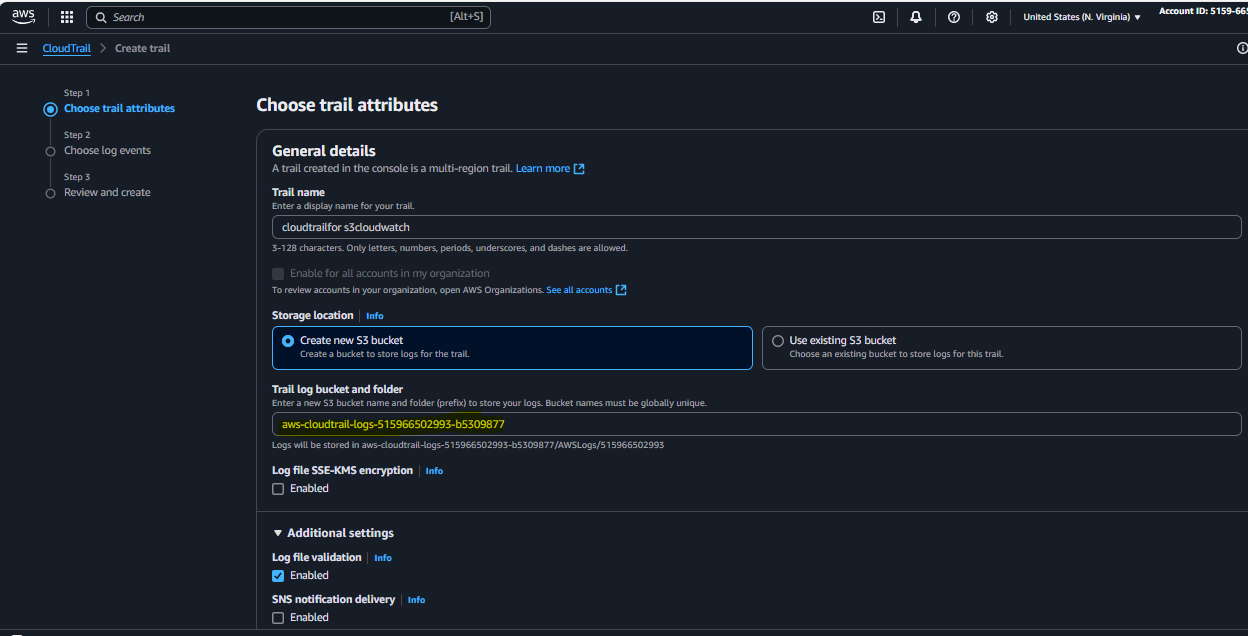
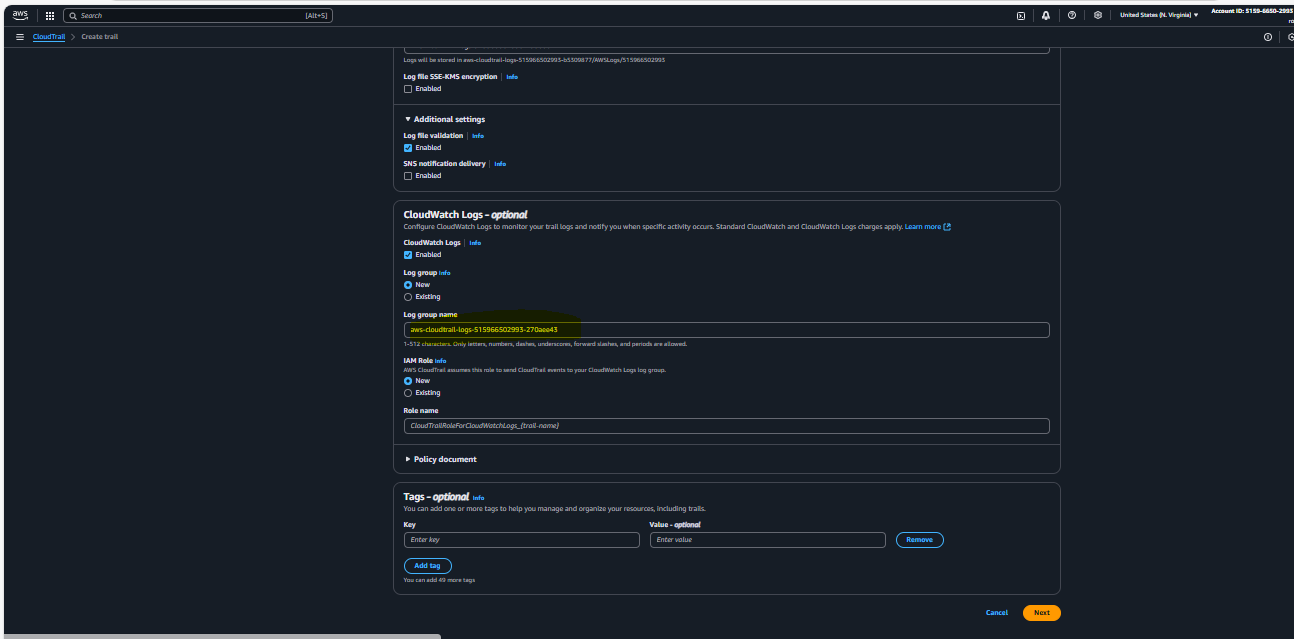
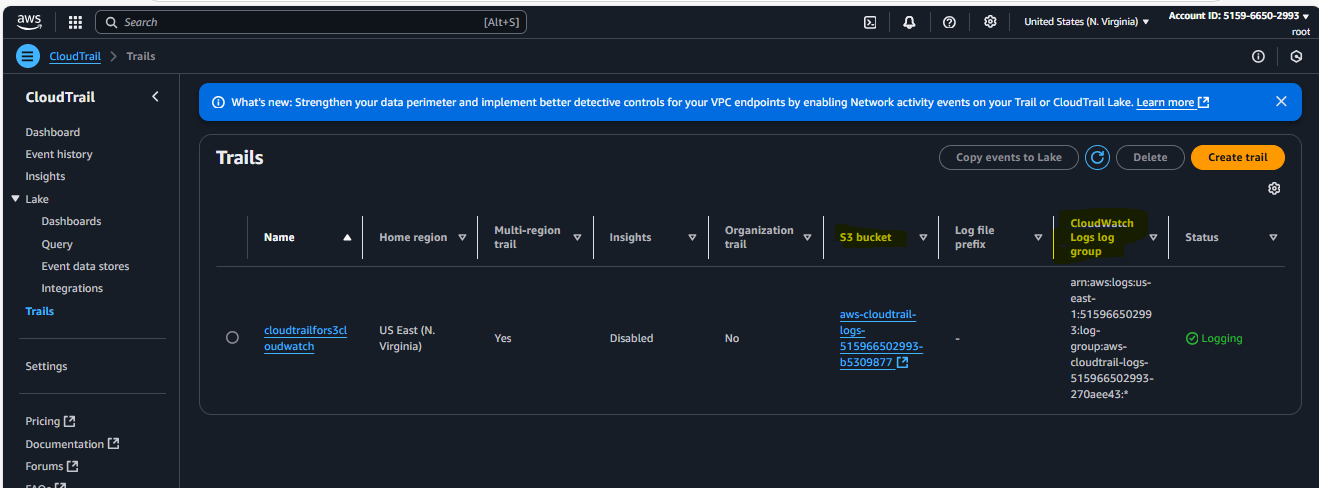
1. Enable cloudtrail monitoring and store the events in s3 and cloudwatch log events.

