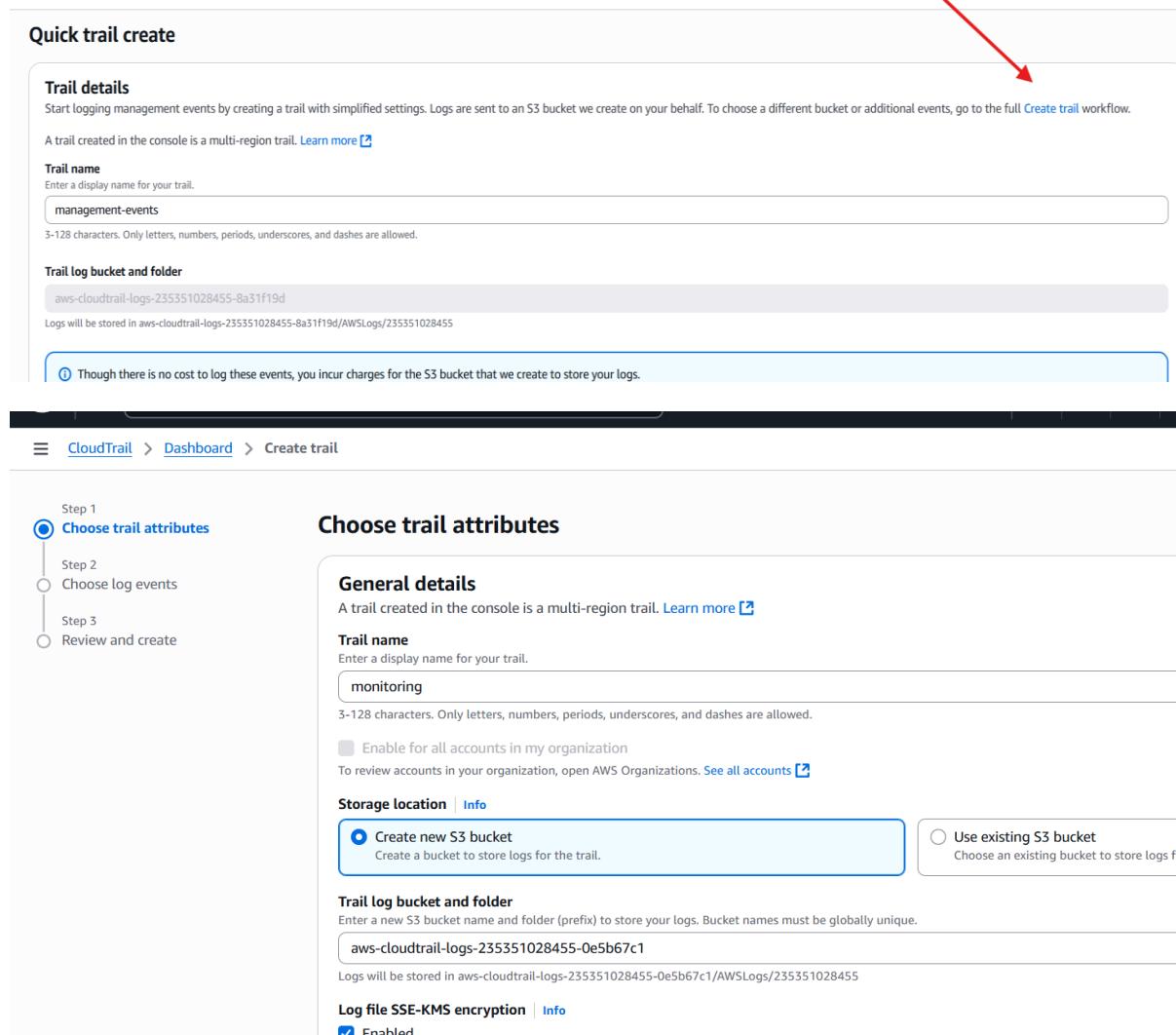


1. Enable cloudtrail monitoring and store the events in s3 and cloudwatch log events.

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.
 3-128 characters. Only letters, numbers, periods, underscores, and dashes are allowed.

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

ⓘ Though there is no cost to log these events, you incur charges for the S3 bucket that we create to store your logs.

CloudTrail > Dashboard > Create trail

Step 1 Choose trail attributes

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.
 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.
 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 deleted

Name	Home region	Multi-region trail	ARN	Insights	Organization trail	S3 bucket	Log file prefix	CloudWatch Logs log group
monitoring	Europe (Stockholm)	Yes	arn:aws:cloudtrail:eu-north-1:235351028455:trail/monitoring	Disabled	No	aws-cloudtrail-logs-235351028455-0e5b67c1	-	-

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

CloudWatch Account ID: 2353-5102-8455 root

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

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 cloudtrial to send alert on email.

