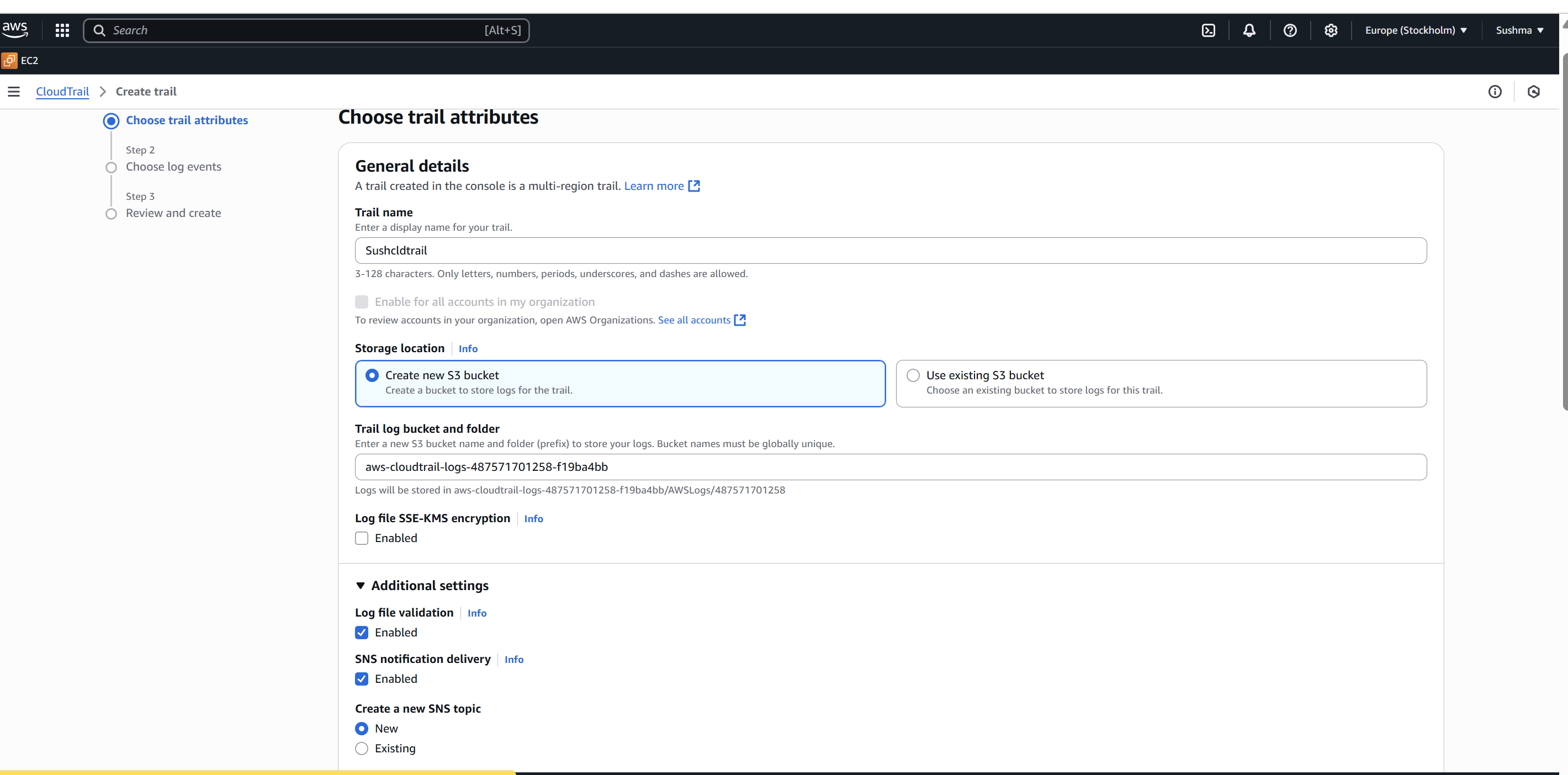
Cloud Trail

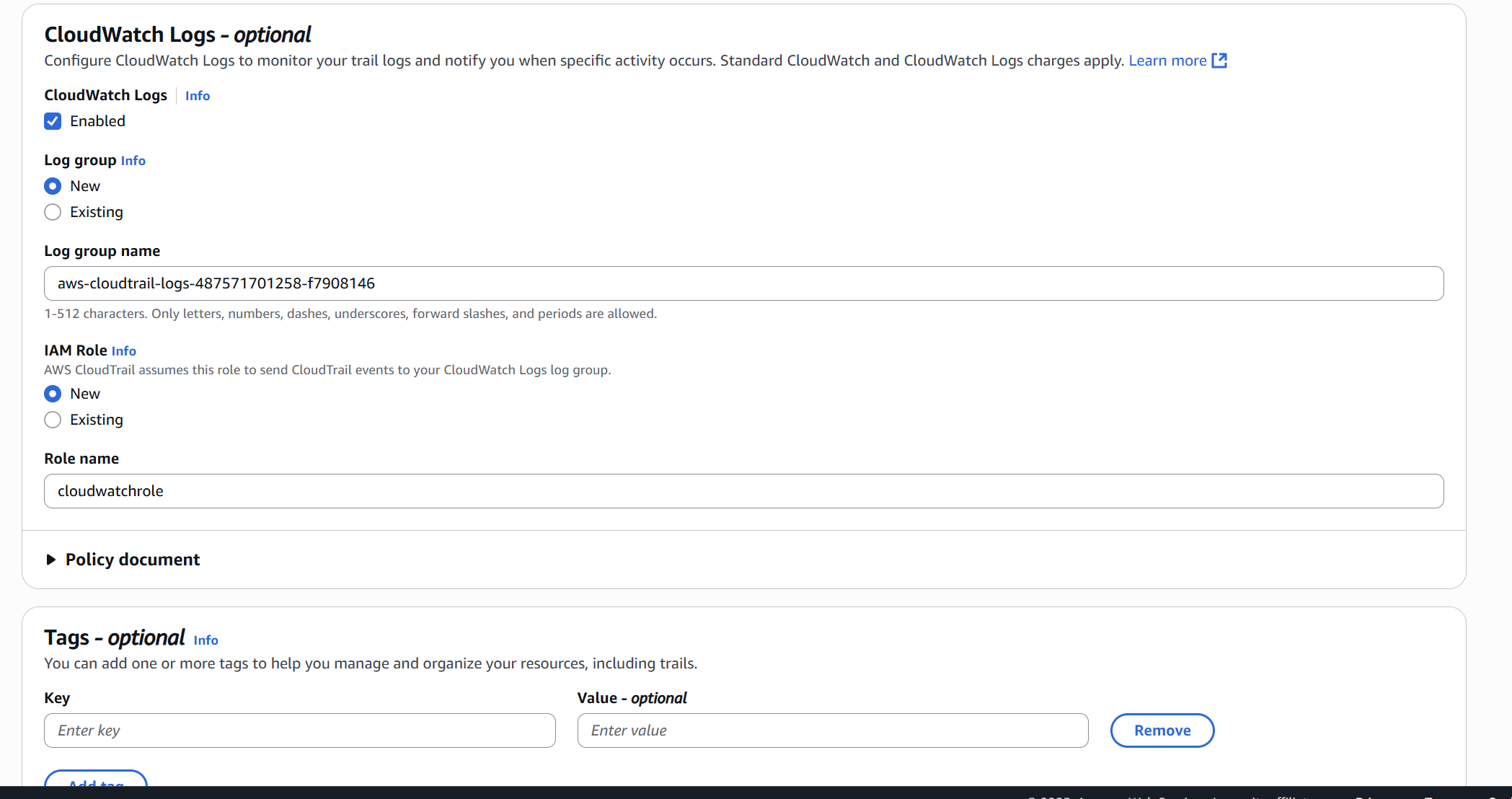
1) Enable cloudtrail monitoring and store the events in s3 and cloudwatch log events.

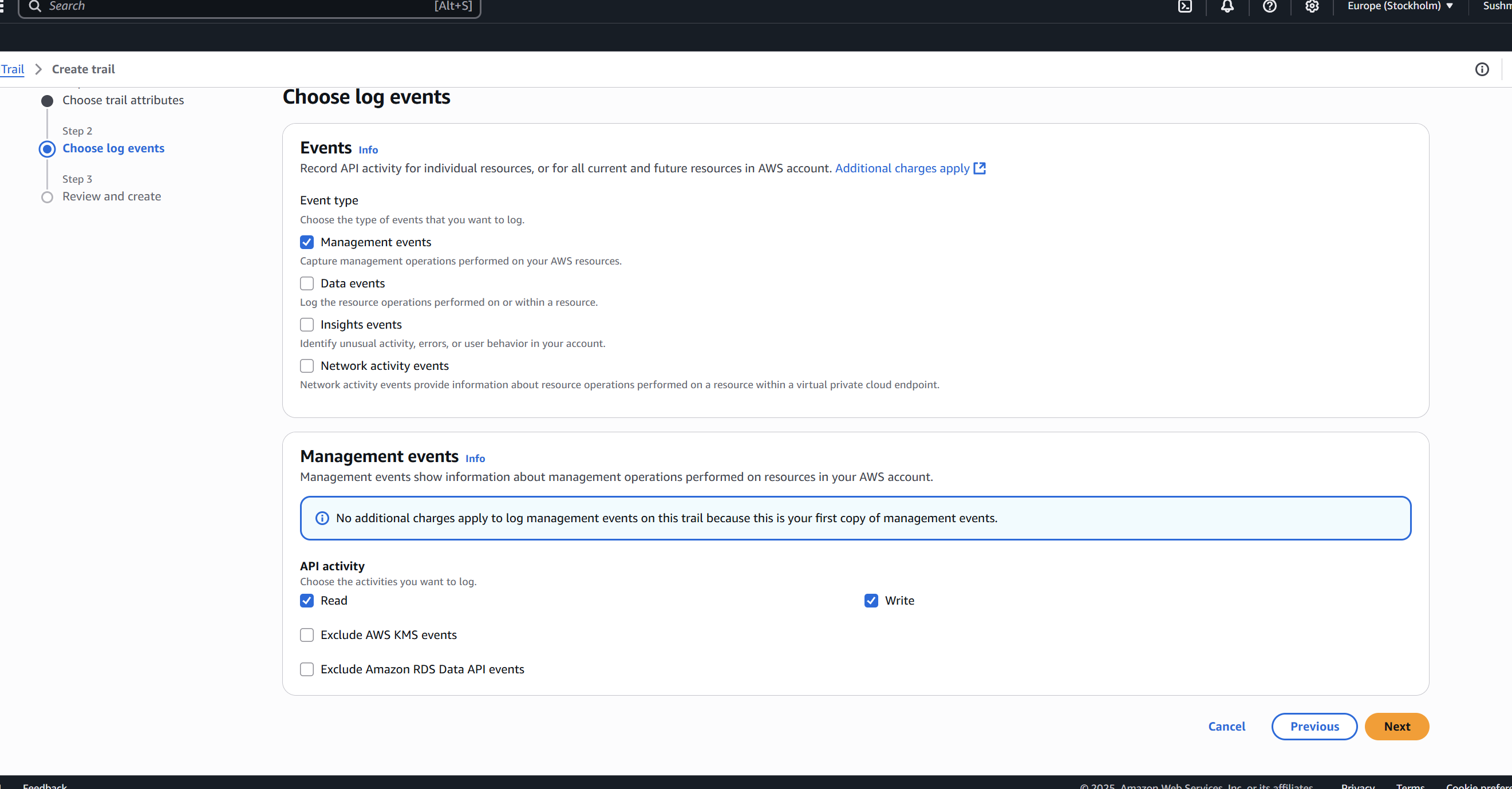
* Create a Trail
* Click **"Trails"** on the left menu.
* Click **"Create trail"**.

