instance metrics**

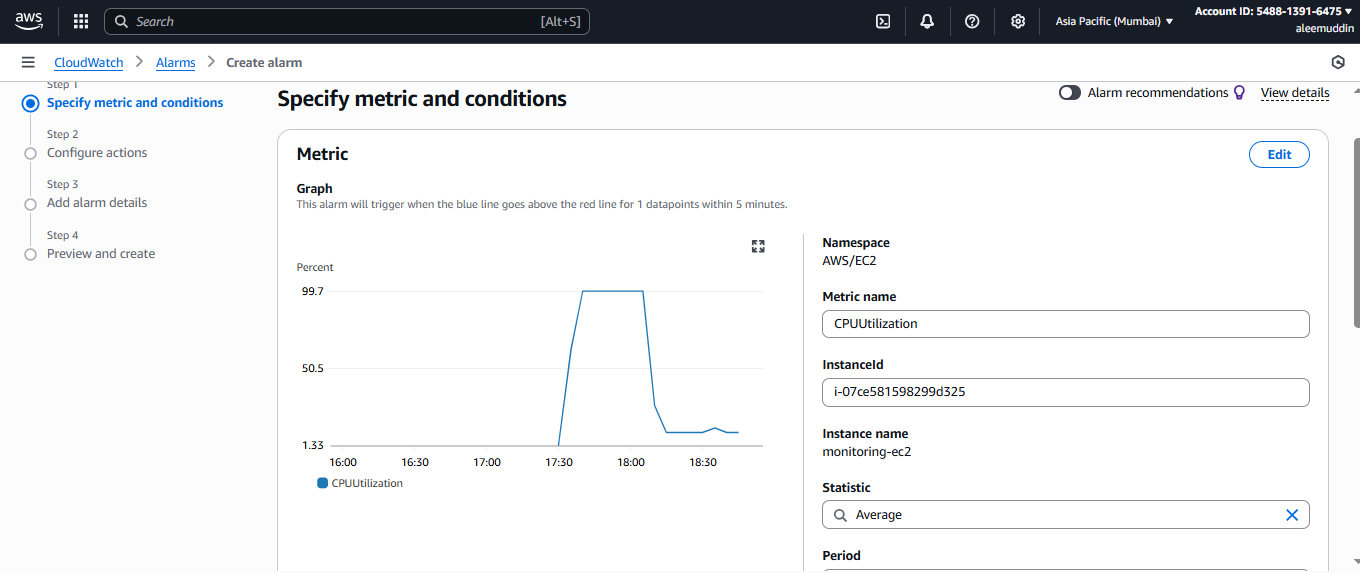


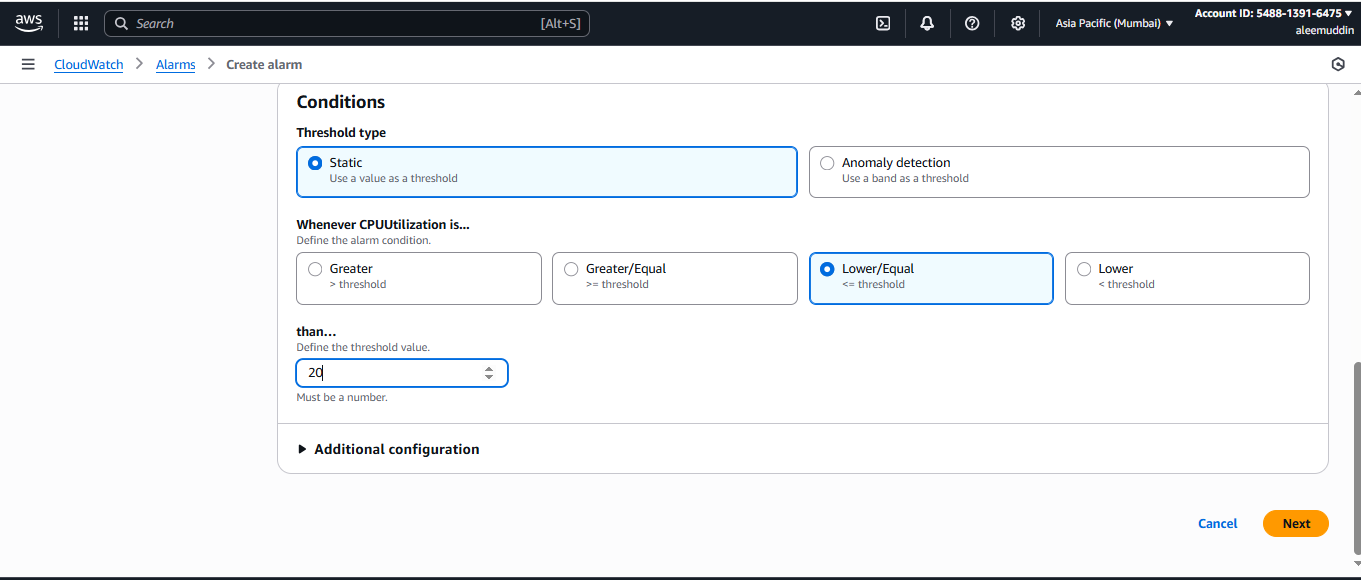
**Go to search bar and paste ec2 instance id**

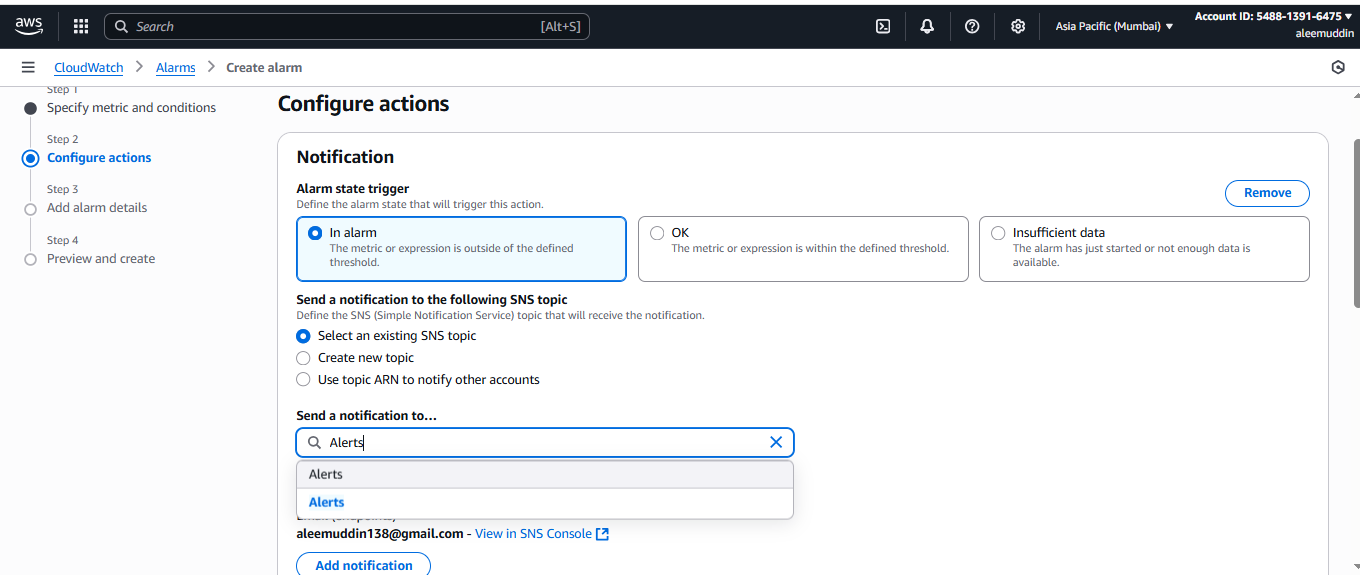
**Select CPU utilization>Select metric & continue**