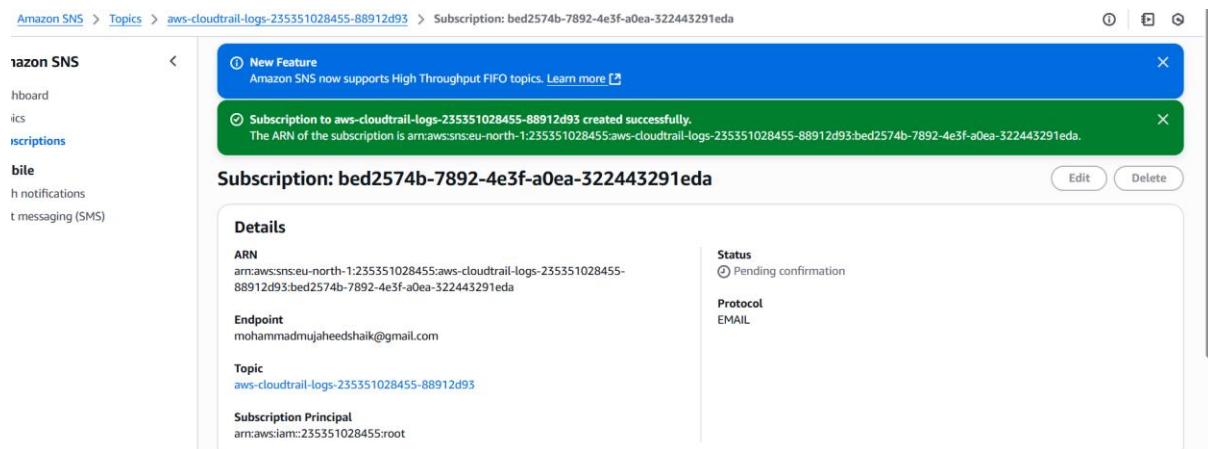
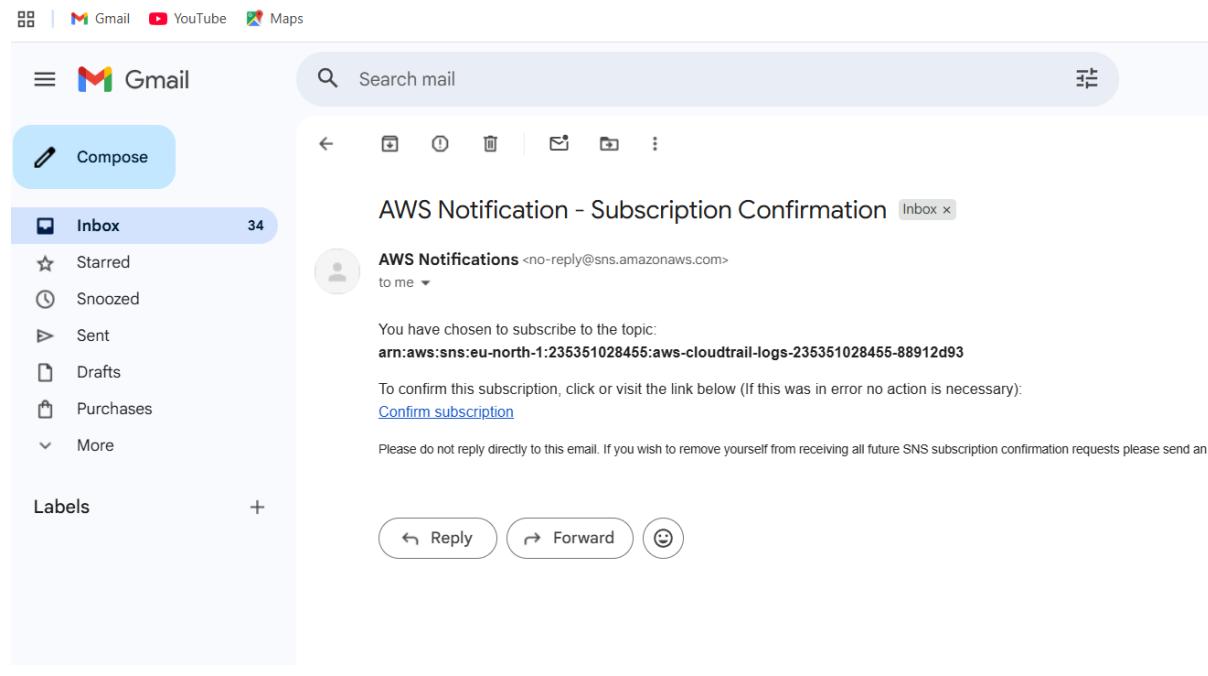
Go to sns

