

CloudTrail-CloudWatch-Task

1. Enable CloudTrail monitoring and store the events in S3 and CloudWatch log events.

Architecture

AWS Account

|

CloudTrail

|-----> S3 Bucket (long-term storage & audits)

|

|-----> CloudWatch Logs (real-time monitoring & alerts)

Step 1:

- Go to AWS console
- Search for S3(simple storage service)
- Click on Create bucket.
- Image shows successfully created bucket “neelimalogs”.

The screenshot shows the AWS S3 Buckets page. At the top, there is a green success message: "Successfully created bucket 'neelimalogs'. To upload files and folders, or to configure additional bucket settings, choose View details." Below this, there are two tabs: "General purpose buckets" (selected) and "Directory buckets". A search bar and a "Create bucket" button are visible. The main table lists two buckets: "aws-athena-query-results-814588452081-us-east-2-0pf98ayy" and "aws-cloudtrail-logs-814588452081-0a7db287". Both were created on October 15, 2025, at 11:15:47 UTC+05:30. On the right side, there are three informational boxes: "Account snapshot" (updated daily), "Storage Lens provides visibility into storage usage and activity trends.", and "External access summary" (updated daily), which helps identify bucket permissions for public access or access from external sources.

CloudTrail-CloudWatch-Task

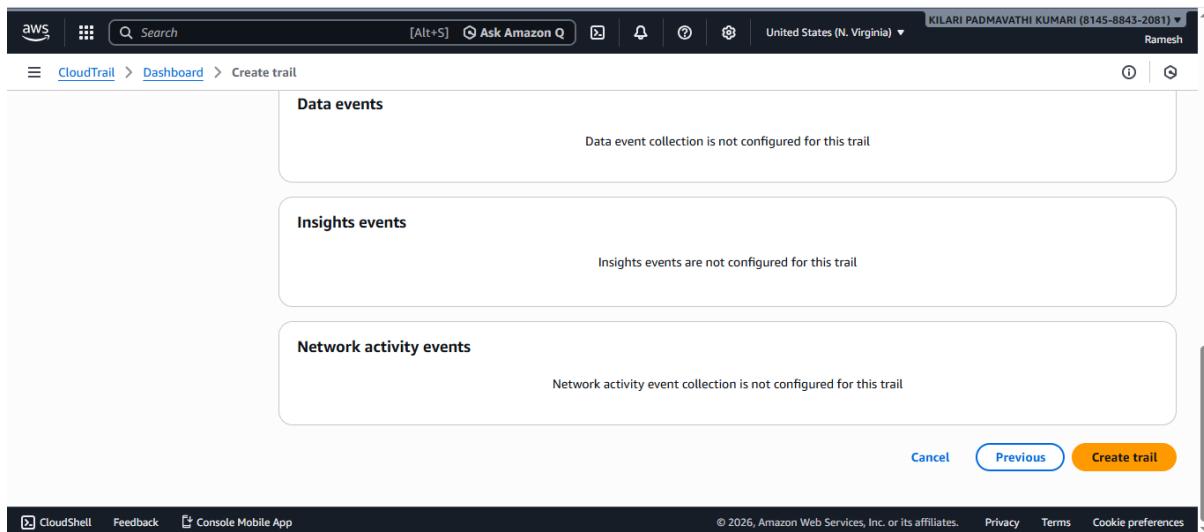
The screenshot shows the 'Create trail' wizard in the AWS CloudTrail console. It's Step 2: Choose log events. The 'General details' section is open, showing a trial named 'neelima-trail'. The 'Storage location' section shows the option 'Use existing S3 bucket' selected, with a prefix 'neelimalogs' entered. The 'Trail log bucket name' section is collapsed.

- Go to Cloud Trail
- Click on Dashboard and Create trail.
- Trail Name:neelima-trail.
- In storage location:use existing s3 bucket
- Browse trail log bucket name to store logs in that bucket.

The screenshot shows the 'Create trail' wizard in the AWS CloudTrail console. It's Step 3: Review and create. The 'Management events' section is expanded, showing a note about multiple management events detected. The 'API activity' section is expanded, showing options for Read (selected) and Write (selected), and checkboxes for Exclude AWS KMS events and Exclude Amazon RDS Data API events. The 'Next' button is visible at the bottom right.

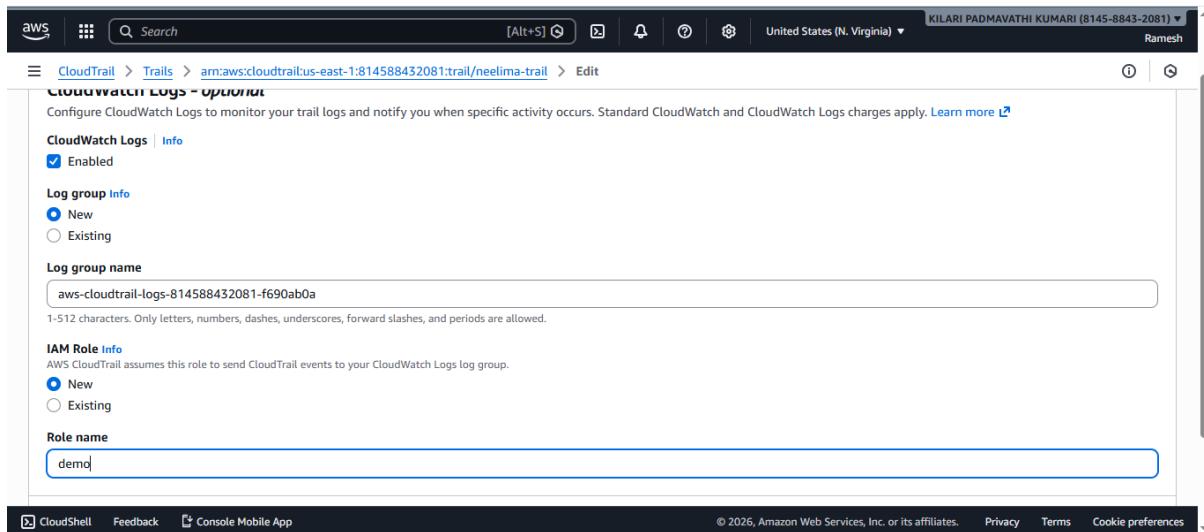
- In Events search for Management events
- API activity as Read and Write
- Click Next.

CloudTrail-CloudWatch-Task



The screenshot shows the 'Create trail' step in the AWS CloudTrail configuration wizard. It displays three sections: 'Data events', 'Insights events', and 'Network activity events'. Each section contains a message stating that event collection is not configured for this trail. At the bottom right, there are 'Cancel', 'Previous', and 'Create trail' buttons.

- Click on Create trail.



The screenshot shows the 'CloudWatch Logs - Overview' page for a specific CloudTrail trail. It includes fields for 'CloudWatch Logs' (Enabled), 'Log group' (New), 'Log group name' (aws-cloudtrail-logs-814588432081-f690ab0a), 'IAM Role' (New), and 'Role name' (demo). The page also includes a note about monitoring trail logs and standard charges apply.

- After creating trail we can edit cloudwatch setting (optional)
- Enable cloudwatch logs to watch logs.
- Log group as new in that logs are stored.
- Create IAM role as new for access permission.
- Click on save.

CloudTrail-CloudWatch-Task

The screenshot shows the AWS CloudTrail Trails page. In the 'General details' section, it lists:

- Trail logging: Logging (green status)
- Trail name: neelima-trail
- Multi-region trail: Yes
- Apply trail to my organization: Enabled for all accounts
- Trail log location: neelimalogs/AWSLogs/o-x662ed459n/814588432081
- Last log file delivered: January 27, 2026, 16:04:21 (UTC+05:30)
- Log file SSE-KMS encryption: Not enabled
- Log file validation: Enabled
- Last file validation delivered: -
- SNS notification delivery: Disabled
- Last SNS notification: -

In the 'CloudWatch Logs' section, it shows:

- Log group: aws-cloudtrail-logs-814588432081-f690ab0a
- IAM Role: arn:aws:iam::814588432081:role/service-role/demo

- The above image shows general details of cloud trail and cloud watch
- Trail logging status → logging.

The screenshot shows the 'Data events' configuration for the 'neelima-trail'. It includes:

- A note: "Choose the type of events that you want to log." with a checked checkbox for "Data events".
- A note: "Log the resource operations performed on or within a resource."
- A section titled "Data events info" with a note: "Data events show information about the resource operations performed on or within a resource. Additional charges apply." and a link to "Advanced event selectors are enabled".
- A note: "Use the following fields for fine-grained control over the data events captured by your trail." with a "Switch to basic event selectors" button.
- A section for "Data event: S3" with a "Remove" button.
- A "Resource type" dropdown set to "S3".
- A "Log selector template" dropdown set to "Log all events".
- A note: "Selector name - optional".

- To check s3 bucket logs we have edit data events
- Data event resource type:S3
- Log selector template:log all events.

CloudTrail-CloudWatch-Task

The screenshot shows the AWS CloudTrail Insights events configuration page. It lists various event types under 'Management events Insights types' (API call rate, API error rate) and 'Data events Insights types' (API call rate, API error rate). Both API call rate and API error rate are checked. At the bottom right are 'Cancel' and 'Save changes' buttons.

- In Insights events management events insight types
- Enable API call rate and API error rate
- Data event insight type
- Enable API call rate and API error rate
- Click on save changes.

The screenshot shows the AWS S3 Buckets list page. Under the 'neelimalogs' bucket, three folders are listed: 'cloudTrail-Digest/', 'cloudTrail-Insight/', and 'cloudTrail/'. The 'Actions' dropdown menu is visible above the objects table.

- Go to S3
- Click on buckets
- Select the bucket which we created as neelimalogs
- In object we can find cloudTrail-Digest/, cloudTrail-insight/,cloudTrail/ folders in it.

CloudTrail-CloudWatch-Task

The screenshot shows the AWS CloudTrail log file details page for the object `814588432081_CloudTrail_us-east-1_20260127T1035Z_E4qjNaeSHpm3zdlx.json.gz`. The left sidebar includes navigation links for Amazon S3, Buckets, Access management and security, and Storage management and insights. The main content area displays the Object overview, showing the Owner (neelimalogs), AWS Region (US East (N. Virginia) us-east-1), Last modified (January 27, 2026, 16:04:17 (UTC+05:30)), Size (1.2 KB), and S3 URI (`s3://neelimalogs/AWSLogs/o-x662ed459n/814588432081/CloudTrail/us-east-1/2026/01/27/814588432081_CloudTrail_us-east-1_20260127T1035Z_E4qjNaeSHpm3zdlx.json.gz`). The Properties tab is selected.

- The above image shows cloudtrail logs.

The screenshot shows the AWS CloudWatch Log Management interface. The left sidebar includes sections for CloudWatch, Favorites and recents, GenAI Observability, Application Signals (APM), Infrastructure Monitoring, Logs (selected), Log Management (New), Log Anomalies, Live Tail, Logs Insights, Contributor Insights, and Metrics. The main content area displays 'Log events' with a search bar, filter options (1m, 1h, UTC timezone), and a 'Display' dropdown. A table lists log entries with columns for Timestamp and Message. The first message is: "There are older events to load. [Load more](#)". Below it are seven log entries from 2026-01-27T10:45:08.442Z, each detailing an AWS Service invocation.

