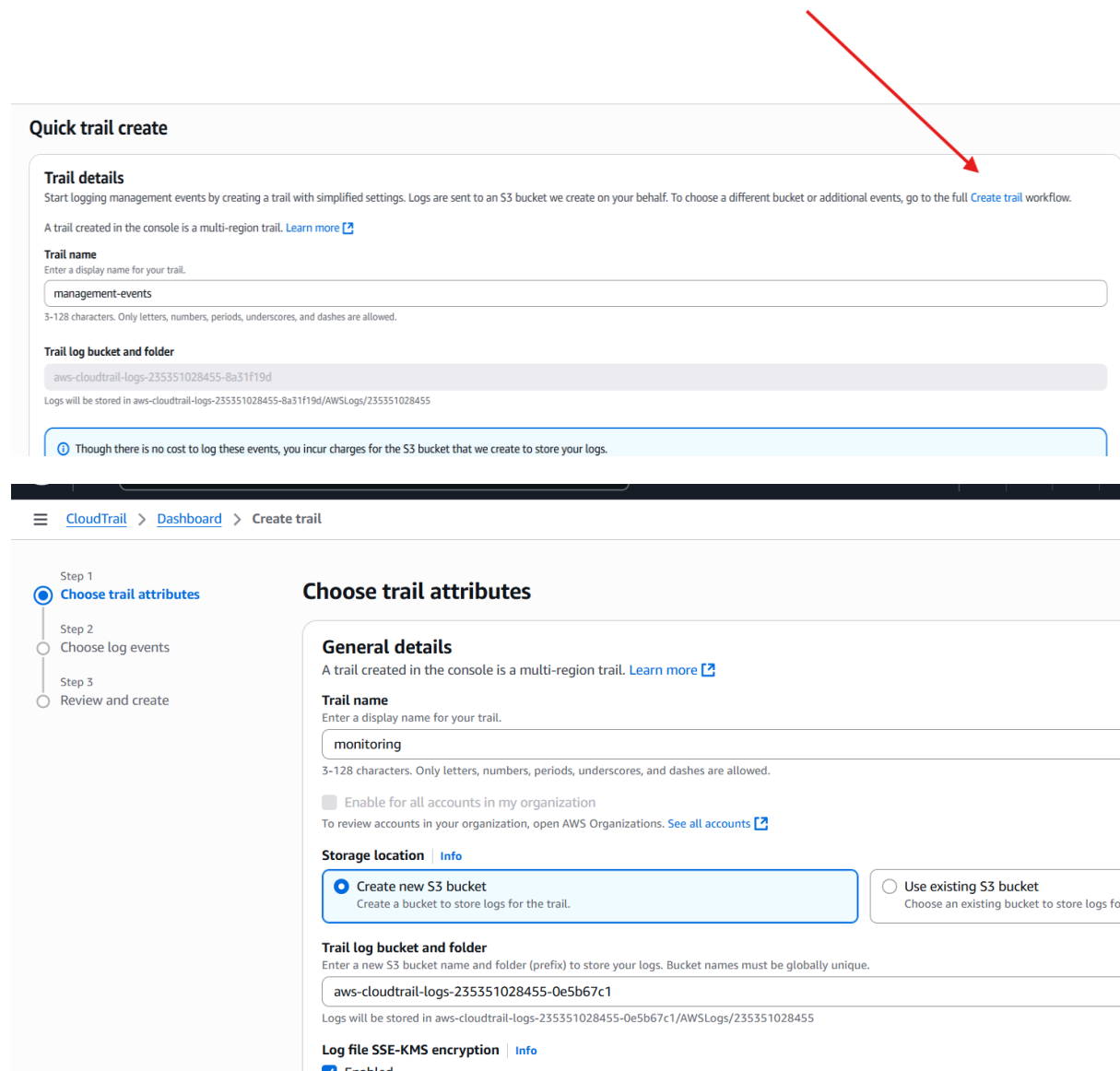


# 1.Enable cloudtrail monitoring and store the events in s3 and cloudwatch log events.

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



**Quick trail create**

**Trail details**  
Start logging management events by creating a trail with simplified settings. Logs are sent to an S3 bucket we create on your behalf. To choose a different bucket or additional events, go to the full [Create trail](#) workflow.

A trail created in the console is a multi-region trail. [Learn more](#)

**Trail name**  
Enter a display name for your trail.  
management-events  
3-128 characters. Only letters, numbers, periods, underscores, and dashes are allowed.

**Trail log bucket and folder**  
aws-cloudtrail-logs-235351028455-8a31f19d  
Logs will be stored in aws-cloudtrail-logs-235351028455-8a31f19d/AWSLogs/235351028455

**Choose trail attributes**

**General details**  
A trail created in the console is a multi-region trail. [Learn more](#)

**Trail name**  
Enter a display name for your trail.  
monitoring  
3-128 characters. Only letters, numbers, periods, underscores, and dashes are allowed.

☐ Enable for all accounts in my organization  
To review accounts in your organization, open AWS Organizations. [See all accounts](#)

**Storage location** | [Info](#)

☒ Create new S3 bucket  
Create a bucket to store logs for the trail.