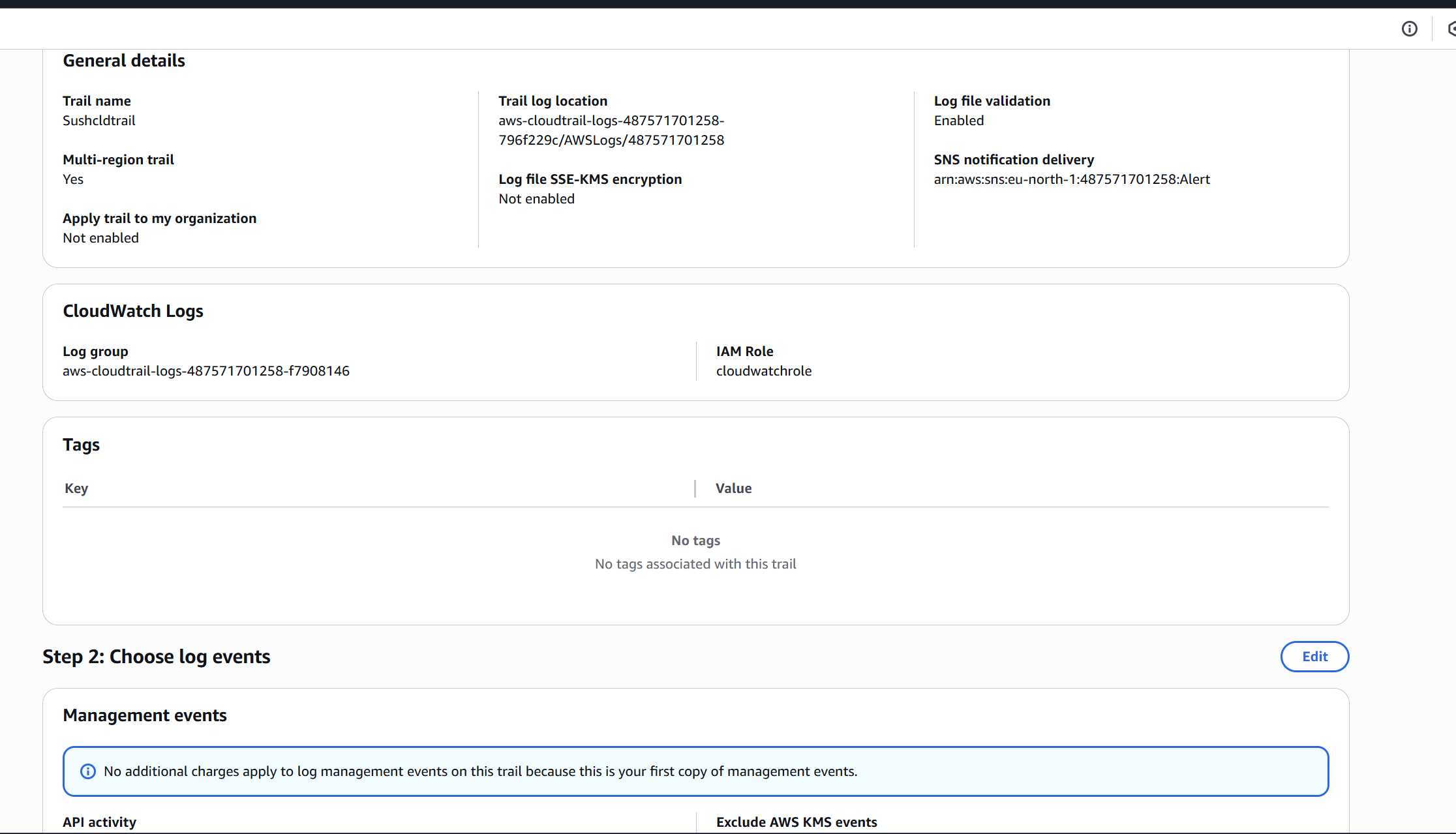
Configure Trail Details

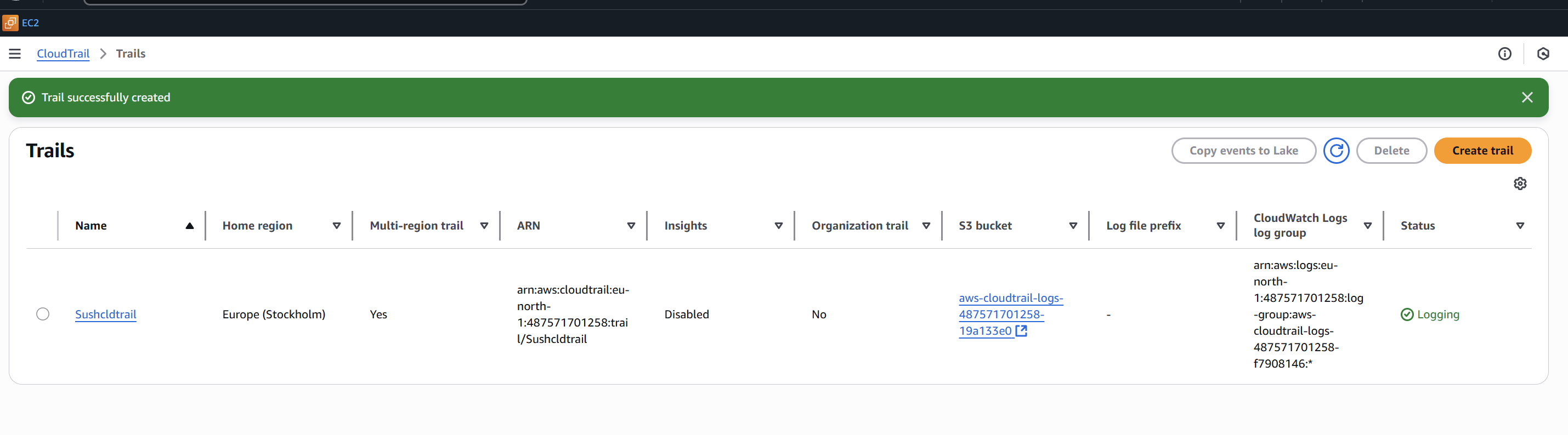
* **Trail name**: e.g., MyOrganizationTrail
* **Apply trail to all regions**: Recommended (**Yes**)
* **Create a new S3 bucket** or use an existing one
* Example: cloudtrail-logs-myaccount
* **Log file SSE encryption**: Optional (default is AES-256)
* **Log file validation**: Optional but recommende

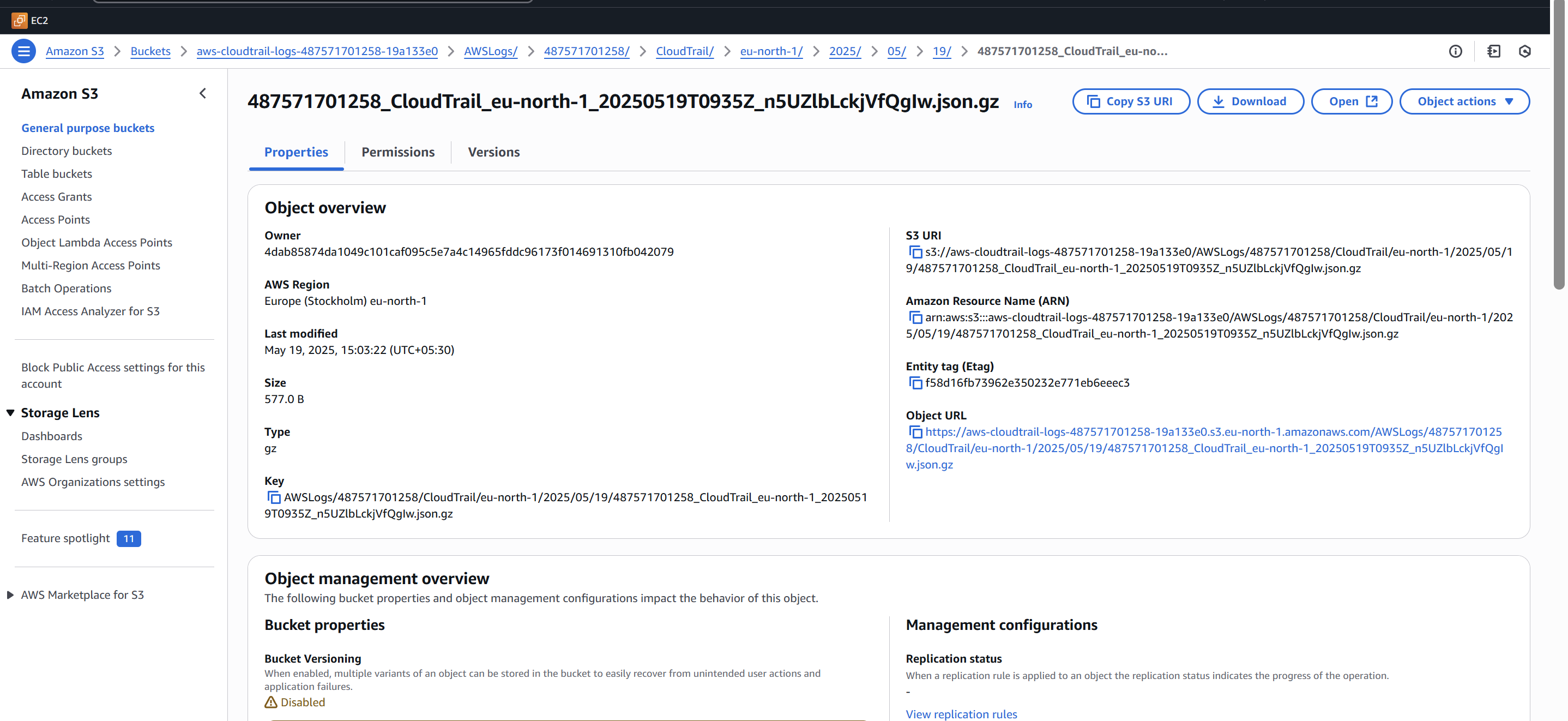












2) Enable SNS for cloudtrial to send alert on email.

