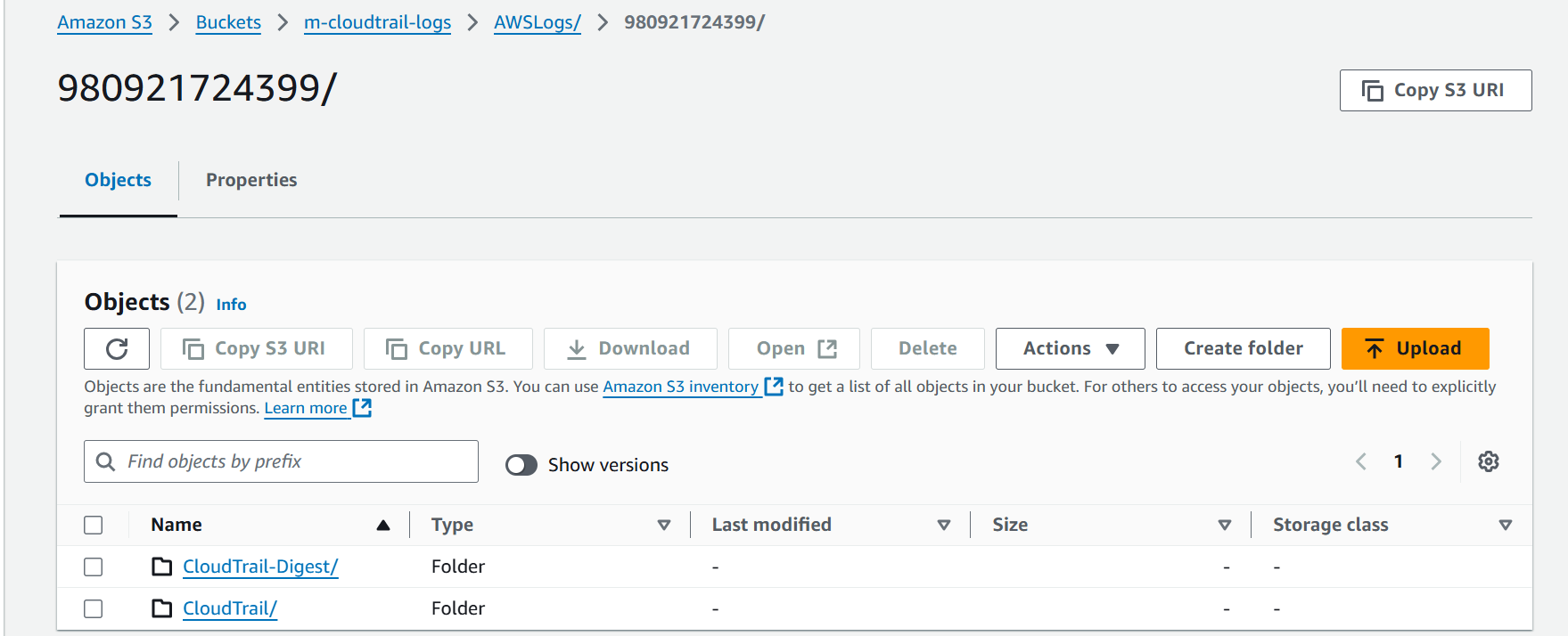
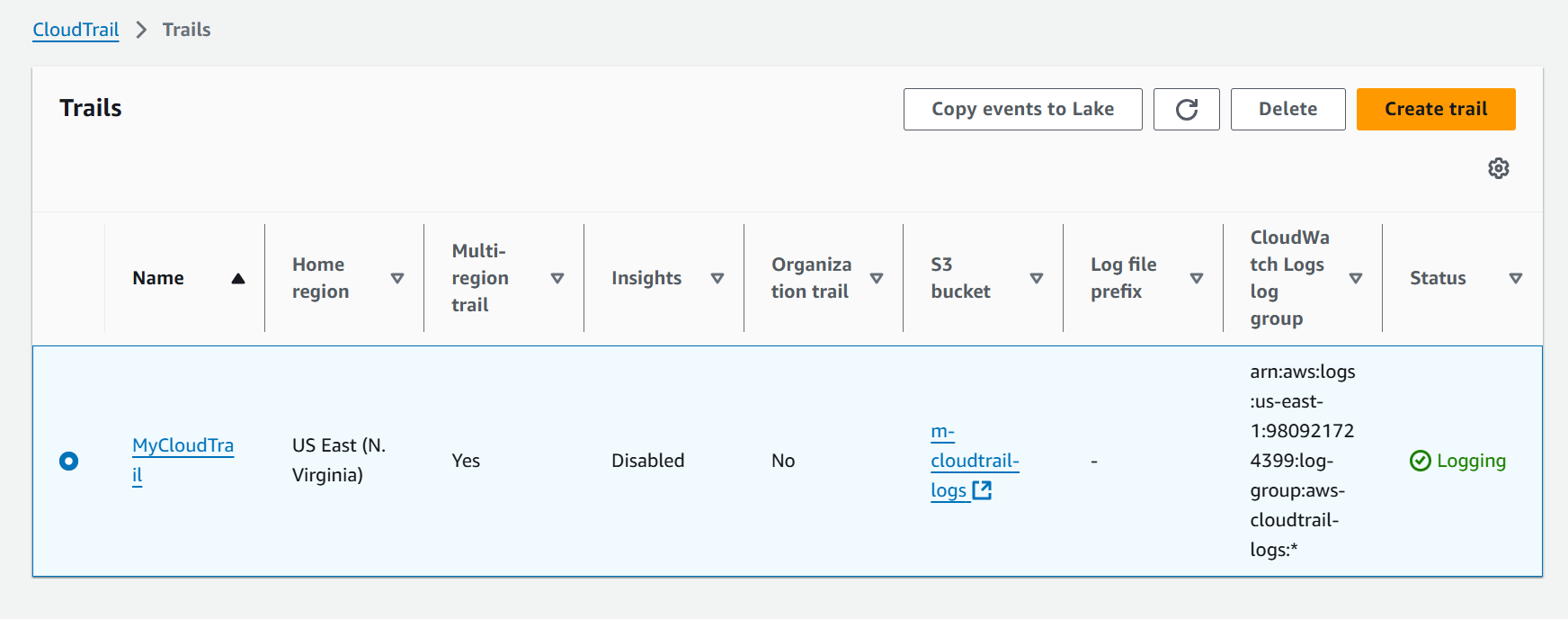
