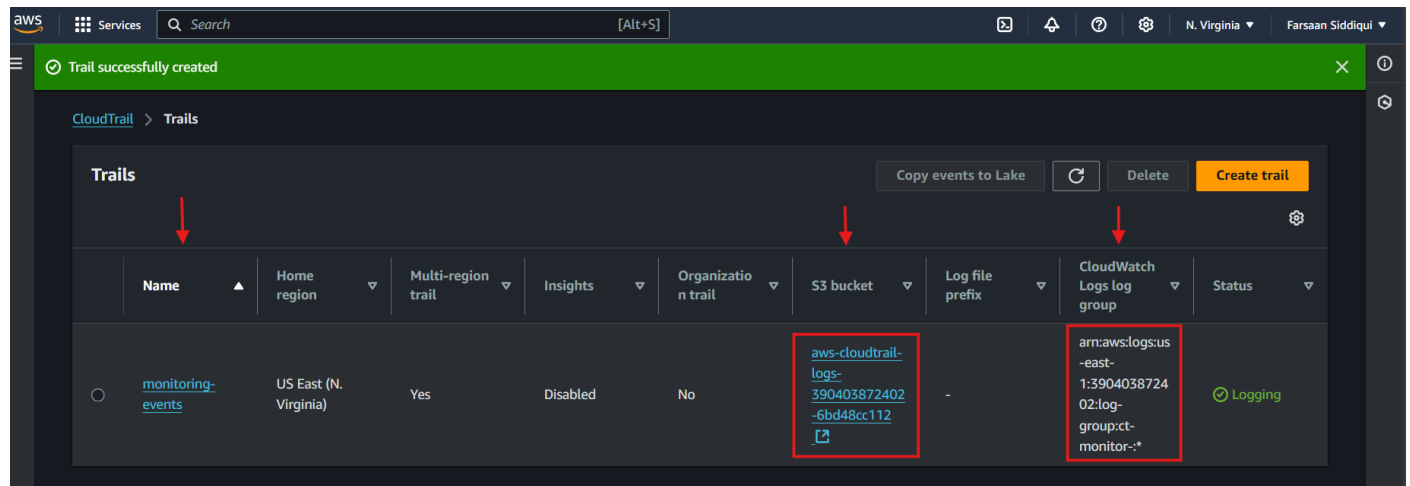
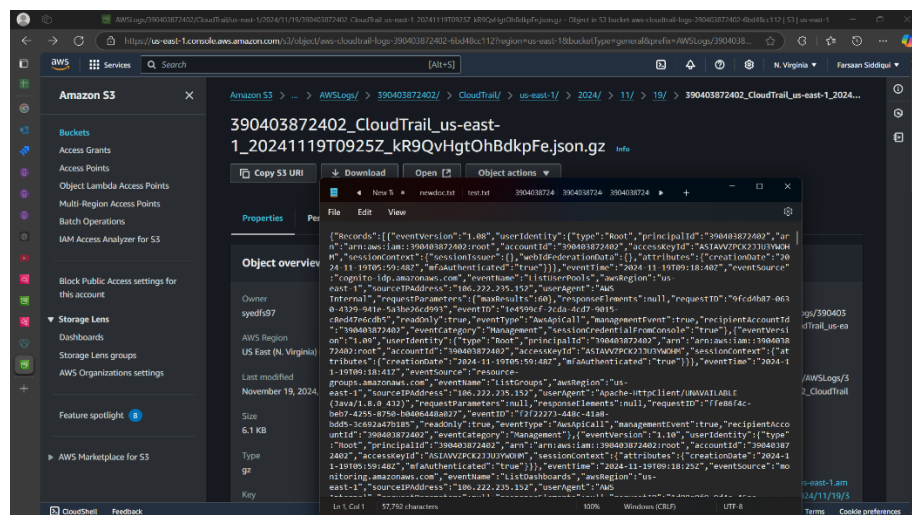


1) Enable cloudtrail monitoring and store the events in s3 and cloudwatch log events.

