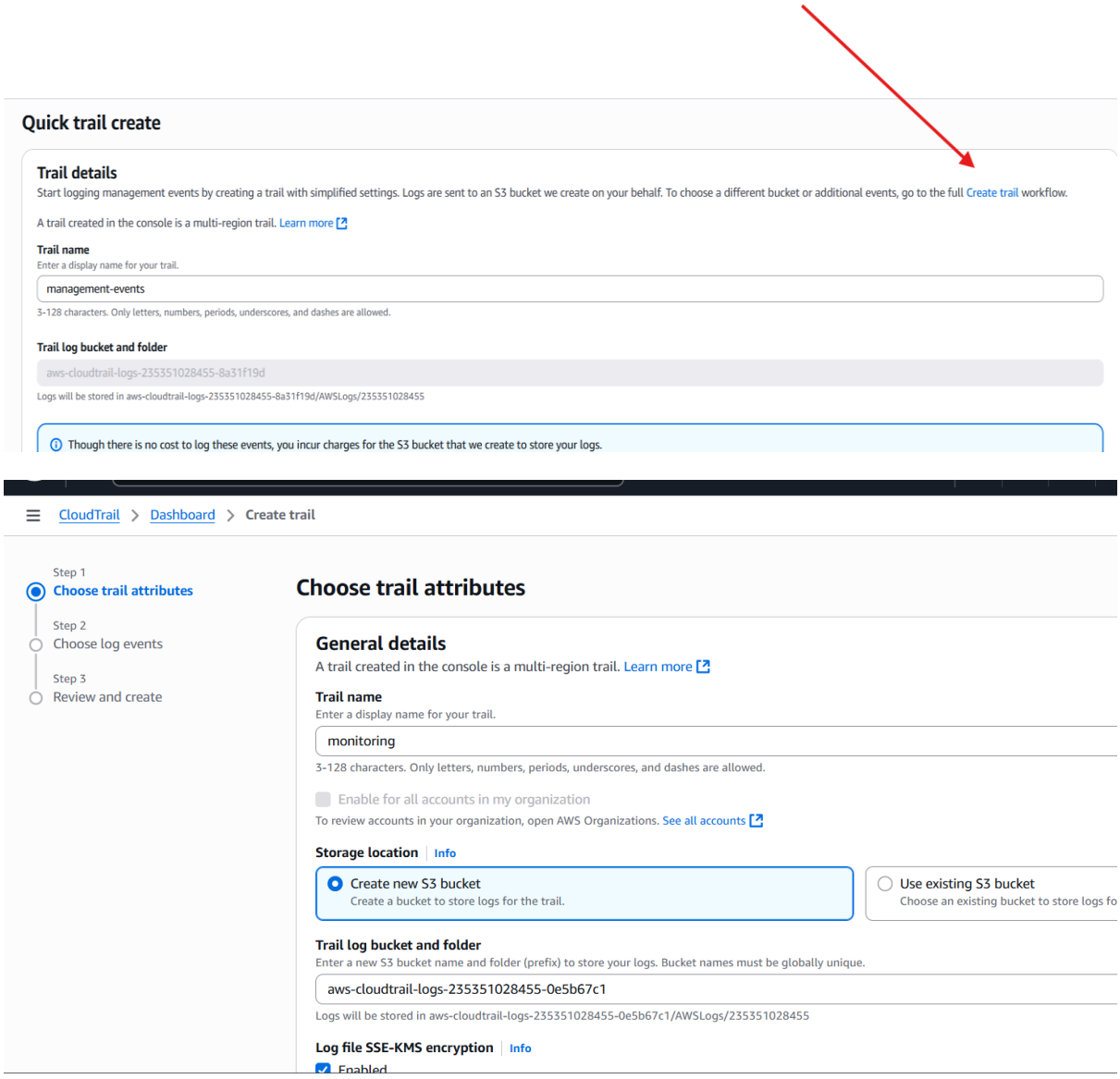
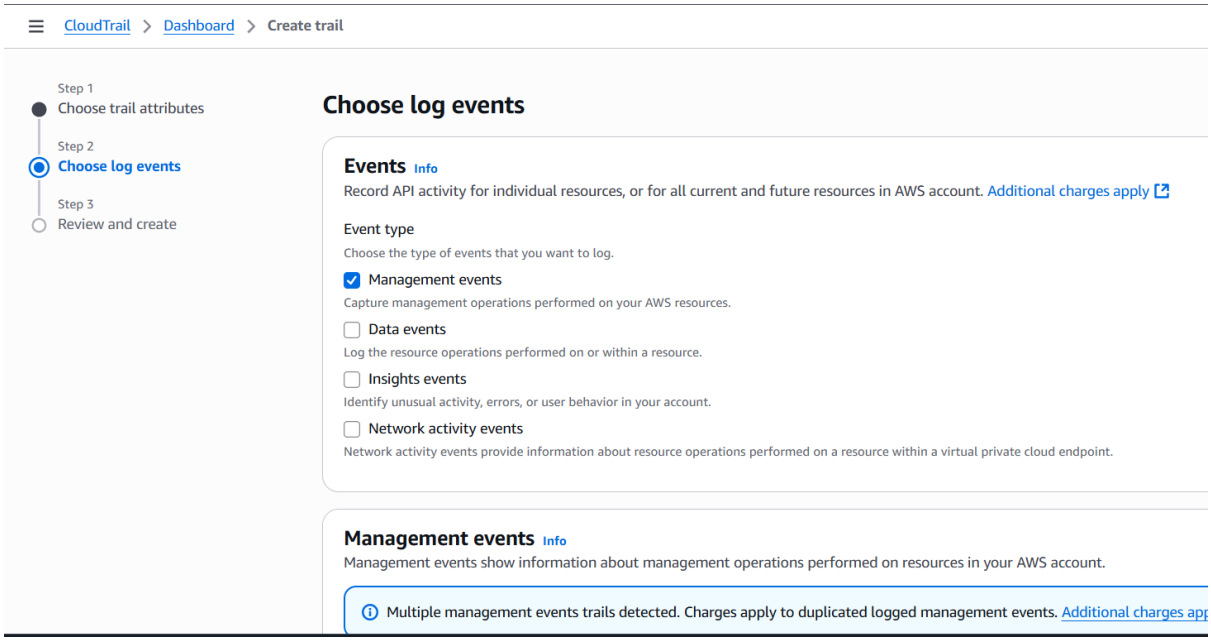
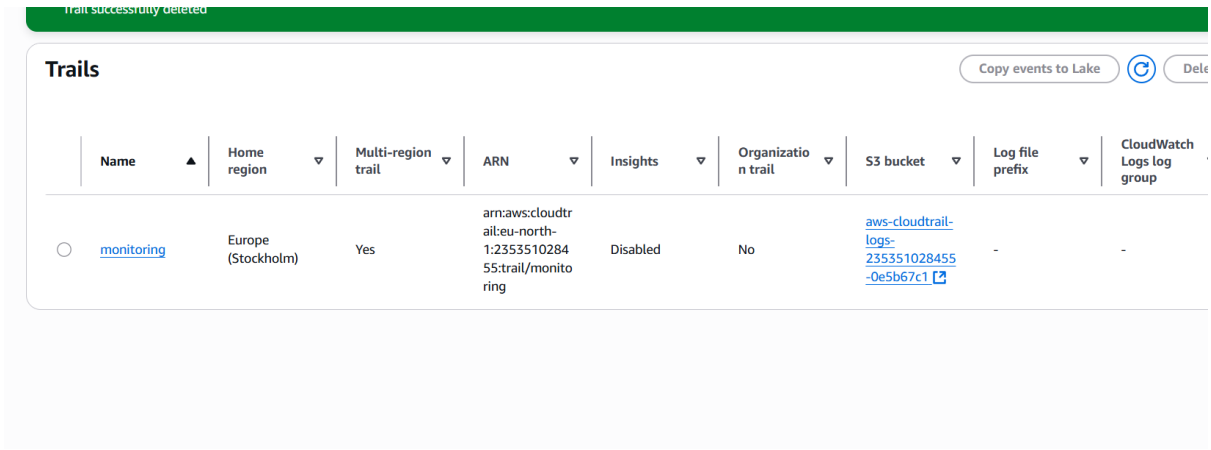
1.Enable cloudtrail monitoring and store the events in s3 and cloudwatch log events.

