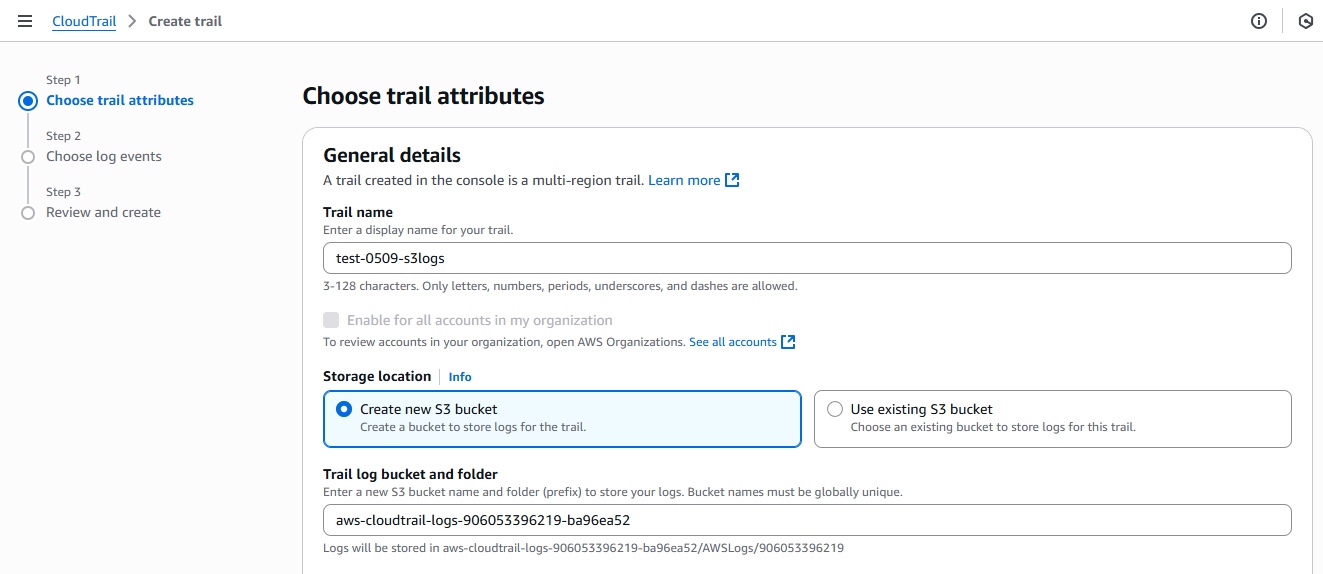
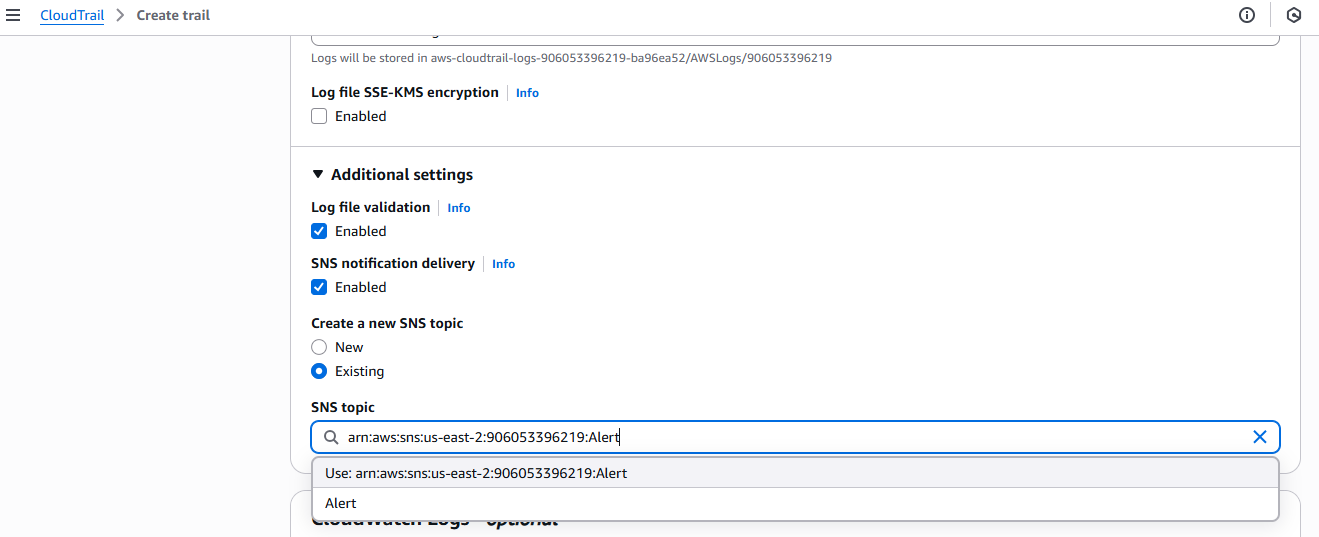
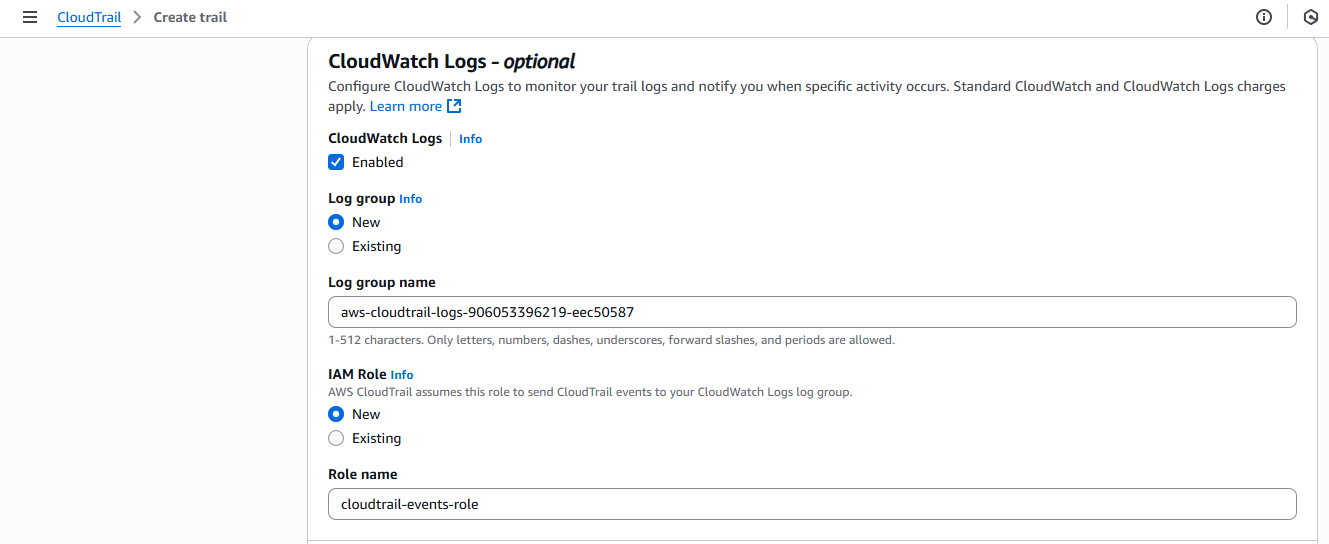
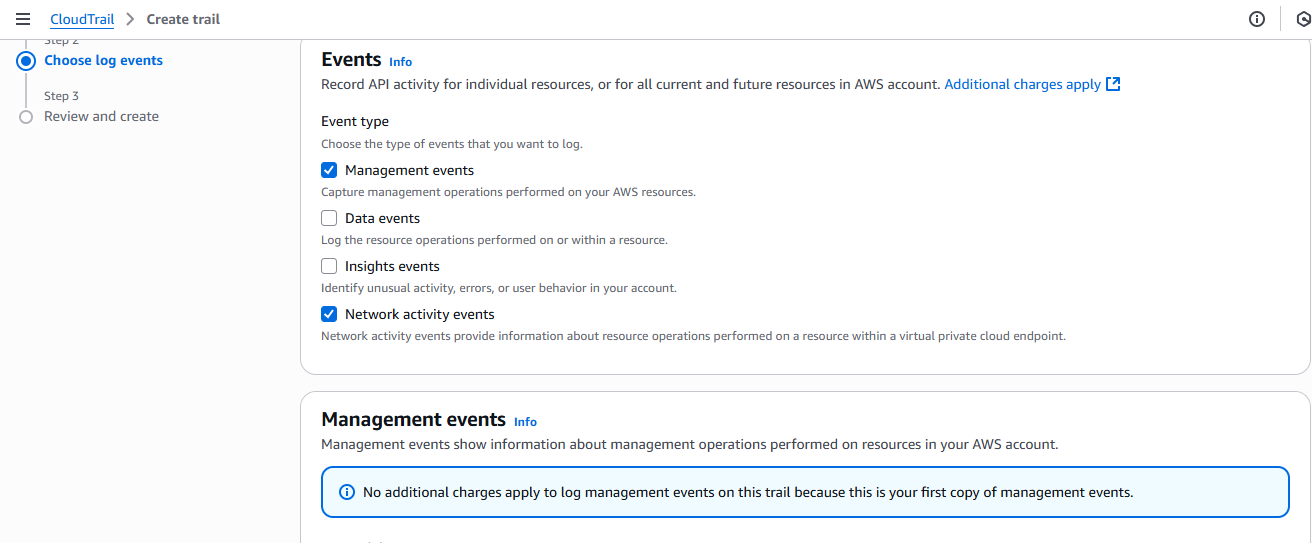
1. Enable cloudtrail monitoring and store the events in s3 and cloudwatch log events.



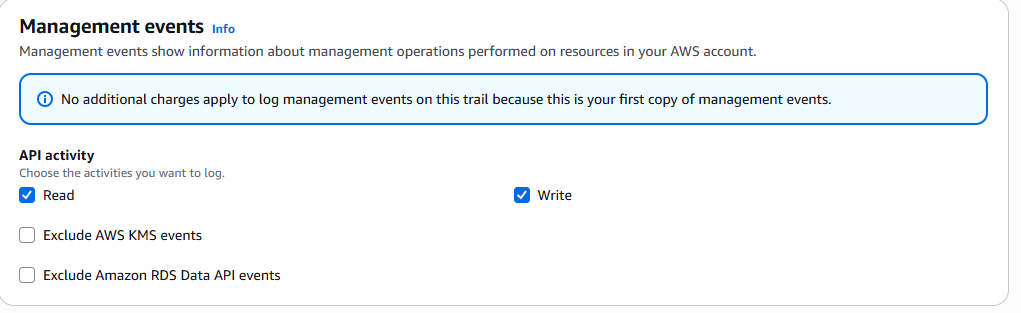




Click on Next

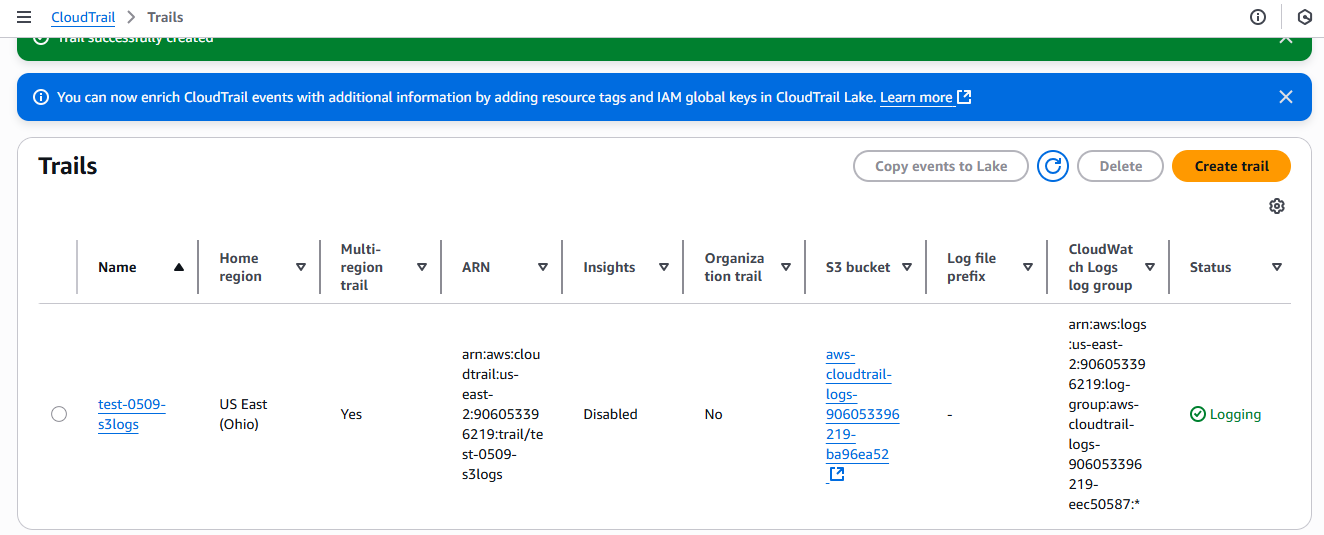


Remove network activity events for the time being iam taking only one event.



