**CLOUDWATCH AND CLOUDTRAIL TASKS**

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

**🔹 Step 1: Go to CloudTrail**

1. Open the **AWS Console**
2. Search for **CloudTrail** in the Services menu
3. Click **“Trails”** on the left
4. Click **“Create trail”**

**🔹 Step 2: Configure Trail**

1. **Trail name**: Give a name (e.g., my-trail)
2. **Apply trail to all regions**: ✅ Yes (recommended)
3. **Management events**: ✅ Read/Write events (recommended)
4. **Data events** (optional): Enable if you want S3 or Lambda tracking

**🔹 Step 3: Choose S3 Bucket for Log Storage**

1. Select **Create new S3 bucket** or use an existing one
2. If creating a new one:
   * Bucket name: cloudtrail-logs-yourname
   * CloudTrail will automatically set permissions

**🔹 Step 4: Send Logs to CloudWatch Logs**

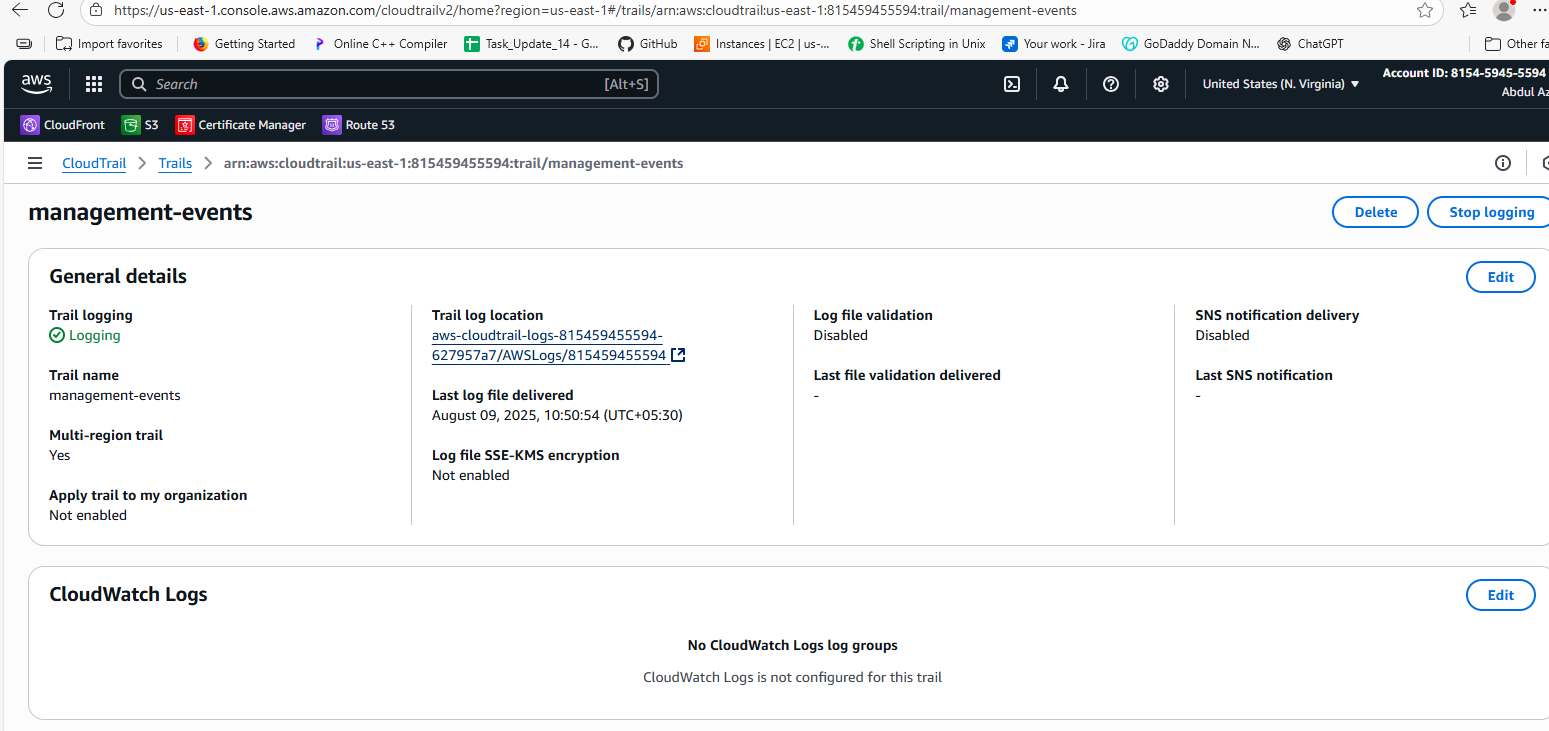
1. **Enable CloudWatch Logs integration** ✅
2. Choose **Create new log group** (e.g., /aws/cloudtrail/mytrail)
3. CloudTrail needs permission to send logs:
   * Click **“Create a new IAM role”**
   * It will create a role like: CloudTrail\_CloudWatchLogs\_Role