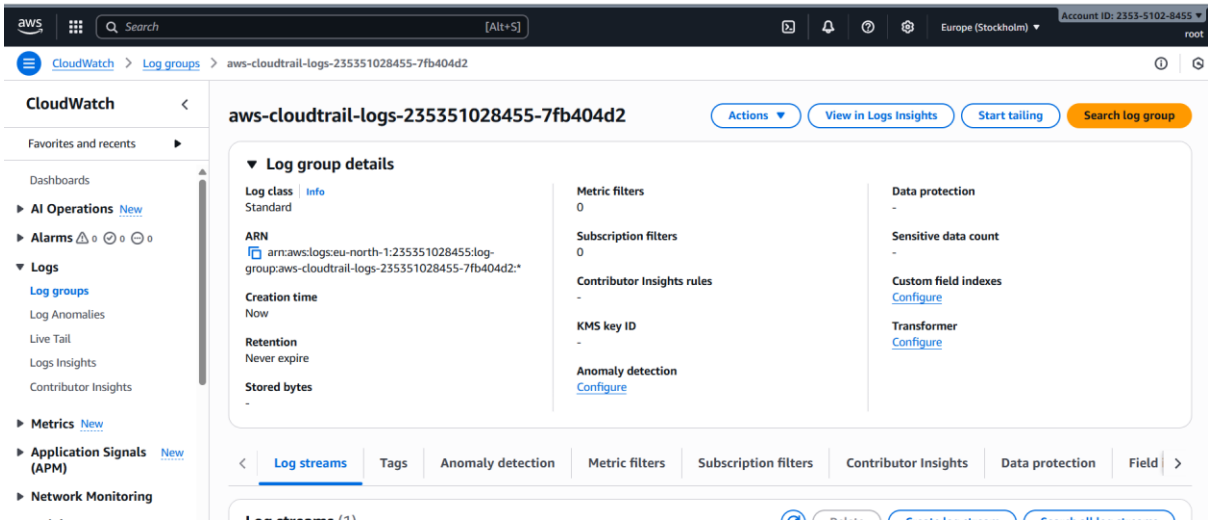
Go to cloud trail and create trail and create trail workflow.





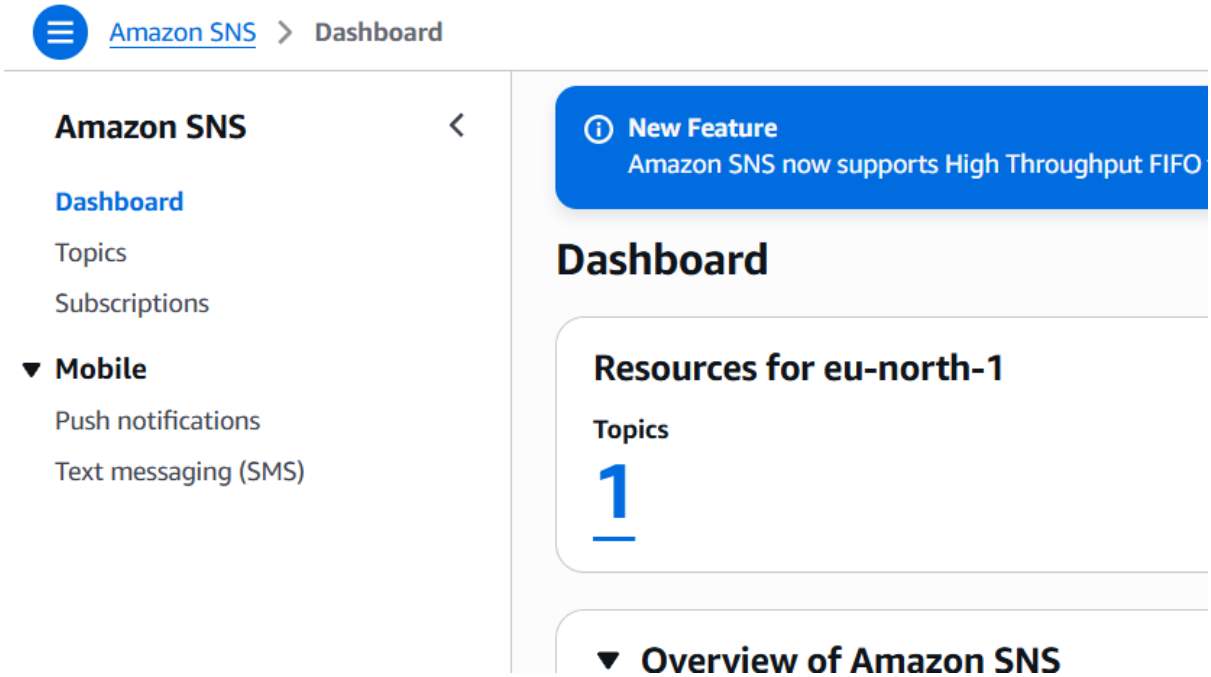


Go to cloudwatch and go to log groups. We can see a log group has been created.

