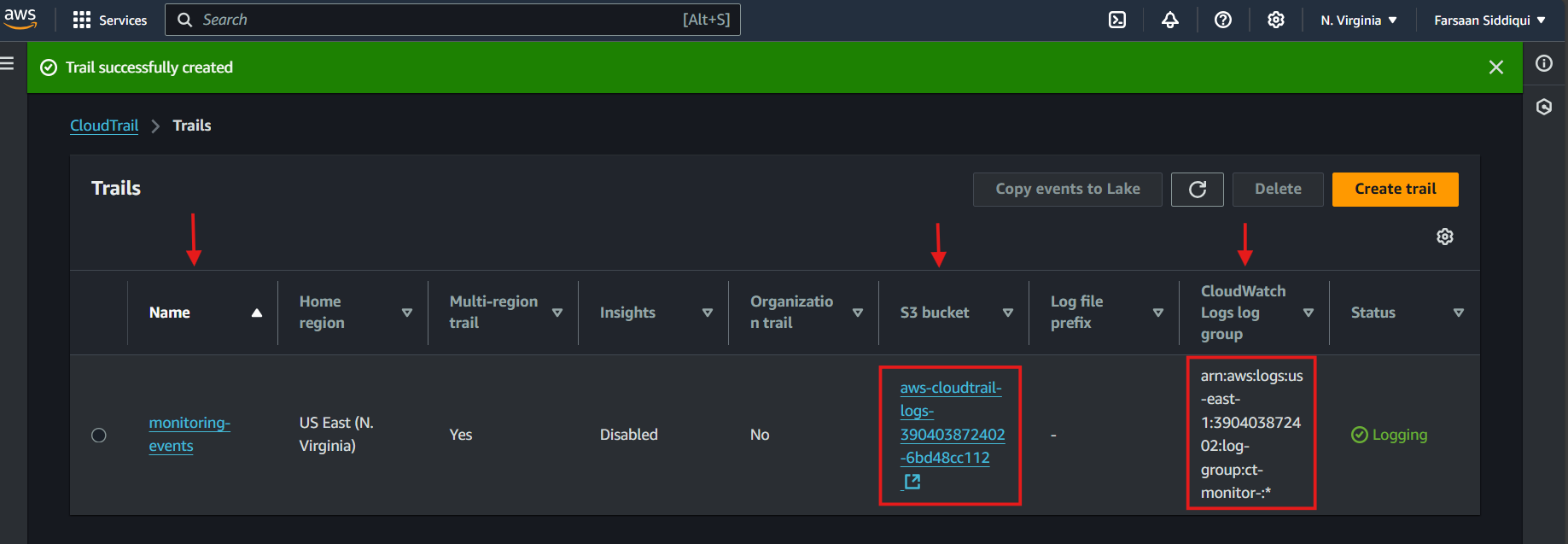
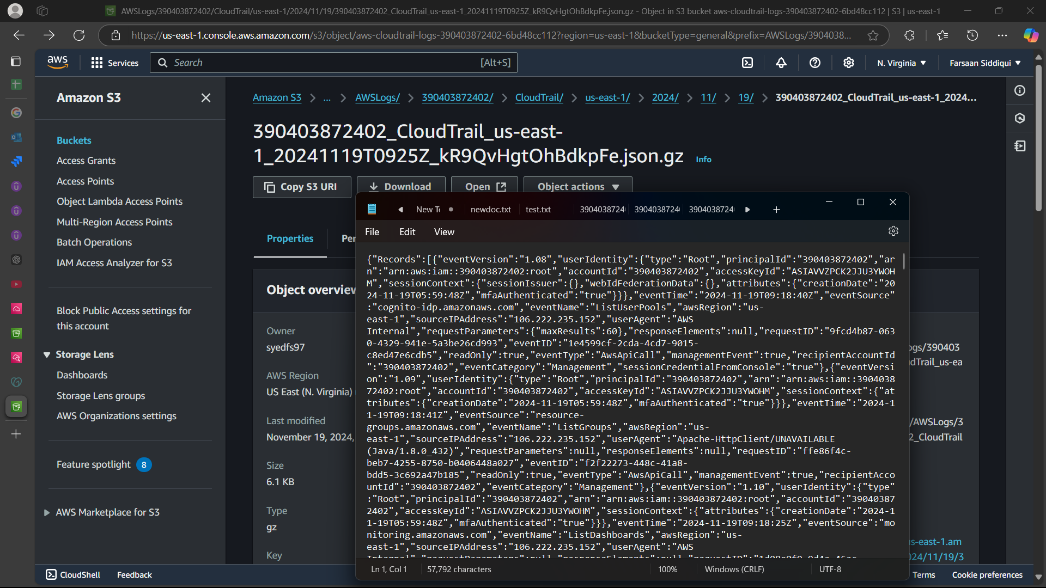
**Cloudwatch and Cloudtrial**