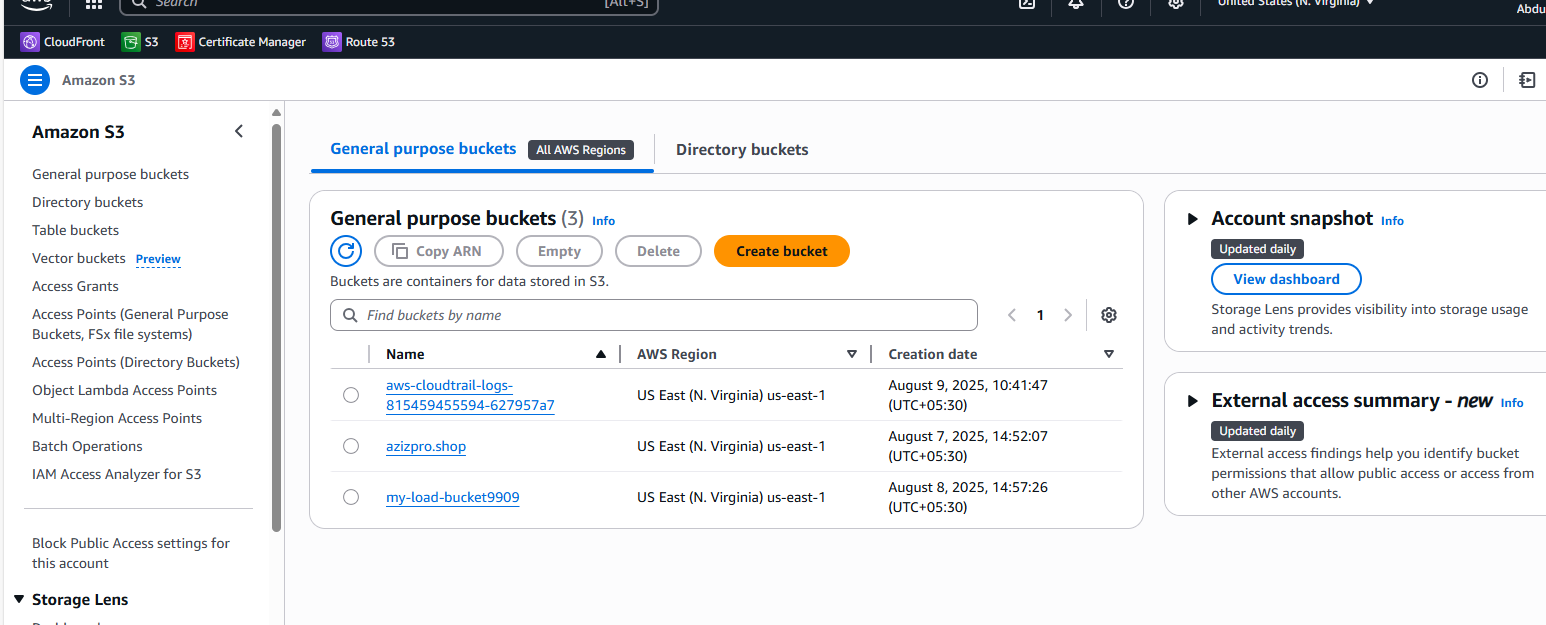
**🔹 Step 5: Review and Create**

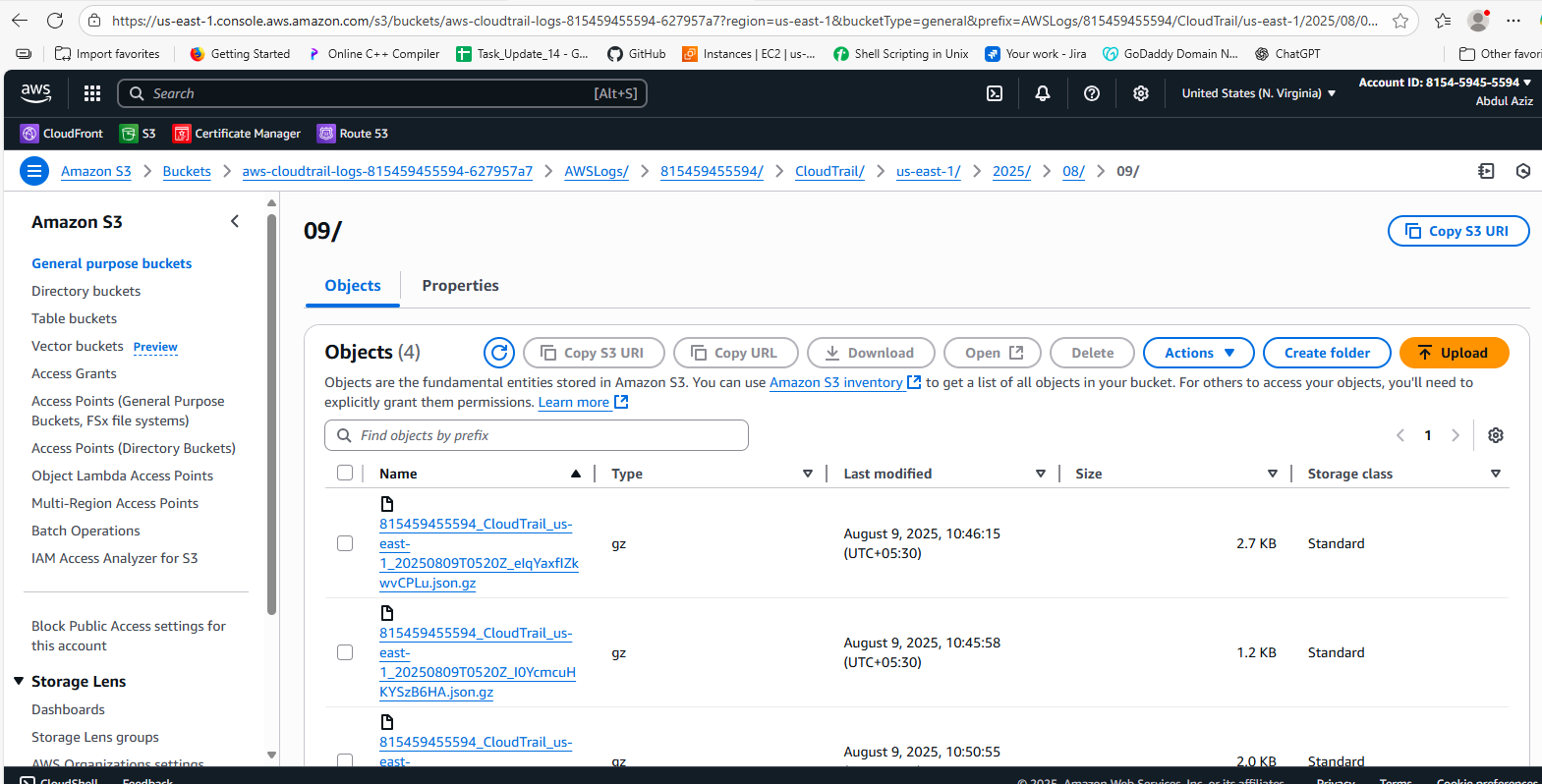
1. Review all your settings
2. Click **Create trail**

✅ Done! Now all CloudTrail events will be:

* Stored in the selected **S3 bucket**
