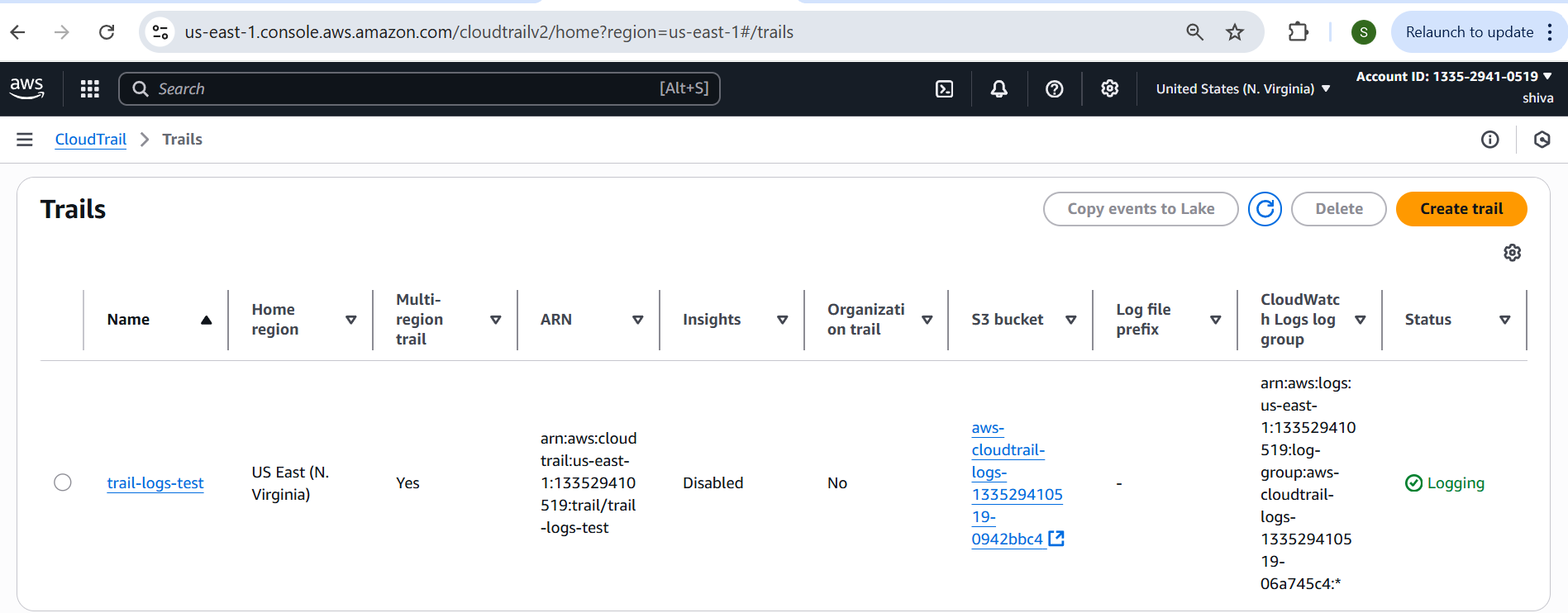
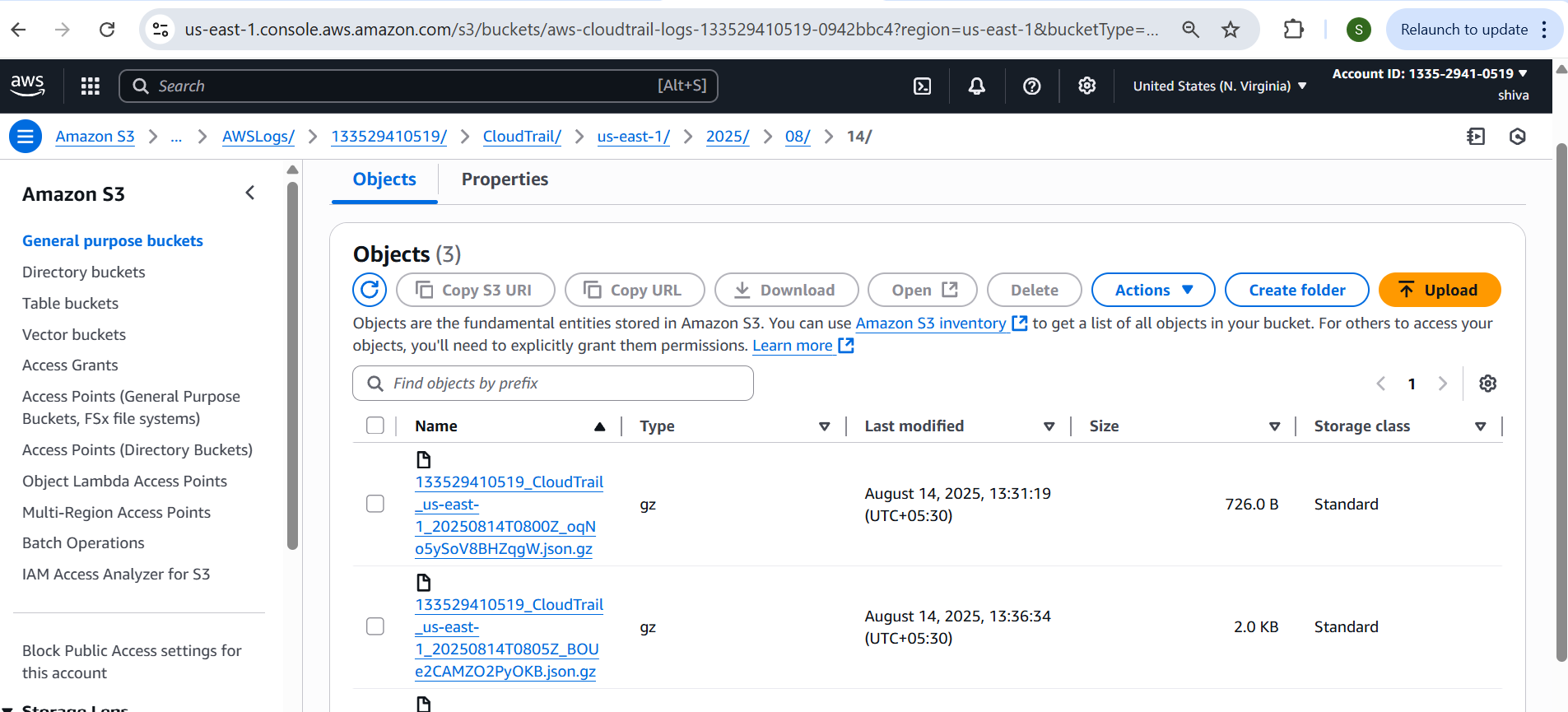
1) Enable cloudtrail monitoring and store the events in s3 and cloudwatch log events.