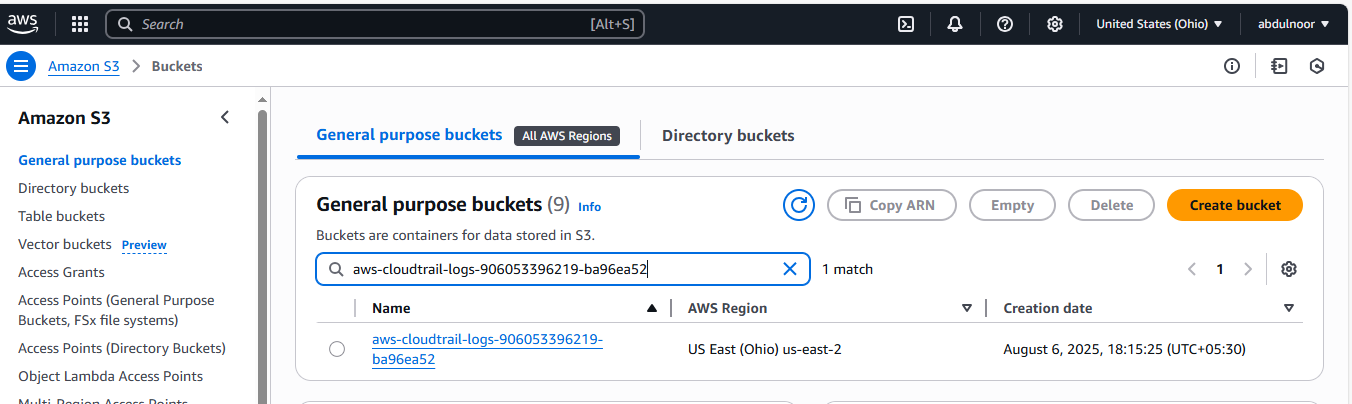
Click on next

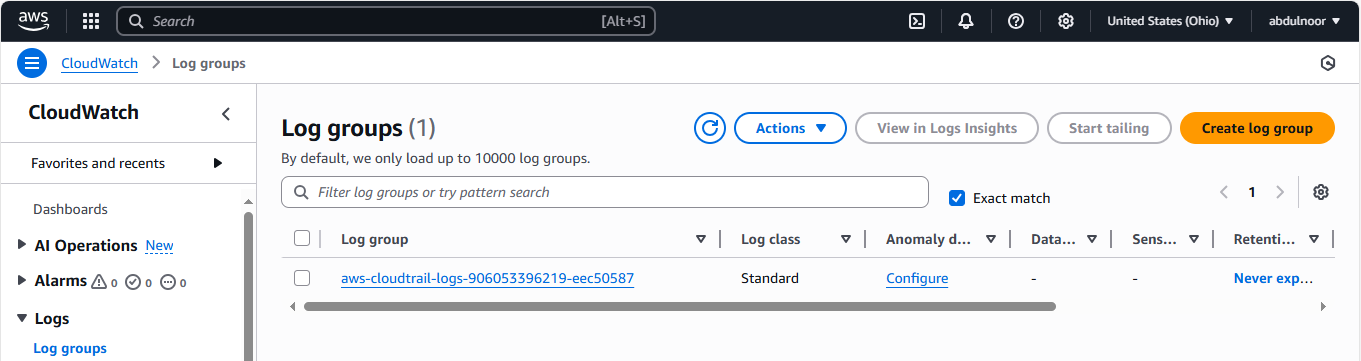
Click on cloud trail



Logs in S3 as in cloud trail I took new and it’s created bydefault and same for cloud watch.

Or else we can create one S3 bucket and one cloud watch an we can assign in it in cloud trail as under existing check box .



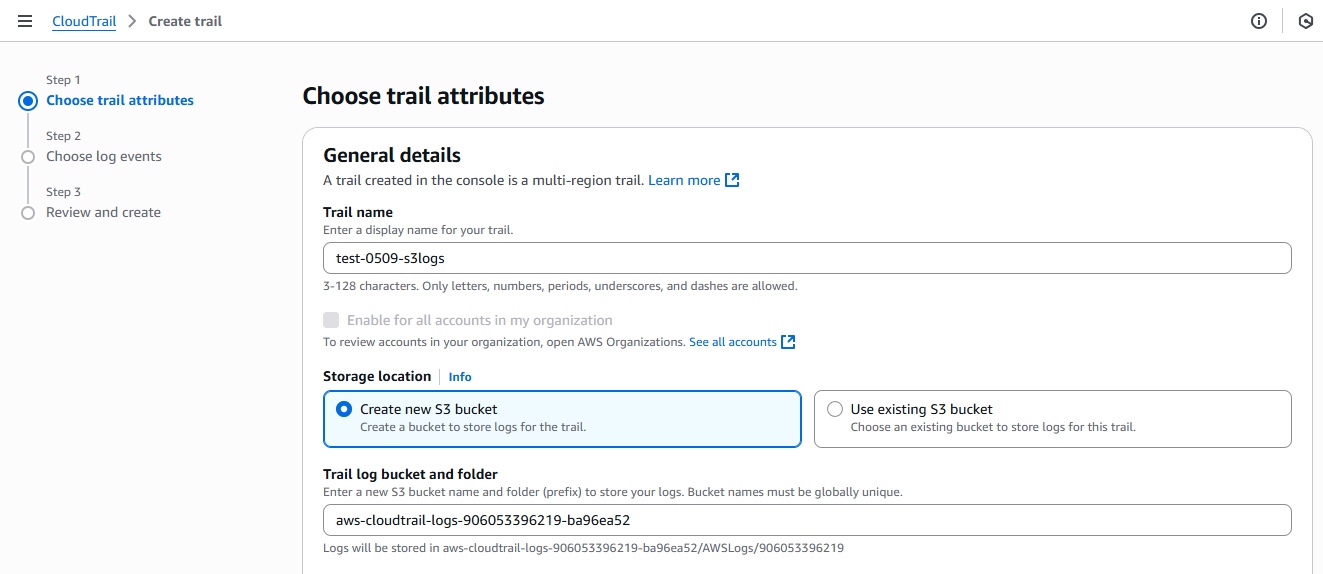


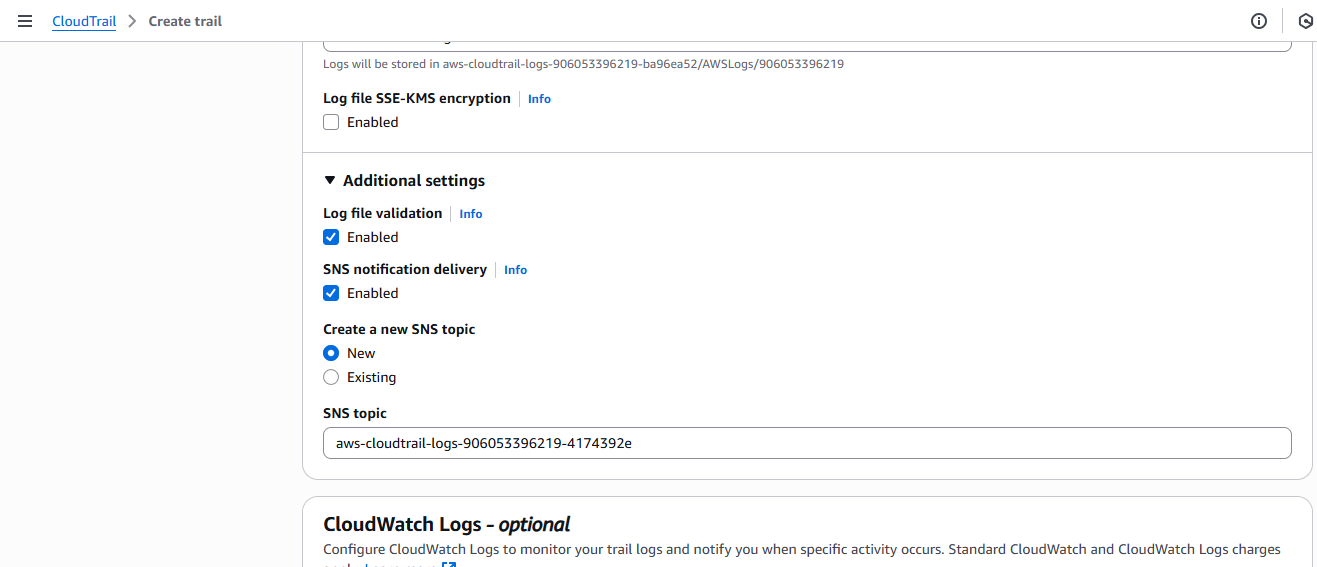
2) Enable SNS for cloudtrial to send alert on email.

**Create full worklow of Cloud Trail:**

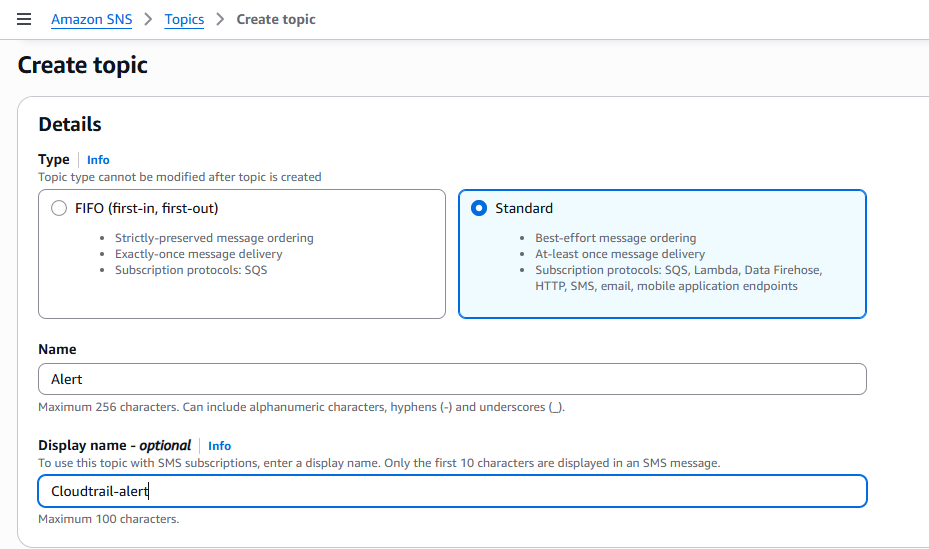
Go to – AWS console – Search – cloudtrail – create a trail – click on create trail (blue colour written)

Now see the diferent version o GUI of cloud trail.





