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

Let’s set up **AWS CloudTrail** to monitor account activity and store events in **S3** and **CloudWatch Logs** — step-by-step and clearly 👇

**Step 1: Open CloudTrail Console**

1. Sign in to **AWS Management Console**.
2. Go to **CloudTrail** → https://console.aws.amazon.com/cloudtrail

**Step 2: Create a New Trail**

1. Click **“Create trail”**.
2. Under **Trail name**, give a name — e.g. MyCloudTrail.

**Step 3: Choose an S3 Bucket (for Event Storage)**

1. Under **Storage location**, choose:
   * **Create new S3 bucket** → enter a unique name like my-cloudtrail-logs-bucket.
   * Or **Use existing bucket** if you already have one.
2. CloudTrail will automatically create a folder structure like:  
   s3://my-cloudtrail-logs-bucket/AWSLogs/<account-id>/CloudTrail/...

✅ **Tip:** Make sure S3 bucket versioning and encryption are enabled for security.

**Step 4: Enable CloudWatch Logs Integration**

1. Under **CloudWatch Logs**, toggle **“Enabled”**.
2. Select or create a **CloudWatch Log group** — e.g. /aws/cloudtrail/MyTrailLogs.
3. Choose an **IAM role** for CloudTrail to send logs to CloudWatch.

If you don’t have one:

* + Click **“Create new role”**
  + It will automatically create a role named like:  
    CloudTrail\_CloudWatchLogs\_Role

This role allows CloudTrail to publish log events to CloudWatch.

**Step 5: Select Events to Record**

1. Under **Event type**, select:
   * **Management events** ✅ (default)
   * Optionally add **Data events** (e.g., S3 object-level API activity)
   * Optionally add **Insights events** (detect unusual activity)

**Step 6: Apply to All Regions (Recommended)**

1. Under **Apply trail to all regions**, choose **Yes** ✅  
   → This ensures CloudTrail monitors all regions.

**Step 7: Review and Create**

1. Review all details.
2. Click **Create trail**.

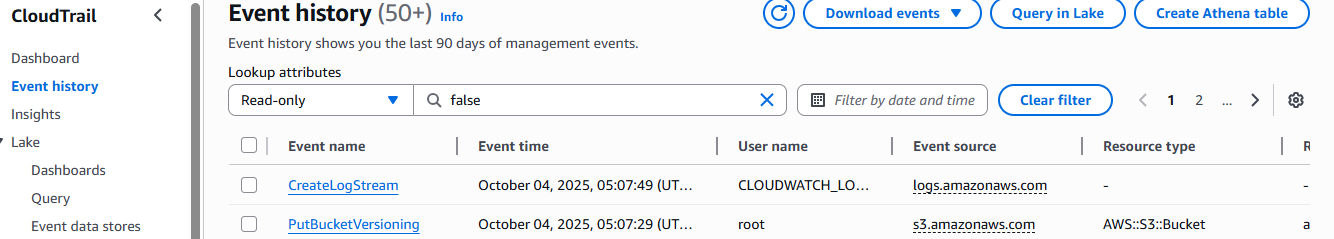
CloudTrail will now:

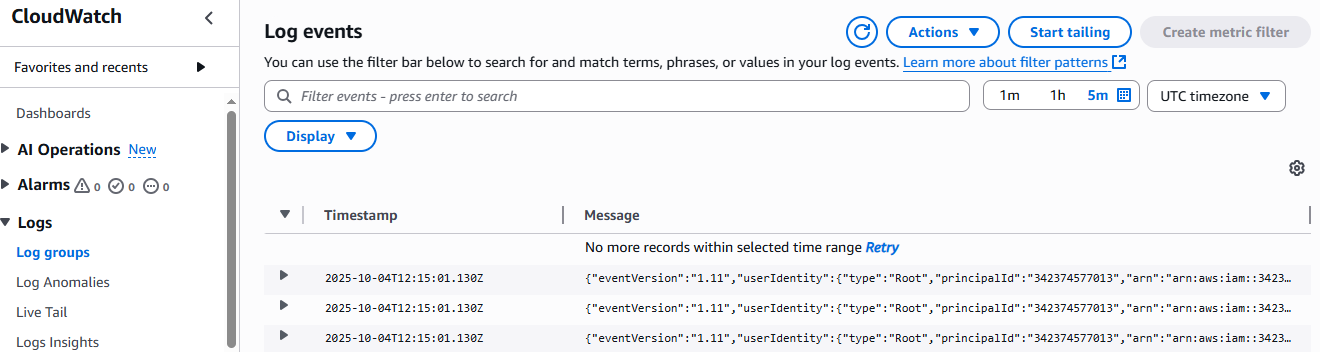
* Record all management (and optionally data) events.
* Store them in your S3 bucket.
* Send copies to CloudWatch Logs for real-time monitoring.

**Verification**

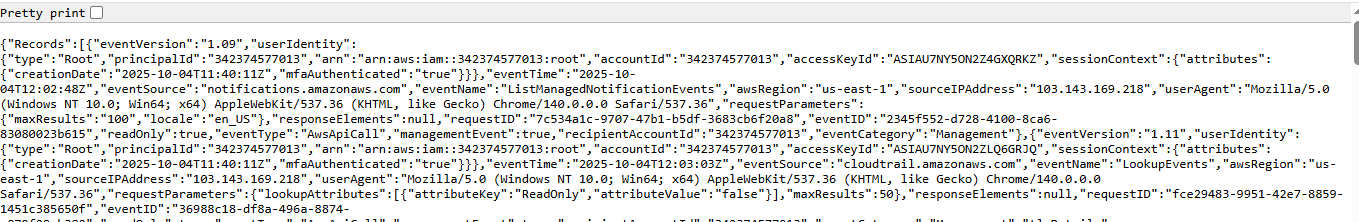
* In **CloudTrail → Event history**, you can view recent activity.
* In **CloudWatch → Log groups**, check the log group you created for incoming events.
* In **S3 → Bucket**, you’ll see folders like:
* AWSLogs/
* <your-account-id>/
* CloudTrail/
* us-east-1/

us-west-2/





**S3:**



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

Perfect 👍 Let’s now enable **Amazon SNS (Simple Notification Service)** for your **CloudTrail** so it can send **email alerts** when new events occur.

Below is a **clear step-by-step guide** 👇

## Step 1: Open CloudTrail Console

1. Go to **AWS Management Console → CloudTrail**  
   👉 https://console.aws.amazon.com/cloudtrail
2. Select your existing trail (for example MyCloudTrail).

## Step 2: Enable SNS Notification

1. In the **Trail details** page, click **Edit**.
2. Scroll down to **“SNS notification delivery”** (in the **Storage location** section).
3. Check ✅ **“Enable SNS notification for new log file delivery”**.

## Step 3: Create or Choose an SNS Topic

1. Choose **Create a new SNS topic** (recommended).
   * Topic name example: CloudTrail-Email-Alerts
2. CloudTrail will automatically configure permissions to publish notifications to that SNS topic.

## Step 4: Subscribe Your Email to the SNS Topic

1. Go to **Amazon SNS → Topics**  
   👉 https://console.aws.amazon.com/sns/v3/home#/topics
2. Select your topic (e.g. CloudTrail-Email-Alerts).
3. Click **“Create subscription”**.
4. Under:
   * **Protocol:** select Email
   * **Endpoint:** enter your **email address** (e.g. you@example.com)
5. Click **Create subscription**.

## Step 5: Confirm Subscription

1. Check your **email inbox**.
2. You’ll get a message from **AWS Notifications**.
3. Click the **“Confirm subscription”** link inside the email.

✅ Once confirmed — your email will start receiving alerts whenever CloudTrail delivers new log files.