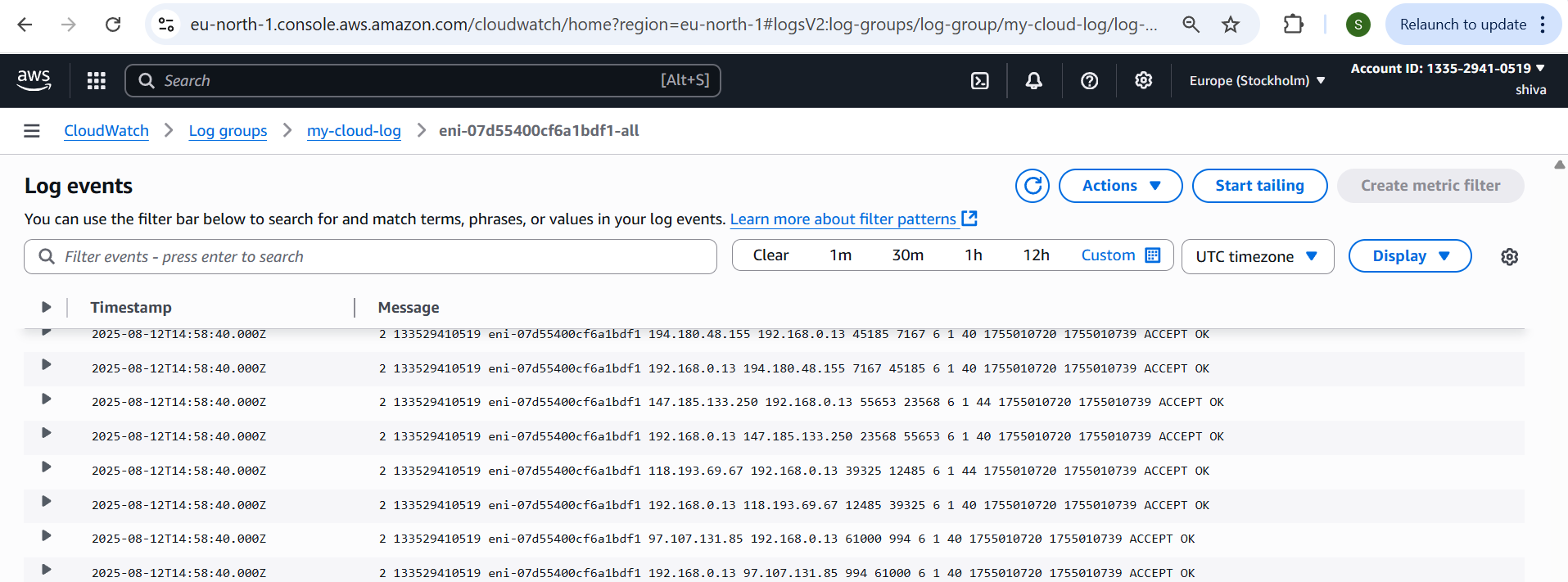
Go to CloudTrail  
1. Open the AWS Console  
2. Search for CloudTrail in the Services menu  
3. Click “Create trail”