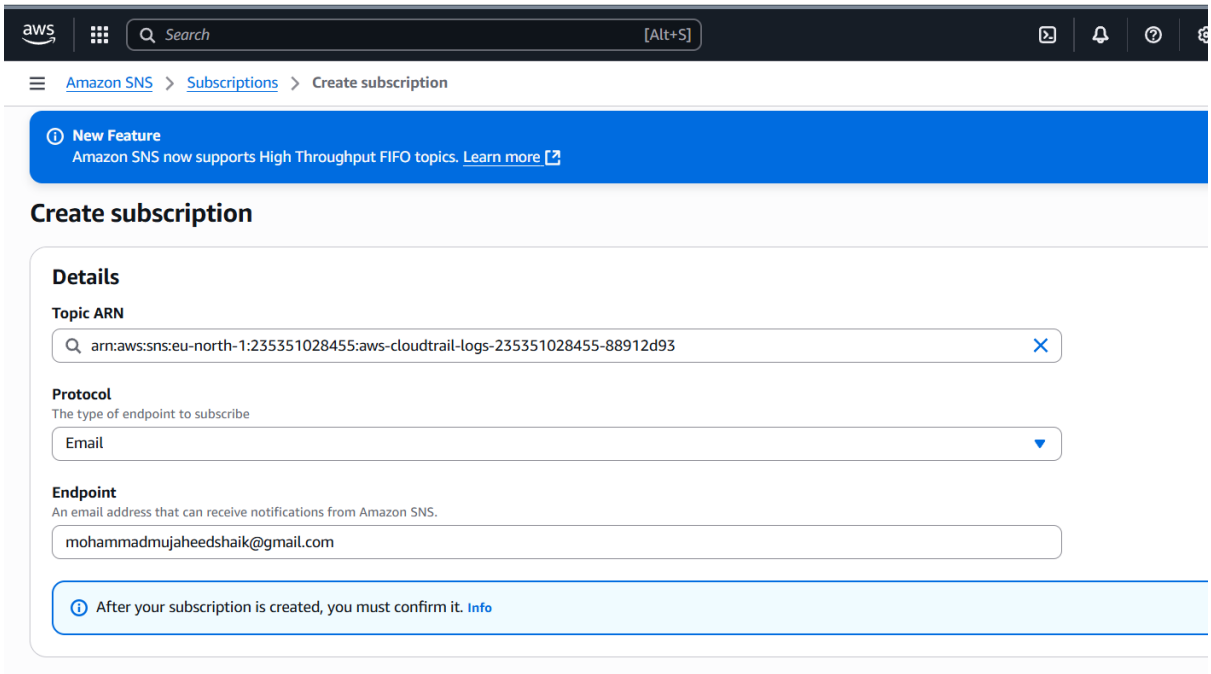


2. Enable SNS for cloudtrial to send alert on email.

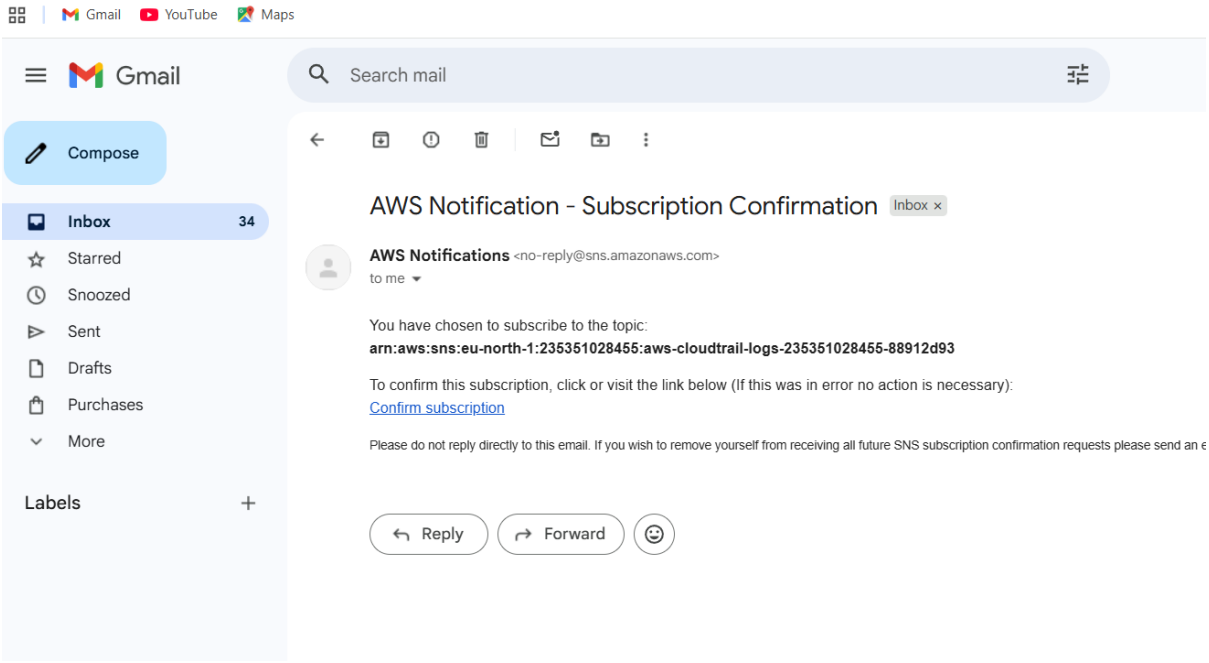
Go to sns

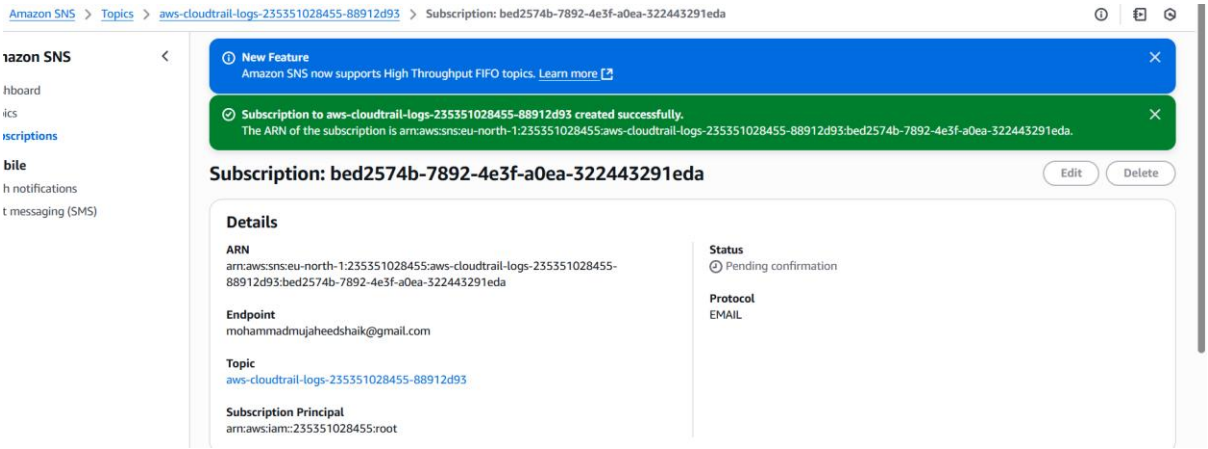


Create a subscription and create subscription by giving the created cloudtrailevent and select email and provide the email.



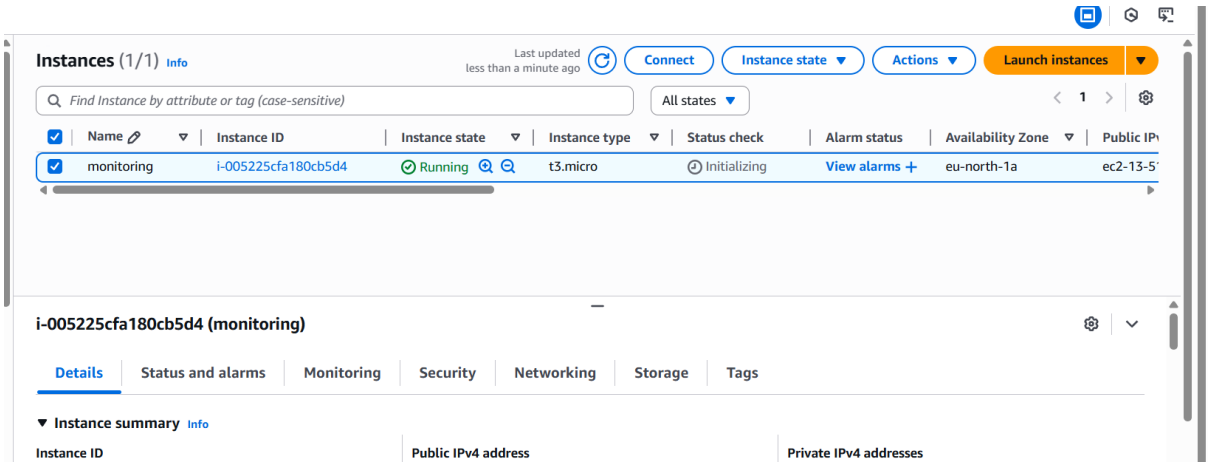
Go to your email and accept the request.