**Goto AWS console and search for Cloudtrail-🡪Create trail work flow**

**Step1: Choose trail attributes and give cloudtrail name and select s3 bucket and cloudwatch logs**

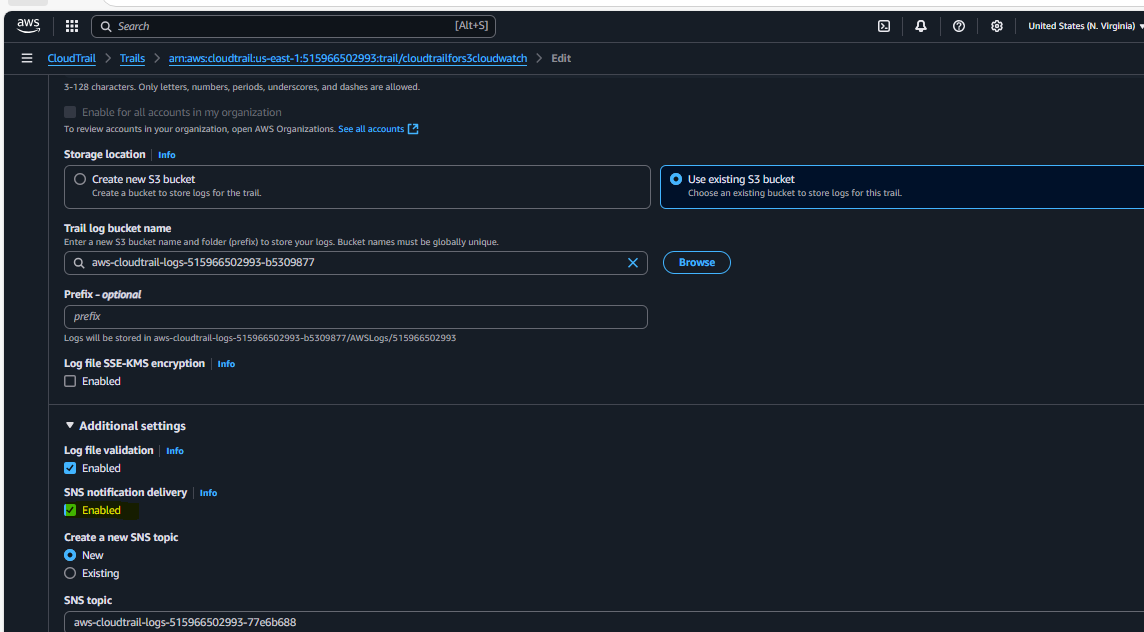
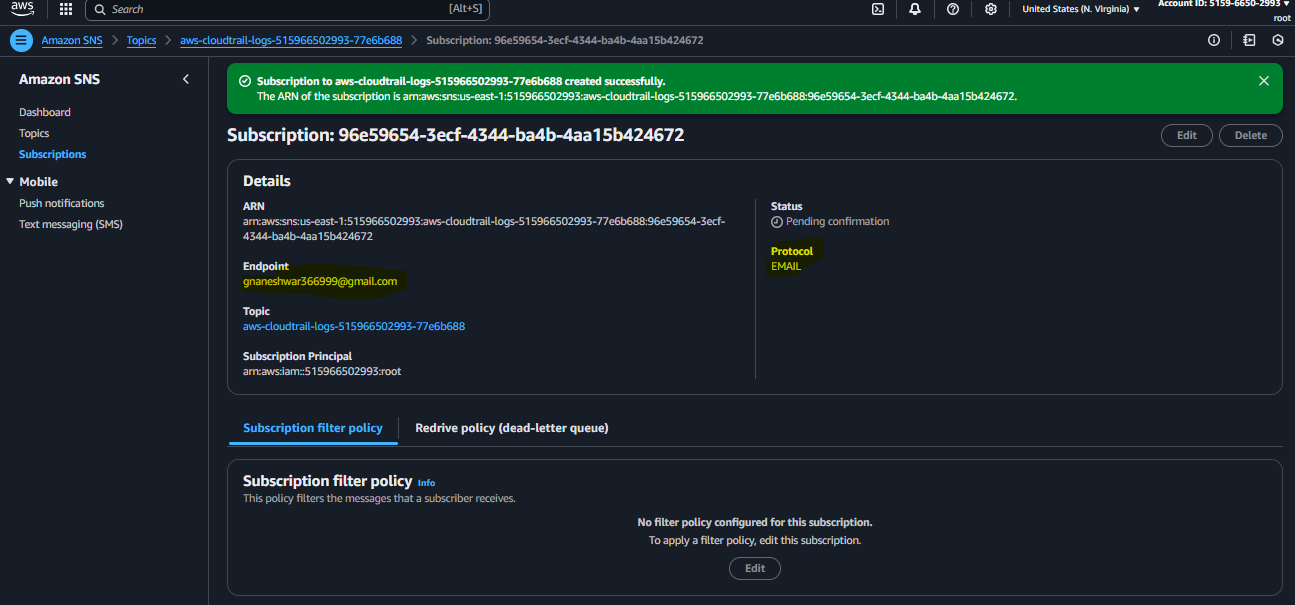
**Step 2:choose log events type and Manage events activity type**

