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

Steps:

1. Go to cloudtrail and create a trail name and

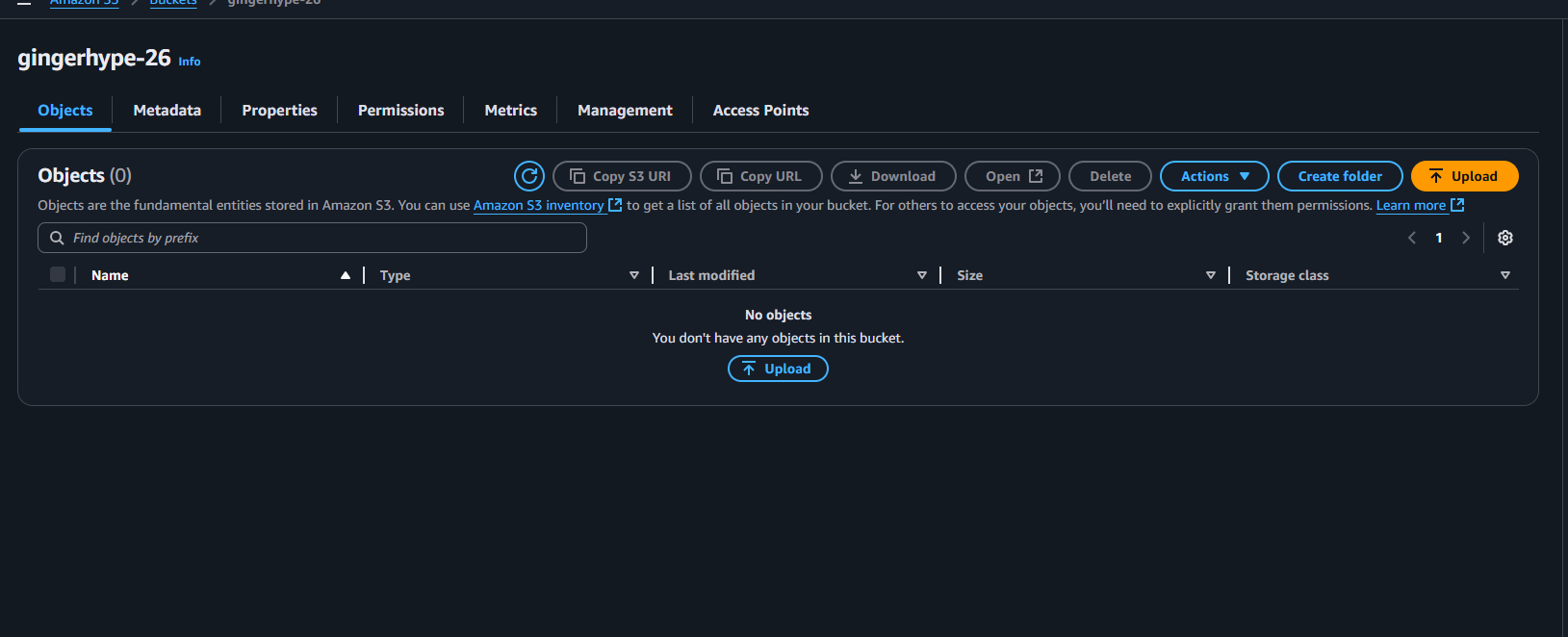