☐ Use existing S3 bucket  
Choose an existing bucket to store logs for the trail.

**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-235351028455-0e5b67c1  
Logs will be stored in aws-cloudtrail-logs-235351028455-0e5b67c1/AWSLogs/235351028455

**Log file SSE-KMS encryption** | [Info](#)  
☒ Enabled

CloudTrail > Dashboard > Create trail

Step 1 Choose trail attributes

Step 2 **Choose log events**

Step 3 Review and create

### Choose log events

**Events** [Info](#)

Record API activity for individual resources, or for all current and future resources in AWS account. [Additional charges apply](#)

**Event type**

Choose the type of events that you want to log.

☒ **Management events**

Capture management operations performed on your AWS resources.

☐ **Data events**

Log the resource operations performed on or within a resource.

☐ **Insights events**

Identify unusual activity, errors, or user behavior in your account.

☐ **Network activity events**

Network activity events provide information about resource operations performed on a resource within a virtual private cloud endpoint.

**Management events** [Info](#)

Management events show information about management operations performed on resources in your AWS account.

[Multiple management events trails detected. Charges apply to duplicated logged management events. \[Additional charges apply\]\(#\)](#)

Trail successfully detected

### Trails

[Copy events to Lake](#) [Refresh](#) [Delete](#)

	Name	Home region	Multi-region trail	ARN	Insights	Organization trail	S3 bucket	Log file prefix	CloudWatch Logs log group
<input type="radio"/>	<a href="#">monitoring</a>	Europe (Stockholm)	Yes	arn:aws:cloudtrail:eu-north-1:235351028455:trail/monitoring	Disabled	No	<a href="#">aws-cloudtrail-logs-235351028455-0e5b67c1</a>	-	-

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

aws CloudWatch > Log groups > aws-cloudtrail-logs-235351028455-7fb404d2

Account ID: 2353-5102-8455

### aws-cloudtrail-logs-235351028455-7fb404d2

[Actions](#) [View in Logs Insights](#) [Start tailing](#) [Search log group](#)

**Log group details**

**Log class** [Info](#)

Standard

**ARN**

[arn:aws:logs:eu-north-1:235351028455:log-group:aws-cloudtrail-logs-235351028455-7fb404d2:](#)

**Creation time**

Now

**Retention**

Never expire

**Stored bytes**

-

**Metric filters**

0

**Subscription filters**

0

**Contributor Insights rules**

-

**KMS key ID**

-

**Anomaly detection**

[Configure](#)

**Data protection**

-

**Sensitive data count**

-

**Custom field indexes**

[Configure](#)

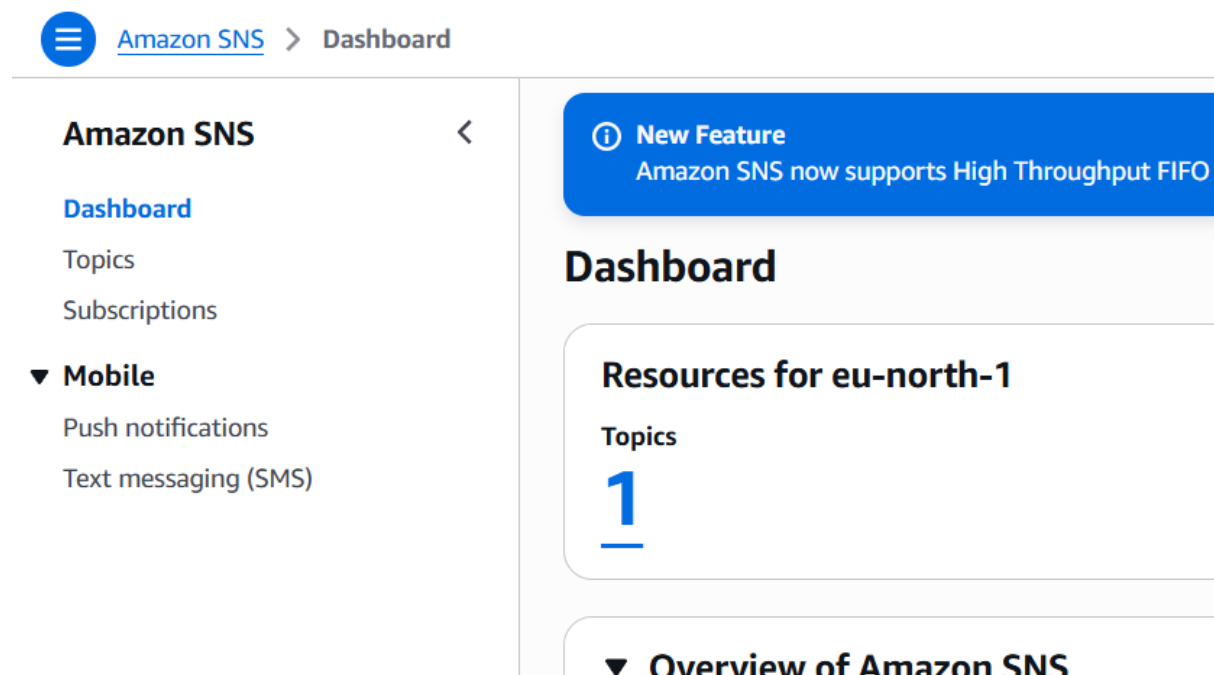
**Transformer**

[Configure](#)