The screenshot shows the Amazon SNS Dashboard. On the left sidebar, there are links for 'Amazon SNS Dashboard', 'Topics', 'Subscriptions', and 'Mobile' (with sub-links for 'Push notifications' and 'Text messaging (SMS)'). A blue banner at the top right says 'New Feature: Amazon SNS now supports High Throughput FIFO'. The main area is titled 'Dashboard' and shows 'Resources for eu-north-1'. It has a 'Topics' section with a count of '1'.

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

The screenshot shows the 'Create subscription' page in the AWS SNS console. The top navigation bar includes the AWS logo, search bar, and other navigation links. The main content area has a 'New Feature' banner about High Throughput FIFO topics. Below it, the 'Create subscription' form has sections for 'Details', 'Topic ARN' (set to 'arn:aws:sns:eu-north-1:235351028455:aws-cloudtrail-logs-235351028455-88912d93'), 'Protocol' (set to 'Email'), and 'Endpoint' (set to 'mohammadmujaheedshaik@gmail.com'). A note at the bottom says 'After your subscription is created, you must confirm it.' with a link to 'Info'.

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.

The screenshot shows the AWS EC2 Instances page. At the top, there's a search bar with placeholder text 'Find Instance by attribute or tag (case-sensitive)' and a dropdown menu set to 'All states'. Below the search bar is a table header with columns: Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IP. A single instance is listed: 'monitoring' (Instance ID i-005225cfa180cb5d4), which is 'Running' (t3.micro). The 'Status check' column shows 'Initializing'. The 'Alarm status' column has a '+' sign. The 'Availability Zone' is 'eu-north-1a' and the 'Public IP' is 'ec2-13-5'. At the bottom of the table, there's a 'Launch instances' button.

i-005225cfa180cb5d4 (monitoring)

Details | Status and alarms | Monitoring | Security | Networking | Storage | Tags