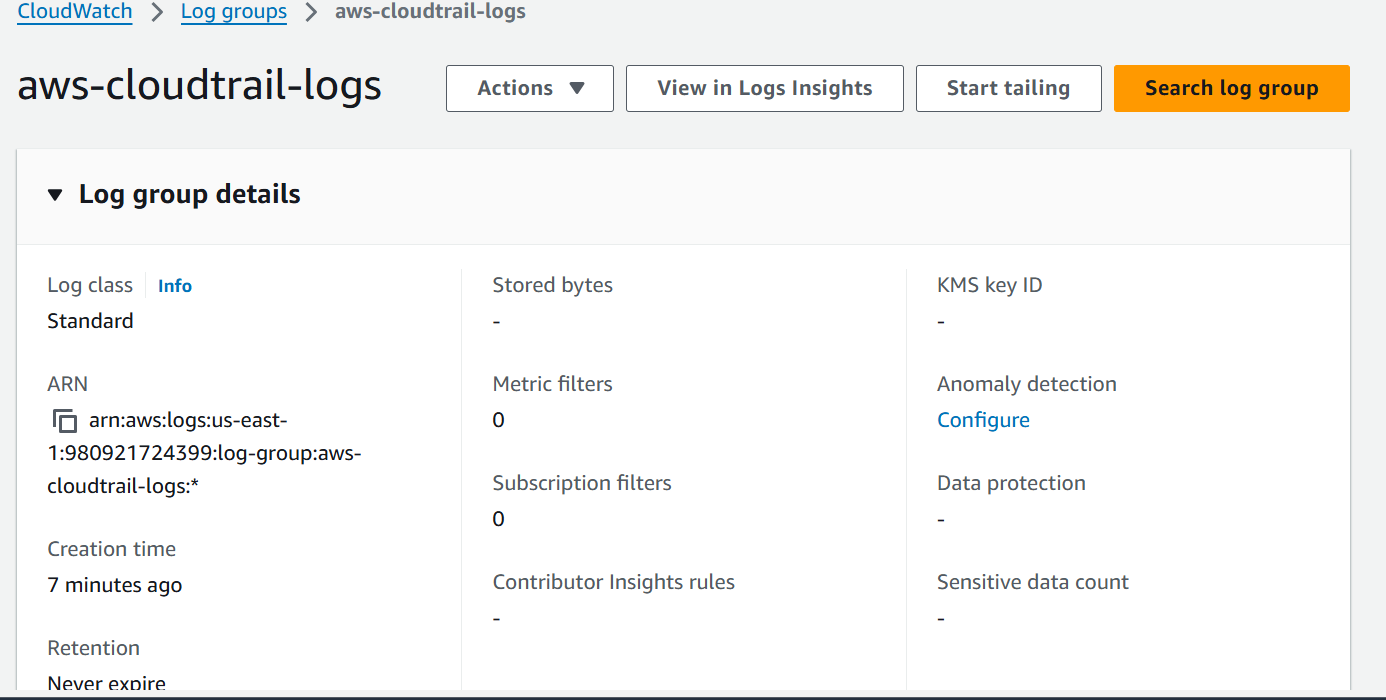
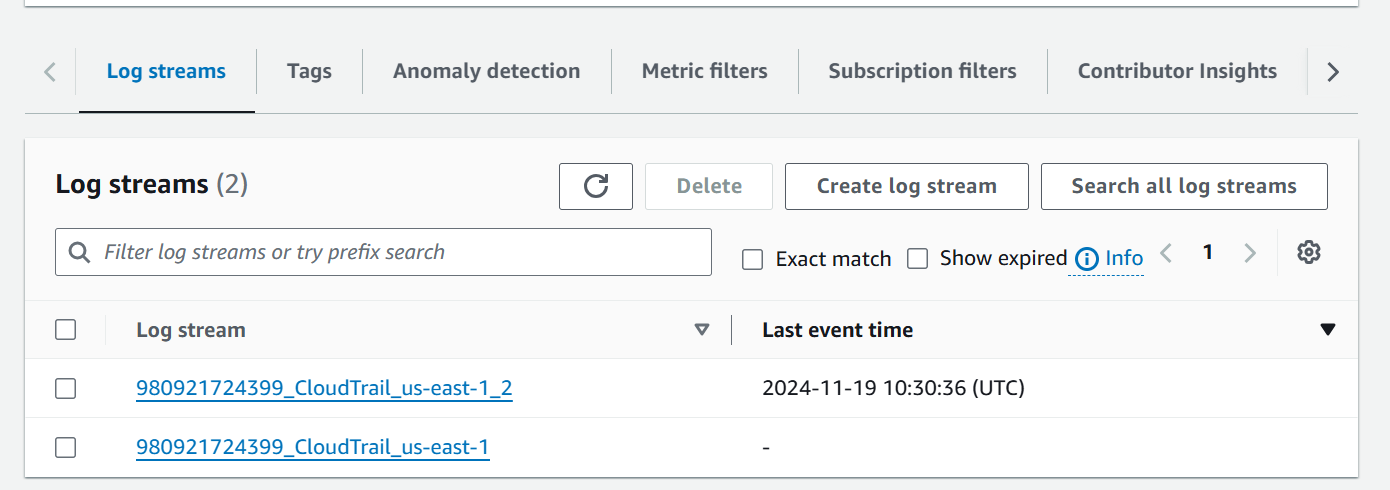