* Streamed to **CloudWatch Logs** for monitoring & alerting







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

**🔹 Step 1: 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**

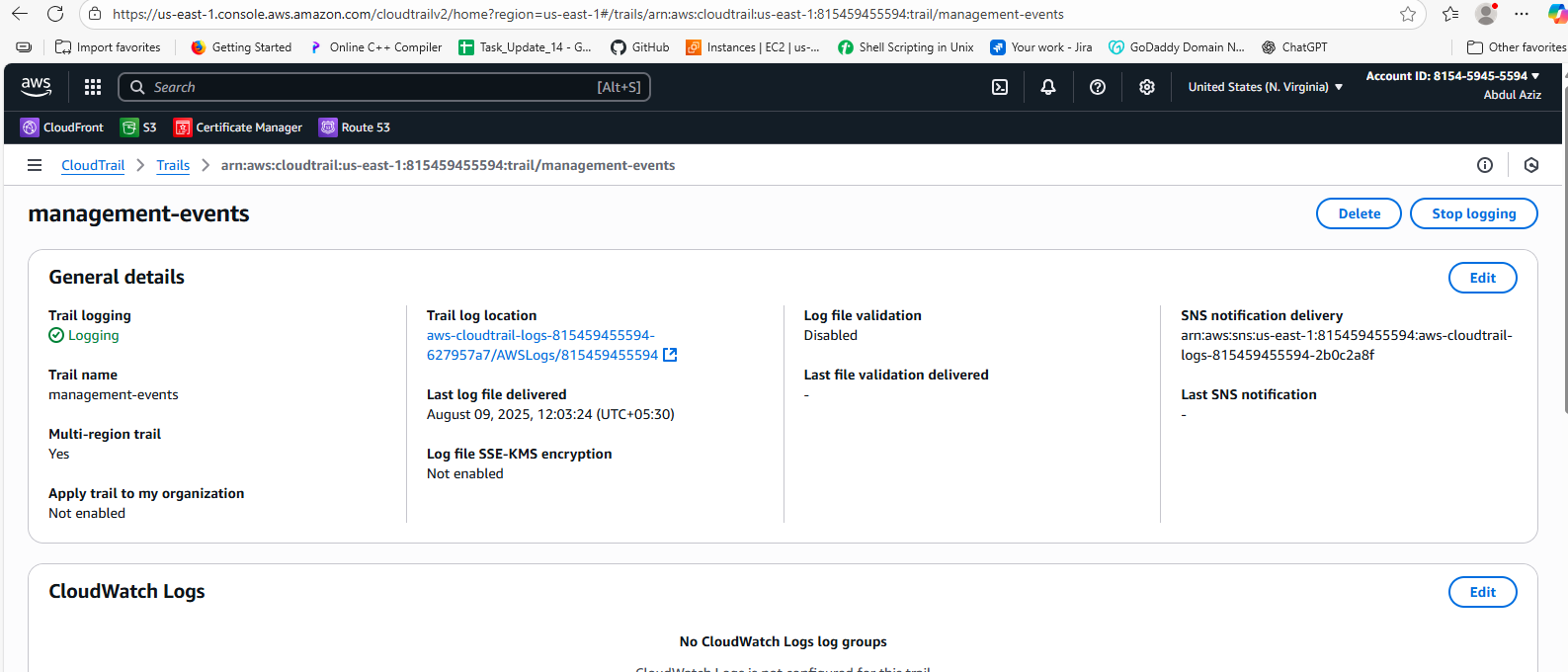
**🔹 Step 2: Subscribe to the Topic via Email**