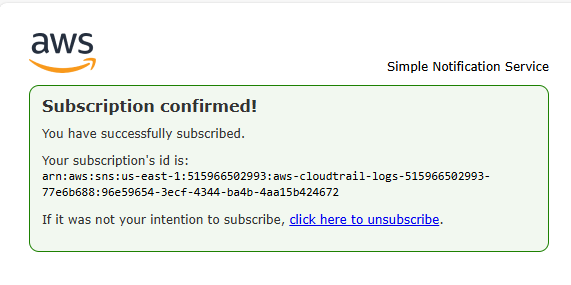
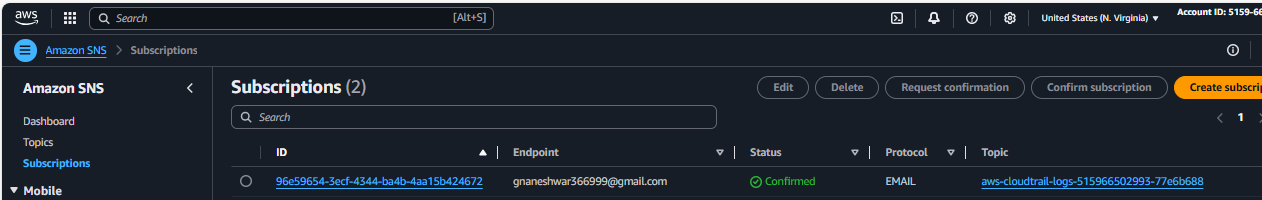
**Step3: Review and create trail.**

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

**Step 1: If you have a existing Topic in SNS use it and enable SNS notification delivery option** **or else create new topic**

**Step 2: Goto Subscription--🡪Create subscription -🡪Choose protocol--🡪Email-🡪verify confirmation**  

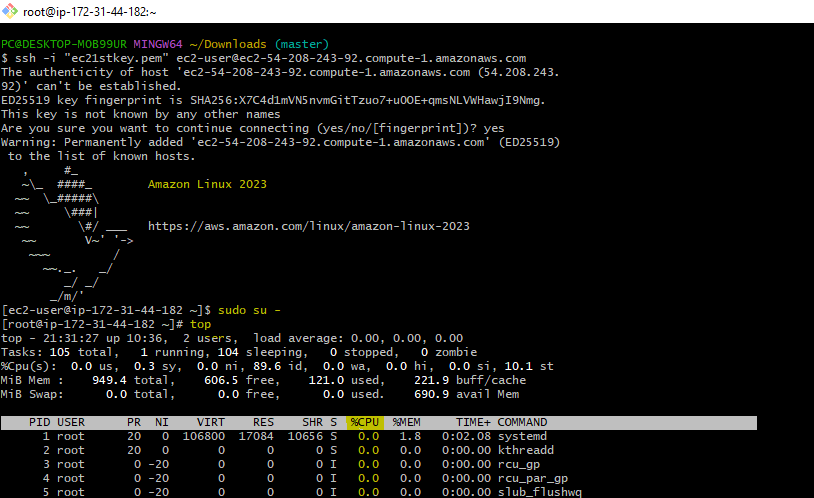
**You will get a subscription confirmation email and**  confirm it 

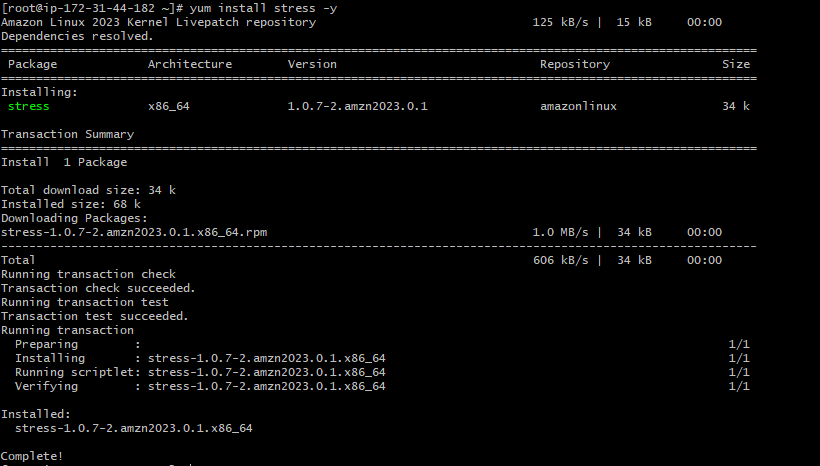
1. Configure cloud watch monitoring and record the cpu utilization and other metrics of ec2.