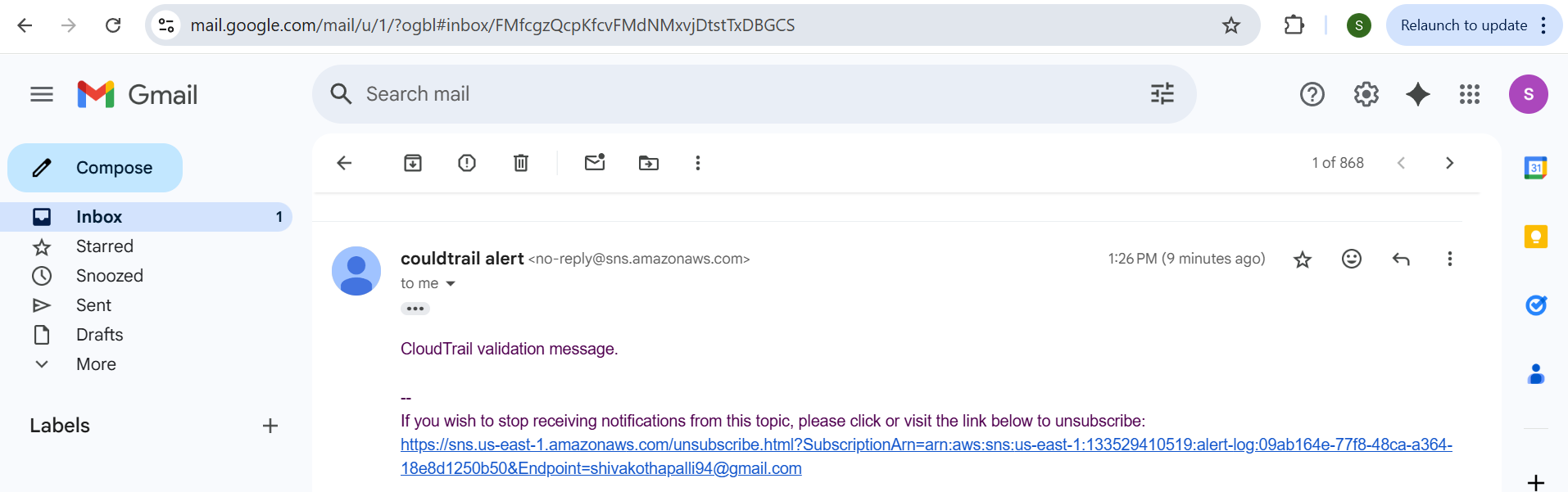
Create new S3 bucket or use an existing one