But here, letus try to create new SNS . For this goto – aws console – search – SNS – Simple Notification Service (SMS managed message topics for pub/sub) – Next step

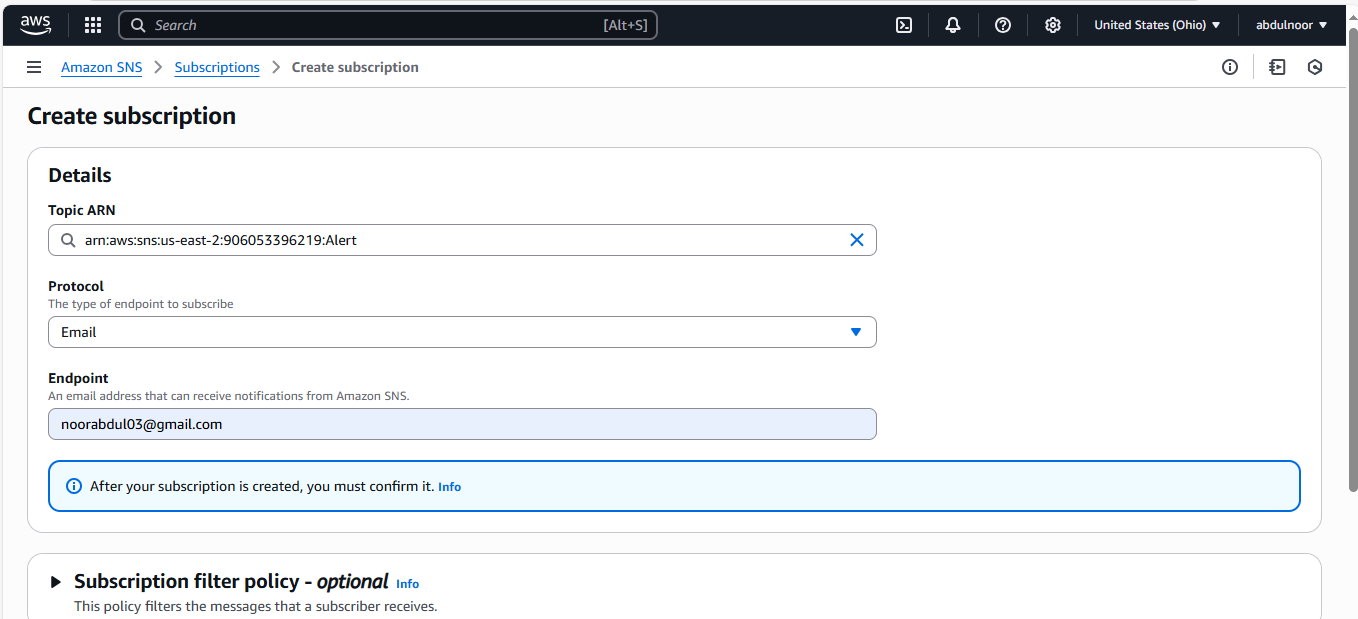


Lets create basic one only without any other settings

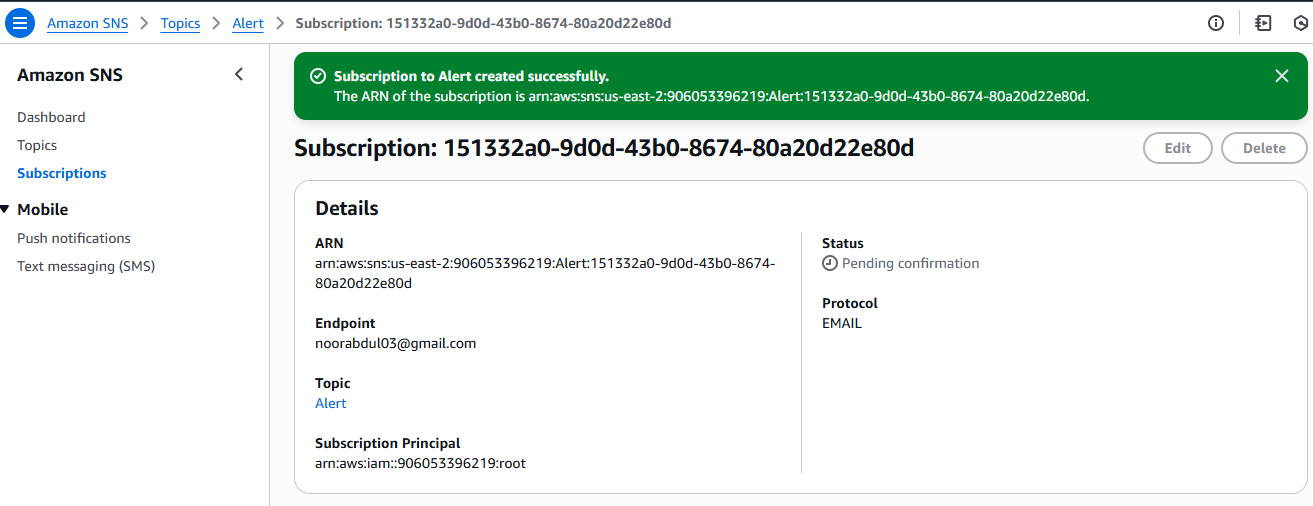
Click on create topic

One SNS topic created and with this now to this particular topic what kind of alert you want to configure

Click on subscription – create subscriptions

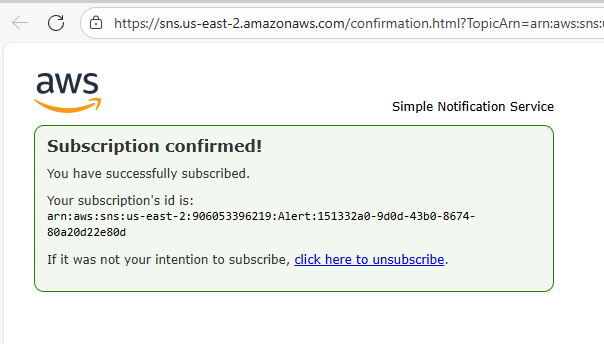


Click on create subscriptions

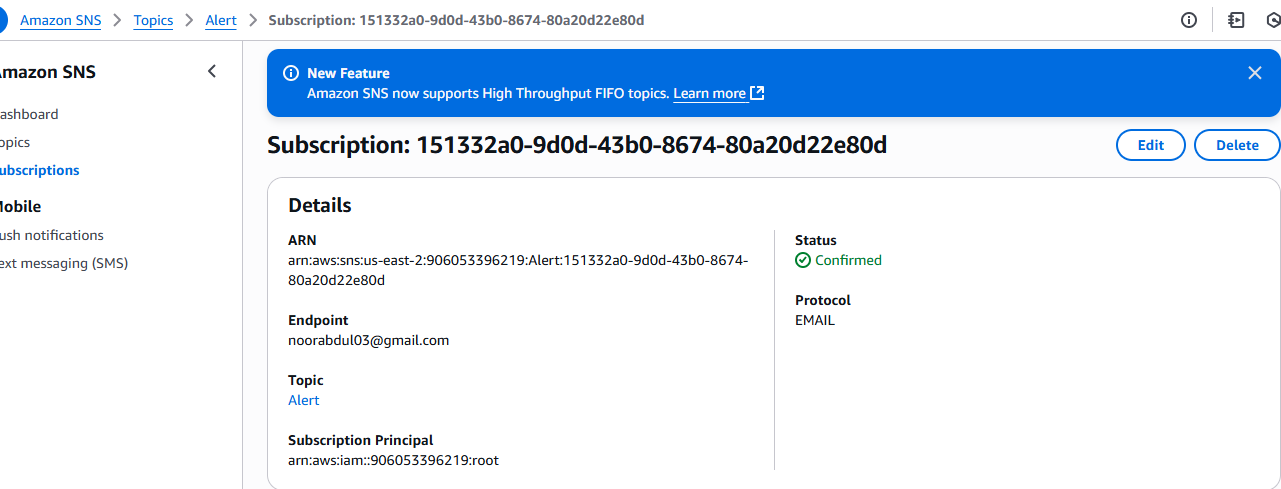


Showing status as pending. Means aws SNS has sent one notification to our email where we need to confirm that this SNS service has enabled.

Go to email and click on confirmation



Reresh the page of SNS and see status as confirmed.



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

To add fake CPU utilization just use the below command because as we don’t have user or any who is using cpu to reach 100%. That’s why adding fake.

Command

yes >/dev/null &

enter the above command ater every one minute an refresh the cw

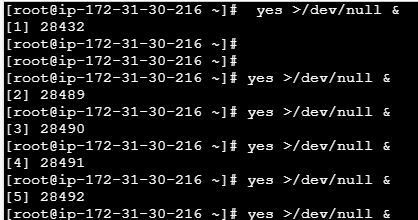
|  |  |
| --- | --- |
| yes | A command that **prints y continuously** to the terminal (nonstop, very fast) |

|  |  |
| --- | --- |
| > | Redirects the output |

|  |  |
| --- | --- |
| /dev/null | A special file that **discards anything written to it** (a “black hole”) |

|  |  |
| --- | --- |
| & | Runs the command **in the background**, so your terminal stays usable |

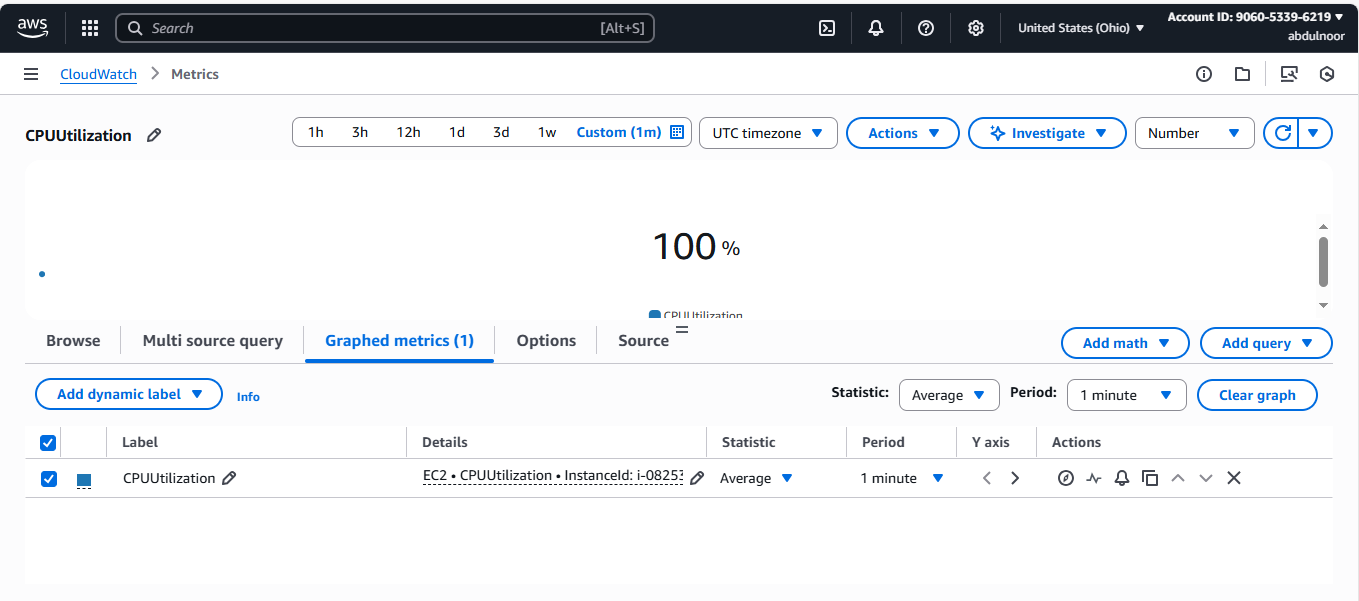
But before running this command make sure that any yes process not running in the back ground. I it is then kill or stop