## (Optional) Step 6: Verify SNS Works

You can test SNS:

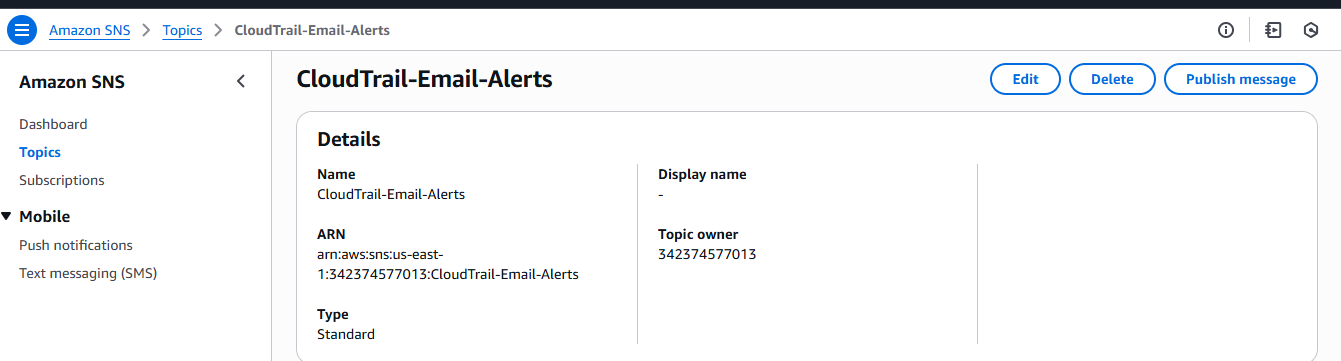
1. Go to the SNS topic.
2. Click **Publish message**.
3. Type a test message and click **Publish**.
4. You should receive an email notification immediately.

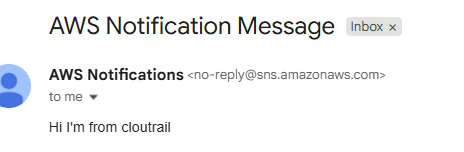
## Example CloudTrail + SNS Architecture

CloudTrail → S3 Bucket (stores logs)

→ CloudWatch Logs (for monitoring)

→ SNS Topic → Email Alert





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

## Step 1: Open CloudWatch Console

Go to:  
👉 https://console.aws.amazon.com/cloudwatch

## ⚙️ Step 2: Enable Detailed Monitoring (Optional but Recommended)

By default, **CloudWatch collects EC2 metrics every 5 minutes**.  
To get **1-minute intervals**, enable **Detailed Monitoring**:

1. Go to **EC2 → Instances**.
2. Select your instance.
3. Click **Actions → Monitor and troubleshoot → Manage detailed monitoring**.
4. Check ✅ **Enable detailed monitoring**, then **Save**.

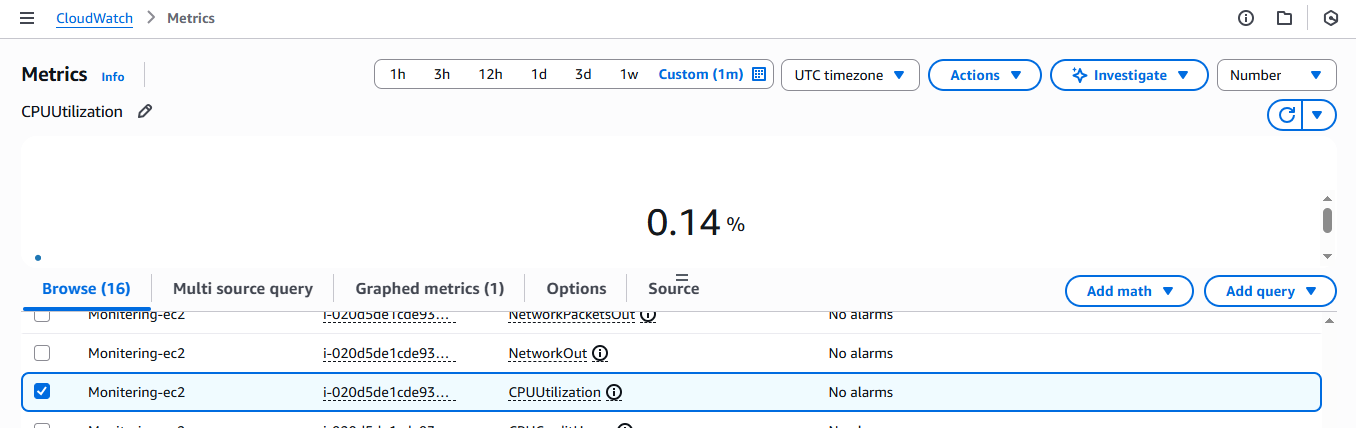
## 📊 Step 3: View CPU and Other Default Metrics