**Cloud trail & Cloud watch.**

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

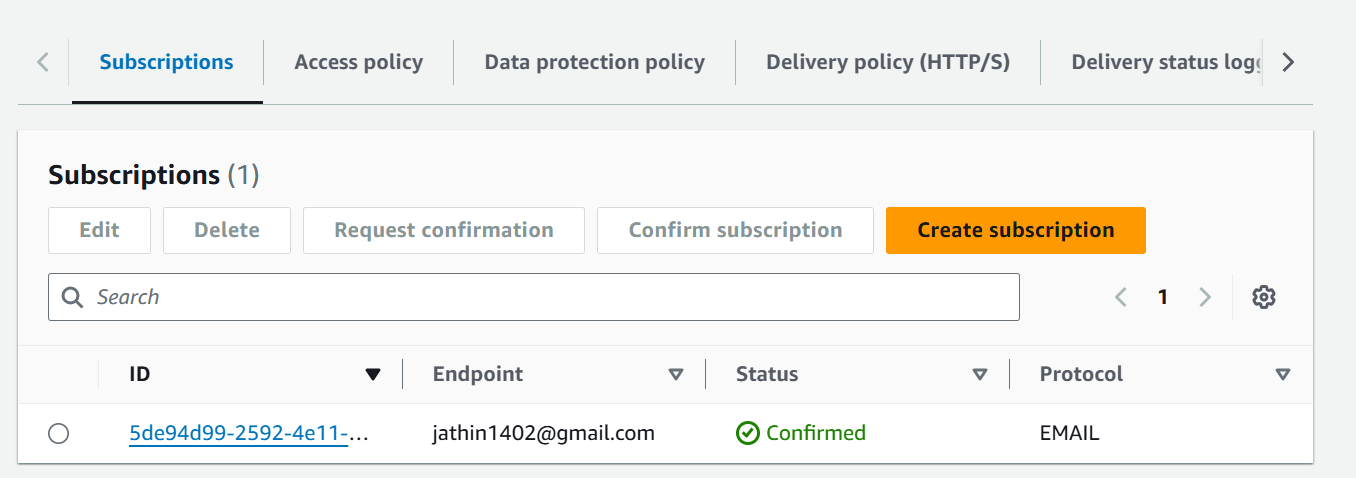