Timestamp	Message
2026-01-27T10:45:08.442Z	{"eventVersion": "1.11", "userIdentity": {"type": "AWSService", "invokedBy": "cloudtrail.amazonaws.com"}, "event...
2026-01-27T10:45:08.442Z	{"eventVersion": "1.11", "userIdentity": {"type": "AWSService", "invokedBy": "cloudtrail.amazonaws.com"}, "event...
2026-01-27T10:45:08.442Z	{"eventVersion": "1.11", "userIdentity": {"type": "AWSService", "invokedBy": "cloudtrail.amazonaws.com"}, "event...
2026-01-27T10:45:08.442Z	{"eventVersion": "1.11", "userIdentity": {"type": "AWSService", "invokedBy": "cloudtrail.amazonaws.com"}, "event...
2026-01-27T10:45:08.442Z	{"eventVersion": "1.11", "userIdentity": {"type": "AWSService", "invokedBy": "cloudtrail.amazonaws.com"}, "event...
2026-01-27T10:45:08.442Z	{"eventVersion": "1.11", "userIdentity": {"type": "AWSService", "invokedBy": "cloudtrail.amazonaws.com"}, "event...
2026-01-27T10:45:08.442Z	{"eventVersion": "1.09", "userIdentity": {"type": "IAMUser", "principalId": "AIDA33KKBB3IY1W3ROMVL", "arn": "arn:...

- In cloudwatch we can see logs in Log management
 - The above image displaying logs in cloudwatch.

CloudTrail-CloudWatch-Task

2. Enable SNS for CloudTrail to send alerts via email.

Objective

- Enable Amazon SNS so that CloudTrail events trigger notifications via email (usually through CloudWatch alarms or EventBridge).

Architecture Flow

- CloudTrail → CloudWatch Logs / EventBridge → SNS → Email

Step 1: Create an SNS Topic

- Go to AWS Console → SNS
- Click Topics → Create topic

Choose Type: Standard

The screenshot shows the 'Create topic' page in the AWS SNS console. At the top, there are tabs for 'Type' (selected) and 'Info'. Below that, a note says 'Topic type cannot be modified after topic is created'. There are two options: 'FIFO (first-in, first-out)' and 'Standard'. The 'Standard' option is selected and highlighted with a blue border. It lists: 'Best-effort message ordering', 'At-least once message delivery', and 'Subscription protocols: SQS, Lambda, Data Firehose, HTTP, SMS, email, mobile application endpoints'. Below this, there is a 'Name' field containing 'alert-cloudtrail'. Under 'Display name - optional', there is a field with 'My Topic'. At the bottom, there are links for 'CloudShell', 'Feedback', and 'Console Mobile App', along with copyright information and links for 'Privacy', 'Terms', and 'Cookie preferences'.

- Name: cloudtrail-alerts-topic
- Click Create topic

CloudTrail-CloudWatch-Task

The screenshot shows the AWS SNS console with a green success message: "Topic alert-cloudtrail created successfully. You can create subscriptions and send messages to them from this topic." The topic name is "alert-cloudtrail". The ARN is "arn:aws:sns:us-east-1:814588432081:alert-cloudtrail". The display name is empty, and the type is "Standard". The "Subscriptions" tab is selected, showing "(0)". There are buttons for "Edit", "Delete", "Request confirmation", "Confirm subscription", and "Create subscription".

- The above image shows topic create successfully.

Step 2: Create Email Subscription

1. Open the created SNS topic
2. Go to **Subscriptions → Create subscription**
3. Select:
 - **Protocol:** Email
 - **Endpoint:** your email address
4. Click **Create subscription**

The screenshot shows the "Create subscription" page. In the "Details" section, the "Topic ARN" is "arn:aws:sns:us-east-1:814588432081:alert-cloudtrail". The "Protocol" is set to "Email". The "Endpoint" is "neelimarani2398@gmail.com". A note at the bottom says "After your subscription is created, you must confirm it." To the right, there is a "Confirm your subscription" sidebar with instructions about subscription confirmation messages and a note that Amazon SNS doesn't send messages to an endpoint until the subscription is confirmed.

CloudTrail-CloudWatch-Task

The screenshot shows the AWS SNS console with a green success message: "Subscription to alert-cloudtrail created successfully. The ARN of the subscription is arn:aws:sns:us-east-1:814588432081:alert-cloudtrail:33b2a1ab-6a8d-4a19-9957-d8b824ea93cb." Below this, a table displays the subscription details:

Details	
ARN	arn:aws:sns:us-east-1:814588432081:alert-cloudtrail:33b2a1ab-6a8d-4a19-9957-d8b824ea93cb
Endpoint	neelimarani2398@gmail.com
Topic	alert-cloudtrail
Subscription Principal	arn:aws:iam::814588432081:user/Ramesh
Status	Pending confirmation
Protocol	EMAIL

5. **Confirm the subscription** from the email you receive, Until confirmed, alerts will NOT be delivered.

The screenshot shows a Gmail inbox with a single email from "AWS Notifications <no-reply@sns.amazonaws.com>" titled "AWS Notification - Subscription Confirmation". The email body contains the following text:

Why is this message in spam? This message is similar to messages that were identified as spam in the past.
[Report not spam](#)

You have chosen to subscribe to the topic:
arn:aws:sns:us-east-1:814588432081:alert-cloudtrail

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

Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to [sns-opt-out](#).

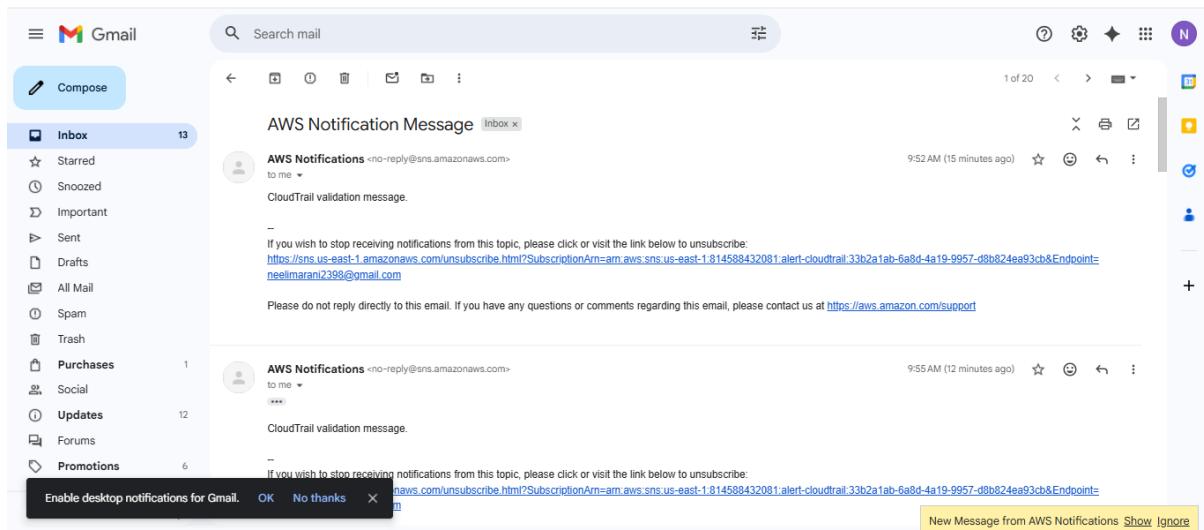
The screenshot shows a confirmation email from AWS SNS. The subject is "Subscription confirmed!" and the body contains the following text:

You have successfully subscribed.

Your subscription's id is:
arn:aws:sns:us-east-1:814588432081:alert-cloudtrail:33b2a1ab-6a8d-4a19-9957-d8b824ea93cb

If it was not your intention to subscribe, [click here to unsubscribe](#).

CloudTrail-CloudWatch-Task



Conclusion: The above image shows that we are receiving AWS Notification message through Email.

CloudTrail-CloudWatch-Task

3. Create one alarm to send an alert to email if the CPU utilization is more than 50 percent.

- Go to Aws console
- Search for EC2 and create an instance.

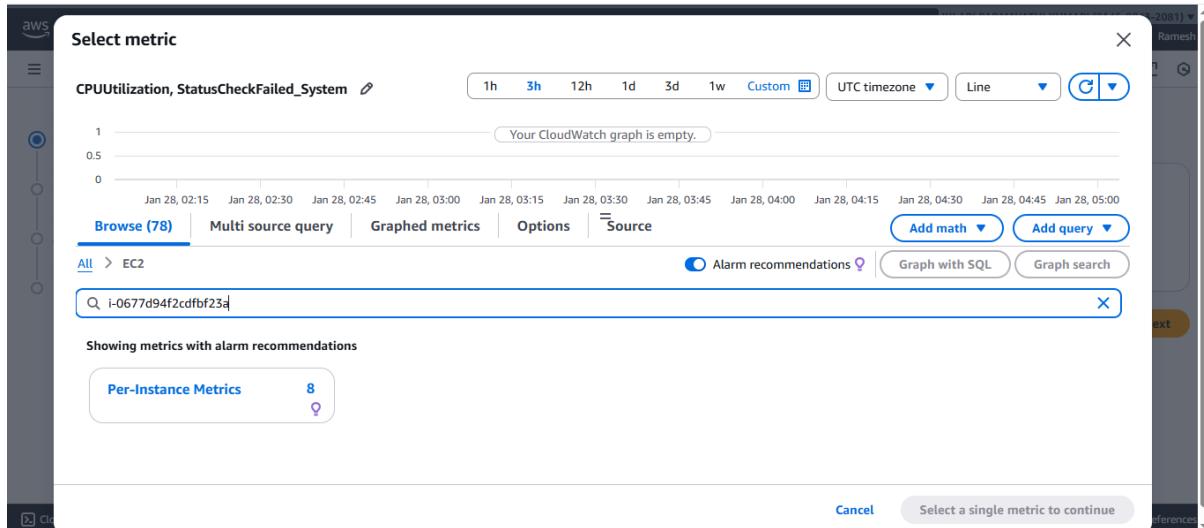
The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation sidebar with options like Dashboard, EC2 Global View, Events, Instances (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, Capacity Manager (New), and Images. The main area displays 'Instances (1/2) Info'. It lists two instances: 'A' (Instance ID: i-02861daeb77063a4f, Status: Stopped, Type: t3.micro) and 'Test-ec2' (Instance ID: i-0677d94f2cdafb23a, Status: Running, Type: t2.micro). Below the table, there's a detailed view for 'Test-ec2' showing its Public IPv4 address (54.84.119.212), Private IPv4 addresses (172.31.37.203), and Public DNS. At the bottom right of the main area, there are links for CloudShell, Feedback, and Console Mobile App, along with copyright information for 2026, Amazon Web Services, Inc. or its affiliates, and links for Privacy, Terms, and Cookie preferences.