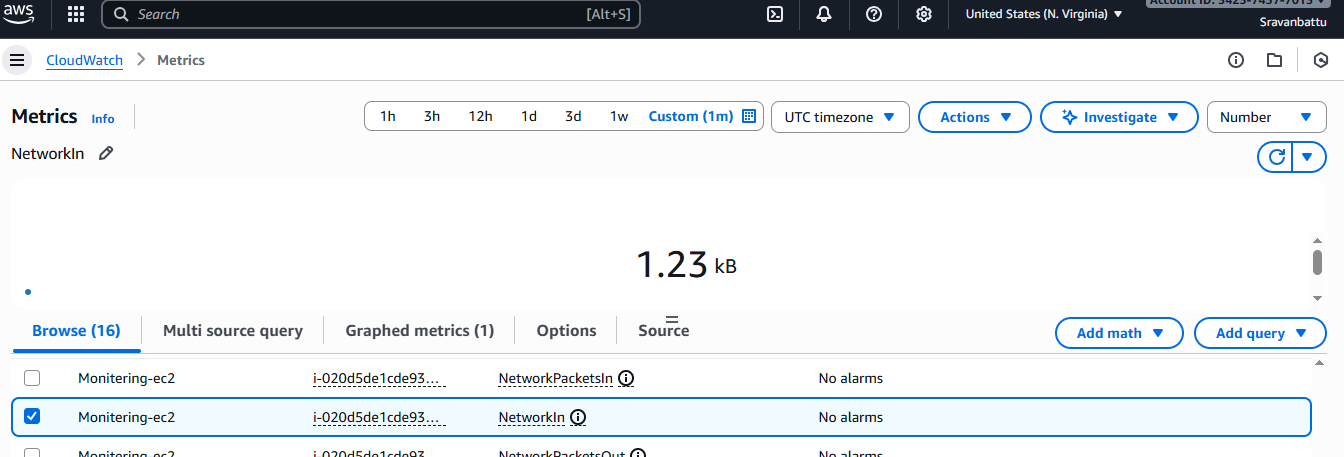
CloudWatch automatically collects these EC2 metrics:

| **Metric Name** | **Description** |
| --- | --- |
| CPUUtilization | % CPU used |
| NetworkIn | Bytes received |
| NetworkOut | Bytes sent |
| DiskReadBytes | Bytes read from disk |
| DiskWriteBytes | Bytes written to disk |
| StatusCheckFailed | Fails if EC2 instance or system check fails |

To view them:

1. In **CloudWatch Console → Metrics**
2. Choose **All metrics → EC2 → Per-Instance Metrics**
3. Select your instance ID → You’ll see charts for CPU, Network, Disk, etc.





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

Perfect 👍 — let’s create a **CloudWatch Alarm** so you get an **email alert if CPU utilization exceeds 70%**.

I’ll give you the **exact step-by-step instructions**.

## **Step 1 — Create an SNS Topic for Email Alerts**

This is where CloudWatch will send the alarm notification.

1. Go to **Amazon SNS Console**:  
   👉 https://console.aws.amazon.com/sns/v3/home
2. Click **Topics → Create topic**.
   * **Topic type:** Standard
   * **Name:** HighCPUAlarmTopic
3. Click **Create topic**.