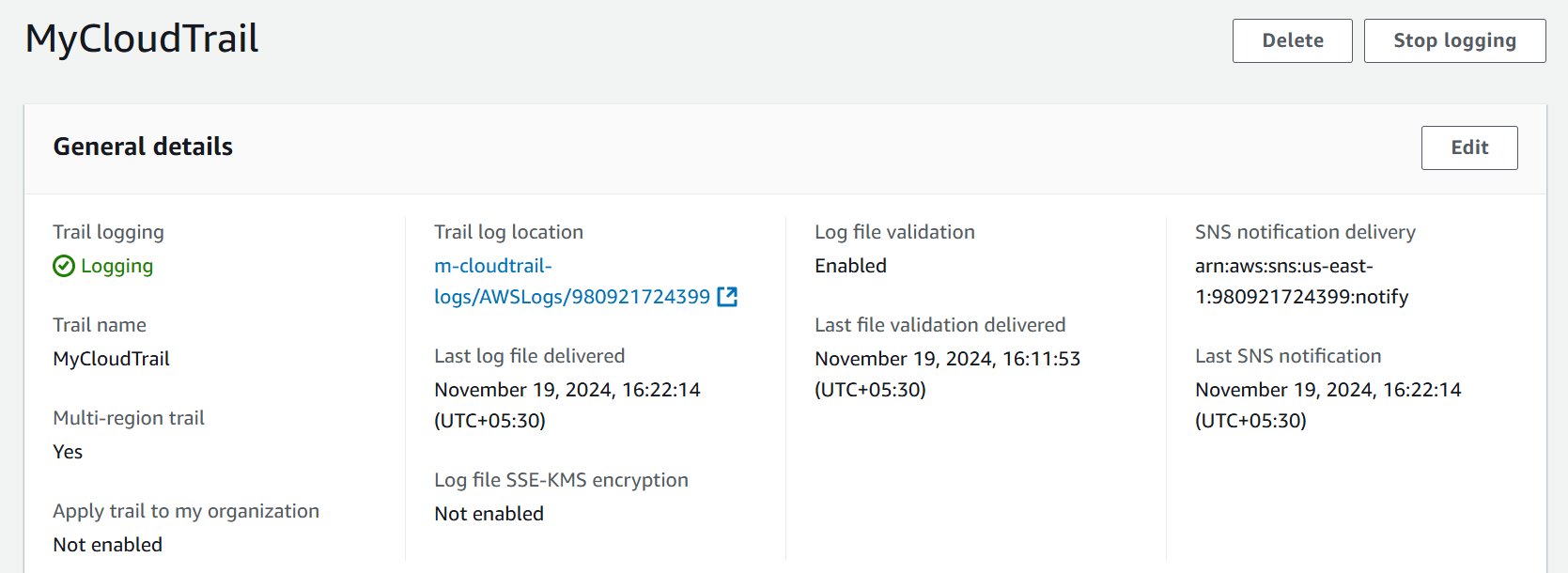




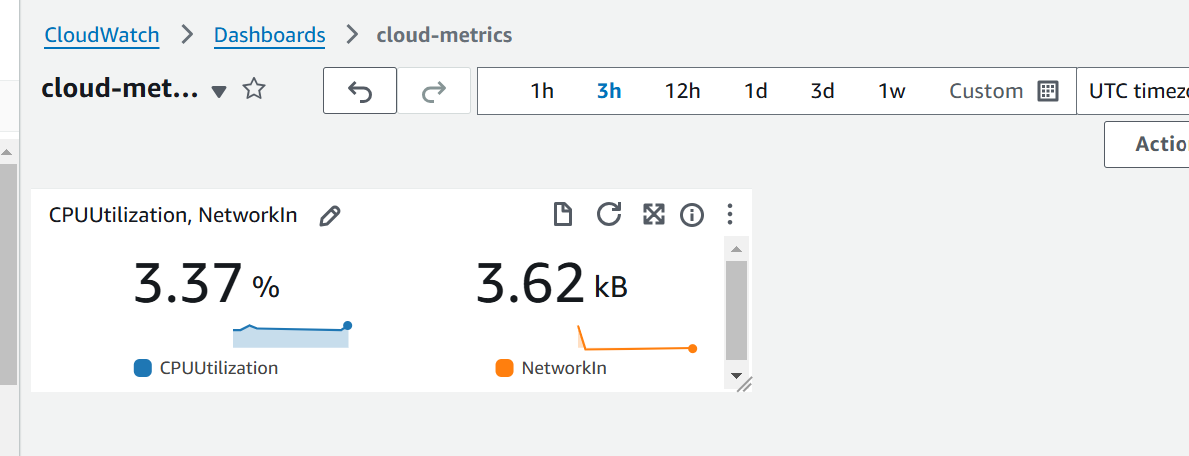