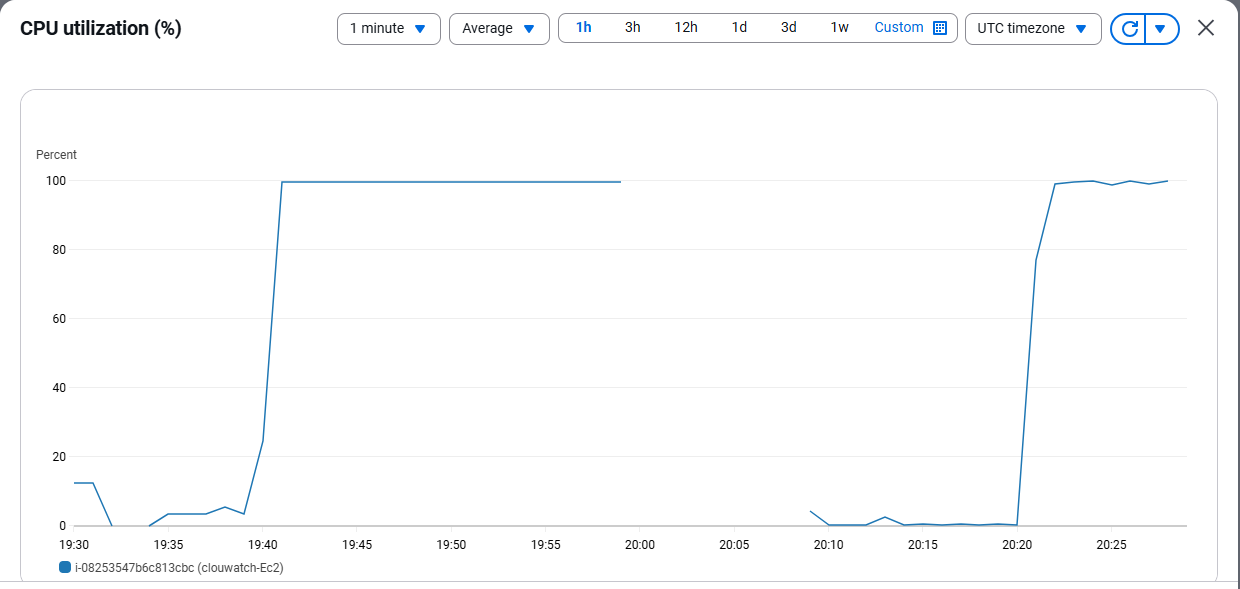


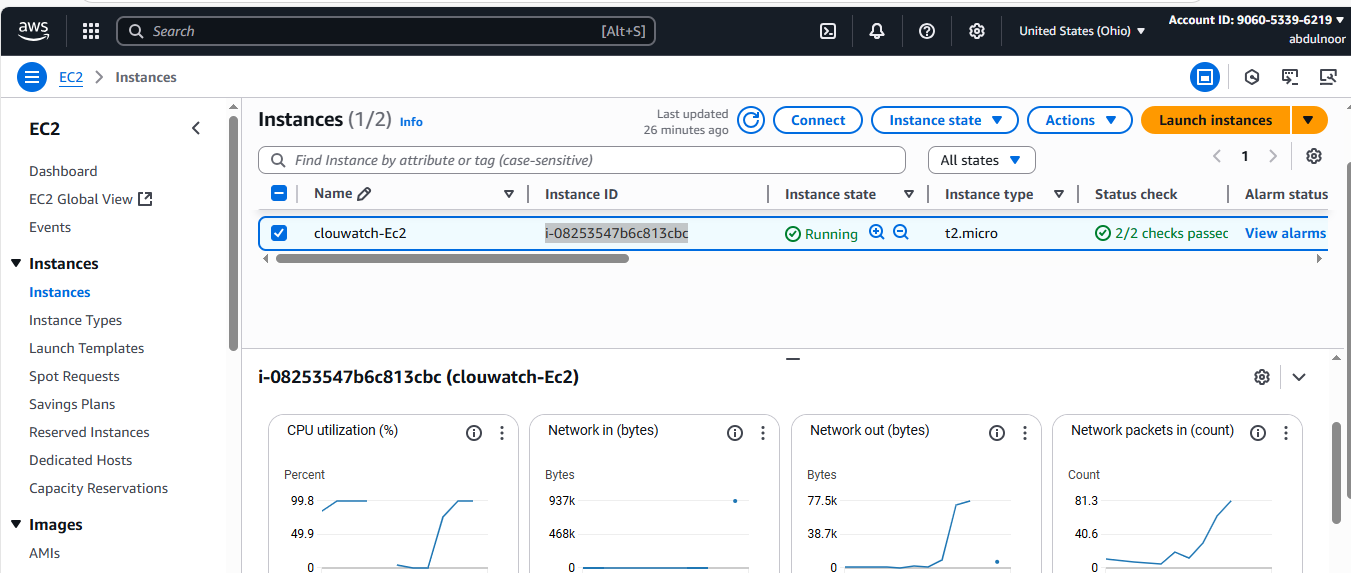
Or I you don’t get 100% use the below

**Yum install stress -y**

**Ex: stress --cpu 2 --timeout 60**



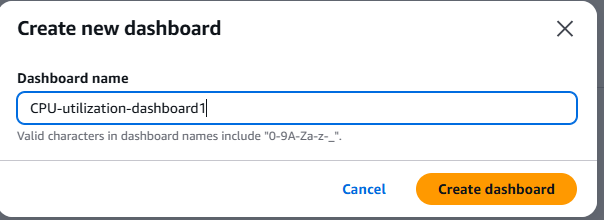




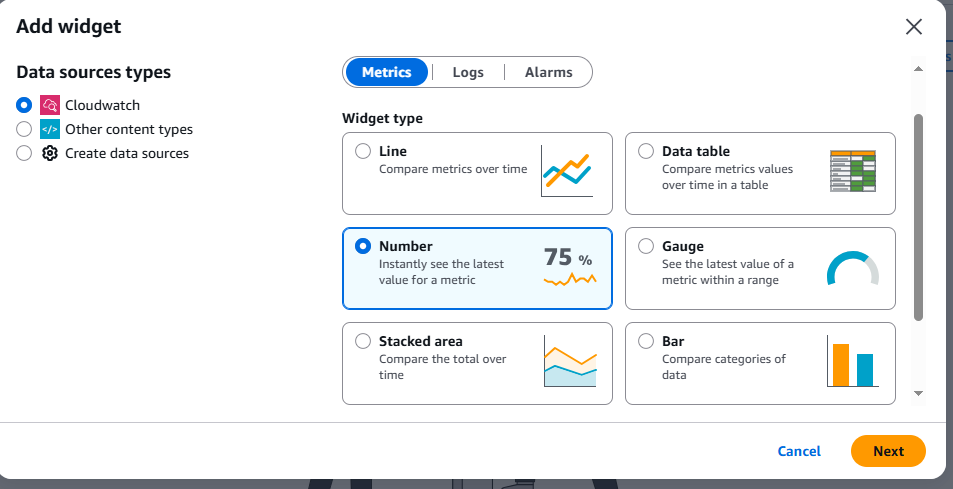
**4) Create one alarm to send alert to email if the cpu utilization is more than 70 percent.**

Create your own dashboard

Goto – cloudwatch – dashboard – create dashboard

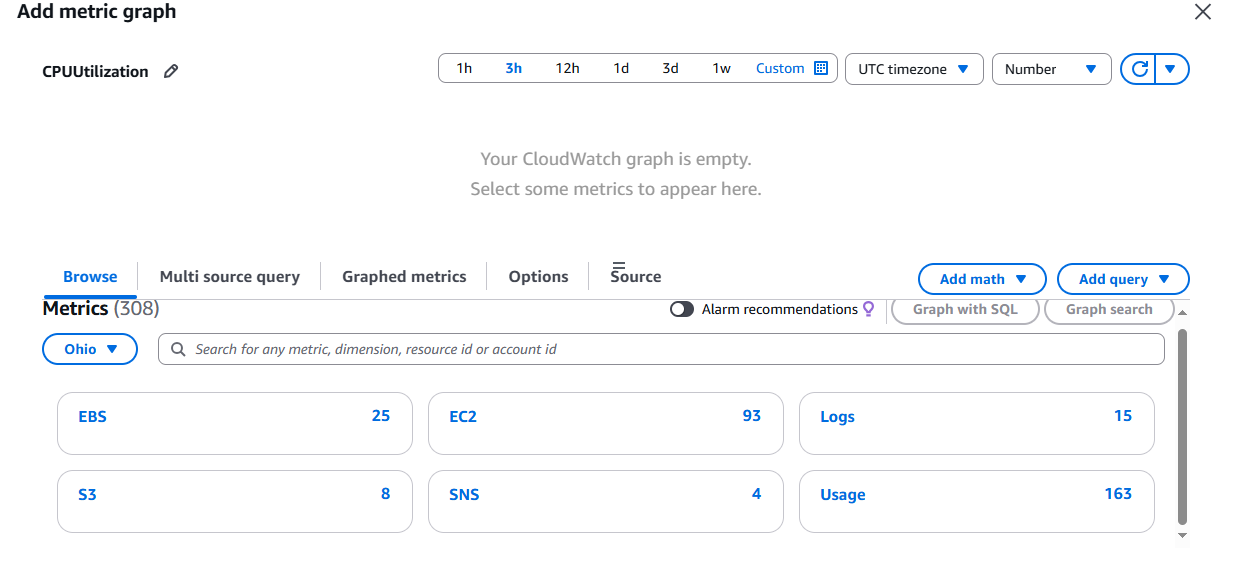


In the one ashboard I can add all the cpu’s like I I 5 instances then I add them here.



These are the widgets means how your dashboard looks as you want you can select and this can be change after creating. Or now I am going with numeric.

Click on next

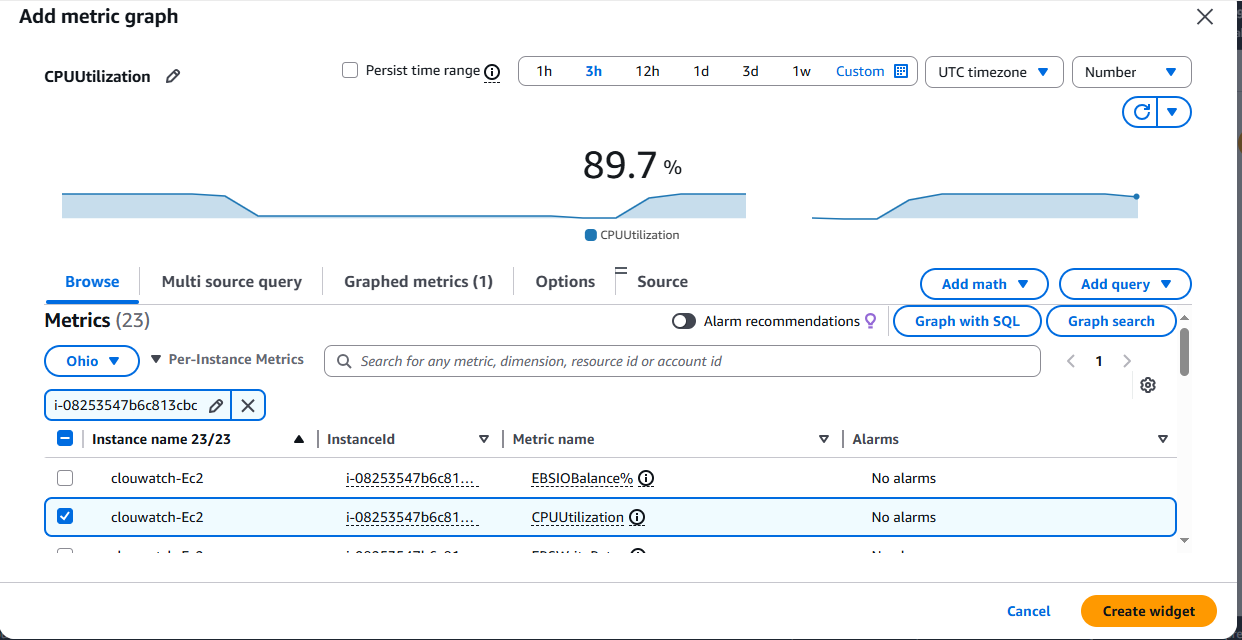


These default matrics given by the aws by default. If you want you can use. Or now I am using for cpu utilizations so I am using EC2.