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

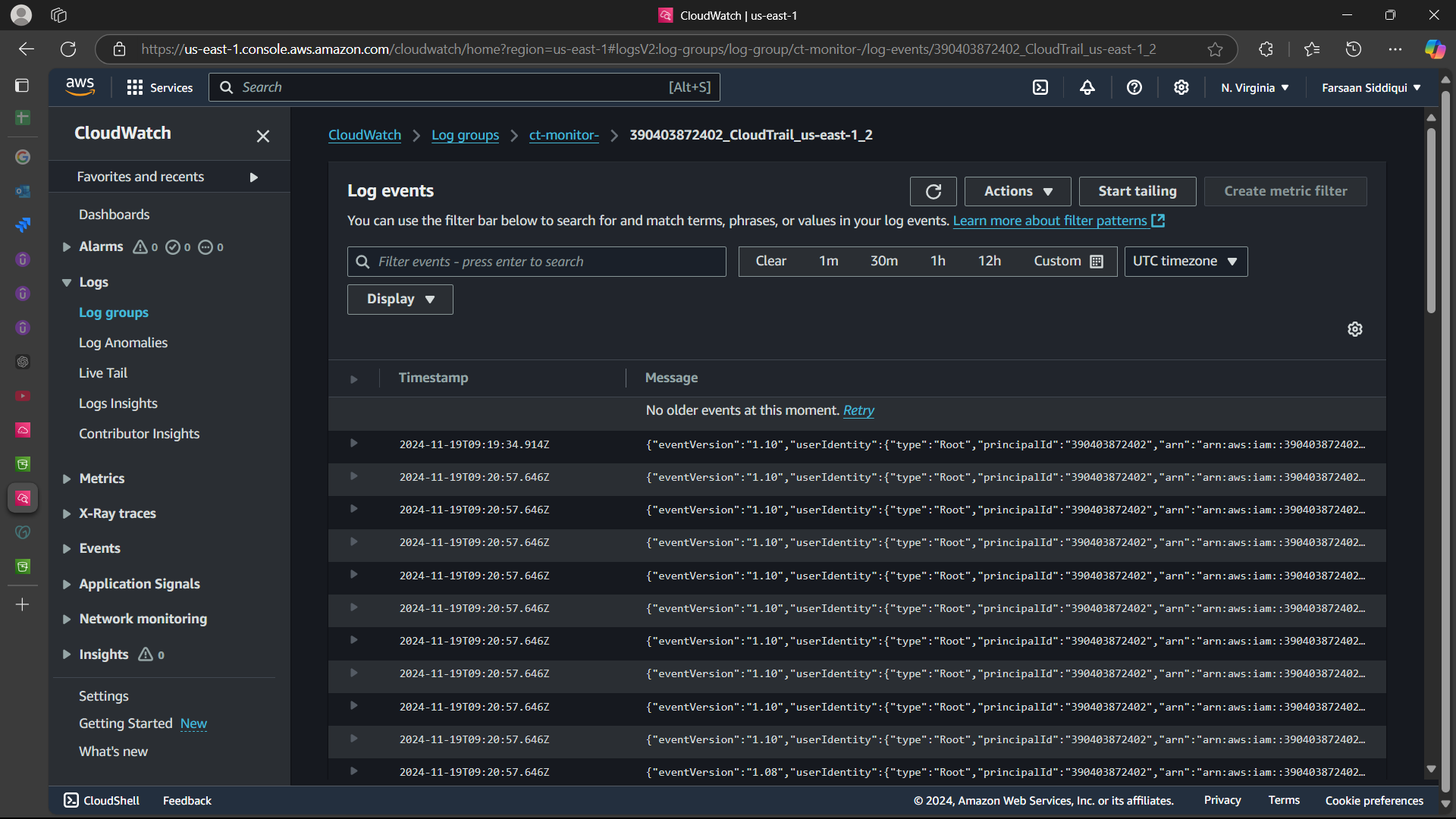
\*creating 1 trial in cloudtrial with management events for aws account and storing logs in s3 and cloudwatch

****

\*deleting old buckets and checking for logs from s3 bucket we created earlier for logs