**Step 1: Create an SNS Topic**

1. Navigate to **Amazon SNS → Topics**
2. Click **“Create topic”**
3. Select:
   * **Type**: Standard
   * **Name**: Alert
4. Click **Create topic**

**Result:** SNS topic Alert created successfully

**Step 2: Subscribe Your Email to the Topic**

1. Click on the topic you created: Alert
2. Click **“Create subscription”**
3. Set:
   * **Protocol**: Email
   * **Endpoint**: your email (e.g., sushmitha16rachakonda@outlook.com)
4. Click **Create subscription**

**Note:** You'll receive a confirmation email — click the link in the email to confirm.

**Step 3: Create an EventBridge Rule for CloudTrail Event**

1. Navigate to **Amazon EventBridge → Rules**
2. Click **“Create rule”**
3. Fill in:
   * **Name**: e.g., cloudtrail-alert-deletebucket
   * **Event Source**: AWS events or custom patterns
   * Choose **Custom pattern (JSON)**

{

"source": ["aws.s3"],

"detail-type": ["AWS API Call via CloudTrail"],

"detail": {

"eventName": ["DeleteBucket"]

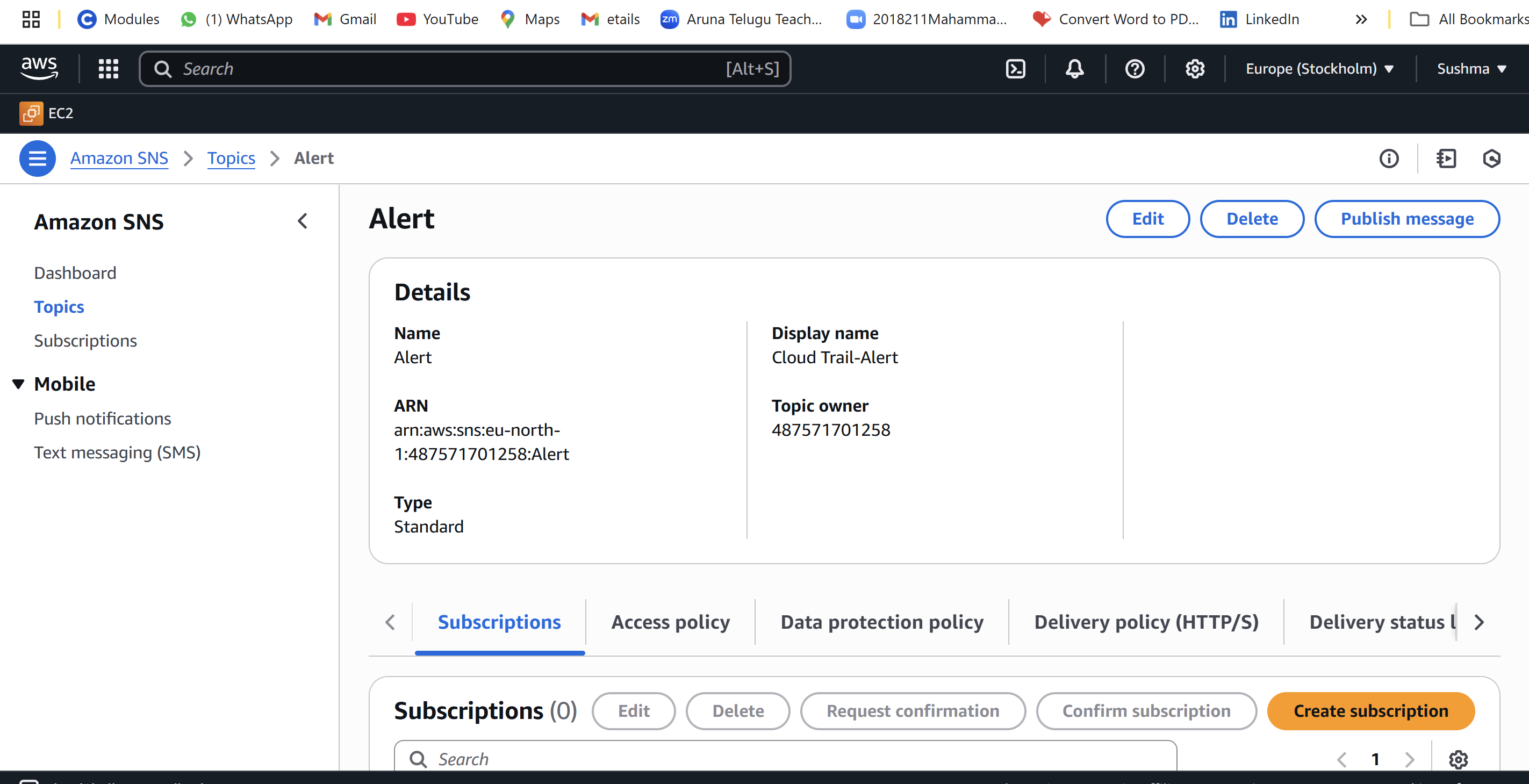
}

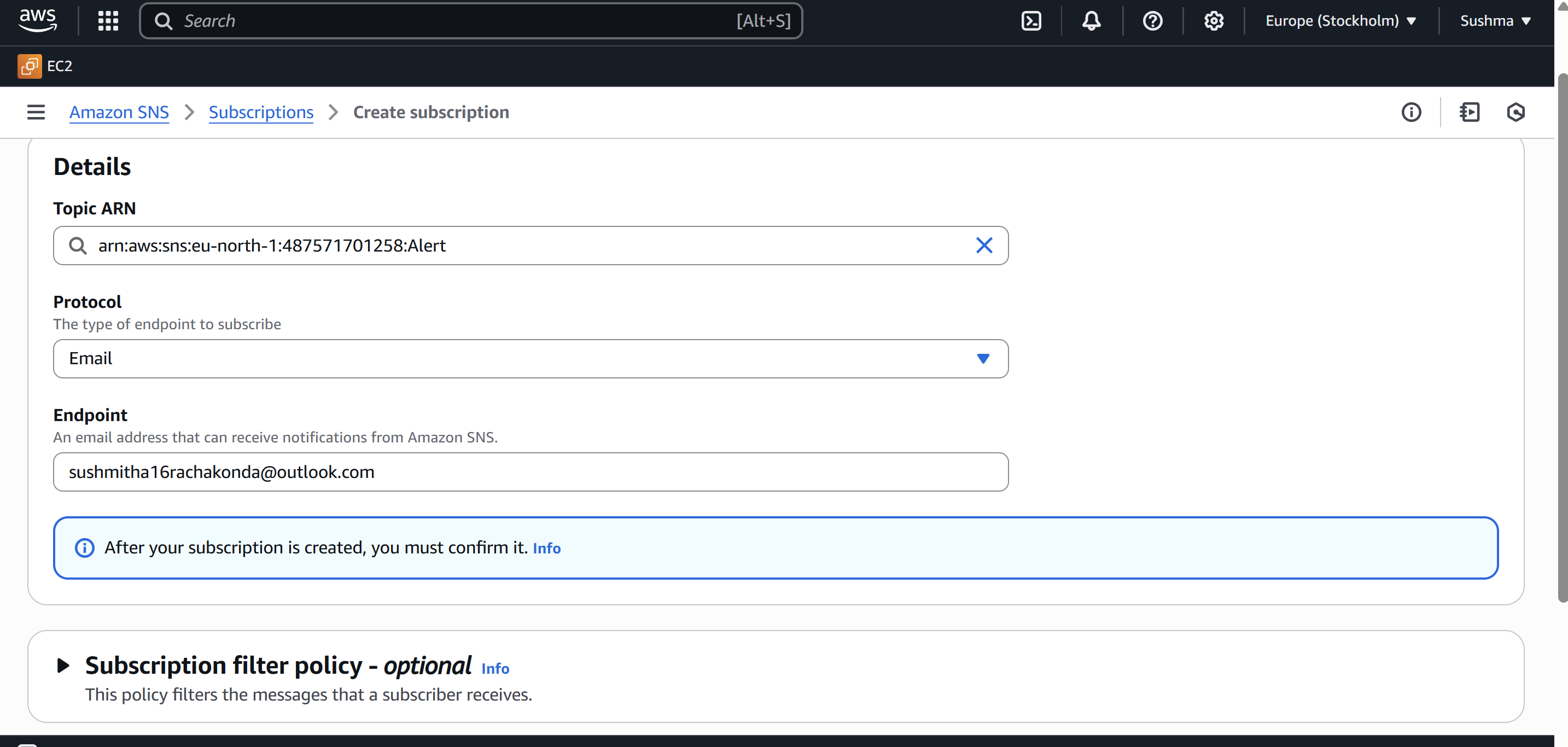
}

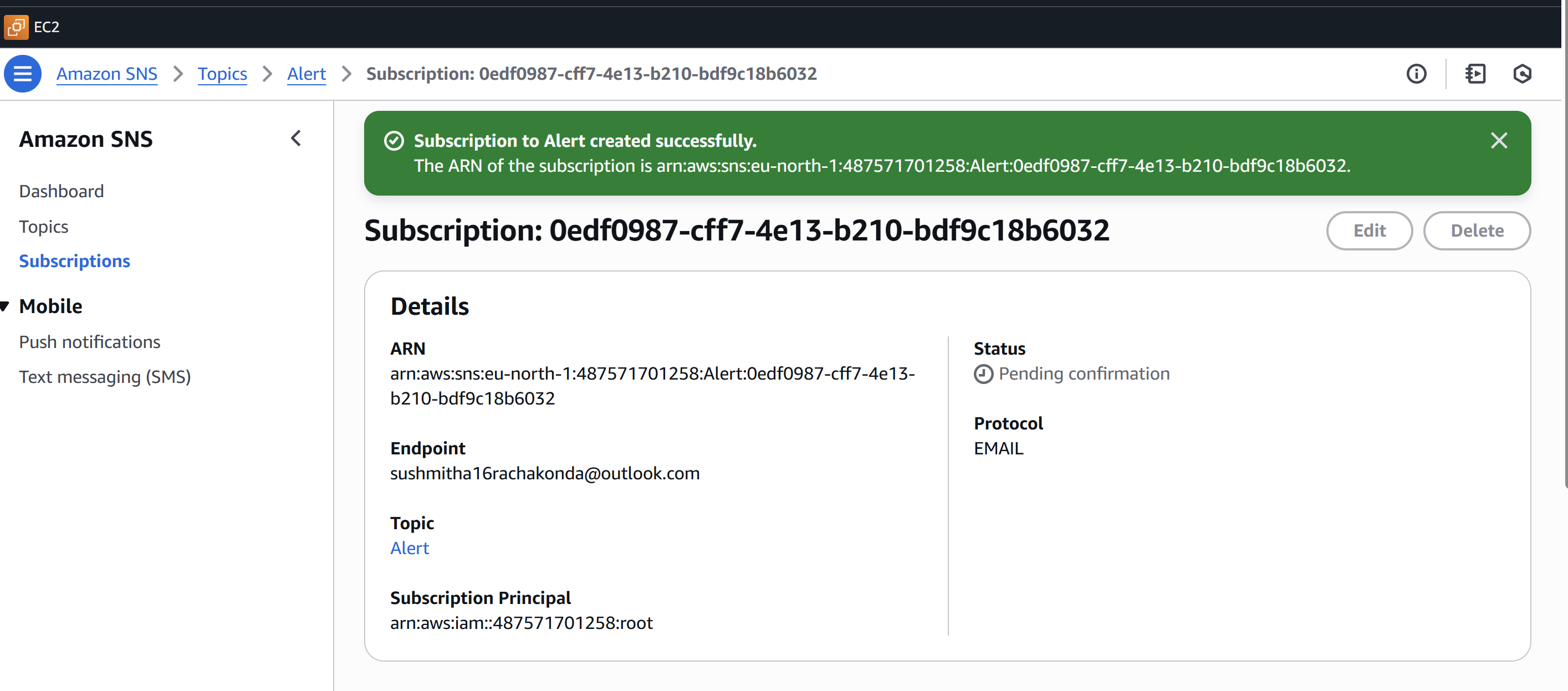
4.In the **Target** section:

* Select **SNS topic**
* Choose the topic: Alert

5.Click **Next**, then **Create rule**







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