1. Enable SNS for cloudtrial to send alert on email.

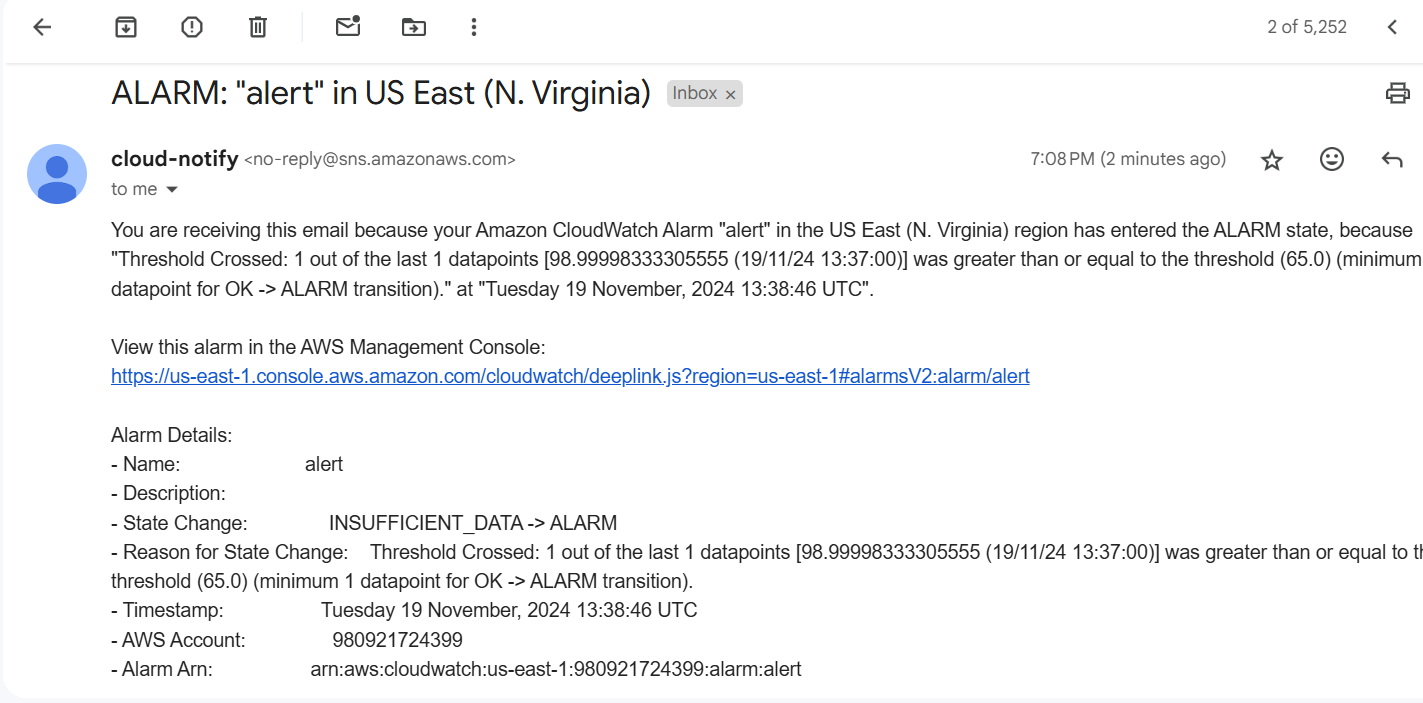
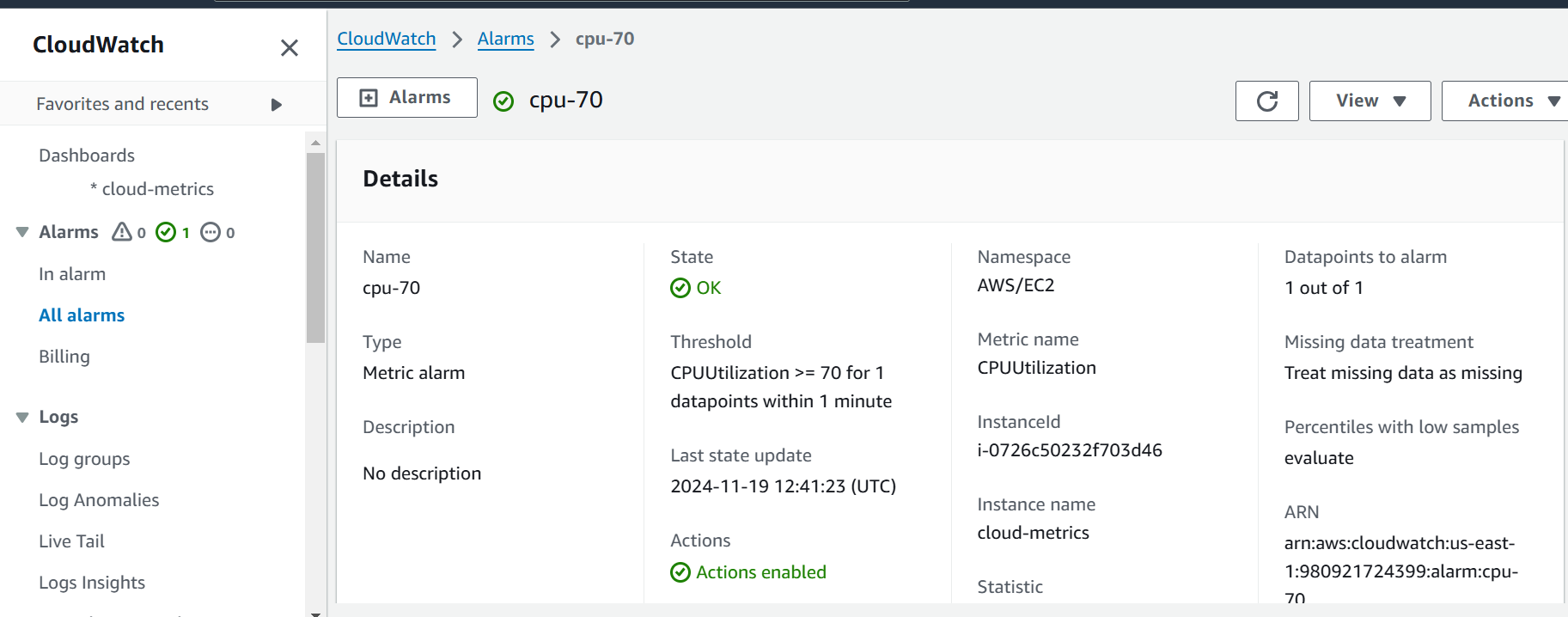




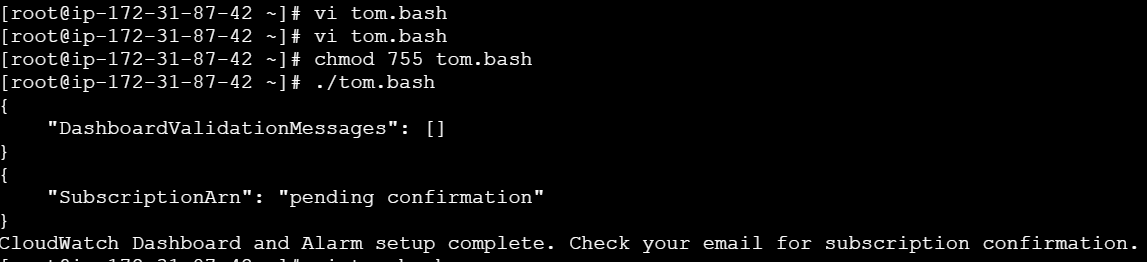
1. Configure cloud watch monitoring and record the cpu utilization and other metrics of ec2.



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

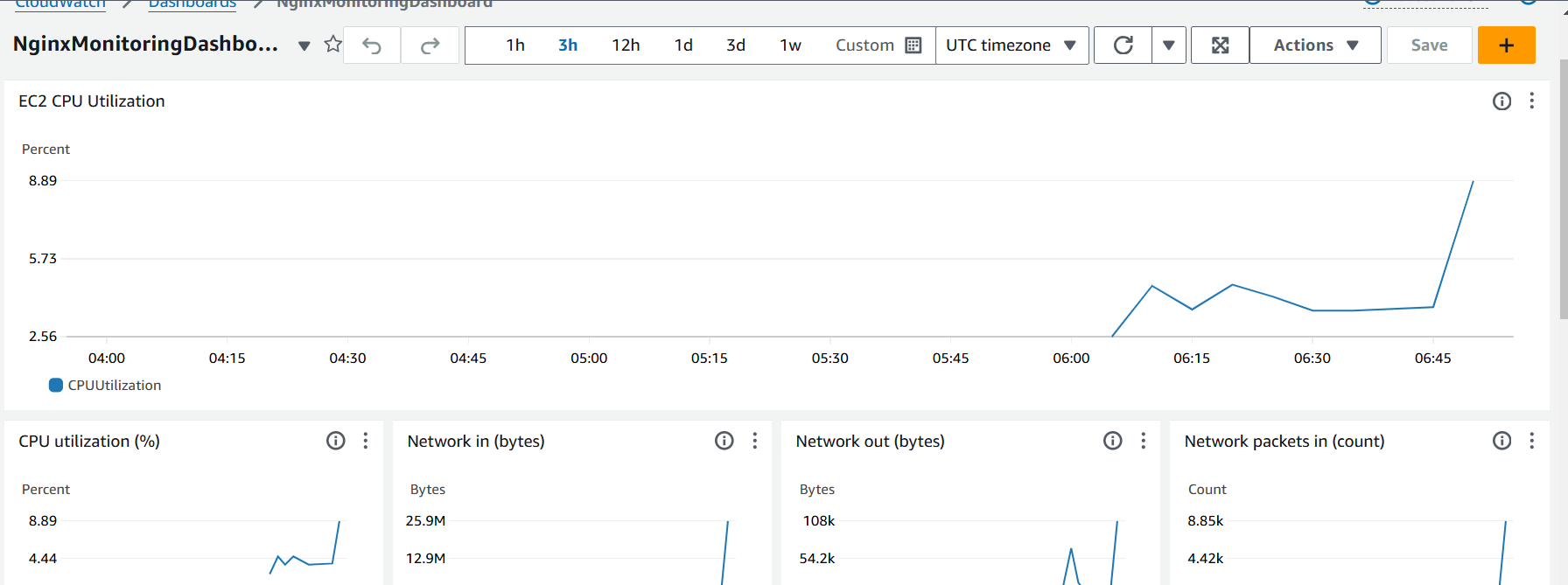


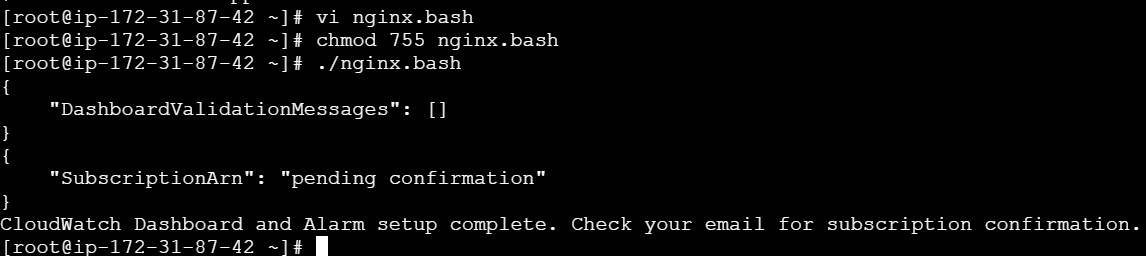
5)Create Dashboard and monitor tomcat service wether it is running or not and send the alert.



#!/bin/bash   # Variables  
   DASHBOARD\_NAME="Tomcat"  
   ALARM\_NAME="alert"  
   TOPIC\_NAME="notify"  
   TOPIC\_EMAIL="[jathin1402@gmail.com](mailto:jathin1402@gmail.com)"  
   INSTANCE\_ID="i-07492b841393e8e52" # 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."

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





#!/bin/bash   # Variables  
   DASHBOARD\_NAME="NginxMonitoringDashboard"  
   ALARM\_NAME="alert"  
   TOPIC\_NAME="notify"  
   TOPIC\_EMAIL="[jathin1402@gmail.com](mailto:jathin1402@gmail.com)"  
   INSTANCE\_ID="i-07492b841393e8e52" # 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 "$TOPIC\_EMAIL"   # Create CloudWatch Alarm for EC2 Status Check  
   aws cloudwatch put-metric-alarm --alarm-name "$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"   # Monitor NGINX logs via CloudWatch logs (optional)  
   # You can set up NGINX access logs or error logs to be pushed to CloudWatch.   echo "CloudWatch Dashboard and Alarm setup complete. Check your email for subscription confirmation."