Enable CloudWatch Logs integration







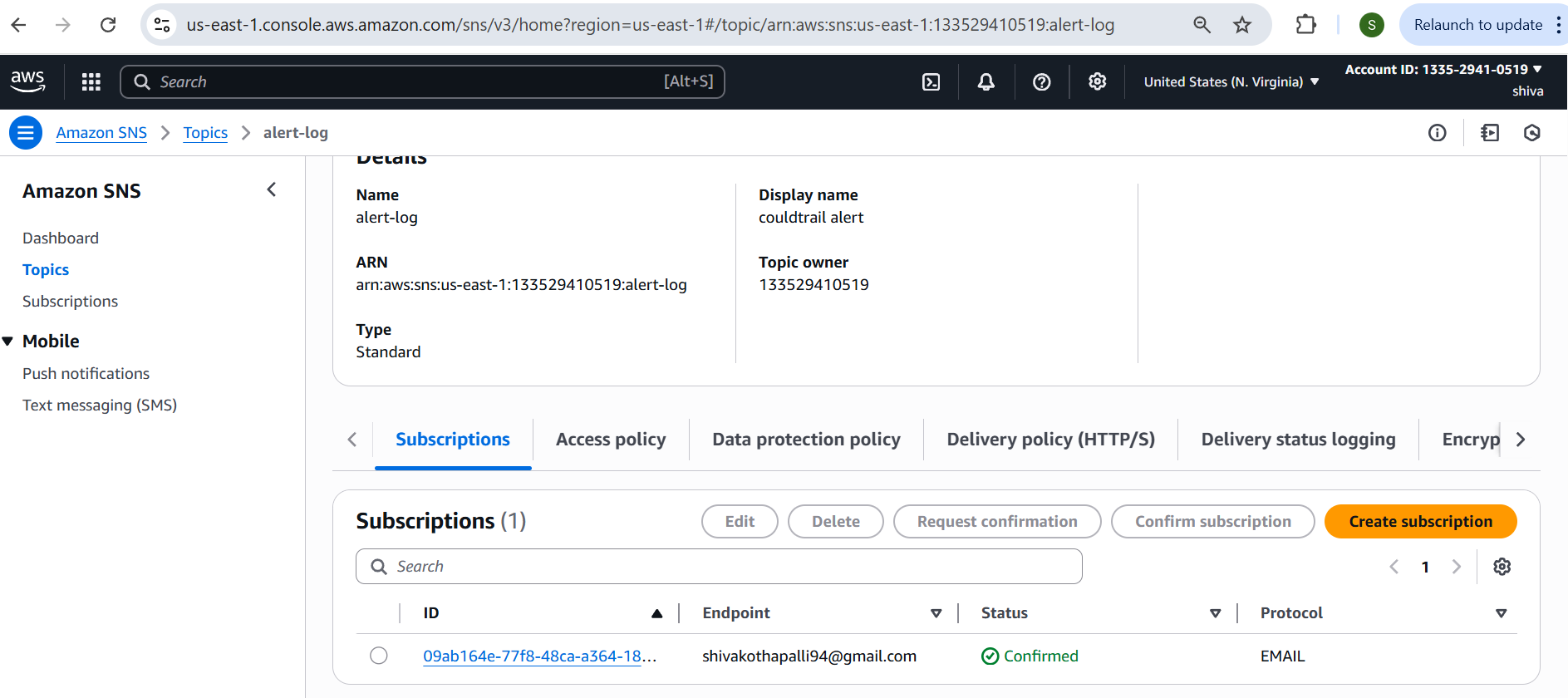


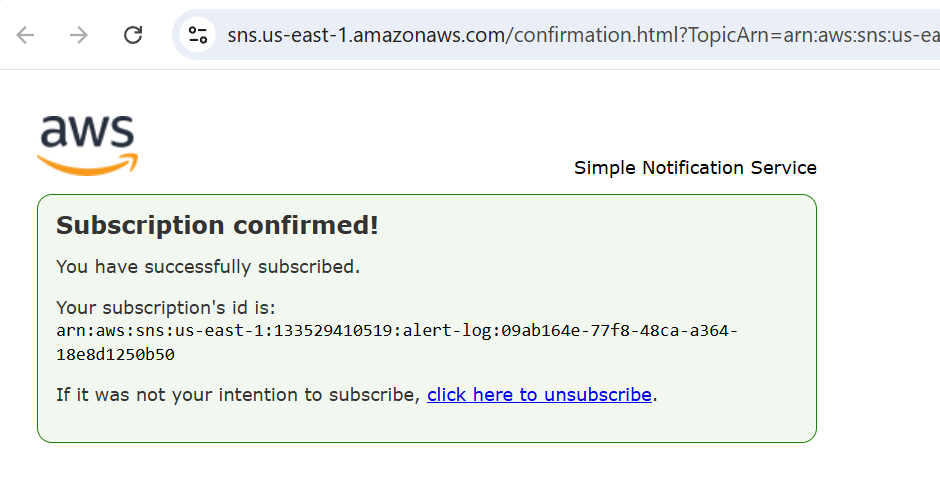
2) Enable SNS for cloudtrial to send alert on email.

Create an SNS Topic  
1. Go to SNS → Topics  
2. Click Create topic  
3. Select Standard  
4. Enter Name (e.g., cloudtrail-alerts)  
5. Click Create topic

Amazon SNS ---> topics ---> select your topic ---> subsciption  
--->create subscription

