1. Enable CloudTrail monitoring and store the events in s3 and CloudWatch log events.

**Step 1: Sign in to AWS Management Console**

* Navigate to: <https://console.aws.amazon.com/cloudtrail/>

**Step 2: Create or Use an S3 Bucket**

CloudTrail needs an S3 bucket to store logs.

**Option A: Create a New S3 Bucket**

1. Go to **S3 Console**: <https://s3.console.aws.amazon.com/s3>
2. Click **Create bucket**.
3. Enter a unique **Bucket name** (e.g., my-cloudtrail-logs-bucket).
4. Choose the region.
5. Keep **Object Ownership** as default (ACLs disabled).
6. Enable **Bucket Versioning** (recommended).
7. Click **Create bucket**.

Note: Make sure that the bucket policy allows CloudTrail to write logs.

**Option B: Use an Existing S3 Bucket**

* Make sure you add the appropriate bucket policy to allow CloudTrail access.

**📜 Sample Bucket Policy:**

Replace your-bucket-name and your-account-id.

json

CopyEdit

{

"Version": "2012-10-17",

"Statement": [

{

"Sid": "AWSCloudTrailWrite",

"Effect": "Allow",

"Principal": {

"Service": "cloudtrail.amazonaws.com"

},

"Action": "s3:PutObject",

"Resource": "arn:aws:s3:::your-bucket-name/AWSLogs/your-account-id/\*",

"Condition": {

"StringEquals": {

"s3:x-amz-acl": "bucket-owner-full-control"

}

}

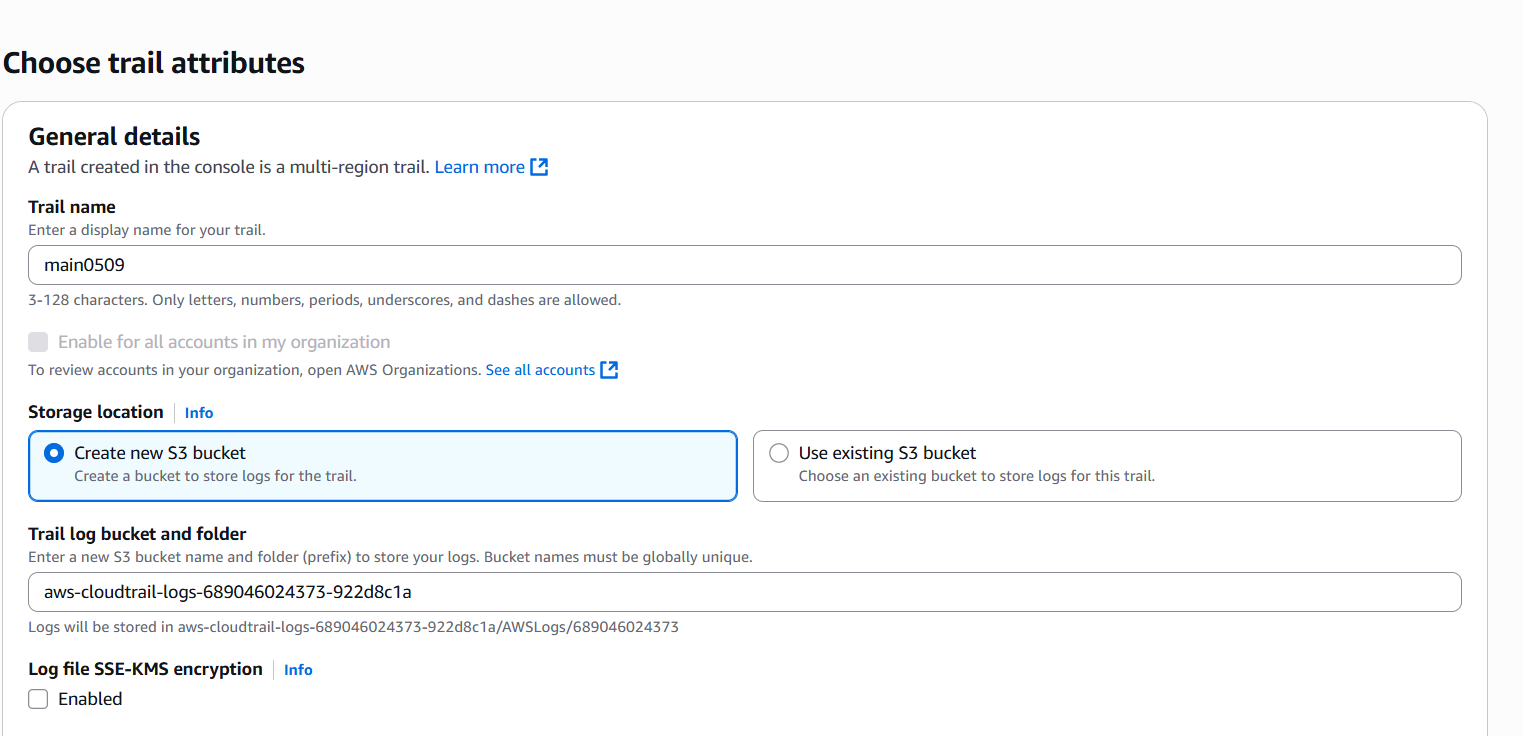
}

]

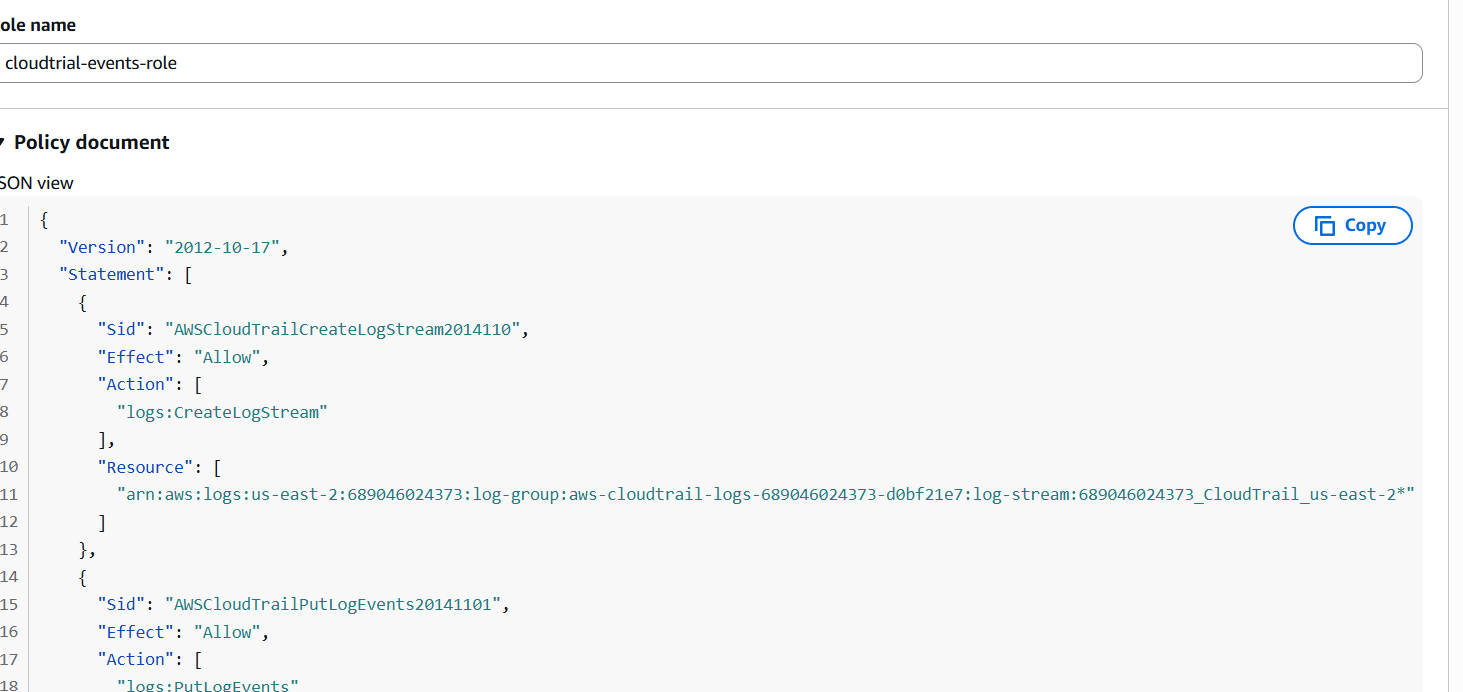
}

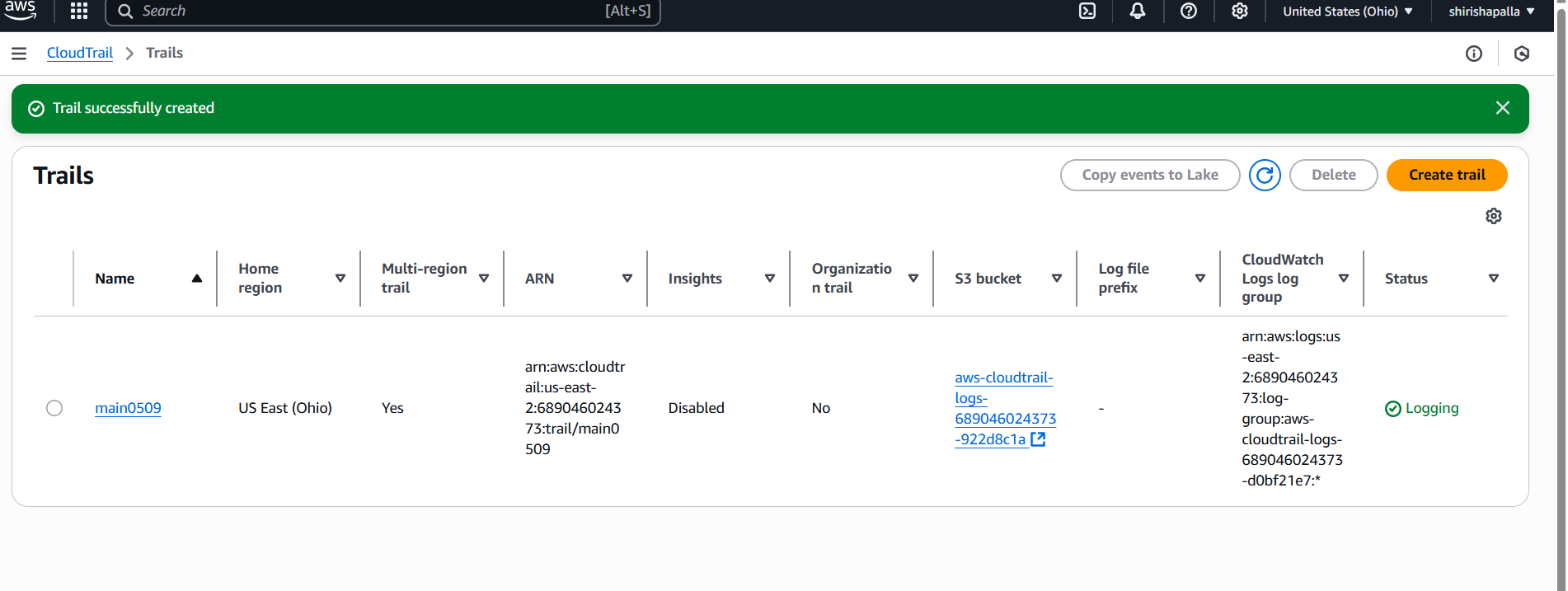
**✅ Step 3: Create a CloudWatch Log Group (Optional but Required for Logs in CloudWatch)**