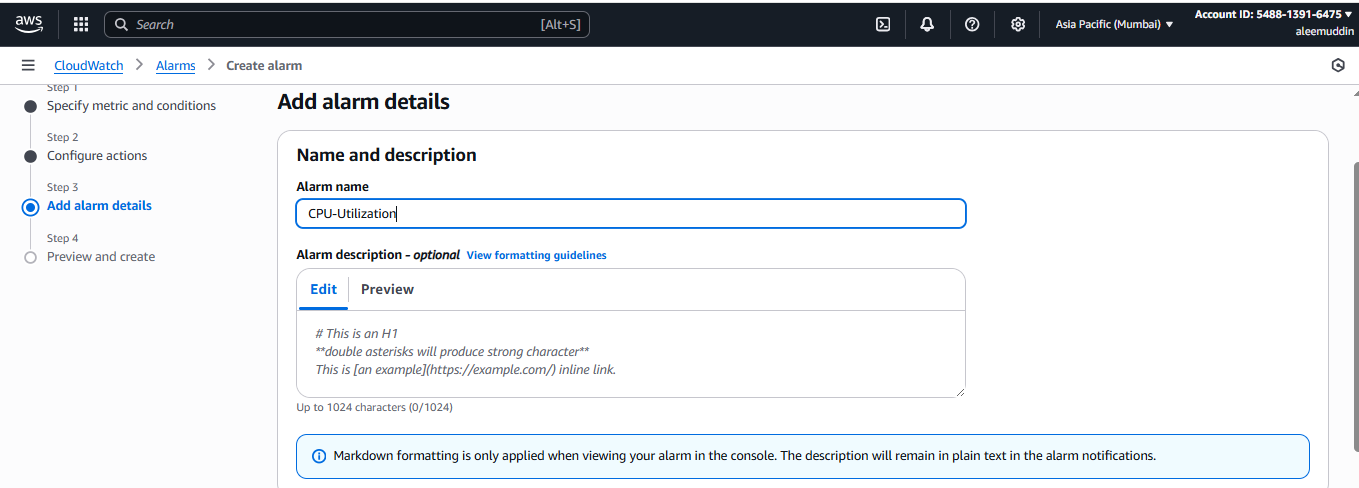


**To create alarms greater then or less then CPU ulilization 70% follow the below steps**

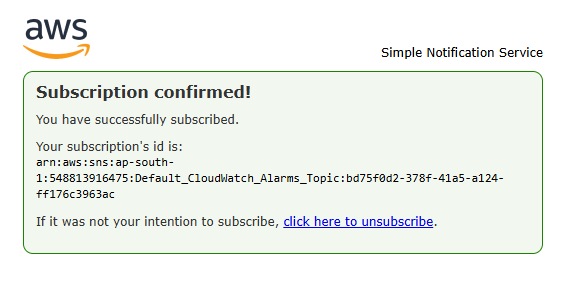




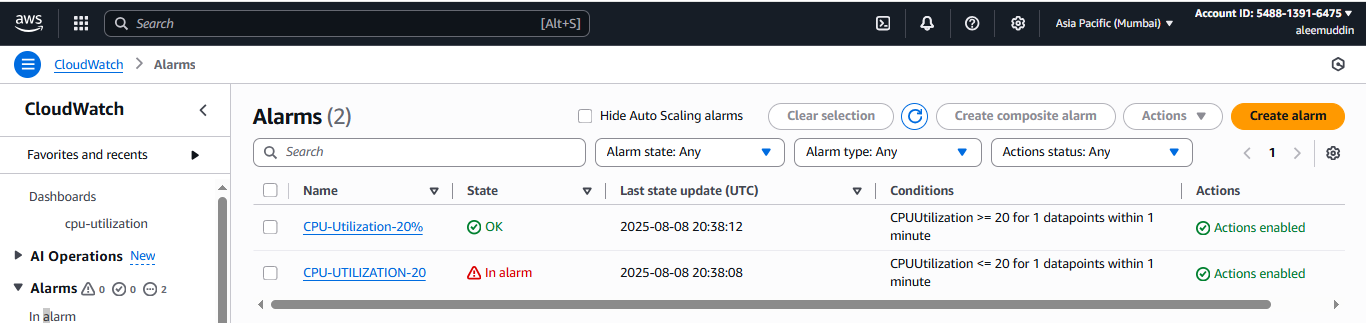




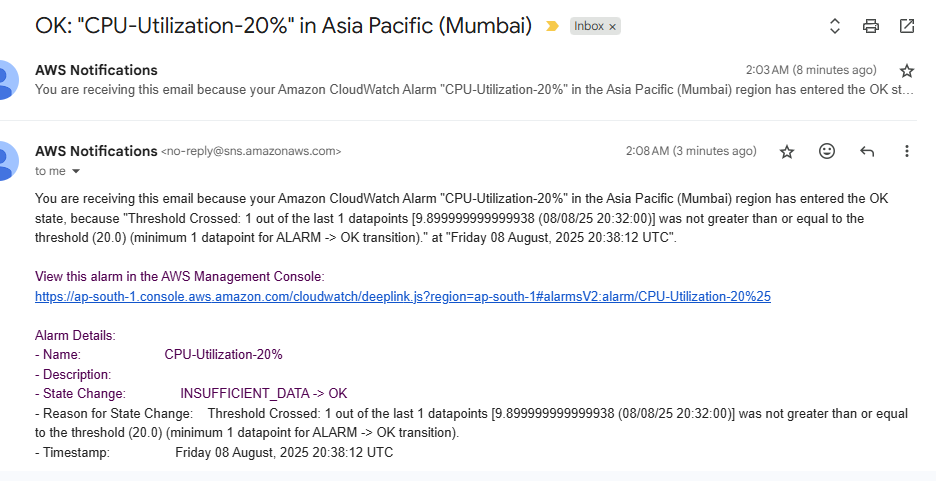
**After this step go to mail confirm the subscription**