▼ Instance summary Info

Instance ID | Public IPv4 address | Private IPv4 addresses

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

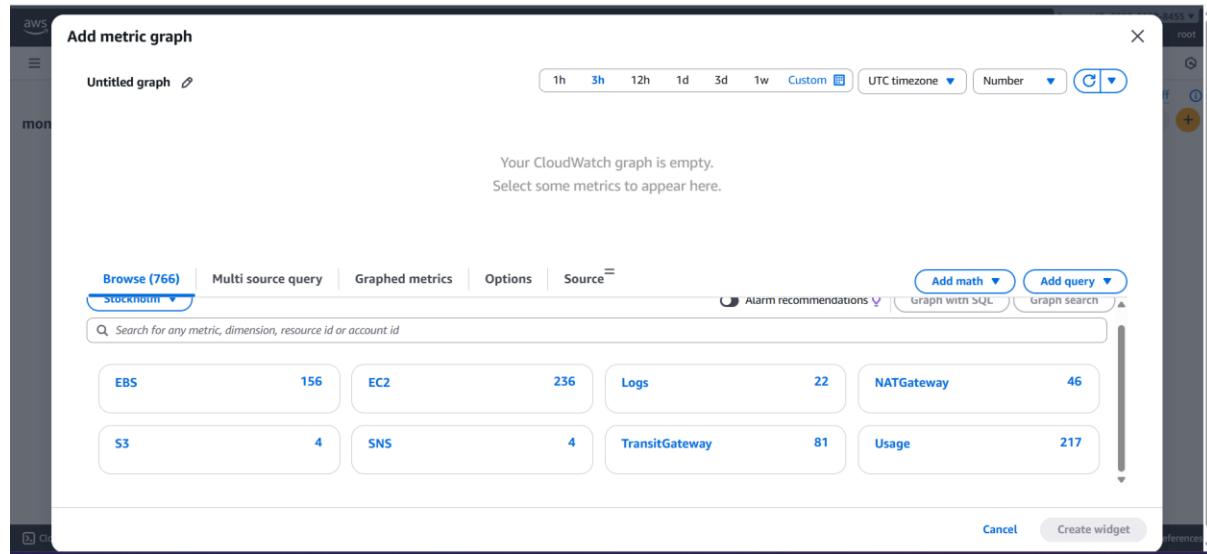
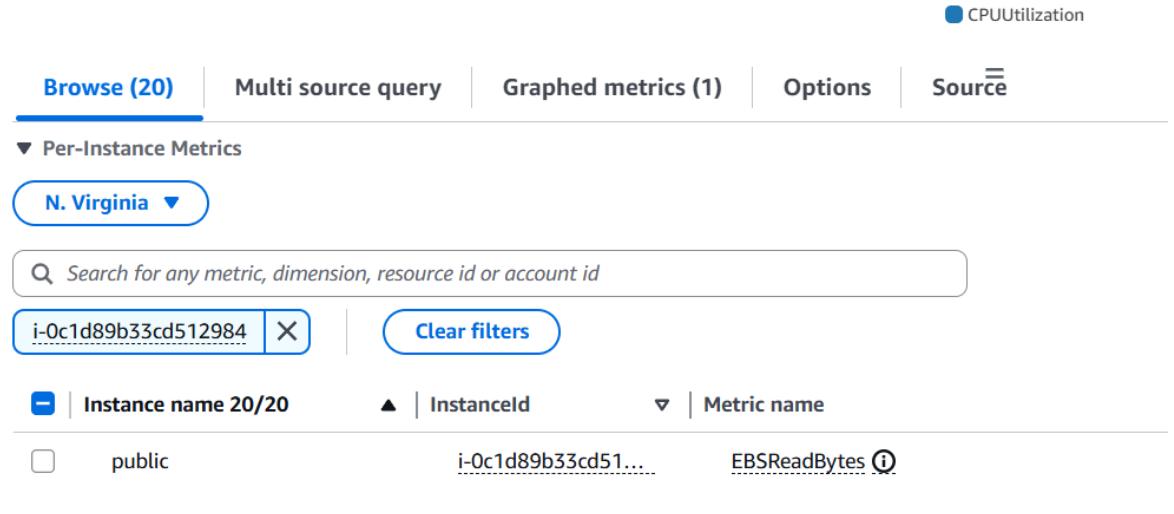
The screenshot shows the AWS CloudWatch Dashboards page. The left sidebar includes sections for Favorites and recents, Dashboards (selected), AI Operations, Alarms, Logs, Metrics, and Application Signals. The main area is titled 'Custom dashboards' and shows a table with one row: 'Custom Dashboards (0)'. The table has columns for Name, Sharing, and Favorite. There are buttons for 'Share dashboard', 'Delete', and 'Create dashboard'. A message at the bottom states 'No dashboards' and 'You have not created any dashboards.' with a 'Create dashboard' button.

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 3c

100 %



CPUUtilization

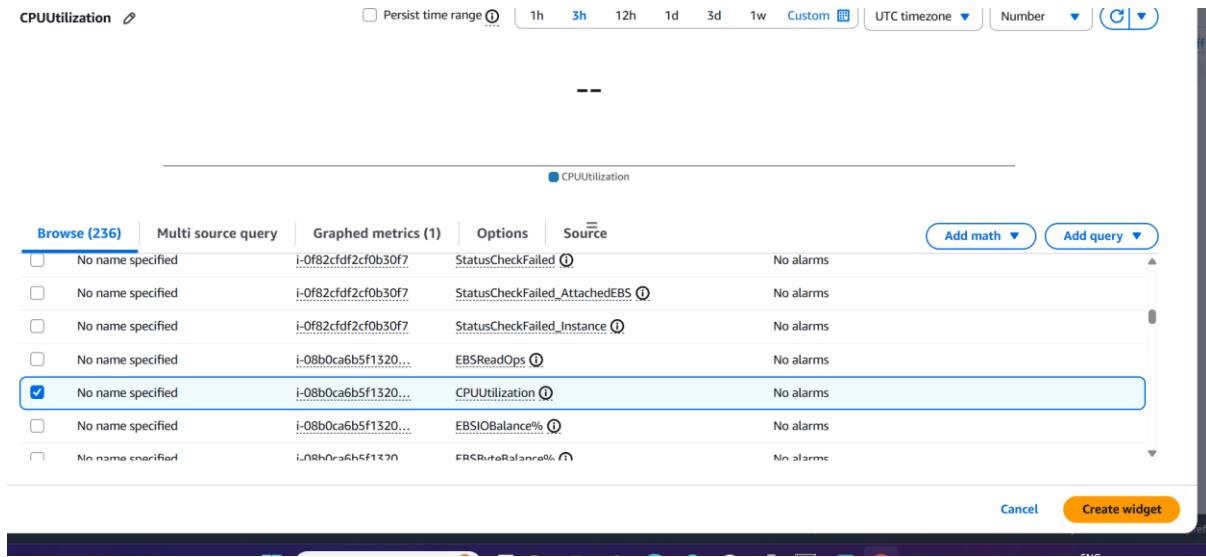
Persist time range 1h 3h 12h 1d 3d 1w Custom UTC timezone Number (G)