2. next select use existing s3 bucket

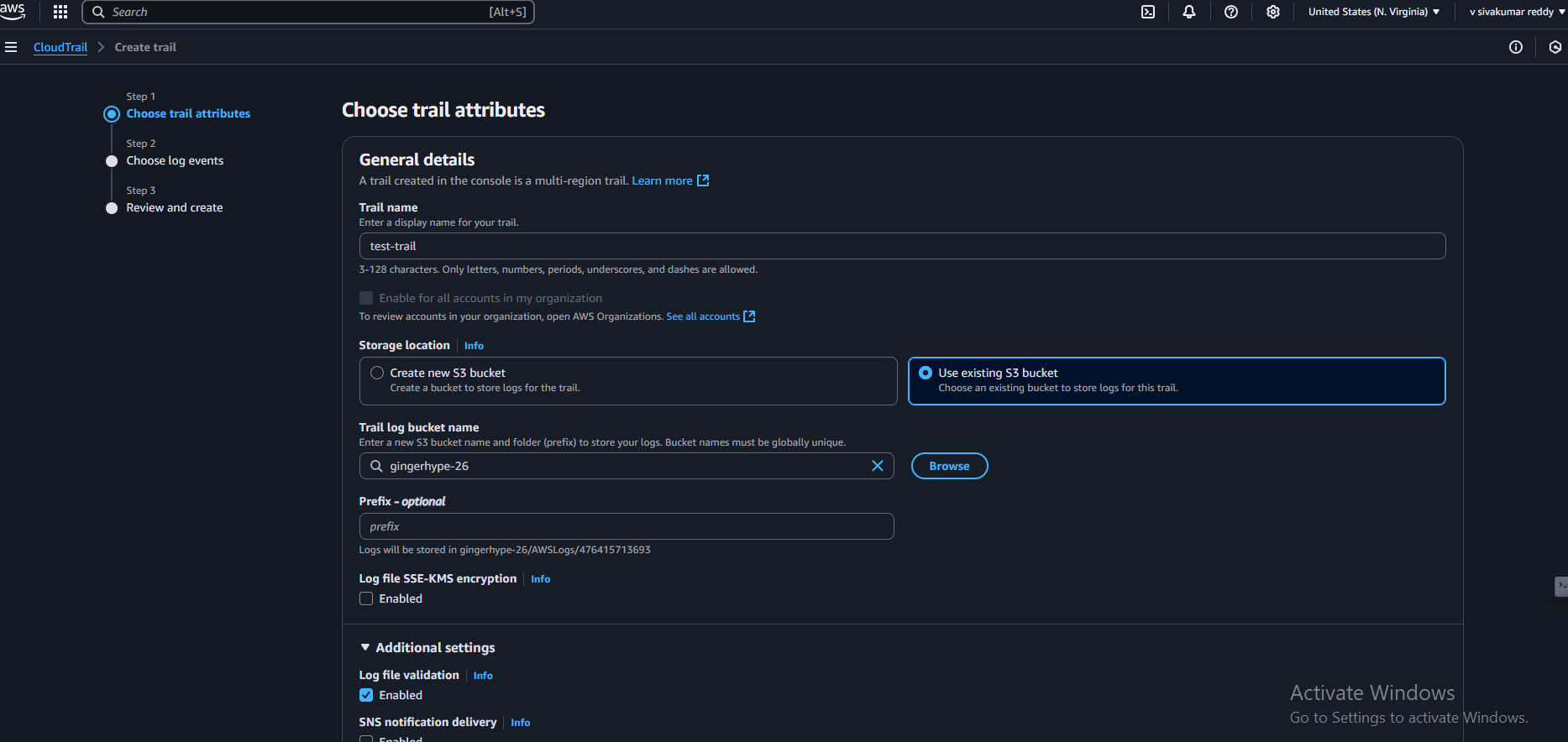
3. enable logfile validation

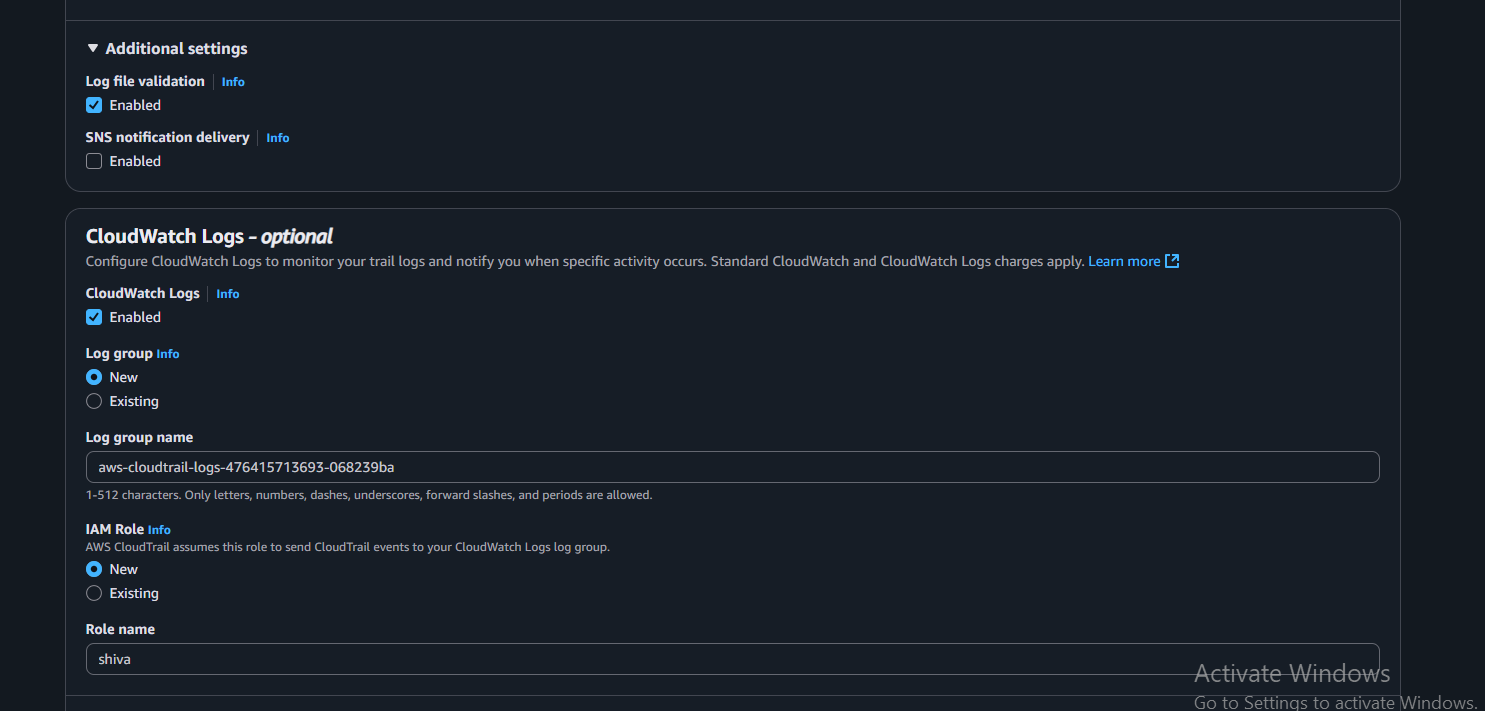