**Step 1: Open CloudWatch Console**

1. Go to AWS Console → Search for **CloudWatch**
2. Click on **“Dashboards”**
3. Click **“Create dashboard”**
4. Enter a name: e.g., Cpuutilization
5. Click **“Create dashboard**

**Step 2: Add a Metric Widget**

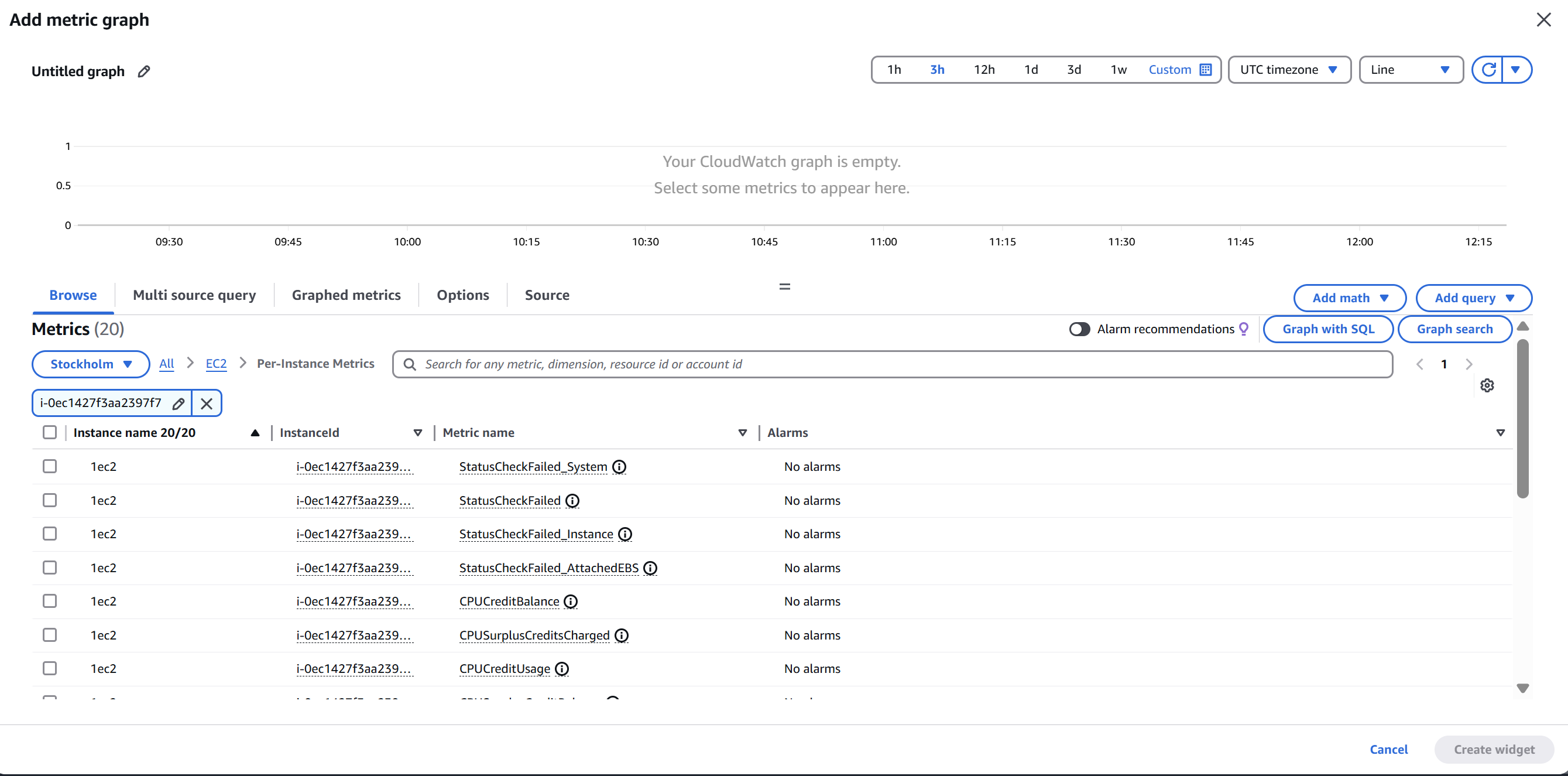
1. In the pop-up, choose **“Line”** or **“Number”** widget → Click **Next**
2. Click **Browse** (from your screenshot)
3. Select:
   * Region: Stockholm
   * Service: EC2
   * Category: Per-Instance Metrics
   * Filter by instance ID or name (e.g., i-0ec142f75aa2397f1)
4. Choose metrics:
   * CPUUtilization
   * NetworkIn
   * NetworkOut
   * MetadataNoToken
   * NetworkPacketsIn
   * NetworkPacketsOut
   * CPUCreditUsage, etc.
5. Click **“Create widget”**

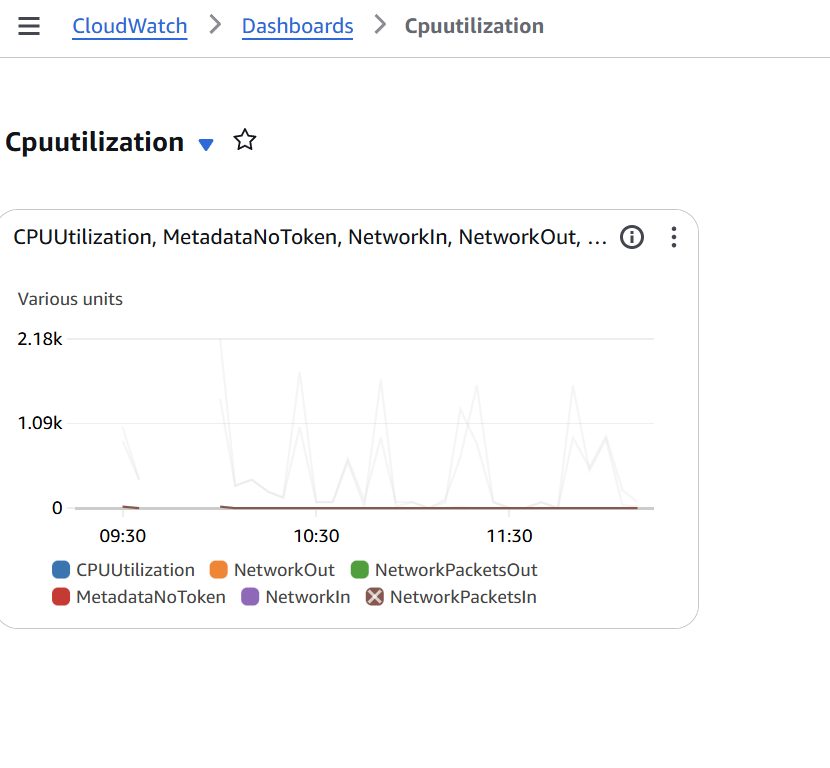
Step 3: View and Save the Dashboard

You will be taken back to your Dashboard view

The widget will now show a line graph for all selected metrics

Click “Save dashboard”





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

Step 1: Open CloudWatch Metrics for EC2

Go to CloudWatch Console → Metrics → Browse by EC2.

Choose your region (e.g., Stockholm).

Select Per-Instance Metrics → Click your instance ID (i-0ec142f75aa2397f7).

Check the box for CPUUtilization (and optionally NetworkIn, NetworkOut, etc.).

Click “Add to graph”.