1. Open the topic you just created
2. Click **Create subscription**
3. **Protocol**: Email
4. **Endpoint**: Enter your email address
5. Click **Create subscription**
6. Open your email and **confirm the subscription**

**🔹 Step 3: Create Event Rule for CloudTrail**

1. Go to **CloudWatch → Rules** (or EventBridge → Rules)
2. Click **Create rule**
3. Choose **Event source**:
   * **Service Name**: CloudTrail
   * **Event Type**: e.g., Management events or specific actions
4. Under **Target**, choose:
   * **SNS Topic**
   * Select the topic you created (cloudtrail-alerts)
5. Click **Next**, give the rule a name, and click **Create rule**

✅ Done! You’ll now receive email alerts when the selected CloudTrail events occur.



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

**🔹 Step 1: Open EC2 Console**

1. Go to **EC2 → Instances**
2. Select the instance you want to monitor

**🔹 Step 2: Enable Detailed Monitoring (Optional for 1-min data)**

1. In the **Details tab**, check **Monitoring**
2. Click **Actions → Monitor and troubleshoot → Manage detailed monitoring**
3. Enable it if needed (extra cost)

**🔹 Step 3: View CPU & Other Metrics**

1. Go to **CloudWatch → Metrics**
2. Click **Browse → EC2 → Per-Instance Metrics**
3. Select your **Instance ID**
4. View metrics like:
   * CPUUtilization
   * DiskReadBytes, DiskWriteBytes
   * NetworkIn, NetworkOut

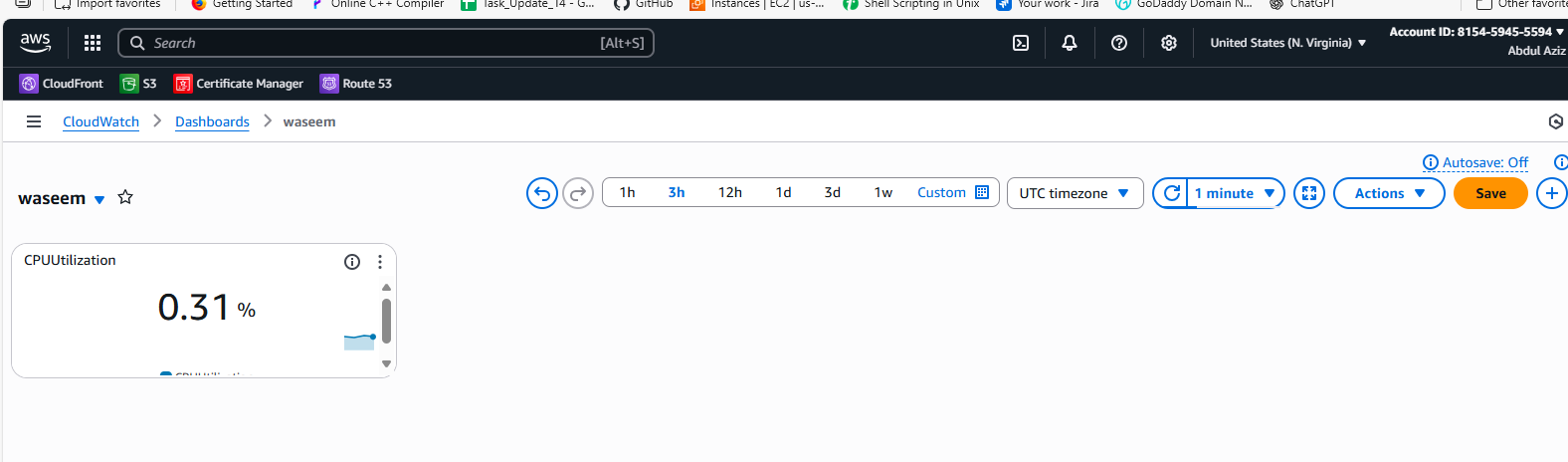
**🔹 Step 4: Create Dashboard (Optional)**

1. Go to **CloudWatch → Dashboards**
2. Click **Create dashboard**
3. Name it (e.g., ec2-monitoring)
4. Add widgets for **CPU**, **network**, etc.

**🔹 Step 5: Create Alarm (Optional)**

1. Go to **CloudWatch → Alarms → Create alarm**
2. Choose **EC2 → CPUUtilization**
3. Set threshold (e.g., > 70%)
4. Choose **SNS topic** (if alerts needed)
5. Create alarm

✅ Done! CloudWatch is now monitoring your EC2 instance's performance.