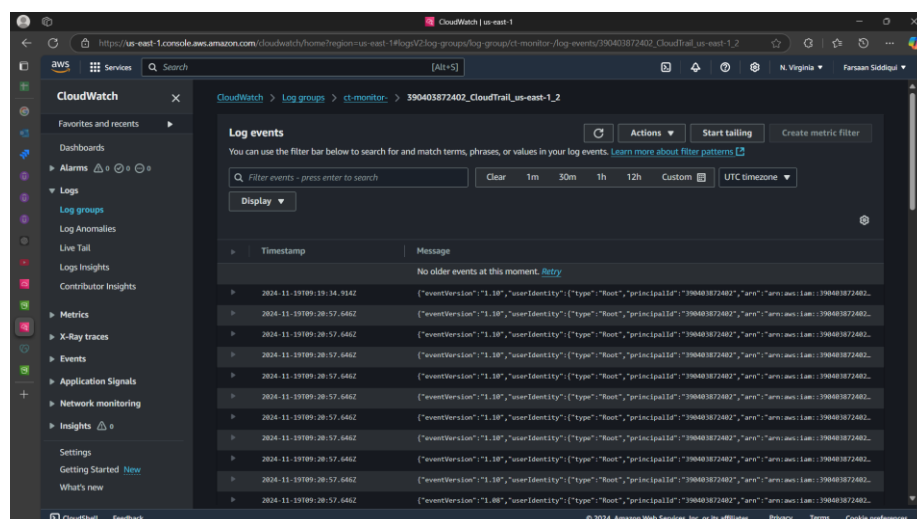
*CREATING 1 TRIAL IN CLOUDTRIAL WITH MANAGEMENT EVENTS FOR AWS ACCOUNT AND STORING LOGS IN S3 AND CLOUDWATCH



*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 cloudtrail to send alert on email.

*CREATING SNS AND ADDING SUBSCRIPTION WITH EMAIL PROTOCOL

The screenshot shows the Amazon SNS console interface. At the top, there are two notification banners: a blue one for a new feature and a green one stating 'Subscription to changes-in-aws created successfully.' Below these, the breadcrumb trail is 'Amazon SNS > Topics > changes-in-aws > Subscription: e95e33d6-1ebf-492c-885a-e502072e0457'. The main heading is 'Subscription: e95e33d6-1ebf-492c-885a-e502072e0457' with 'Edit' and 'Delete' buttons. The 'Details' section shows the ARN, Endpoint (syeds97@outlook.com), Topic (changes-in-aws), and Subscription Principal. The 'Status' is 'Pending confirmation' with a red arrow pointing to it. The 'Protocol' is 'EMAIL', also highlighted with a red box.

*CONFIRMING THE EMAIL FOR SNS REQUEST



Simple Notification Service

Subscription confirmed!

You have successfully subscribed.

Your subscription's id is:

arn:aws:sns:us-east-1:390403872402:changes-in-aws:e95e33d6-1ebf-492c-885a-e502072e0457

If it was not your intention to subscribe, [click here to unsubscribe](#).

*ENABLING THE SNS SERVICE BY EDITING AND SELECTING THE ONE JUST CREATED ABOVE

The screenshot shows the AWS CloudTrail console. The breadcrumb trail is 'CloudTrail > Trails > arn:aws:cloudtrail:us-east-1:390403872402:trail/monitoring-events', with a red arrow pointing to the last part. The main heading is 'monitoring-events' with 'Delete' and 'Stop logging' buttons. The 'General details' section has an 'Edit' button with a red arrow pointing to it. The 'SNS notification delivery' field is highlighted with a red box and contains the ARN 'arn:aws:sns:us-east-1:390403872402:changes-in-aws'.