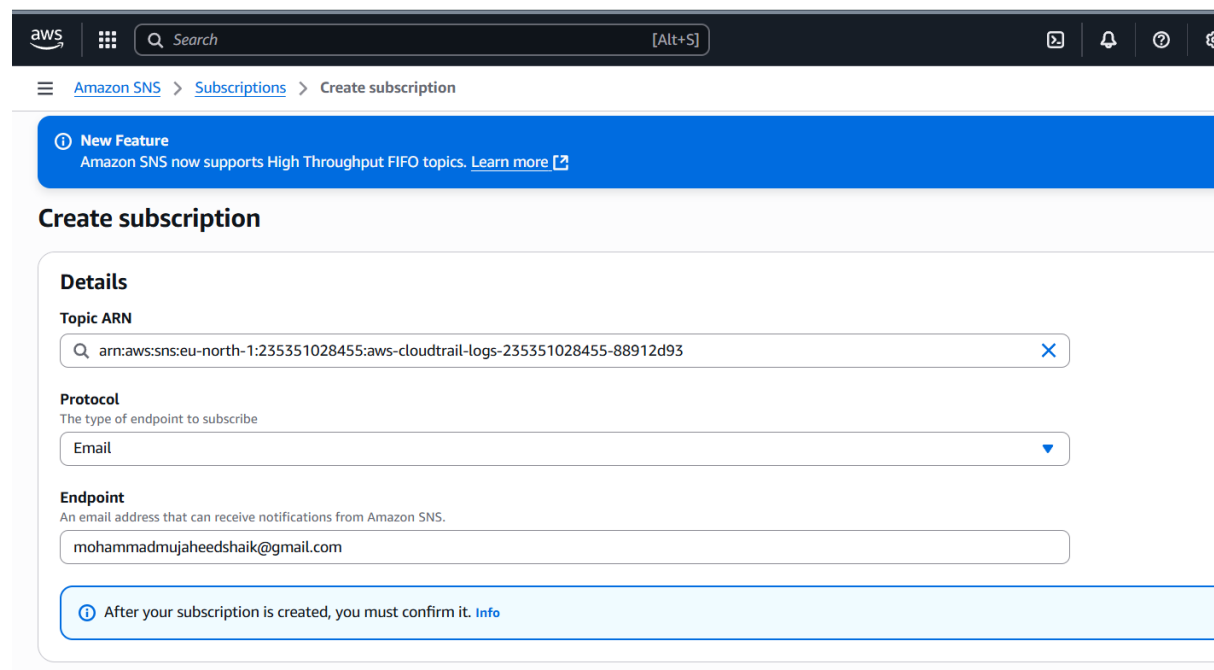
[Log streams](#) [Tags](#) [Anomaly detection](#) [Metric filters](#) [Subscription filters](#) [Contributor Insights](#) [Data protection](#) [Field](#)