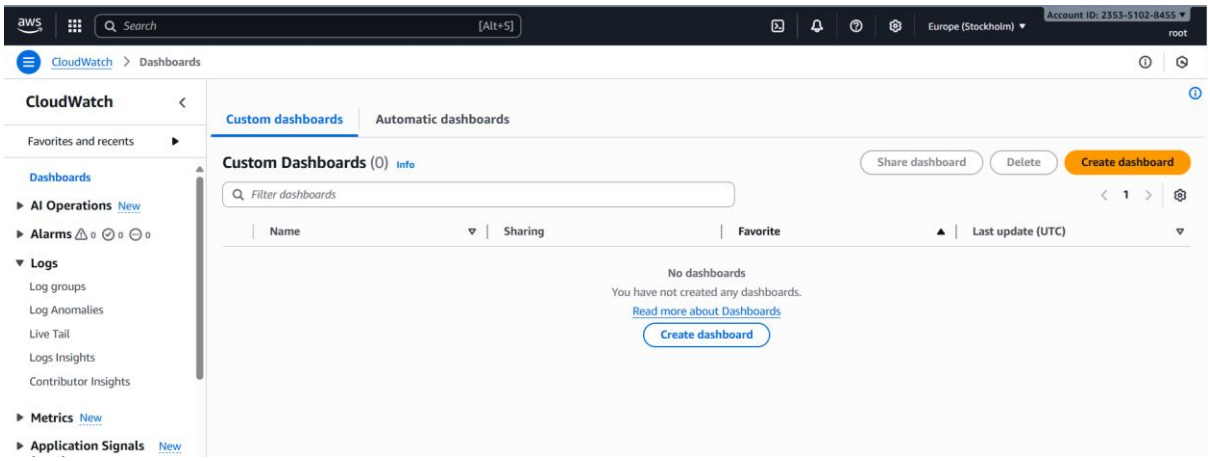


3. Configure cloud watch monitoring and record the cpu utilization and other metrics of ec2.

Launch one instance in ec2.



Go to cloudwatch and click on dashboards and click on create dashboard.



Go to metrics and add numeric type in that select ec2 and select cpu utilization and create widget.

Select the instance id and paste in the bar.

CPUUtilization

Persist time range

1h 3h 12h 1d 3d

100 %

CPUUtilization

Browse (20) | Multi source query | Graphed metrics (1) | Options | Source

Per-Instance Metrics

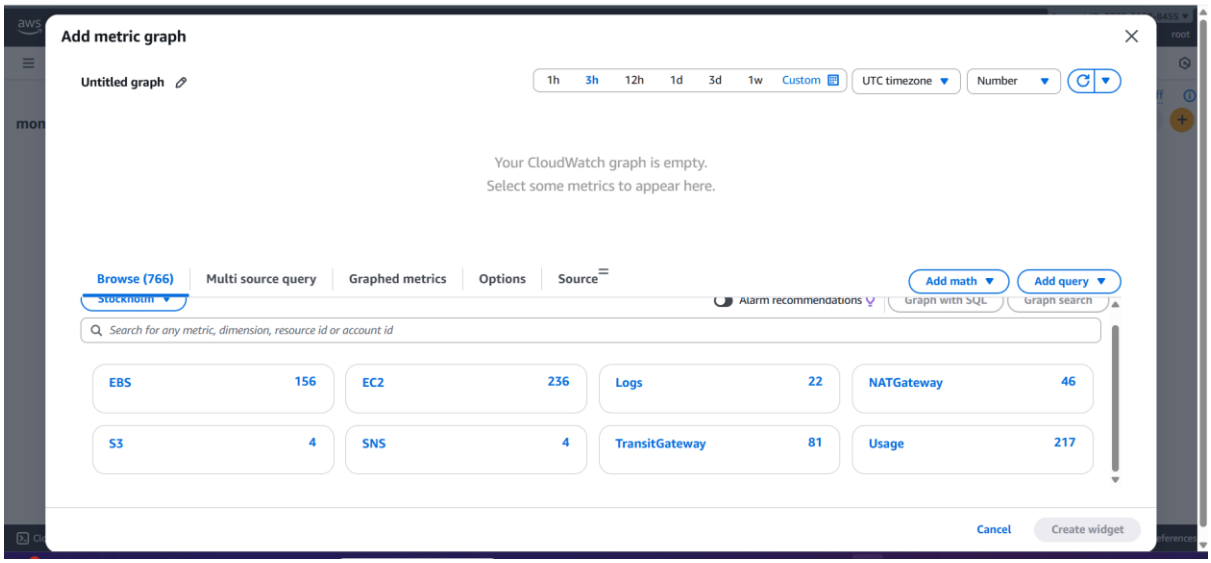
N. Virginia

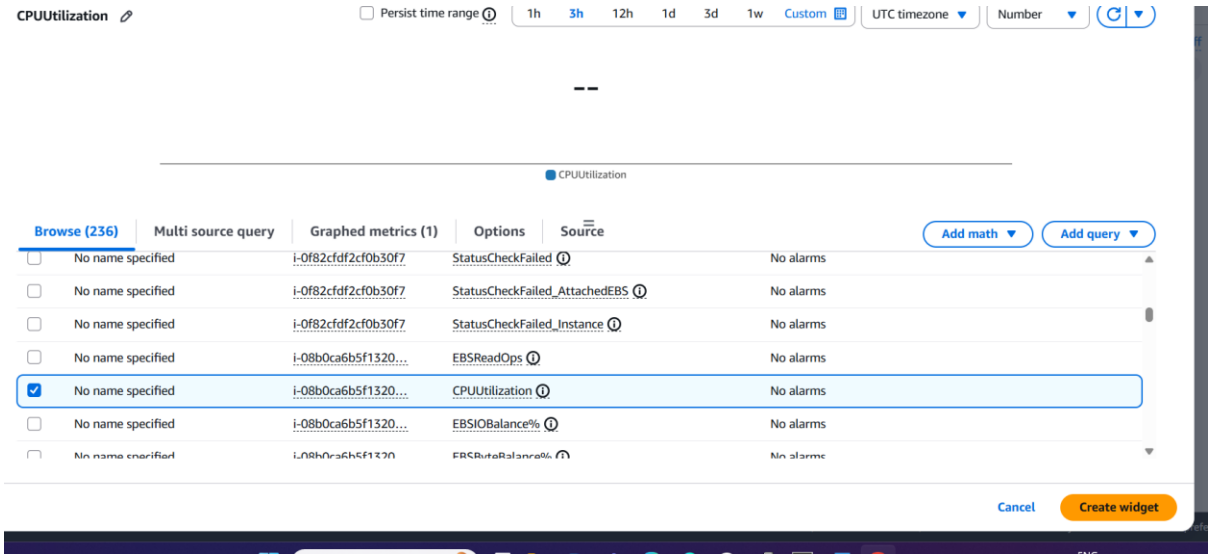
Search for any metric, dimension, resource id or account id