- Go to Aws console
- Search for SNS (simple Notification service)
- Create a topic: -alert CloudTrail and
- Create a Subscription
- click on save.

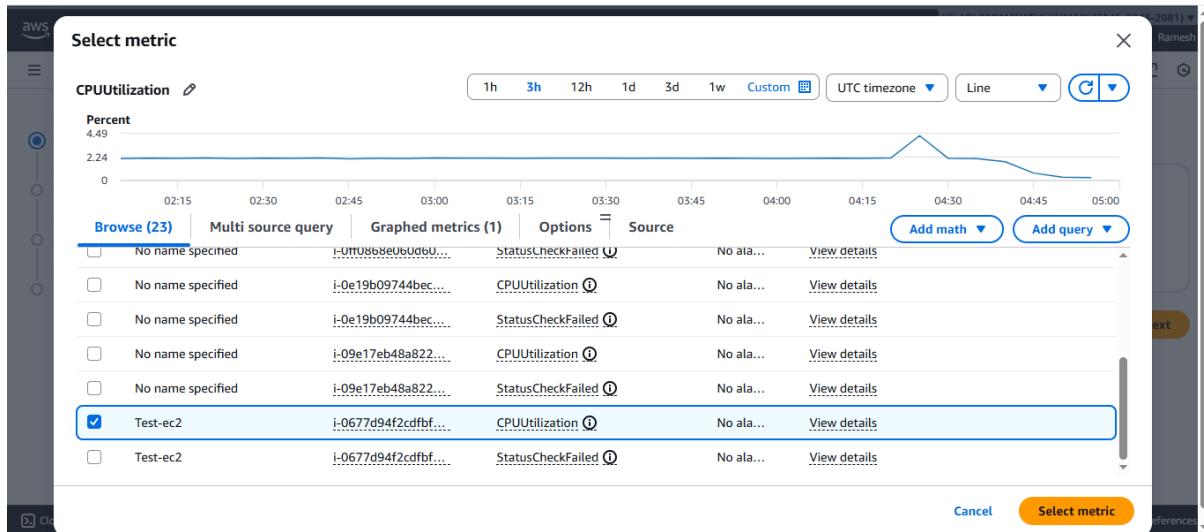
The screenshot shows the AWS Amazon SNS Topics page. The left sidebar includes options for Dashboard, Topics (selected), Subscriptions, and Mobile (Push notifications, Text messaging (SMS)). The main content area shows a topic named 'alert-cloudtrail'. A green success message at the top says 'Save changes Topic alert-cloudtrail saved successfully.' Below the message, there's a 'Details' section with fields for Name (alert-cloudtrail), ARN (arn:aws:sns:us-east-1:814588432081:alert-cloudtrail), Display name (cloud-watch alert), and Type (Standard). Underneath the details, tabs for Subscriptions, Access policy, Data protection policy, Delivery policy (HTTP/S), and Delivery status logging are visible. The 'Subscriptions' tab is selected, showing one subscription (1) with an 'Edit' button. At the bottom, there are buttons for Request confirmation, Confirm subscription, and Create subscription, along with links for CloudShell, Feedback, and Console Mobile App, and standard footer links for 2026, Amazon Web Services, Inc. or its affiliates, Privacy, Terms, and Cookie preferences.

CloudTrail-CloudWatch-Task

- Go to CloudWatch
- And select Metric in that search for EC2.

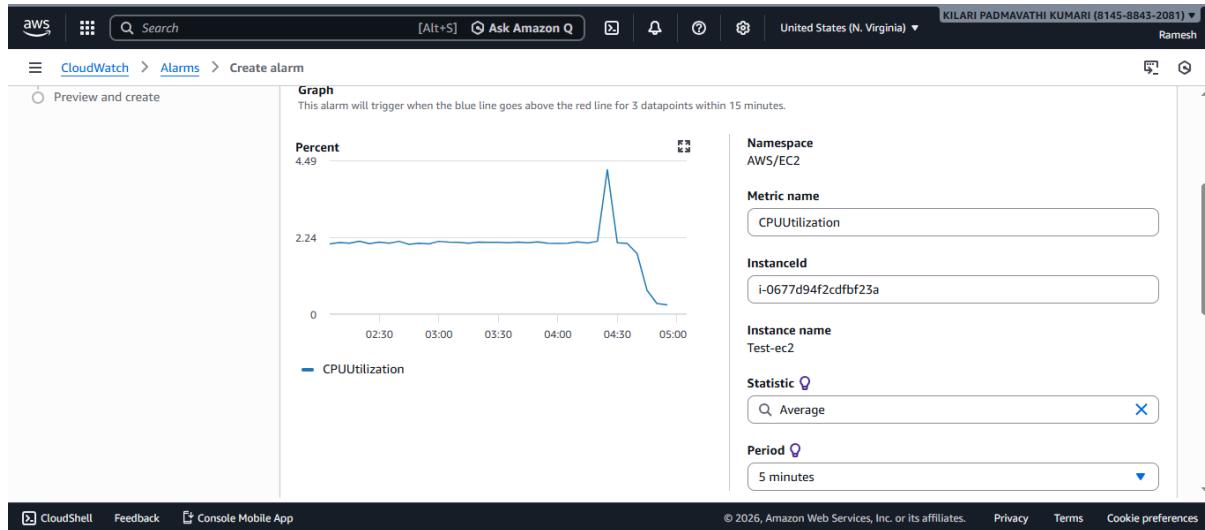


- Find desired EC2 we required and search CPU utilizations



CloudTrail-CloudWatch-Task

- In CloudWatch click on alarms
- Create Alaram
- Select Metric Name CPU utilization.



- In conditions Threshold Type as Static
- Whenever CPUUtilization is Greater and Condition in that define the threshold value.

The screenshot shows the 'Create alarm' step in the AWS CloudWatch Metrics Metrics Explorer. The 'Conditions' section is displayed, allowing users to define an alarm based on specific metrics and thresholds. The 'Threshold type' is set to 'Static' (radio button selected). The 'Whenever CPUUtilization is...' section contains four options: 'Greater' (radio button selected), 'Greater/Equal', 'Lower/Equal', and 'Lower'. The 'than...' section specifies a threshold value of '50'. Below this, a link 'Additional configuration' is visible.

CloudTrail-CloudWatch-Task

Step 2
Configure actions

Notification

Alarm state trigger
Define the alarm state that will trigger this action.

In alarm
The metric or expression is outside of the defined threshold.

OK
The metric or expression is within the defined threshold.

Insufficient data
The alarm has just started or not enough data is available.

Remove

Send a notification to the following SNS topic
Define the SNS (Simple Notification Service) topic that will receive the notification.

Select an existing SNS topic
alert-cloudtrail

Create new topic
alert-cloudtrail

Q alert-cloudtrail X

Only topics belonging to this account are listed here. All persons and applications subscribed to the selected topic will receive notifications.

Email (endpoints)
neelimarani2398@gmail.com - View in SNS Console ↗

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- Configure actions
- Send a notification to the following SNS topic we can select or create Existing SNS topic
- Save it.

EC2 action

Alarm state trigger
Define the alarm state that will trigger this action.

In alarm
The metric or expression is outside of the defined threshold.

OK
The metric or expression is within the defined threshold.

Insufficient data
The alarm has just started or not enough data is available.

Remove

Take the following action...
Define what will happen to the EC2 instance with the Instance ID i-0677d94f2cd9bf23a when this alarm is triggered.

Recover this instance
You can only recover certain EC2 instance types. See documentation

Stop this instance
You can only stop an instance if it is backed by an EBS volume. AWS will use the existing Service Linked Role (AWSServiceRoleForCloudWatchEvents) to perform this action. Show IAM policy document

Terminate this instance
You will not be able to terminate this instance if termination protection is enabled. AWS will use the existing Service Linked Role (AWSServiceRoleForCloudWatchEvents) to perform this action. Show IAM policy document

Reboot this instance
An instance reboot is equivalent to an operating system reboot. AWS will use the existing Service Linked Role (AWSServiceRoleForCloudWatchEvents) to perform this action. Show IAM policy document

Add EC2 action

CloudShell Feedback Console Mobile App © 2026, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

CloudTrail-CloudWatch-Task

The screenshot shows the 'Add alarm details' step of the CloudWatch 'Create alarm' wizard. On the left, a sidebar lists steps: Step 1 (Specify metric and conditions), Step 2 (Configure actions), Step 3 (Add alarm details), and Step 4 (Preview and create). Step 3 is selected. The main area is titled 'Name and description'. It contains an 'Alarm name' input field with the value 'AWS/EC2 CPUUtilization InstanceId=i-0677d94f2cdffb23a', an 'Alarm description - optional' text area with the message 'CPU utilization crossed threshold and stopping instance.', and a note about Markdown formatting. At the bottom right of the main area is a link to 'View formatting guidelines'.

- Add alarm details Name and description
- Alarm description Edit Message.
- Preview and create

The screenshot shows 'Step 1: Specify metric and conditions' of the CloudWatch 'Create alarm' wizard. The sidebar shows steps 1 through 4, with Step 1 selected. The main area is titled 'Metric'. It includes a 'Graph' section showing a blue line above a red horizontal line at 50, with a note that the alarm triggers when the blue line goes above the red line for 3 datapoints within 15 minutes. To the right, configuration fields show 'Namespace: AWS/EC2', 'Metric name: CPUUtilization', and 'InstanceId: i-0677d94f2cdffb23a'. Below these fields is a 'Details' button.

- The below image shows alarm successfully created alarm AWS/EC2 CPUUtilizations for instance.

The screenshot shows the 'Alarms' page in the CloudWatch console. The sidebar has sections for 'CloudWatch', 'Favorites and recents', and 'Alarms'. Under 'Alarms', there are links for 'In alarm' and 'All alarms'. The main area displays a green banner stating 'Successfully created alarm AWS/EC2 CPUUtilization InstanceId=i-0677d94f2cdffb23a.' Below this is a table titled 'Alarms (1)'. The table has columns for 'Name' (AWS/EC2), 'State' (not visible), 'Last state update (UTC)' (not visible), and 'Cond' (not visible). A search bar and filter options are also present.