**First launch an ec2 instance and connect to it and switch to super user by using “sudo su –”**

**And for practical purpose we are adding some fake load to the instance by installing stress package**

**yum install stress –y**



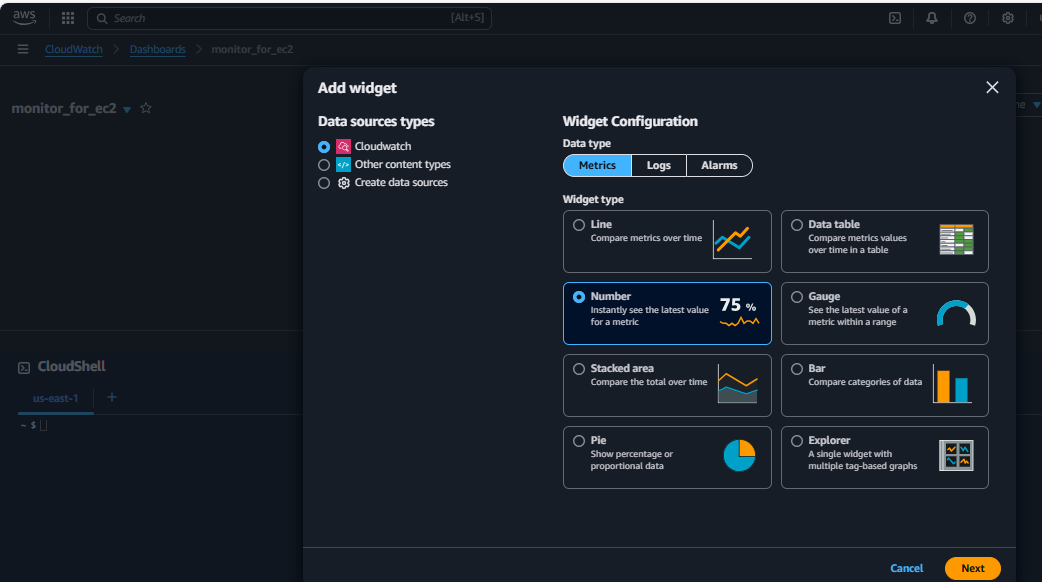


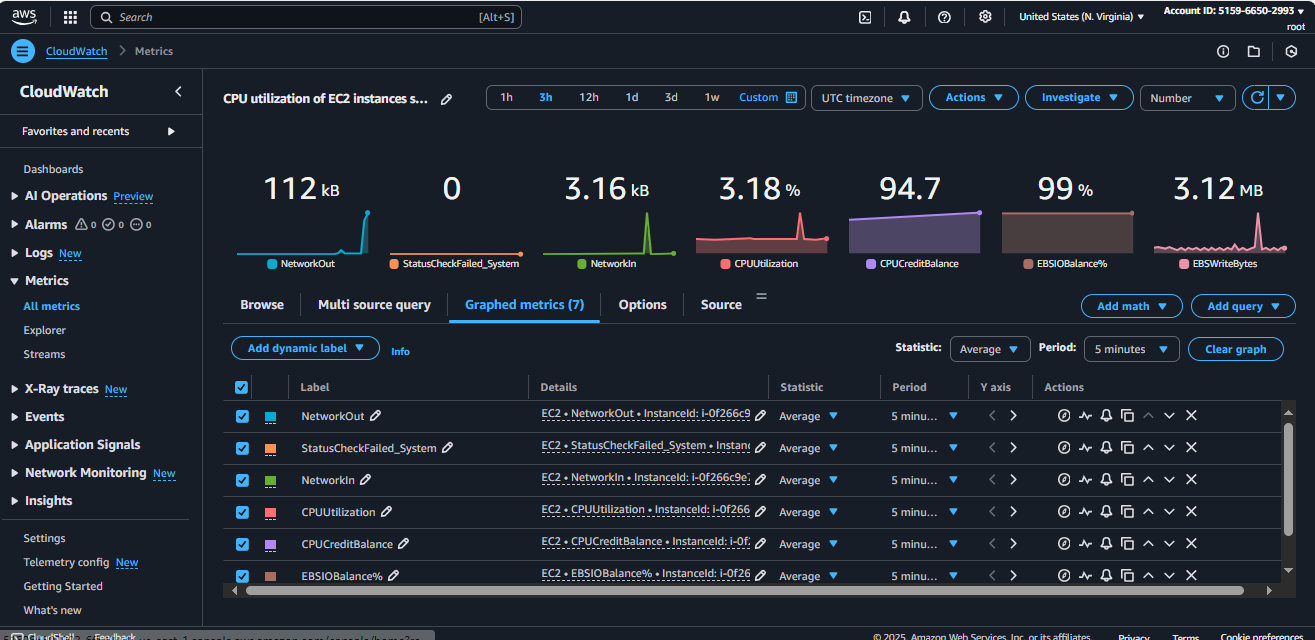
**Now goto Cloudwatch-🡪create dashboard-🡪give name -🡪Add widget🡪Cloudwatch-🡪choose data type as Metrics --🡪number click next**

**Now create a dashboard**

**Goto Cloudwatch--🡪create dashboard🡪give name & create**

**In the next step Add Widget -🡪choose 1 type (Number) click on Next ,then after-🡪 “Add metric graph” u can see AWS default metrics like (ec2,EBS,ALB,NAT etc.,)**