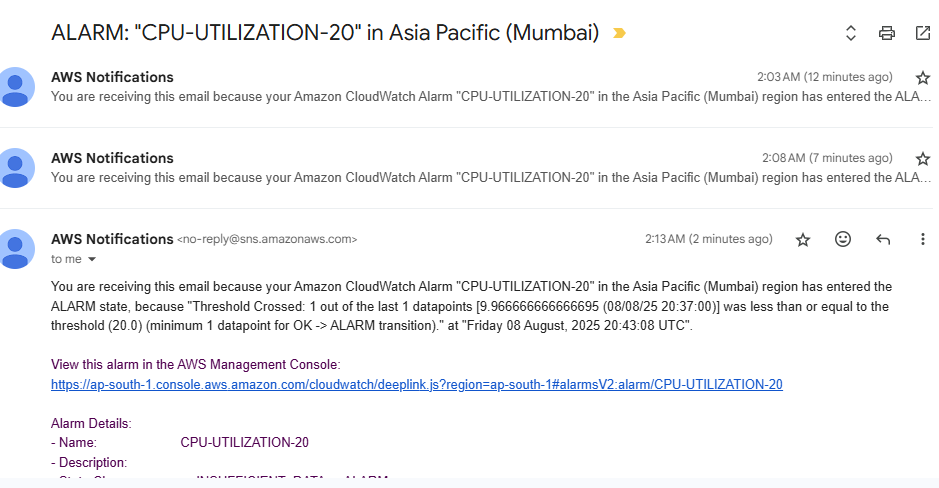


**After confirmation will get the alarm status**



**If the utilization is greater or less we get the notification in th email**





1. **Create Dashboard and monitor tomcat service wether it is running or not and send the alert.**
2. **In ec2 installed java and then apache 9.0.107 and create the script → chmod 777 → and run**
3. **Sudo wget** [**https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.107/bin/apache-tomcat-9.0.107.tar.gz**](https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.107/bin/apache-tomcat-9.0.107.tar.gz)
4. **sudo tar -xzf** [**apache-tomcat-9.0.107.tar.gz**](http://apache-tomcat-9.0.107.tar.gz)
5. **sudo mv apache-tomcat-9.0.107 /opt/tomcat9**
6. **cd /opt/tomcat9/bin**
7. **sudo chmod +x \*.sh**
8. **[root@ip-172-31-44-227 ~]# vi monitoring.bash**
9. **[root@ip-172-31-44-227 ~]# vi monitoring.bash**
10. **[root@ip-172-31-44-227 ~]# chmod 777 monitoring.bash**

**### script taken**

**#!/bin/bash**

**# Get instance ID from metadata (v2)**

**TOKEN=$(curl -sX PUT "http://169.254.169.254/latest/api/token" \**

**-H "X-aws-ec2-metadata-token-ttl-seconds: 21600")**

**INSTANCE\_ID=$(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" \**

**http://169.254.169.254/latest/meta-data/instance-id)**

**# Check if Tomcat is running**

**checkTomcatStatus() {**

**pgrep -f tomcat > /dev/null**

**if [ $? -eq 0 ]; then**

**echo 1**

**else**

**echo 0**

**fi**

**}**

**i=$(checkTomcatStatus)**

**# Push to CloudWatch**

**aws --region ap-south-1 cloudwatch put-metric-data \**

**--metric-name tomcat \**

**--value "$i" \**

**--namespace tomcat \**

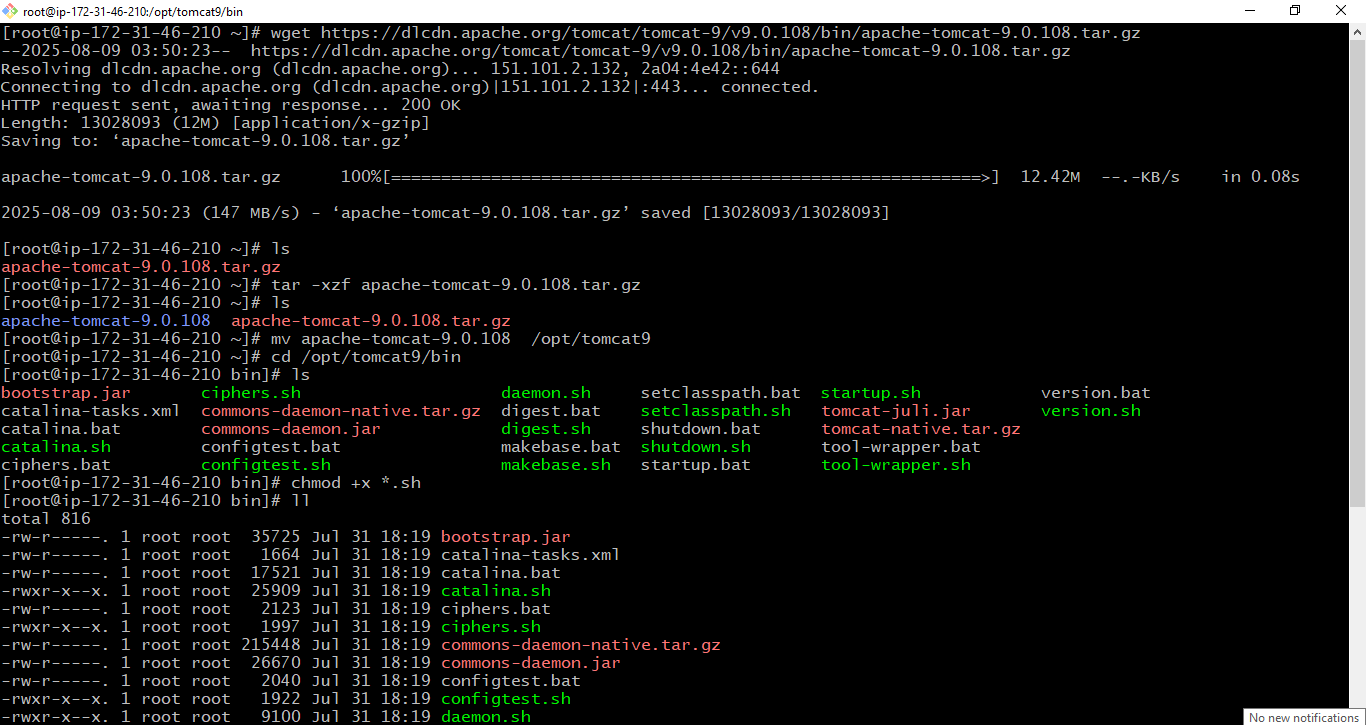
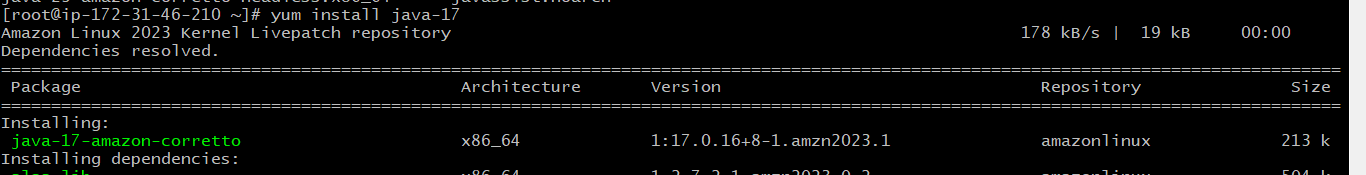
**--dimensions InstanceId="$INSTANCE\_ID"**