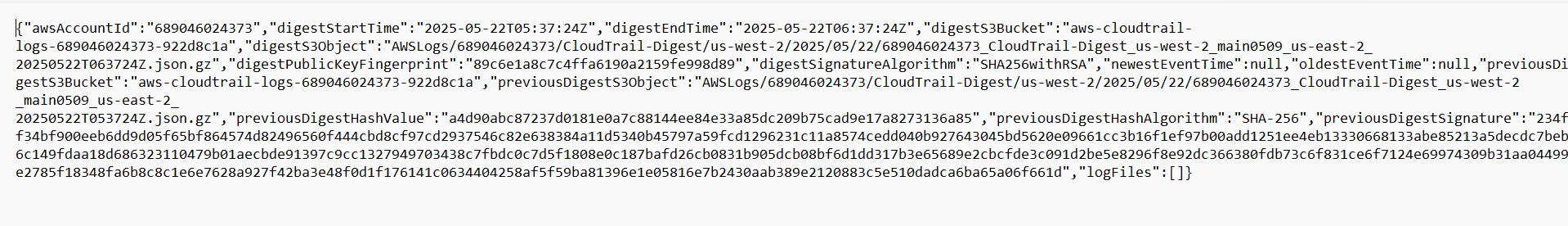
1. Go to **CloudWatch Console**: <https://console.aws.amazon.com/cloudwatch/>
2. Click **Log groups** in the left menu.
3. Click **Create log group**.
4. Provide a name (e.g., /aws/cloudtrail/logs).
5. Click **Create**.





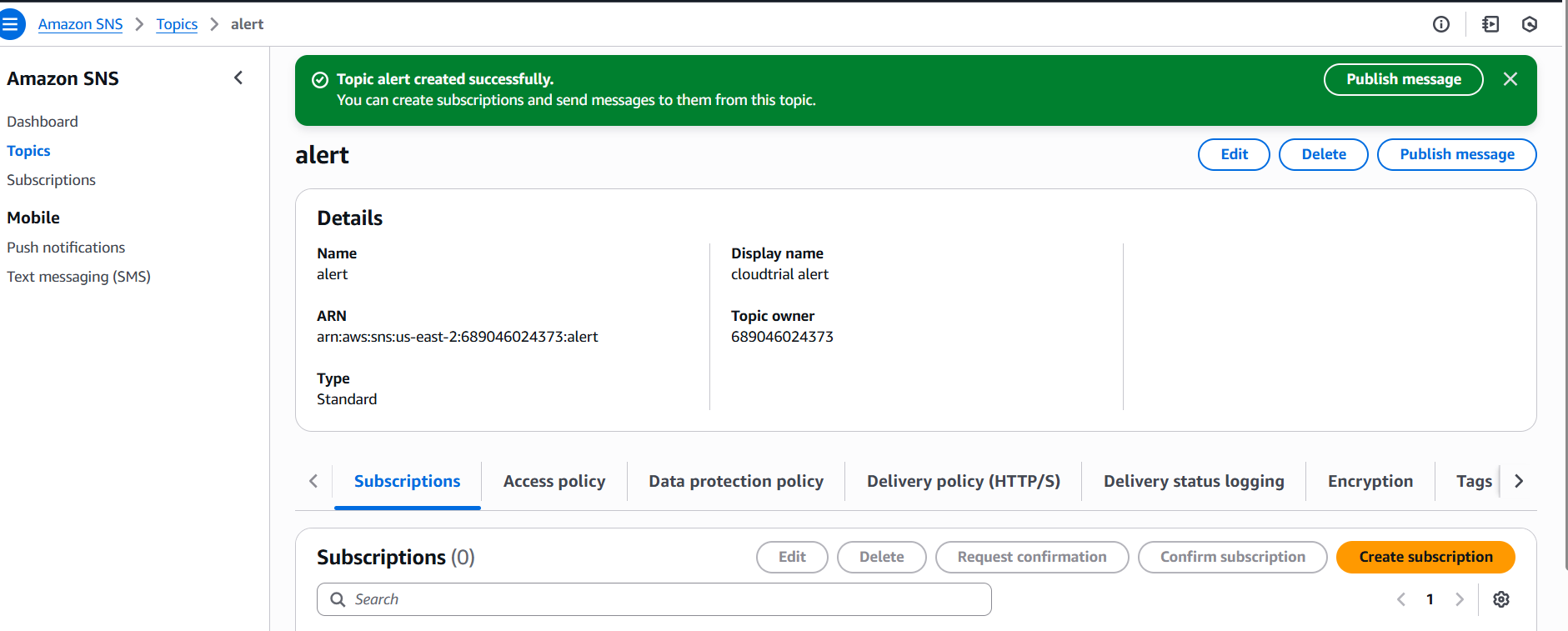




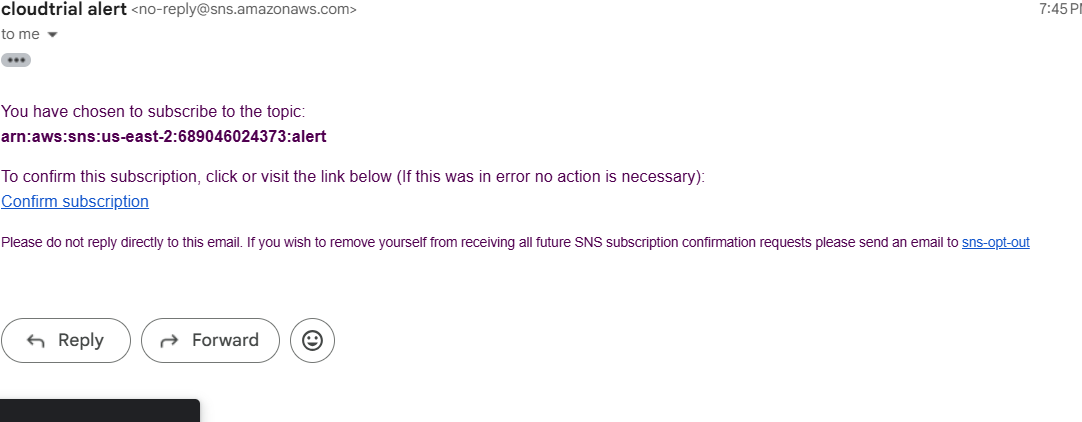


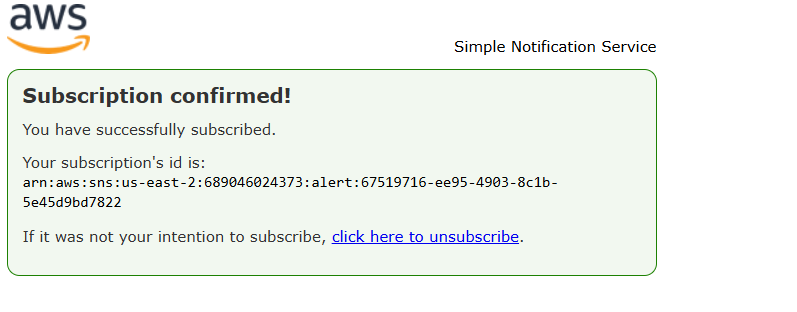
1. Enable SNS for CloudTrail to send alert on email.

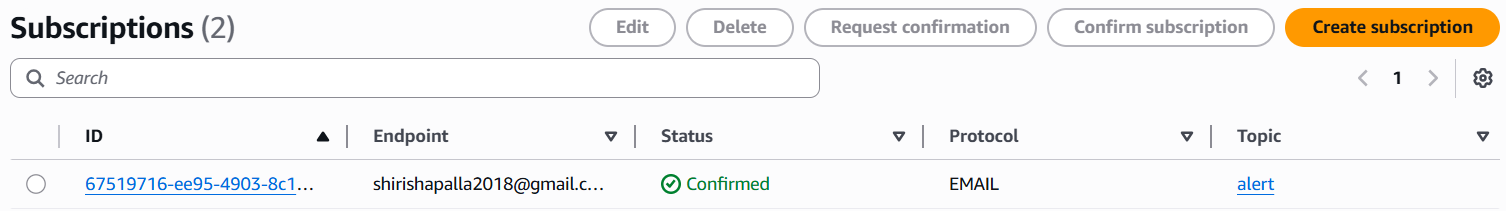
* Go to SNS
* Click on topic and provide name then create topic
* After creating topic
* In topic , Go to subscription --->create subscription
* Provide email id/SMS(number)
* Click on create subscription
* Notification sent email/SMS
* Click on confirm subscription