## 2. Enable SNS for cloudtrail to send alert on email.

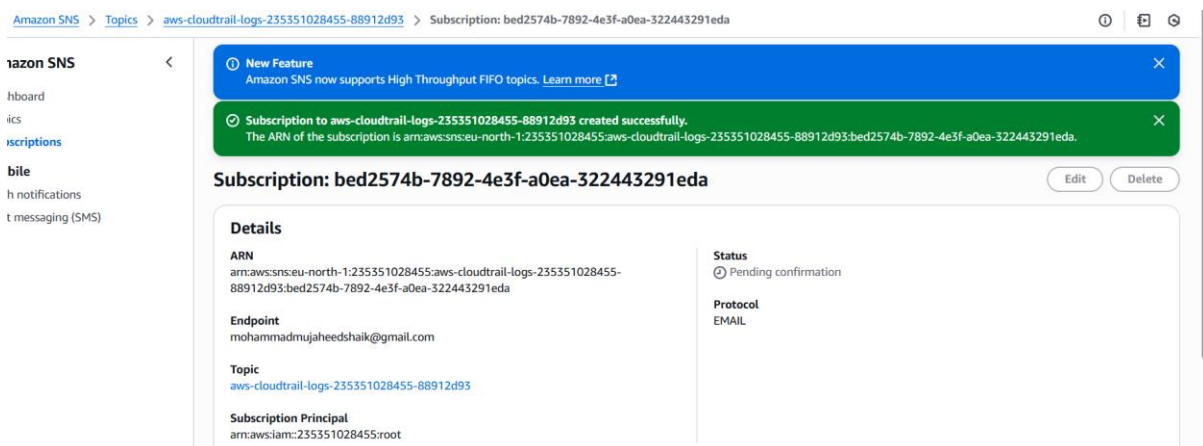
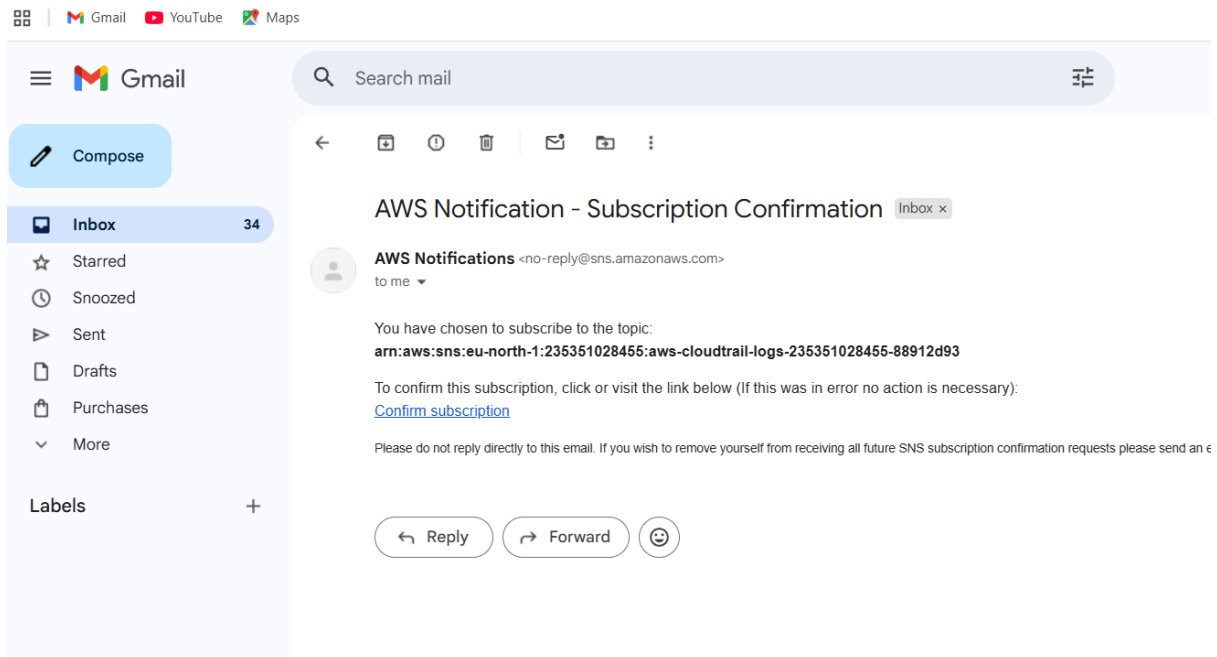
Go to sns



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

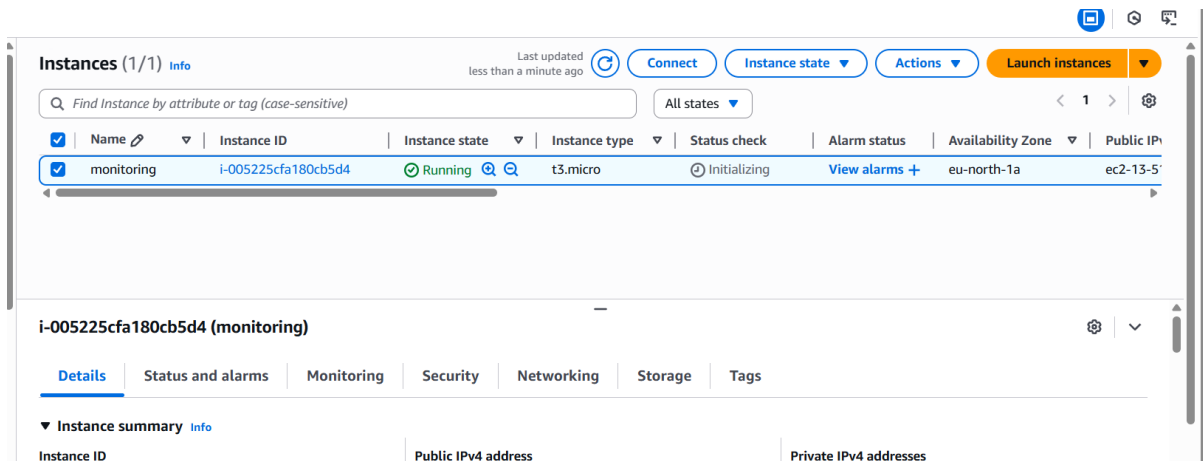


Go to your email and accept the request.