**Now create script in crontab**

1. **Crontab -e**

**\* \* \* \* \* /root/monitoring.bash**

1. **Once you run the monitoring.bash it will reflect in CloudWatch Dashboard.**
2. **Now create Dashboard for Tomcat you can see the tomcat select metric in number and then if metric >=1 tomcat running <=0 tomcat nor running then select SNS Topic(if not there then create) → give alarm name and done.**
3. **Now you can see in dashboard if we start the service then 1 showing , if stop service 0 showing and accordingly ALARM mail will be sent.**



**sudo chmod +x \*.sh**

**using the above command will all files of .sh become executable.**

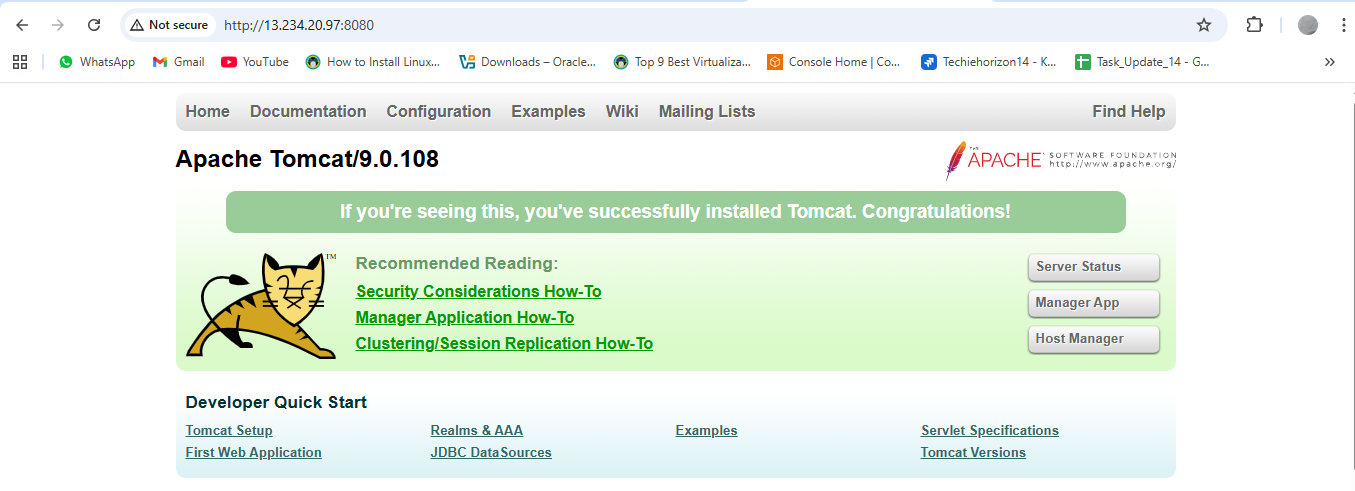
**chmod +x \*.sh → only adds execute permission,  
keeping the existing read/write rules the same.**

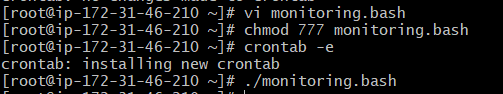
**If the file was rw-r--r--, after +x it becomes rwxr-xr-x (owner can write, others cannot).**

**Why cant we use chmod 777**

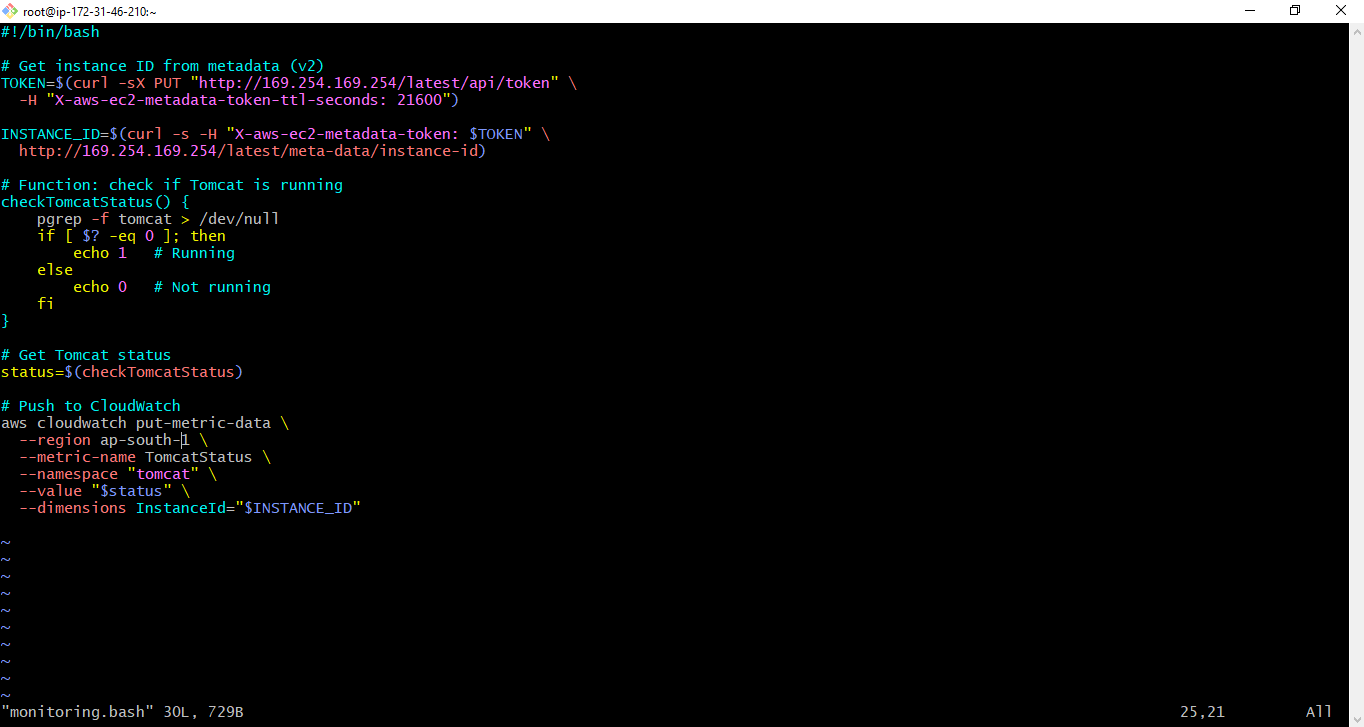
* **chmod 777 → read + write + execute for everyone  
  That means any user** **(even non-admin accounts) could modify or run Tomcat scripts.**
* **This is risky — someone could edit startup.sh or shutdown.sh to run malicious code.**