CloudTrail-CloudWatch-Task

The screenshot shows the AWS CloudWatch Alarms console. On the left, there's a navigation sidebar with 'CloudWatch' selected. The main area displays a green banner at the top stating 'Successfully created alarm AWS/EC2 CPUUtilization InstancId=i-0677d94f2cdffb23a.' Below this, a table lists the alarm details:

Name	State	Last state update (UTC)	Conditions
AWS/EC2 CPUUtilization InstancId=i-0677d94f2cdffb23a	OK	2026-01-28 05:17:19	CPUUtilization > 50 for 3 datapoints within 15 minutes

At the bottom of the page, there are links for 'CloudShell', 'Feedback', 'Console Mobile App', and standard footer links for 'Privacy', 'Terms', and 'Cookie preferences'.

```
root@ip-172-31-37-203:~#
user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)
$ ssh -i "Neelima-Jenkins.pem" ec2-user@ec2-54-84-119-212.compute-1.amazonaws.co
m
The authenticity of host 'ec2-54-84-119-212.compute-1.amazonaws.com (54.84.119.2
12)' can't be established.
ED25519 key fingerprint is SHA256:lcCn5I1JGTKQJPEVf5+aVt5foqMLuHAc3bMfgMdhzlc.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-84-119-212.compute-1.amazonaws.com' (ED25519)
to the list of known hosts.
,
  #_
  ~\_ ####_      Amazon Linux 2023
  ~\_ #####\_
  ~~ \###|
  ~~  \#/ __| https://aws.amazon.com/linux/amazon-linux-2023
  ~~   V~' , ->
  ~~
  ~~ .-. / \
  ~~ / , / \
  ~~ /m/
[ec2-user@ip-172-31-37-203 ~]$ sudo -i
[root@ip-172-31-37-203 ~]# |
```

- Go to Downloads open Gitbash and Connect ssh
- Change to root user to avoid errors.

CloudTrail-CloudWatch-Task

```
[root@ip-172-31-37-203 ~]# yum update
Amazon Linux 2023 Kernel Livepatch repository   232 kB/s |  30 kB     00:00
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-37-203 ~]# yum install stress-ng
Last metadata expiration check: 0:00:27 ago on Wed Jan 28 05:29:30 2026.
Dependencies resolved.
=====
  Package        Arch      Version       Repository      Size
=====
Installing:
  stress-ng     x86_64    0.15.05-1.amzn2023      amazonlinux    2.3 M
Installing dependencies:
  Judy          x86_64    1.0.5-25.amzn2023.0.3      amazonlinux    153 k
  libbsd         x86_64    0.10.0-7.amzn2023.0.2      amazonlinux    109 k
  lksctp-tools  x86_64    1.0.18-9.amzn2023.0.3      amazonlinux    92 k
Transaction Summary
=====
Install 4 Packages

Total download size: 2.7 M
Installed size: 9.7 M
Is this ok [y/N]: y|
```

- To provide stress to CPU we have to install stress-ng.

Command:yum install stress-ng

```
root@ip-172-31-37-203~:
Last metadata expiration check: 0:00:27 ago on Wed Jan 28 05:29:30 2026.
Dependencies resolved.
=====
  Package        Arch      Version       Repository      Size
=====
Installing:
  stress-ng     x86_64    0.15.05-1.amzn2023      amazonlinux    2.3 M
Installing dependencies:
  Judy          x86_64    1.0.5-25.amzn2023.0.3      amazonlinux    153 k
  libbsd         x86_64    0.10.0-7.amzn2023.0.2      amazonlinux    109 k
  lksctp-tools  x86_64    1.0.18-9.amzn2023.0.3      amazonlinux    92 k
Transaction Summary
=====
Install 4 Packages

Total download size: 2.7 M
Installed size: 9.7 M
Is this ok [y/N]: y
Downloading Packages:
(1/4): libbsd-0.10.0-7.amzn2023.0.2.x86_64.rpm           2.7 MB/s | 109 kB     00:00
(2/4): lksctp-tools-1.0.18-9.amzn2023.0.3.x86_64.rpm      2.0 MB/s | 92 kB     00:00
(3/4): Judy-1.0.5-25.amzn2023.0.3.x86_64.rpm            2.3 MB/s | 153 kB     00:00
(4/4): stress-ng-0.15.05-1.amzn2023.x86_64.rpm          10 MB/s | 2.3 MB     00:00
Total
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing      :
  Installing     : lksctp-tools-1.0.18-9.amzn2023.0.3.x86_64      1/1
  Installing     : libbsd-0.10.0-7.amzn2023.0.2.x86_64        2/4
  Installing     : Judy-1.0.5-25.amzn2023.0.3.x86_64        3/4
  Installing     : stress-ng-0.15.05-1.amzn2023.x86_64       4/4
  Running scriptlet: stress-ng-0.15.05-1.amzn2023.x86_64
  Verifying      : Judy-1.0.5-25.amzn2023.0.3.x86_64        4/4
  Verifying      : libbsd-0.10.0-7.amzn2023.0.2.x86_64        1/4
  Verifying      : lksctp-tools-1.0.18-9.amzn2023.0.3.x86_64      2/4
  Verifying      : stress-ng-0.15.05-1.amzn2023.x86_64       3/4
  Verifying      : stress-ng-0.15.05-1.amzn2023.x86_64       4/4
Installed:
  Judy-1.0.5-25.amzn2023.0.3.x86_64                   libbsd-0.10.0-7.amzn2023.0.2.x86_64
  lksctp-tools-1.0.18-9.amzn2023.0.3.x86_64           stress-ng-0.15.05-1.amzn2023.x86_64
Complete!
[root@ip-172-31-37-203 ~]#
```

- Installation completed

CloudTrail-CloudWatch-Task

```
[root@ip-172-31-37-203 ~]# stress-ng --cpu 4 --cpu-load 100 --timeout 60s
stress-ng: info: [27322] setting to a 60 second run per stressor
stress-ng: info: [27322] dispatching hogs: 4 cpu
stress-ng: info: [27322] successful run completed in 60.15s (1 min, 0.15 secs)
[root@ip-172-31-37-203 ~]# stress-ng --cpu 4 --cpu-load 100 --timeout 350s
stress-ng: info: [27580] setting to a 350 second (5 mins, 50.00 secs) run per s
tressor
stress-ng: info: [27580] dispatching hogs: 4 cpu

Broadcast message from root@ip-172-31-37-203.ec2.internal (Wed 2026-01-28 05:55:20 UTC):
The system will power off now!

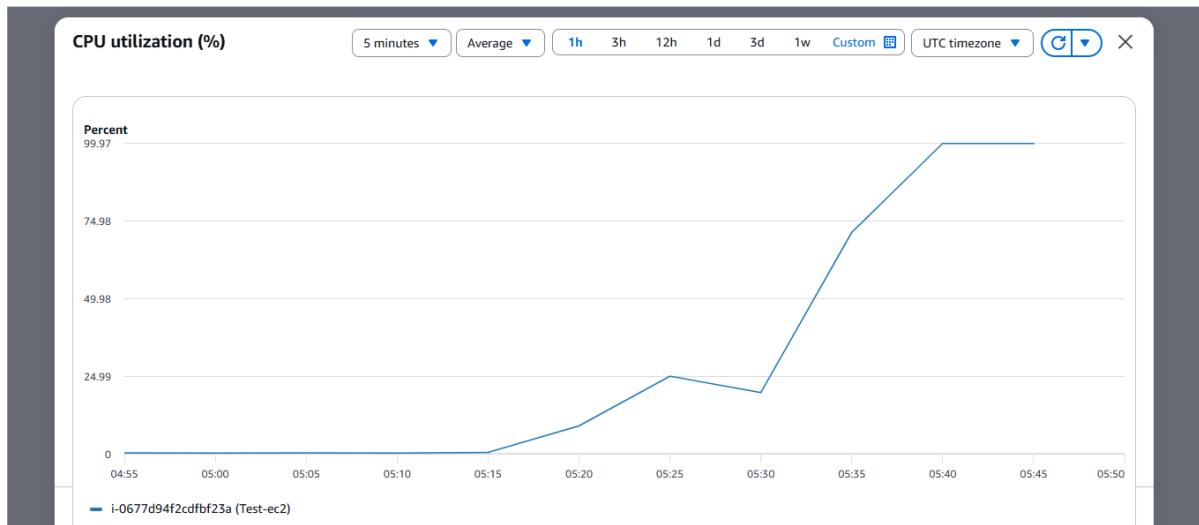
Broadcast message from root@ip-172-31-37-203.ec2.internal (Wed 2026-01-28 05:55:20 UTC):
The system will power off now!

Connection to ec2-54-84-119-212.compute-1.amazonaws.com closed by remote host.
Connection to ec2-54-84-119-212.compute-1.amazonaws.com closed.

user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)
$ |
```

- Command to increase stress as follows

Stress-ng –cpu 4 –cpu-load 100 –timeout 60s



- In CPU monitoring we can see increase of CPU utilization(%)



- The above image shows trigger a notification for our Email through SNS.

CloudTrail-CloudWatch-Task

The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed. The main area displays two instances: 'A' (Stopped) and 'Test-ec2' (Stopping). The 'Test-ec2' instance is selected. Below the table, there are four detailed monitoring charts for the 'Test-ec2' instance:

- CPU utilization (%)**: Percent 25.12 (at 05:30), 12.56 (at 05:00).
- Network in (bytes)**: Bytes 1.12M (at 05:30), 561.71K (at 05:00).
- Network out (bytes)**: Bytes 486.42K (at 05:30), 243.21K (at 05:00).
- Network packets in (count)**: Count 724.2 (at 05:30), 362.1 (at 05:00).

Conclusion:- After reaching threshold the instance stopped automatically and it provides alarm and SNS also.

CloudTrail-CloudWatch-Task

4. Configure CloudWatch monitoring and record the CPU utilization and other metrics of EC2.

Step 1: Verify CloudWatch Agent (default vs detailed)

Default Monitoring (automatic)

By default, EC2 sends **basic metrics every 5 minutes**:

- CPUUtilization
- NetworkIn / NetworkOut
- DiskReadOps / DiskWriteOps
- StatusCheckFailed

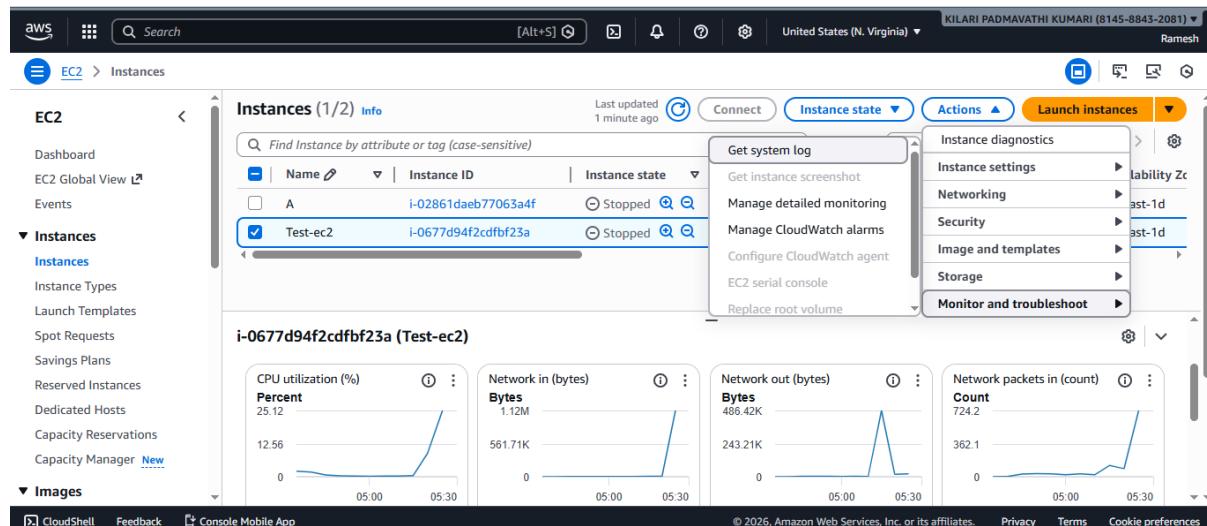
No setup needed

Detailed Monitoring (recommended)

- Records metrics **every 1 minute**.