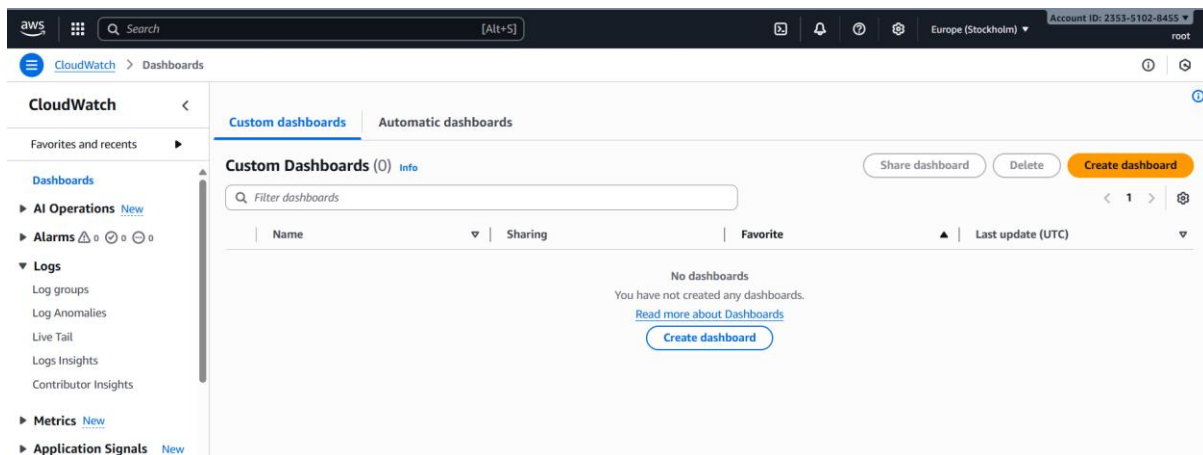


3. Configure cloud watch monitoring and record the cpu utilization and other metrics of ec2.

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](#) 


Add metric graph

Untitled graph 

1h 3h 12h 1d 3d 1w Custom  UTC timezone  Number   

Your CloudWatch graph is empty.  
Select some metrics to appear here.


Browse (766)



Stockholm 

Multi source query


Graphed metrics


Options

Source 

 Alarm recommendations 

Graph with SQL

Graph search 

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

EBS156

EC2236

Logs22

NATGateway46

S34

SNS4

TransitGateway81

Usage217

Cancel

Create widget

CPUUtilization [🔗](#)

☐ Persist time range ⓘ 1h 3h 12h 1d 3d 1w Custom ⓘ UTC timezone ▼ Number ▼ [🔄](#) [📄](#)

--

■ CPUUtilization

Browse (236)	Multi source query	Graphed metrics (1)	Options	Source	
<input type="checkbox"/>	No name specified	i-0f82cfd2cf0b30f7	StatusCheckFailed ⓘ		No alarms
<input type="checkbox"/>	No name specified	i-0f82cfd2cf0b30f7	StatusCheckFailed_AttachedEBS ⓘ		No alarms
<input type="checkbox"/>	No name specified	i-0f82cfd2cf0b30f7	StatusCheckFailed_Instance ⓘ		No alarms
<input type="checkbox"/>	No name specified	i-08b0ca6b5f1320...	EBSReadOps ⓘ		No alarms
<input checked="" type="checkbox"/>	No name specified	i-08b0ca6b5f1320...	CPUUtilization ⓘ		No alarms
<input type="checkbox"/>	No name specified	i-08b0ca6b5f1320...	EBSIOPBalance% ⓘ		No alarms
<input type="checkbox"/>	No name specified	i-08b0ca6b5f1320...	EBSPutBalance% ⓘ		No alarms

[Cancel](#) [Create widget](#)

aws [☰](#)  [Alt+S]