4. enable cloudwatch log and give iam role name

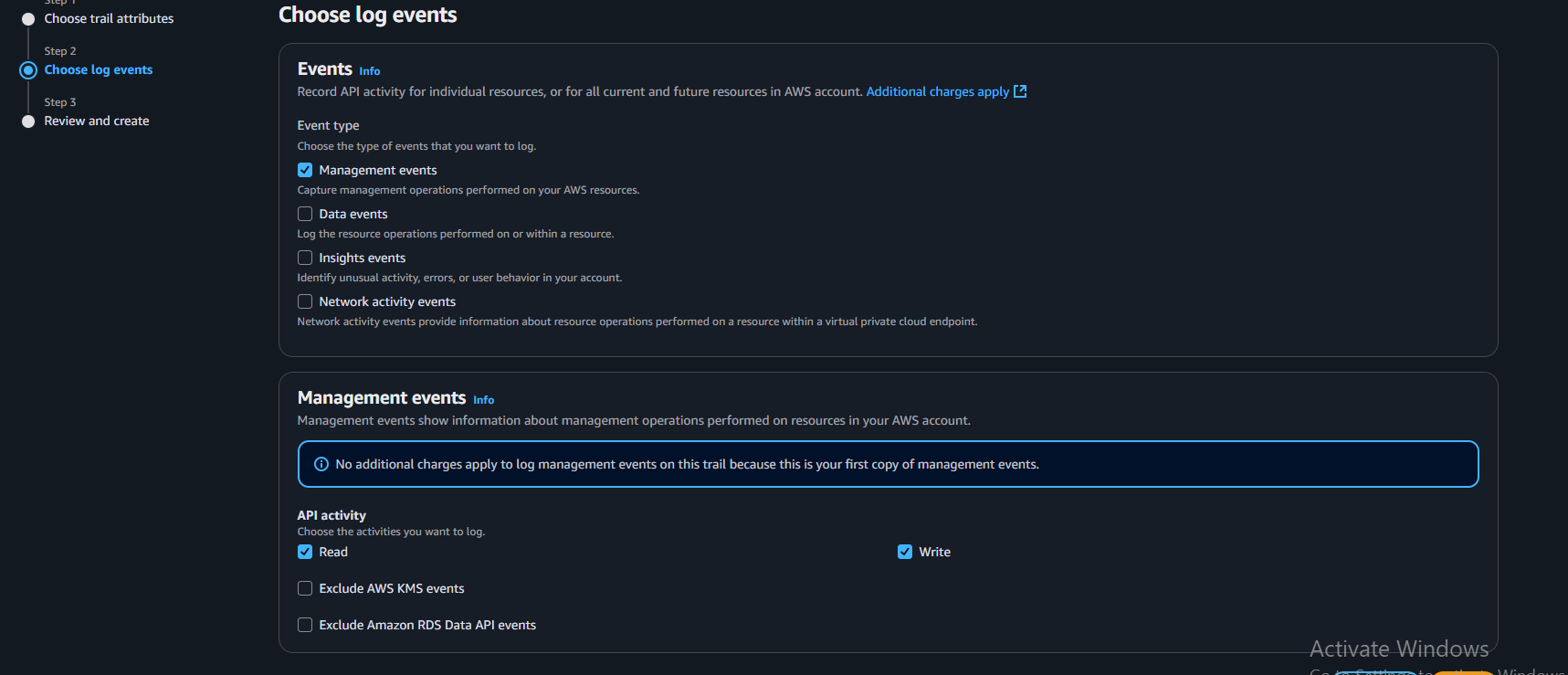
5. go to events and select management event