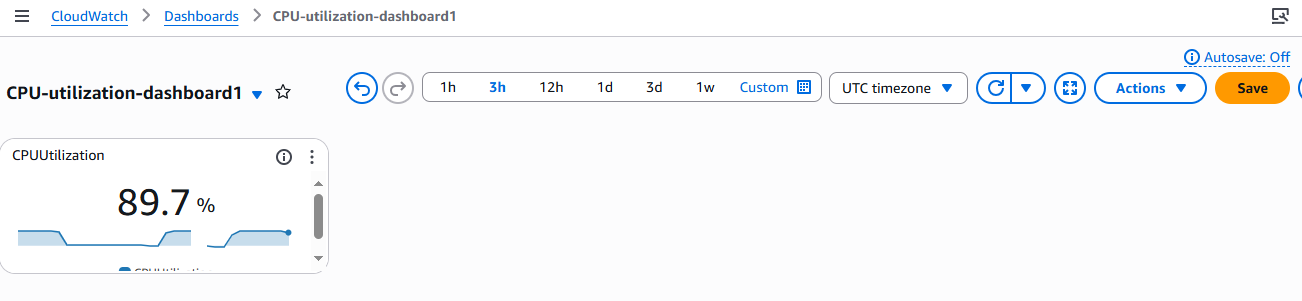
Click on Ec2

Then click on preinstance matrics

Here in search give your EC2 instance ID and select CPU utilization as we are monitoring.



Click on create widget



I have created dash board

I I have one more CPU utilization then I come back CPUU and add by instance ID and I create different dash boards.

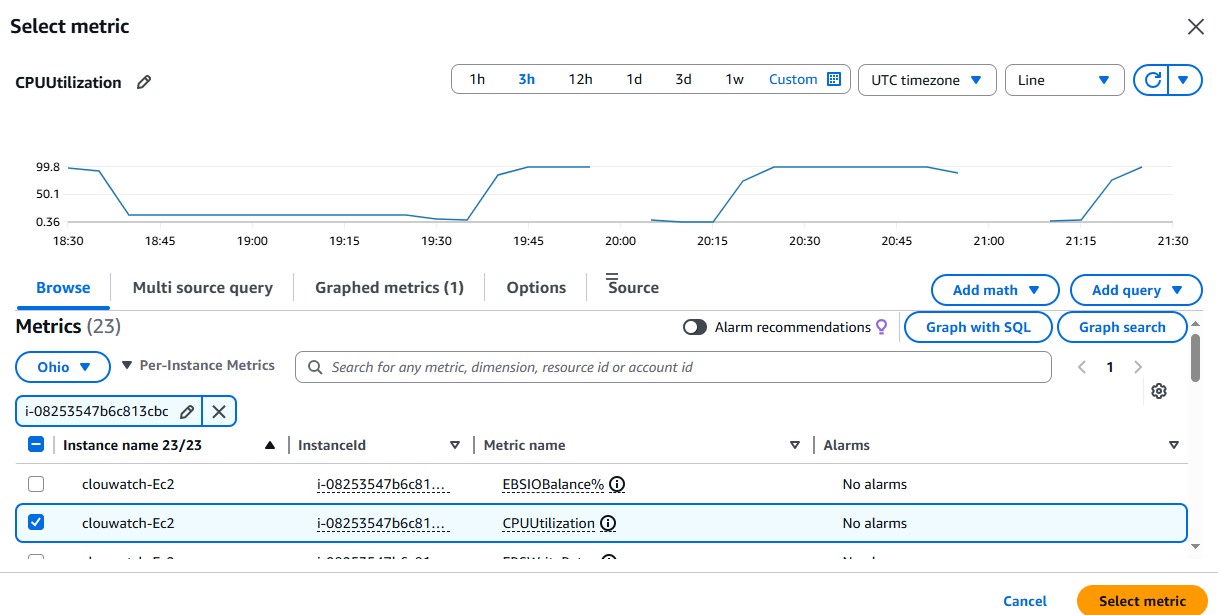
You can customize dashboard and you can edit – widgets – numbers

NOW\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

I want to create Alarm If the CPU utilization reches morethan 70 > then I should get the Alarm

Create one Alarm

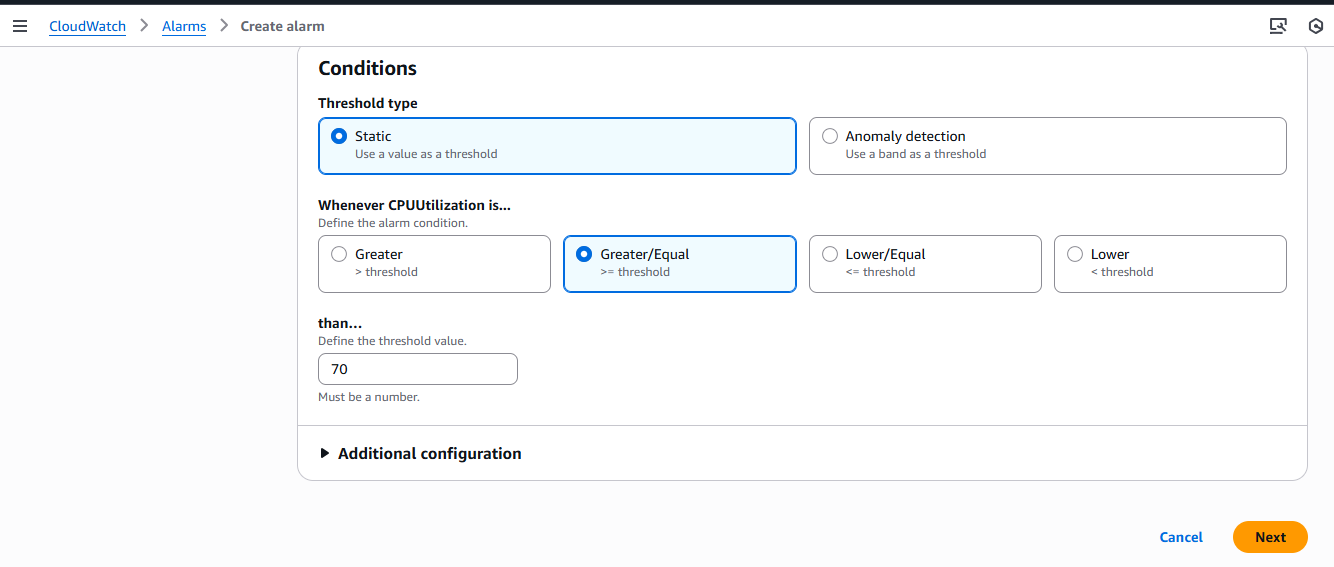
Goto – Cloudwatch – Alarm – In Alarm – Create Alarm – Select Matric – EC2 – Preinstances – Give in search Instance ID –



Click on select matric

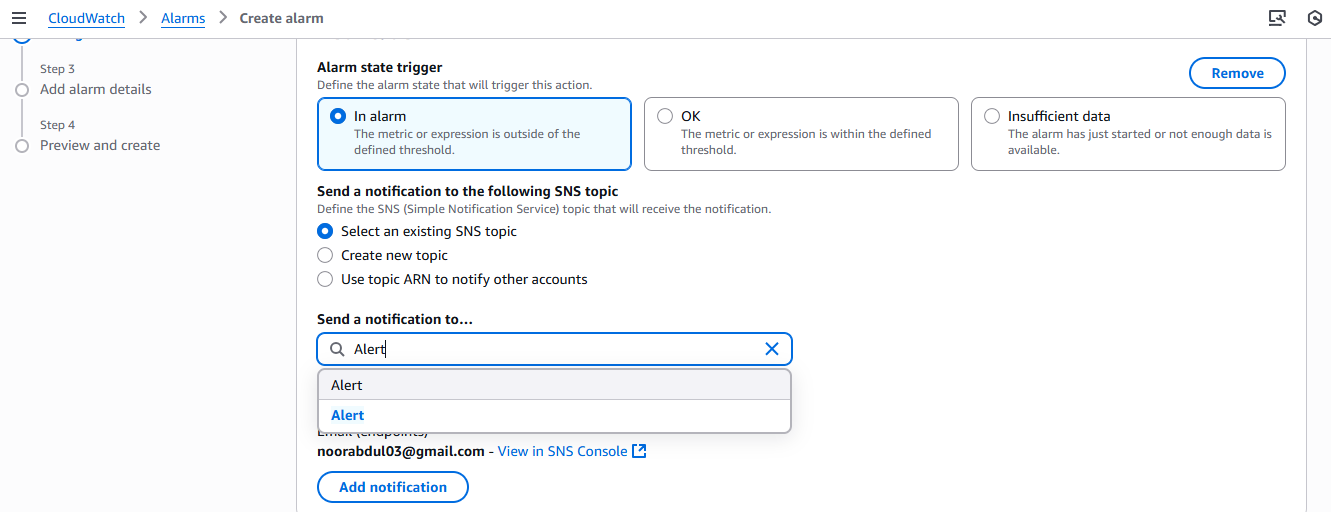
Here you can find diferent type of configuration

For ex: give greter than/eual and give 70



Click on next

How the notification will send like SNS topic alert we created alreay use it



And I you want to take any EC2 action. If it is more than 70 % then like I want stop this instance or terminate you can but for now I am not taking any EC2 actions.