*CHECKING EMAIL IF WE ARE GETTING ALERTS AS WE HAVE NOW ENABLED SNS

The screenshot shows an email inbox interface. On the left, there's a sidebar with 'Favorites' (Inbox, Drafts, Archive) and 'Folders' (Inbox). The main area shows a list of emails. Two emails from 'AWS Notifications' are highlighted with red arrows. The first email has a purple 'AN' icon and the subject 'AWS Notification Message'. The second email has a blue square icon and the same subject. Both emails have a timestamp of '9:42 AM' and a truncated body text starting with '{\"s3Bucket\": \"aws-cloudtrail-logs-390403872402-6...\"}'.

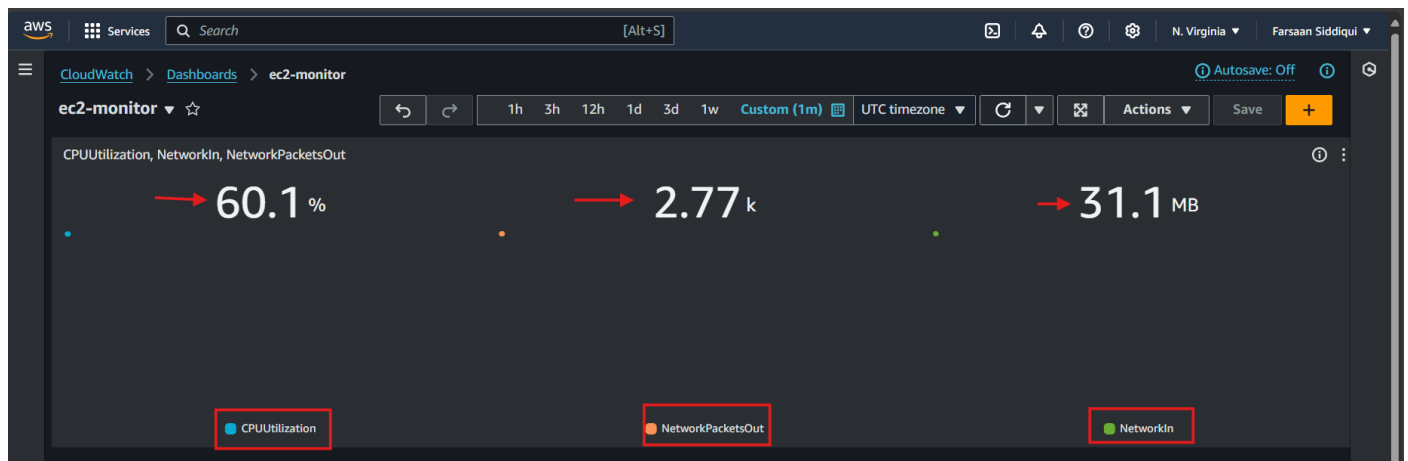
3) Configure cloud watch monitoring and record the cpu utilization and other metrics of ec2.