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

**Step 1: Goto cloudwatch-🡪Alarms-🡪Create alarm-🡪select metric**

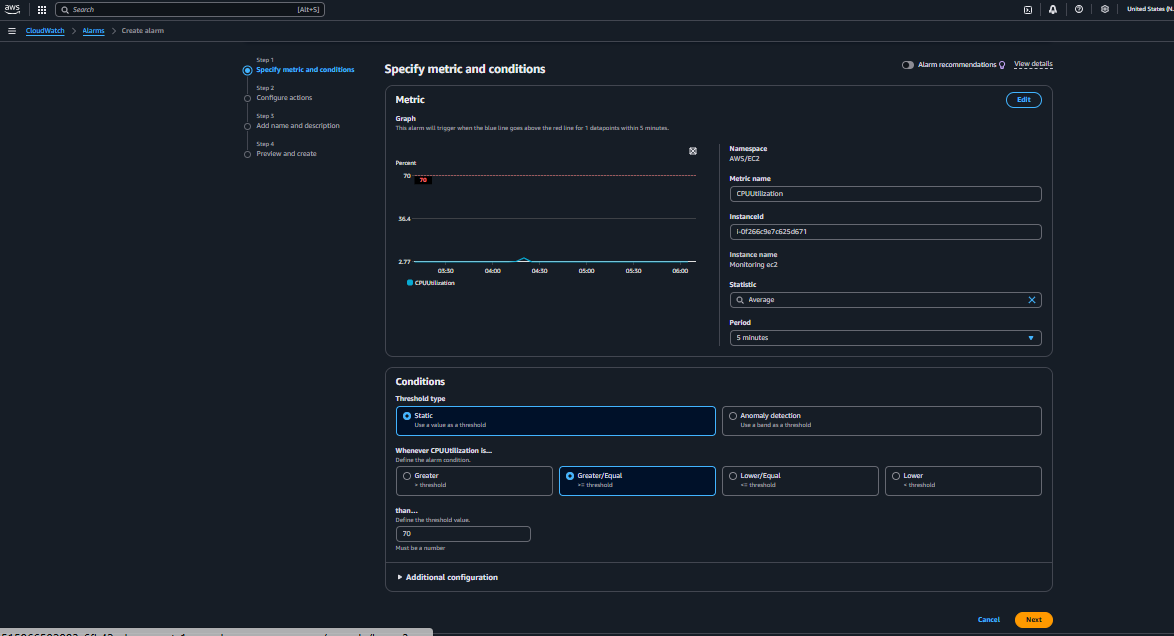
* **Metric-**🡪**choose threshold type as “Static”**
* **Choose WHENEVER CPU IS UTILISATION IS -**🡪 **Grater than or equals to**
* **Than--**🡪 **give value 70**

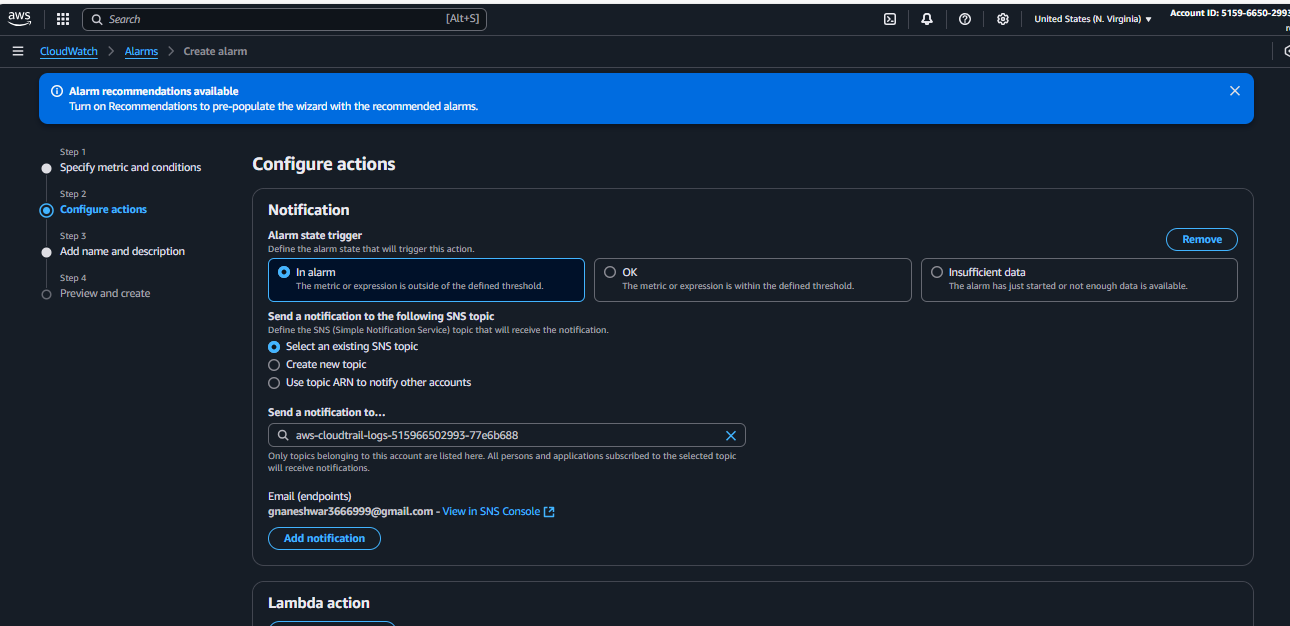
**Step 2:Configure Action -🡪Alarm state trigger**

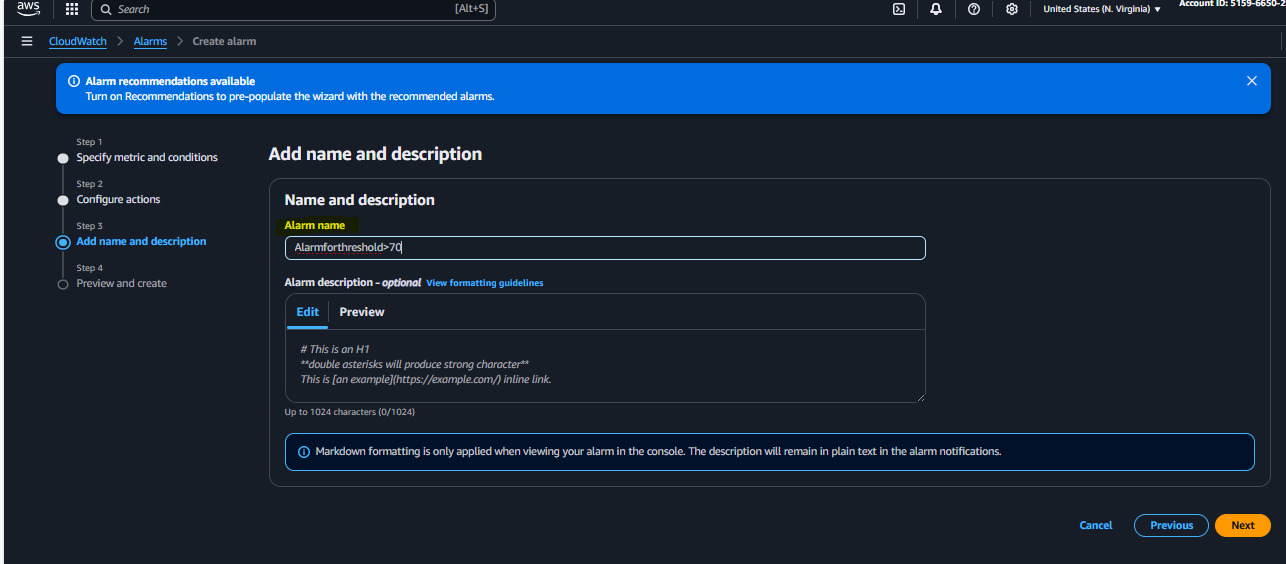
* **ALARM STATE TRIGGER-🡪In Alarm**
* **Send notification to SNS type-🡪 choose email and give your email id**
* **And use all default otions**