[☰](#) [CloudWatch](#) > [Dashboards](#) > dd

dd ▼ ☆ [↶](#) [↷](#) 1h 3h 12h

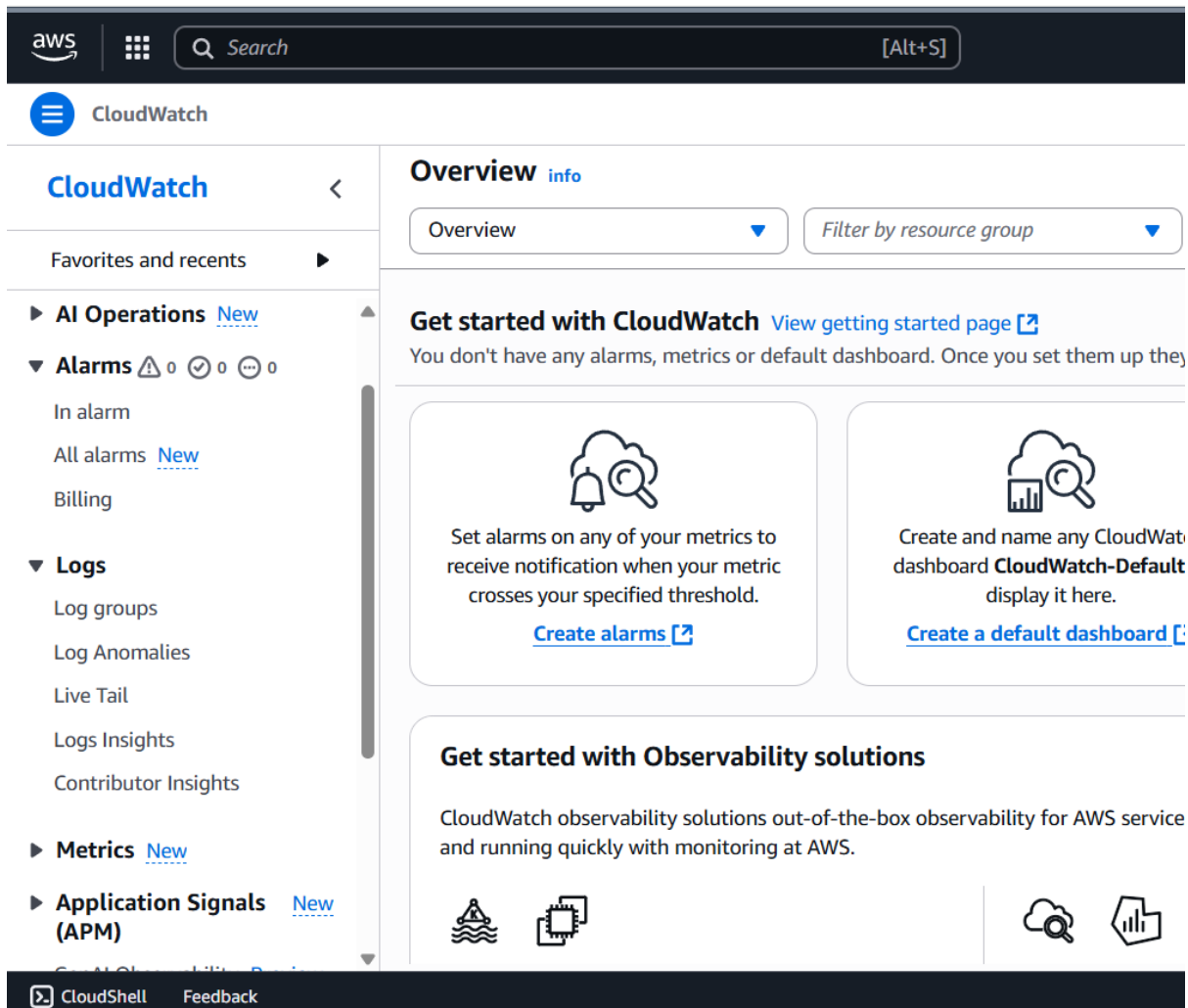
CPUUtilization ⓘ ⋮

100 %

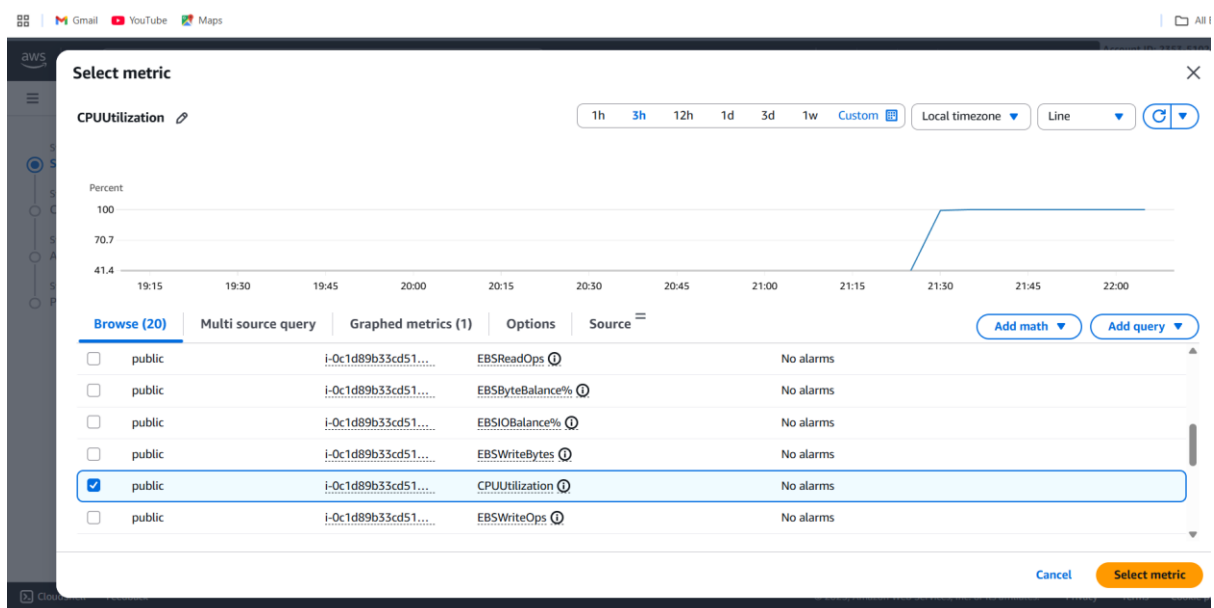
●

**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.





This screenshot shows the 'Create alarm' wizard in AWS CloudWatch, specifically the 'Specify metric and conditions' step. The interface includes a breadcrumb trail (CloudWatch > Alarms > Create alarm) and a timeline view showing CPUUtilization from 19:30 to 22:00. On the right, the 'Statistic' is set to 'Average' and the 'Period' is '5 minutes'. The 'Conditions' section is divided into 'Threshold type' (Static is selected) and 'Whenever CPUUtilization is...' (Greater > threshold is selected). A threshold value of '70' is entered in the 'than...' field. The bottom of the screen shows 'CloudShell' and 'Feedback' links, along with a copyright notice for Amazon Web Services.