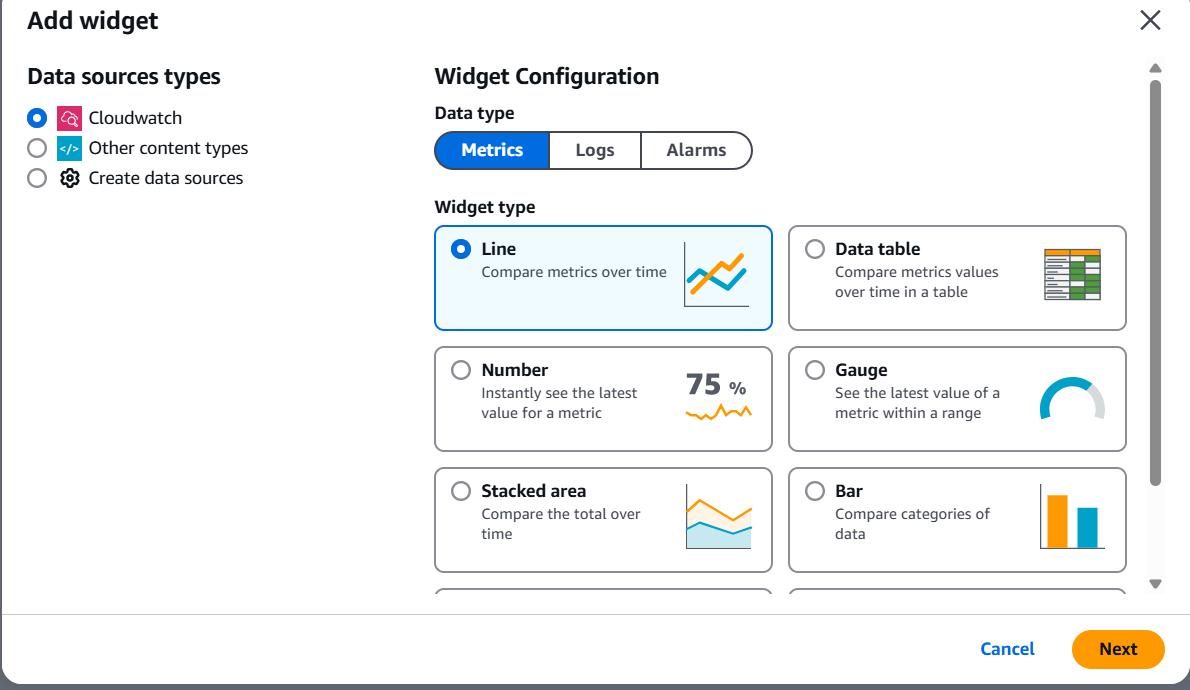


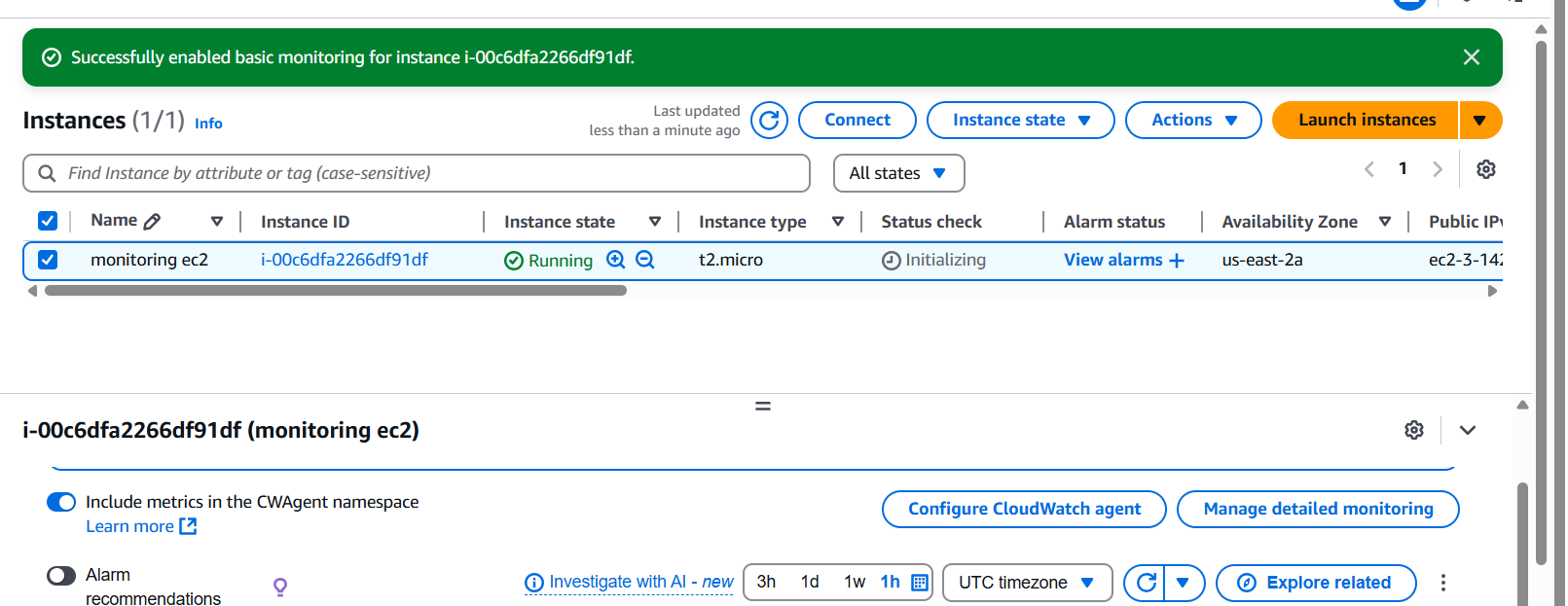


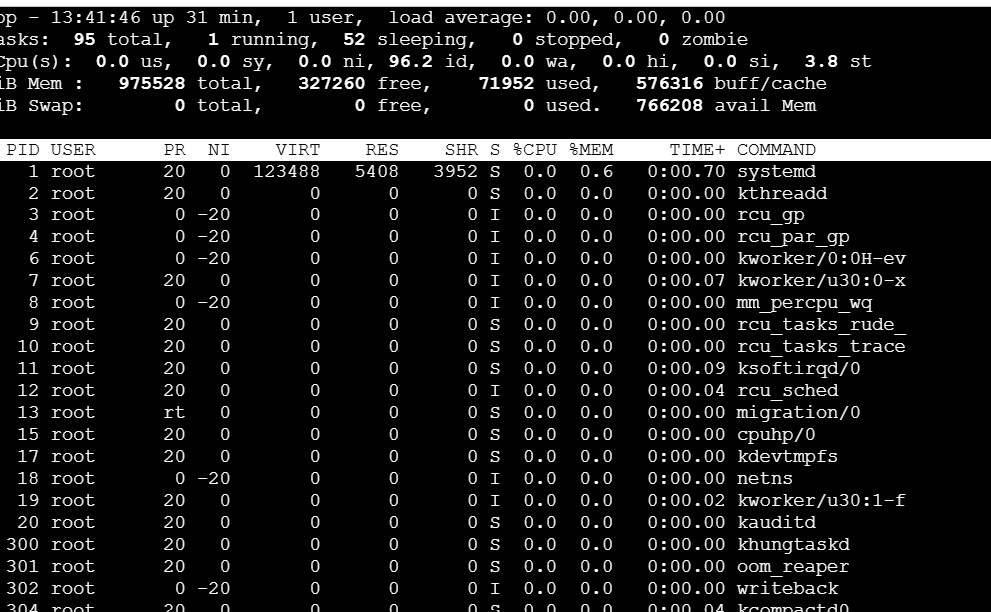


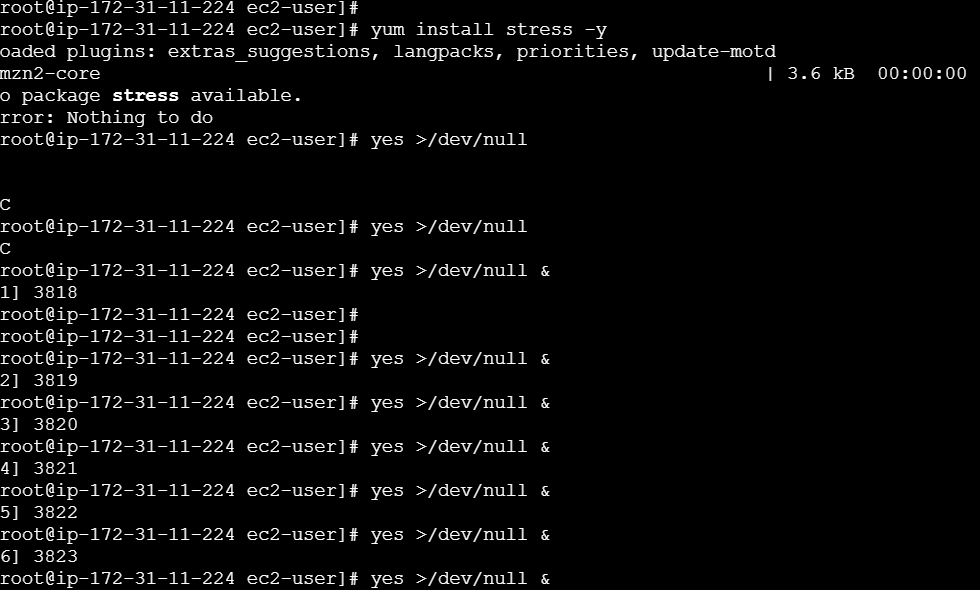
1. Configure cloud watch monitoring and record the CPU utilization and other metrics of ec2.