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

**🔹 Step 1: Go to CloudWatch Console**

1. Open **CloudWatch**
2. Click **Alarms → Create alarm**

**🔹 Step 2: Select Metric**

1. Click **Select metric**
2. Go to **EC2 → Per-Instance Metrics**
3. Select your **Instance ID**
4. Choose **CPUUtilization**
5. Click **Select metric**

**🔹 Step 3: Define Alarm Conditions**

1. **Threshold type**: Static
2. **Whenever CPUUtilization is...**: Greater than
3. **Threshold value**: 70
4. **Period**: 5 minutes (or as needed)

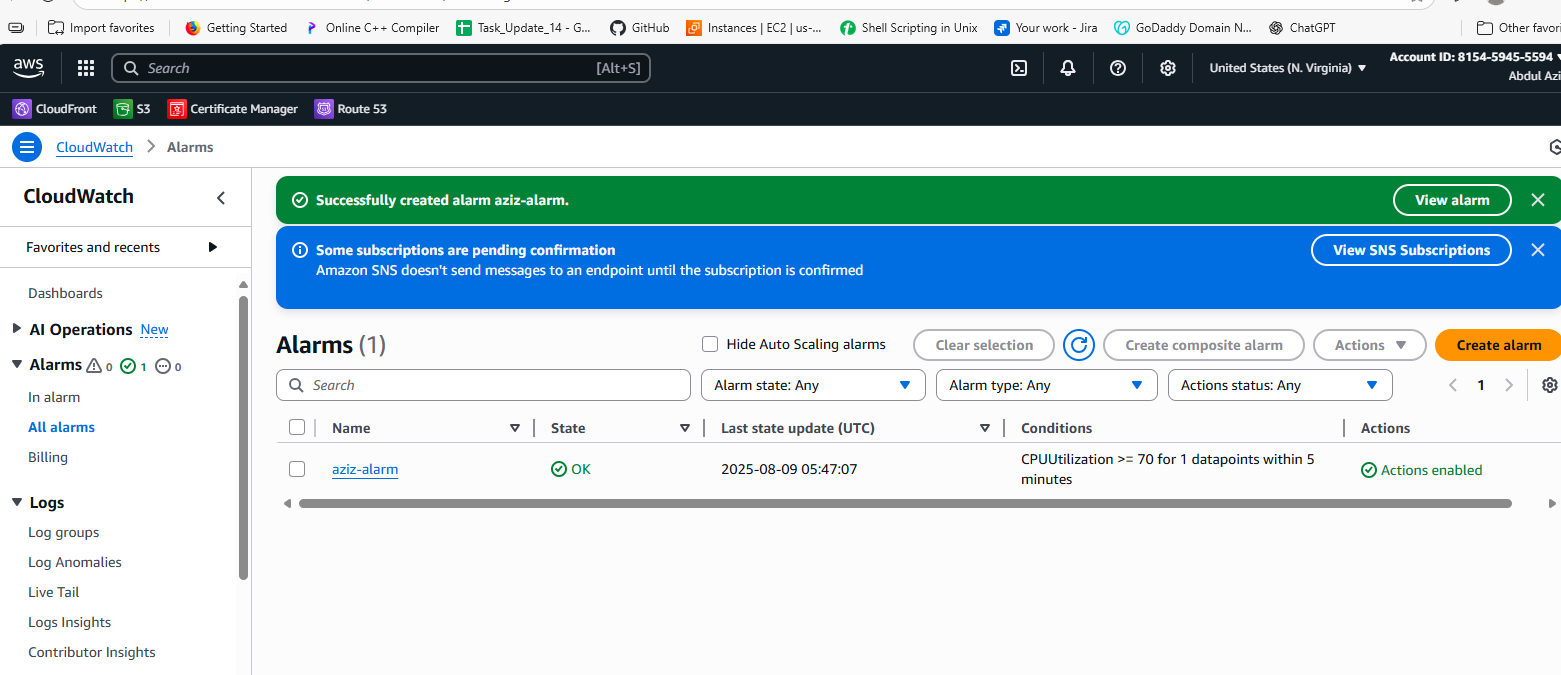
**🔹 Step 4: Configure Notifications**

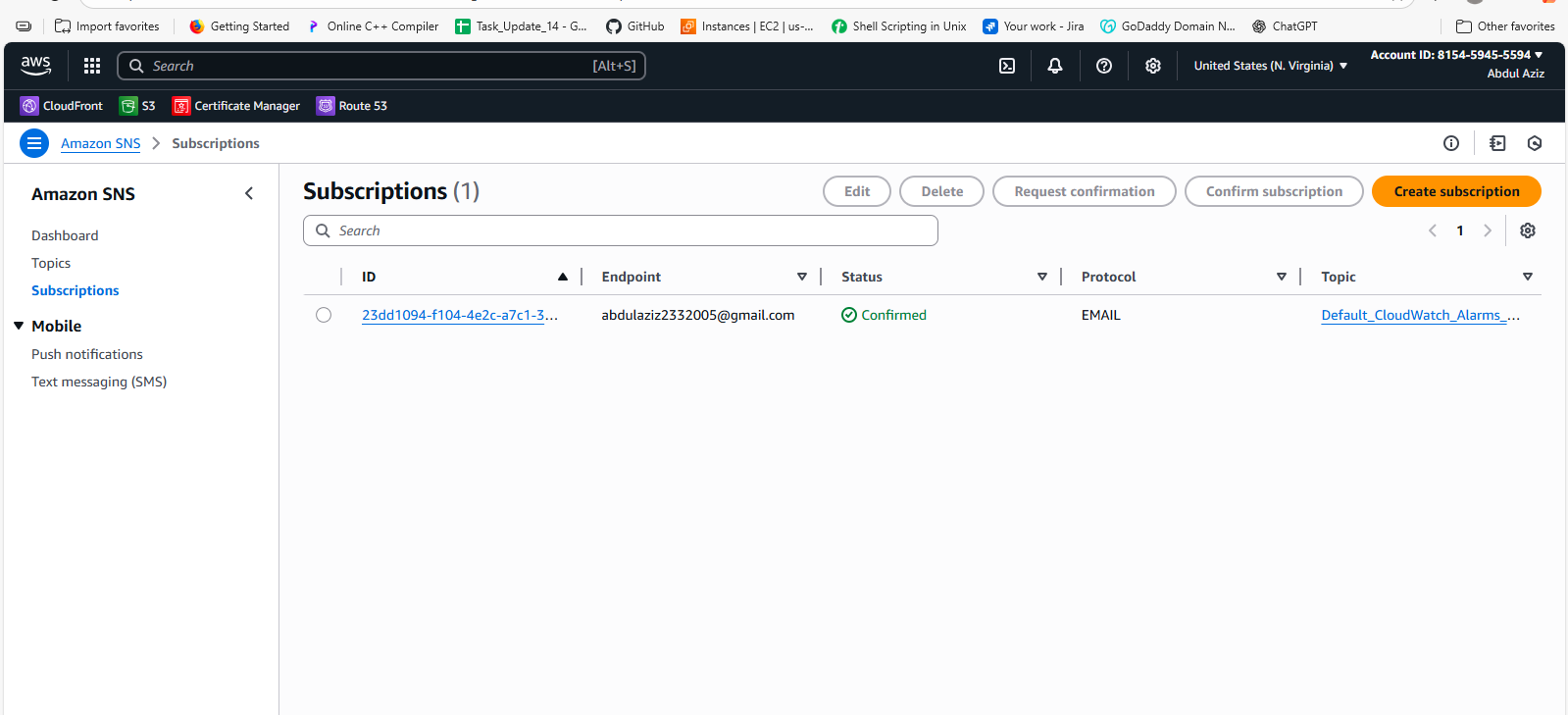
1. Under **Notification**, choose:
   * **In alarm** → Send notification to an **SNS topic**
2. If needed, click **Create SNS topic**:
   * Topic name: cpu-alert-topic
   * Email endpoint: your-email@example.com
3. Confirm the subscription via email