**Step 2: Create an Alarm for CPUUtilization**

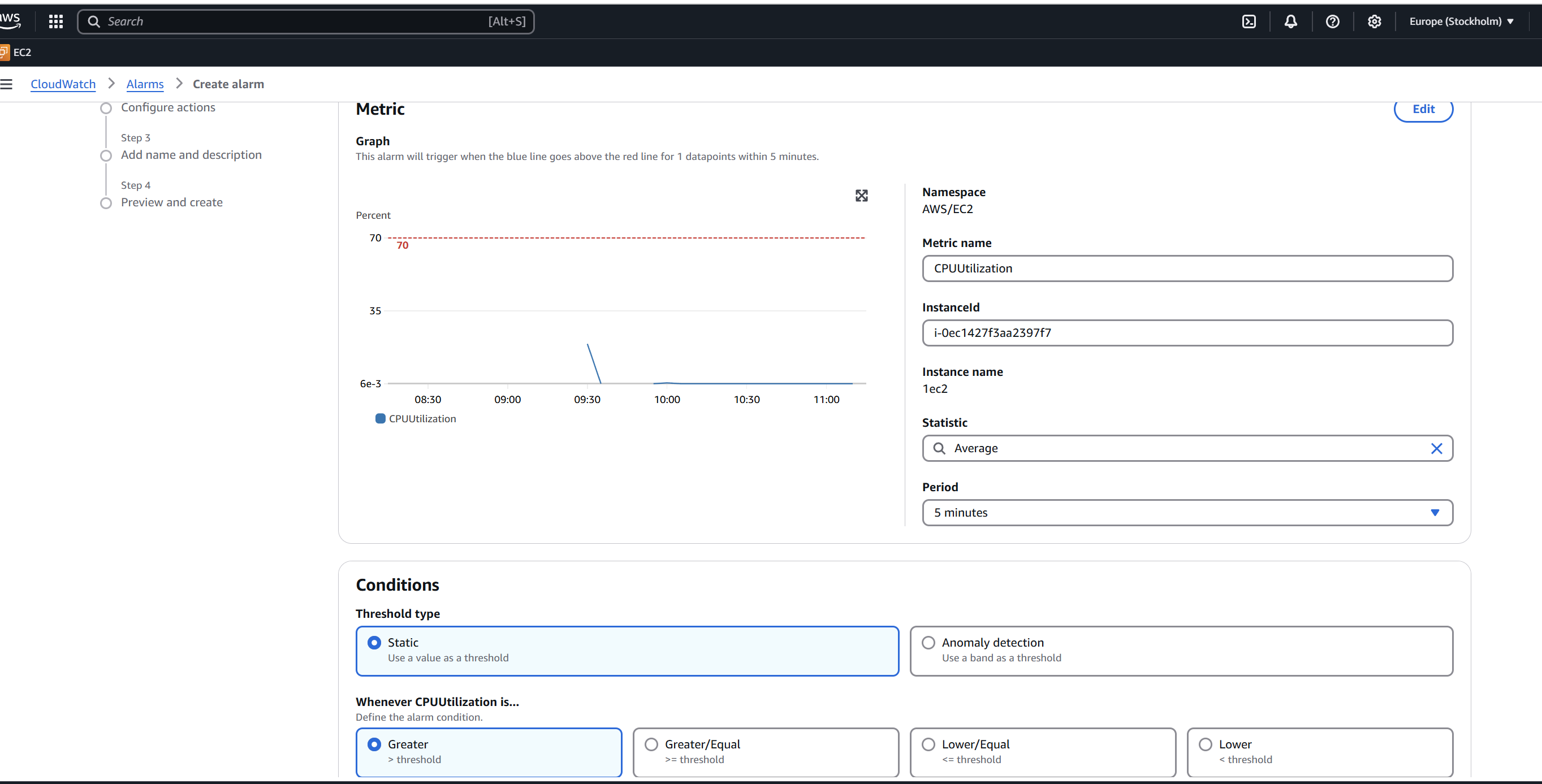
1. From the metrics graph, select **CPU Utilization** and click **“Create Alarm”**.
2. Set the **threshold conditions**:
   * **Threshold type**: Static
   * **Whenever CPUUtilization is**: Greater than 70
   * **Period**: 5 minutes
   * **Statistic**: Average

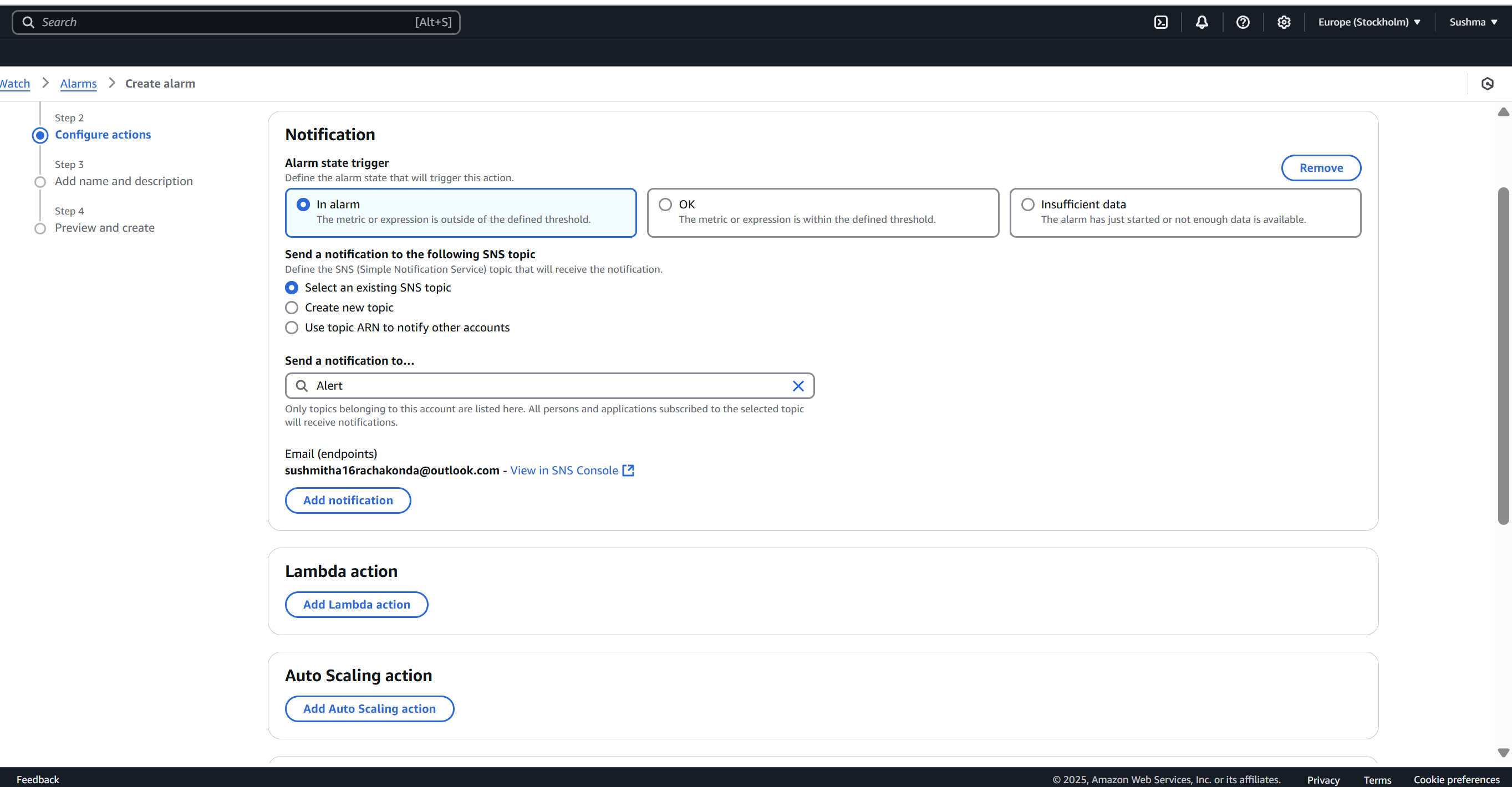
**Step 3: Configure Actions**

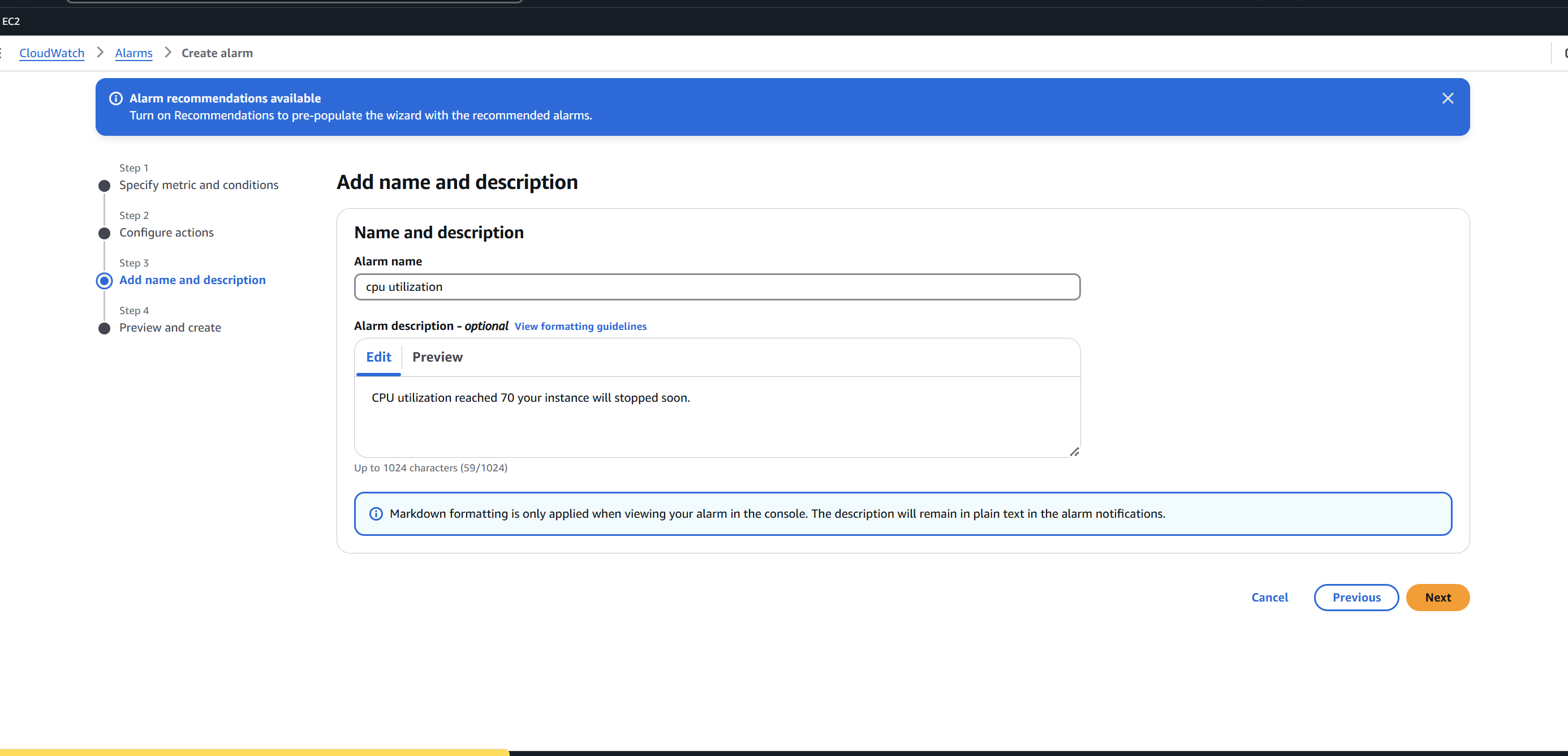
1. Under **Notification**:
   * Select **“In Alarm”**.
   * Choose an existing SNS topic or create a new one.
   * If creating new:
     + Set topic name (e.g., HighCPUAlertTopic)
     + Add your **email address** to receive notifications.
     + You must **confirm the subscription** via a link sent to your email.