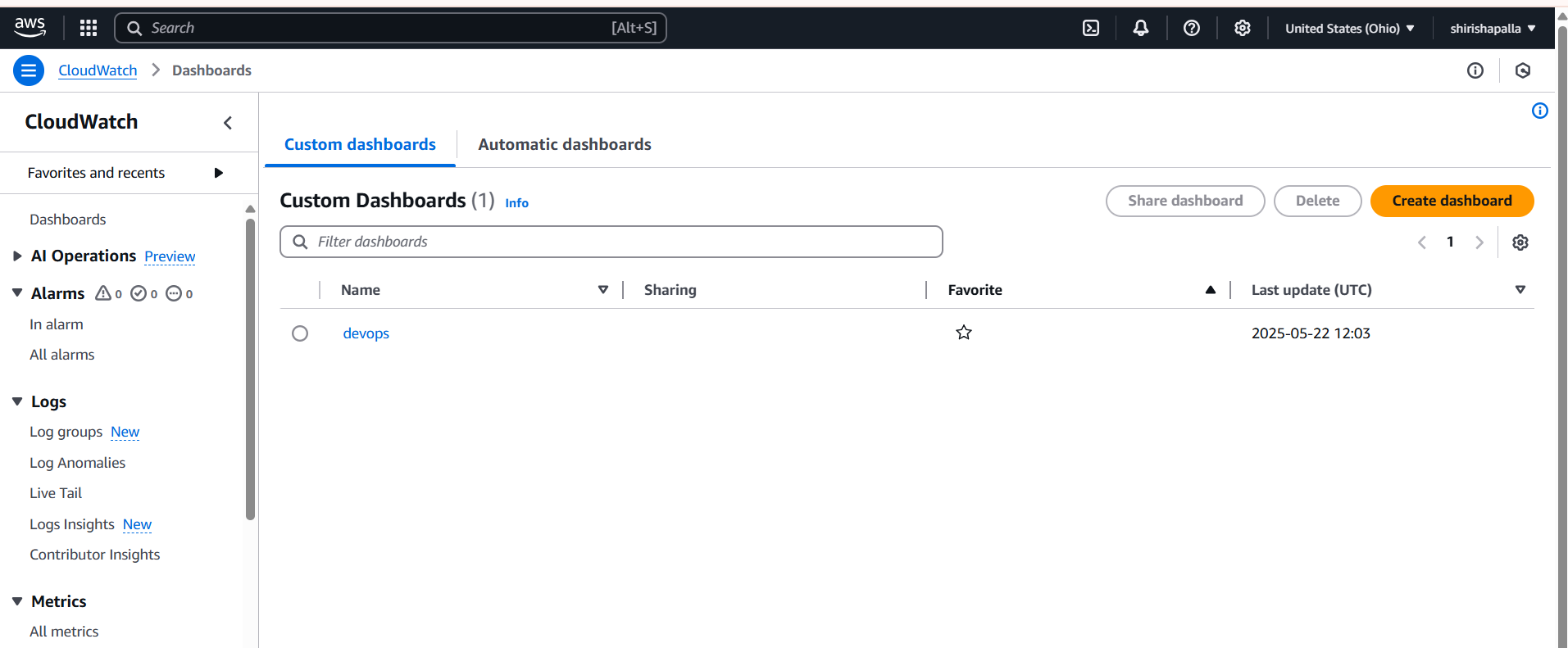
* Go to cloud watch dashboard
* Click on create dashboard
* Select widget type
* Select metric (eg:ec2 and ec2 id)
* In this instance, Select monitor (CPU utilization)
* Then save