Browse (236) Multi source query Graphed metrics (1) Options Source

Add math Add query

No name specified	i-0f82cfdf2cf0b30f7	StatusCheckFailed	No alarms
No name specified	i-0f82cfdf2cf0b30f7	StatusCheckFailed_AttachedEBS	No alarms
No name specified	i-0f82cfdf2cf0b30f7	StatusCheckFailed_Instance	No alarms
No name specified	i-08b0ca6b5f1320...	EBSReadOps	No alarms
<input checked="" type="checkbox"/> No name specified	i-08b0ca6b5f1320...	CPUUtilization	No alarms
No name specified	i-08b0ca6b5f1320...	EBSIOBalance%	No alarms
No name specified	i-08b0ca6b5f1320...	EBSReadBalance%	No alarms

Cancel Create widget



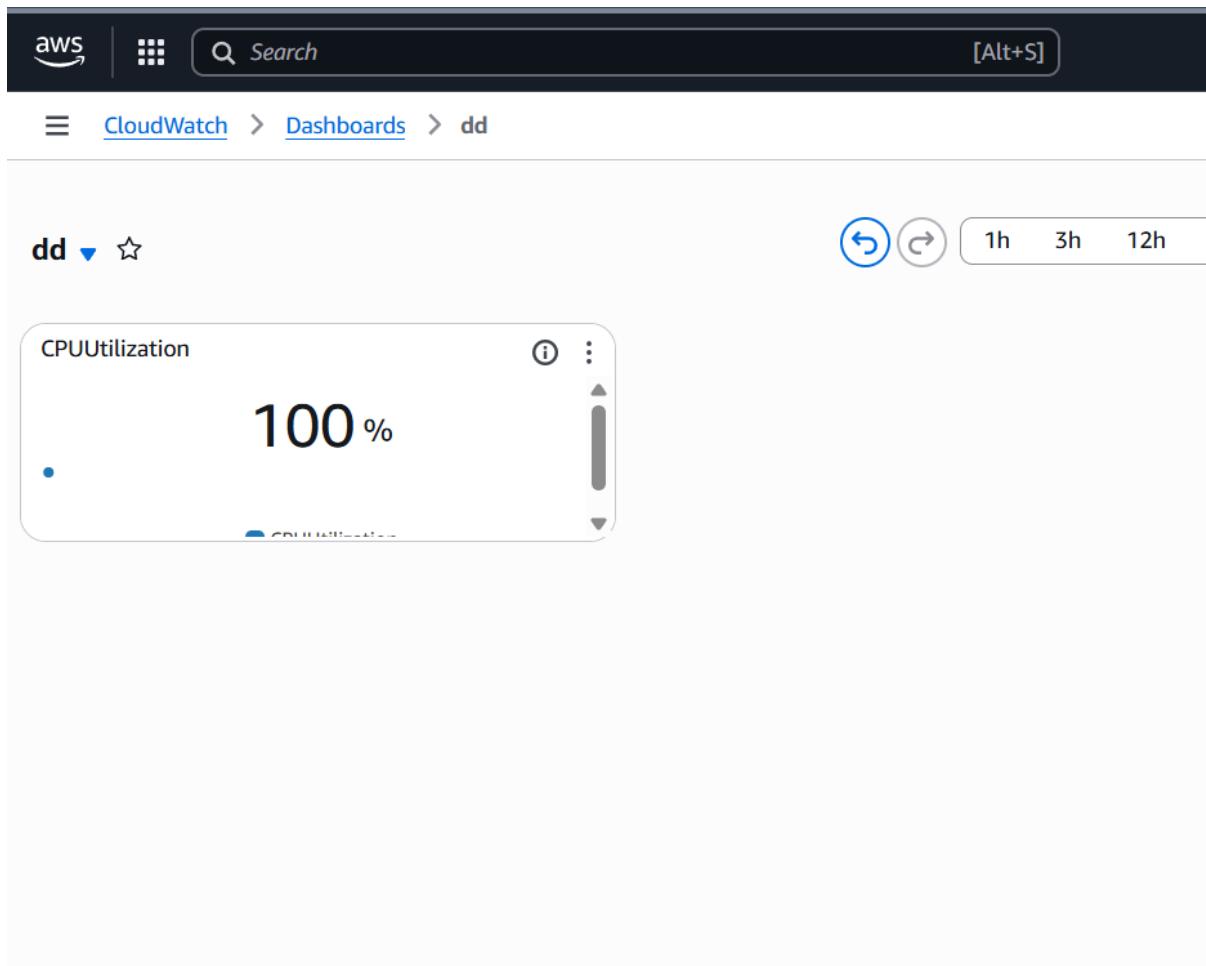
aws | Search [Alt+S]