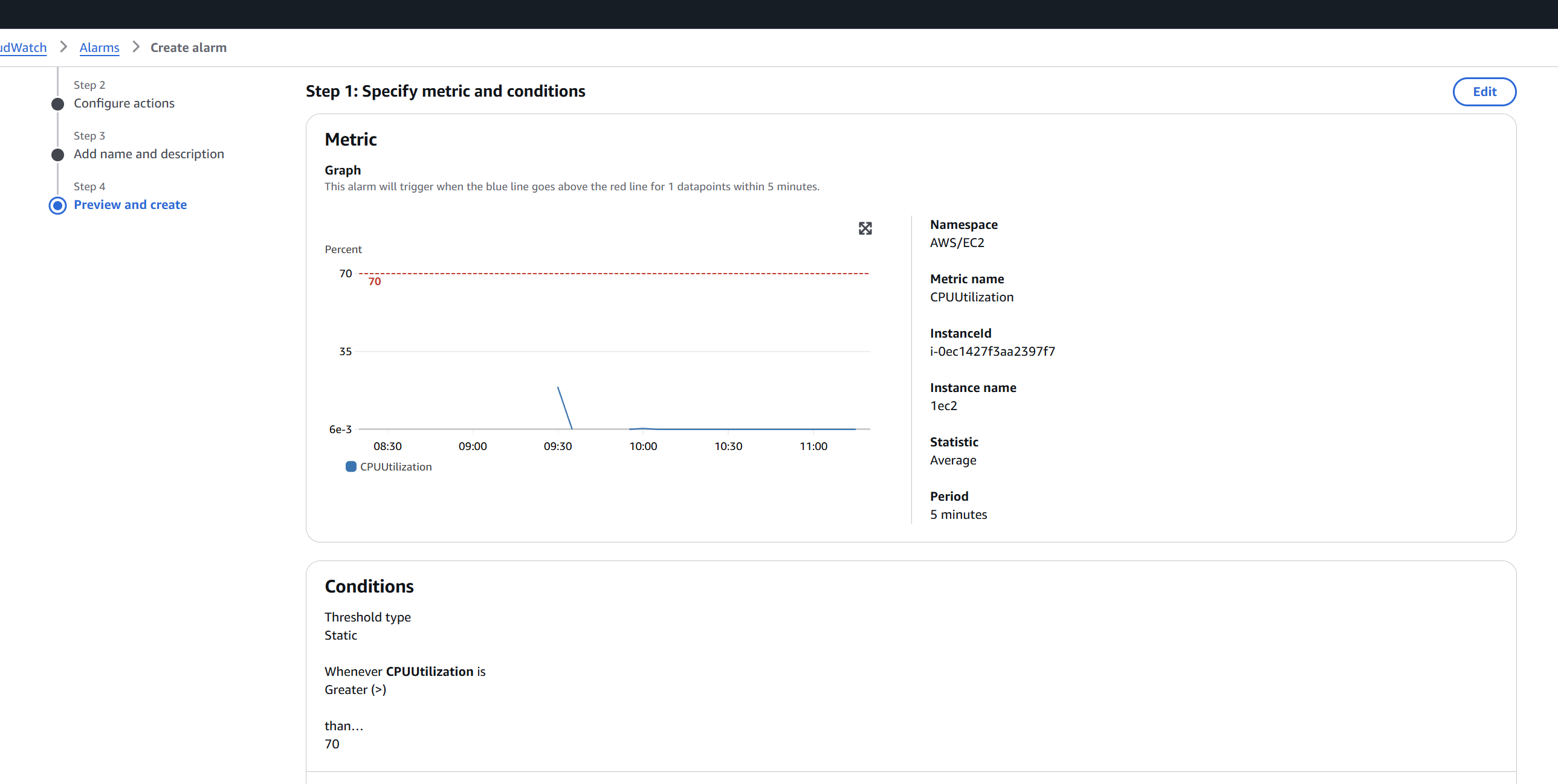
**Step 4: Add Alarm Name and Description**

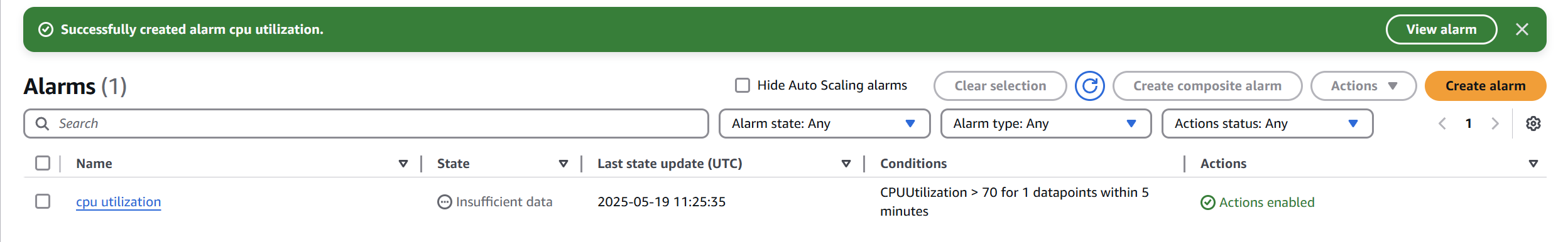
1. Name: cpu utilization
2. Description: e.g., *“CPU utilization reached 70%. Your instance will stop soon.”*











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

* Install CloudWatch Agent on the EC2 instance
* Create Shell Script to Monitor Tomcat

Create the /opt directory (if it doesn't exist)

sudo mkdir -p /opt

cd /opt

* Create/Edit the Tomcat check script

sudo vi check\_tomcat.sh

#!/bin/bash

# Check Tomcat status

TOMCAT\_PORT=8080

TOMCAT\_HOST=localhost

STATUS=$(curl -s -o /dev/null -w "%{http\_code}" http://${TOMCAT\_HOST}:${TOMCAT\_PORT})

if [ "$STATUS" -eq 200 ]; then

METRIC\_VALUE=1

else

METRIC\_VALUE=0

fi

# Get IMDSv2 token

TOKEN=$(curl -s -X PUT "http://169.254.169.254/latest/api/token" \

-H "X-aws-ec2-metadata-token-ttl-seconds: 21600")

# Use token to get instance ID

INSTANCE\_ID=$(curl -s -H "X-aws-ec2-metadata-token: $TOKEN" \

http://169.254.169.254/latest/meta-data/instance-id)

if [ -z "$INSTANCE\_ID" ]; then

echo "Error: Could not retrieve instance ID"

exit 1

fi

# Push to CloudWatch

aws cloudwatch put-metric-data \

--metric-name TomcatStatus \

--namespace "Custom/Tomcat" \

--value "$METRIC\_VALUE" \

--dimensions InstanceId="$INSTANCE\_ID"

* sudo chmod +x /opt/check\_tomcat.sh
* sudo /opt/check\_tomcat.sh
* **IAM Role Permissions for EC2**

{

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

"Statement": [

{

"Effect": "Allow",

"Action": "cloudwatch:PutMetricData",

"Resource": "\*"

}

]

}

* Attach the role to the EC2 instance.
* Create CloudWatch Alarm
* Go to **CloudWatch > Alarms > Create Alarm**:

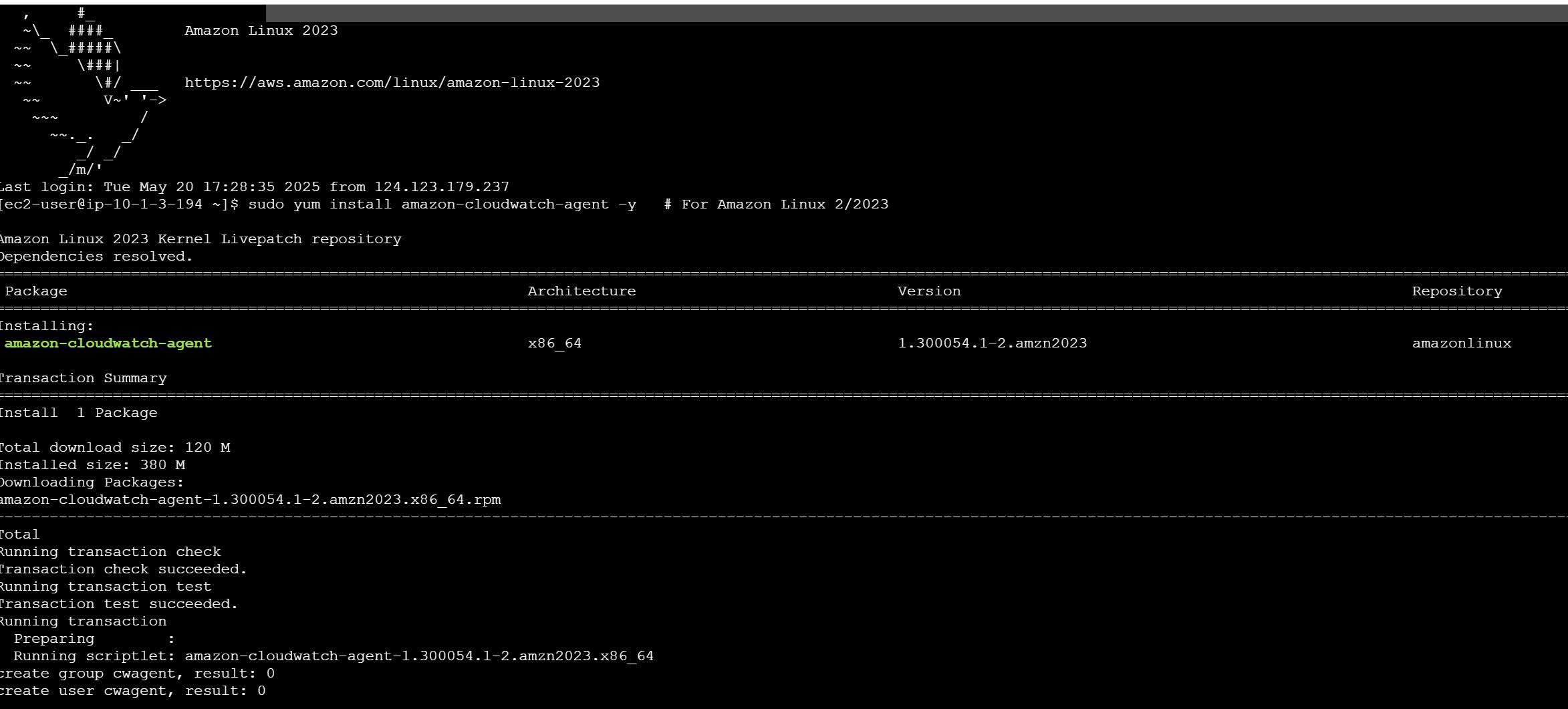
**Metric namespace**: Custom/Tomcat

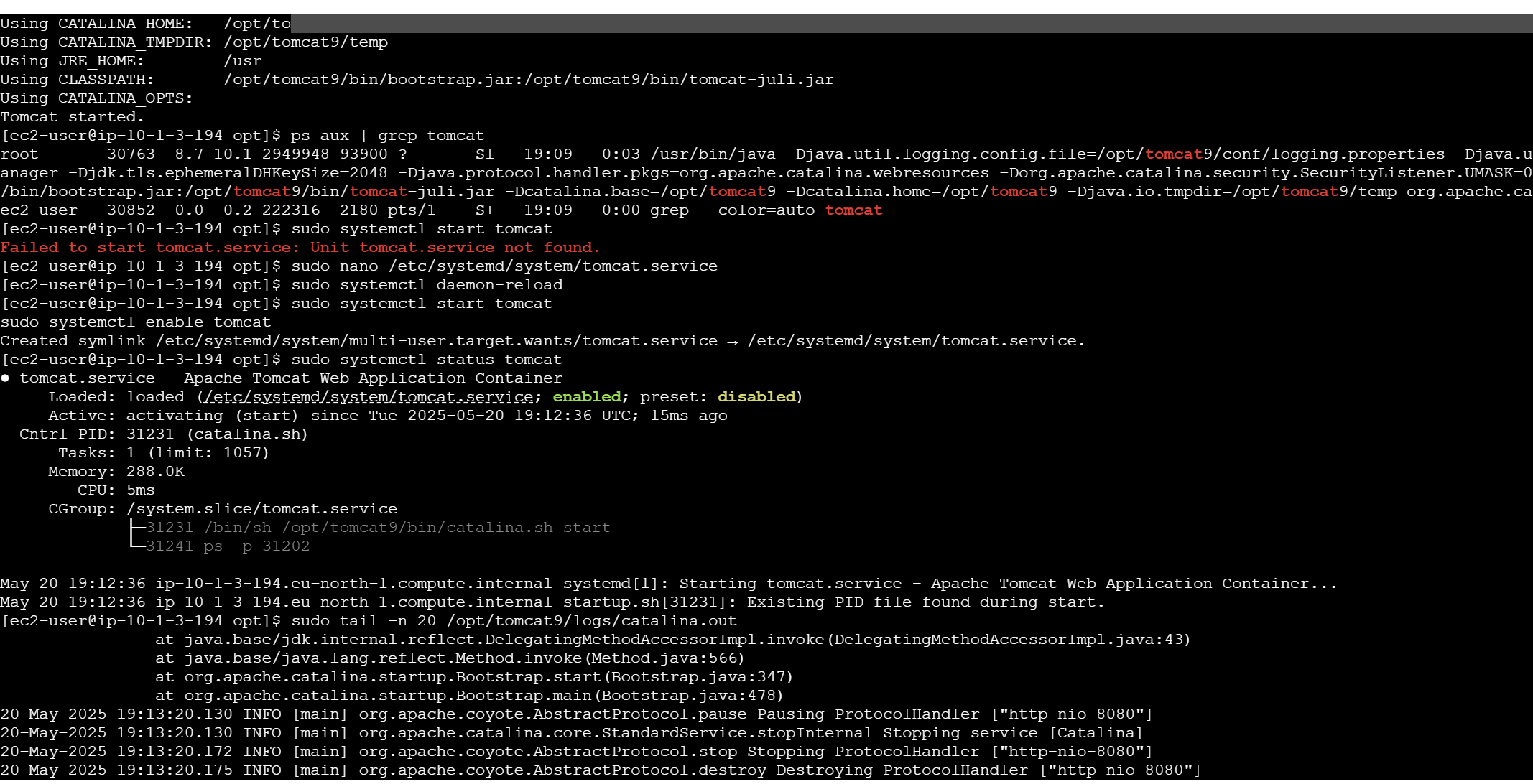
**Metric name**: TomcatStatus

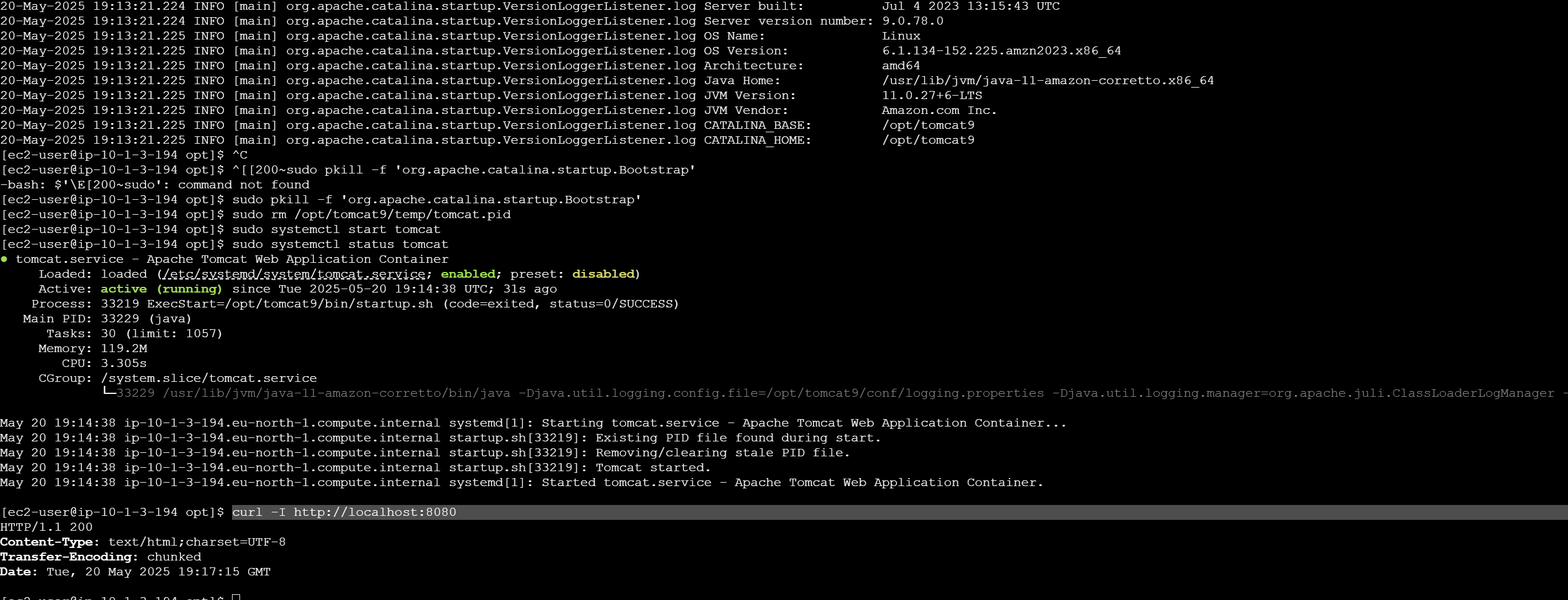
**Statistic**: Minimum

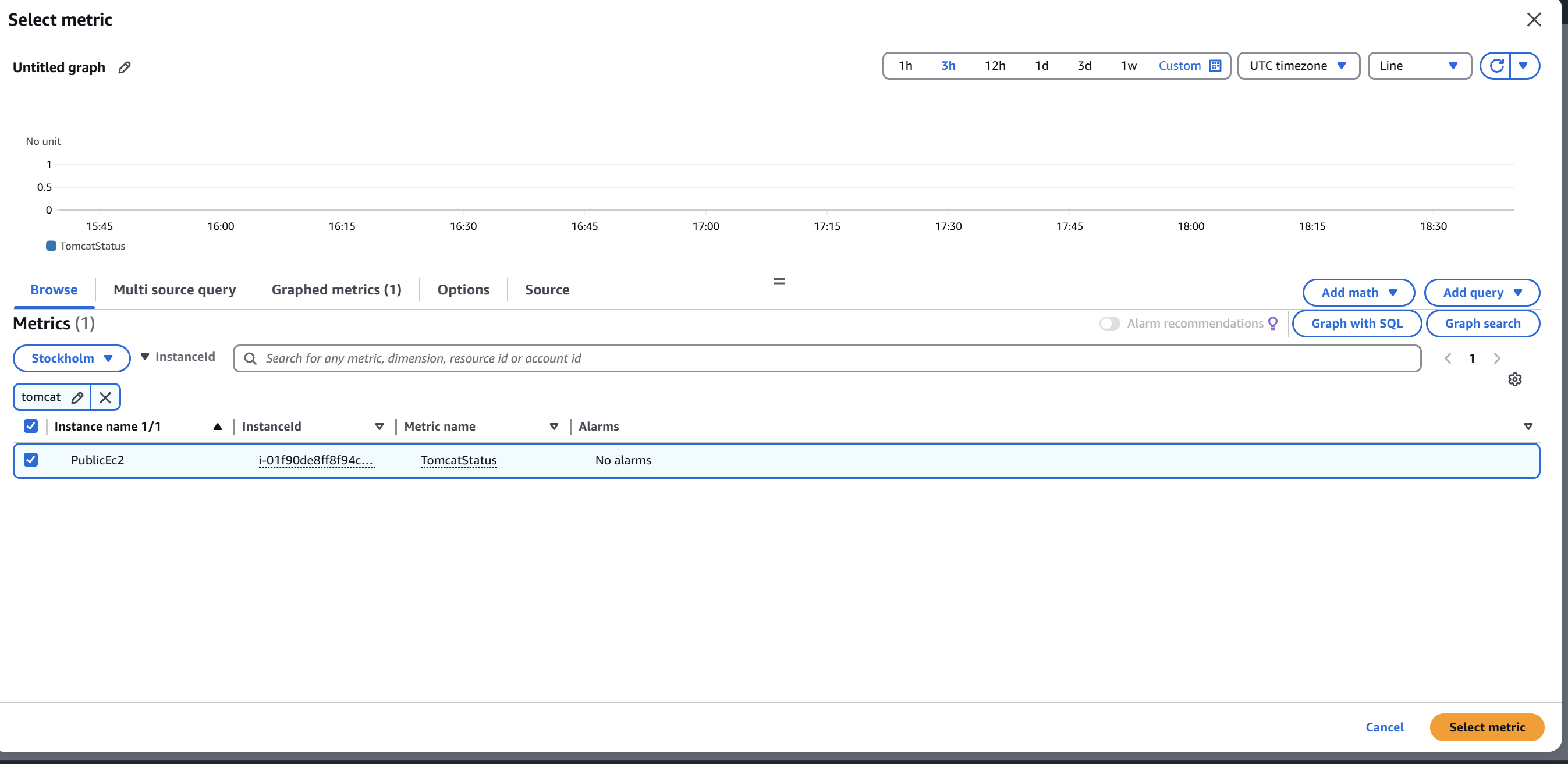
**Threshold**: If TomcatStatus < 1 for 1 datapoint within 1 minute

