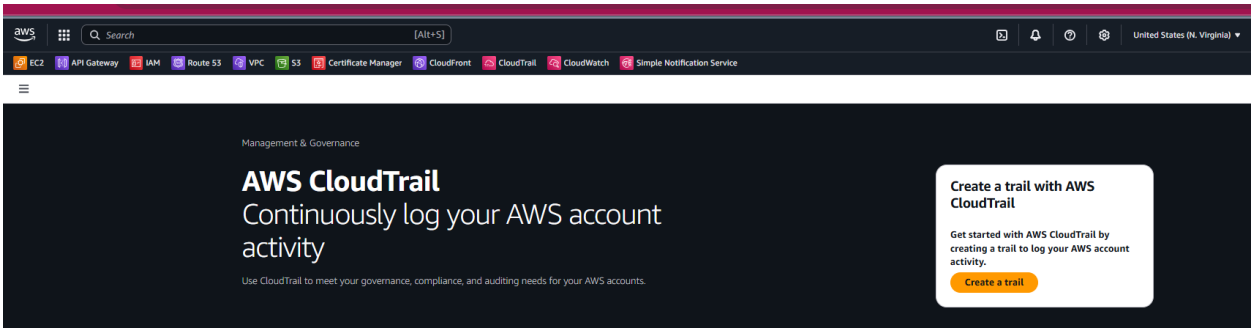
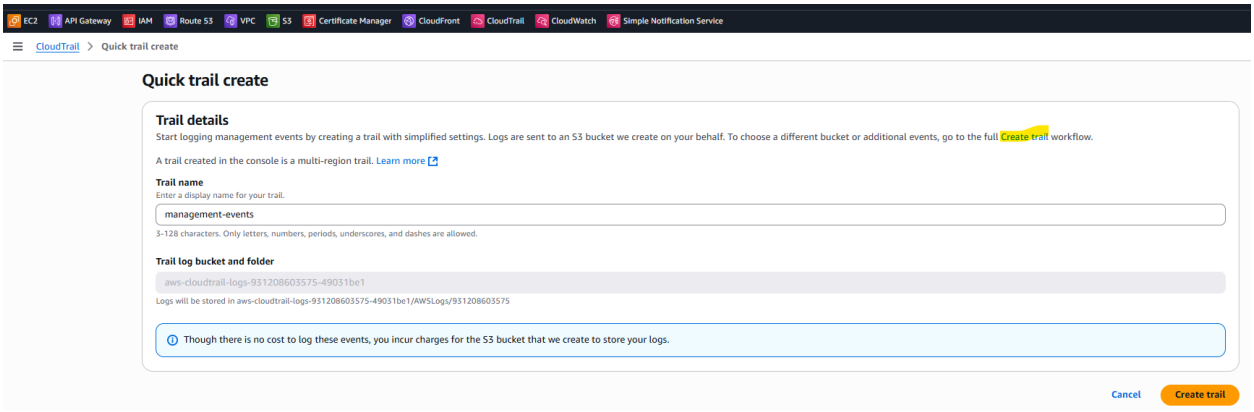
1) Enable CloudTrail monitoring and store the events in s3 and CloudWatch log events.

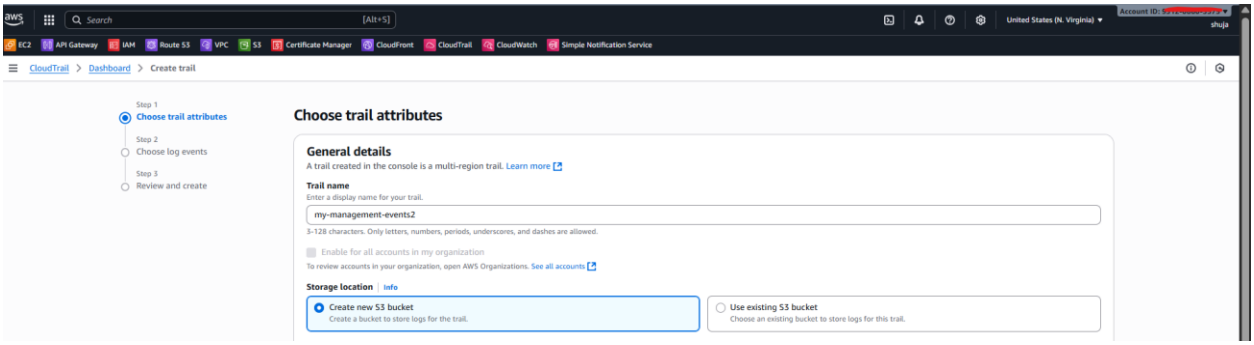
Goto CloudTrail service and click on Create a trail:



Then click on Create trail to get the complete workflow of cloudtrail:



Enter the trail name, Storage Location, Create a new s3 bucket :



and give the bucket name , Log file validation and enable SNS delivery notification:

Trail log bucket and folder

Enter a new S3 bucket name and folder (prefix) to store your logs. Bucket names must be globally unique.

aws-cloudtrail-logs-horizon-bucket

Logs will be stored in aws-cloudtrail-logs-horizon-bucket/AWSLogs/931208603575

Log file SSE-KMS encryption | Info

☐ Enabled

▼ **Additional settings**

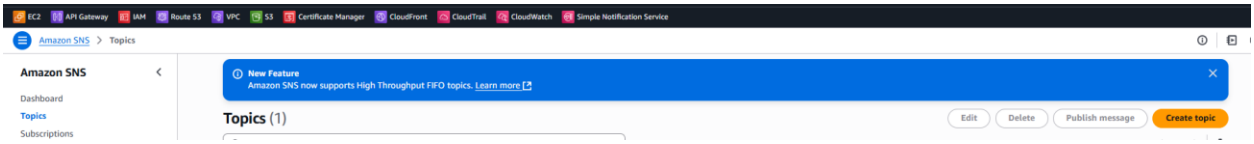
Log file validation | Info

☑ Enabled

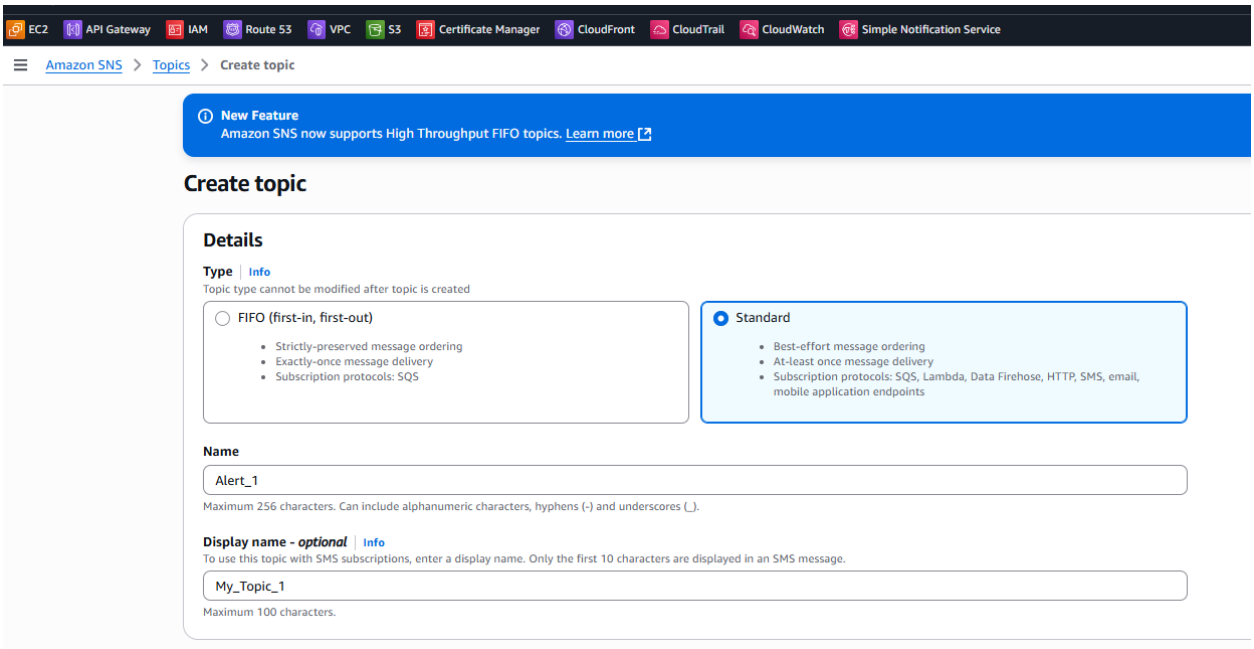
SNS notification delivery | Info

☑ Enabled

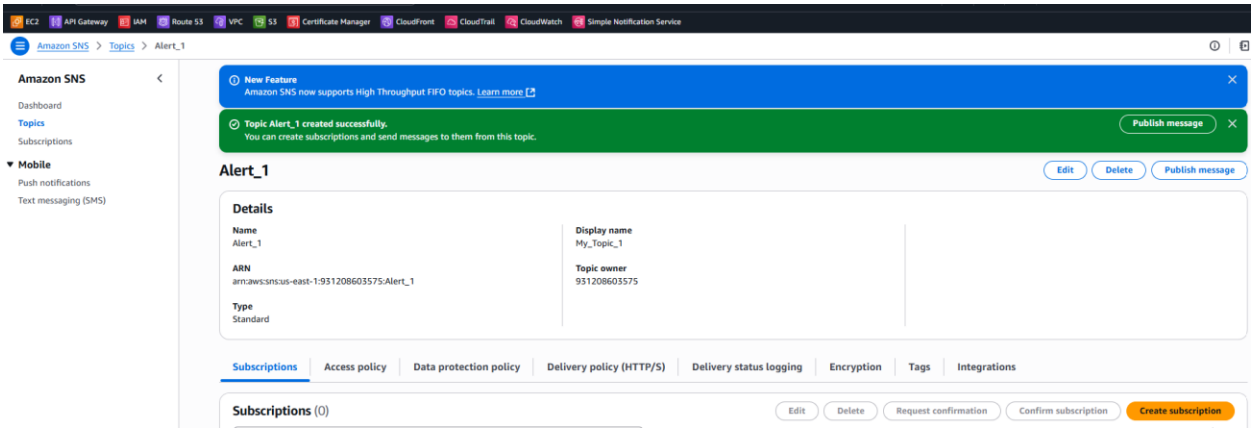
Goto SNS service and Create topic:



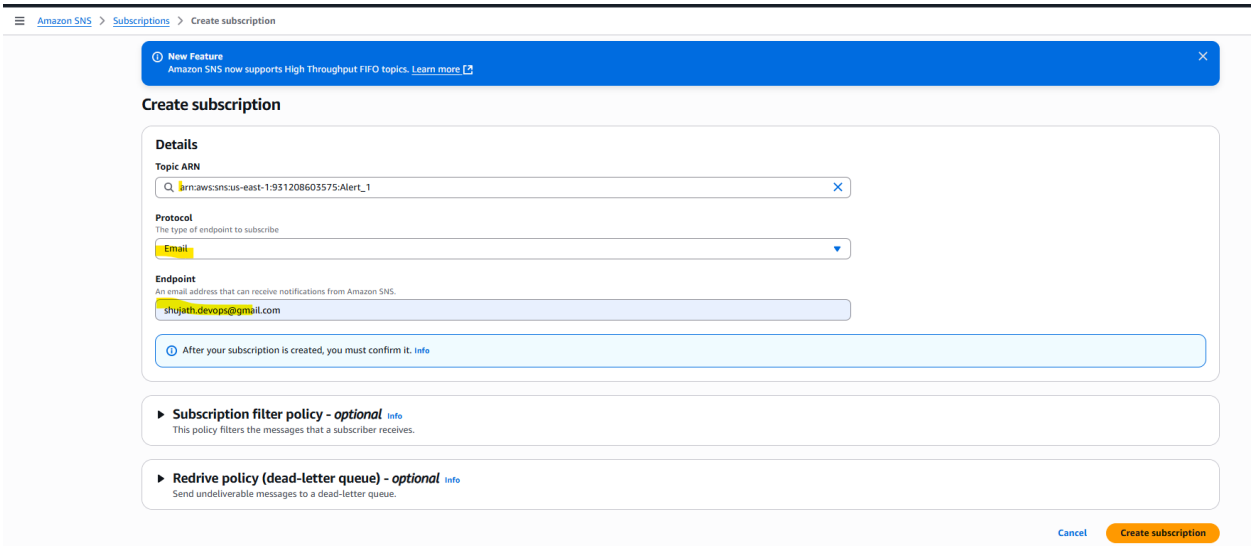
Select the standard and give the name of the topic and Display name and Create topic:



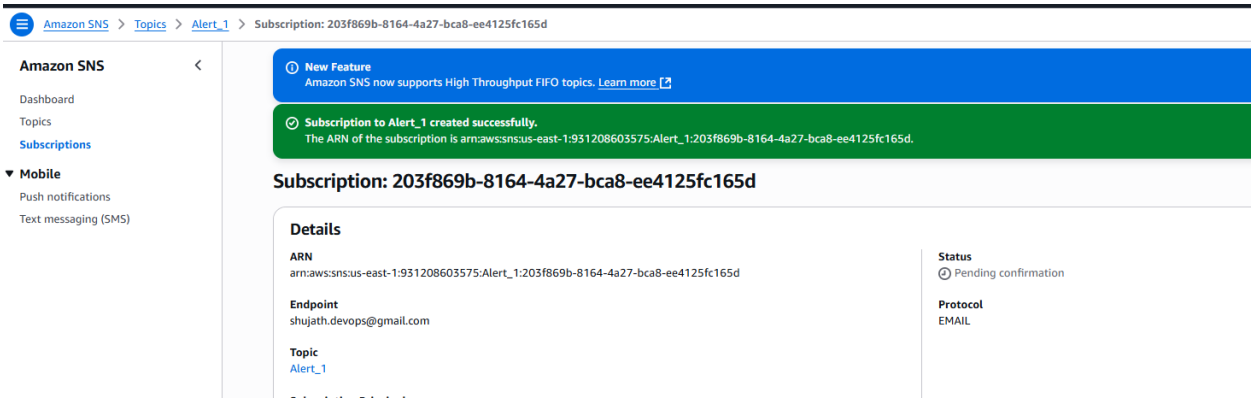
Next, Create the Subscription:



Select the protocol as ‘Email’ and Endpoint as ‘your email id’ and click Create Subscription:

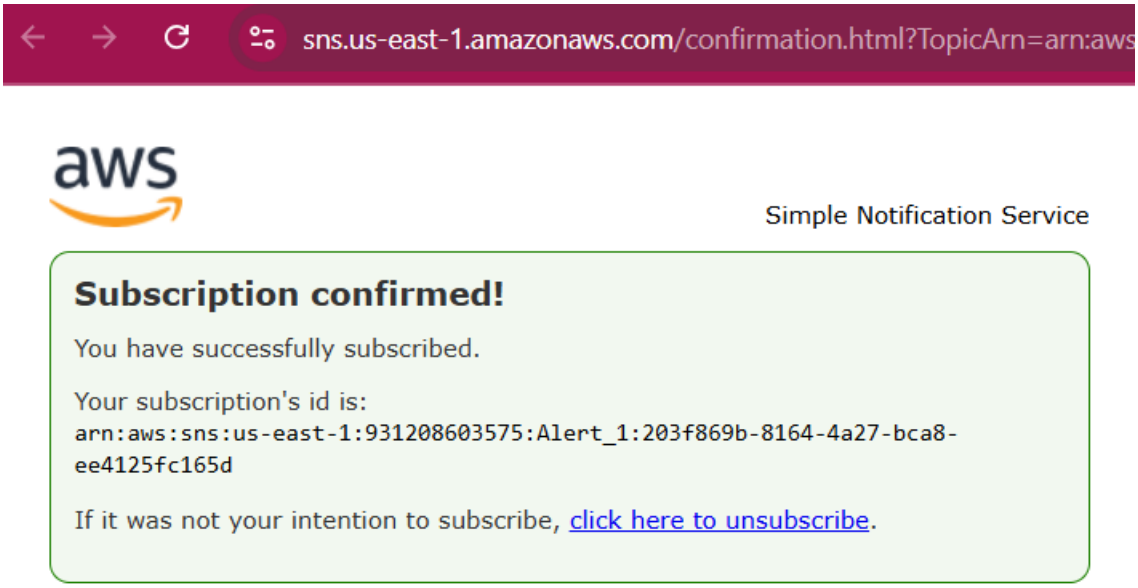


Then you will get status as Pending confirmation:

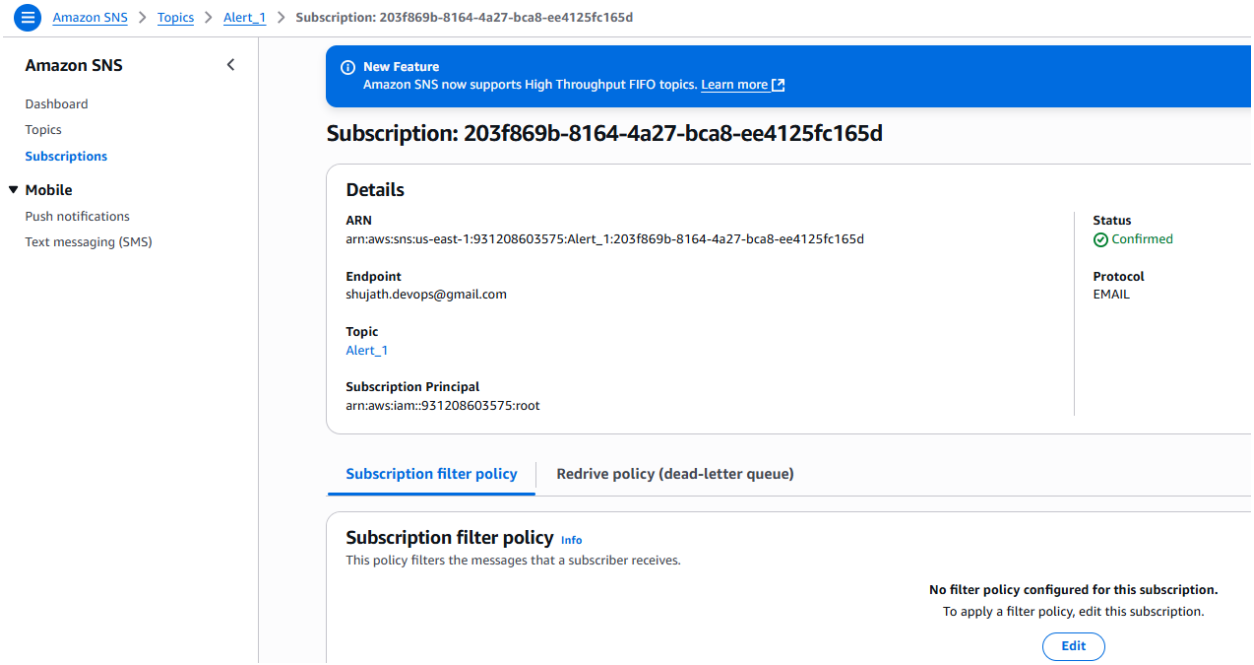


Goto your email and confirm the subscription:

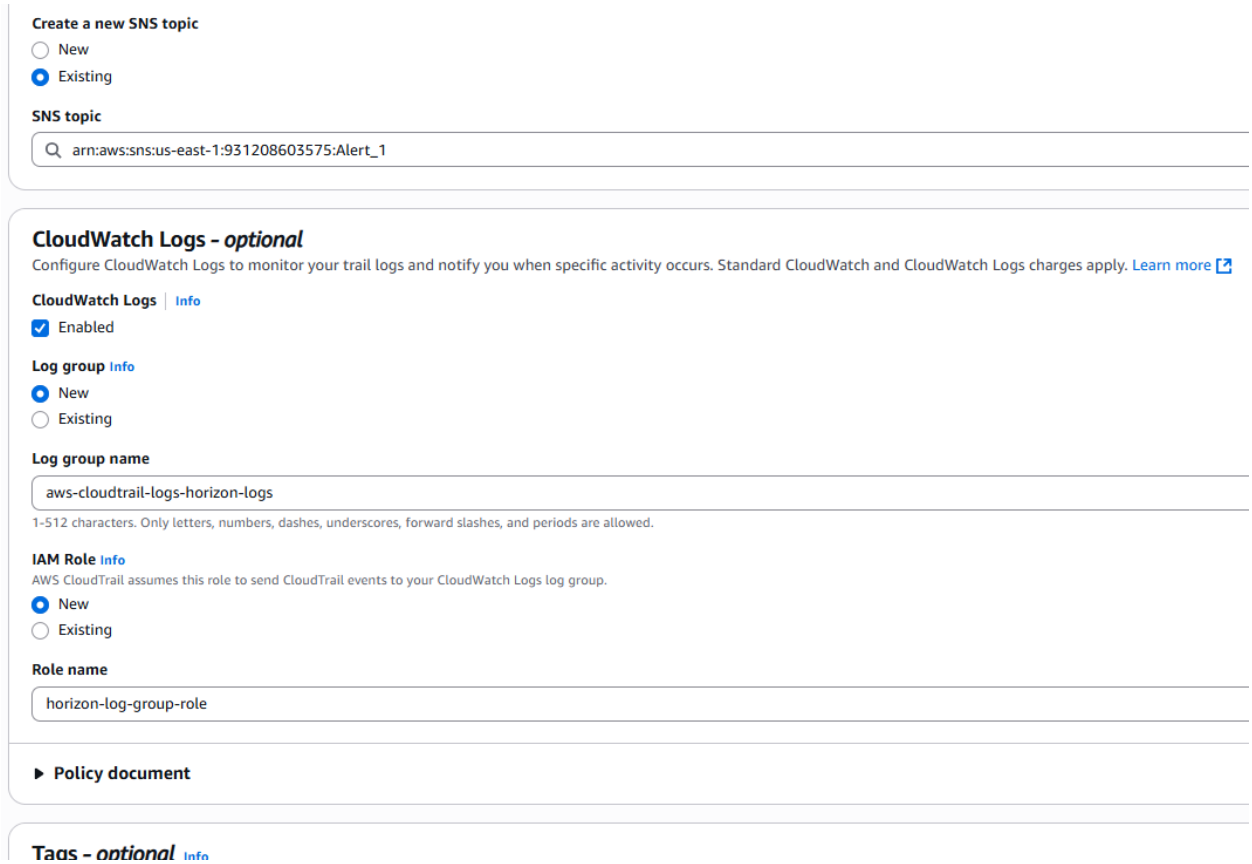
To confirm this subscription, click or visit the link below (If this was in error no action is necessary):
Confirm subscription



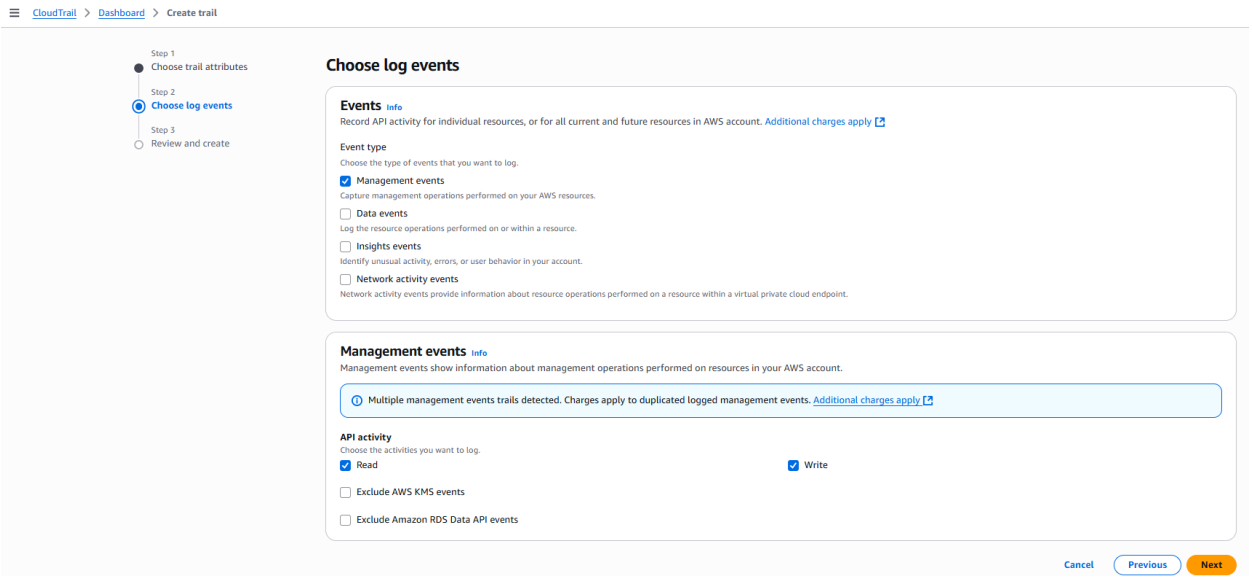
Then you will get the status as confirmed:



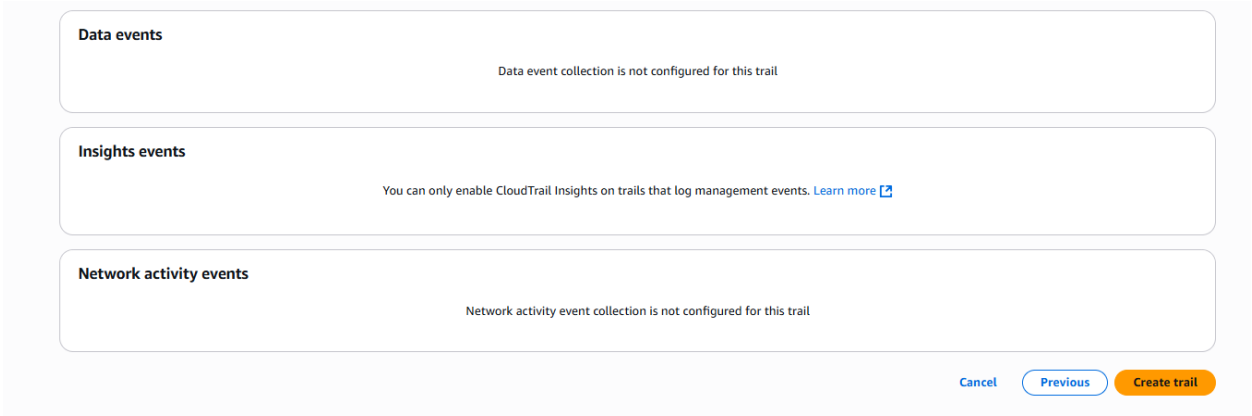
Then Enable the CloudWatch logs and give the log group name and iam role name[log group and iam role will be created by default by aws cloudTrail we just need to give any name here:



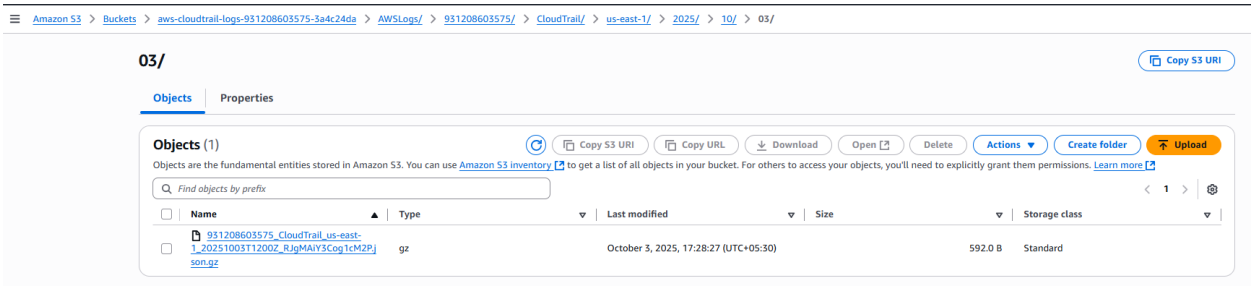
Next, select which events you want to create, here creating ‘Management events’ and select the API Activity as ‘Read’ ‘Write’:



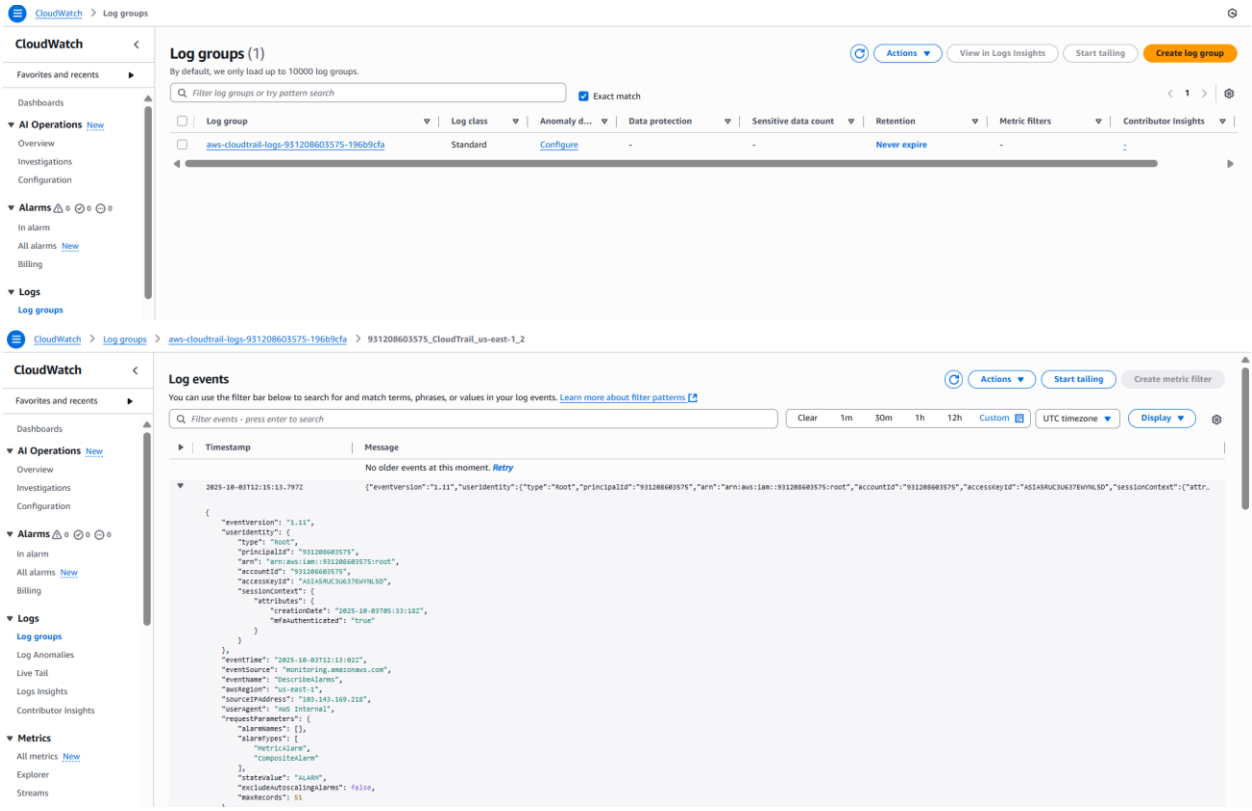
And click on create trail :



Do some activity in s3 like create or delete some buckets and check the logs in the s3 bucket:

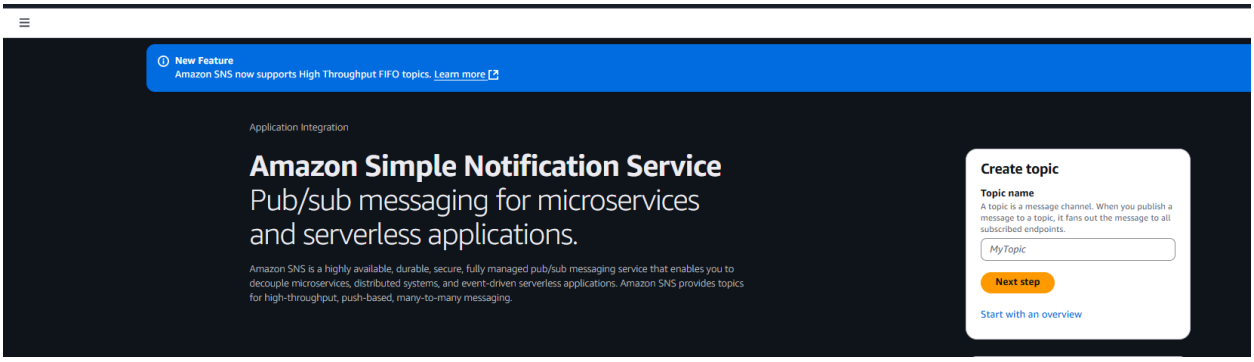


And check the logs in cloudWatch log group:

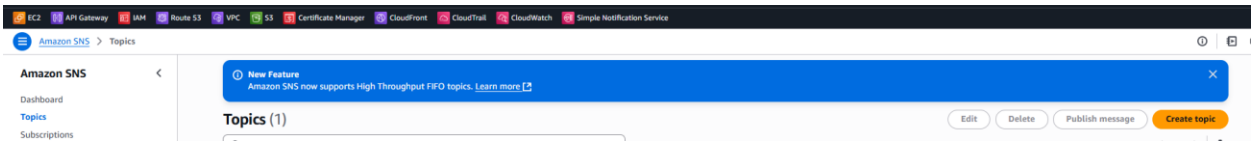


----------done----------

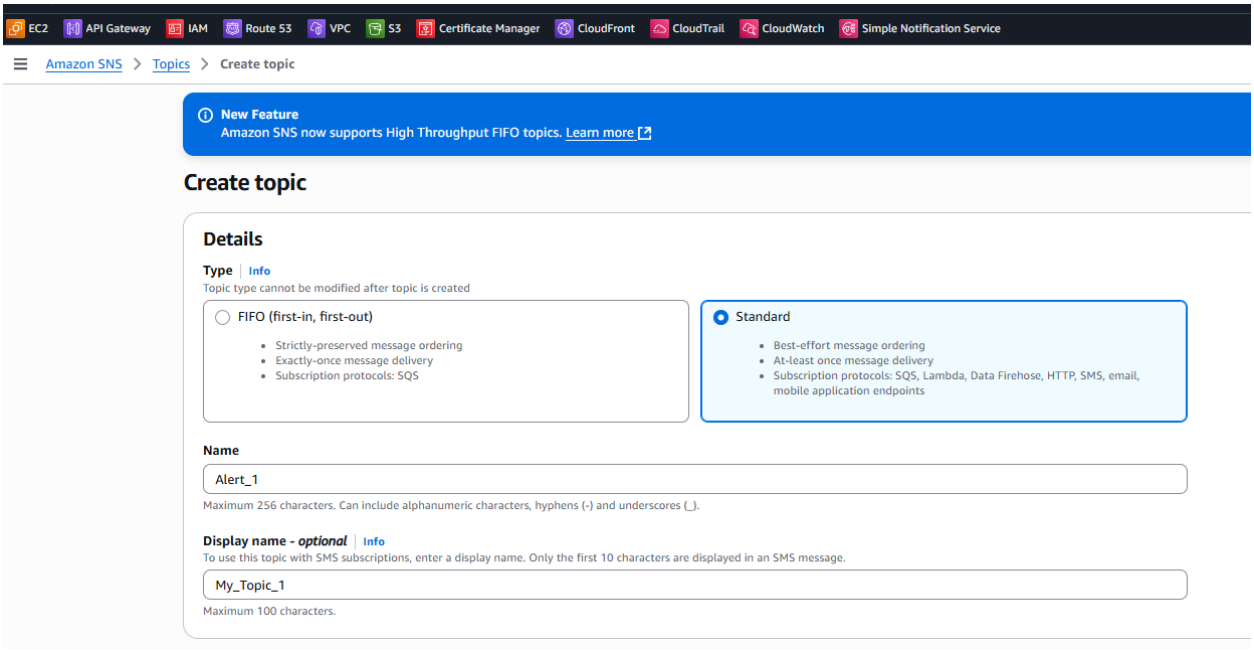
2) Enable SNS for cloudtrial to send alert on email.



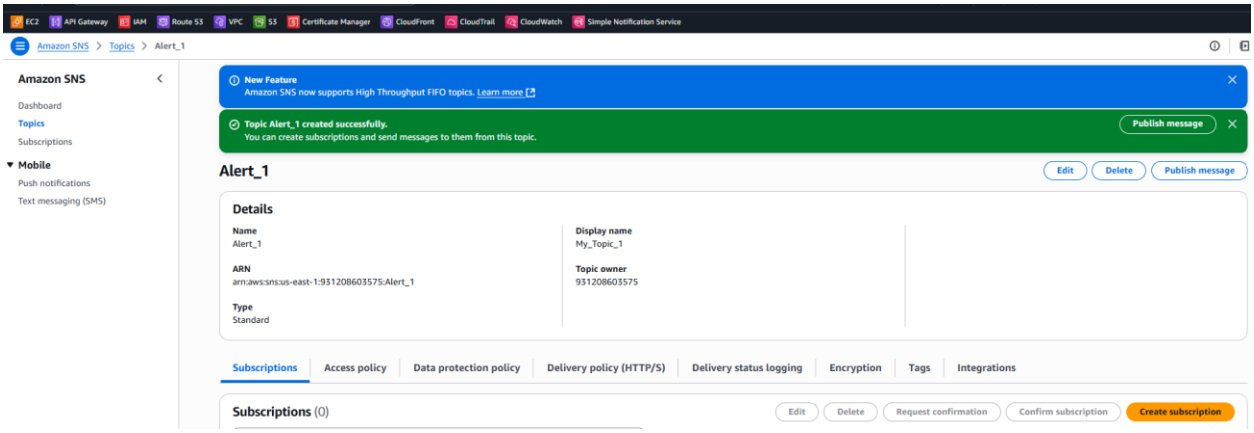
Goto SNS service and Create topic:



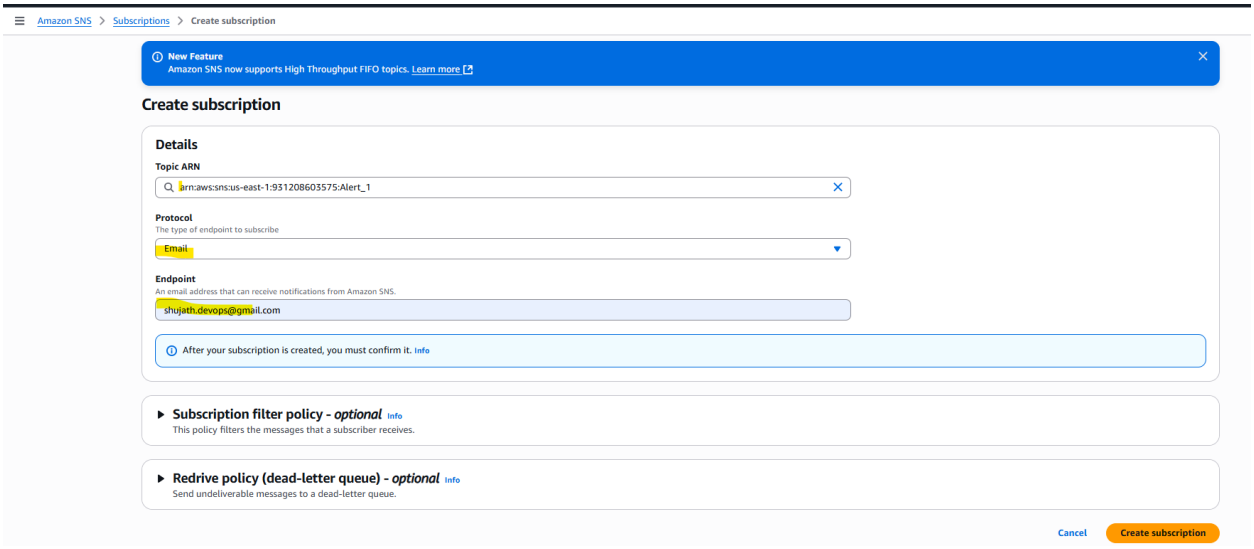
Select the standard and give the name of the topic and Display name and Create topic:



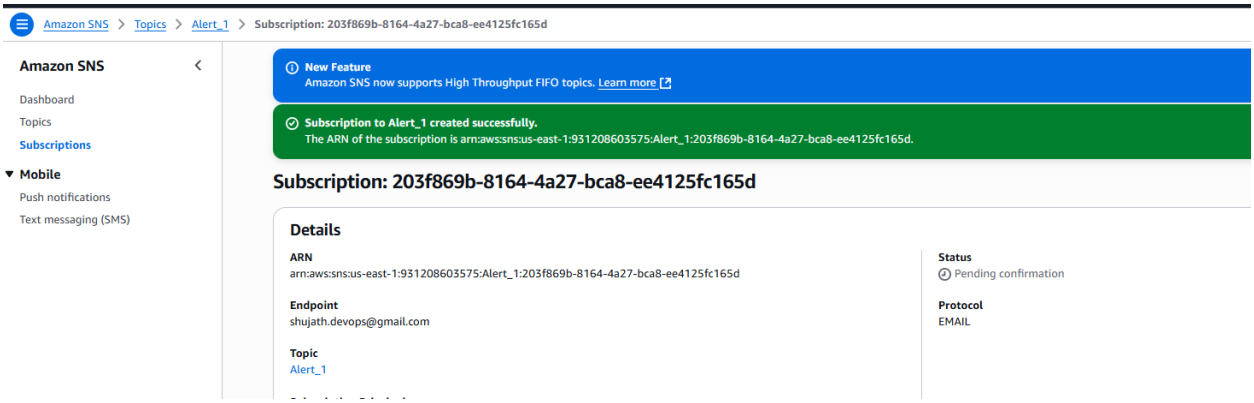
Next, Create the Subscription:



Select the protocol as 'Email' and Endpoint as 'your email id' and click Create Subscription:

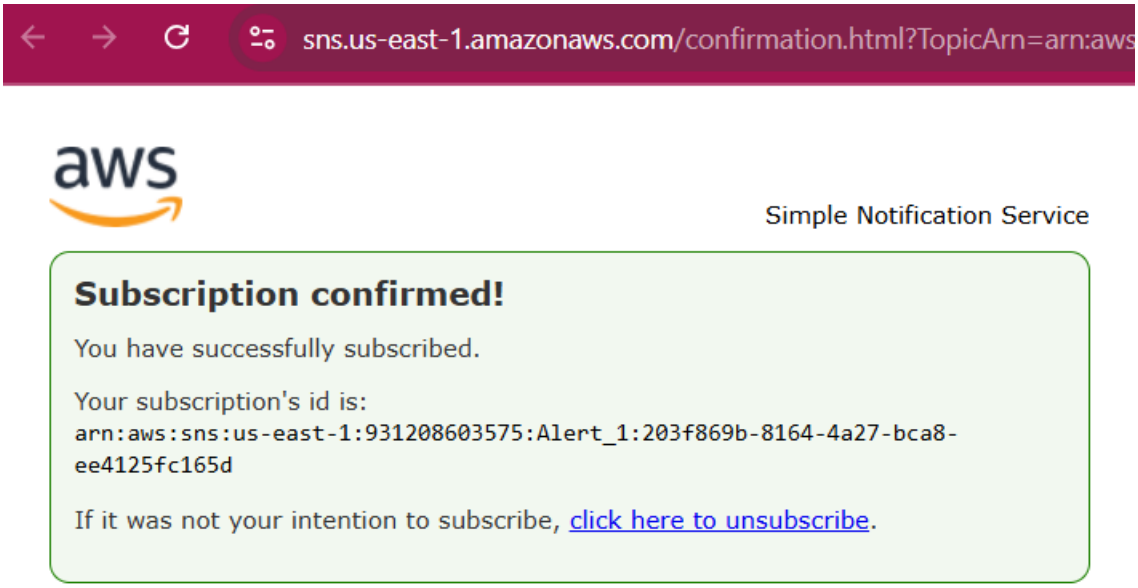


Then you will get status as Pending confirmation:

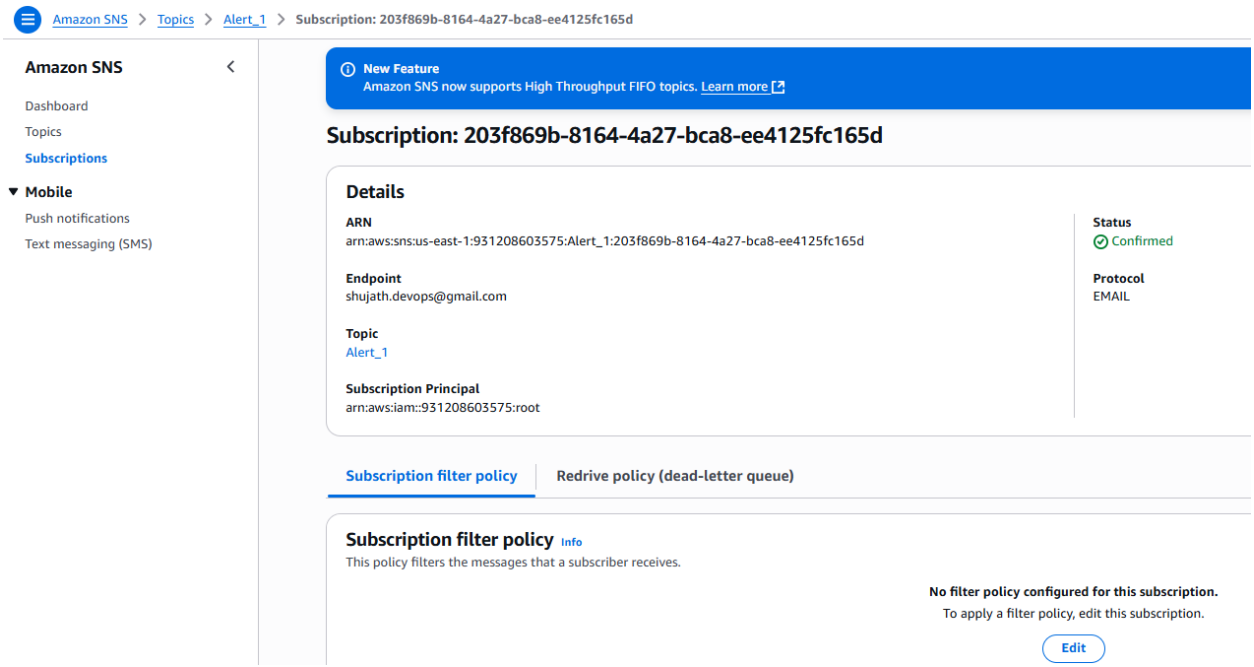


Goto your email and confirm the subscription:

To confirm this subscription, click or visit the link below (If this was in error no action is necessary):
Confirm subscription



Then you will get the status as confirmed:



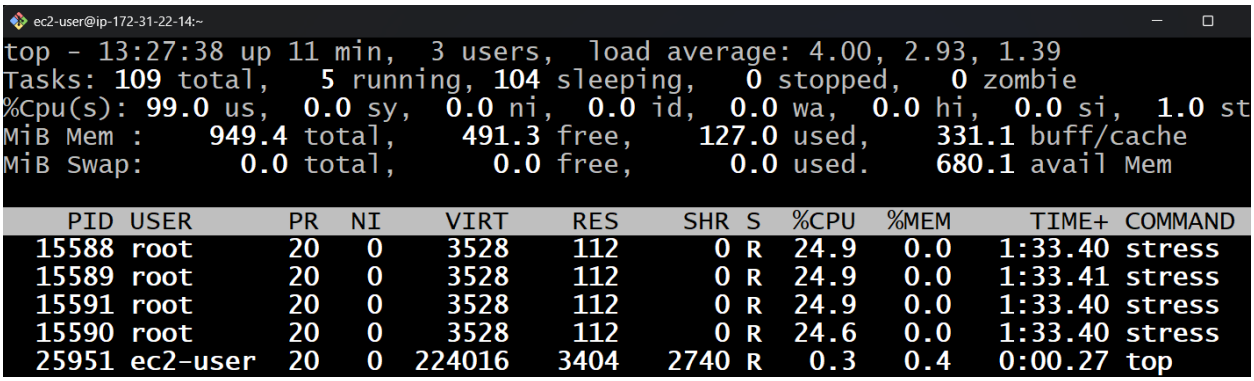
----------done----------

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

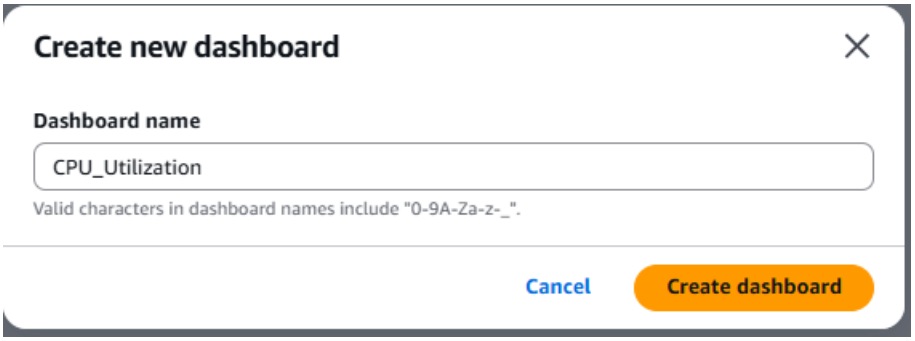
Create an EC2 instance and SSH into it and add some test load:

```
[ec2-user@ip-172-31-22-14 ~]$ sudo stress -c 4
stress: info: [15587] dispatching hogs: 4 cpu, 0 io, 0 vm, 0 hdd
```

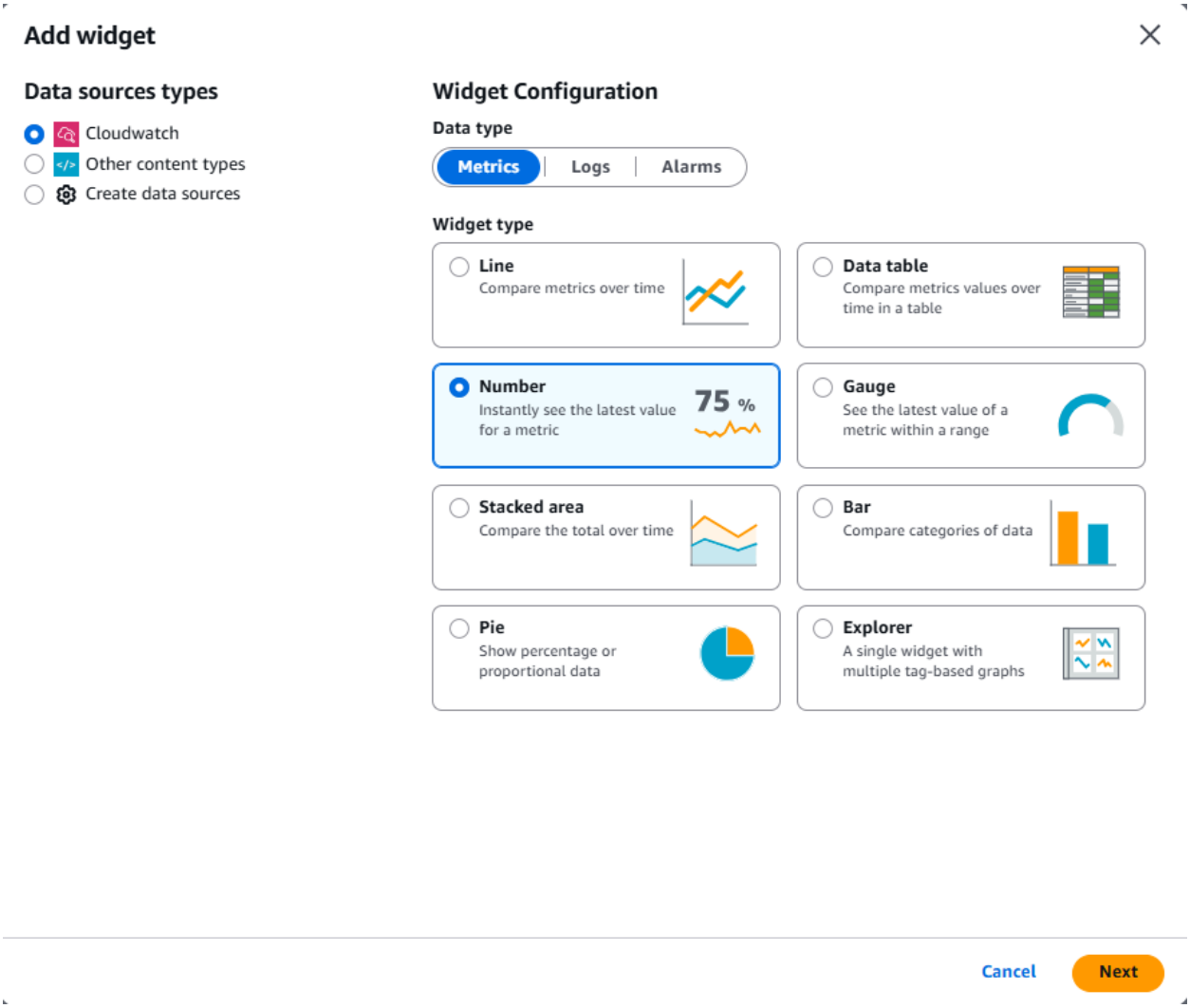
We can check the cpu usage:



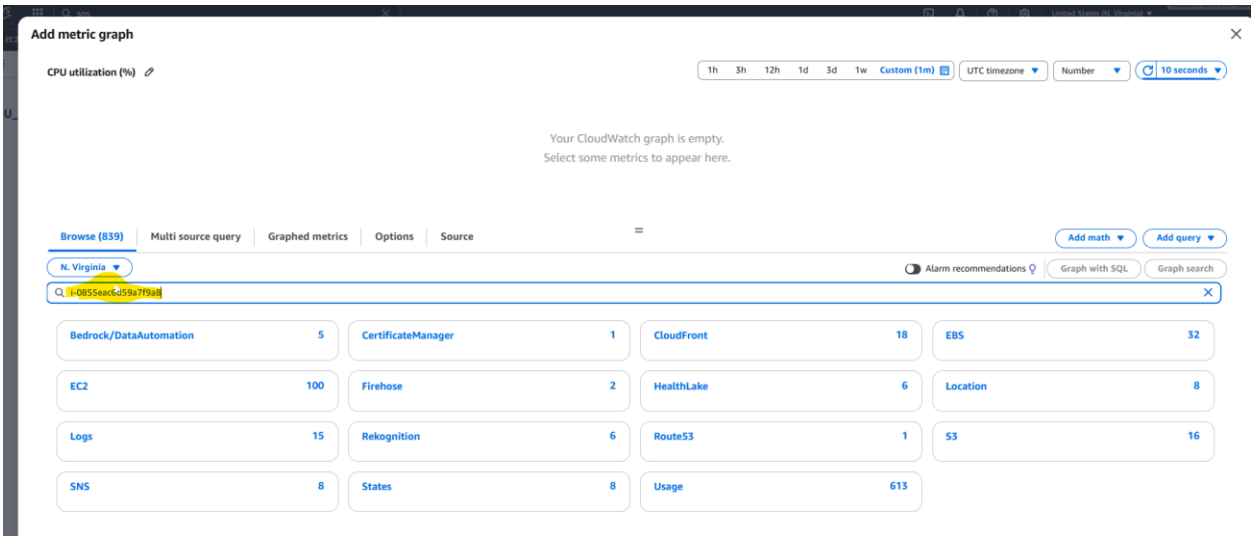
Goto CloudWatch and create Dashboard and give the name:



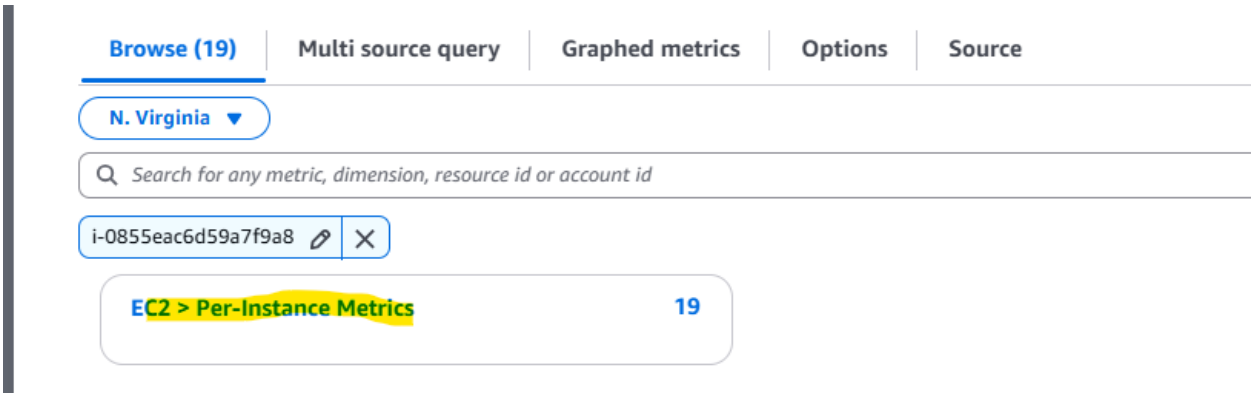
Select Number in widget:



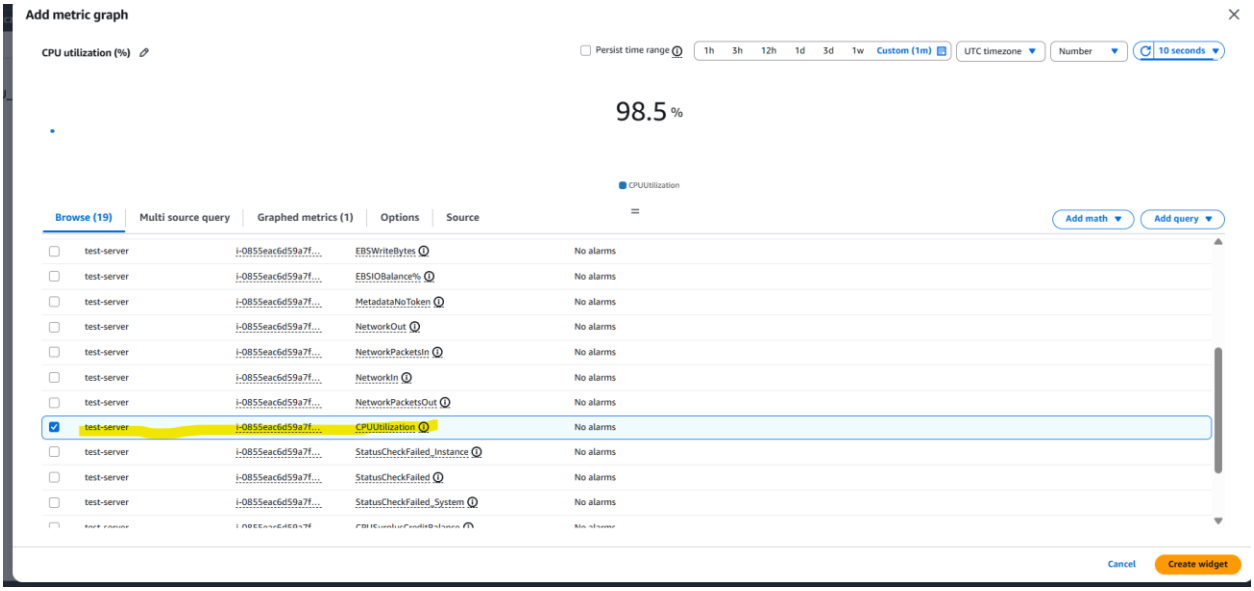
Give instance id and press Enter:



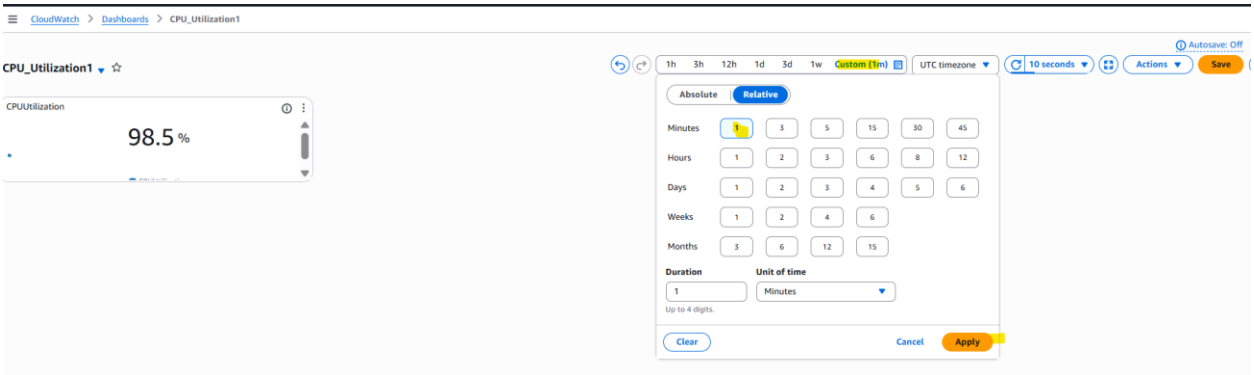
And click on EC2->Pre-Instance Metrics



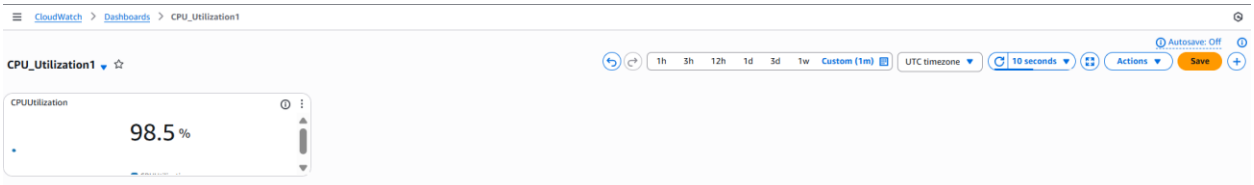
Select yours and click on create widget:



Select custom 1 min and apply :



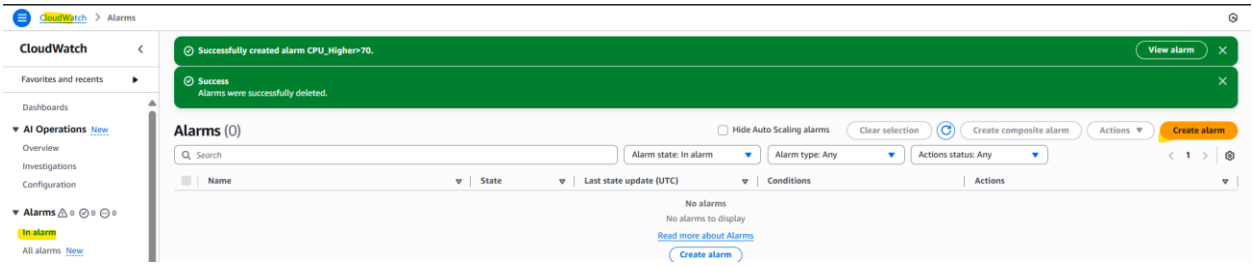
We can see the recorded CPU utilization of our instance:



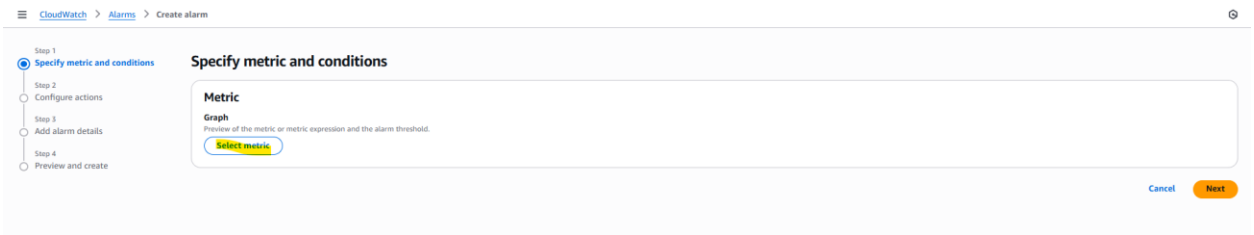
---------done----------

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

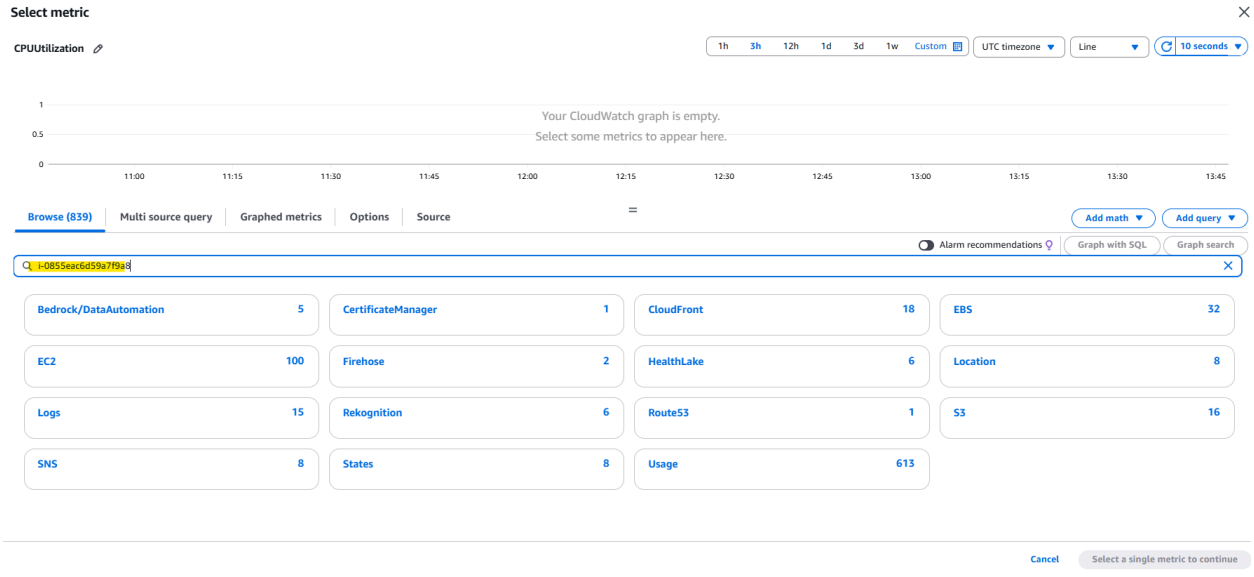
Goto CloudWatch service and select ‘In alarm’ and click on Create alarm:



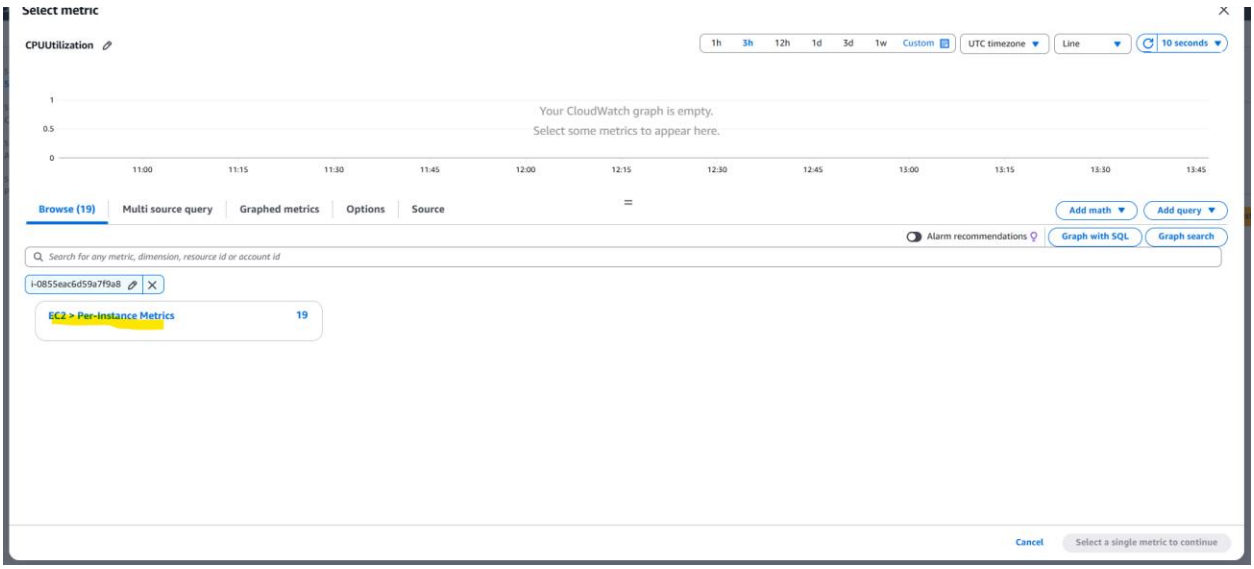
Then click on Select metric:



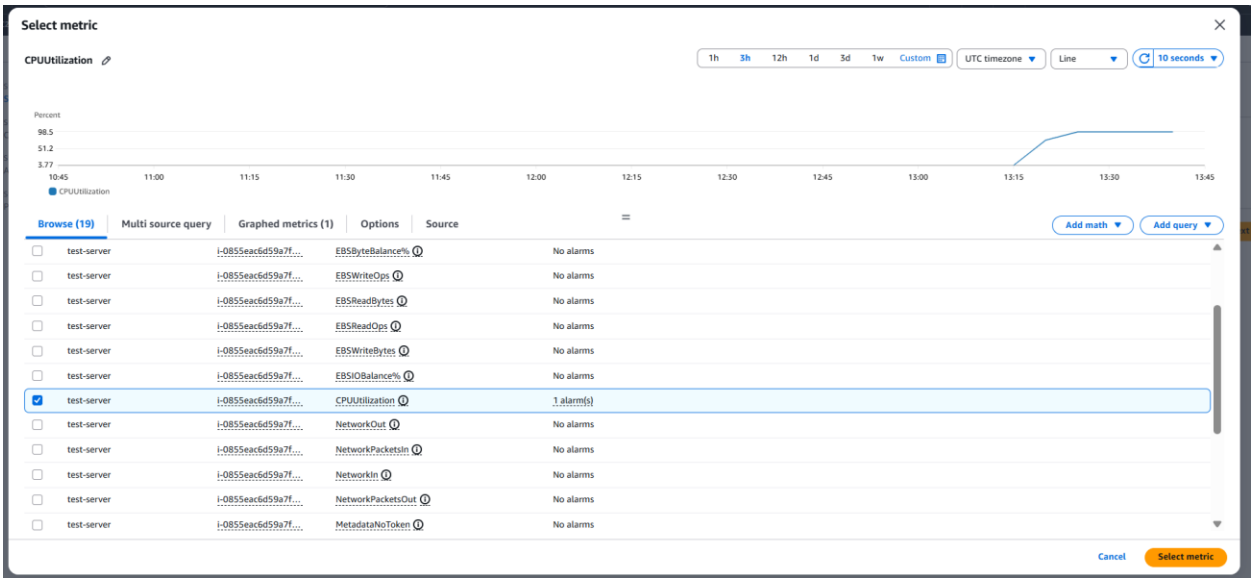
Give your instance id and press enter:



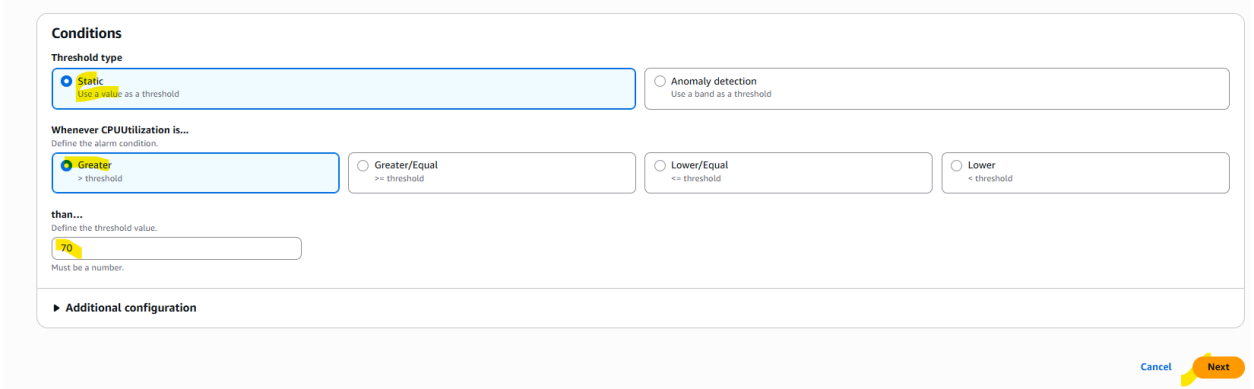
And click on Ec2->Pre-instance-metrics:



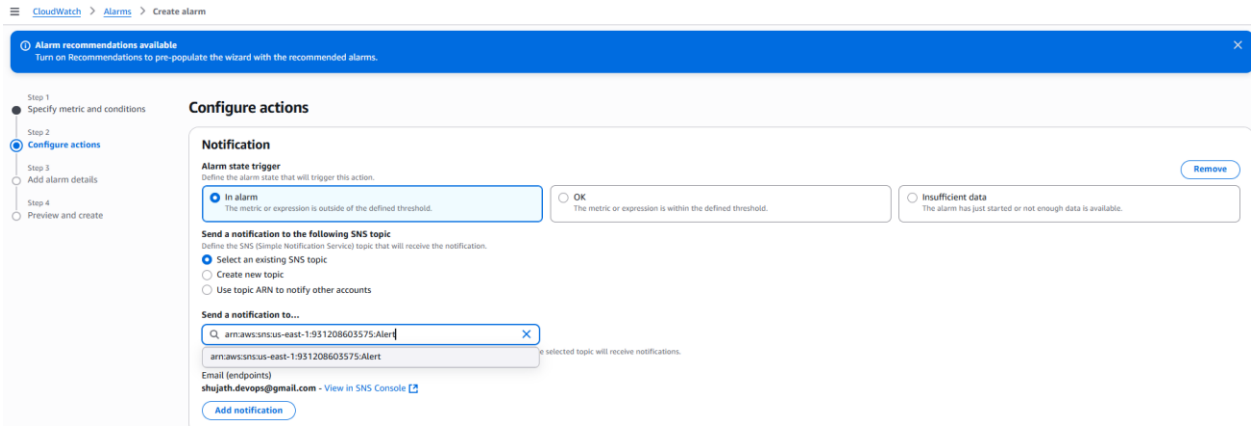
Select your metric and click on ‘select metric’:



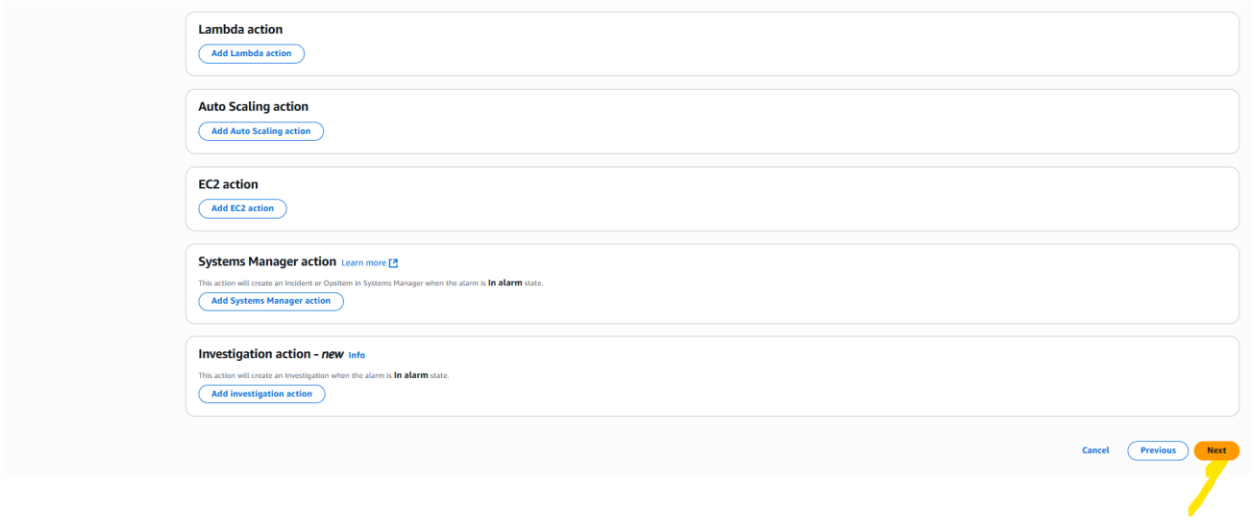
Give your value here with condition:



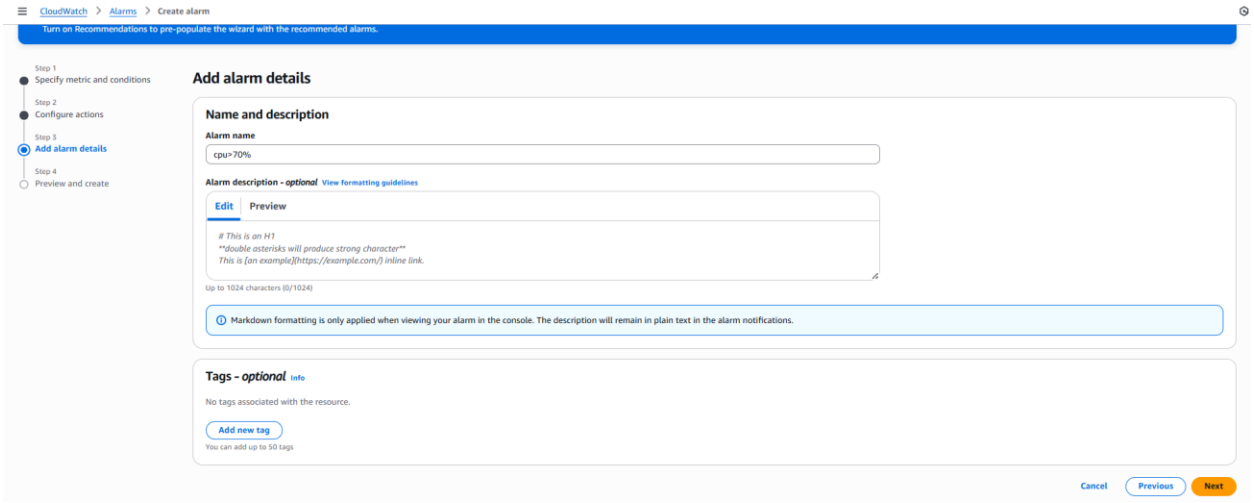
Select your Notification type ‘here select In alarm’, and select your SNS topics[SNS topic is already created, if not then create topic and subscription]:



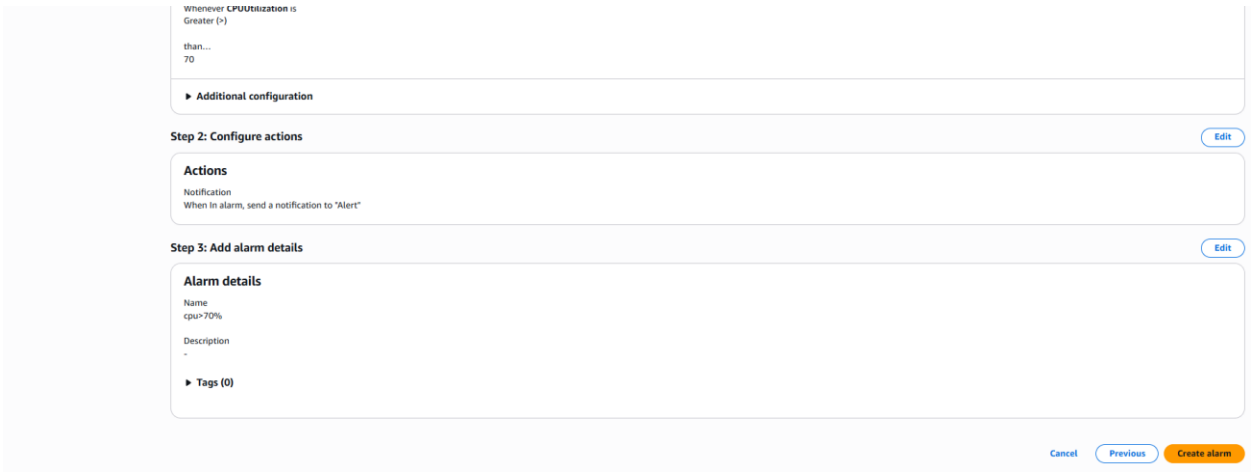
and click on Next:



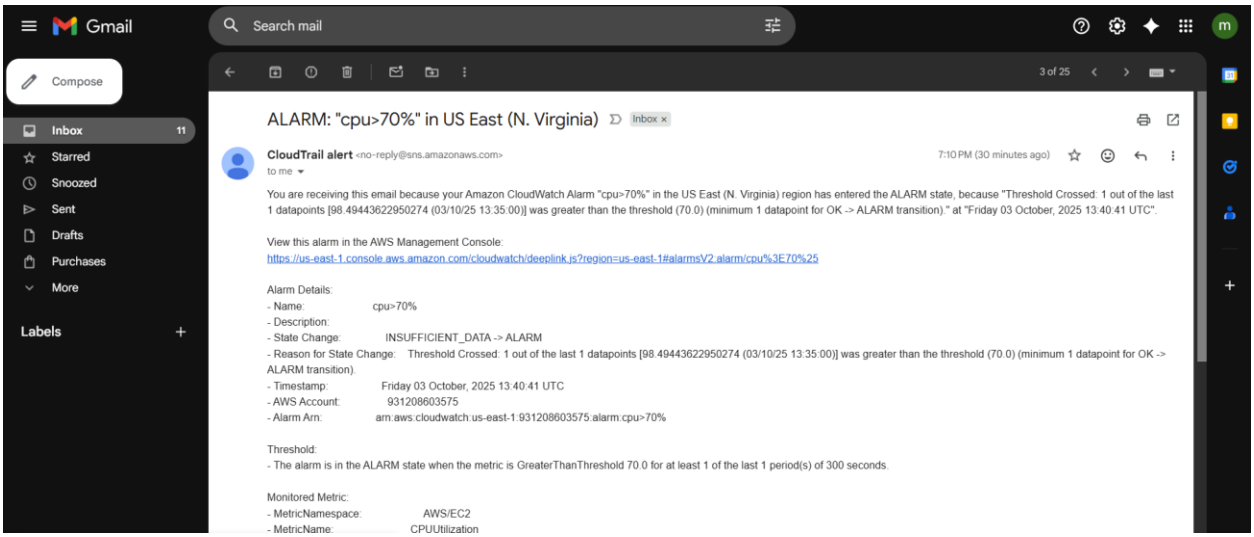
And give the Alarm name:



Preview and create:



And goto email and check the alert:



----------done----------

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

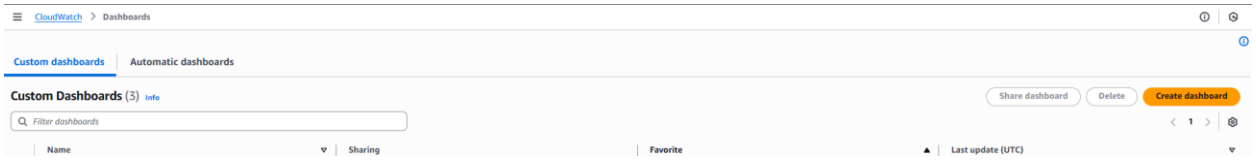
Install tomcat and all its dependencies.