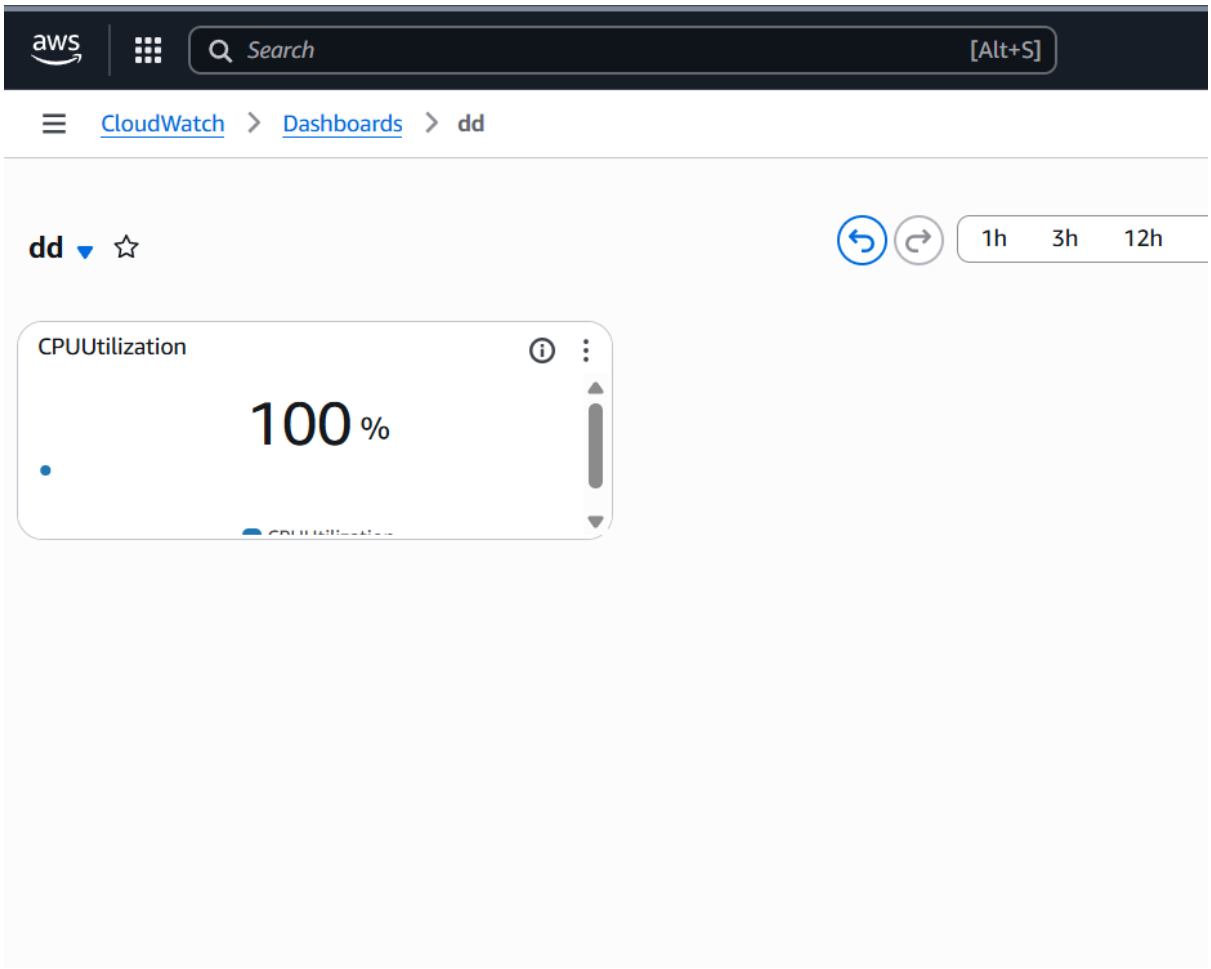
i-0c1d89b33cd512984

Clear filters

Instance name 20/20	InstanceId	Metric name
public	i-0c1d89b33cd51...	EBSReadBytes