Enable it:

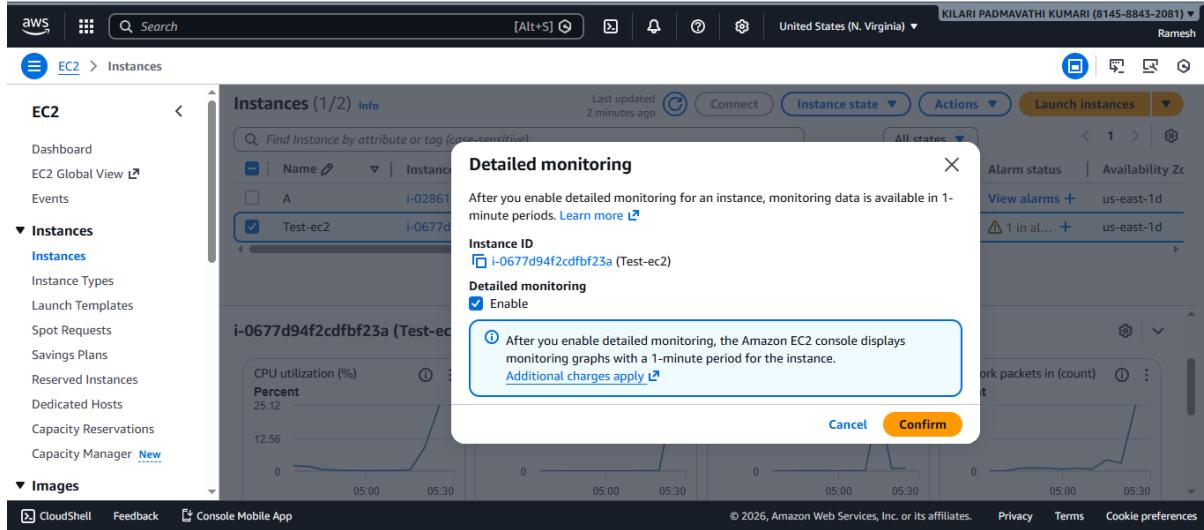
1. Go to **EC2 Console**
2. Select your instance
3. Click **Actions → Monitor and troubleshoot → Manage detailed monitoring**



4. Enable Detailed monitoring

CloudTrail-CloudWatch-Task

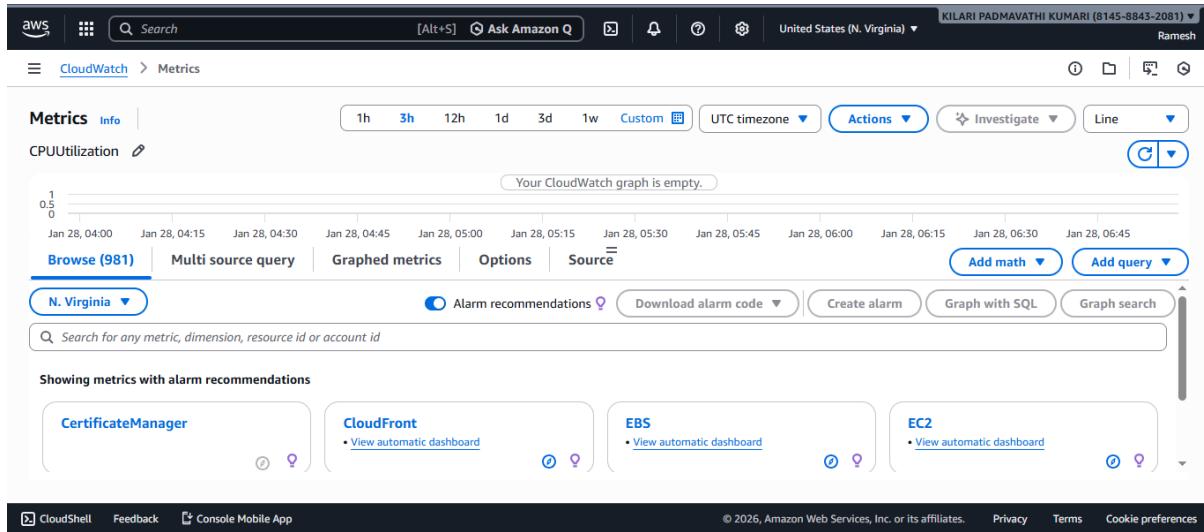
5. Save



Step 2: View CPU Utilization

1. Go to CloudWatch → Metrics

2. Select EC2



3. Choose Per-Instance Metrics

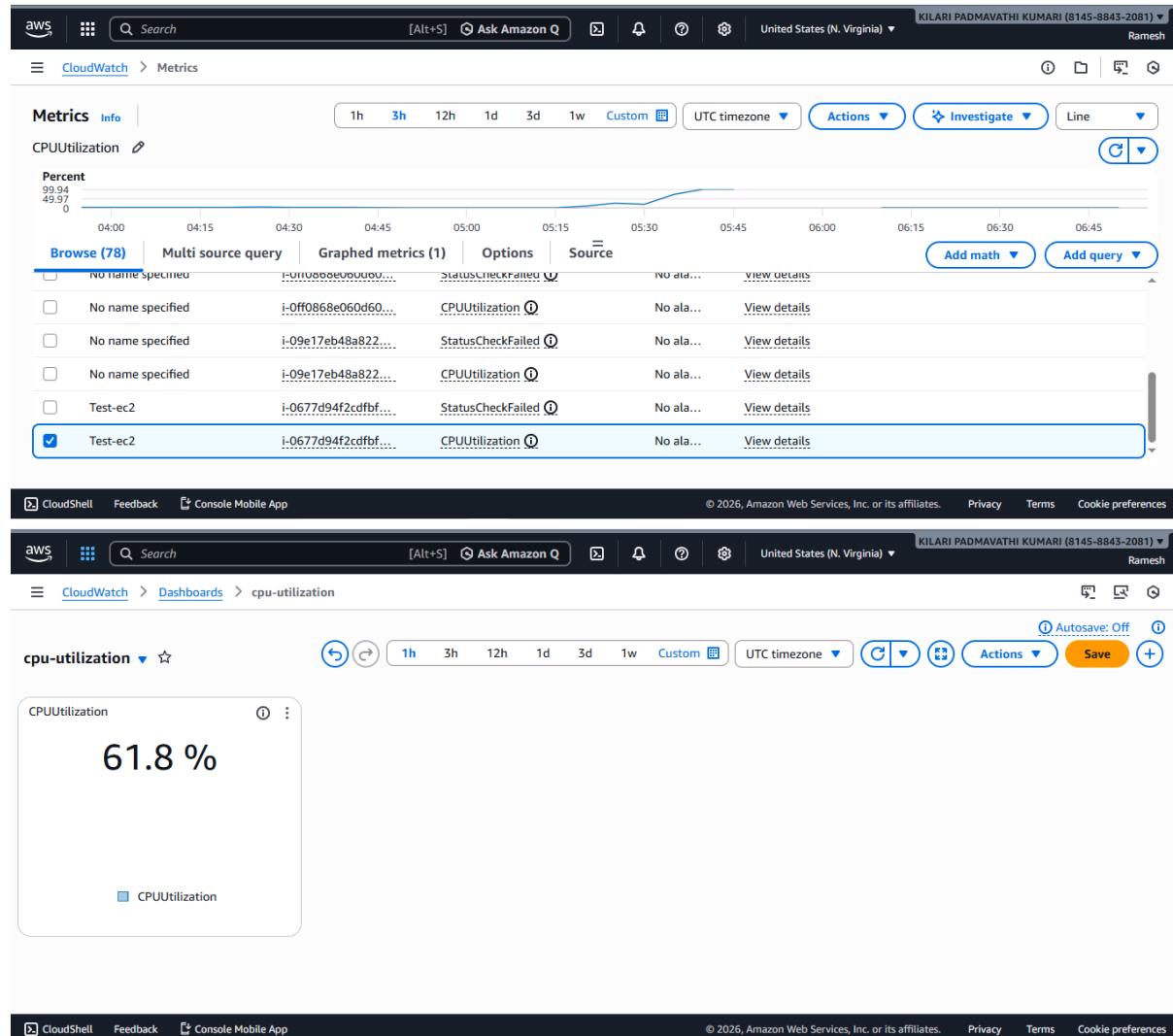
4. Select:

- CPUUtilization

5. Choose your instance

You'll see real-time CPU graphs

CloudTrail-CloudWatch-Task



Conclusion: The above image shows Dashboard for CPUUtilizations.

CloudTrail-CloudWatch-Task

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

- Go AWS console and create instance
- Install tomcat in EC2
- Start tomcat by command sudo systemctl start tomcat.
- Check whether it is running or not.

```
user@DESKTOP-3KHL1RE MINGW64 ~/Downloads (master)
$ ssh -i "Neelima-Jenkins.pem" ec2-user@ec2-54-158-247-79.compute-1.amazonaws.com
Last Login: Wed Jan 28 12:24:04 2026 from 115.96.62.22
[ec2-user@ip-172-31-37-203 ~]$ sudo -i
[ec2-user@ip-172-31-37-203 ~]# sudo systemctl status tomcat
● tomcat.service - Apache Tomcat 10 Web Application Container
   Loaded: loaded (/etc/systemd/system/tomcat.service; enabled; preset: disabled)
     Active: active (running) since Wed 2026-01-28 08:59:10 UTC; 3h 53min ago
       PID: 12327 (java)
      Tasks: 28 (limit: 1120)
     Memory: 121.4M
        CPU: 21.346s
       CGroup: /system.slice/tomcat.service
               └─12327 /usr/bin/java -Djava.util.logging.config.file=/opt/tomcat/conf/logging.properties -Djava.util.logging.manager=org.apache.juli.ClassLoaderLogManager -Djdk.tls.ephemeralDH

Jan 28 08:59:13 ip-172-31-37-203.ec2.internal catalina.sh[12327]: 28-Jan-2026 08:59:13.127 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application director...
Jan 28 08:59:13 ip-172-31-37-203.ec2.internal catalina.sh[12327]: 28-Jan-2026 08:59:13.187 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application director...
Jan 28 08:59:13 ip-172-31-37-203.ec2.internal catalina.sh[12327]: 28-Jan-2026 08:59:13.188 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application director...
Jan 28 08:59:13 ip-172-31-37-203.ec2.internal catalina.sh[12327]: 28-Jan-2026 08:59:13.713 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application director...
Jan 28 08:59:13 ip-172-31-37-203.ec2.internal catalina.sh[12327]: 28-Jan-2026 08:59:13.714 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application director...
Jan 28 08:59:13 ip-172-31-37-203.ec2.internal catalina.sh[12327]: 28-Jan-2026 08:59:13.773 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application director...
Jan 28 08:59:13 ip-172-31-37-203.ec2.internal catalina.sh[12327]: 28-Jan-2026 08:59:13.774 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deploying web application director...
Jan 28 08:59:13 ip-172-31-37-203.ec2.internal catalina.sh[12327]: 28-Jan-2026 08:59:13.814 INFO [main] org.apache.catalina.startup.HostConfig.deployDirectory Deployment of web application director...
Jan 28 08:59:13 ip-172-31-37-203.ec2.internal catalina.sh[12327]: 28-Jan-2026 08:59:13.829 INFO [main] org.apache.coyote.AbstractProtocol.start Starting ProtocolHandler ["http-nio-8080"]
Jan 28 08:59:13 ip-172-31-37-203.ec2.internal catalina.sh[12327]: 28-Jan-2026 08:59:13.873 INFO [main] org.apache.catalina.startup.Catalina.start Server startup in [1705] milliseconds
[lines 1-20/20 (END)]
```

- Go to IAM
- Create user as CW-tomcat-user
- Click on Next.