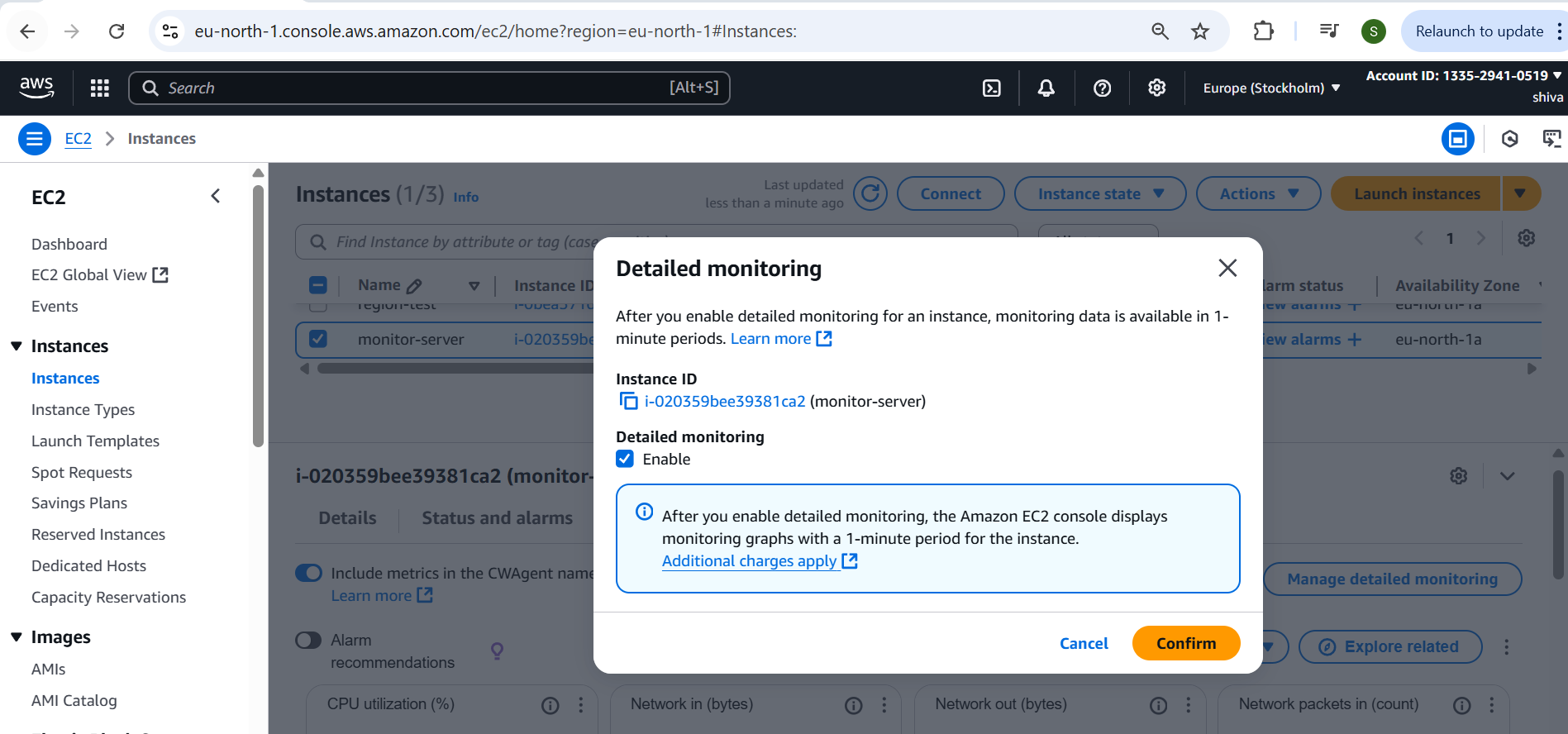
Go to email to confirm the subscription

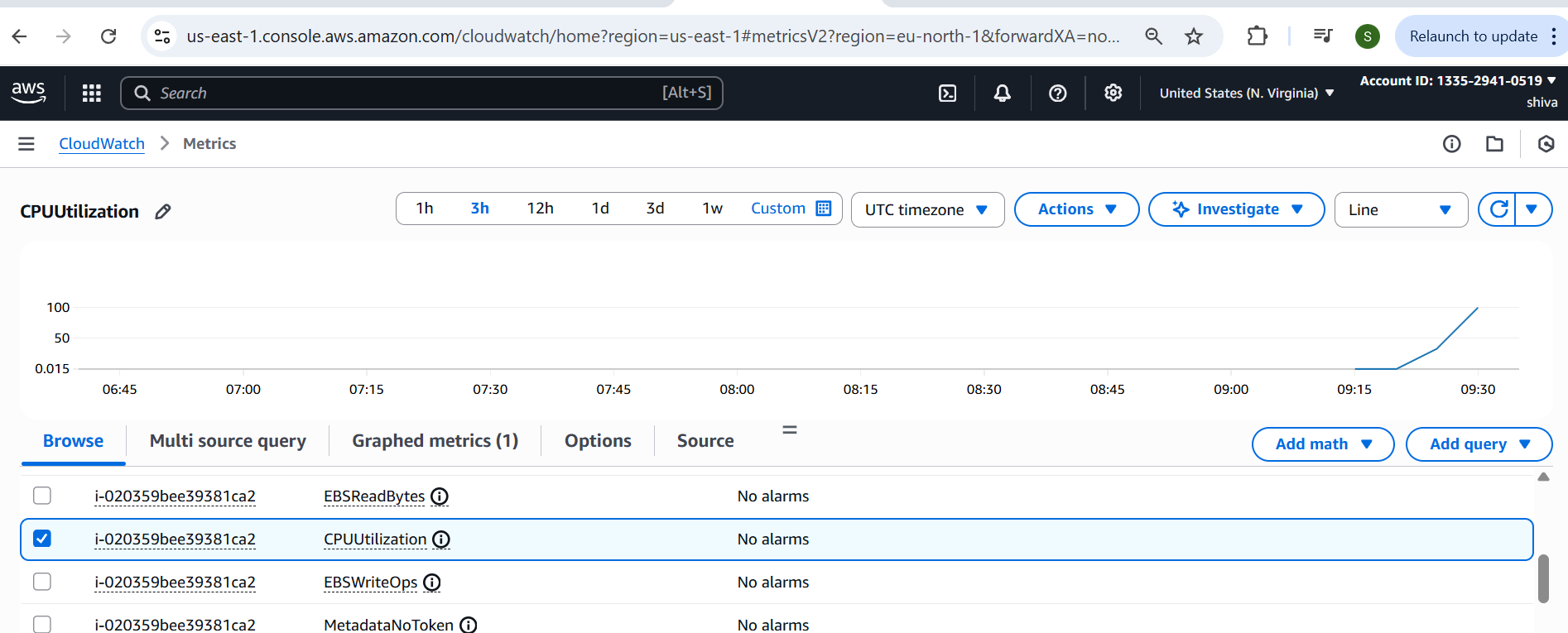




3) Configure cloud watch monitoring and record the cpu utilization and other metrics of ec2.

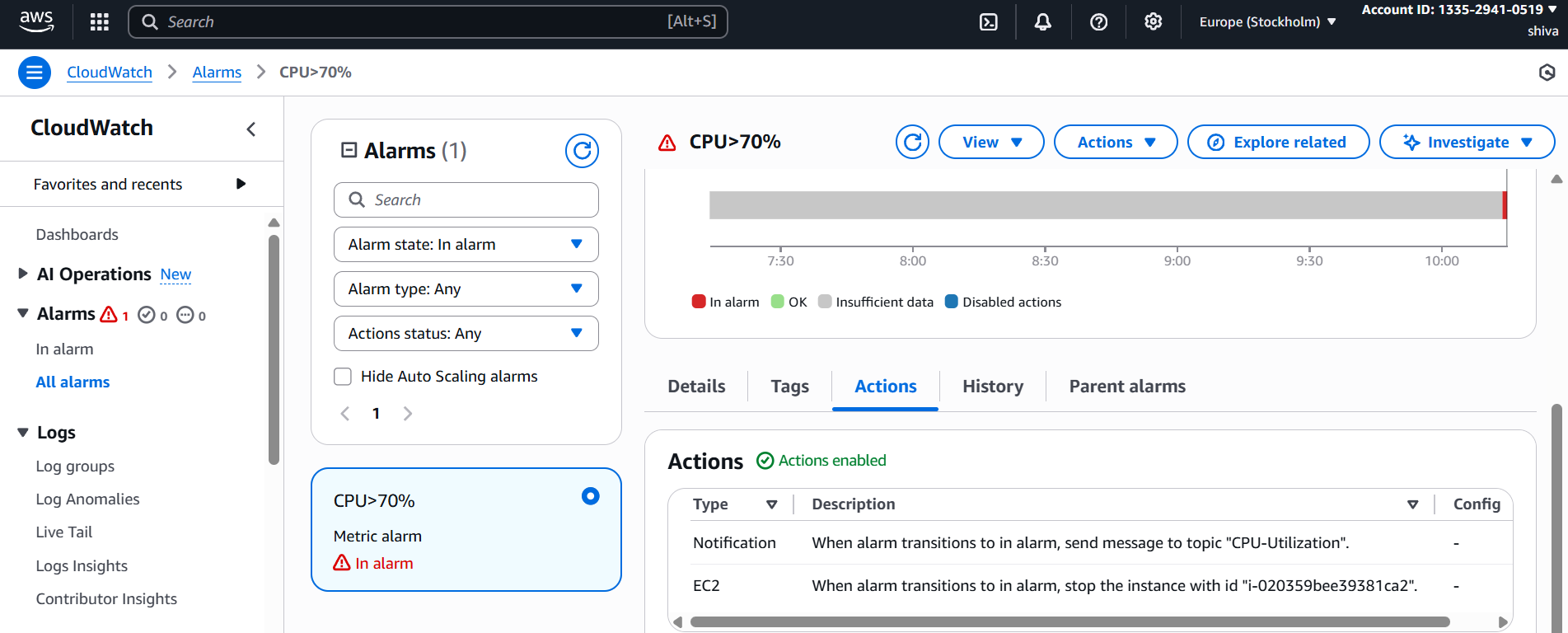
1. Go to EC2 → Instances  
2. Select the instance you want to monitor  
Enable Detailed Monitoring (Optional for 1-min data)  
1. In the Details tab, check Monitoring  
2. Click Actions → Monitor and troubleshoot → Manage detailed