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

aws | Search [Alt+S]

CloudWatch

CloudWatch

Favorites and recents

AI Operations New

Alarms ⚠ 0 ✓ 0 ... 0

- In alarm
- All alarms New
- Billing

Logs

- Log groups
- Log Anomalies
- Live Tail
- Logs Insights
- Contributor Insights

Metrics New

Application Signals (APM) New

CloudShell Feedback

Overview info

Overview Filter by resource group

Get started with CloudWatch View getting started page

You don't have any alarms, metrics or default dashboard. Once you set them up they will appear here.

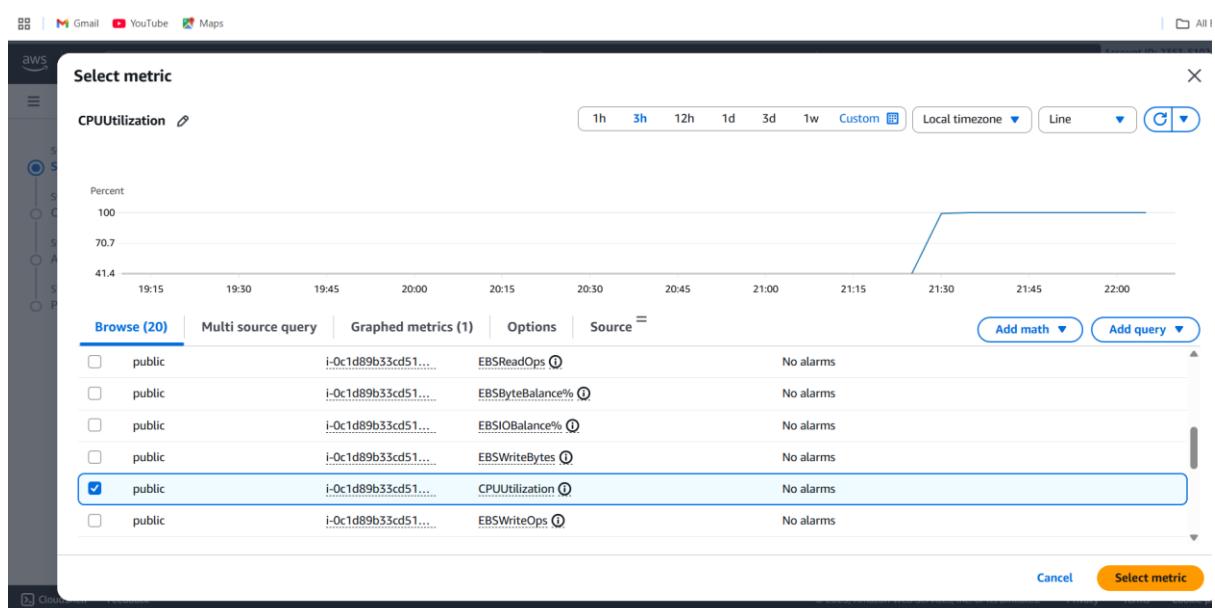
Set alarms on any of your metrics to receive notification when your metric crosses your specified threshold. [Create alarms](#)

Create and name any CloudWatch dashboard **CloudWatch-Default** and display it here. [Create a default dashboard](#)

Get started with Observability solutions

CloudWatch observability solutions out-of-the-box observability for AWS service and running quickly with monitoring at AWS.

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



The screenshot shows the 'Create alarm' wizard in the AWS CloudWatch console. In Step 1, 'Specify metric and conditions', the 'CPUUtilization' metric is selected from a dropdown. On the right, the 'Statistic' is set to 'Average' and the 'Period' is set to '5 minutes'. In Step 2, 'Configure actions', the 'Notification' section is active. It shows three options: 'In alarm' (selected), 'OK', and 'Insufficient data'. Below this, a list of SNS topics is shown, with 'aws-cloudtrail-logs-235351028455-56089bf9' selected. The sidebar on the left indicates the current step is Step 2.

Select in alarm and select sns topic.

The screenshot shows the 'Create alarm' wizard in the AWS CloudWatch console, specifically the 'Configure actions' step. The 'Notification' section is active. Under 'Alarm state trigger', the 'In alarm' option is selected. Below it, under 'Send a notification to the following SNS topic', the 'Select an existing SNS topic' option is selected, and the topic 'aws-cloudtrail-logs-235351028455-56089bf9' is listed in the dropdown. The sidebar on the left indicates the current step is Step 2.