The screenshot shows the 'Specify user details' step of the IAM user creation wizard. On the left, a sidebar lists steps: Step 1 (radio button selected), Step 2, Set permissions, Step 3, and Review and create. The main area is titled 'User details' and contains a 'User name' field with 'cw-tomcat-user' typed in. Below the field is a note: 'The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , @ _ - (hyphen)'. There is also an optional checkbox for 'Provide user access to the AWS Management Console' with the note: 'In addition to console access, users with SigninLocalDevelopmentAccess permissions can use the same console credentials for programmatic access without the need for access keys.' At the bottom right are 'Cancel' and 'Next' buttons.

CloudTrail-CloudWatch-Task

The screenshot shows the AWS IAM 'Create user' wizard at Step 3: Review and create. The user details section shows a user name 'cw-tomcat-user'. The permissions summary table lists a single policy: 'CloudWatchFullAccess' (AWS managed). The tags section is optional.

- Set permission as cloudwatchfullaccess

The screenshot shows the AWS IAM 'Users' page for the user 'cw-tomcat-user'. The 'Permissions' tab is selected, showing the attached policy 'CloudWatchFullAccess' (AWS managed) directly.

- Click on create user and verify permission policies.

```
user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)
$ aws configure
AWS Access Key ID [*****YT7C]: AKIA33KKBB3I7IJYYT7C
AWS Secret Access Key [*****KSt9]: nRPm+hey+jw+yVoU2ISd7+frv6AgzscOTy
kgKSt9
Default region name [us-west-2]: us-east-1
Default output format [json]: json

user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)
$ aws sts get-caller-identity

{
    "UserId": "AIDA33KKBB3I6AJPYKG5H",
    "Account": "814588432081",
    "Arn": "arn:aws:iam::814588432081:user/cw-tomcat-user"
}
```

- In EC2 we have to configure as Aws configure as shown above.

CloudTrail-CloudWatch-Task

```
user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)
$ ssh -i "Neelima-Jenkins.pem" ec2-user@ec2-54-158-247-79.compute-1.amazonaws.com
  ,      #
  ~\_ #####
  ~~ \#####      Amazon Linux 2023
  ~~ \|###|
  ~~   \#/ __ https://aws.amazon.com/linux/amazon-linux-2023
  ~~     V~' '-->
  ~~     / \
  ~~   /_/
  _/m/
Last login: Wed Jan 28 12:52:51 2026 from 115.96.62.22
[ec2-user@ip-172-31-37-203 ~]$ sudo -i
[root@ip-172-31-37-203 ~]# aws configure
AWS Access Key ID [None]: AKIA33KKBB3I7IJYYT7C
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]: json
[root@ip-172-31-37-203 ~]# aws sts get-caller-identity
{
    "UserId": "AIDA33KKBB3I6AJPYKG5H",
    "Account": "814588432081",
    "Arn": "arn:aws:iam::814588432081:user/cw-tomcat-user"
}
[root@ip-172-31-37-203 ~]# |
```

- Check add configuration by command
Aws sts get-caller-identity

Create Tomcat Metric Script

- sudo vi /opt/tomcat_monitor.sh

```
root@ip-172-31-37-203:~
#!/bin/bash

if systemctl is-active --quiet tomcat; then
    aws cloudwatch put-metric-data \
        --namespace "TomcatService" \
        --metric-name "TomcatRunning" \
        --value 1
else
    aws cloudwatch put-metric-data \
        --namespace "TomcatService" \
        --metric-name "TomcatRunning" \
        --value 0
fi
|
|
|
```

Make executable:

```
sudo chmod +x /opt/tomcat_monitor.sh
```

CloudTrail-CloudWatch-Task

Test Script Manually

```
sudo /opt/tomcat_monitor.sh
```

```
[root@ip-172-31-37-203 ~]# sudo vi /opt/tomcat_monitor.sh
[root@ip-172-31-37-203 ~]# sudo chmod +x /opt/tomcat_monitor.sh
[root@ip-172-31-37-203 ~]# sudo /opt/tomcat_monitor.sh
[root@ip-172-31-37-203 ~]# |
```

No output = **SUCCESS**

Add Cron Job

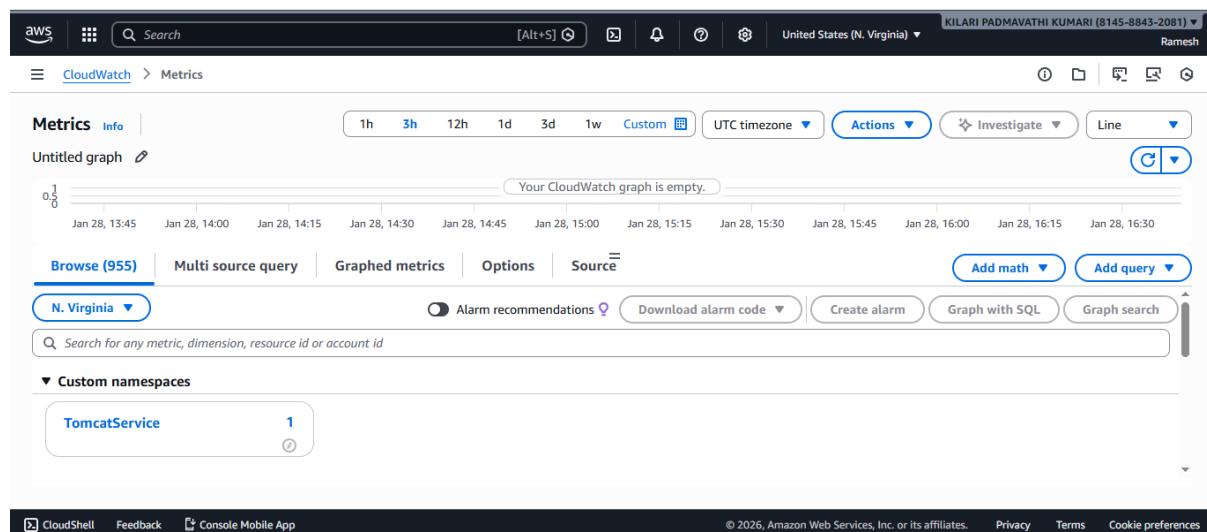
```
sudo crontab -e
```

Add:

```
*/1 * * * * /opt/tomcat_monitor.sh
```

```
[root@ip-172-31-37-203 ~]# sudo crontab -e
crontab: installing new crontab
[root@ip-172-31-37-203 ~]# sudo crontab -l
*/1 * * * * /opt/tomcat_monitor.sh
```

```
[root@ip-172-31-37-203 ~]# |
```



- The above image shows custom namespaces in that we have tomcat service as 1 it means it is running.

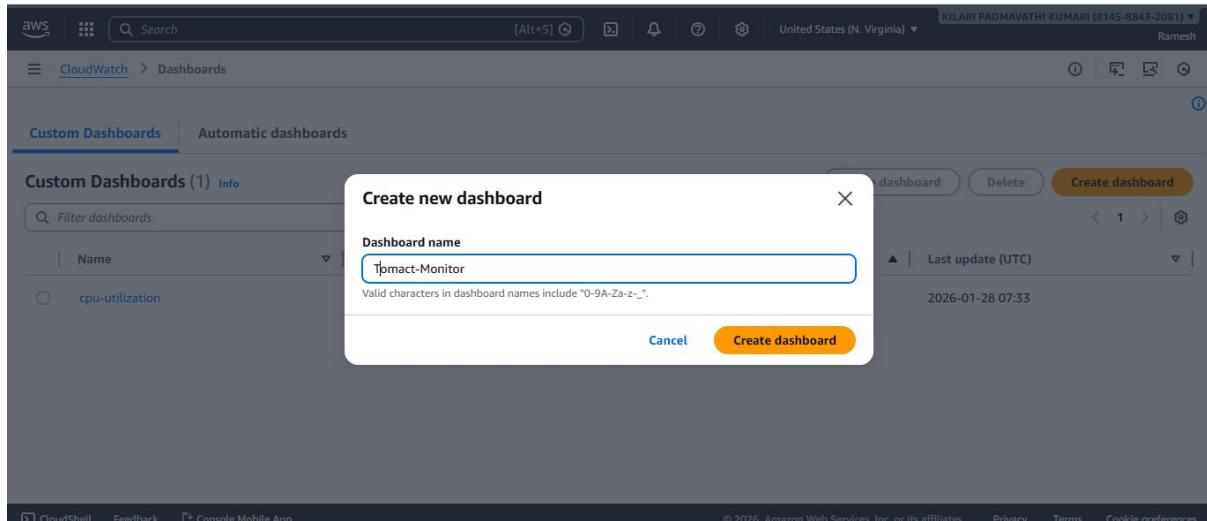
Create CloudWatch Dashboard

- AWS Console → **CloudWatch**
- Left menu → **Dashboards**
- Click **Create dashboard**

CloudTrail-CloudWatch-Task

4. Name: Tomcat-Monitor

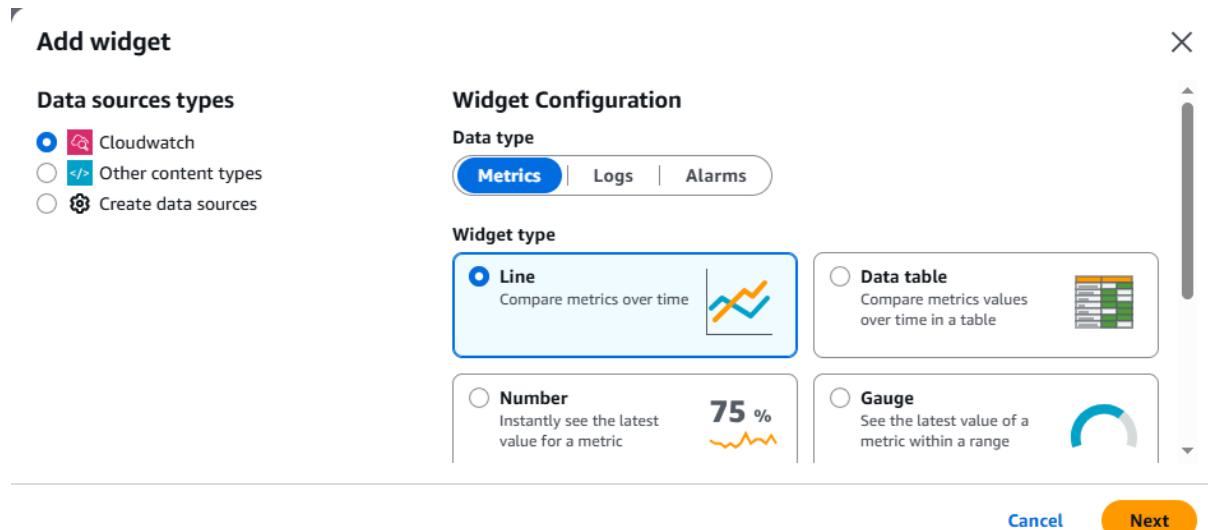
5. Click create dashboard.