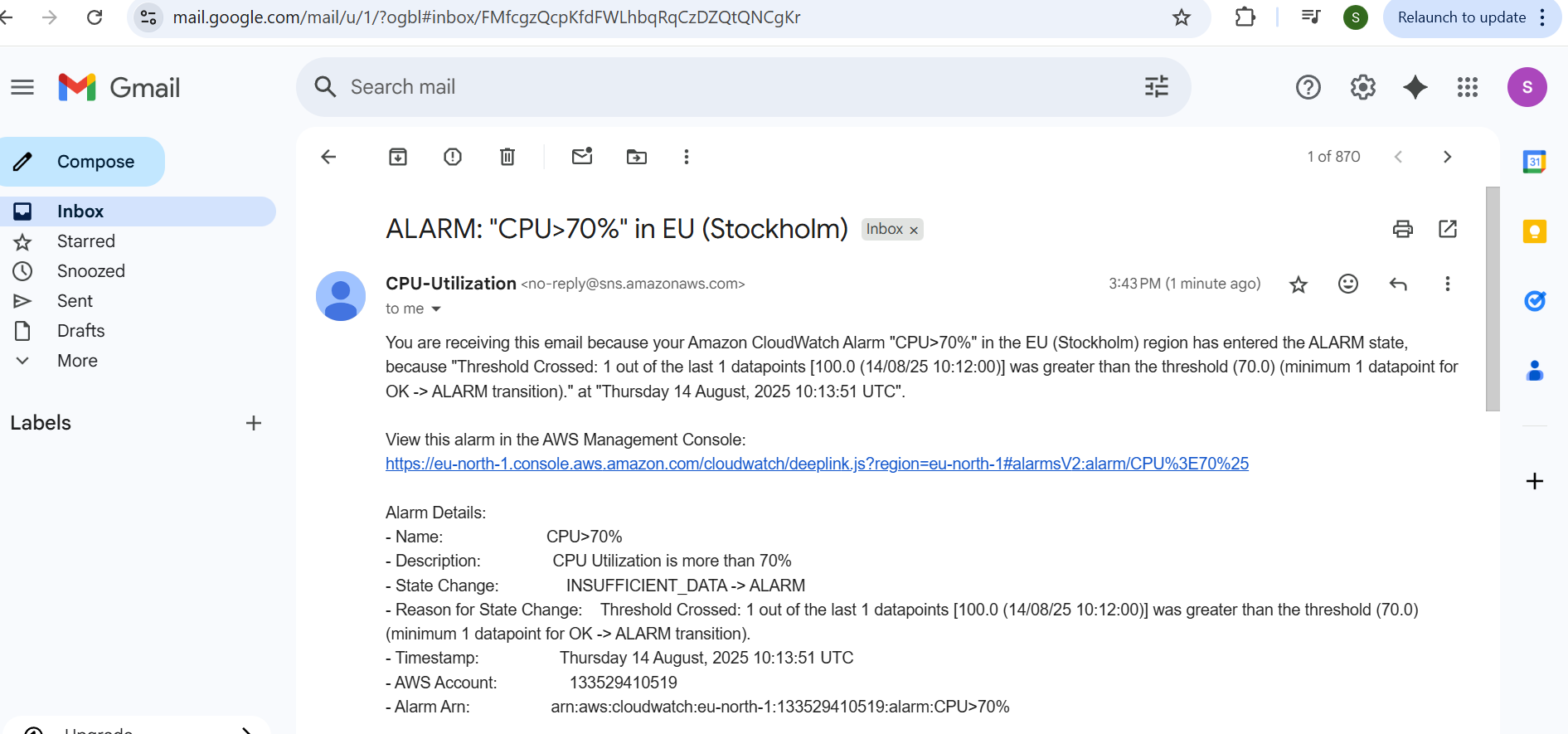


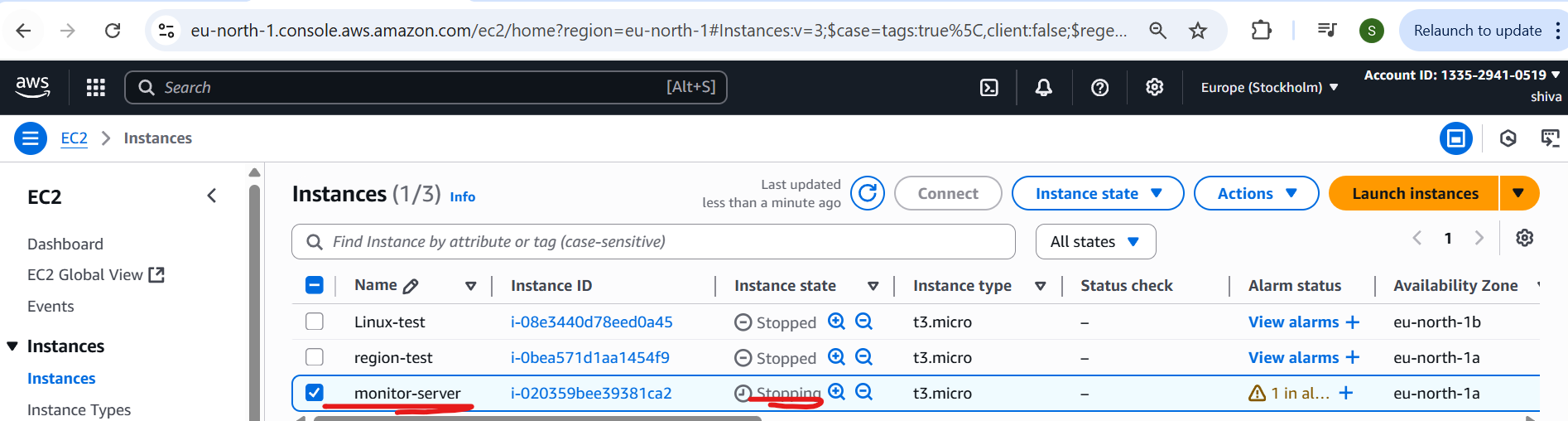


4) Create one alarm to send alert to email if the cpu utilization is more than 70 percent.

Create CloudWatch Alarm for EC2 CPU  
Cloudwatch ---> Alarms ---> create alarm --->select metric --->  
and select datapoint 1 → condition static → select SNS (if not then   
create) → create







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

Install tomcat

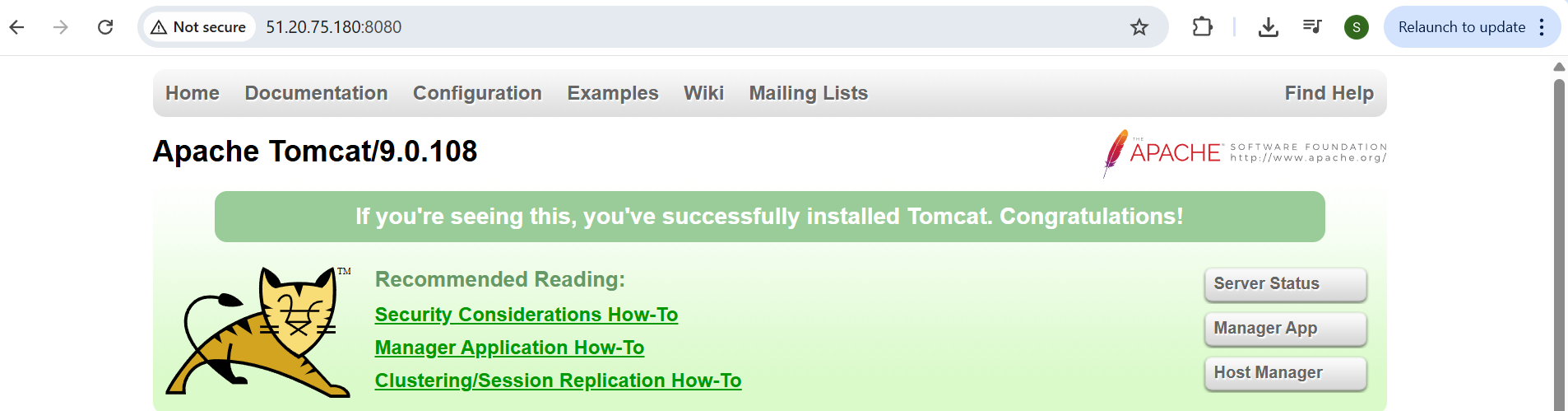
Create Custom Metric for Tomcat  
1. Create a shell script like:  
Bash  
 CopyEdit  
#!/bin/bash  
if systemctl is-active --quiet tomcat; then  
 /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl   
\  
 -a put-metric-data -m "TomcatStatus" -n "TomcatRunning" -v 1  
else  
 /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl   
\  
 -a put-metric-data -m "TomcatStatus" -n "TomcatRunning" -v 0