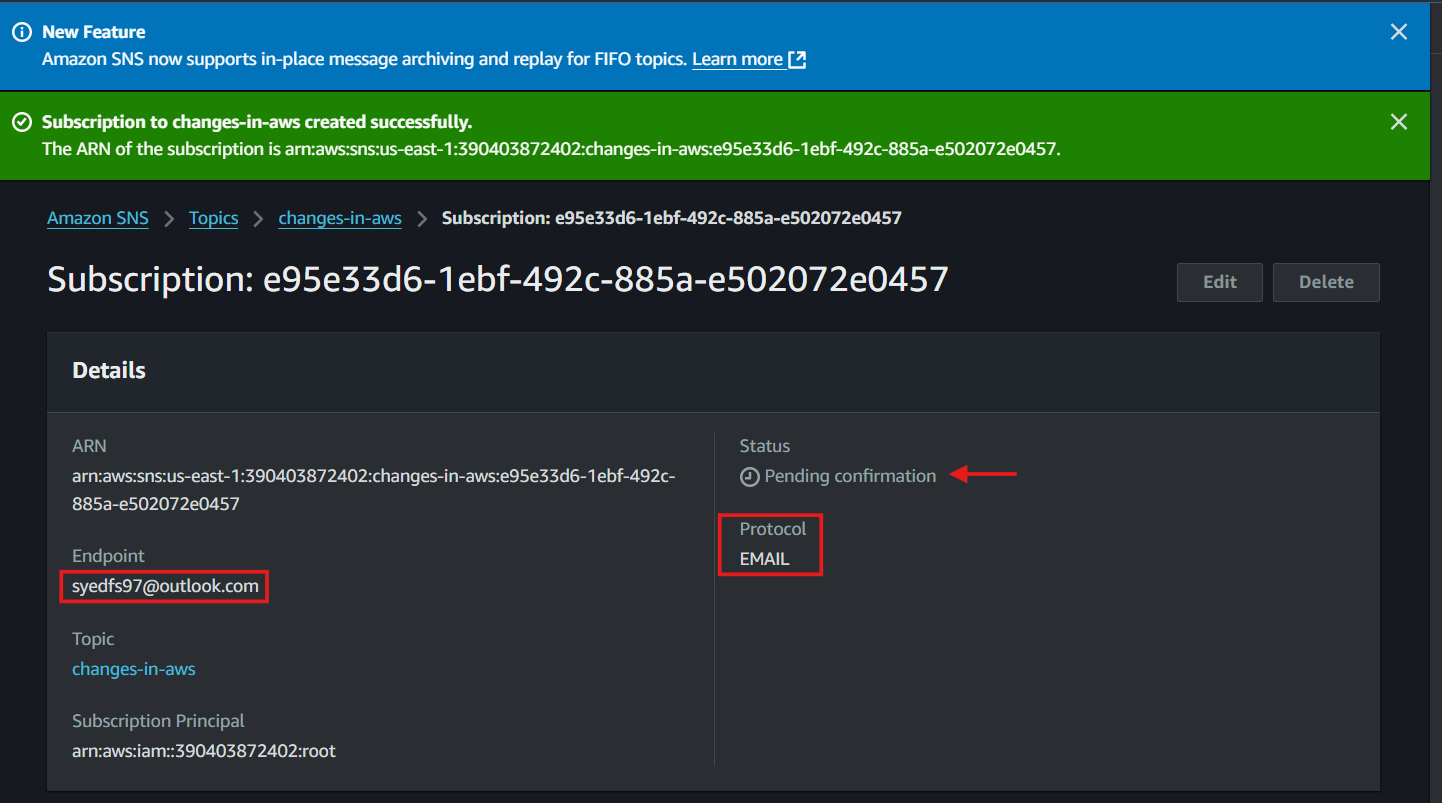


\*checking the cloudwatch log groups to make sure the logs are been captured

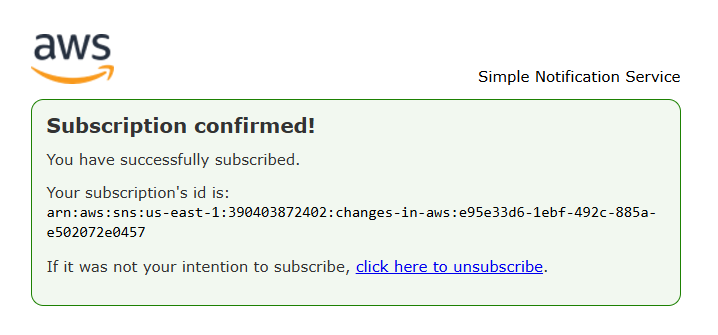


**2) Enable SNS for cloudtrial to send alert on email.**

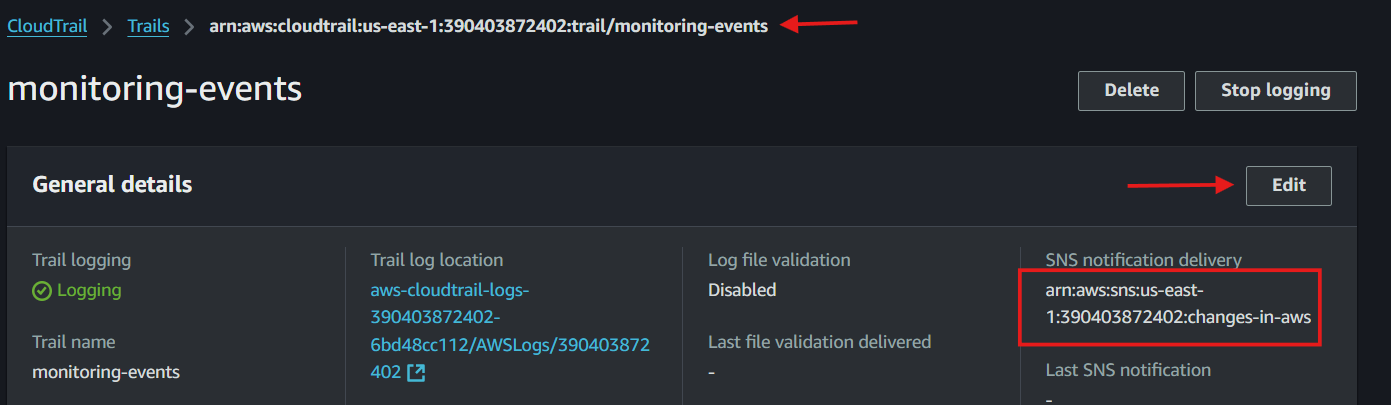
\*creating sns and adding subscription with email protocol

****

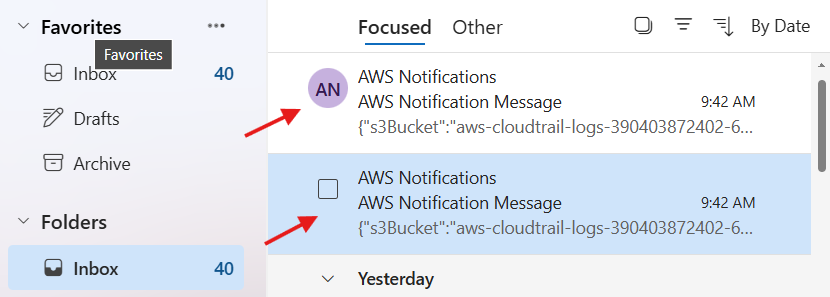