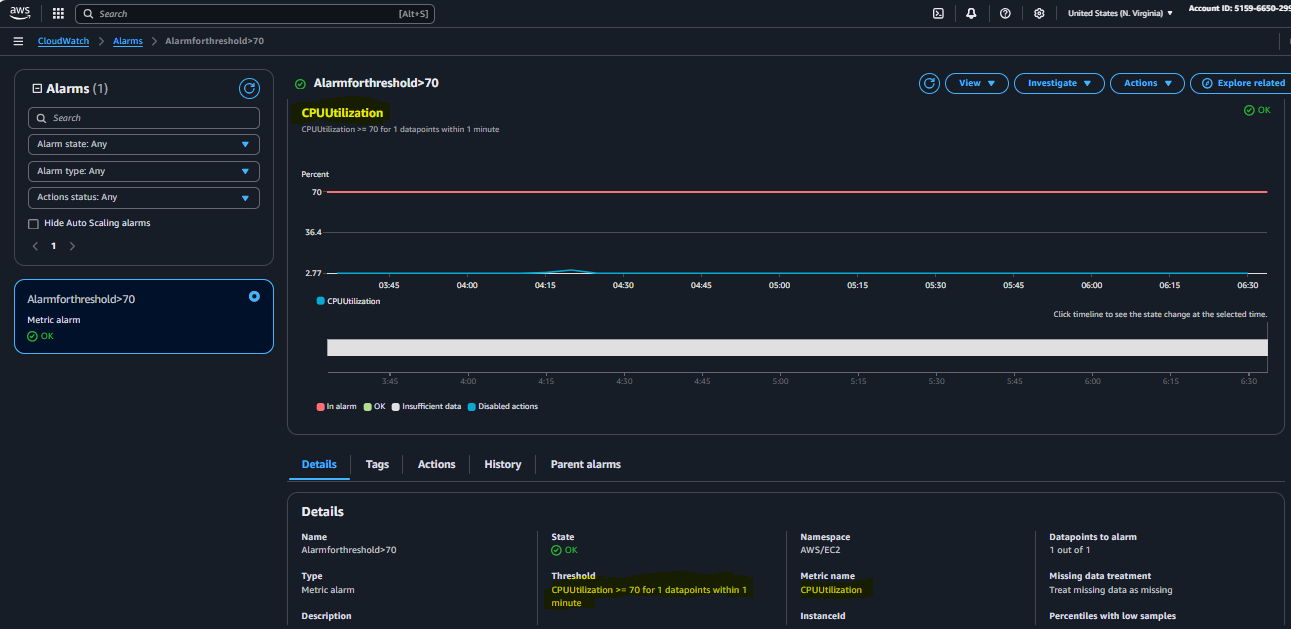
**Step 3: Add name and description**

**\*Give alarm name \_\_\_> make any required changes and create**







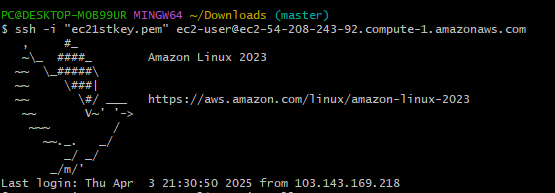


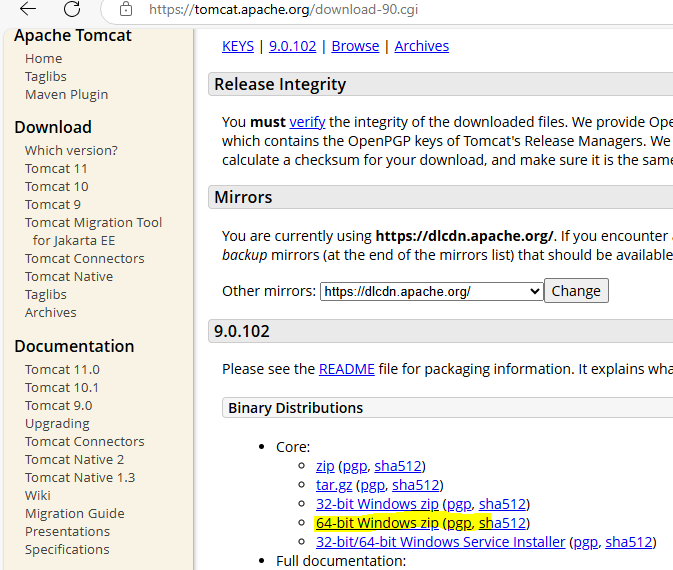
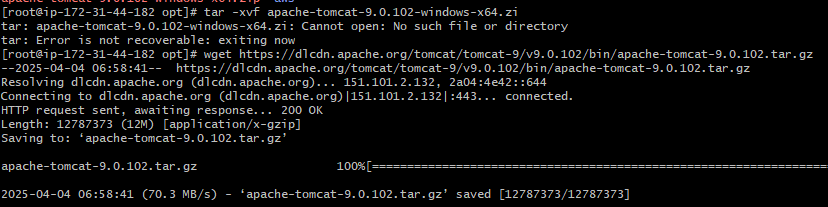
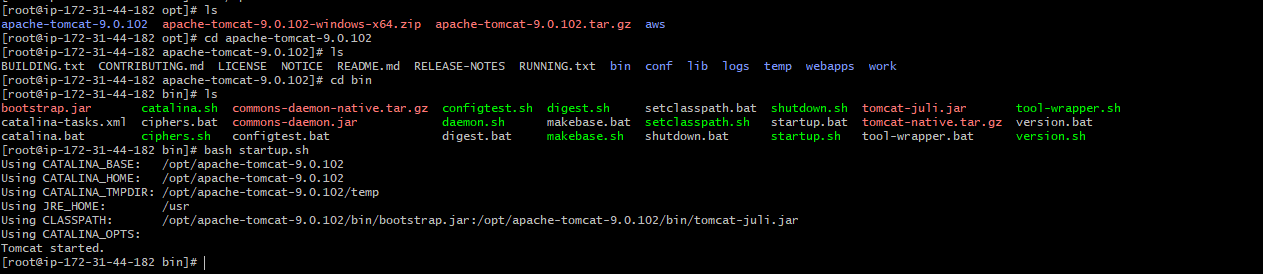
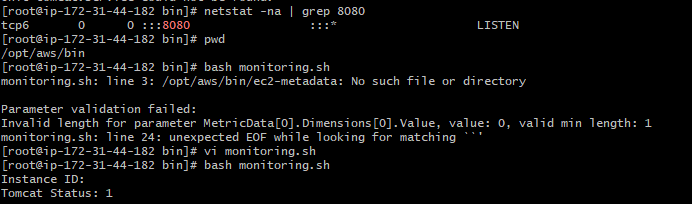
1. Create Dashboard and monitor tomcat service whether it is running or not and send the alert.