**Go to browser by using ec2 instance public ip check by adding default port number**





[root@ip-172-31-30-216 ~]# vi monitoring.bash



#!/bin/bash

# Get instance ID from metadata (v2)

TOKEN=$(curl -sX PUT "http://169.254.169.254/latest/api/token" \

-H "X-aws-ec2-metadata-token-ttl-seconds: 21600")

INSTANCE\_ID=$(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" \

http://169.254.169.254/latest/meta-data/instance-id)

# Function: check if Tomcat is running

checkTomcatStatus() {

pgrep -f tomcat > /dev/null

if [ $? -eq 0 ]; then

echo 1 # Running

else

echo 0 # Not running

fi

}

# Get Tomcat status

status=$(checkTomcatStatus)

# Push to CloudWatch

aws cloudwatch put-metric-data \

--region ap-south-1 \

--metric-name TomcatStatus \

--namespace "tomcat" \

--value "$status" \

--dimensions InstanceId="$INSTANCE\_ID"

**[root@ip-172-31-30-216 ~]# chmod 777 monitoring.bash**

**[root@ip-172-31-30-216 ~]# sudo crontab -e**

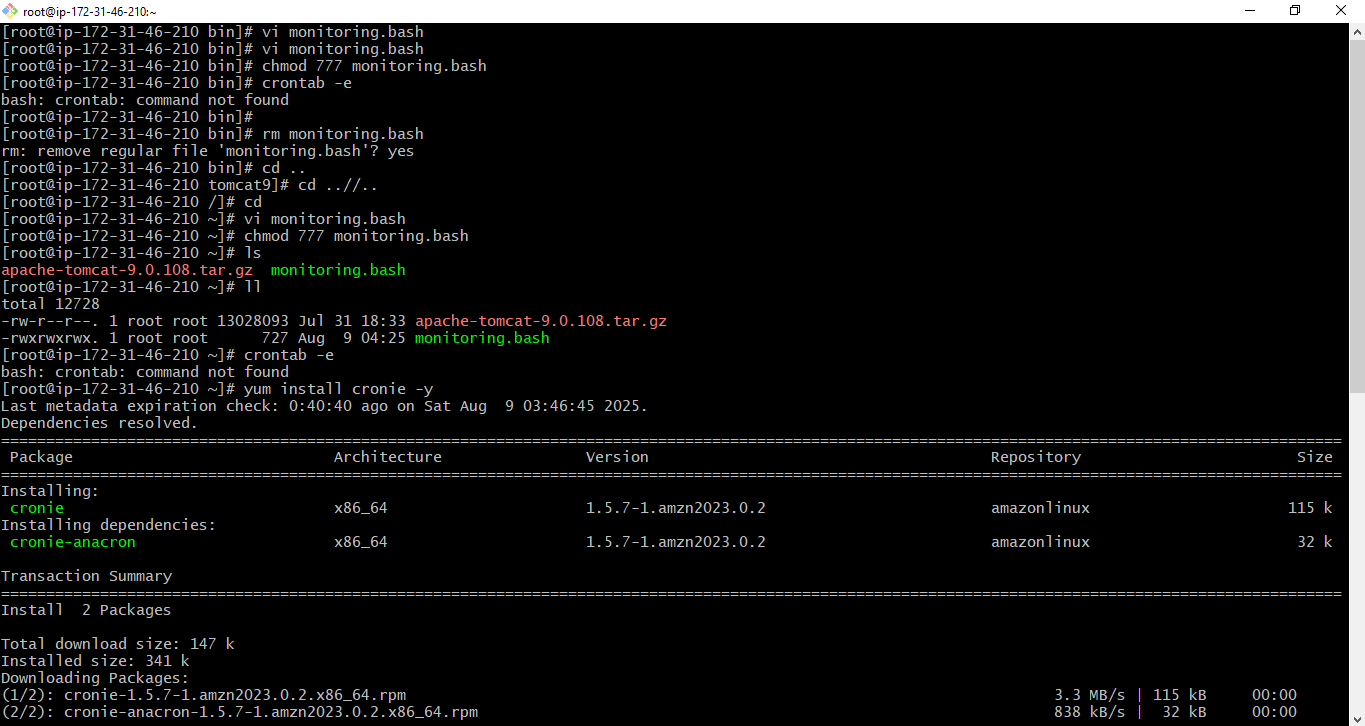
**sudo: crontab: command not found**

**I this error came it’s mean you don’t have crontab configured in your ec2**

**Or this we need to configure**

**Since you’re on Amazon Linux ,run:**

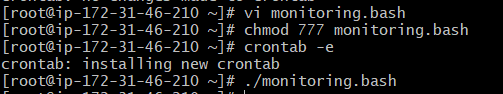
**[root@ip-172-31-30-216 ~]# yum install cronie –y**

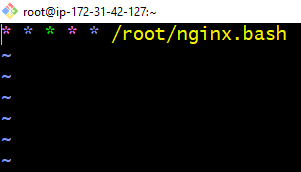


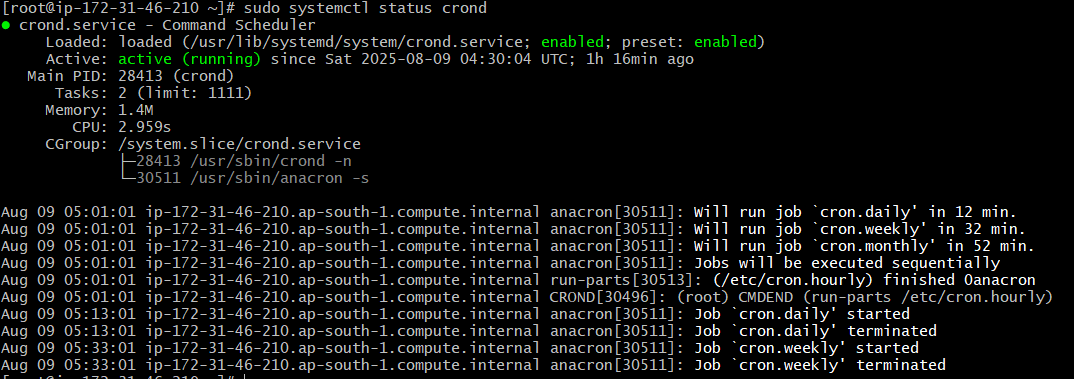
**Then start and enable the cron service:**