\*confirming the email for sns request



\*enabling the sns service by editing and selecting the one just created above



\*checking email if we are getting alerts as we have now enabled sns

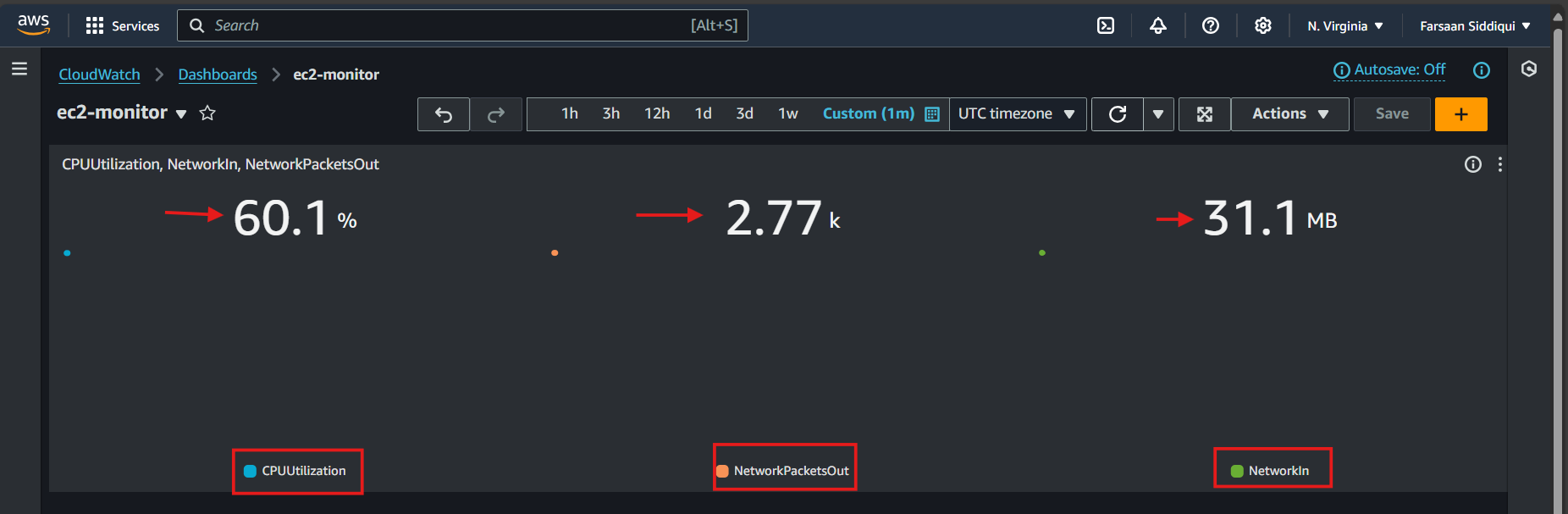
****

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

\*create one dashboard in cloudwatch for monitoring cpu utilization , networks packets sent and received.