**First of all, create 1 ec2 instance and connect to the instance by using ssh**

**Switch to Super user “Sudo su –”**

**Install java and download apache tomcat by using “wget tomcat URL” (COPIED FROM WEB BROWSER) and untar the file and start the service by executing startup.sh and check on the browser using public IP**

**.**

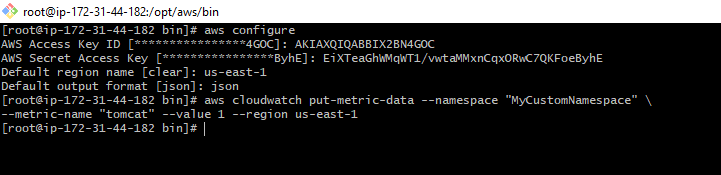
* You cannot create a namespace directly from the CloudWatch UI
* Instead, push a metric with a new namespace using AWS CLI

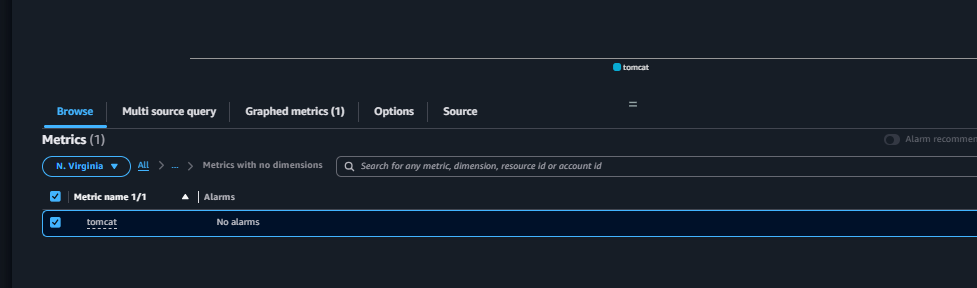
Run the following command to send a dummy metric:

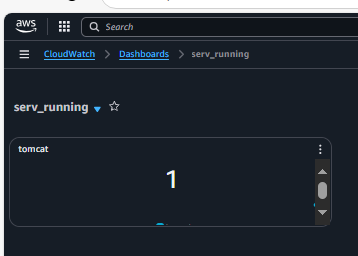
aws cloudwatch put-metric-data --namespace "MyCustomNamespace" \