*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 WGET

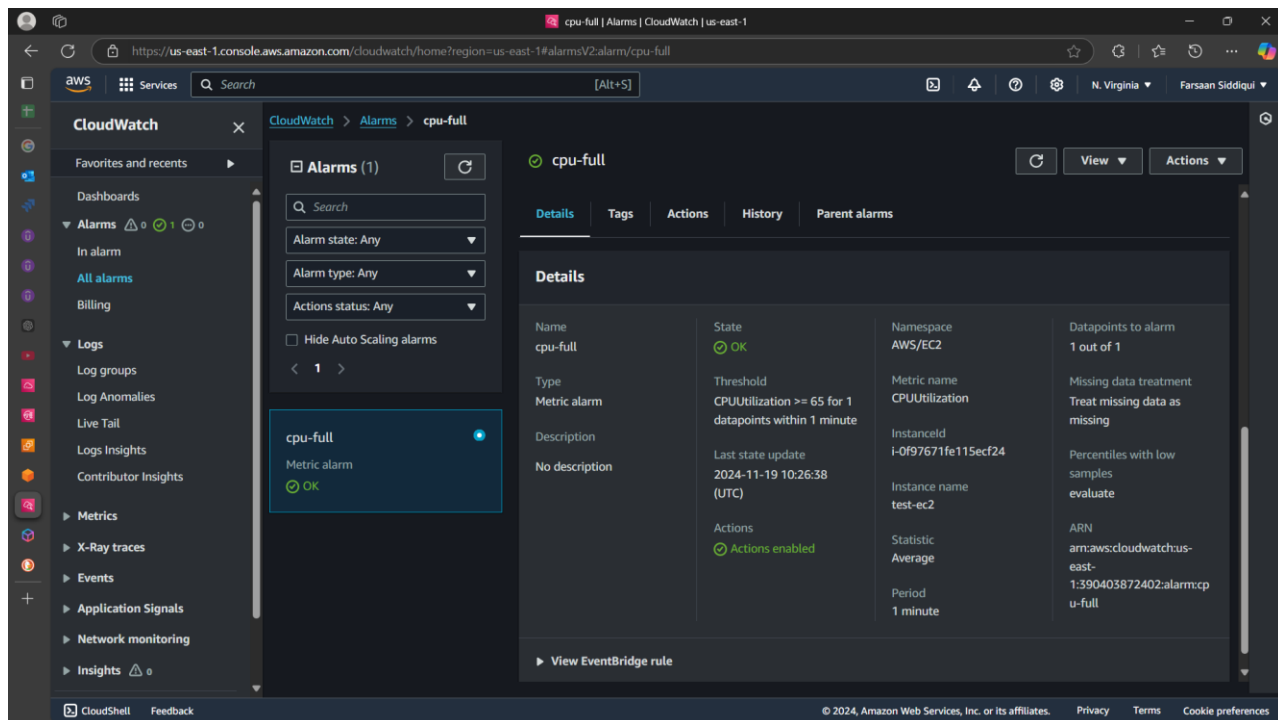
*INSTALL STRESS FROM AMAZON-LINUX-EXTRA AND USE STRESS COMMAND TO ADD LOAD TO CPU

*SET THE TIME FRAME TO 1 MINUTES AND CHECK THE DASHBOARD

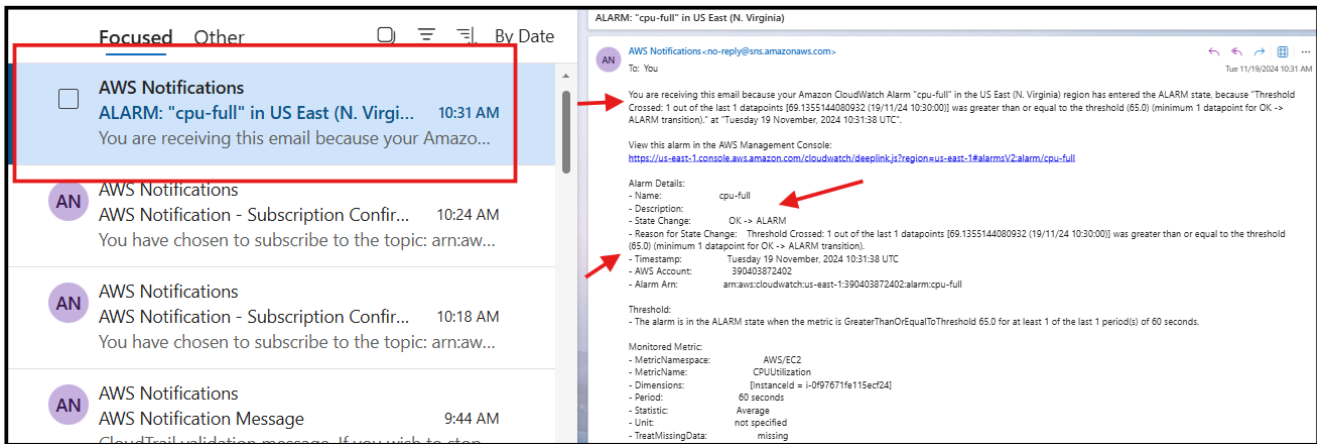


4) Create one alarm to send alert to email if the cpu utilization is more than 70 percent.