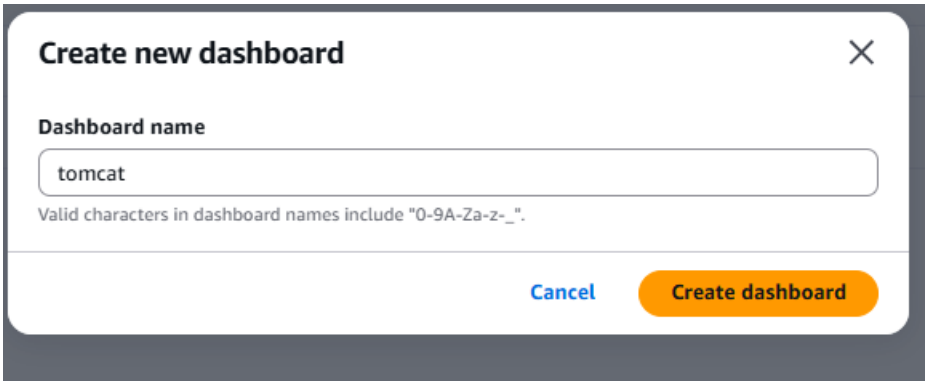
Create a script to check the status of Tomcat:

```
[root@ip-172-31-30-63 ~]# cat monitor.sh
#!/bin/bash
# Check if Tomcat service is active
if systemctl is-active --quiet tomcat; then
  STATUS=1
else
  STATUS=0
fi
# Push custom metric to CloudWatch
aws cloudwatch put-metric-data \
  --namespace "TomcatMonitoring" \
  --metric-name "TomcatStatus" \
  --value $STATUS \
  --region us-east-1
```

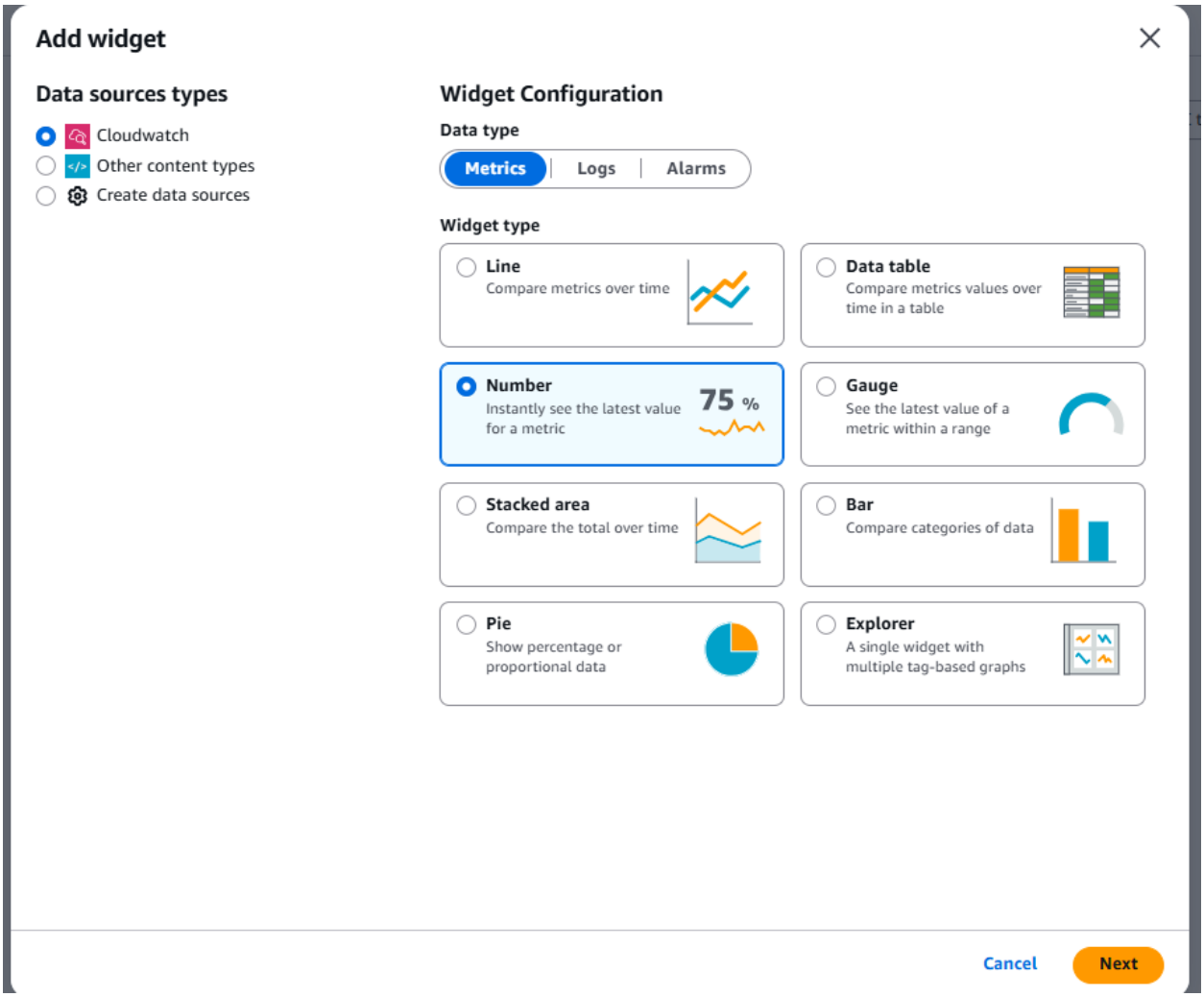
Run the script

And create the dashboard in CloudWatch:

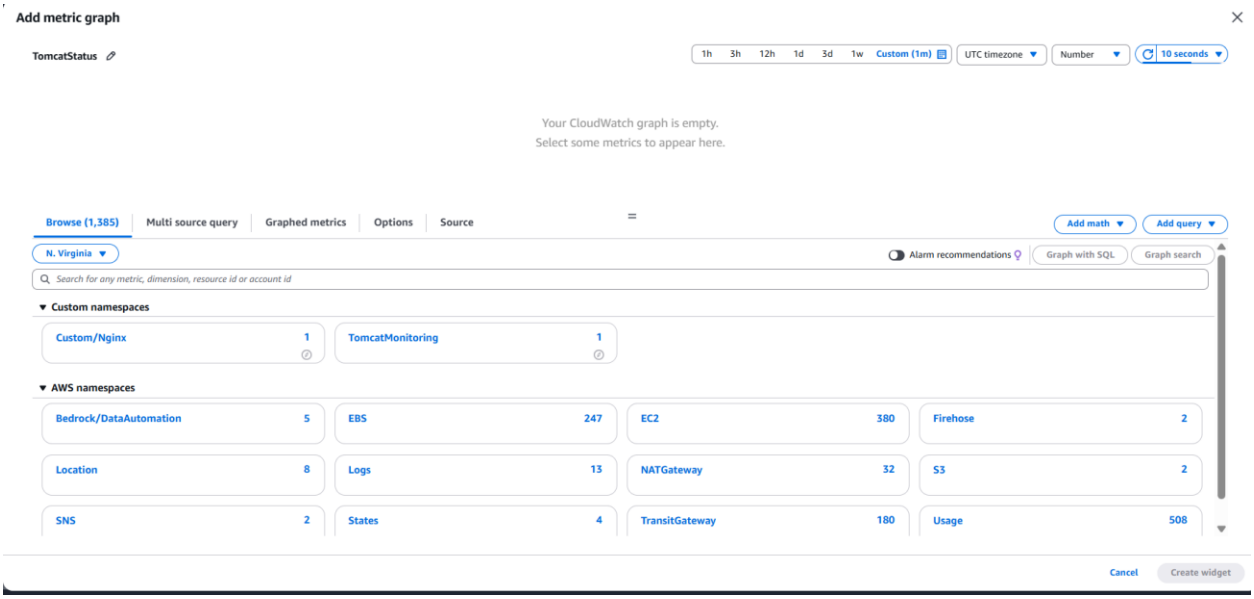


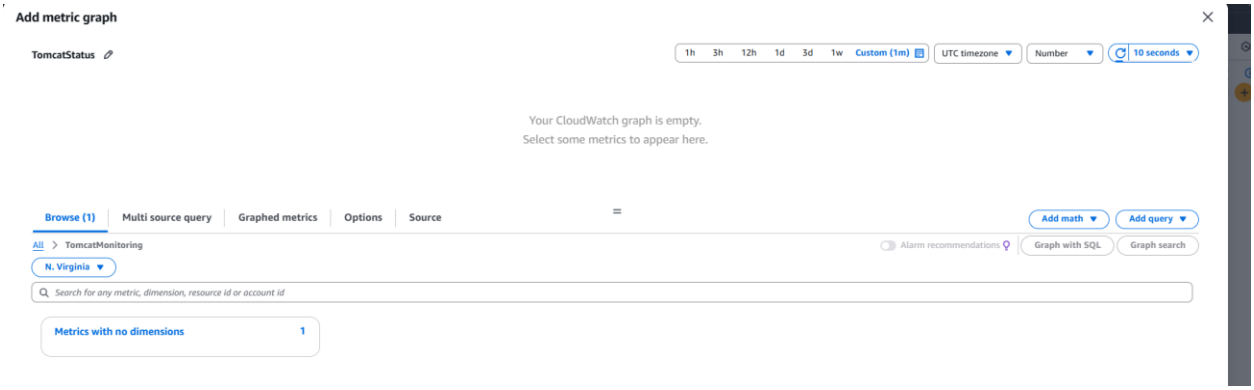


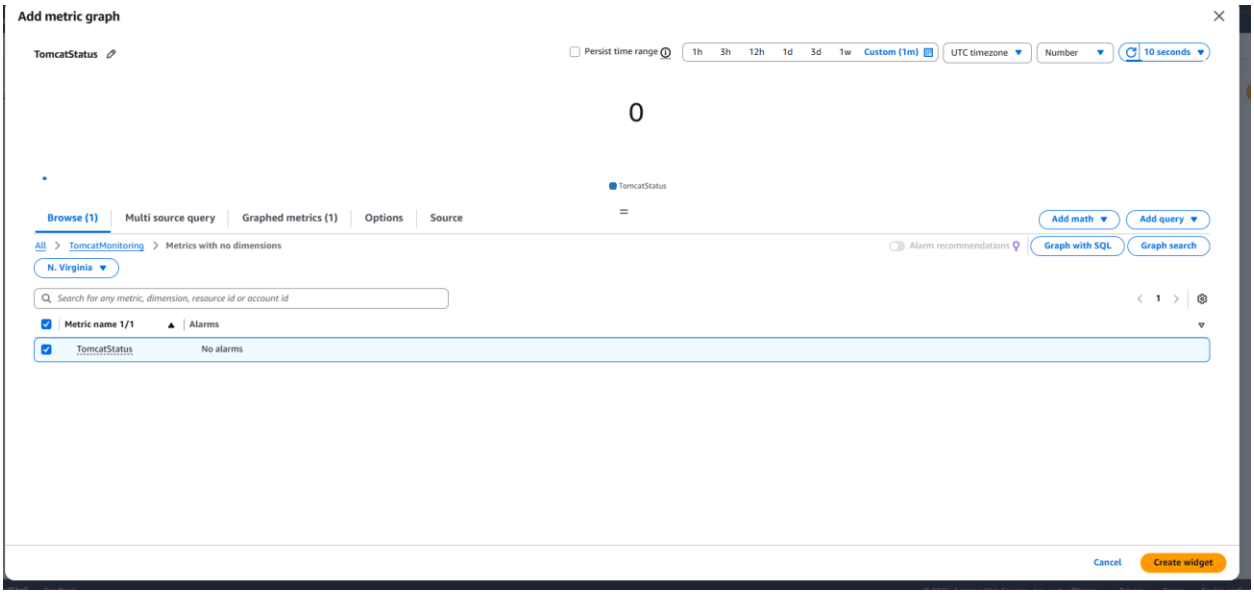
Select your required widget:



Select here the custom namespace i.e., tomcat:







---

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

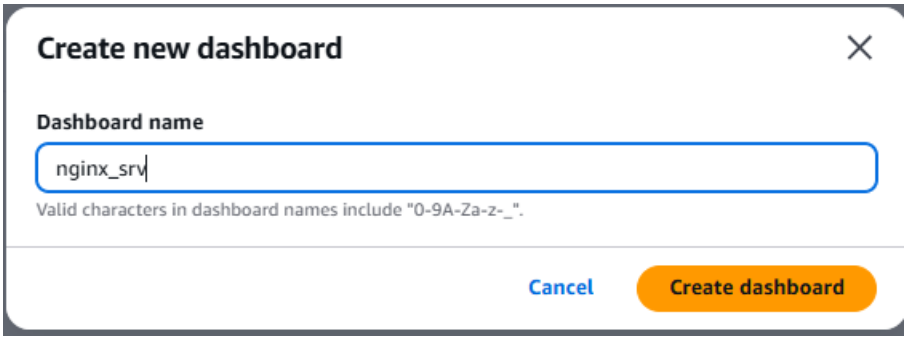
Write a script to send the nginx status to cloudwatch:

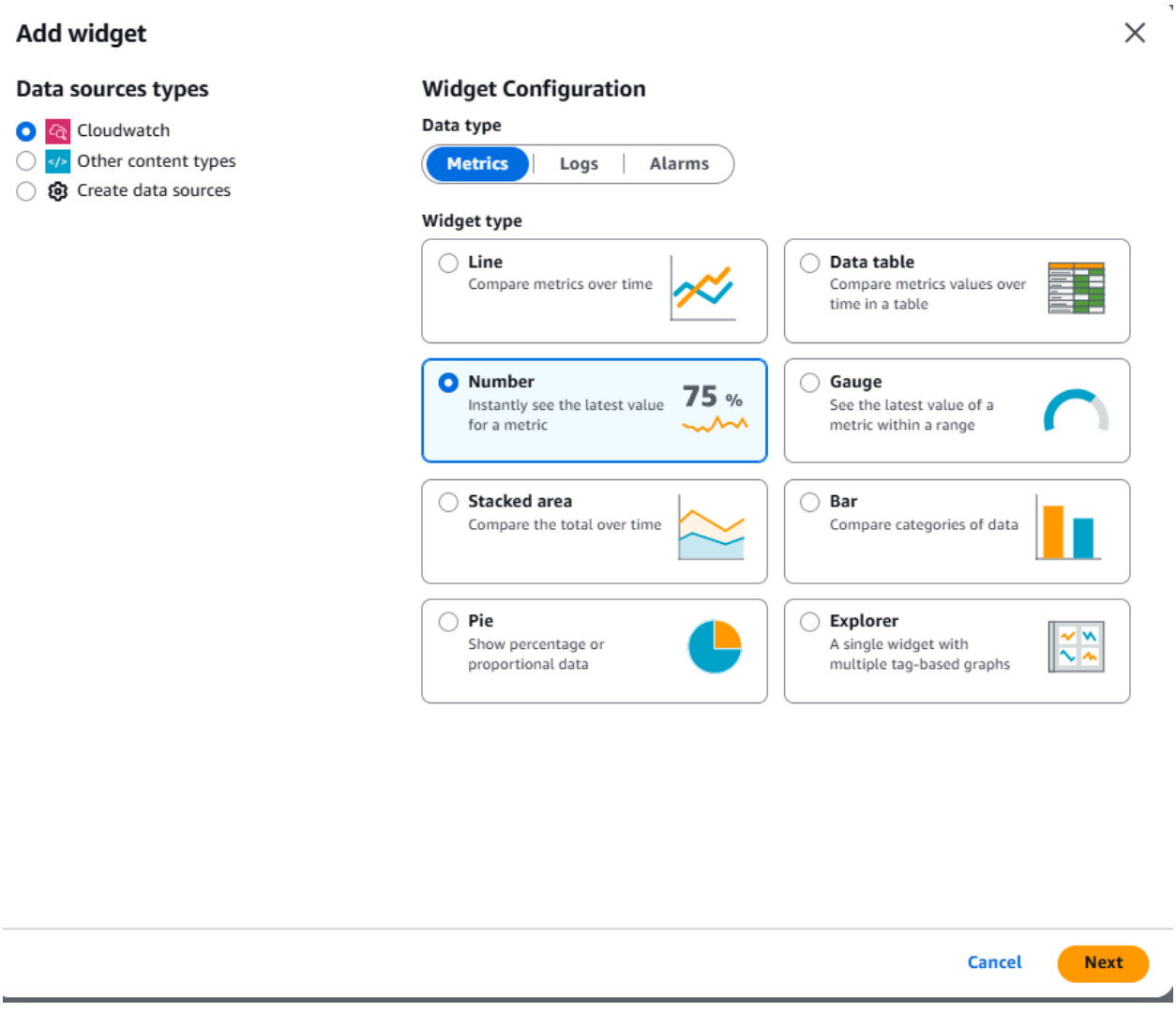
```bash
#!/bin/bash

# Set region (update this)
AWS_REGION="us-east-1"

# Check if Nginx is running
if systemctl is-active --quiet nginx; then
    nginx_status=1
else
    nginx_status=0
fi

# Push custom metric to CloudWatch
/usr/bin/aws cloudwatch put-metric-data \
  --namespace "Custom/Nginx" \
  --metric-name "NginxRunning" \
  --dimensions "Service=Nginx" \
  --value "$nginx_status" \
  --unit Count \
  --region "$AWS_REGION"
```

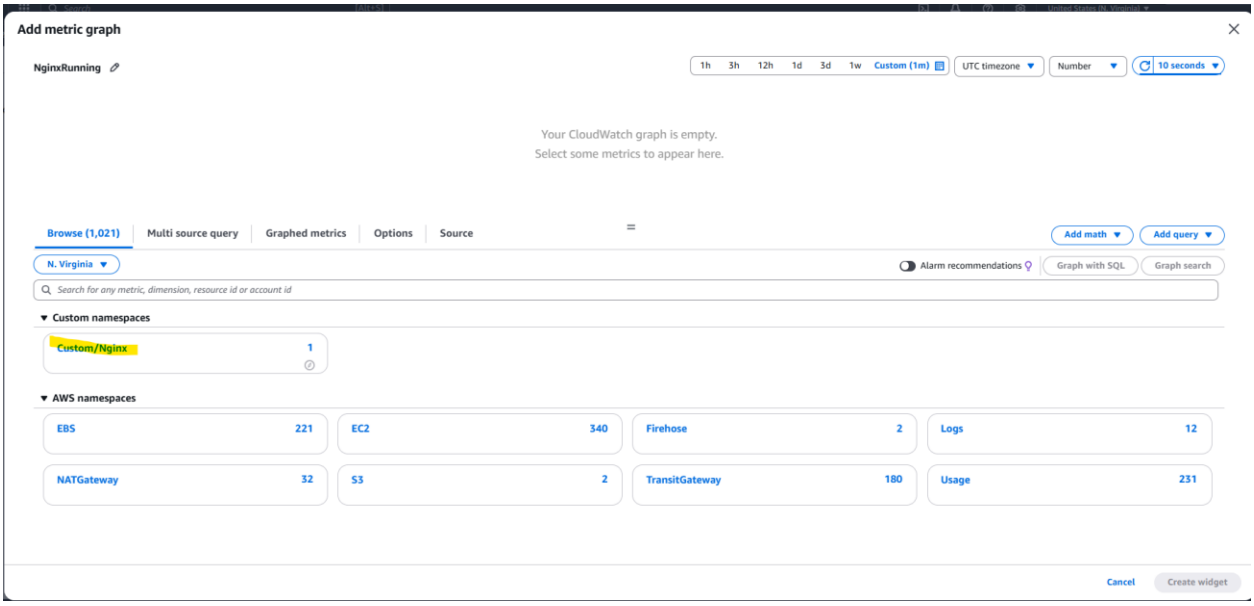
And create a cronjob to send metric to every minute:

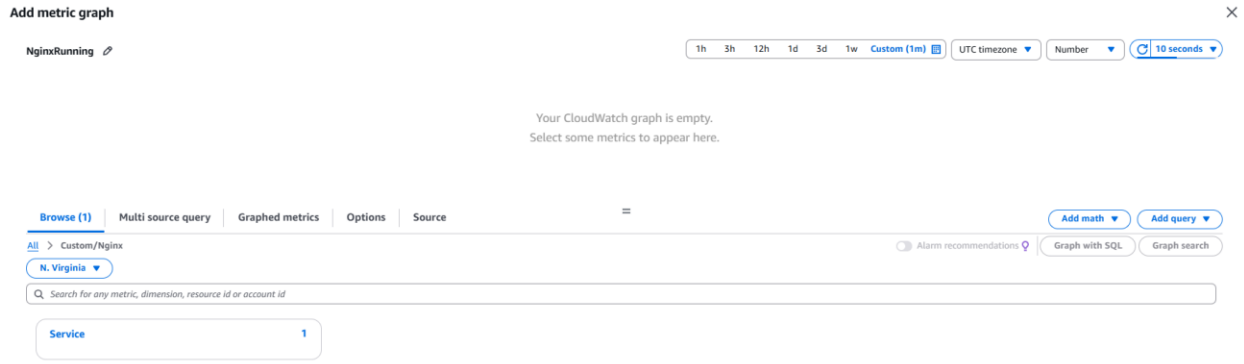
```
[root@ip-172-31-27-88 ~]# crontab -l
* * * * * /root/nginx.sh > /root/cron.log
```

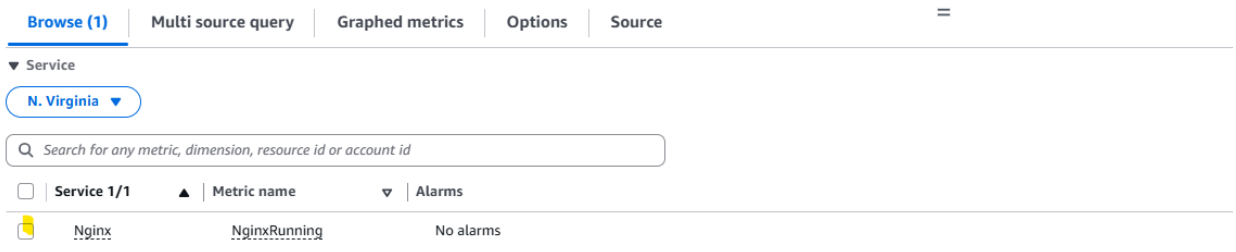
Create Dashboard in CloudWatch :

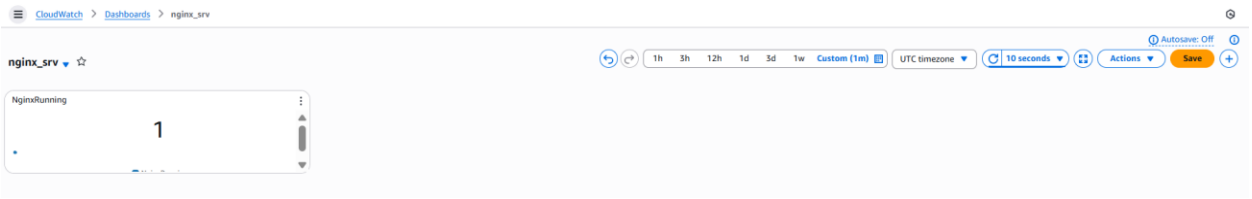


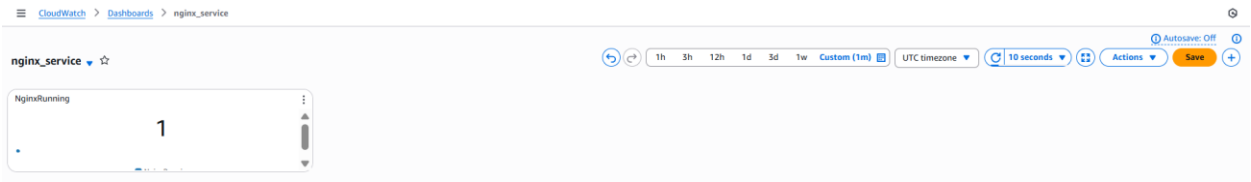

Add widget
Data sources types
Cloudwatch
Other content types
Create data sources
Widget Configuration
Data type
Metrics
Logs
Alarms
Widget type
Line
Compare metrics over time
Data table
Compare metrics values over time in a table
Number
Instantly see the latest value for a metric
75 %
Gauge
See the latest value of a metric within a range
Stacked area
Compare the total over time
Bar
Compare categories of data
Pie
Show percentage or proportional data
Explorer
A single widget with multiple tag-based graphs
Cancel
Next


Add metric graph
NginxRunning
1h
3h
12h
1d
3d
1w
Custom (1m)
UTC timezone
Number
10 seconds
Your CloudWatch graph is empty.
Select some metrics to appear here.
Browse (1,021)
Multi source query
Graphed metrics
Options
Source
Add math
Add query
N. Virginia
Alarm recommendations
Graph with SQL
Graph search
Search for any metric, dimension, resource id or account id
Custom namespaces
Custom/Nginx
1
AWS namespaces
EBS
221
EC2
340
Firehose
2
Logs
12
NATGateway
32
S3
2
TransitGateway
180
Usage
231
Cancel
Create widget


Add metric graph
NginxRunning
1h 3h 12h 1d 3d 1w Custom (1m)
UTC timezone
Number
10 seconds
Your CloudWatch graph is empty.
Select some metrics to appear here.
Browse (1)
Multi source query
Graphed metrics
Options
Source
Add math
Add query
All > Custom/Nginx
Alarm recommendations
Graph with SQL
Graph search
N. Virginia
Search for any metric, dimension, resource id or account id
Service 1


Browse (1)
Multi source query
Graphed metrics
Options
Source
Service
N. Virginia
Search for any metric, dimension, resource id or account id
Service 1/1
Metric name
Alarms
Nginx
NginxRunning
No alarms


CloudWatch > Dashboards > nginx_srv
nginx_srv
Autosave: Off
1h 3h 12h 1d 3d 1w Custom (1m)
UTC timezone
10 seconds
Actions
Save
NginxRunning
1


CloudWatch > Dashboards > nginx_service
nginx_service
Autosave: Off
1h 3h 12h 1d 3d 1w Custom (1m)
UTC timezone
10 seconds
Actions
Save
NginxRunning
1

----------completed----------