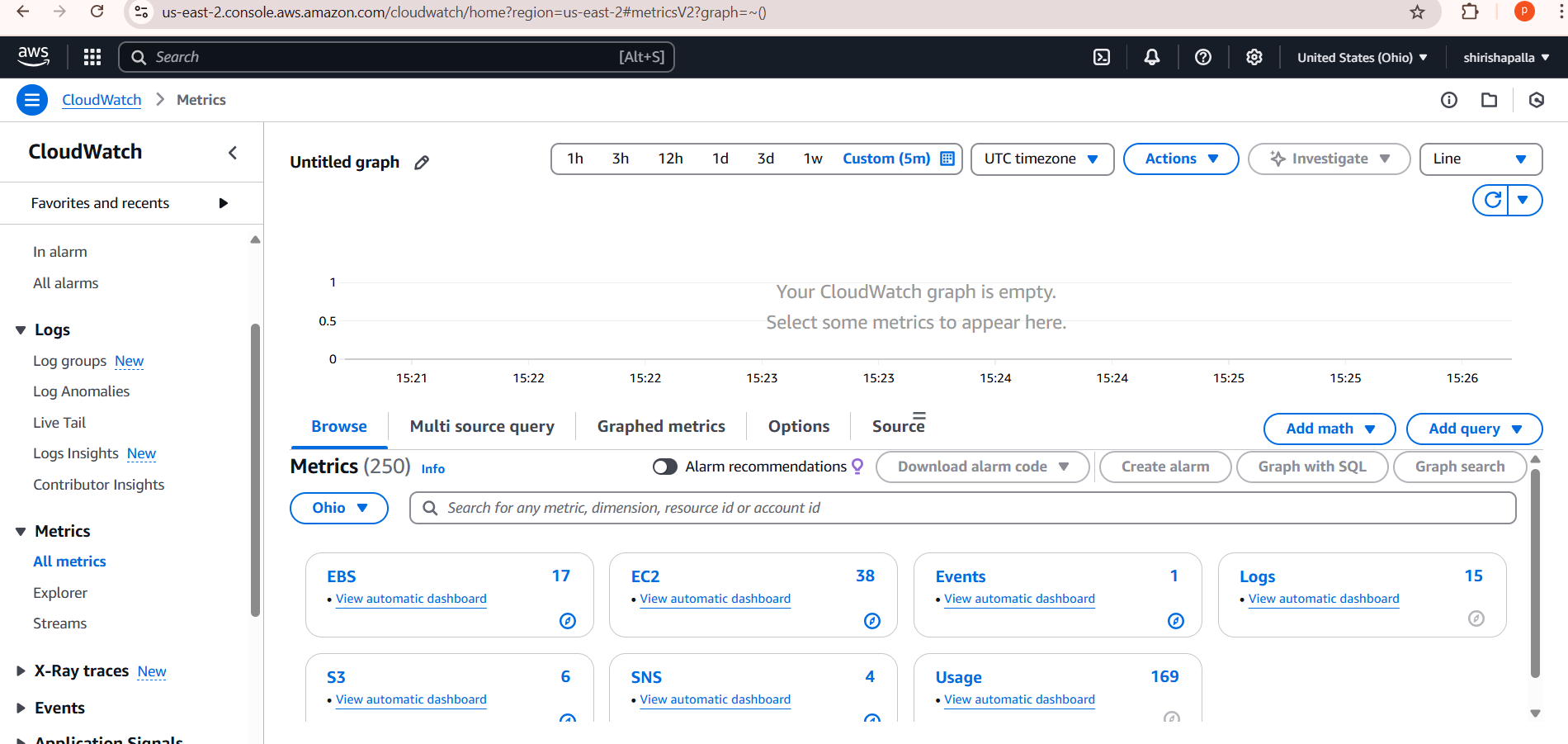


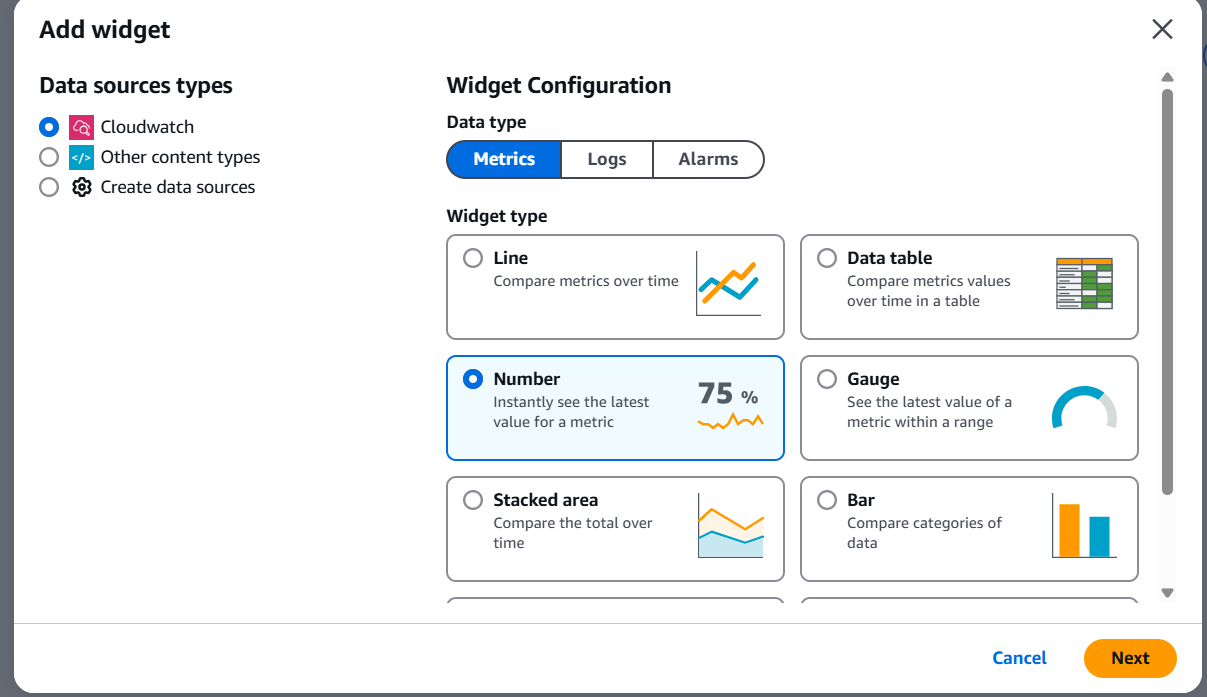


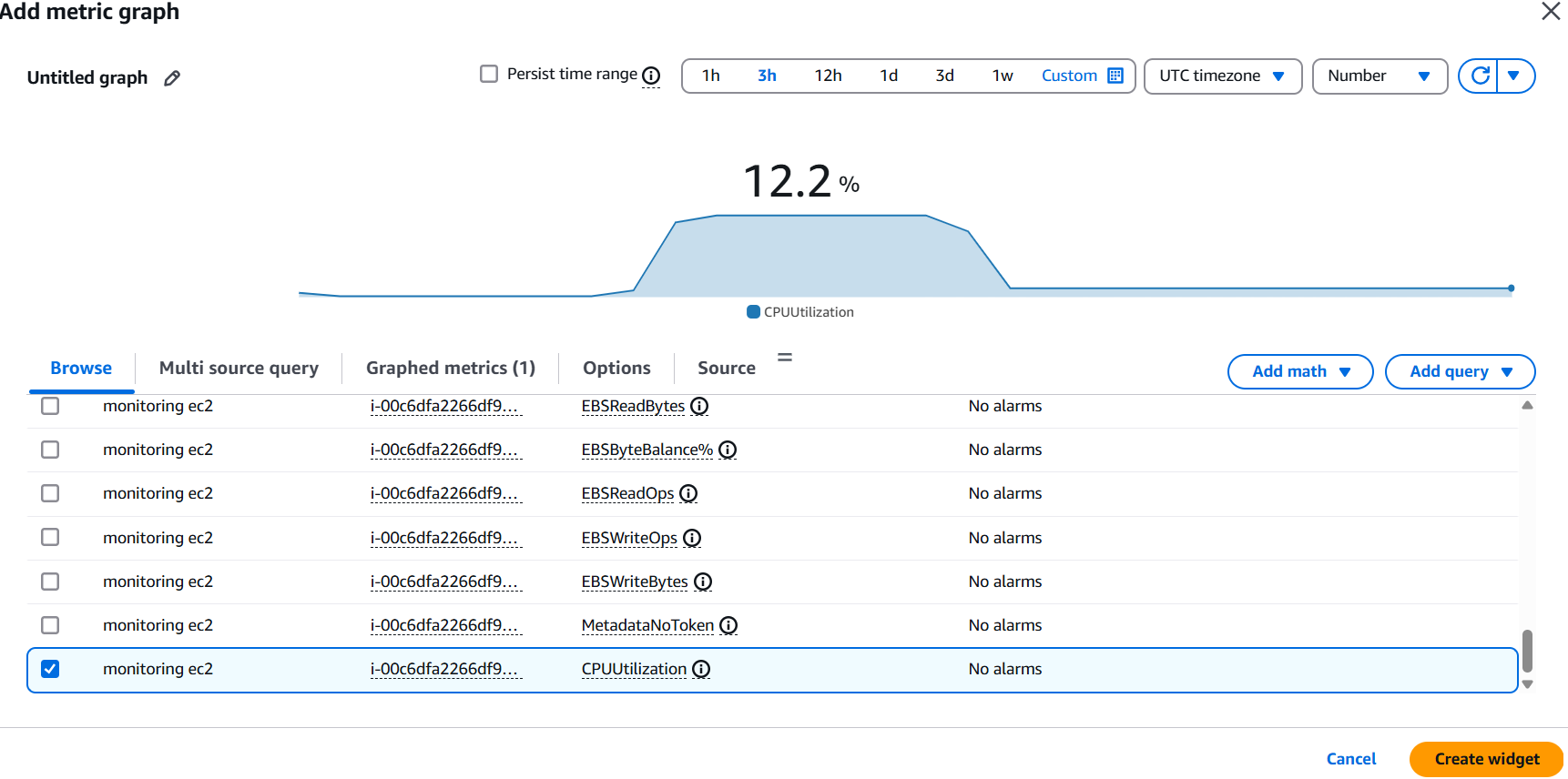


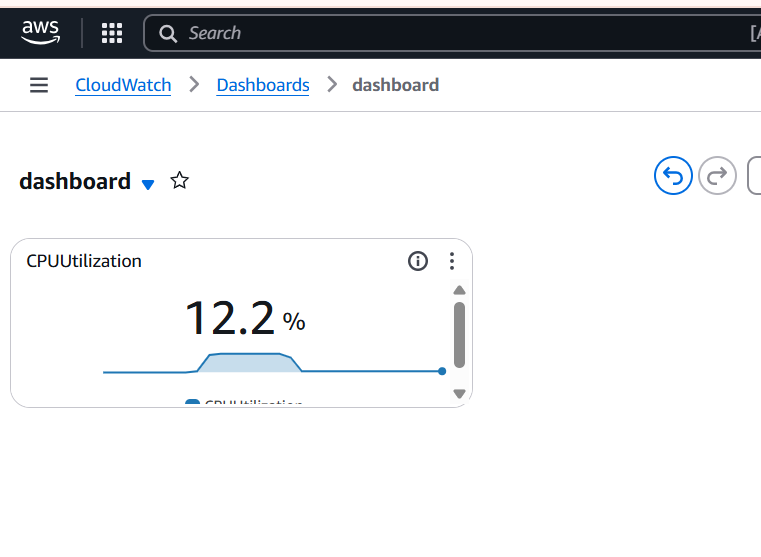






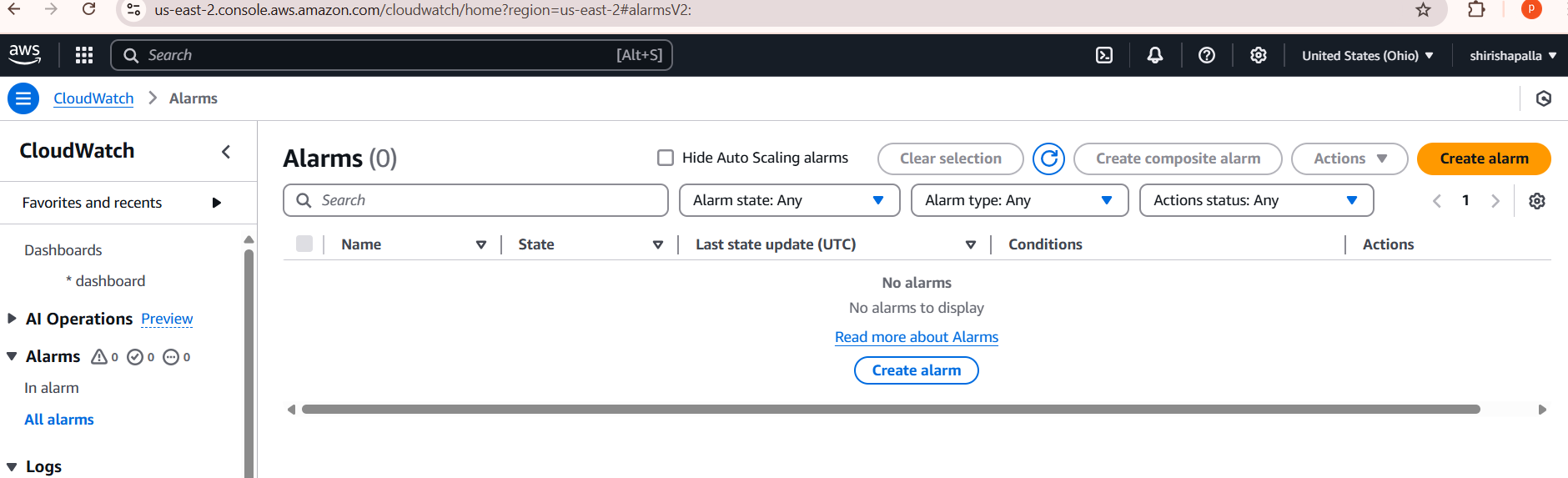


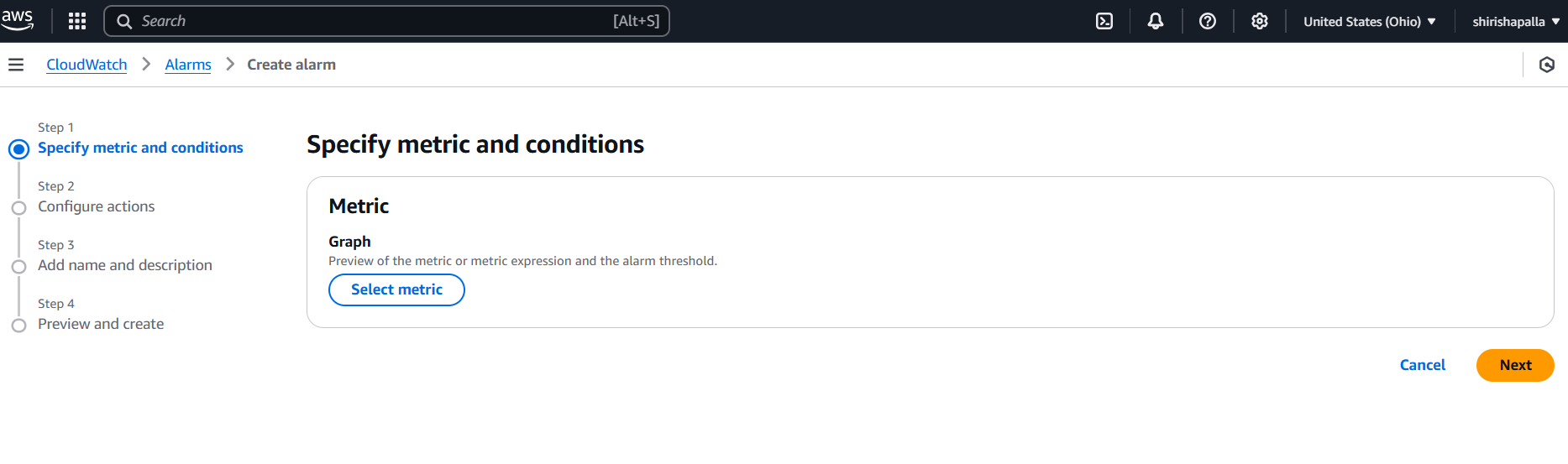


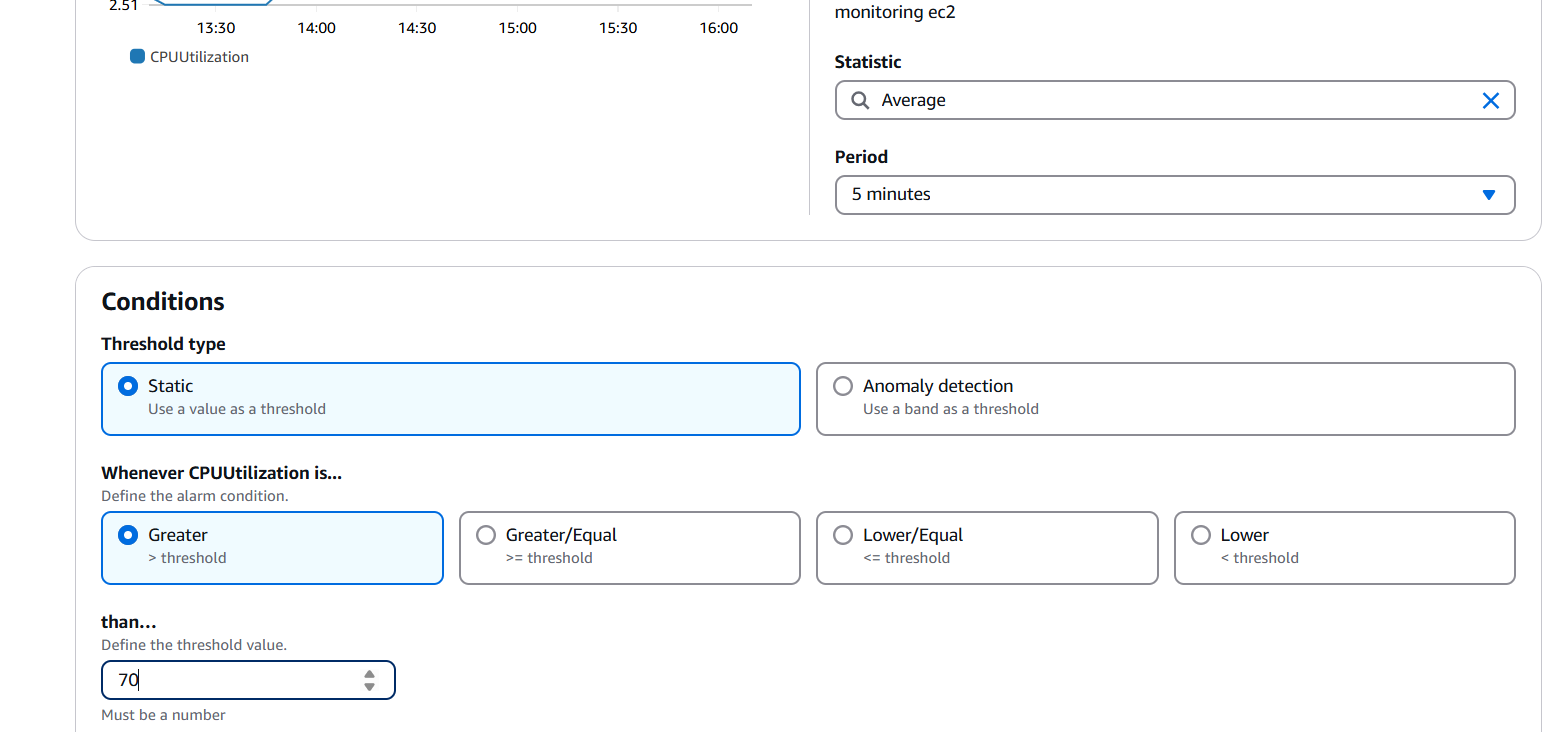


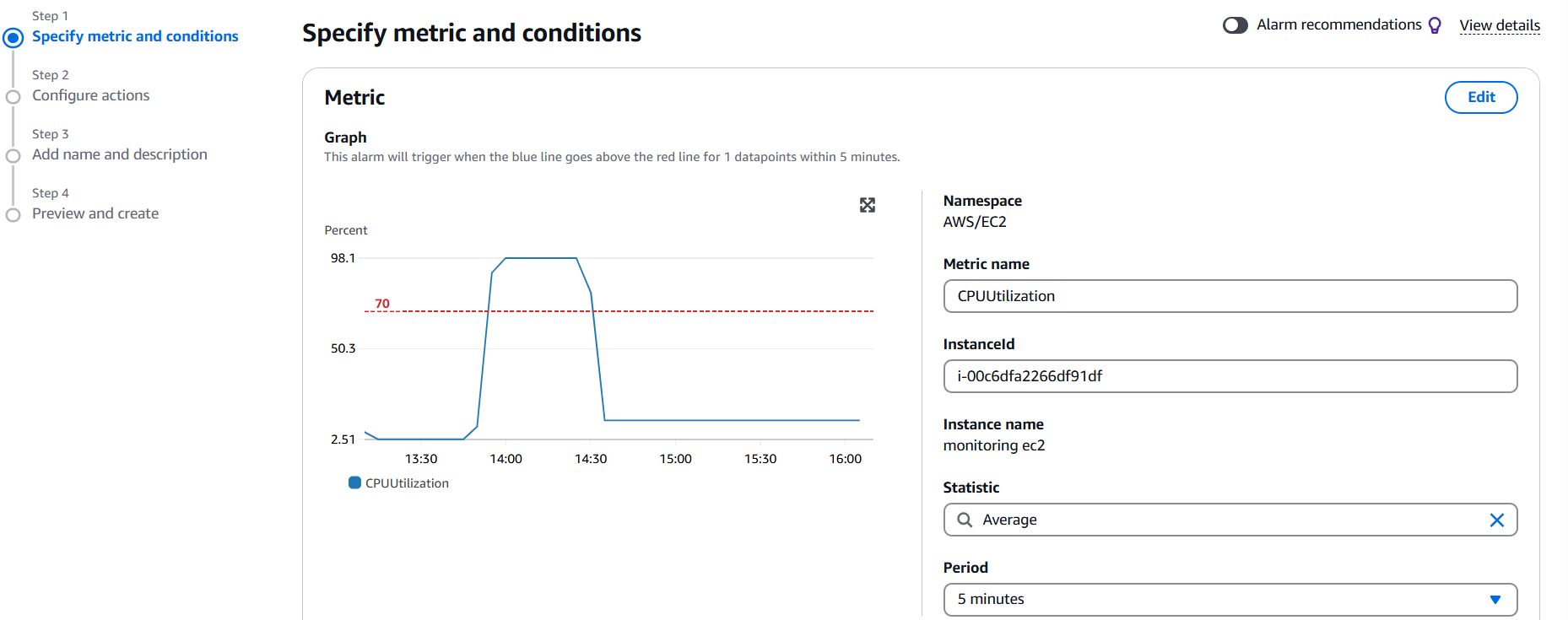