## **Step 2 — Subscribe Your Email**

1. In the topic details, click **Create subscription**.
2. **Protocol:** Email
3. **Endpoint:** Your email address
4. Click **Create subscription**.

✅ You will receive an email asking to confirm the subscription — click the confirmation link.

## **Step 3 — Create the CloudWatch Alarm**

1. Go to **CloudWatch Console**:  
   👉 https://console.aws.amazon.com/cloudwatch
2. On the left menu → **Alarms → All alarms → Create alarm**.
3. Click **Select metric** → **EC2 → Per-Instance Metrics → CPUUtilization**.
4. Select your EC2 instance and click **Select metric**.

### ****Step 4 — Configure the Alarm****

* **Statistic:** Average
* **Period:** 1 minute (if detailed monitoring enabled)
* **Threshold type:** Static
* **Condition:** Greater than
* **Value:** 70

Example:

Whenever CPUUtilization is > 70% for 1 consecutive period(s) of 1 minute

Click **Next**.

### ****Step 5 — Configure Actions****

* **Select an existing SNS topic**: HighCPUAlarmTopic (created in Step 1)
* Click **Next**.

### ****Step 6 — Name and Create Alarm****

* **Name:** HighCPUUtilizationAlarm
* **Description:** “Alarm if CPU > 70% for 1 minute”
* Click **Create alarm**.

## **Step 7 — Verify**

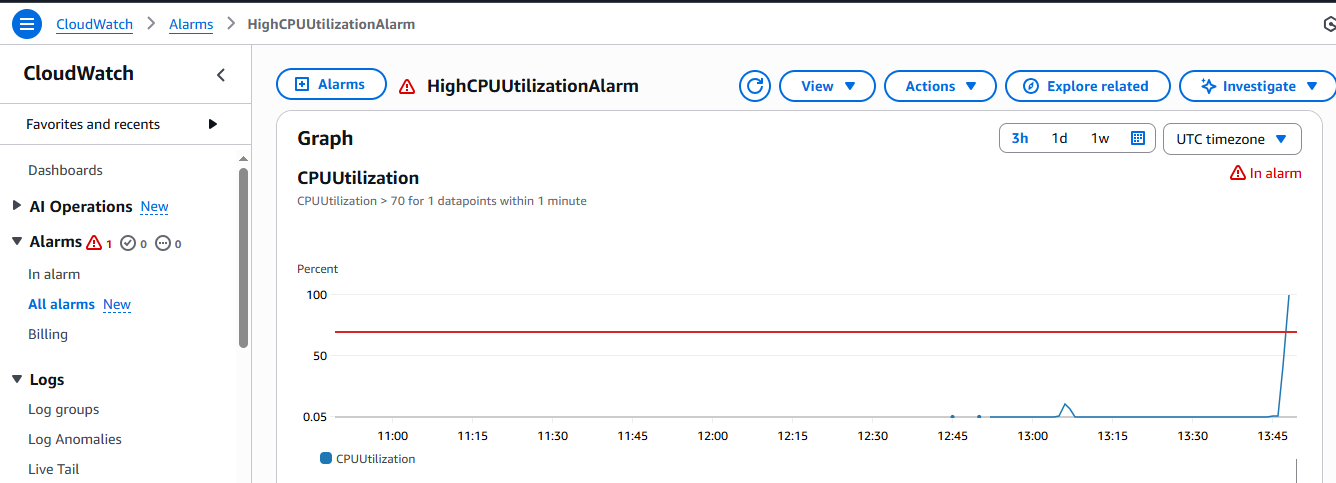
* CloudWatch will now monitor your EC2 CPU utilization.
* If CPU exceeds 70% → SNS will send an email to your subscribed address.