Click on next

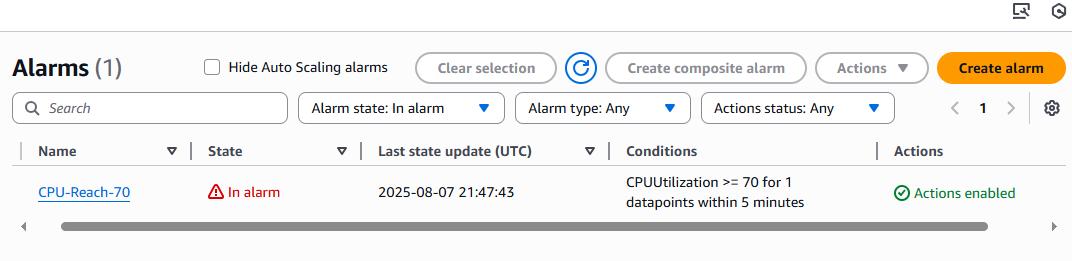
Give alarm name

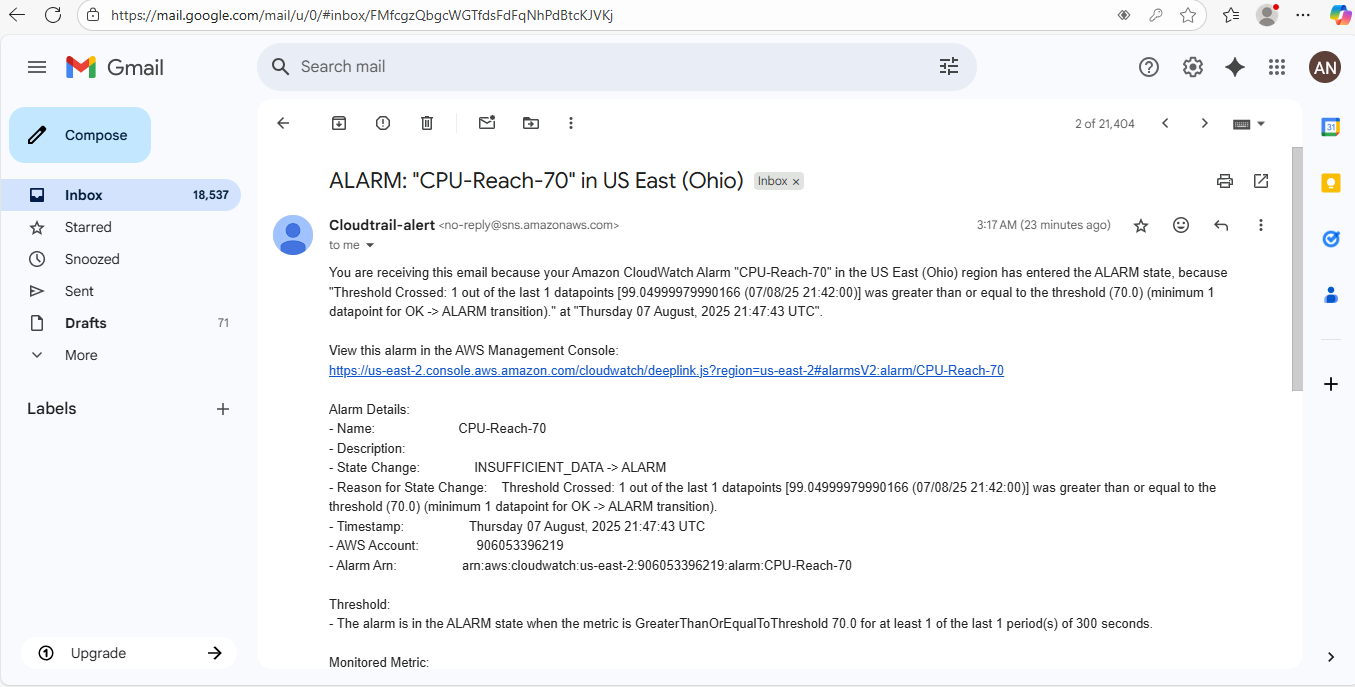
CPU-Reach-70%

Click on next

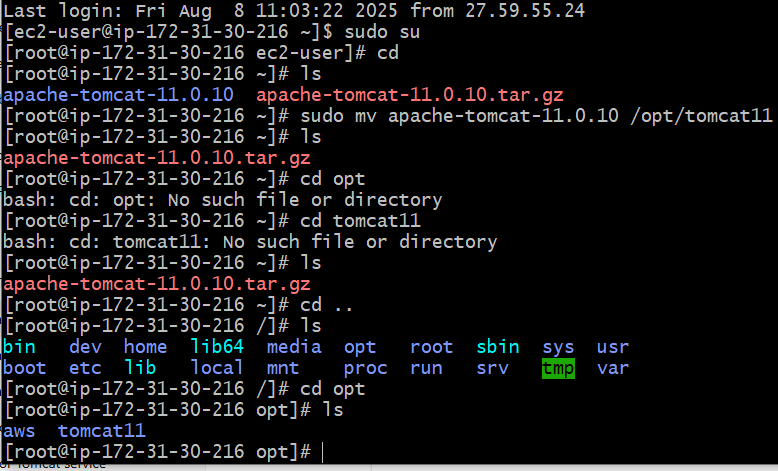
Click on create alarm

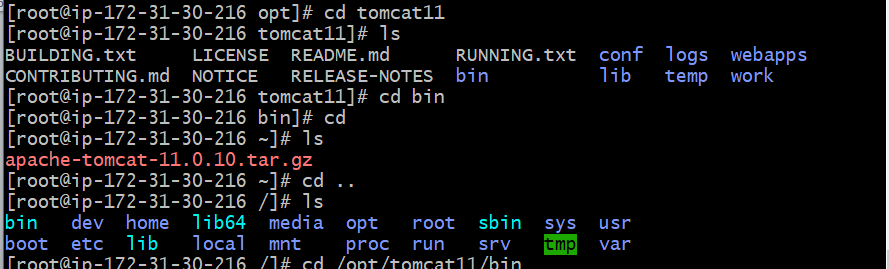
See as o now it is in-suicient data and then it will creat Alarm

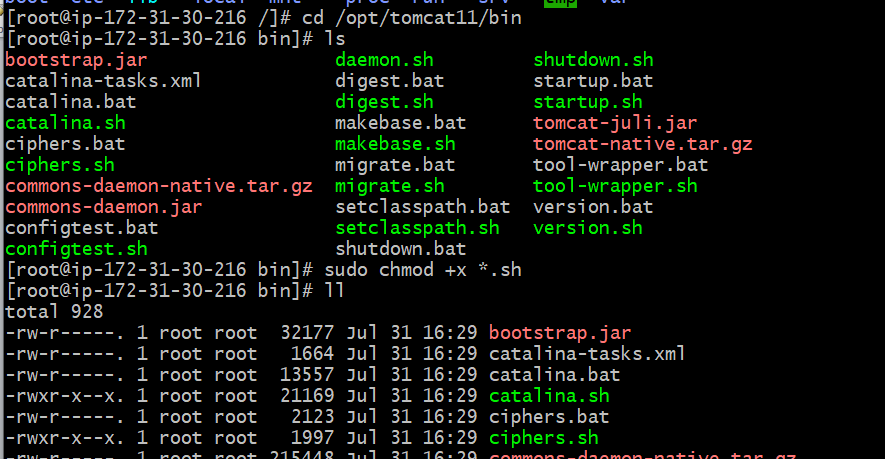




**5) Create Dashboard and monitor tomcat service wether it is running or not and send the alert.**







**cd /opt/tomcat9/bin**

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

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

**Notes……**

 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.

Notes end……

**Step 2:**

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

**Chmod 777 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 us-east-2 \

--metric-name TomcatStatus \

--namespace "tomcat" \

--value "$status" \

--dimensions InstanceId="$INSTANCE\_ID"

**step 3:**

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

**Step 4:**

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

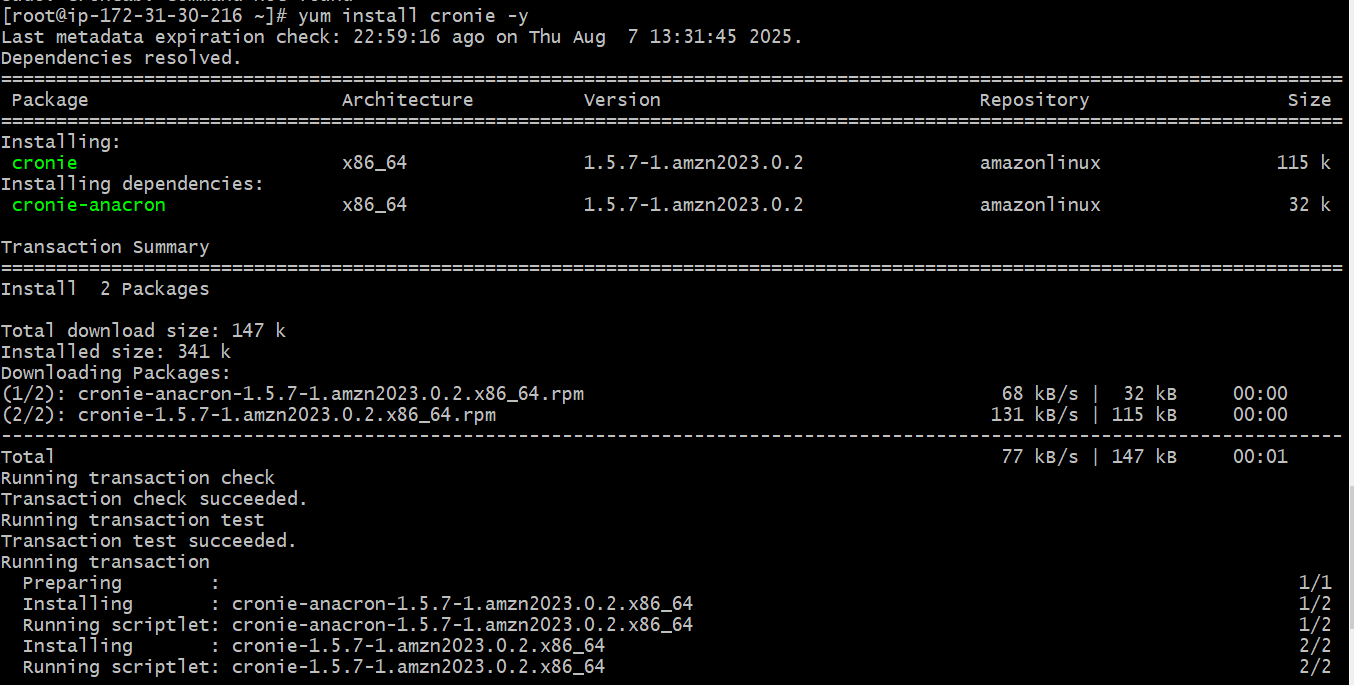
sudo: crontab: command not found

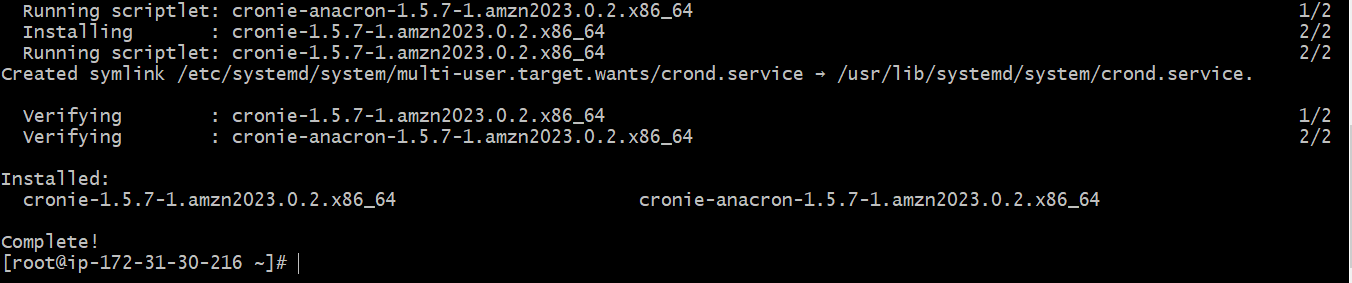
**I this error came it’s mean you don’t have crontab configured in your ec2**

**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



After installing, run:

Check if it’s working command

crontab –e

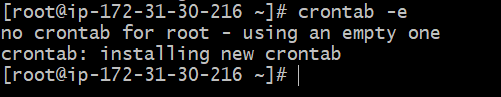


when you run crontab -e as **root**, you should add exactly this below line



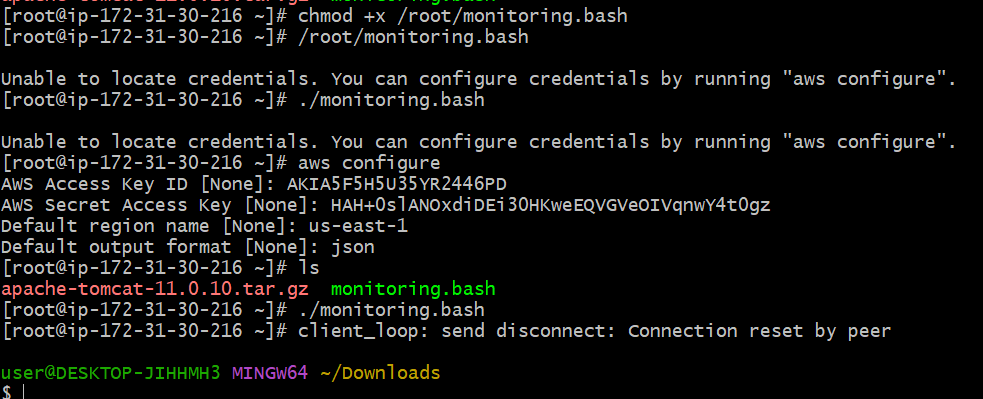
\* \* \* \* \* /root/monitoring.bash

Save it

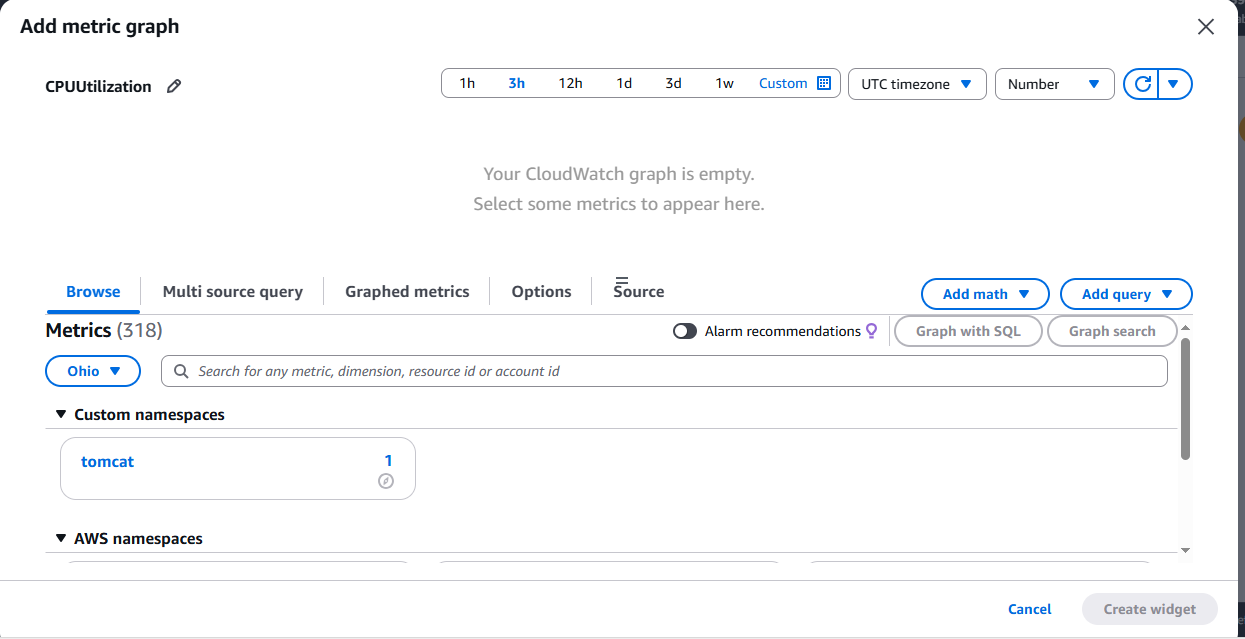


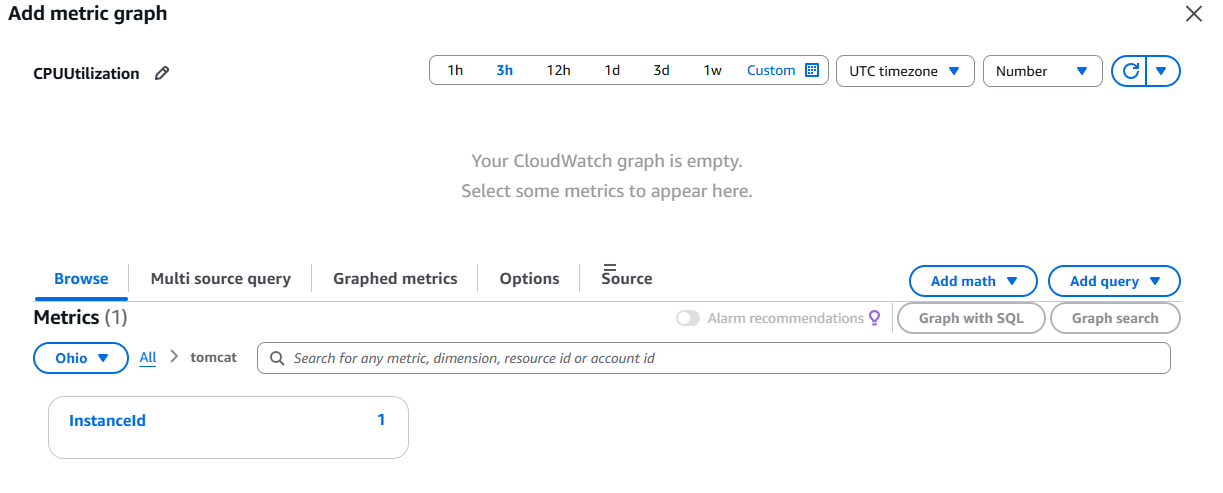
Step

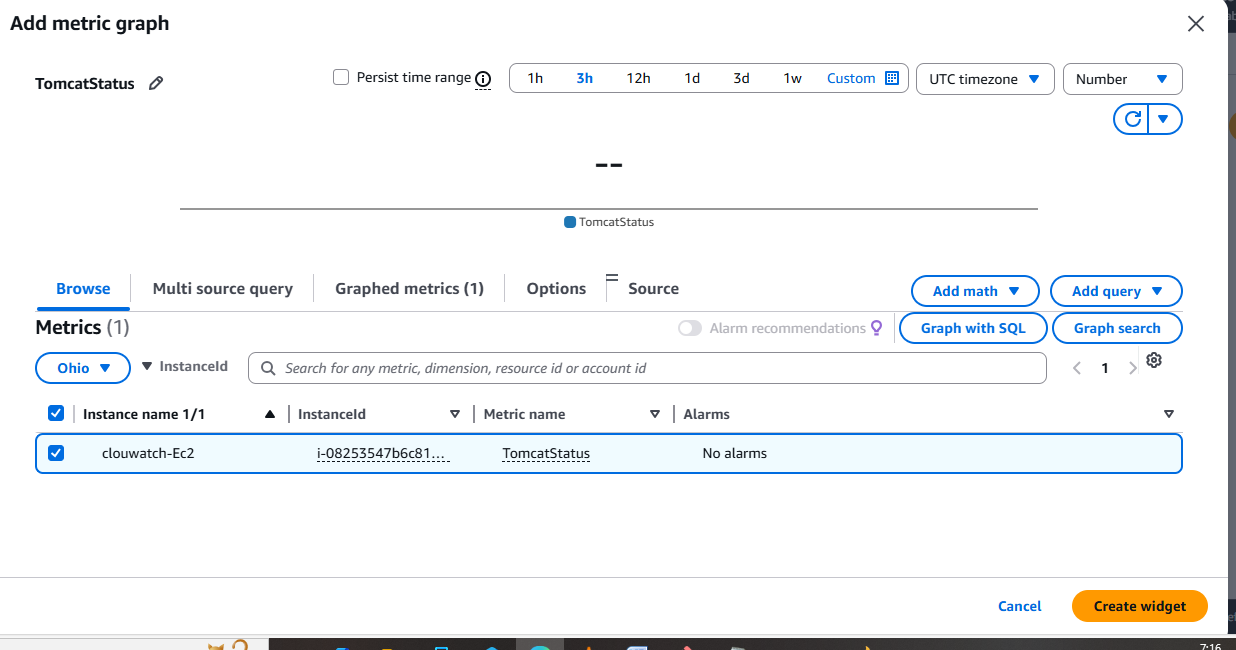
Execute bash scripting file

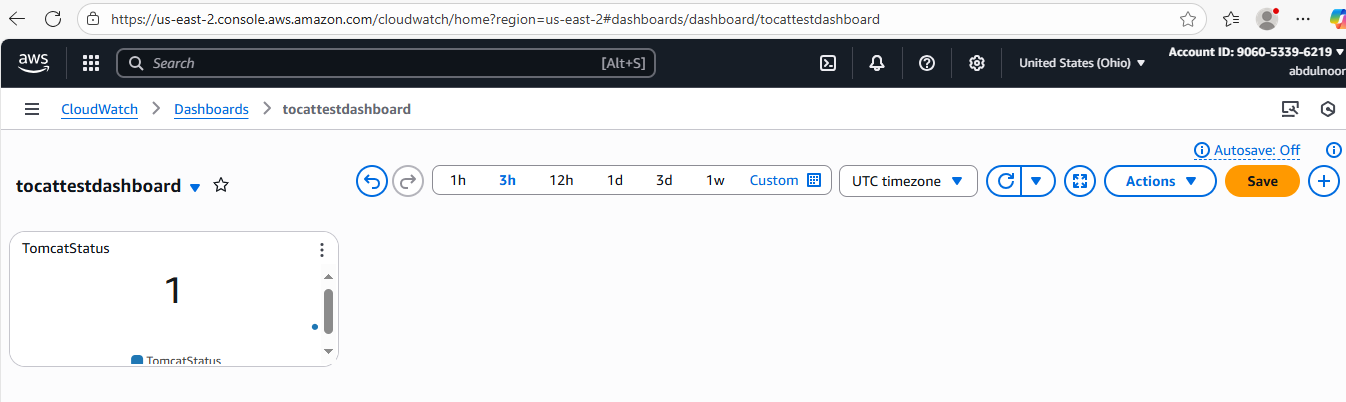


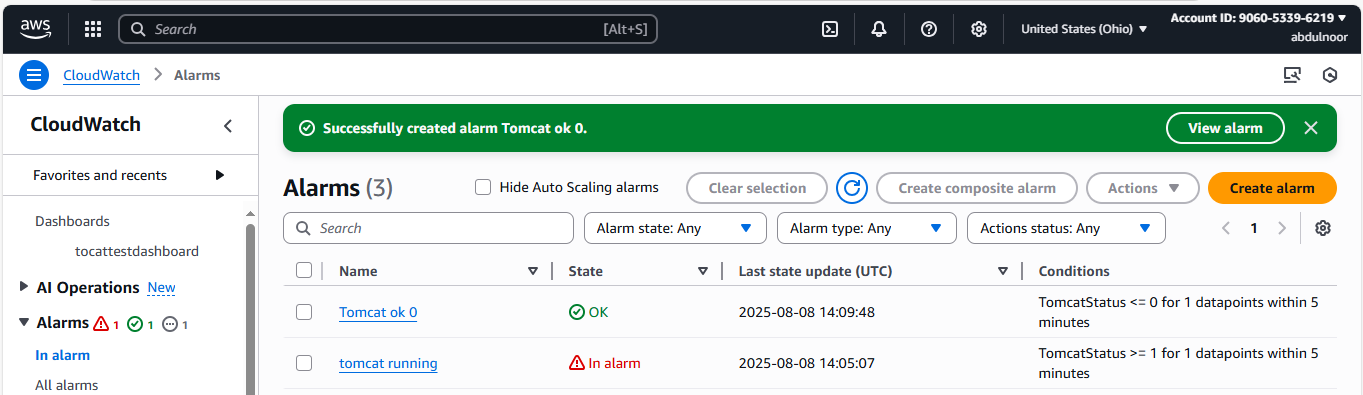
1. **Once you run the monitoring.bash it will reflect in CloudWatch Dashboard.**
2. **Now create Dashboard for Tomcat you can see the tomcat select metric in number and then if metric >=1 tomcat running <=0 tomcat nor running then select SNS Topic(if not there then create) → give alarm name and done.**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