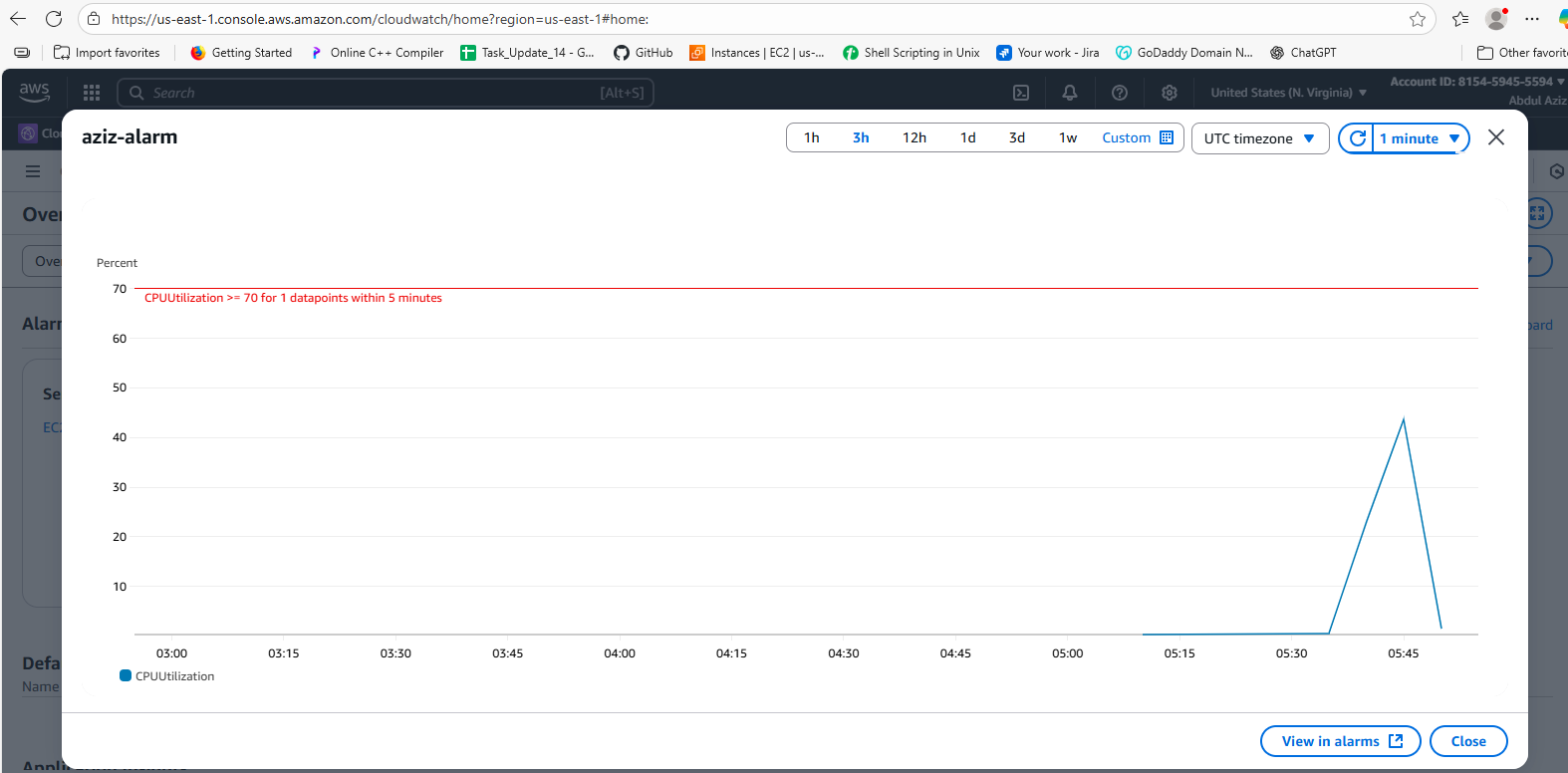
**🔹 Step 5: Name and Create Alarm**

1. Name your alarm (e.g., HighCPUAlarm)
2. Click **Create alarm**

✅ Done! You’ll now receive an **email alert** if CPU usage exceeds **70%**.







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

**🔹 Step 1: Install & Configure CloudWatch Agent**

1. Install the **CloudWatch Agent** on the EC2 instance
2. Create agent config file to monitor **Tomcat status**
   * Use a script to check tomcat service and push custom metric