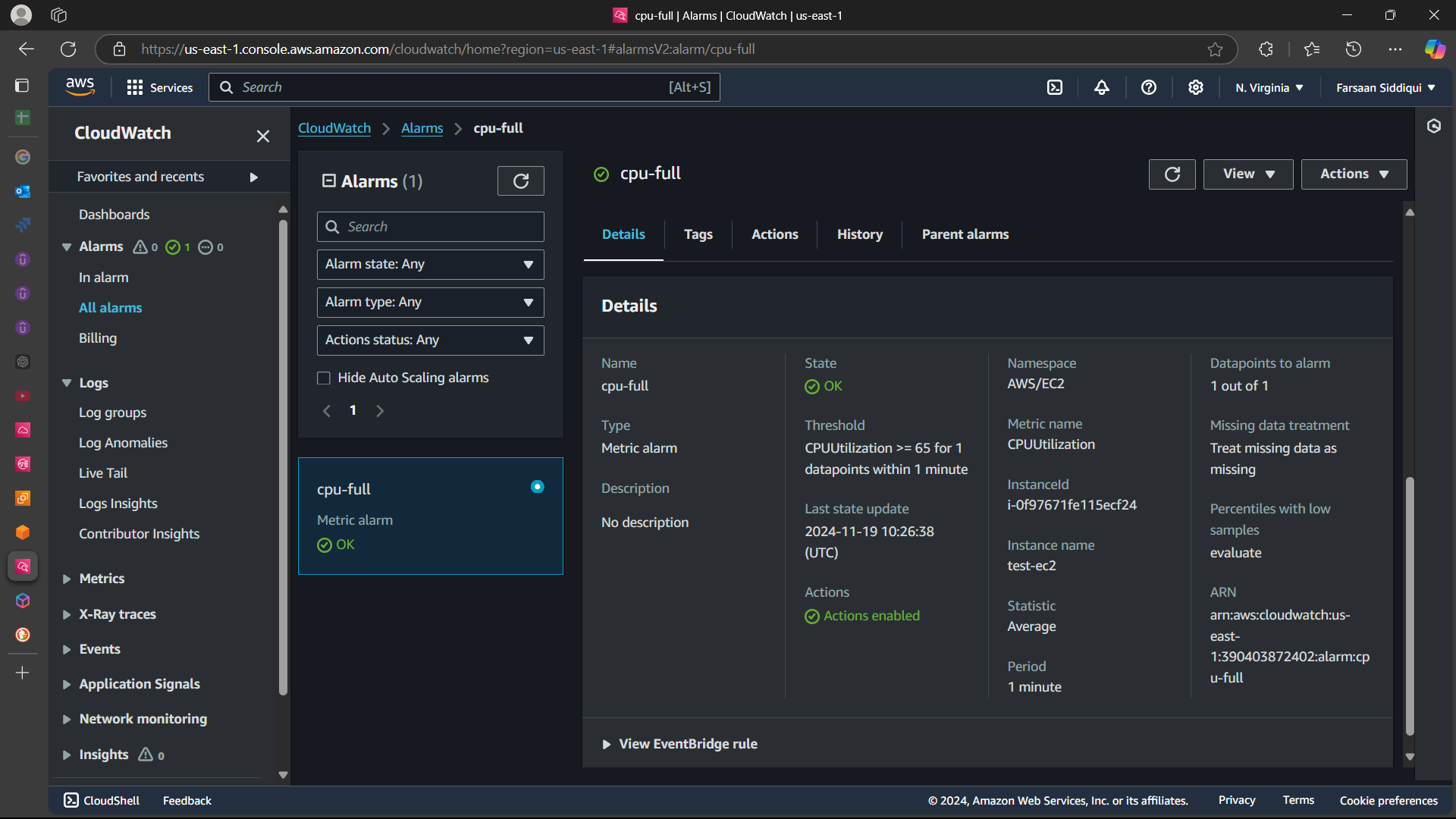
\*login in to ec2 and ping any website also dowload any image using wget0

\*install stress from amazon-linux-extra and use stress command to add load to cpu