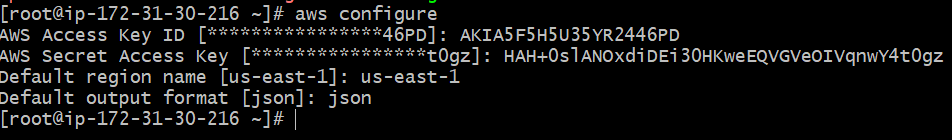






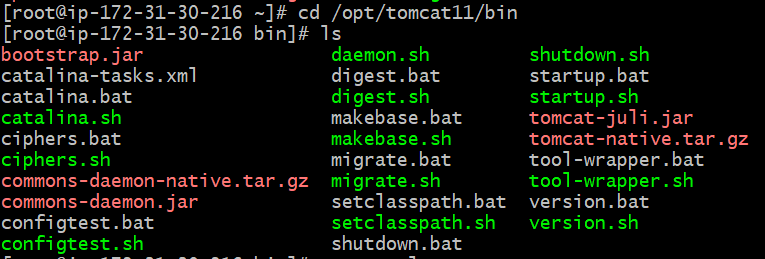


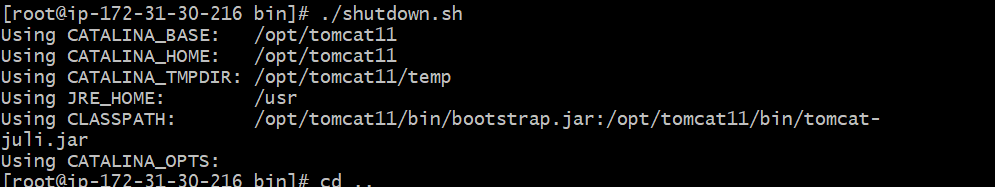
To stop tomcat : first check aws cli is configured or not If not then do

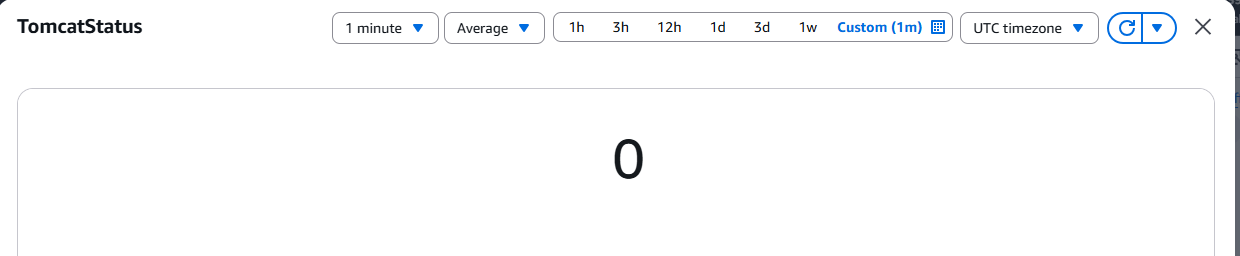


Stop tomcat

As my tomcat file is in the below location so to shut down also I need to be there

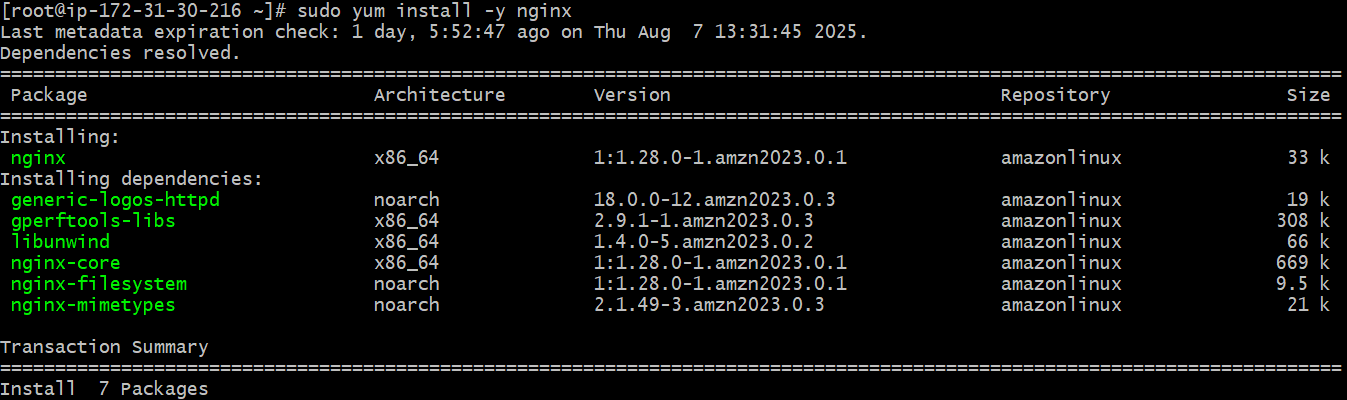


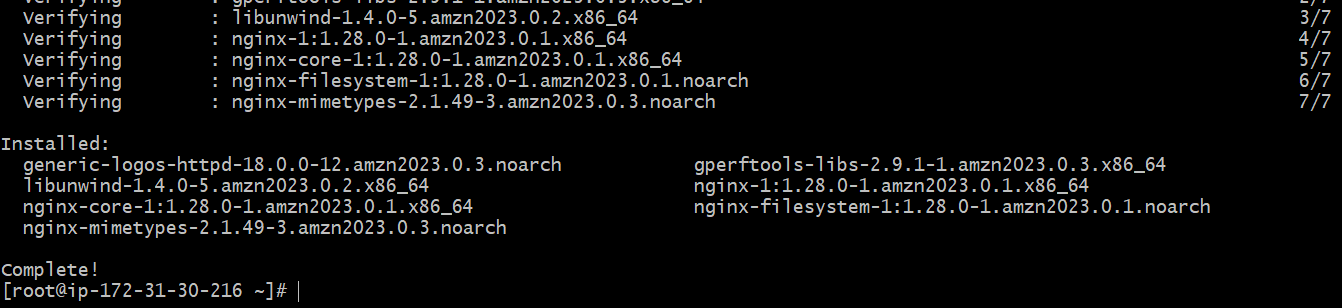


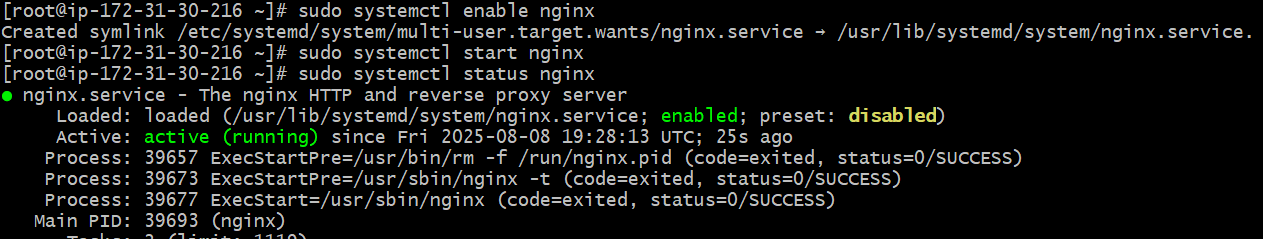


**6) Create Dashboard and monitor nginx service to send the alert if nginx is not running.**

**Step 1 :Install Nginx**







Step 2:



Why creating nginx script ile in opt because mostly sotware related files create in this directory location. for future reference.

Insert the file and paste the below script and save it. In this script check the region where your aws cloudwatch dashboard created and ec2 according custom namespace appears. So change according to region the below script just by changing region. Or run aws configure command by keeing your related region.

For region ohio

#!/bin/bash

SERVICE="nginx"

REGION="us-east-2" # Ohio region

NAMESPACE="nginx-monitor"

if systemctl is-active --quiet $SERVICE; then

STATUS=1

else

STATUS=0

fi

/usr/bin/aws cloudwatch put-metric-data \

--metric-name NginxStatus \

--namespace "$NAMESPACE" \

--value "$STATUS" \

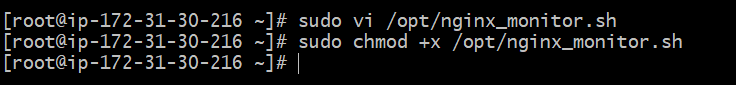
--region "$REGION"

Step 3:

Give permission to file

sudo chmod +x /opt/nginx\_monitor.sh

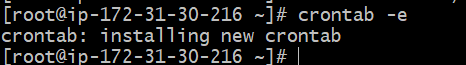
or give chmod 777 /opt/nginx\_monitor.sh (but not preferable)

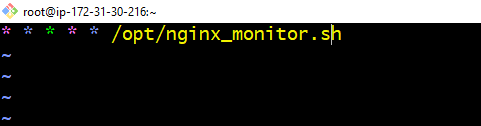


Step 4:

Add to Cron

crontab –e





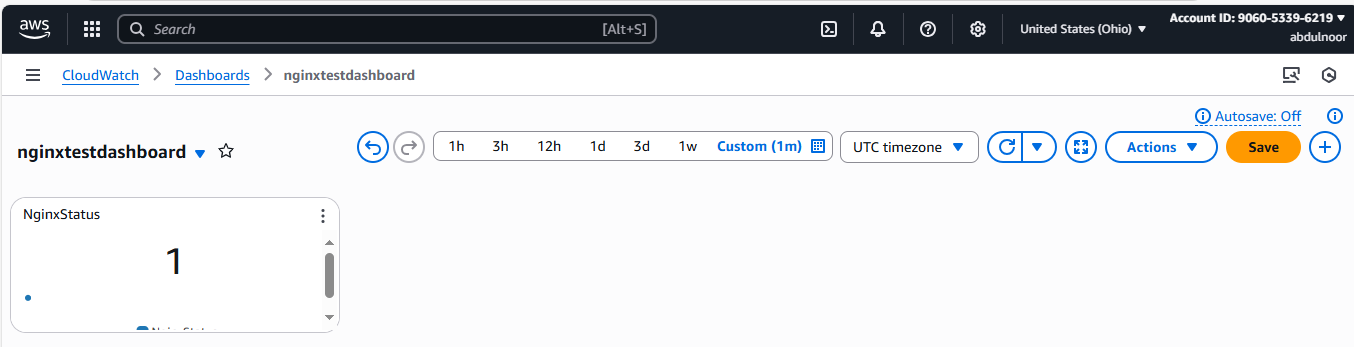
Now run the script

bash /opt/nginx\_monitor.sh

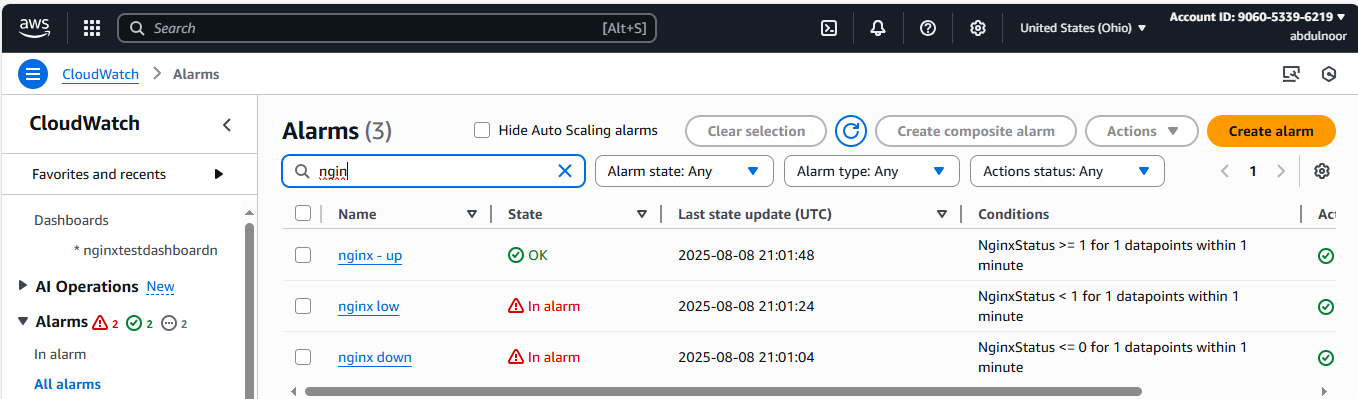
or

./opt/nginx\_monitor.sh

Before creating dashboard in cloudwatch



Now create alarm



I have deleted the SNS subscription that’s why not getting an email notification. I again create SNS subscription.