*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 CREATES 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

The screenshot shows an email inbox on the left and the details of an AWS Notification email on the right. The email is titled "OK: TomcatServiceAlarm" in US East (N. Virginia) and is dated 11:33 AM. The email body contains details about the alarm state change from "INSUFFICIENT_DATA" to "OK". A red arrow points to the "OK" state change in the email body. Another red arrow points to the "Name: TomcatServiceAlarm" field in the "Alarm Details" section.

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 CREATES 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

FocusedOther

By Date

AWS Notifications

OK: "NginxServiceAlarm" in US East (... 11:45 AM

You are receiving this email because your Amazo...

AN

AWS Notifications

AWS Notification - Subscription Confir... 11:44 AM

You have chosen to subscribe to the topic: arn:aw...

AN

AWS Notifications

OK: "TomcatServiceAlarm" in US East (... 11:33 AM

You are receiving this email because your Amazo...

AN

AWS Notifications

AWS Notification - Subscription Confir... 11:32 AM

You have chosen to subscribe to the topic: arn:aw...

AN

AWS Notifications

ALARM: "cpu-full" in US East (N. Virgi... 10:33 AM

You are receiving this email because your Amazo...

AN

AWS Notifications

ALARM: "cpu-full" in US East (N. Virgini... 10:31 AM

You are receiving this email because your Amazo...

OK: "NginxServiceAlarm" in US East (N. Virginia)

AN

AWS Notifications <no-reply@sns.amazonaws.com>

To: You

Tue 11/19/2024 11:45 AM

Reply

Reply all

Forward

You are receiving this email because your Amazon CloudWatch Alarm "NginxServiceAlarm" in the US East (N. Virginia) region has entered the OK state, because "Threshold Crossed: 1 datapoint [0.0 (19/11/24 11:40:00)] was not greater than or equal to the threshold (1.0)." at "Tuesday 19 November, 2024 11:45:43 UTC".

View this alarm in the AWS Management Console:
<https://us-east-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=us-east-1#alarmsV2:alarm/NginxServiceAlarm>

Alarm Details:

- Name:	NginxServiceAlarm
- Description:	
- State Change:	INSUFFICIENT_DATA -> OK
- Reason for State Change:	Threshold Crossed: 1 datapoint [0.0 (19/11/24 11:40:00)] was not greater than or equal to the threshold (1.0).
- Timestamp:	Tuesday 19 November, 2024 11:45:43 UTC
- AWS Account:	390403872402
- Alarm Arn:	arn:aws:cloudwatch:us-east-1:390403872402:alarm:NginxServiceAlarm

Threshold:

- The alarm is in the ALARM state when the metric is GreaterThanOrEqualToThreshold 1.0 for at least 1 of the last 1 period(s) of 300 seconds.

Monitored Metric:

- MetricNamespace:	AWS/EC2
- MetricName:	StatusCheckFailed
- Dimensions:	[InstanceId = i-097671fe115ecf24]
- Period:	300 seconds
- Statistic:	Average
- Unit:	not specified

State Change Actions:

- OK: [arn:aws:sns:us-east-1:390403872402:NginxServiceAlert]
- ALARM: [arn:aws:sns:us-east-1:390403872402:NginxServiceAlert]
- INSUFFICIENT_DATA: [arn:aws:sns:us-east-1:390403872402:NginxServiceAlert]

..

If you wish to stop receiving notifications from this topic, please click or visit the link below to unsubscribe:
<https://sns.us-east-1.amazonaws.com/unsubscribe.html?SubscriptionArn=arn:aws:sns:us-east-1:390403872402:NginxServiceAlert:0eb46e5d-ce13-4548-8f99-60e8d6f164cb&Endpoint=sjedf597@outlook.com>

Please do not reply directly to this email. If you have any questions or comments regarding this email, please contact us at <https://aws.amazon.com/support>

Reply

Forward

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.