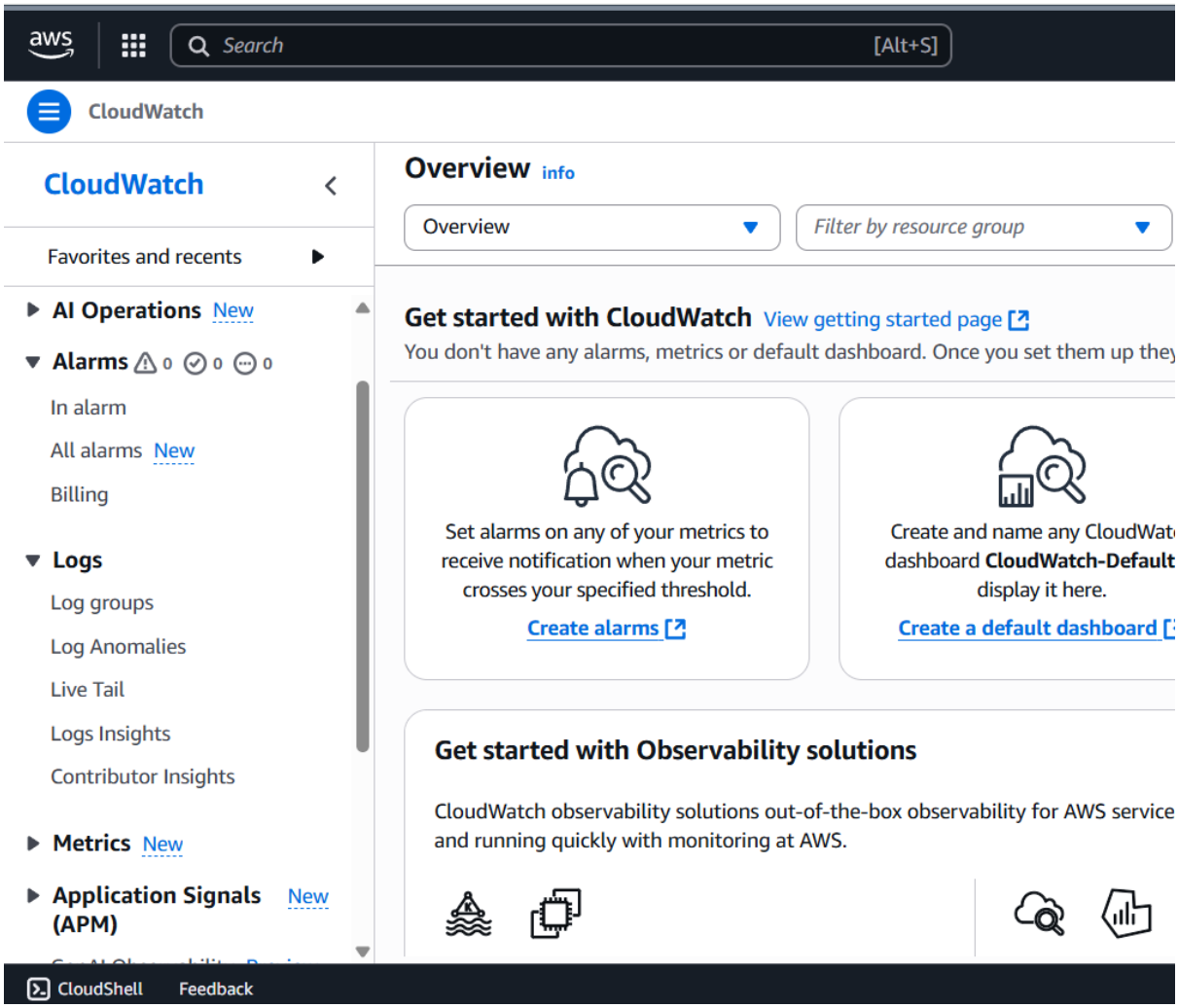




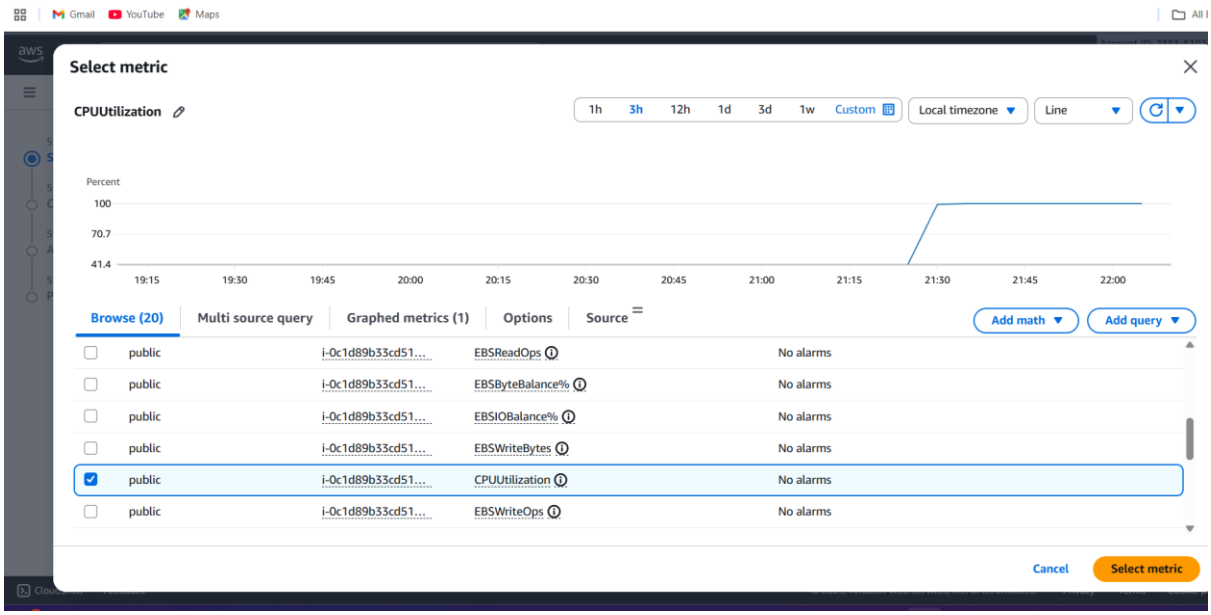


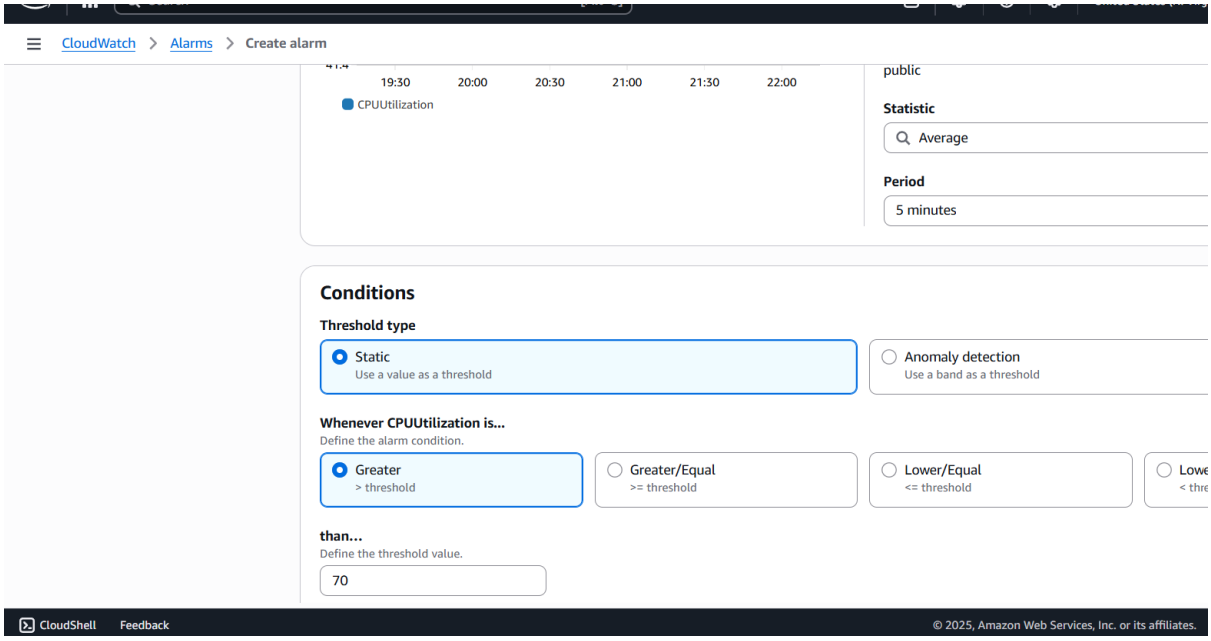
4.Create one alarm to send alert to email if the cpu utilization is more than 70 percent.

go to cloudwatch and create alarm

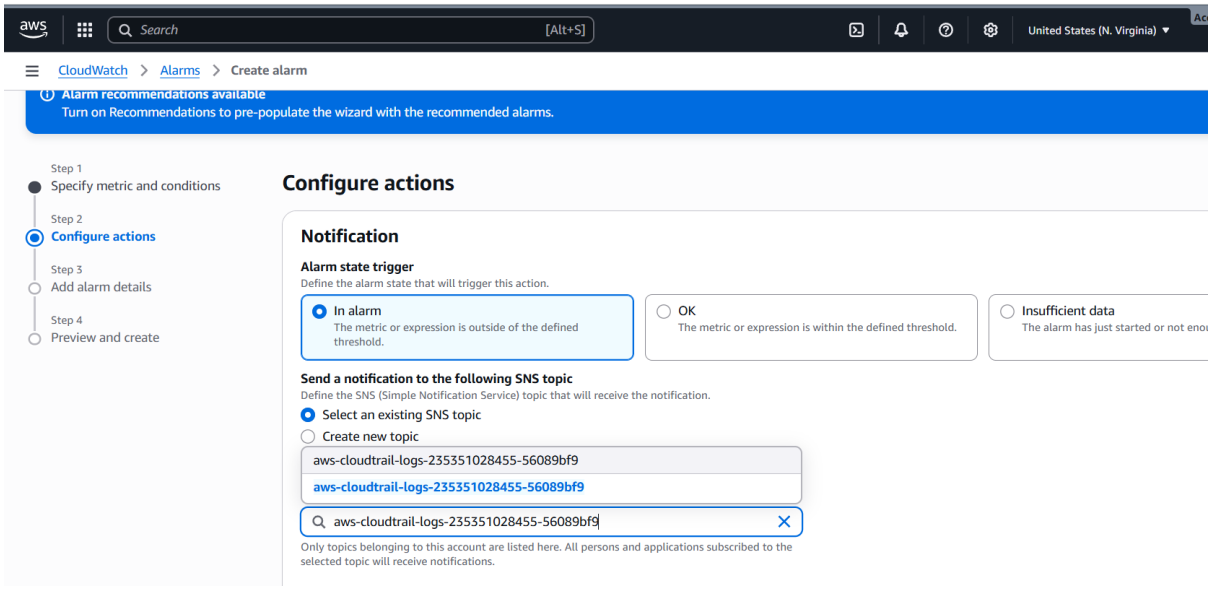


Create metrics and select your instance id and select cpu utilization.