**systemctl start crond**

**systemctl enable crond**

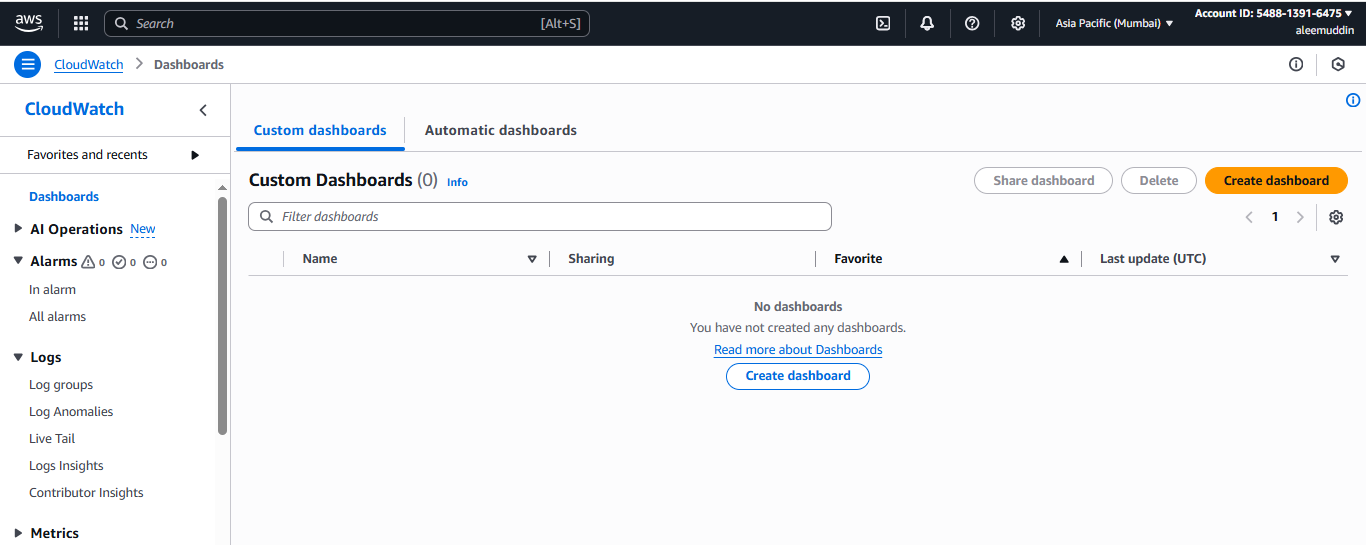


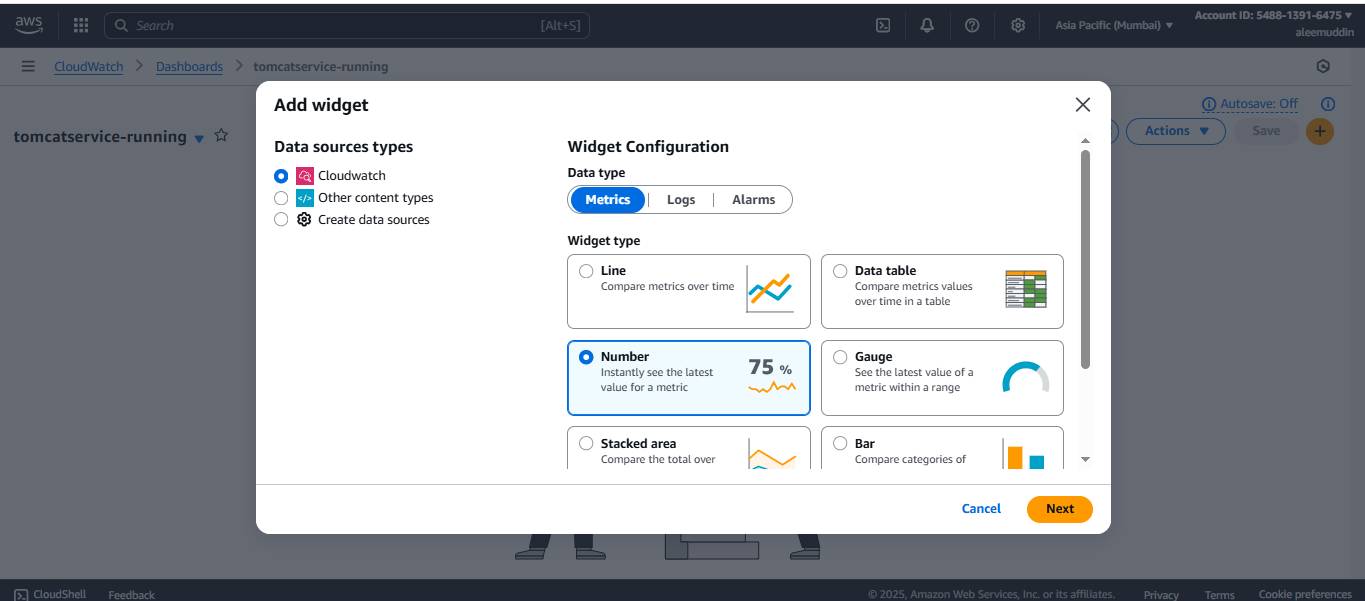


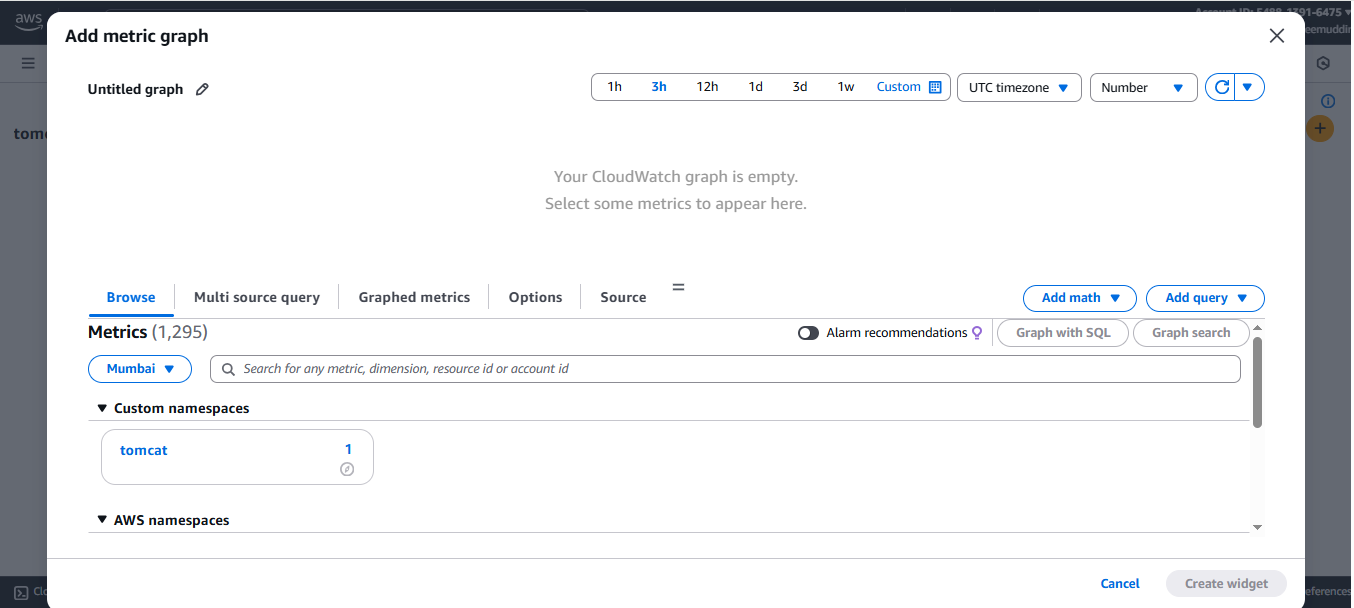


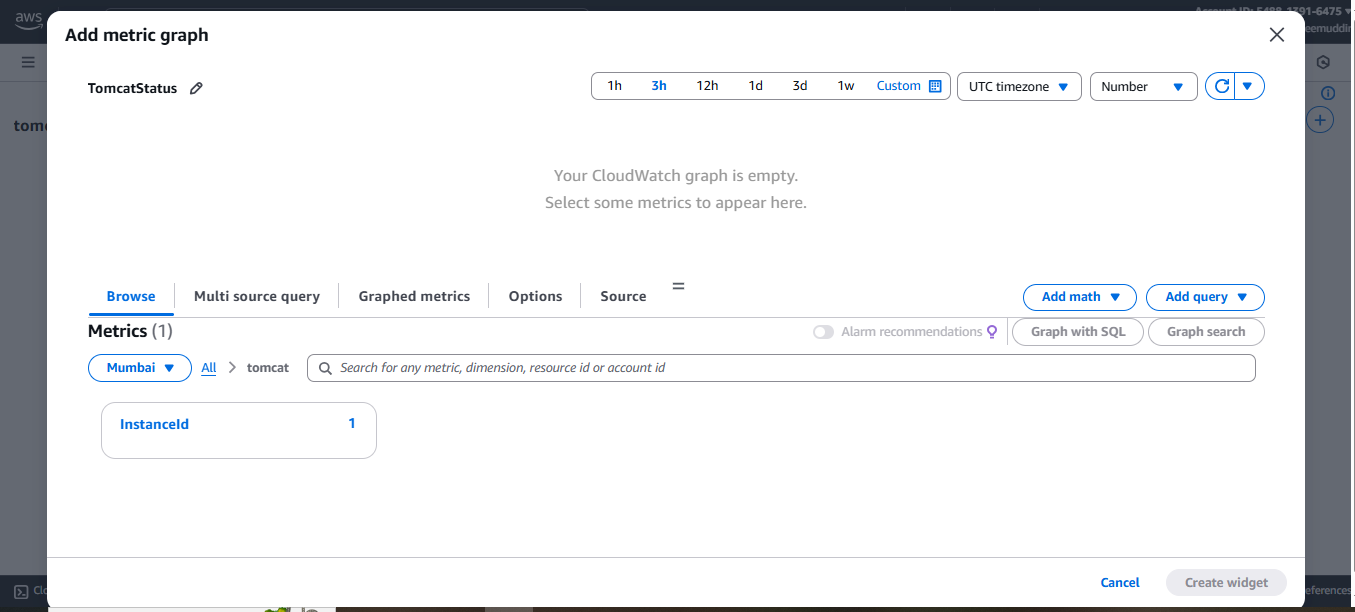
1. **Once you run ./monitoring.bash it will reflect in CloudWatch Dashboard.**