**🔹 Step 2: 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

1. Schedule it via **cron** to run every 1 minute:

bash

CopyEdit

\* \* \* \* \* /path/to/tomcat\_status.sh

**🔹 Step 3: Create CloudWatch Alarm**

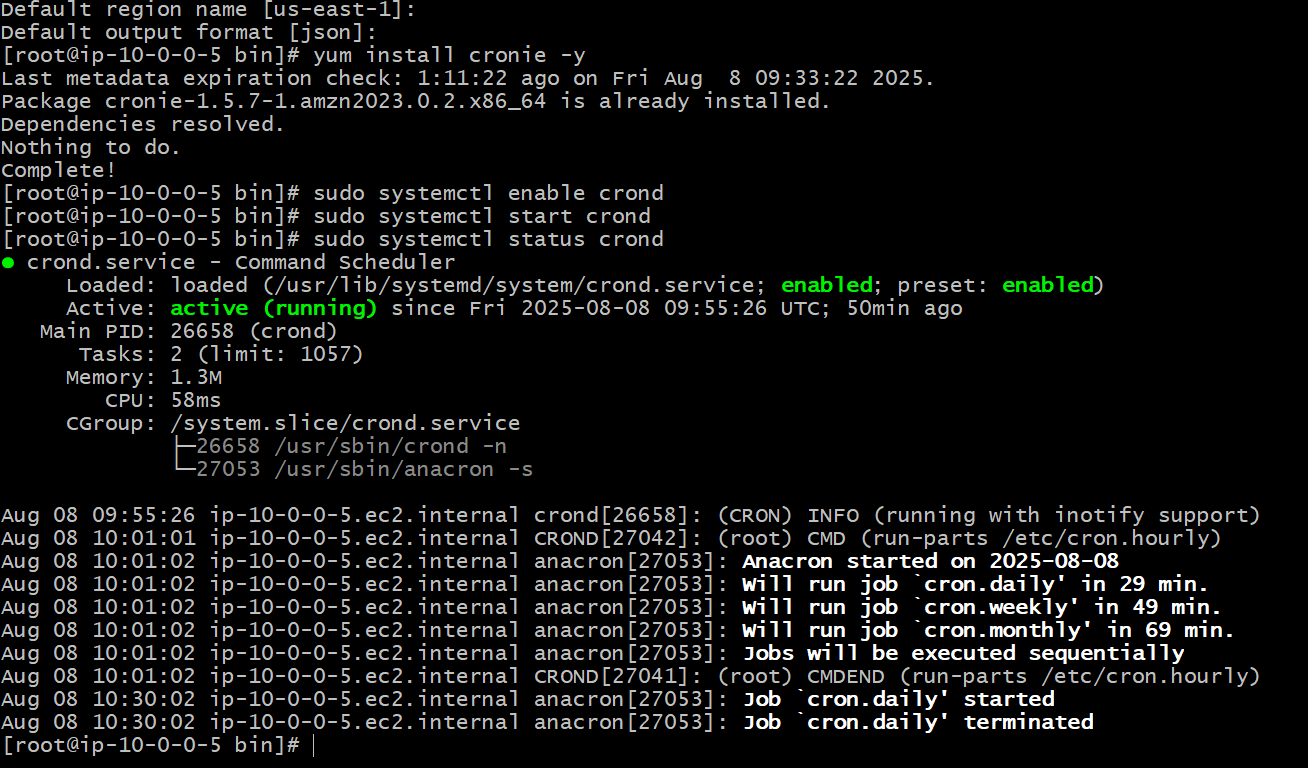
1. Go to **CloudWatch → Alarms → Create alarm**
2. Select custom namespace: TomcatStatus
3. Choose metric: TomcatRunning
4. Set threshold: **is equal to 0** (Tomcat not running)
5. Choose **SNS topic** for email alerts
6. Create alarm