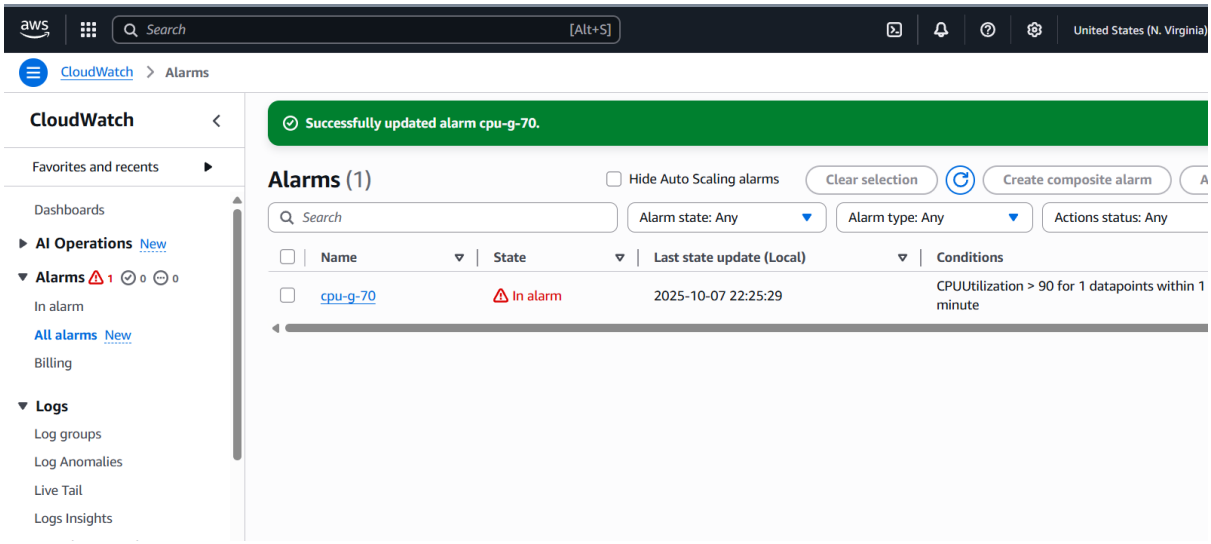


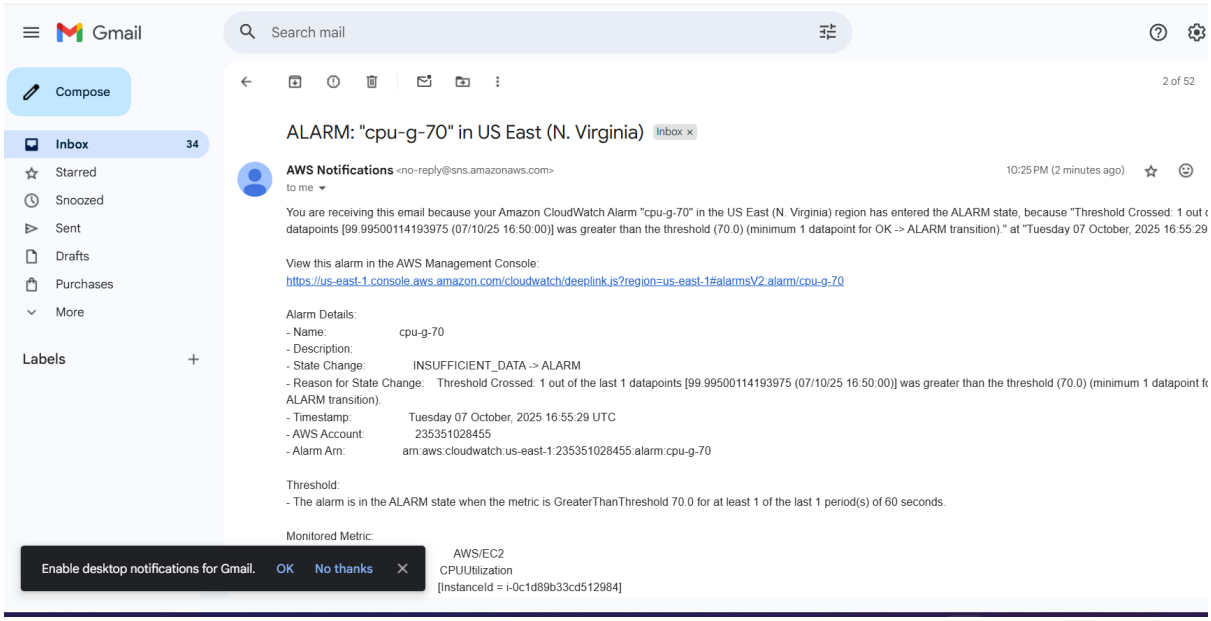


Select in alarm and select sns topic.



if cpu utilization is greater than treshhold value then a email alarm will notify.





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

Open created instance and install java.

Sudo yum install java-17 -y

```
[ec2-user@ip-10-0-4-253 ~]$ sudo yum install java-17 -y
Amazon Linux 2023 Kernel Livepatch repository
Dependencies resolved.
================================================================================
 Package                                    Architecture       Versi
================================================================================
Installing:
 java-17-amazon-corretto                    x86_64             1:17.
Installing dependencies:
 alsa-lib                                   x86_64             1.2.7
 cairo                                      x86_64             1.18.
 dejavu-sans-fonts                          noarch             2.37-
 dejavu-sans-mono-fonts                     noarch             2.37-
 dejavu-serif-fonts                         noarch             2.37-
 fontconfig                                 x86_64             2.13.
 fonts-filesystem                           noarch             1:2.0
 freetype                                   x86_64             2.13.
 giflib                                     x86_64             5.2.1
 google-noto-fonts-common                   noarch             20240
 google-noto-sans-vf-fonts                  noarch             20240
 graphite2                                  x86_64             1.3.1
 harfbuzz                                   x86_64             7.0.0
 java-17-amazon-corretto-headless           x86_64             1:17.
 javapackages-filesystem                    noarch             6.0.0
 langpacks-core-font-en                     noarch             3.0-2
 libICE                                     x86_64             1.1.1
 libSM                                      x86_64             1.2.4
 libX11                                     x86_64             1.8.1
```

Download apache tomcat tarfile by using wget

```
[ec2-user@ip-10-0-4-253 ~]$ wget https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.110/bin/apache-tomcat
--2025-10-08 10:16:42--  https://dlcdn.apache.org/tomcat/tomcat-9/v9.0.110/bin/apache-tomcat-9.0.110
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 13036068 (12M) [application/x-gzip]
Saving to: 'apache-tomcat-9.0.110.tar.gz'

apache-tomcat-9.0.110.tar.gz      100%[=============================================================>

2025-10-08 10:16:43 (308 MB/s) - 'apache-tomcat-9.0.110.tar.gz' saved [13036068/13036068]
```

Vim monitoring.sh and paste this code

```
[root@ip-10-0-4-253 ~]# cat monitoring.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
[root@ip-10-0-4-253 ~]#
```

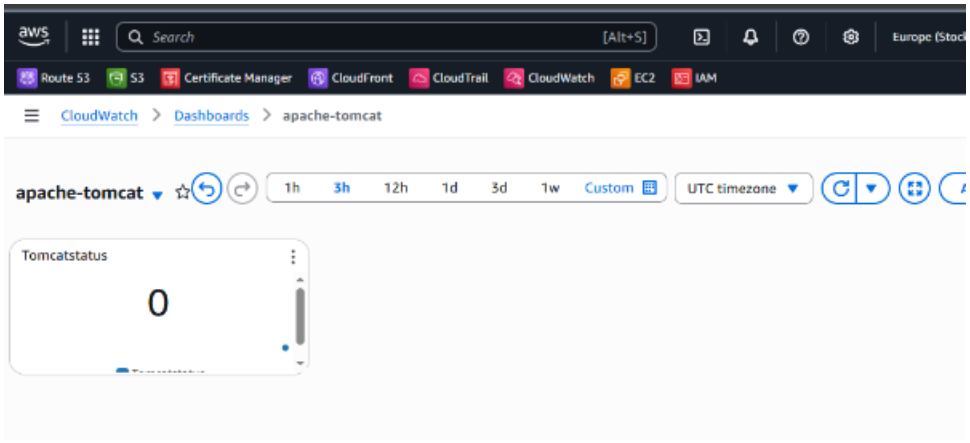
Change permission to chmod 777 monitoring.sh

```
[ec2-user@ip-10-0-4-253 ~]$ vim monitoring.sh
[ec2-user@ip-10-0-4-253 ~]$ chmod 777 monitoring.sh
[ec2-user@ip-10-0-4-253 ~]$ aws configure
AWS Access Key ID [None]: AKIATNTADWLT4ZEROEGX
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]: json
```

Aws configure and give the key,password.region,format.

Execute command -e and execute * * * * */opt/monitoring.sh

Go to cloudwatch and create dashboard and add widgets and select apache-tomcat.