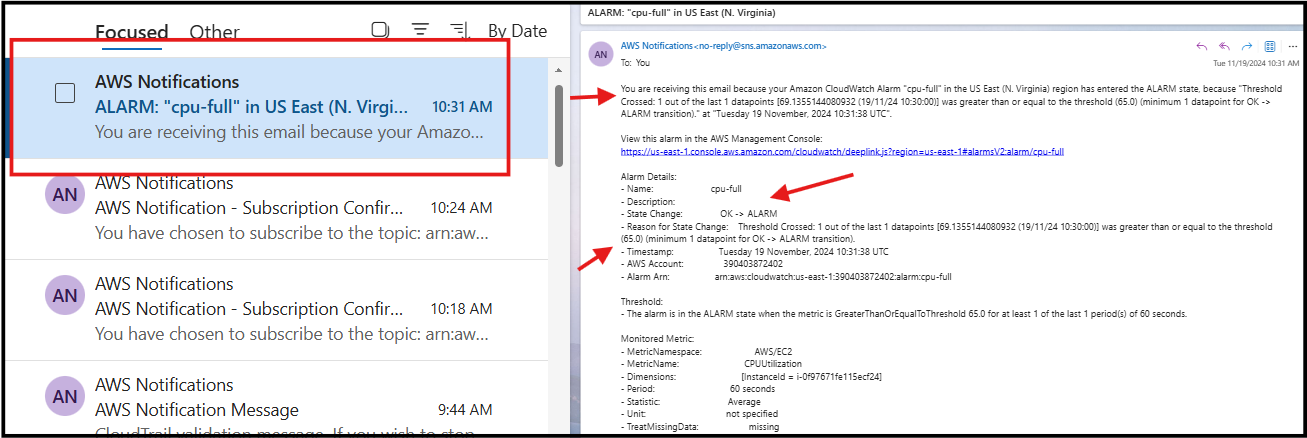
\*set the time frame to 1 minutes and check the dashboard 

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

\*create 1 alarm ,add sns alarm should send notification to email if it crosses 65% cpu usage

****

\*add stress to cpu and wait , will get the notification on email .

****

**What the Script Does**

1. **Creates a CloudWatch Dashboard**: The dashboard includes widgets for monitoring EC2 CPU utilization and status checks.
2. **Creates an SNS Topic**: The topic is used for sending alerts.
3. **Subscribes to the SNS Topic**: Your email address is subscribed to the topic to receive alerts.
4. **Creates a CloudWatch Alarm**: The alarm monitors the StatusCheckFailed metric and sends an alert if the status check fails.

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

\*login to instance and install java and tomcat

\*write a bash script which cretates cloudwatch dashboard , sns topic, subscription for sns topic and creates a cloudwatch alarm

**#!/bin/bash**

**# Variables**

**DASHBOARD\_NAME**=**"TomcatMonitoringDashboard"**

**ALARM\_NAME**=**"TomcatServiceAlarm"**

**TOPIC\_NAME**=**"TomcatServiceAlert"**

**TOPIC\_EMAIL**=**"syedfs97@outlook.com"**

**INSTANCE\_ID**=**"i-0f97671fe115ecf24"** **# Replace with your EC2 instance ID**

**# Create CloudWatch Dashboard**

aws cloudwatch put-dashboard **--dashboard-name** **$DASHBOARD\_NAME** **--dashboard-body** **'{**