6. Click Add widget

7. Choose Line

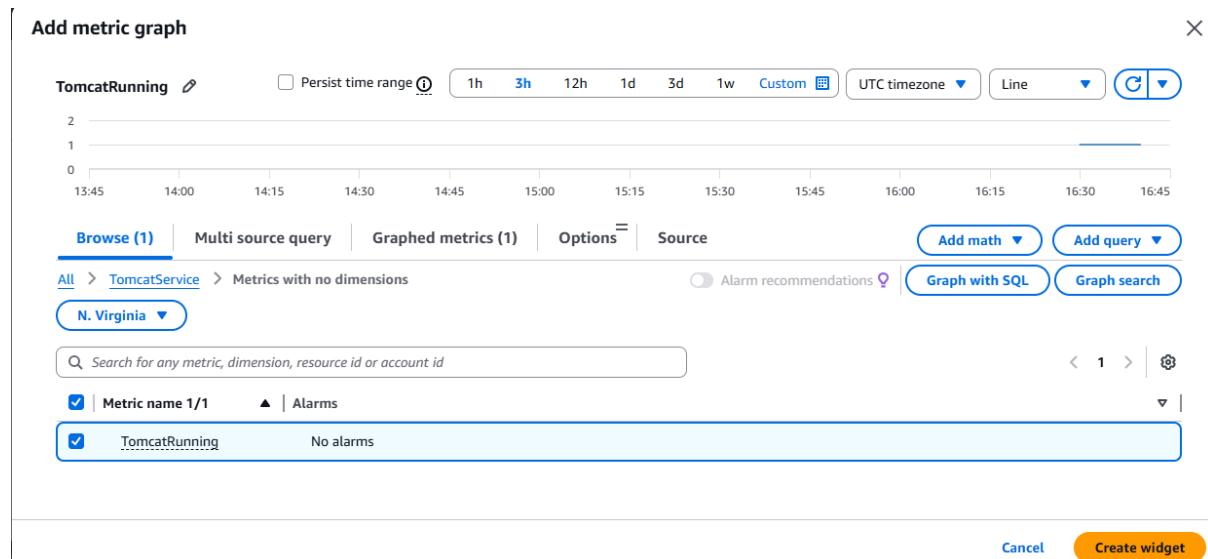
8. Select:



9. Custom namespaces → TomcatService → TomcatRunning

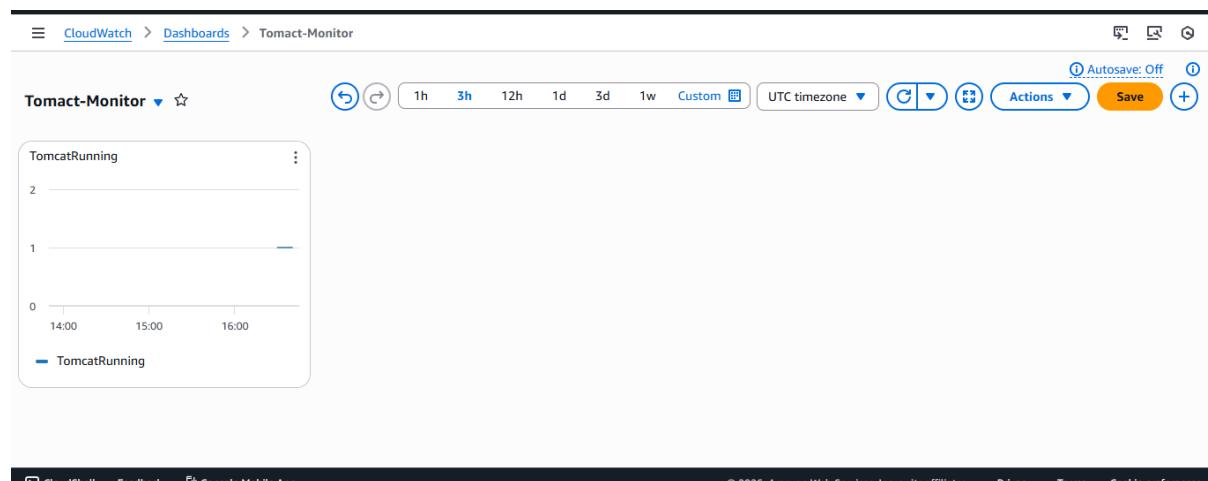
10. Click Create widget

CloudTrail-CloudWatch-Task



Now you'll see Tomcat status graph:

- 1 = Running
- 0 = Down



- Create Alarm
- CloudWatch → Alarms → Create alarm

CloudTrail-CloudWatch-Task

The screenshot shows the AWS CloudWatch Alarms console. On the left sidebar, under the 'Alarms' section, there is a notification: 'Successfully created alarm Monitor-tomcat.' Below it, another message says 'Some subscriptions are pending confirmation'. The main area displays a table of alarms, with one entry for 'Monitor-tomcat'.

Name	State	Last state update (UTC)	Conditions
Monitor-tomcat	Insufficient data	2026-01-28 16:57:26	TomcatRunning < 1 for 1 datapoints within 1 minute

- Create SNS by creating topic

The screenshot shows a confirmation page from the AWS Simple Notification Service. The URL in the browser is 'sns.us-east-1.amazonaws.com/confirmation.html?TopicArn=arn:aws:sns:us-east-1:814588432081:tomcat-alert'. The page title is 'Simple Notification Service'. A green box contains the message: 'Subscription confirmed! You have successfully subscribed. Your subscription's id is: arn:aws:sns:us-east-1:814588432081:tomcat-alert:ca7a9c96-8612-4dac-8ffd-e471a8df604c. If it was not your intention to subscribe, [click here to unsubscribe](#).'

- Add subscription to confirm subscription.

CloudTrail-CloudWatch-Task

The screenshot shows the Amazon SNS console with the 'Topics' section selected. A blue banner at the top indicates a new feature: 'Amazon SNS now supports High Throughput FIFO topics.' Below this, the 'Topic owner' is listed as '814588432081'. The 'Subscriptions' tab is active, showing a single subscription entry:

ID	Endpoint	Status	Protocol
ca7a9c96-8612-4dac-8ffd-e...	neelimarani2398@gmail.com	Confirmed	EMAIL

- To verify
- Stop tomcat service by sudo systemctl stop tomcat.

```
[root@ip-172-31-37-203 ~]# sudo systemctl stop tomcat
[root@ip-172-31-37-203 ~]# |
```

The screenshot shows a Gmail inbox with an email from 'AWS Notifications <no-reply@sns.amazonaws.com>' titled 'ALARM: "Monitor-tomcat" in US East (N. Virginia)'. The email body contains the following information:

You are receiving this email because your Amazon CloudWatch Alarm "Monitor-tomcat" in the US East (N. Virginia) region has entered the ALARM state, because "Threshold Crossed: 1 out of the last 1 datapoints [0.0 (28/01/26 17:01:00)] was less than the threshold (1.0) (minimum 1 datapoint for OK -> ALARM transition)." at "Wednesday 28 January, 2026 17:02:22 UTC".

View this alarm in the AWS Management Console:
<https://us-east-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=us-east-1#alarmsV2:alarm/Monitor-tomcat>

Alarm Details:

- Name: Monitor-tomcat
- Description: Hi!.
- StopTomcat...!
- State Change: OK -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [0.0 (28/01/26 17:01:00)] was less than the threshold (1.0) (minimum 1 datapoint for OK -> ALARM transition).
- Timestamp: Wednesday 28 January, 2026 17:02:22 UTC
- AWS Account: 814588432081
- Alarm Arn: arn:aws:cloudwatch:us-east-1:814588432081:alarm:Monitor-tomcat

Threshold:
- The alarm is in the ALARM state when the metric is LessThanThreshold 1.0 for at least 1 of the last 1 period(s) of 60 seconds.

Conclusion: SNS trigger a Email which in created by cloudwatch alarm.

CloudTrail-CloudWatch-Task

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

Monitor **Nginx service** on EC2 and:

- Push status to **CloudWatch Custom Metric**
- Show on **Dashboard**
- Send **Email alert** if Nginx stops

```
user@DESKTOP-3KH1IRE MINGW64 ~/Downloads (master)
$ ssh -i "Neelima-Jenkins.pem" ec2-user@ec2-54-87-109-74.compute-1.amazonaws.com
The authenticity of host 'ec2-54-87-109-74.compute-1.amazonaws.com (54.87.109.74)' can't be established.
ED25519 key fingerprint is SHA256:jMXHPDUx2GIYL34hgK8QEyWY3CMIXGjw/Puf8g7UY.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-87-109-74.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

,
  #
  ~\_\_#####
  ~~\_\#\#\#\_
  ~~ \#\#\#
  ~~ \#\#
  ~~ \#/ ___
  ~~ V~' '-->
  ~~
  ~~ .-.
  ~~ /_/
  ~~ /m/'

[ec2-user@ip-172-31-40-146 ~]$ |
```

Step 1: Install & Start Nginx (if not already)

```
sudo yum install -y nginx
```

```
sudo systemctl start nginx
```

```
sudo systemctl enable nginx
```

Check:

```
systemctl status nginx
```

```
Complete!
[root@ip-172-31-40-146 ~]# sudo systemctl start nginx
[root@ip-172-31-40-146 ~]# 
[root@ip-172-31-40-146 ~]# sudo systemctl enable nginx
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
[root@ip-172-31-40-146 ~]# systemctl status nginx
● nginx.service - The nginx HTTP and reverse proxy server
   Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: disabled)
   Active: active (running) since Wed 2026-01-28 17:13:40 UTC; 53s ago
     Main PID: 26159 (nginx)
        Tasks: 2 (limit: 1120)
       Memory: 2.5M
          CPU: 59ms
        CGroup: /system.slice/nginx.service
                  └─26159 "nginx: master process /usr/sbin/nginx"
                     ├─26163 "nginx: worker process"

Jan 28 17:13:40 ip-172-31-40-146.ec2.internal systemd[1]: Starting nginx.service - The nginx HTTP and reverse proxy server...
Jan 28 17:13:40 ip-172-31-40-146.ec2.internal nginx[26136]: nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
Jan 28 17:13:40 ip-172-31-40-146.ec2.internal nginx[26136]: nginx: configuration file /etc/nginx/nginx.conf test is successful
Jan 28 17:13:40 ip-172-31-40-146.ec2.internal systemd[1]: Started nginx.service - The nginx HTTP and reverse proxy server.
[root@ip-172-31-40-146 ~]# |
```