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

Yum install nginx -y

```
[root@ip-10-0-4-253 ~]# yum install nginx -y
Last metadata expiration check: 1:44:52 ago on Wed Oct  8 10:14:22 2025.
Dependencies resolved.
================================================================================
 Package                         Architecture              Version
================================================================================
Installing:
 nginx                           x86_64                    1:1.28.0-1.
Installing dependencies:
 generic-logos-httpd             noarch                    18.0.0-12.a
 gperftools-libs                 x86_64                    2.9.1-1.amz
 libunwind                       x86_64                    1.4.0-5.amz
 nginx-core                      x86_64                    1:1.28.0-1.
 nginx-filesystem                noarch                    1:1.28.0-1.
 nginx-mimetypes                 noarch                    2.1.49-3.am

Transaction Summary
================================================================================
Install  7 Packages

Total download size: 1.1 M
Installed size: 3.7 M
Downloading Packages:
(1/7): generic-logos-httpd-18.0.0-12.amzn2023.0.3.noarch.rpm
(2/7): libunwind-1.4.0-5.amzn2023.0.3.x86_64.rpm
(3/7): gperftools-libs-2.9.1-1.amzn2023.0.3.x86_64.rpm
(4/7): nginx-1.28.0-1.amzn2023.0.2.x86_64.rpm
(5/7): nginx-filesystem-1.28.0-1.amzn2023.0.2.noarch.rpm
(6/7): nginx-mimetypes-2.1.49-3.amzn2023.0.3.noarch.rpm
```

Sudo systemctl start nginx

Sudo yum install cronie

Sudo systemctl start crond

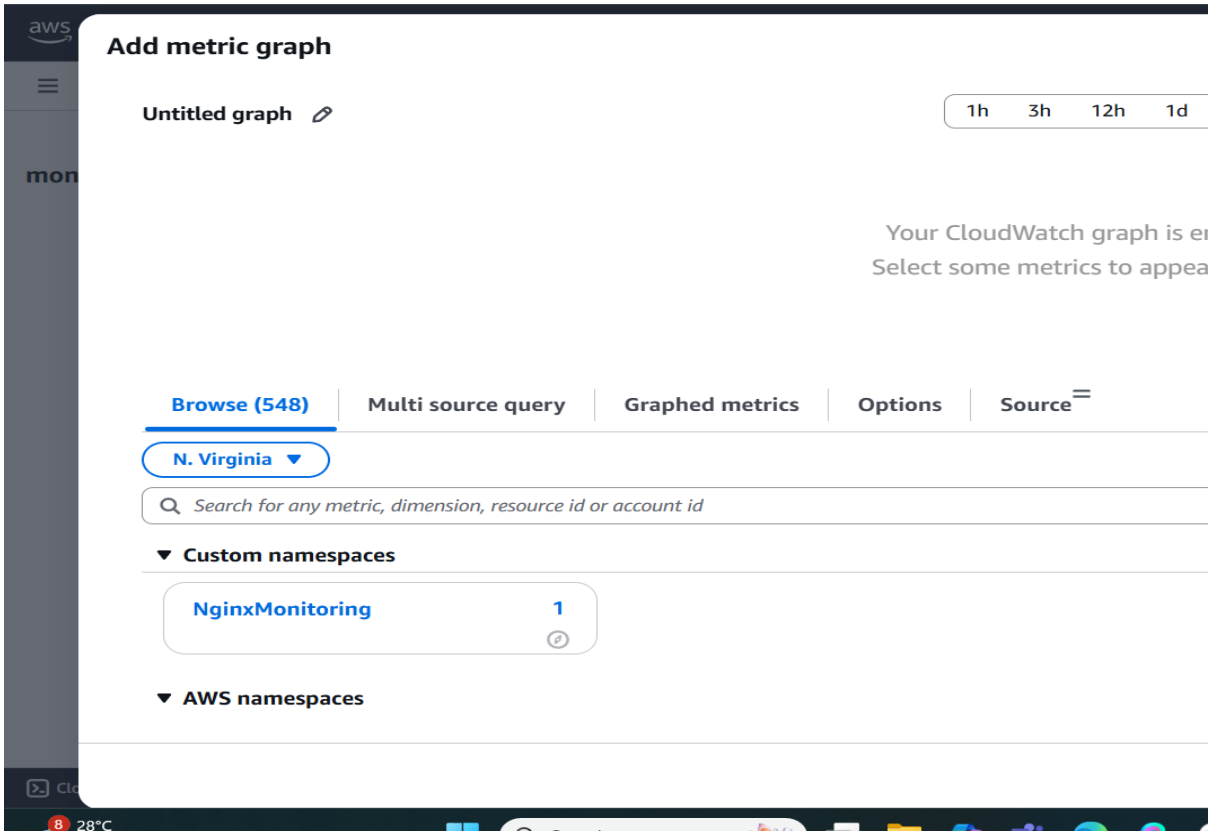
Crontab -e in that * * * * * /address of ngix you downloaded/nginix.sh

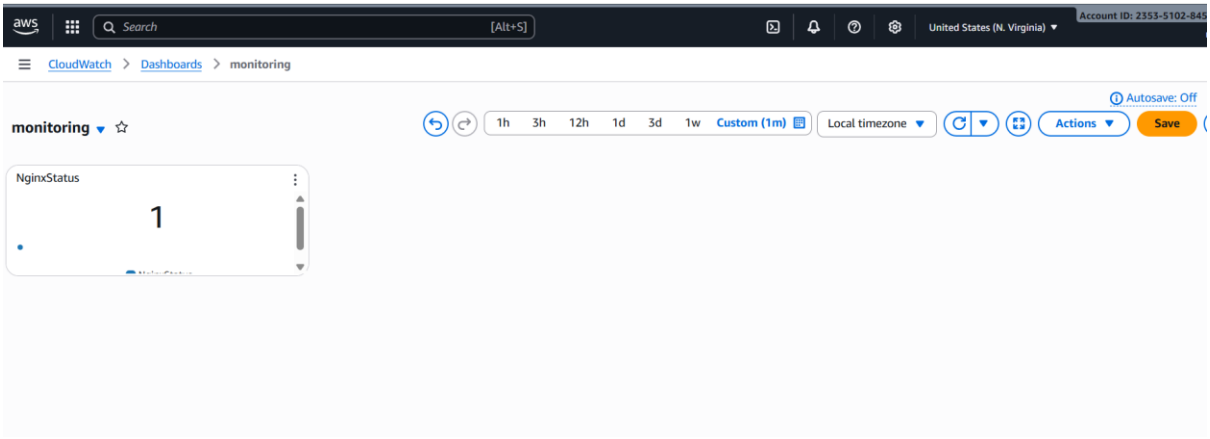
```
* * * * * /root/nginx.sh
~
```

Chang permissions to chmod 777.

```
/home/ec2-user
[ec2-user@ip-10-0-4-253 ~]$ sudo su -
Last login: Wed Oct  8 11:58:47 UTC 2025 on pts/1
[root@ip-10-0-4-253 ~]# pwd
/root
[root@ip-10-0-4-253 ~]# crontab -e
crontab: installing new crontab
[root@ip-10-0-4-253 ~]# aws configure
AWS Access Key ID [****************PVBX]: AKIATNTADWLTZJVQPVBX
AWS Secret Access Key [****************Eoeh]: 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
Default region name [us-east1]: us-east-1
Default output format [json]: json
[root@ip-10-0-4-253 ~]# vi nginx.sh
[root@ip-10-0-4-253 ~]# vi nginx.sh
[root@ip-10-0-4-253 ~]# chmod 777 nginx.sh
```

Go to cloud watch and see the nginx monitoring and select widget .





Go to alarms and create alarm select metric keep the threshold value less than 1 then it will create a alarm when its stopped.