**"widgets": [**

**{**

**"type": "metric",**

**"x": 0,**

**"y": 0,**

**"width": 24,**

**"height": 6,**

**"properties": {**

**"metrics": [**

**[ "AWS/EC2", "CPUUtilization", "InstanceId", "'$INSTANCE\_ID'" ]**

**],**

**"period": 300,**

**"stat": "Average",**

**"region": "us-east-1",**

**"title": "EC2 CPU Utilization"**

**}**

**},**

**{**

**"type": "metric",**

**"x": 0,**

**"y": 6,**

**"width": 24,**

**"height": 6,**

**"properties": {**

**"metrics": [**

**[ "AWS/EC2", "StatusCheckFailed", "InstanceId", "'$INSTANCE\_ID'" ]**

**],**

**"period": 300,**

**"stat": "Average",**

**"region": "us-east-1",**

**"title": "EC2 Status Check Failed"**

**}**

**}**

**]**

**}'**

**# Create SNS Topic**

**TOPIC\_ARN**=**$(aws sns create-topic --name $TOPIC\_NAME --query 'TopicArn' --output text)**

**# Subscribe to SNS Topic**

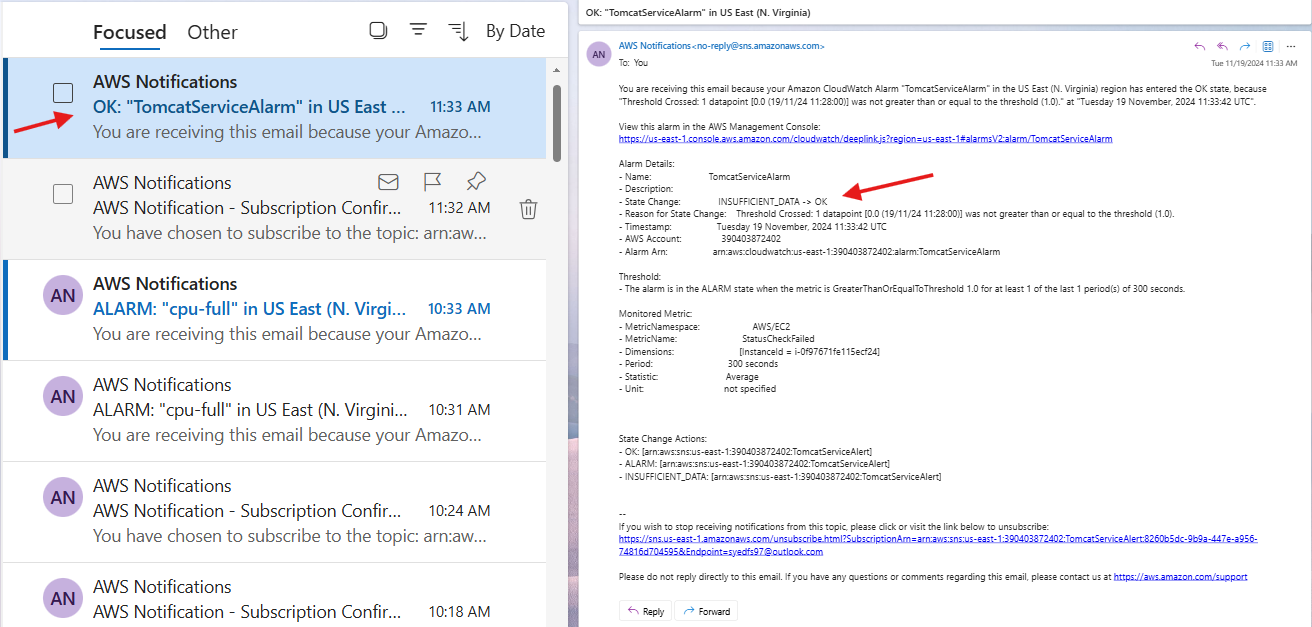
aws sns subscribe **--topic-arn** **$TOPIC\_ARN** **--protocol** email **--notification-endpoint** **$TOPIC\_EMAIL**

**# Create CloudWatch Alarm**

aws cloudwatch put-metric-alarm **--alarm-name** **$ALARM\_NAME** **--metric-name** StatusCheckFailed **--namespace** AWS/EC2 **--statistic** Average **--period** **300** **--threshold** **1** **--comparison-operator** GreaterThanOrEqualToThreshold **--dimensions** **Name**=InstanceId,**Value**=**$INSTANCE\_ID** **--evaluation-periods** **1** **--alarm-actions** **$TOPIC\_ARN** **--insufficient-data-actions** **$TOPIC\_ARN** **--ok-actions** **$TOPIC\_ARN**

**echo "CloudWatch Dashboard and Alarm setup complete. Check your email for subscription confirmation."**

**>>>**check email if we got alert about the service

****

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

\*login to instance and install nginx

\*write a bash script which cretates cloudwatch dashboard , sns topic, subscription for sns topic and creates a cloudwatch alarm

**#!/bin/bash**

**# Variables**

**DASHBOARD\_NAME**=**"NginxMonitoringDashboard"**

**ALARM\_NAME**=**"NginxServiceAlarm"**

**TOPIC\_NAME**=**"NginxServiceAlert"**

**TOPIC\_EMAIL**=**"syedfs97@outlook.com"**

**INSTANCE\_ID**=**"i-0f97671fe115ecf24"** **# Replace with your EC2 instance ID**

**# Create CloudWatch Dashboard**

aws cloudwatch put-dashboard **--dashboard-name** **$DASHBOARD\_NAME** **--dashboard-body** **'{**

**"widgets": [**

**{**

**"type": "metric",**

**"x": 0,**

**"y": 0,**

**"width": 24,**

**"height": 6,**

**"properties": {**

**"metrics": [**

**[ "AWS/EC2", "CPUUtilization", "InstanceId", "'$INSTANCE\_ID'" ]**

**],**

**"period": 300,**

**"stat": "Average",**

**"region": "us-east-1",**

**"title": "EC2 CPU Utilization"**

**}**

**},**

**{**

**"type": "metric",**

**"x": 0,**

**"y": 6,**

**"width": 24,**

**"height": 6,**

**"properties": {**

**"metrics": [**

**[ "AWS/EC2", "StatusCheckFailed", "InstanceId", "'$INSTANCE\_ID'" ]**

**],**

**"period": 300,**

**"stat": "Average",**

**"region": "us-east-1",**

**"title": "EC2 Status Check Failed"**

**}**

**}**

**]**

**}'**

**# Create SNS Topic**

**TOPIC\_ARN**=**$(aws sns create-topic --name $TOPIC\_NAME --query 'TopicArn' --output text)**

**# Subscribe to SNS Topic**

aws sns subscribe **--topic-arn** **$TOPIC\_ARN** **--protocol** email **--notification-endpoint** **$TOPIC\_EMAIL**