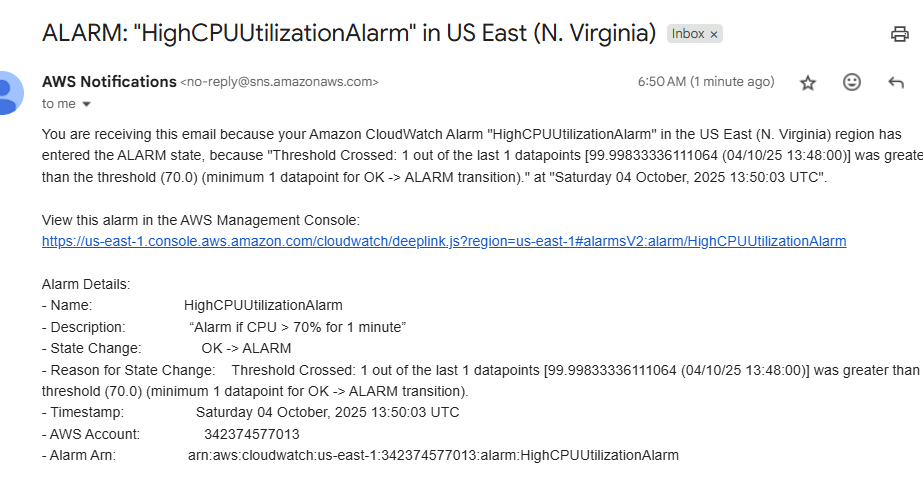
📩 **Extra Tip:**  
You can test the alarm by running a CPU stress test:

sudo yum install stress -y

stress --cpu 2 --timeout 300

That will increase CPU usage for 5 minutes and trigger the alarm.





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

### Step 2: Download from Apache archive (this URL works)

Use this archived version link (Tomcat **9.0.83**, a stable long-term version):

sudo wget https://archive.apache.org/dist/tomcat/tomcat-9/v9.0.83/bin/apache-tomcat-9.0.83.tar.gz

✅ You should see output like:

Saving to: ‘apache-tomcat-9.0.83.tar.gz’

apache-tomcat-9.0.83.tar.gz 100%[==============================>] 10.6M 3.3MB/s in 3.2s

### 🧩 Step 3: Extract and Set Up

sudo tar -xvzf apache-tomcat-9.0.83.tar.gz

sudo mv apache-tomcat-9.0.83 tomcat

### 🧩 Step 4: Start Tomcat

sudo sh /opt/tomcat/bin/startup.sh

Expected output:

Using CATALINA\_BASE: /opt/tomcat

Using CATALINA\_HOME: /opt/tomcat

Tomcat started.

### 🧩 Step 5: Verify and Access

Check it’s running:

ps aux | grep tomcat

### Install cronie

sudo dnf install cronie -y

### 🧩 Step 2: Enable and start the cron service

sudo systemctl enable crond

sudo systemctl start crond

You can confirm it’s running:

sudo systemctl status crond

You should see:

Active: active (running)

### 🧩 Step 3: Now open the crontab editor

crontab -e

✅ It should now open in vi or nano.

Add this line to run your Tomcat check script every 5 minutes:

\*/5 \* \* \* \* /opt/tomcat\_check.sh

#### **Edit your Tomcat check script**

Open:

sudo vi /opt/tomcat\_check.sh

Replace the contents with:

#!/bin/bash

SERVICE="tomcat"