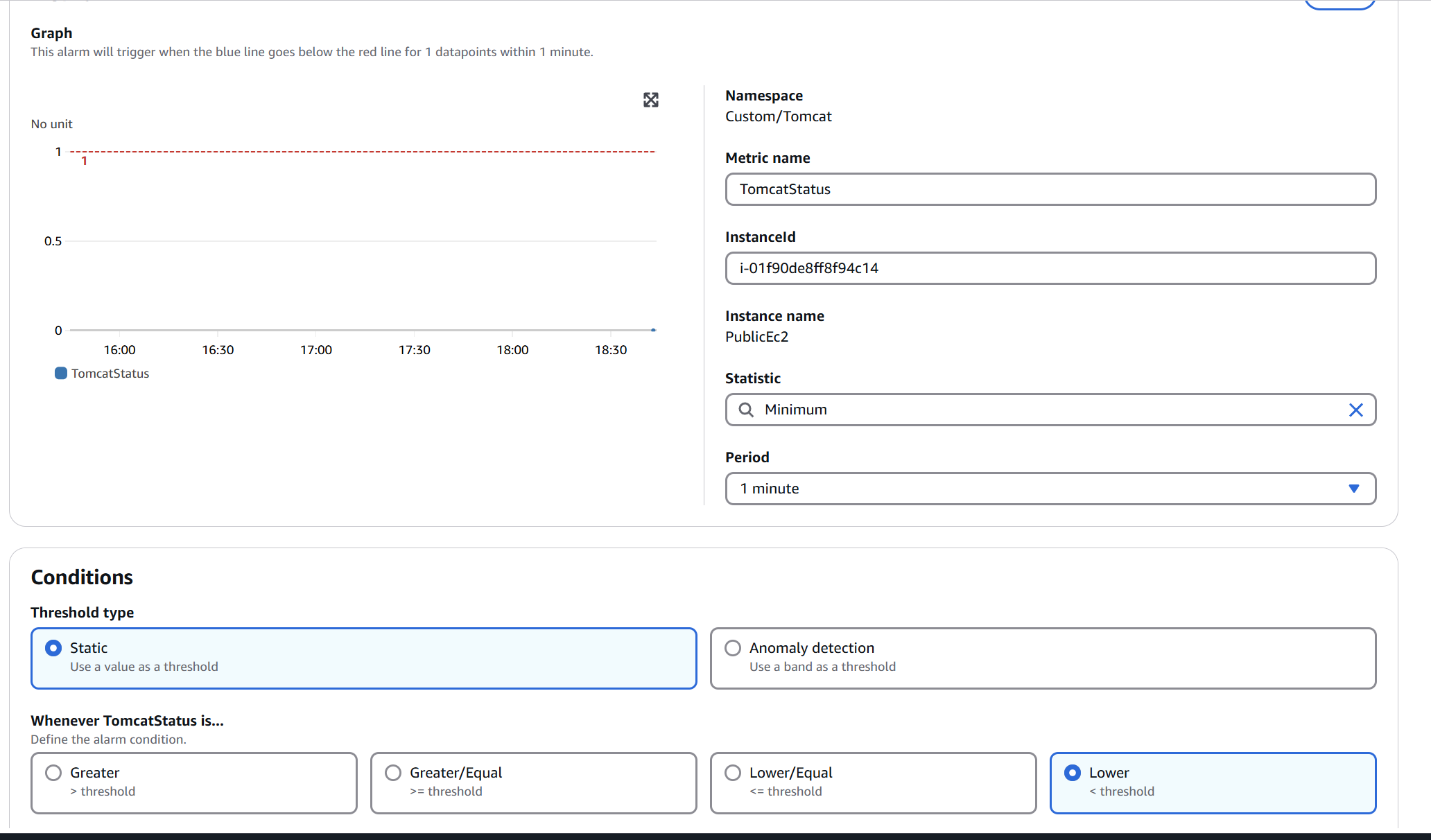
**Action**: Choose SNS topic (or create one to receive email/SMS)

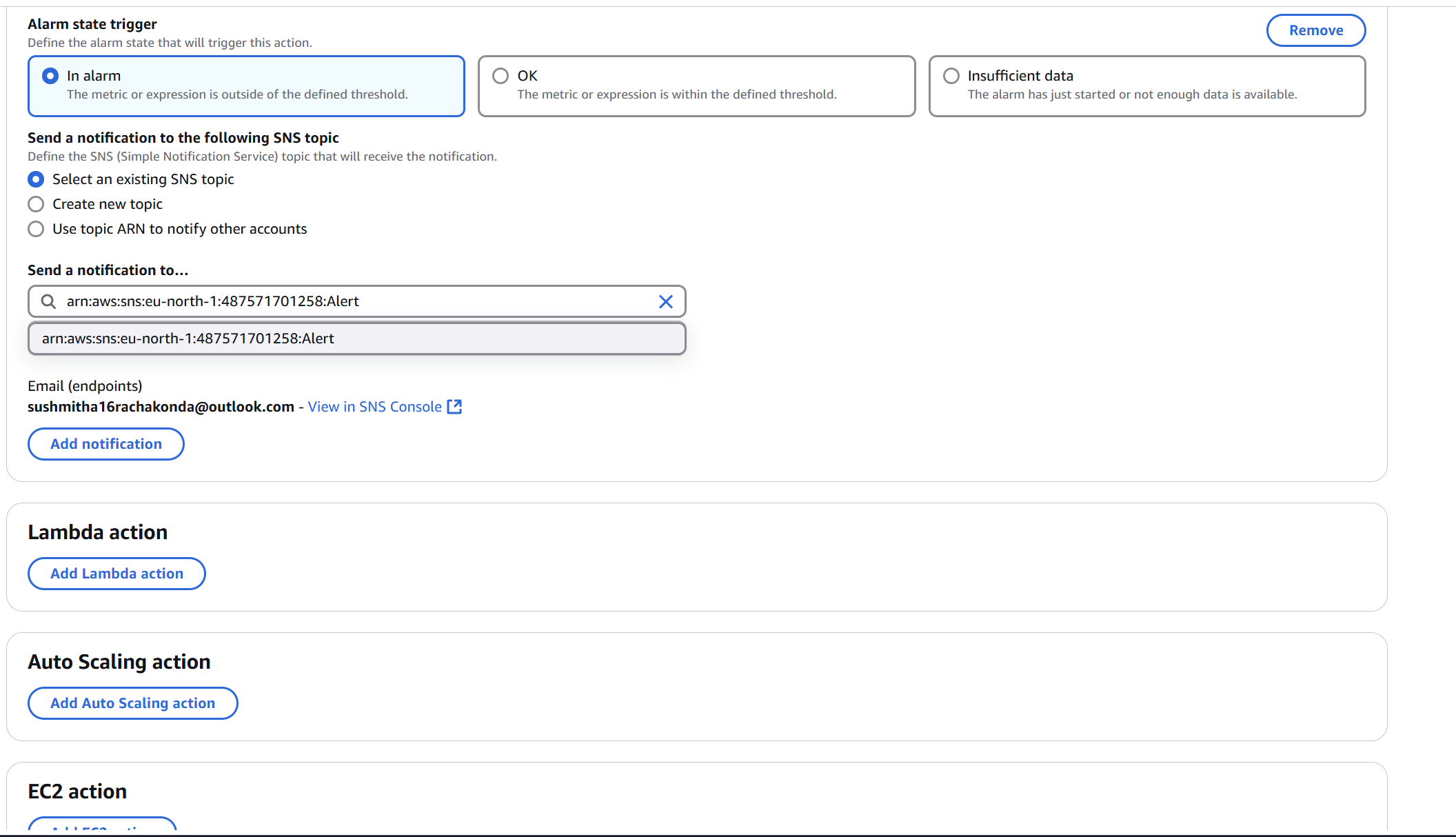


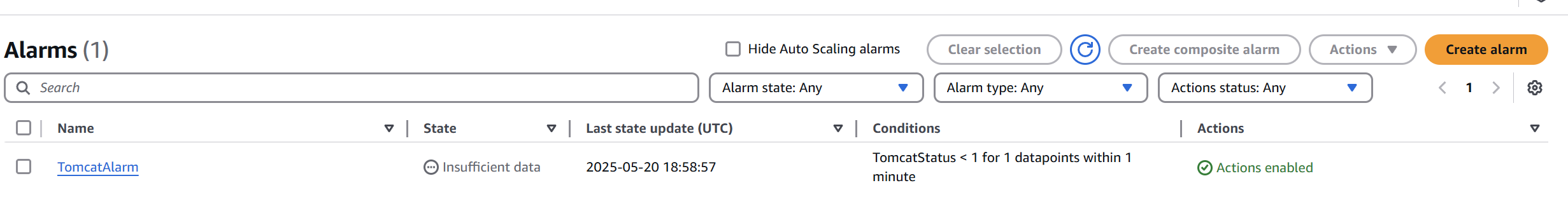












6) Create Dashboard and monitor nginx service to send the alert if nginx is not running.

* **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)