**# Create CloudWatch Alarm for Nginx**

aws cloudwatch put-metric-alarm **--alarm-name** **$ALARM\_NAME** **--metric-name** StatusCheckFailed **--namespace** AWS/EC2 **--statistic** Average **--period** **300** **--threshold** **1** **--comparison-operator** GreaterThanOrEqualToThreshold **--dimensions** **Name**=InstanceId,**Value**=**$INSTANCE\_ID** **--evaluation-periods** **1** **--alarm-actions** **$TOPIC\_ARN** **--insufficient-data-actions** **$TOPIC\_ARN** **--ok-actions** **$TOPIC\_ARN**

**# Create a custom metric for Nginx status**

cat **<<EOL > nginx\_status.sh**

**#!/bin/bash**

**if systemctl is-active --quiet nginx; then**

**aws cloudwatch put-metric-data --metric-name NginxStatus --namespace CustomMetrics --value 1 --dimensions InstanceId=$INSTANCE\_ID**

**else**

**aws cloudwatch put-metric-data --metric-name NginxStatus --namespace CustomMetrics --value 0 --dimensions InstanceId=$INSTANCE\_ID**

**fi**

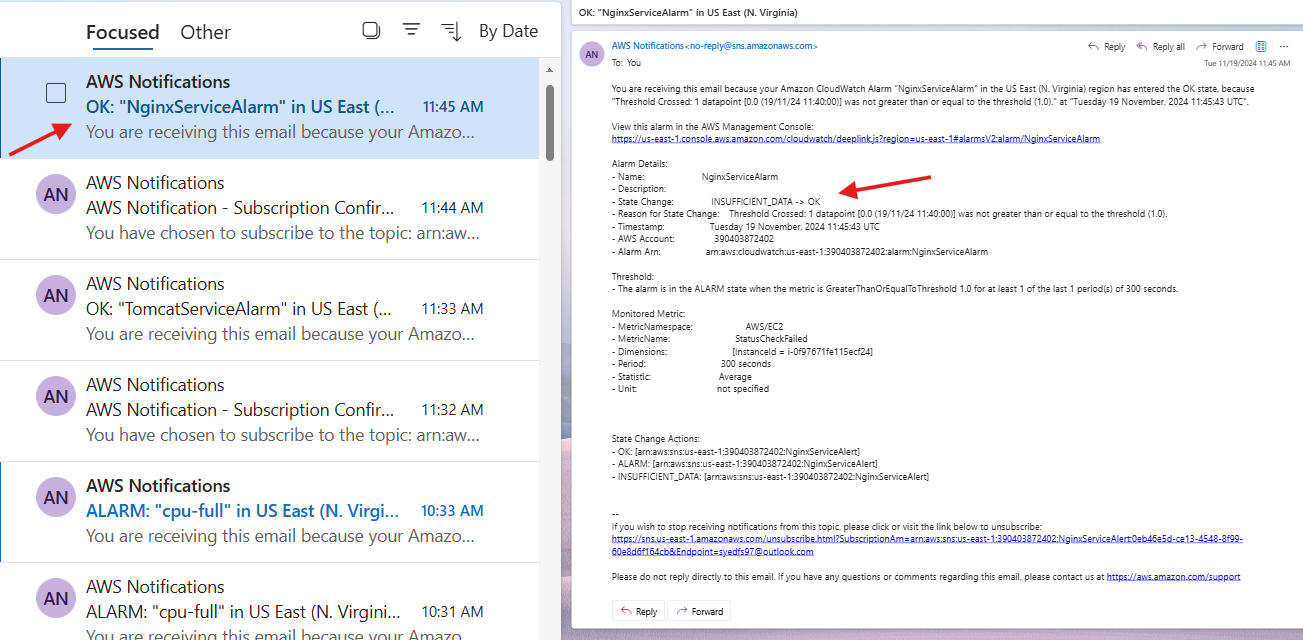
**EOL**

**chmod** **+x** nginx\_status.sh

**# Schedule the script to run every minute using cron**

**(crontab -l 2>/dev/null; echo "\* \* \* \* \* /path/to/nginx\_status.sh")** | crontab –

**>>>**check email if we got alert about the service

****

**What the Script Does**

1. **Creates a CloudWatch Dashboard**: The dashboard includes widgets for monitoring EC2 CPU utilization and status checks.
2. **Creates an SNS Topic**: The topic is used for sending alerts.
3. **Subscribes to the SNS Topic**: Your email address is subscribed to the topic to receive alerts.
4. **Creates a CloudWatch Alarm**: The alarm monitors the StatusCheckFailed metric and sends an alert if the status check fails.
5. **Creates a Custom Metric for Nginx Status**: The script nginx\_status.sh checks the status of the Nginx service and sends the metric to CloudWatch.
6. **Schedules the Custom Metric Script**: The script is scheduled to run every minute using cron.
7. **Creates a CloudWatch Alarm for Nginx Custom Metric**: The alarm monitors the custom metric and sends an alert if the Nginx service is not running.