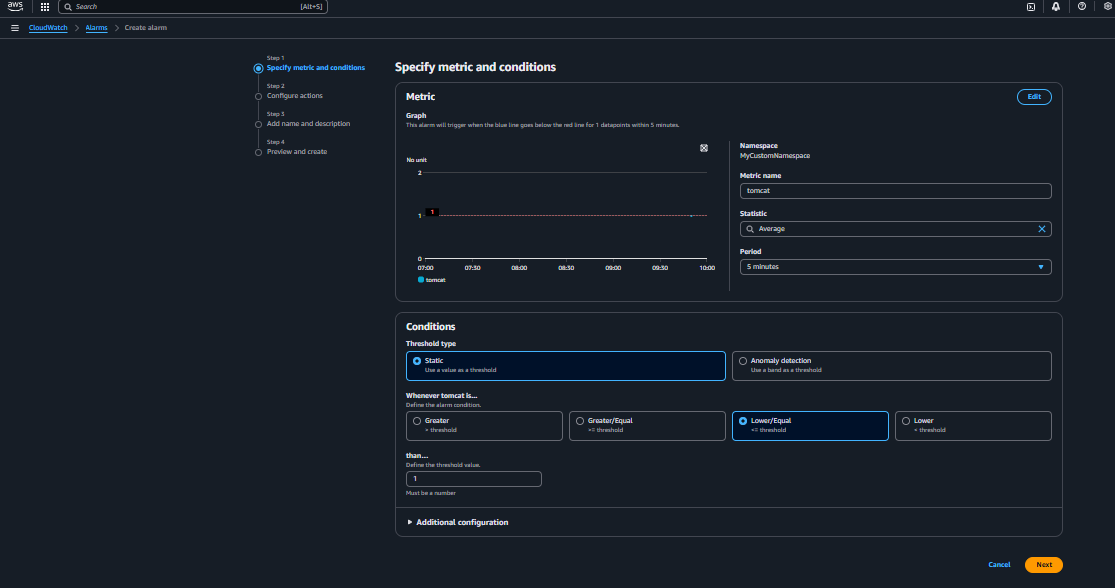
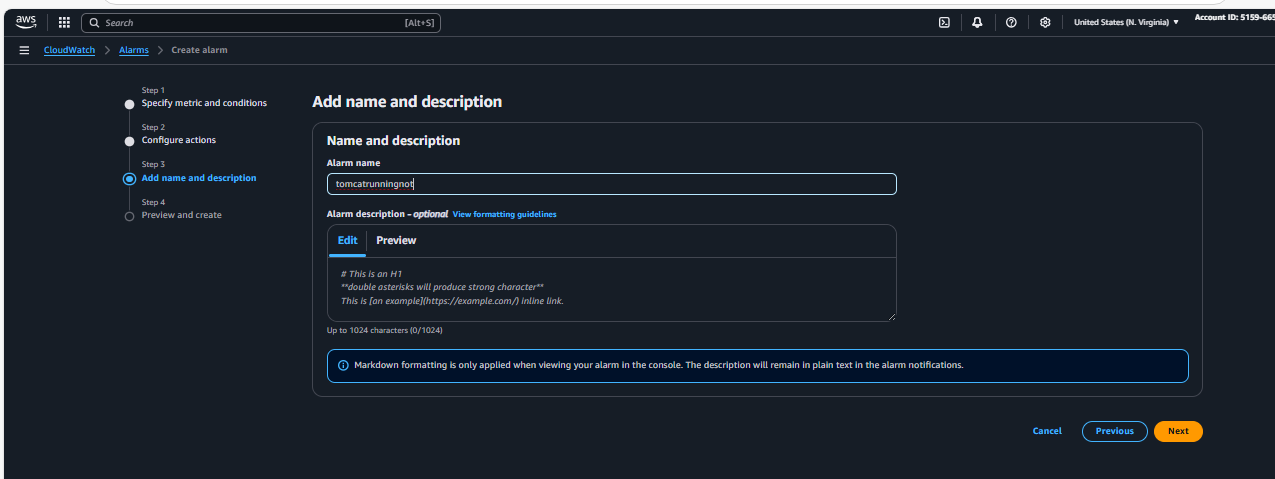
--metric-name "MyMetric" --value 1 --region us-east-1

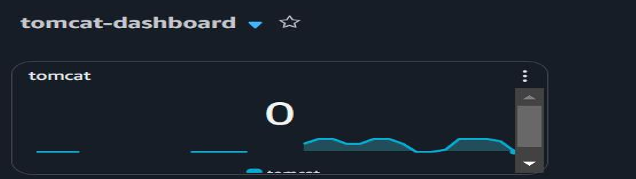
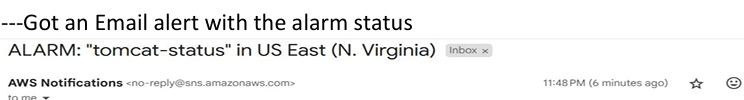
* Replace **"MyCustomNamespace"** with your desired namespace
* Change **"MyMetric"** to your metric name





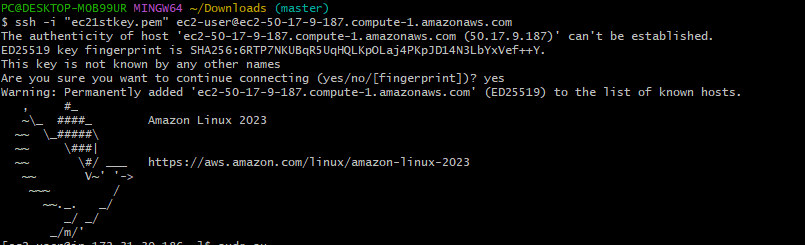
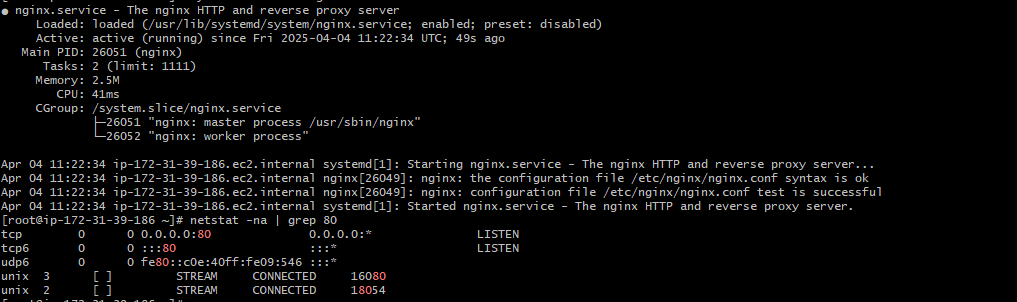
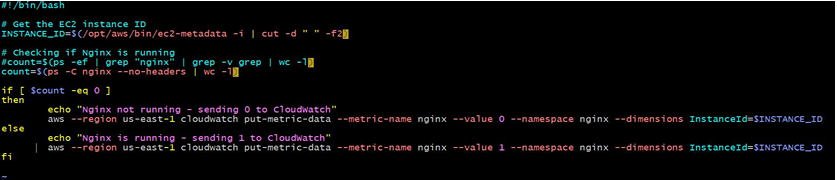


  now add crontab to this job to check whether its running or not for every minute and if we stop the service it will show “0” in the dashboard

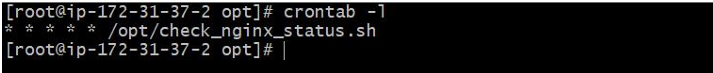
 

1. Create Dashboard and monitor nginx service to send the alert if nginx is not running.

**Create an ec2 instance with nginx configuration and connect to the instance by using ssh**

Set a crontab for this script every minute-



Monitoring Nginx running status through Cloudwatch dashboard





IF nginx is stopped (if not stopped,do stop manually) alarm will changed from OK to IN ALARM and SNS alerts will be sen to email