Select in alarm and select sns topic.

This screenshot shows the 'Configure actions' step of the 'Create alarm' wizard. A blue banner at the top indicates 'Alarm recommendations available'. A sidebar on the left shows the progress: Step 1 (Specify metric and conditions), Step 2 (Configure actions - selected), Step 3 (Add alarm details), and Step 4 (Preview and create). The 'Notification' section has 'Alarm state trigger' set to 'In alarm'. Under 'Send a notification to the following SNS topic', 'Select an existing SNS topic' is chosen, and 'aws-cloudtrail-logs-235351028455-56089bf9' is selected from the dropdown list. The bottom of the screen contains a note about the listed topics.

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

The image shows two screenshots. The top screenshot is the AWS CloudWatch Alarms console. It features a green banner at the top stating "Successfully updated alarm cpu-g-70." Below this, the "Alarms (1)" section shows a table with one alarm: "cpu-g-70". The alarm is in the "In alarm" state, with a last state update of "2025-10-07 22:25:29". The condition is "CPUUtilization > 90 for 1 datapoints within 1 minute". The left sidebar shows the "Alarms" section expanded, with options for "In alarm", "All alarms", and "Billing".

The bottom screenshot is a Gmail interface showing an email notification from "AWS Notifications". The subject is "ALARM: 'cpu-g-70' in US East (N. Virginia)". The email body states: "You are receiving this email because your Amazon CloudWatch Alarm 'cpu-g-70' in the US East (N. Virginia) region has entered the ALARM state, because 'Threshold Crossed: 1 out of 1 datapoints [99.99500114193975 (07/10/25 16:50:00)] was greater than the threshold (70.0) (minimum 1 datapoint for OK -> ALARM transition).'" at "Tuesday 07 October, 2025 16:55:29". It includes a link to view the alarm in the AWS Management Console. Below the email body, there is a section for "Alarm Details" and "Monitored Metric".

**Alarm Details:**

- Name: cpu-g-70
- Description: INSUFFICIENT\_DATA -> ALARM
- State Change: INSUFFICIENT\_DATA -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [99.99500114193975 (07/10/25 16:50:00)] was greater than the threshold (70.0) (minimum 1 datapoint for ALARM transition).
- Timestamp: Tuesday 07 October, 2025 16:55:29 UTC
- AWS Account: 235351028455
- Alarm Arn: arn:aws:cloudwatch:us-east-1:235351028455:alarm:cpu-g-70

**Threshold:**

- The alarm is in the ALARM state when the metric is GreaterThanThreshold 70.0 for at least 1 of the last 1 period(s) of 60 seconds.

**Monitored Metric:**

- AWS/EC2
- CPUUtilization
- [Instanceid = i-0c1d89b33cd512984]

At the bottom of the Gmail interface, there is a notification bar that says "Enable desktop notifications for Gmail." with "OK" and "No thanks" buttons.

**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-9.0.110.tar.gz
--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 ~]# |
```

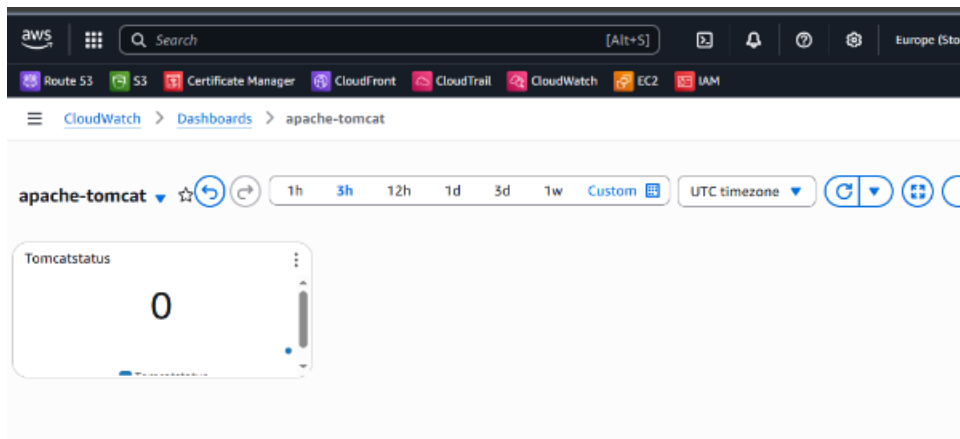
Change permission to chmod 777 monitoring.sh

```
[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.an
=====
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