2. **Now create Dashboard for Tomcat you can see the tomcat select metric in number and then if metric >=1 tomcat running <=0 tomcat nor running then select SNS Topic(if not there then create) → give alarm name and done.**

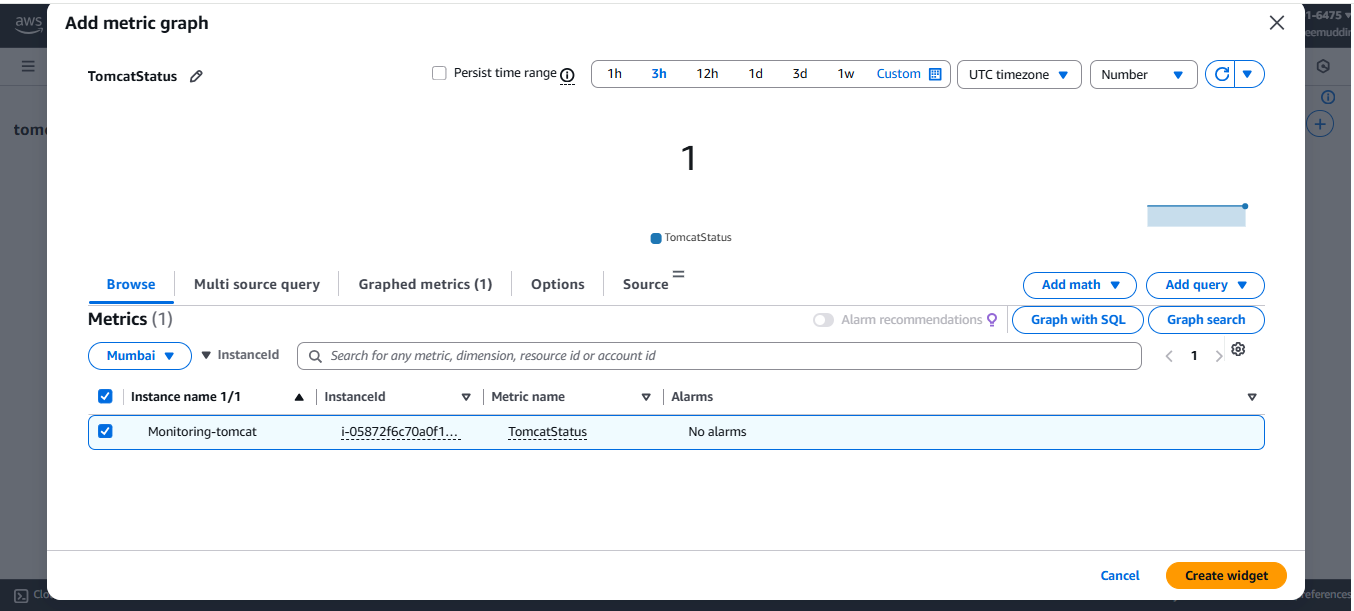
**Now you can see in dashboard if we start the service then 1 showing , if stop service 0 showing and accordingly ALARM mail will be sent.**