CloudTrail-CloudWatch-Task

Step 2: Create Nginx Metric Script

```
sudo vi /opt/nginx_monitor.sh
```

```
root@ip-172-31-40-146:~  
#!/bin/bash  
  
if systemctl is-active --quiet nginx; then  
    aws cloudwatch put-metric-data \  
        --namespace "NginxService" \  
        --metric-name "NginxRunning" \  
        --value 1  
else  
    aws cloudwatch put-metric-data \  
        --namespace "NginxService" \  
        --metric-name "NginxRunning" \  
        --value 0  
fi  
~
```

Step 3: Test Script

```
[root@ip-172-31-40-146 ~]# sudo /opt/nginx_monitor.sh  
Unable to locate credentials. You can configure credentials by running "aws login".  
[root@ip-172-31-40-146 ~]# aws configure  
AWS Access Key ID [None]: AKIA33KKBB3I7IJYYT7C  
AWS Secret Access Key [None]: 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  
Default region name [None]: us-east-1  
Default output format [None]: json  
[root@ip-172-31-40-146 ~]# aws sts get-caller-identity  
{  
    "UserId": "AIDA33KKBB3I6AJPYKG5H",  
    "Account": "814588432081",  
    "Arn": "arn:aws:iam::814588432081:user/cw-tomcat-user"  
}  
[root@ip-172-31-40-146 ~]# sudo /opt/nginx_monitor.sh  
[root@ip-172-31-40-146 ~]# |
```

Step 4: Add Cron Job

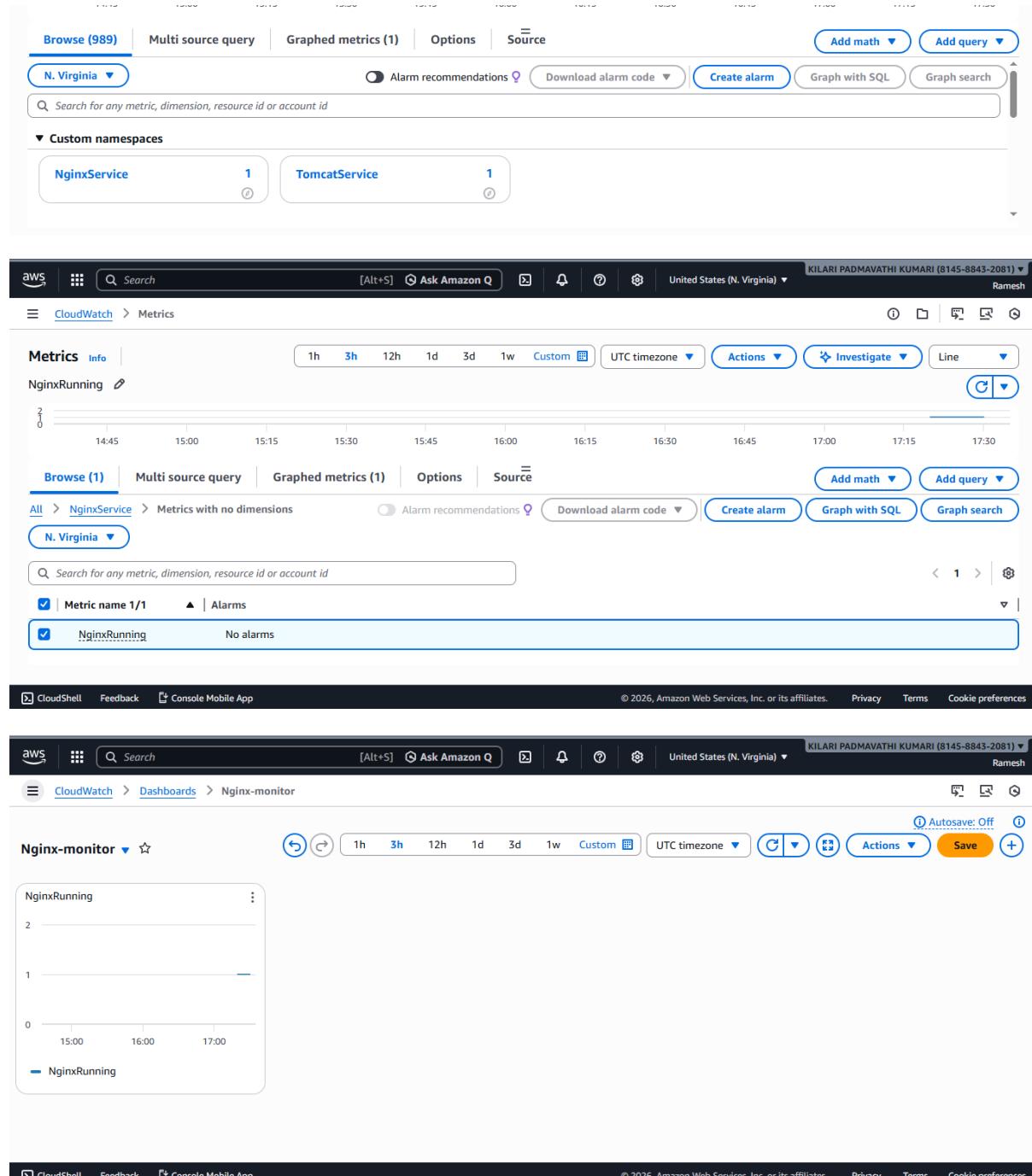
```
[root@ip-172-31-40-146 ~]# sudo crontab -e  
no crontab for root - using an empty one  
crontab: installing new crontab  
[root@ip-172-31-40-146 ~]# sudo crontab -l  
*/1 * * * * /opt/nginx_monitor.sh  
  
[root@ip-172-31-40-146 ~]# |
```

CloudTrail-CloudWatch-Task

Step 5: Verify in CloudWatch

AWS Console → CloudWatch → Metrics

Custom namespaces → NginxService → NginxRunning



- Go Dashboard → Nginx Monitor
- Click on save.

CloudTrail-CloudWatch-Task

The screenshot shows the AWS CloudWatch Alarms interface. On the left sidebar, under the 'Alarms' section, there are two items: 'Nginx status' and 'Monitor-tomcat'. The main area displays a summary message: 'Successfully created alarm Nginx status.' and 'Some subscriptions are pending confirmation'. Below this, a table lists the two alarms:

Name	State	Last state update (UTC)	Conditions
Nginx status	Insufficient data	2026-01-28 17:41:33	NginxRunning < 1 for 1 datapoints within 5 minutes
Monitor-tomcat	Insufficient data	2026-01-28 17:14:22	TomcatRunning < 1 for 1 datapoints within 1 minute

- Create Alarm for Nginx.
- Add SNS topic and confirm subscription.

The screenshot shows an SNS confirmation email from sns.us-east-1.amazonaws.com. The subject is 'Subscription confirmed!'. The body of the email contains the following text:

You have successfully subscribed.
Your subscription's id is:
arn:aws:sns:us-east-1:814588432081:Nginx-Monitoring:f7a70a50-9932-49e5-8a2f-82ada1ff4ea0
If it was not your intention to subscribe, [click here to unsubscribe](#).

Test

Stop nginx:

sudo systemctl stop nginx

```
[root@ip-172-31-40-146 ~]# sudo systemctl stop nginx
[root@ip-172-31-40-146 ~]# |
```

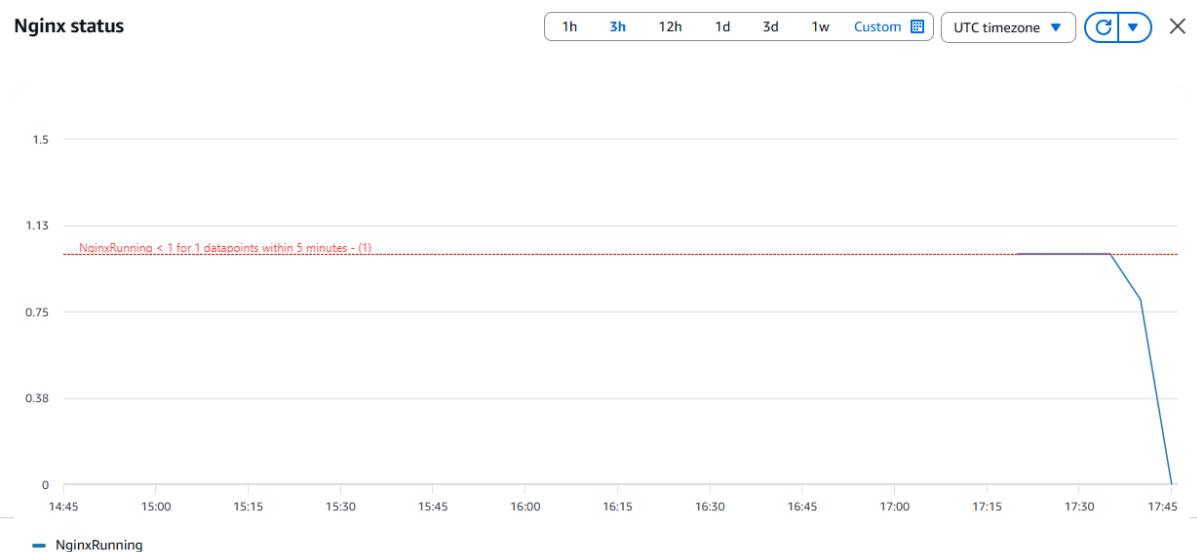
- Alarm → ALARM

CloudTrail-CloudWatch-Task

- Email received

The screenshot shows a Gmail inbox with 12 messages. A message from 'AWS Notifications' is selected, titled 'ALARM: "Nginx status" in US East (N. Virginia)'. The email content details an alarm state change for the 'Nginx status' metric in the US East (N. Virginia) region. The alarm was triggered due to a threshold cross-over, where one out of the last 1 datapoints (0.8 at 28/01/26 17:40:00) was less than the threshold (1.0). The alarm was created on Wednesday, January 28, 2026, at 17:45:18 UTC. The alarm name is 'Nginx status', and it is associated with the AWS account 814588432081. The alarm arm identifier is 'arn:aws:cloudwatch:us-east-1:814588432081:alarm:Nginx%20status'. The threshold is set to 1.0, and the alarm is in the ALARM state when the metric is LessThanThreshold 1.0 for at least 1 of the last 1 period(s) of 300 seconds.

- Dashboard shows 0



Conclusion: Dashboard and monitor the Nginx service to send the alert if Nginx is not running.