1. Create one alarm to send alert to email if the CPU utilization is more than 70 percent.

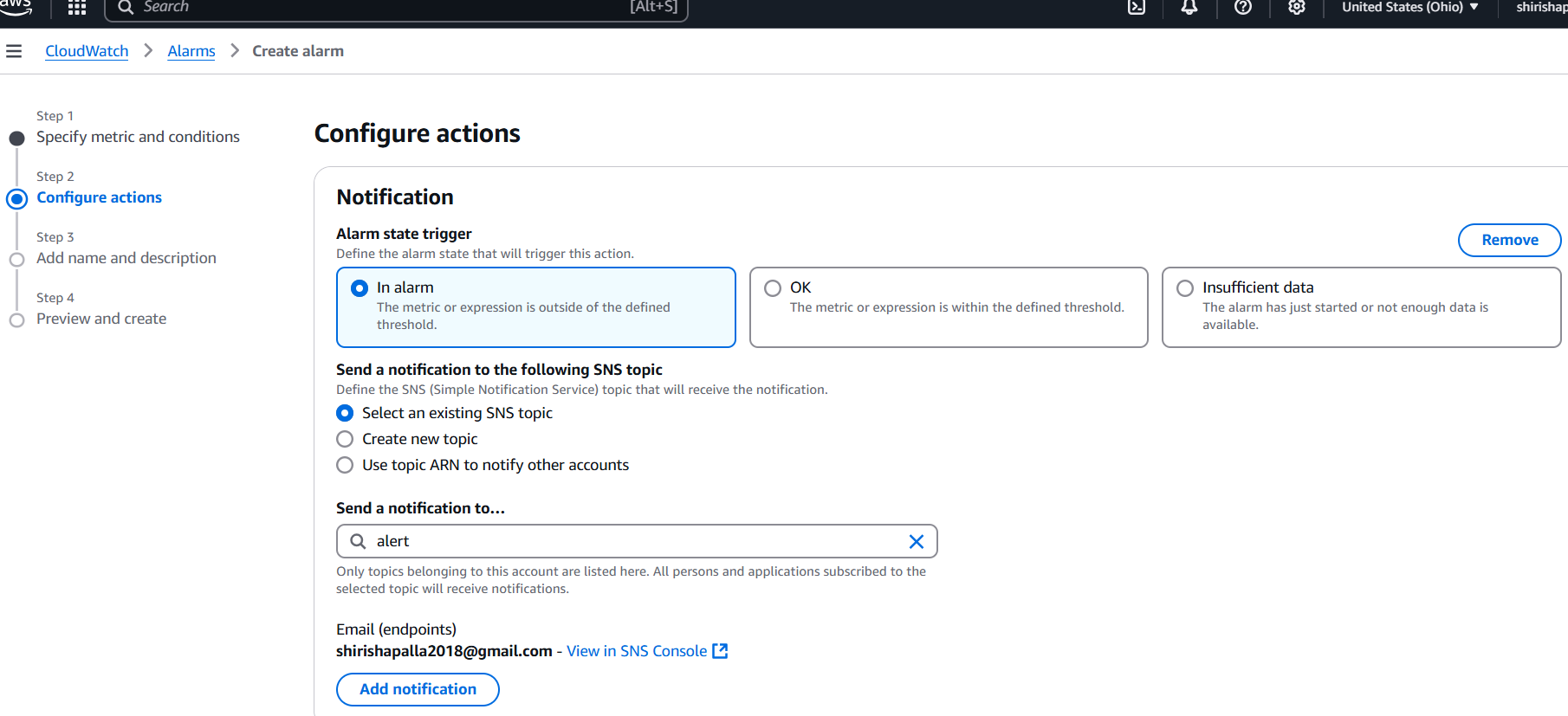
* Go to CloudWatch--->Alarms
* Click Create alarms select metric, Threshold type, and configuration actions
* Select SNS to get notification
* Then click on create Alarm
* Notification sent or email/SMS Then confirm
* Then set alerts set emails