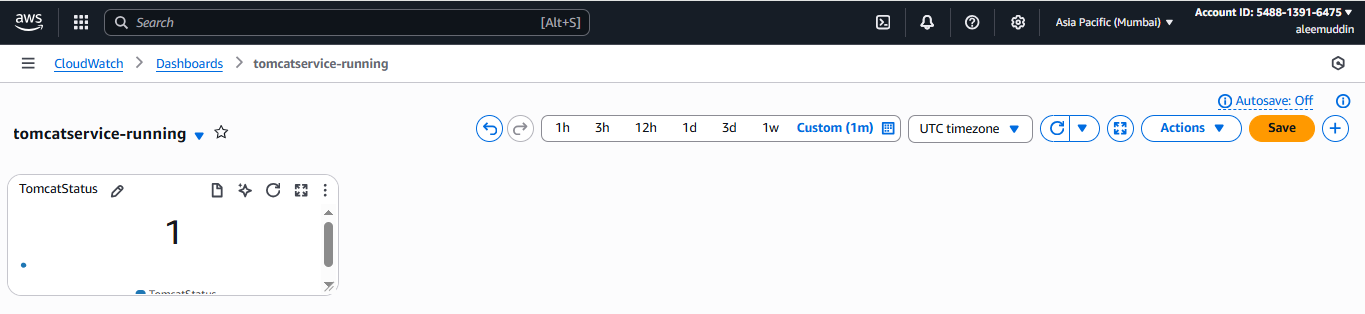


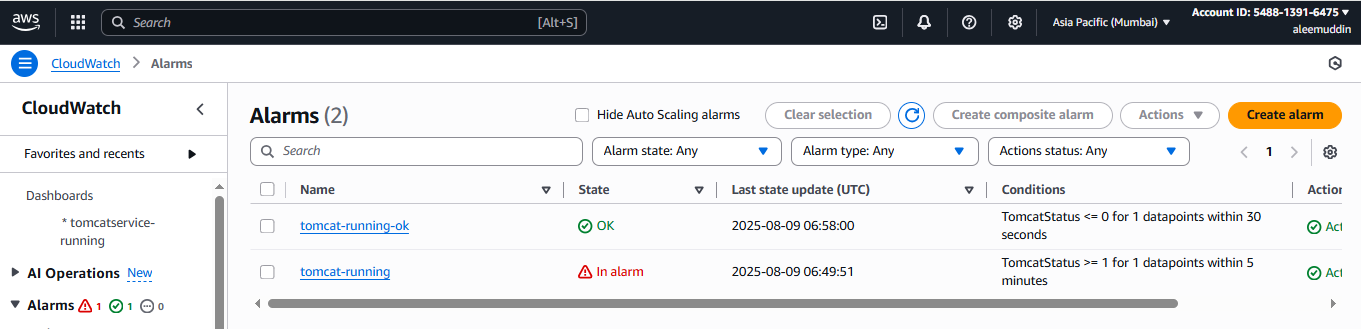






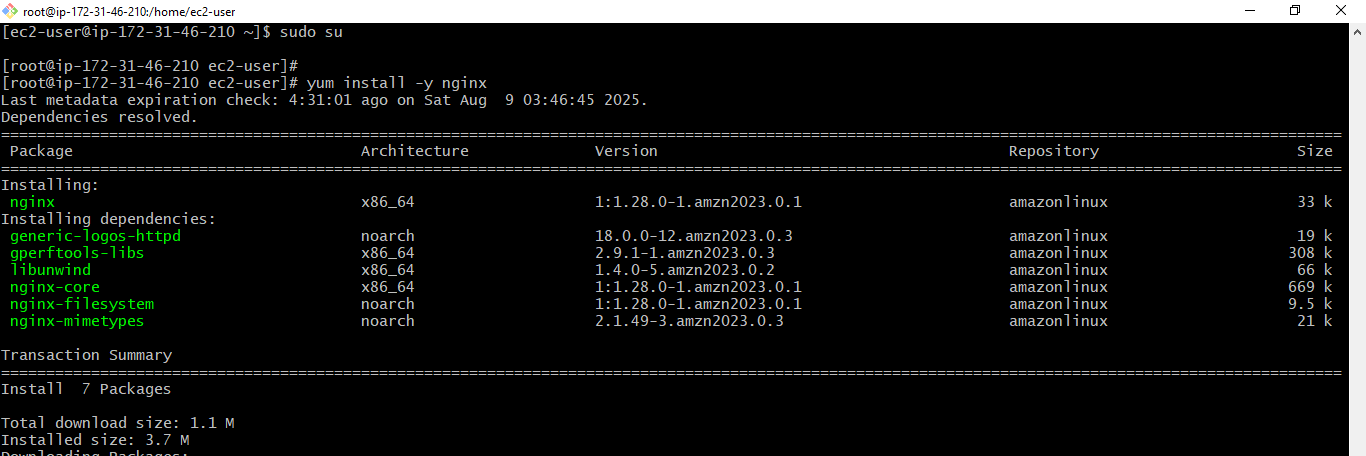




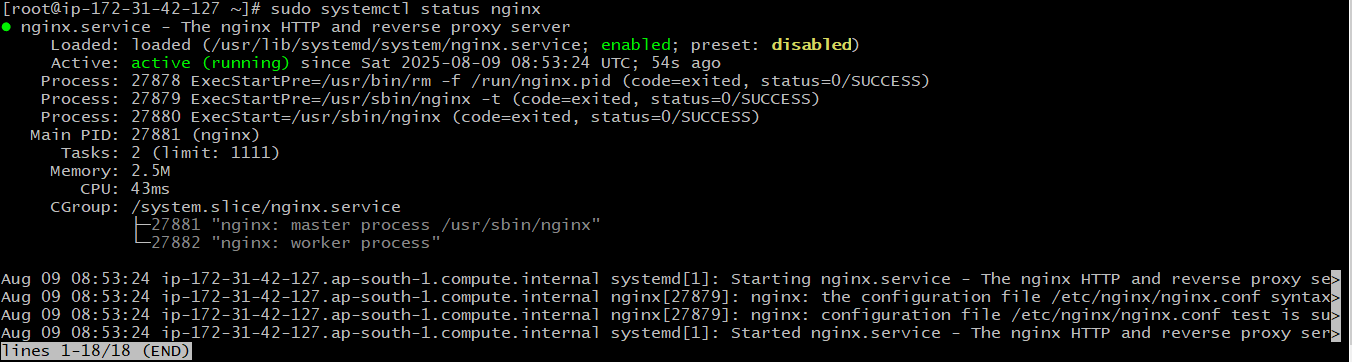


1. **Create Dashboard and monitor nginx service to send the alert if nginx is not running.**

**Configure the nginx**



**After configuration check the status of the service**



**If crontab command then install cronie**