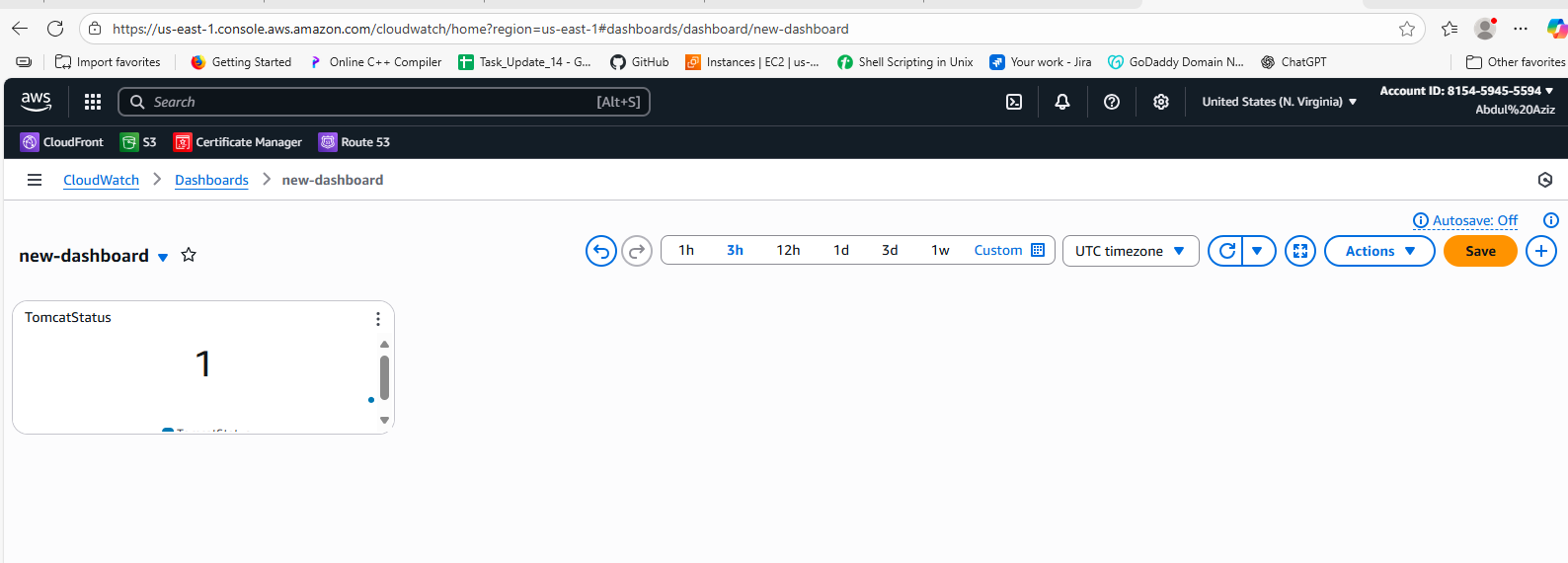
**🔹 Step 4: Create CloudWatch Dashboard**

1. 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

✅ Done! You now have:

* A **dashboard** showing Tomcat status





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

**🔹 Step 1: Install & Configure CloudWatch Agent**

1. Install the **CloudWatch Agent** on the EC2 instance
2. Ensure the agent is running and configured to accept custom metrics

**🔹 Step 2: 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

1. Save as: /opt/nginx\_status.sh
2. Give execute permission:

bash

CopyEdit

chmod +x /opt/nginx\_status.sh

**🔹 Step 3: Schedule the Script to Run Regularly**

1. Open crontab:

bash

CopyEdit

crontab -e

1. Add this line to run it every minute:

bash

CopyEdit

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

**🔹 Step 4: Create CloudWatch Alarm**

1. Go to **CloudWatch → Alarms → Create alarm**
2. Select namespace: NginxStatus
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**

**🔹 Step 5: 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

✅ Done! You now have:

* A working **Nginx monitor**