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

The top half of the image shows the AWS CloudWatch Alarms console. A green banner at the top indicates "Successfully updated alarm cpu-g-70." The main area displays a table of alarms, with one row for "cpu-g-70" which is in an "In alarm" state. The condition for the alarm is "CPUUtilization > 90 for 1 datapoints within 1 minute". The bottom half of the image shows a Gmail inbox with an incoming email from "AWS Notifications" with the subject "ALARM: 'cpu-g-70' in US East (N. Virginia)". The email body provides details about the alarm, including its name, description, state change, reason for state change, timestamp, AWS account, and alarm ARN. It also includes a link to view the alarm in the AWS Management Console and a threshold section. A notification bar at the bottom of the Gmail window says "Enable desktop notifications for Gmail.".

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   Version
=====
Installing:
 java-17-amazon-corretto           x86_64        1:17.0
Installing dependencies:
 alsa-lib                            x86_64        1.2.7
 cairo                               x86_64        1.18.0
 dejavu-sans-fonts                  noarch       2.37-
 dejavu-sans-mono-fonts             noarch       2.37-
 dejavu-serif-fonts                noarch       2.37-
 fontconfig                          x86_64        2.13.0
 fonts-filesystem                   noarch       1:2.0
 freetype                            x86_64        2.13.0
 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        1:17.0
 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.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:e42::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%[=====] 13036068/13036068
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 ~]# |
```

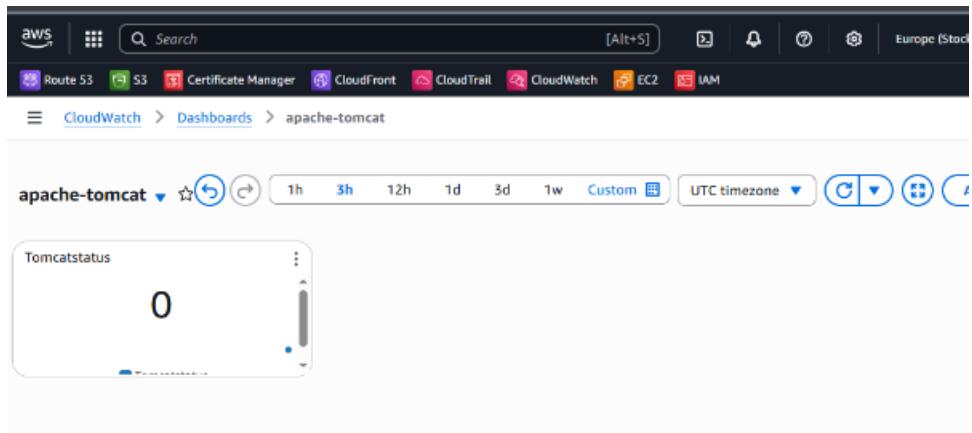
Change permission to chmod 777 monitoring.sh