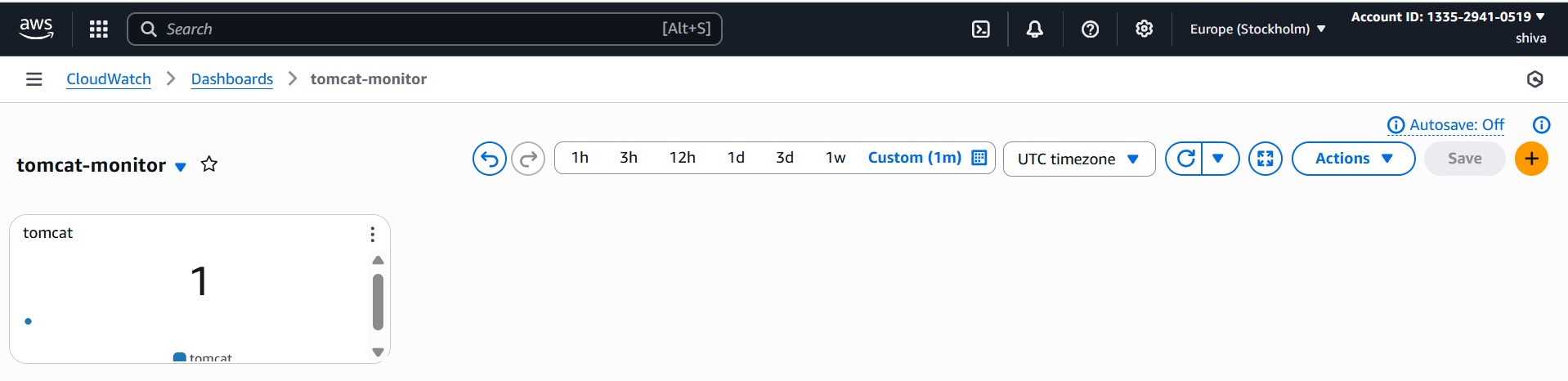
Fi

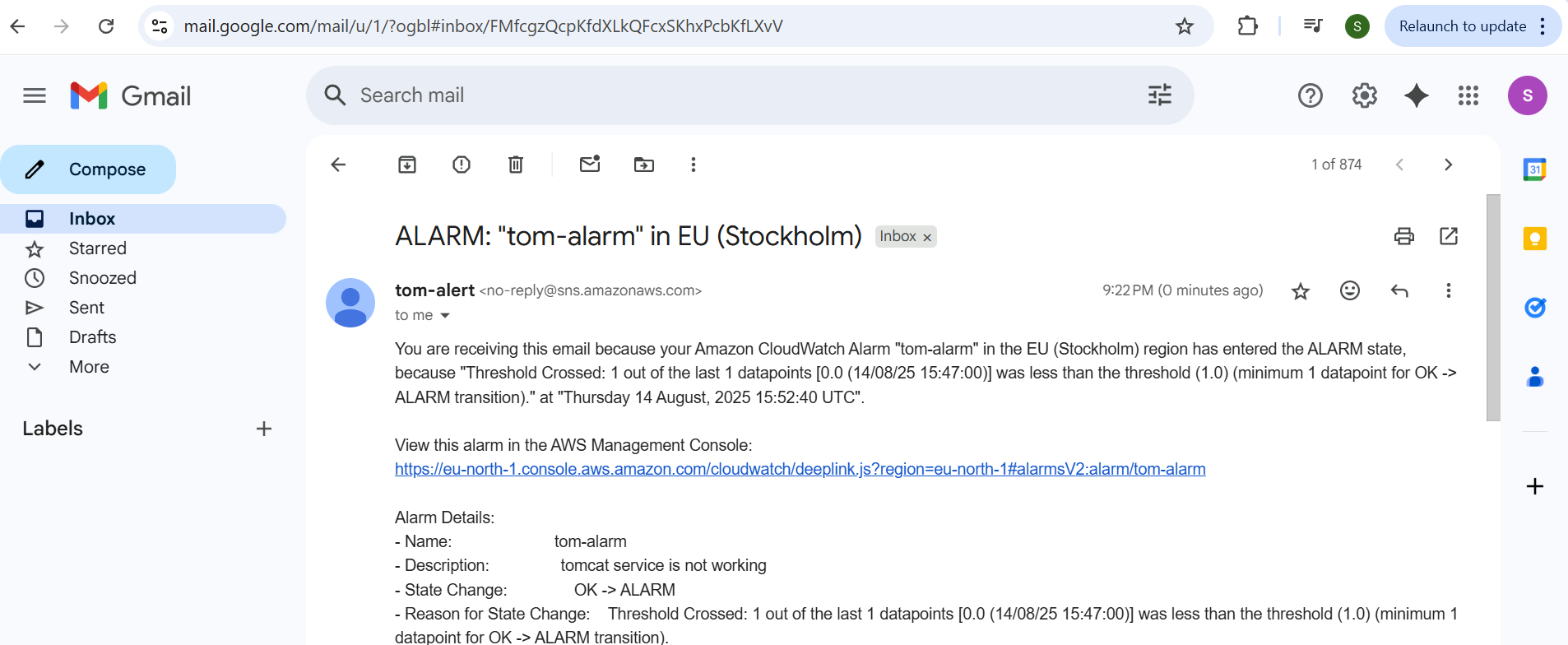
2. Schedule it via cron to run every 1 minute:  
bash  
CopyEdit  
\* \* \* \* \* /root/tomcat.sh

Go to CloudWatch → Dashboards  
2. Click Create dashboard  
3. Name the dashboard (e.g., Tomcat-Monitoring)  
4. Add “Number” widget  
Select the TomcatRunning metric  
Shows 1 if running, 0 if not

1. Go to CloudWatch → Alarms → Create alarm  
2. Select custom namespace: TomcatStatus  
3. Choose metric: TomcatRunning  
4. Set threshold: is equal to 0 (Tomcat not running)  
5. Choose SNS topic for email alerts  
6. Create alarm







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

Install nginx

Create a Script to Check Nginx Status  
1. Create a shell script:  
bash  
CopyEdit  
#!/bin/bash  
if systemctl is-active --quiet nginx; then  
 aws cloudwatch put-metric-data --metric-name NginxRunning --  
namespace "NginxStatus" --value 1  
else  
 aws cloudwatch put-metric-data --metric-name NginxRunning --  
namespace "NginxStatus" --value 0  
fi  
2. Save as: /opt/nginx.sh  
3. Give execute permission:  
bash  
CopyEdit  
chmod +x /opt/nginx.sh  
Step 3: Schedule the Script to Run Regularly  
1. Open crontab:  
bash

crontab –e

\* \* \* \* \* /opt/nginx\_status.sh

Create CloudWatch Dashboard  
1. Go to CloudWatch → Dashboards  
2. Click Create dashboard  
3. Name your dashboard (e.g., nginx-monitor)  
4. Add a Number or Line widget  
Select metric: NginxRunning under NginxStatus namespace

Go to CloudWatch → Alarms → Create alarm  
2. Select namespace: Nginx  
3. Metric: NginxRunning  
4. Set condition:  
Threshold: is equal to 0  
Period: 1 minute  
5. Set notification:  
Choose or create SNS topic  
Add your email and confirm the subscription  
6. Click Create alarm