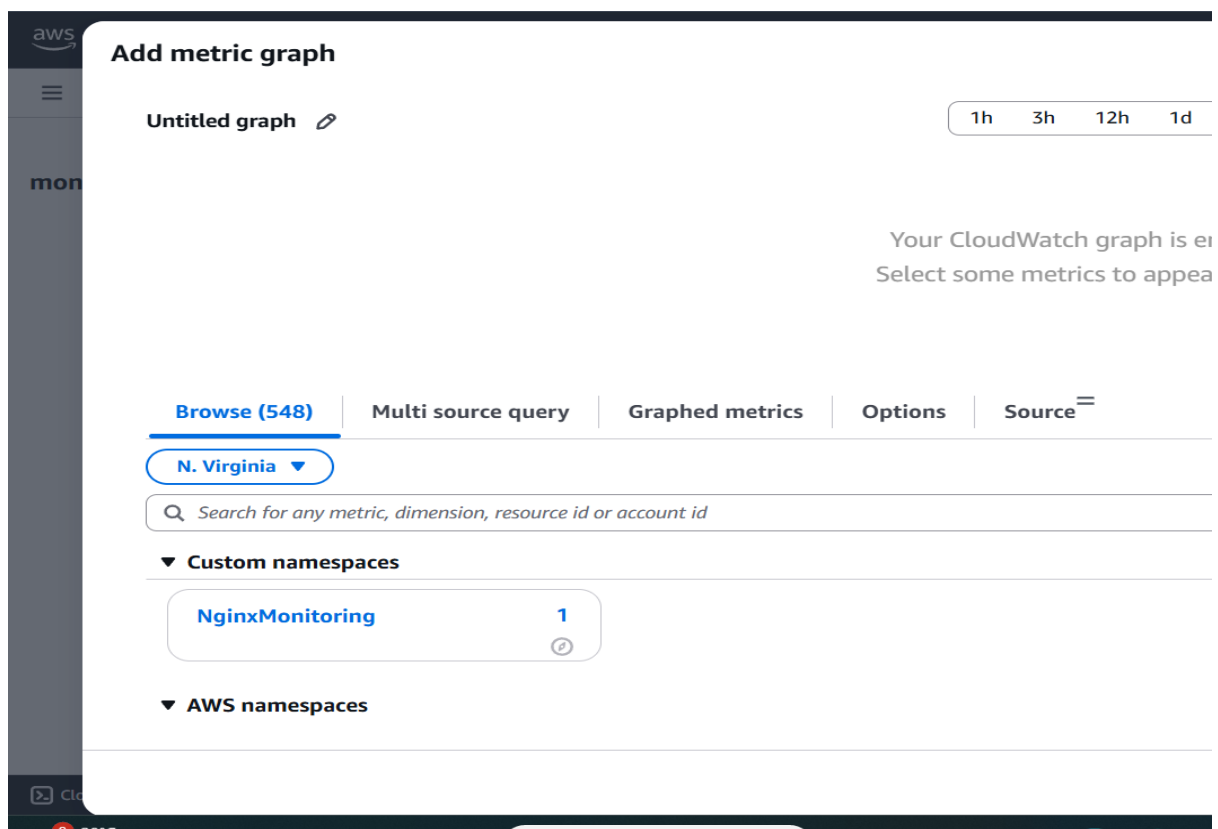
Crontab -e in that \* \* \* \* \* /address of nginx you downloaded/nginx.sh

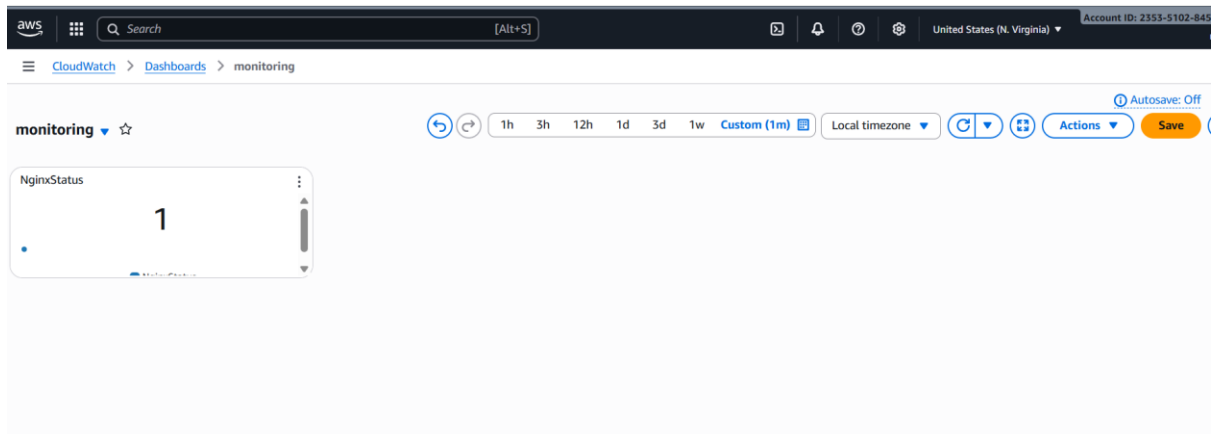
```
* * * * * /root/nginx.sh
```

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.

Alarms (1)

☐ Hide Auto Scaling alarms

Clear selection

Create composite alarm

Actions

Create alarm

Alarm state: OK

Alarm type: Any

Actions status: Any

1

<input type="checkbox"/>	Name	State	Last state update (Local)	Conditions	Actions
<input type="checkbox"/>	<a href="#">nginx stopped</a>	OK	2025-10-08 17:54:06	NginxStatus < 1 for 1 datapoints within 5 minutes	Actions enabled <span>Warning</span>