REGION="us-east-1" # Change to your AWS region

if ps aux | grep -v grep | grep $SERVICE > /dev/null

then

STATUS=1

else

STATUS=0

fi

aws cloudwatch put-metric-data \

--metric-name TomcatStatus \

--namespace "TomcatMonitoring" \

--value $STATUS \

--region $REGION

#### **2️⃣ Make it executable**

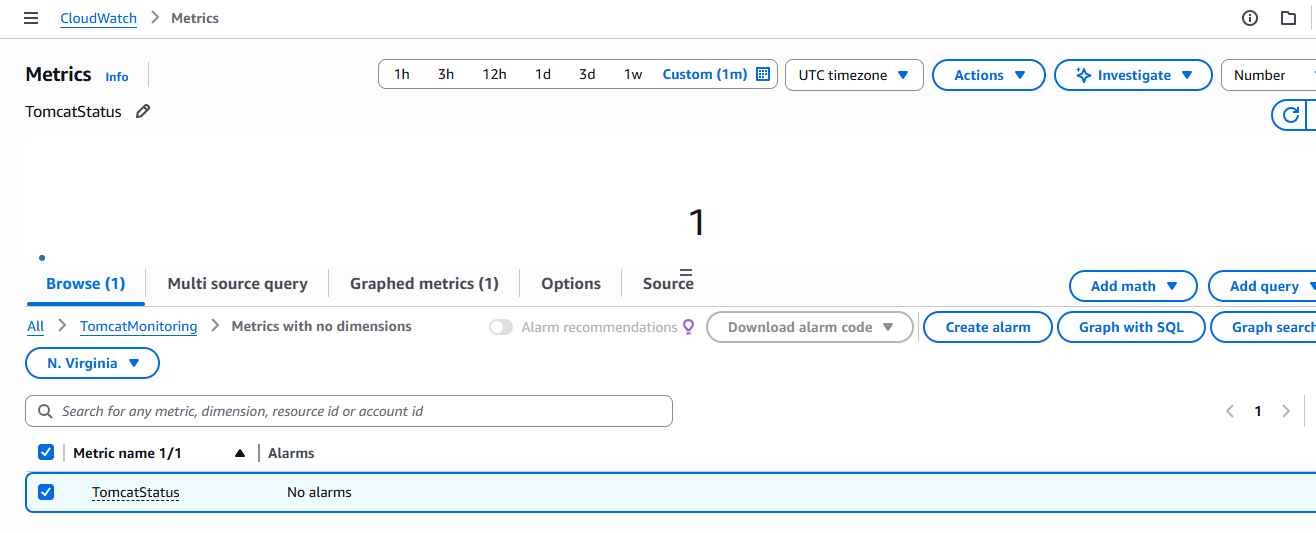
sudo chmod +x /opt/tomcat\_check.sh

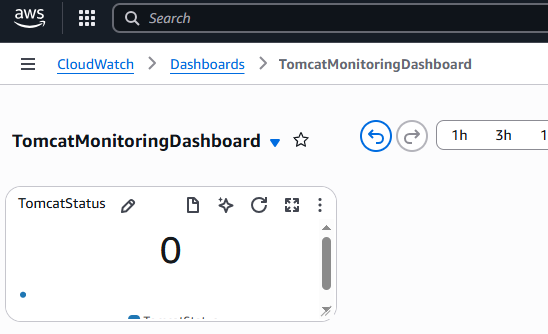
#### **3️⃣ Test it manually**

/opt/tomcat\_check.sh

#### **Check CloudWatch**

Go to AWS Console → CloudWatch → Metrics → **Custom Namespaces** → TomcatMonitoring.  
You should now see TomcatStatus





## **Step 1 — Go to CloudWatch Dashboards**

1. Log in to AWS Console.
2. Search **CloudWatch** in the search bar.
3. Click **Dashboards** in the left menu.
4. Click **Create dashboard**.

## **Step 2 — Name the Dashboard**

* Give it a name, e.g.:

TomcatMonitoringDashboard

* Click **Create dashboard**.

## **Step 3 — Add Widget**

* Choose **Line** or **Number** widget.  
  (Number widget is good for seeing running/stopped state instantly, Line widget is good for trends over time).
* Click **Configure**.

## **Step 4 — Select Metric**

* Click **Add metric**.
* Go to:

Custom Namespaces → TomcatMonitoring → TomcatStatus

* Select the metric.
* Click **Create widget**.

## **Step 5 — Configure the Widget**

* For a **Line graph**, set:
  + Statistic: Average
  + Period: 1 minute
* For a **Number widget**, just select the metric and it will show the latest value (1 or 0).

## **Step 6 — Save Dashboard**

* Click **Save dashboard**