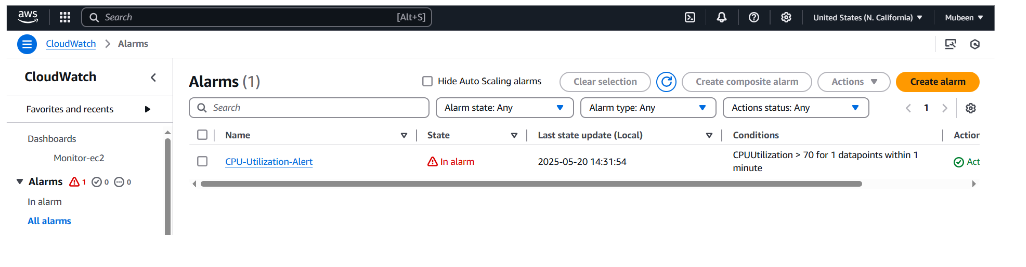




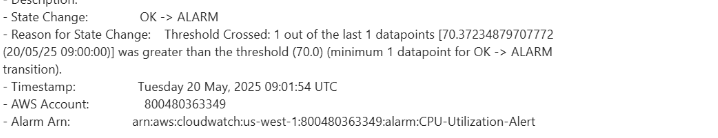












1. Create Dashboard and monitor tomcat service whether it is

Go to Alarms → Create Alarm

**Step 1: Install and Configure CloudWatch Agent**  
 Install the CloudWatch Agent (if not already installed)  
 sudo yum install amazon-cloudwatch-agent -y  
 **Create CloudWatch Agent configuration file**:

Save this file as /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-

agent.json

{

"metrics": {

"append\_dimensions": {

"InstanceId": "${aws:InstanceId}"

},

"metrics\_collected": {

"procstat": [

{

"exe": "java",

"measurement": [

"cpu\_usage",

"memory\_rss"

],

"metrics\_collection\_interval": 60,

"custom\_dimensions": {

"Application": "Tomcat"

}

}

]

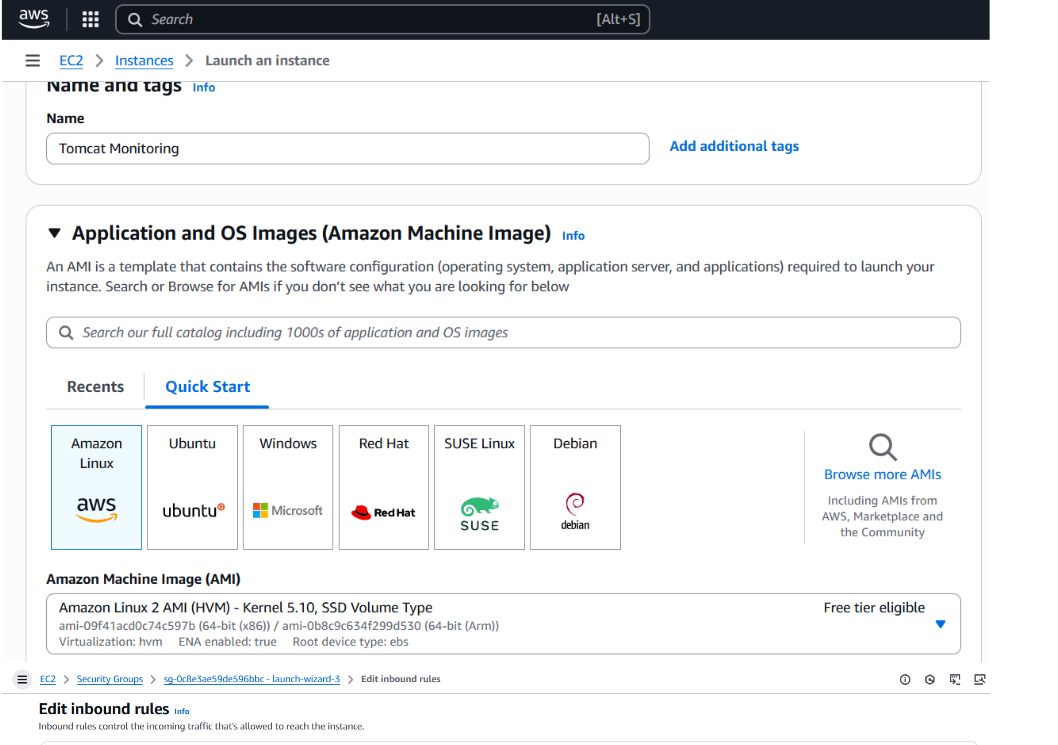
}

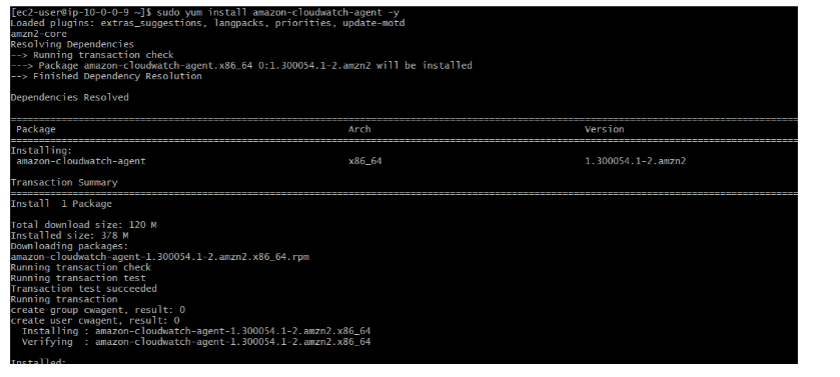
}

}

Apply and start the agent:

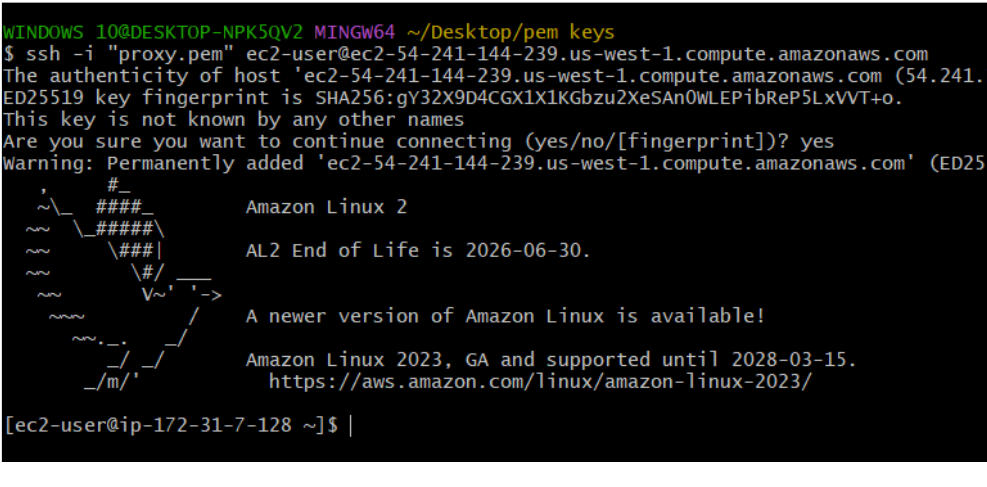
sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl \

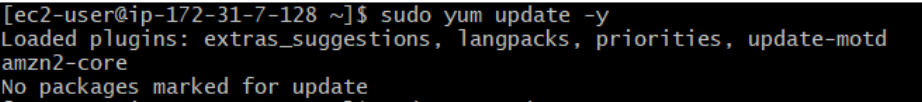
* -a fetch-config \  
   -m ec2 \  
   -c file:/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json \  
  

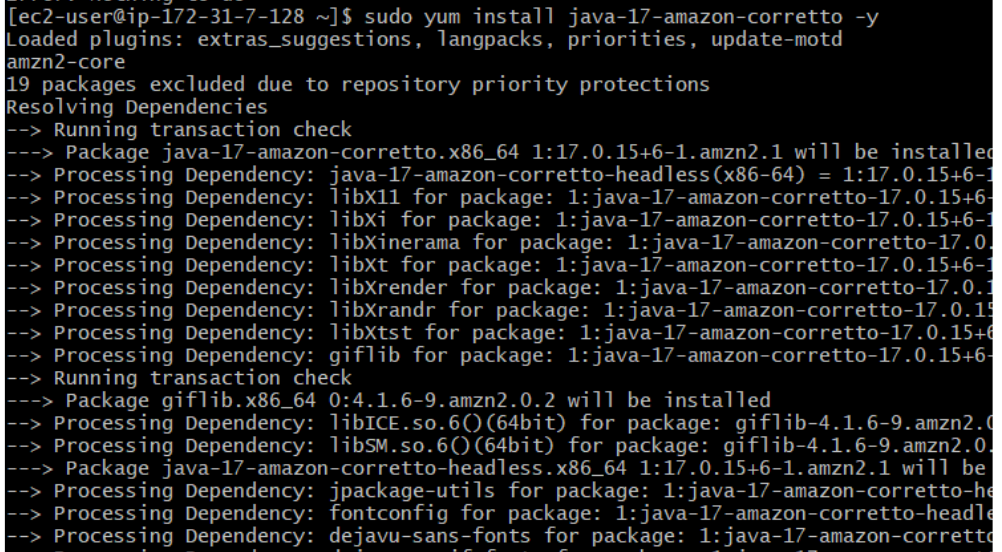


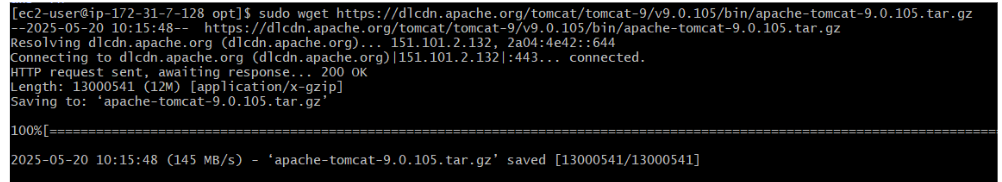
**Step 2: Install and Start Apache Tomcat**  
Download and extract the latest Tomcat:  
wgethttps://downloads.apache.org/tomcat/tomcat9/v9.0.105/bin/apache-  
tomcat-9.0.105.tar.gz  
sudo mkdir -p /opt/tomcat  
sudo tar -xvzf apache-tomcat-9.0.105.tar.gz -C /opt/tomcat --strip-components=1  
sudo chmod +x /opt/tomcat/bin/\*.sh  
sudo /opt/tomcat/bin/startup.sh