```
[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

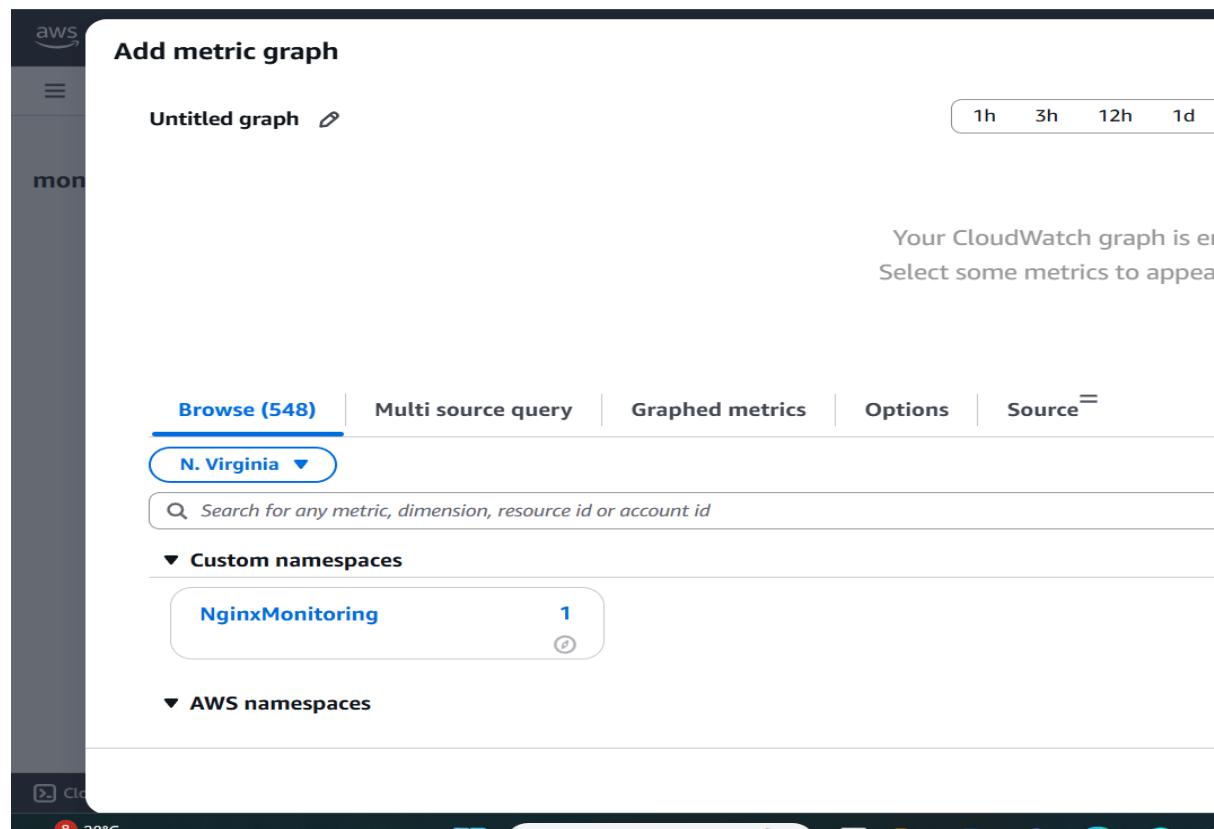
Crontab -e in that * * * * * /address of nginx you downloaded/nginx.sh

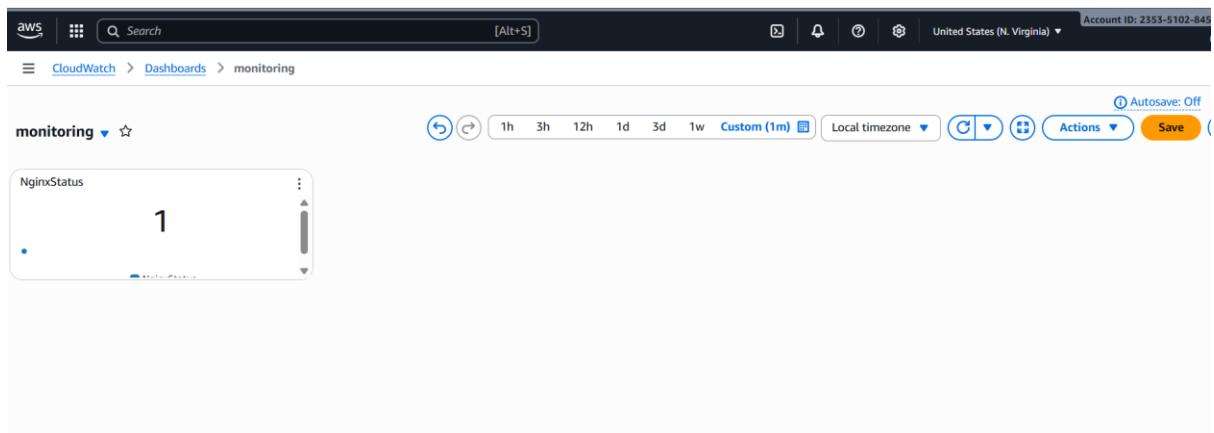
* * * * * /root/nginx.sh

Chang permissions to chmod 777.

```
/home/ecz-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]: AKIATNTADWL TZJVQPVBX  
AWS Secret Access Key [*****Eohh]: LBK4VK/zAn2P/j91TJIq900y2LVJp213l43AEoeh  
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)		<input type="checkbox"/> Hide Auto Scaling alarms	<input type="button" value="Clear selection"/>	<input type="button" value="Create composite alarm"/>	<input type="button" value="Actions"/>	<input type="button" value="Create alarm"/>
		<input type="text"/> Search	Alarm state: OK	Alarm type: Any	Actions status: Any	<input type="button" value="Actions"/>
Name	State	Last state update (Local)	Conditions	Actions		
nginx stopped	OK	2025-10-08 17:54:06	NginxStatus < 1 for 1 datapoints within 5 minutes	<input checked="" type="checkbox"/> Actions enabled	Warn	<input type="button" value="Edit"/>