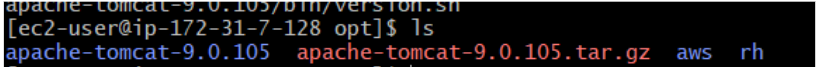
ps aux | grep tomcat

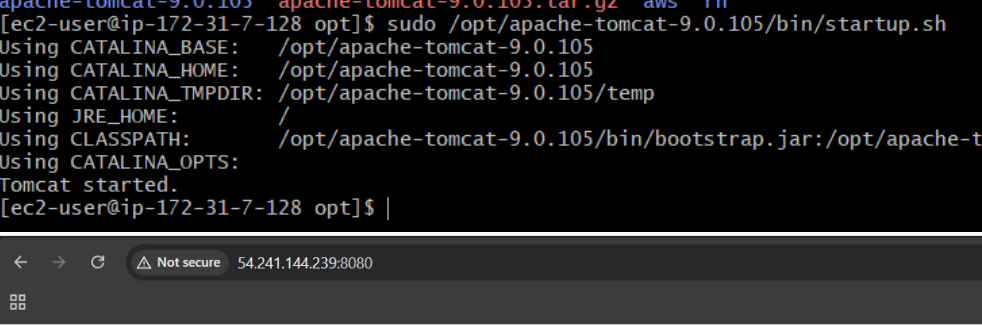


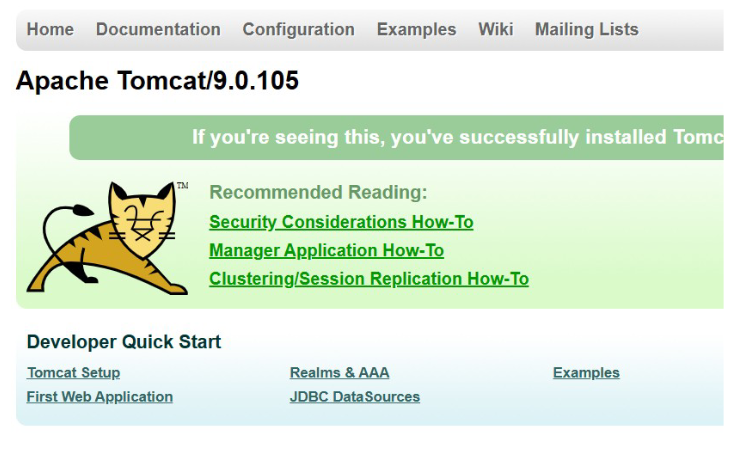




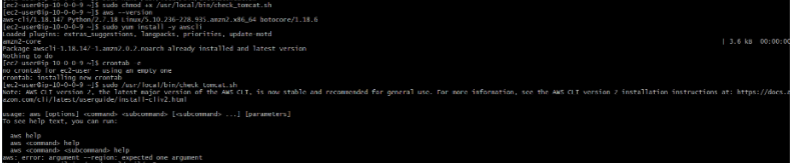




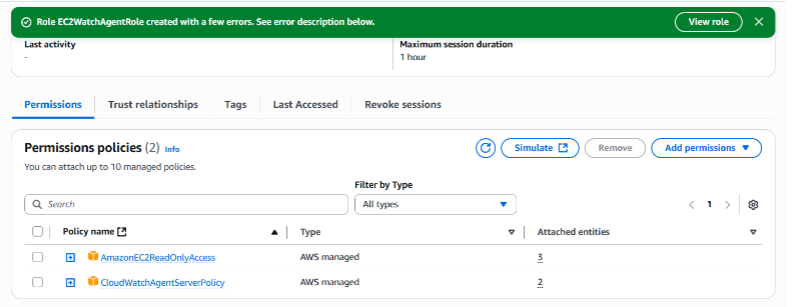


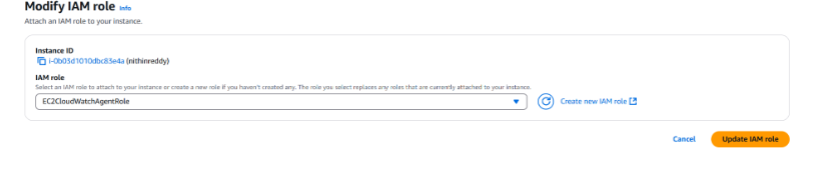


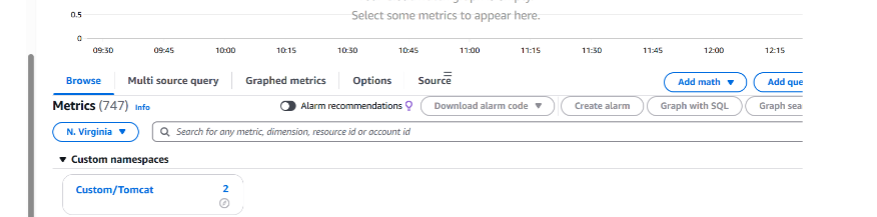
**Step 3: Push Custom Metric "TomcatStatus"**  
Create health check script:  
Create /usr/local/bin/check\_tomcat.sh:  
#!/bin/bash  
if pgrep -f tomcat > /dev/null  
then  
 /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl \  
 -a put-metric-data \  
 -n "Custom/Tomcat" \  
 -m "TomcatStatus" \  
 --value 1  
else  
 /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl \  
 -a put-metric-data \  
 -n "Custom/Tomcat" \  
 -m "TomcatStatus" \  
 --value 0  
Fi  
sudo chmod +x /usr/local/bin/check\_tomcat.sh  
Add it to crontab to run every minute:  
crontab -e



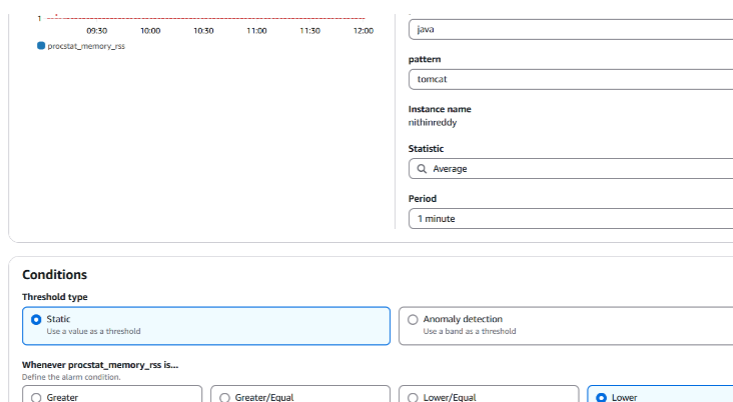
Step 4: Create IAM Role (If Not Already Done)  
1. Go to IAM → Roles → Create Role  
2. Select EC2 as trusted entity  
3. Attach CloudWatchAgentServerPolicy  
4. Attach the role to the EC2 instance

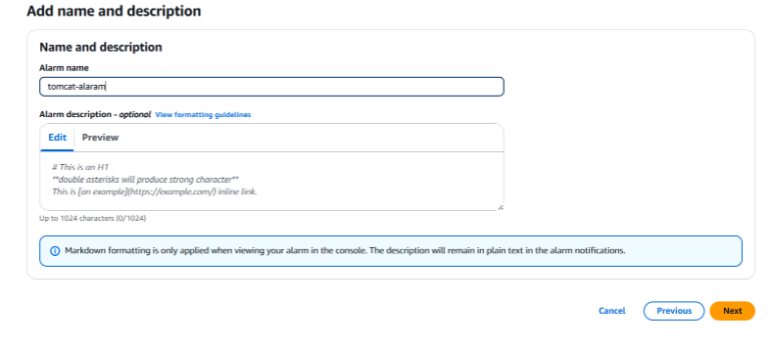


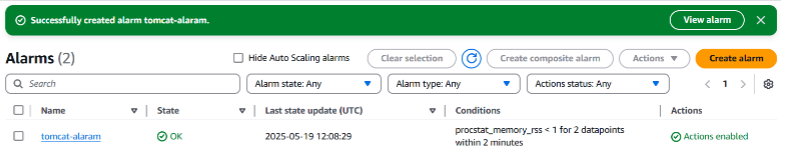
 Step 5: View Custom Metric in CloudWatch  
1. Go to CloudWatch Console  
2. Navigate to Metrics → Custom → Custom/Tomcat



Step 6: Create CloudWatch Alarm  
1. Go to CloudWatch → Alarms → Create Alarm  
2. Choose Custom → Custom/Tomcat  
3. Select TomcatStatus  
4. Set condition:  
Whenever TomcatStatus < 1 for 2 datapoints within 2 minutes  
5. Set notification to an SNS Topic (Email)  
Confirm the subscription email if not done before  
6. Name and create the alarm







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

Install and Configure   
CloudWatch Agent  
Install CloudWatch Agent:  
sudo yum install -y amazon-CloudWatch-agent  
•Create a Script to Report Nginx   
Status as a Custom Metric  
sudo nano /opt/nginx\_status.sh  
#!/bin/bash  
STATUS=$(systemctl is-active   
nginx)  
METRIC\_VALUE=0  
if [ "$STATUS" == "active" ]; then  
 METRIC\_VALUE=1  
fi

aws cloudwatch put-metric-data \  
 --namespace "Custom/Nginx" \  
 --metric-name nginx\_up \  
 --value $METRIC\_VALUE \  
 --unit Count \  
 --region YOUR\_REGION  
•sudo chmod +x   
/opt/nginx\_status.sh  
•Run the Script Automatically   
(Without Cron)  
sudo nano   
/etc/systemd/system/nginx-monitor-  
cw.service  
[Unit]  
Description=Monitor Nginx and   
send status to CloudWatch  
[Service]  
ExecStart=/opt/nginx\_status.sh  
Restart=always

RestartSec=60  
[Install]  
WantedBy=multi-user.target  
•sudo systemctl daemon-reload  
•sudo systemctl enable --now   
nginx-monitor-cw  
5. View Metrics in CloudWatch  
Go to CloudWatch > Metrics  
Navigate to Custom/Nginx  
Find the nginx\_up metric  
 6. Create an Alarm  
1. Go to CloudWatch >   
Alarms > Create Alarm  
2. Select the nginx\_up metric  
3. Condition: Threshold < 1  
4. Action: Send notification to   
SNS topic (email alert)