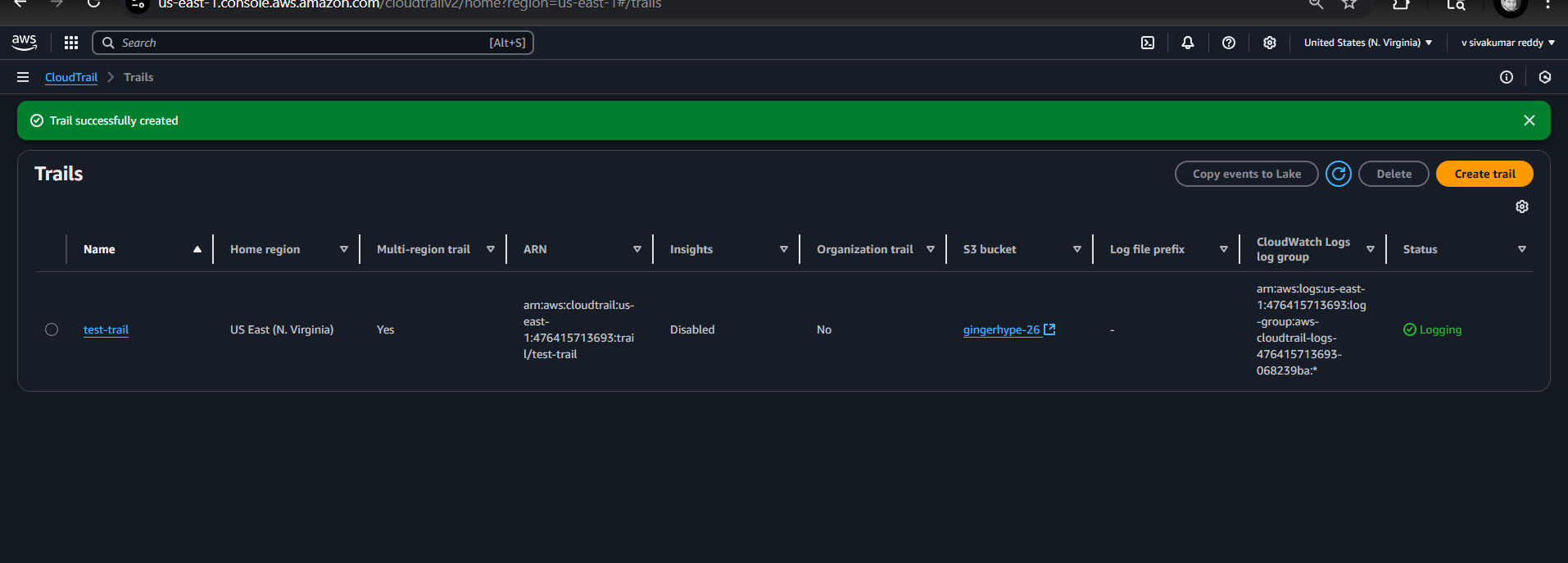
6. next click on create trail







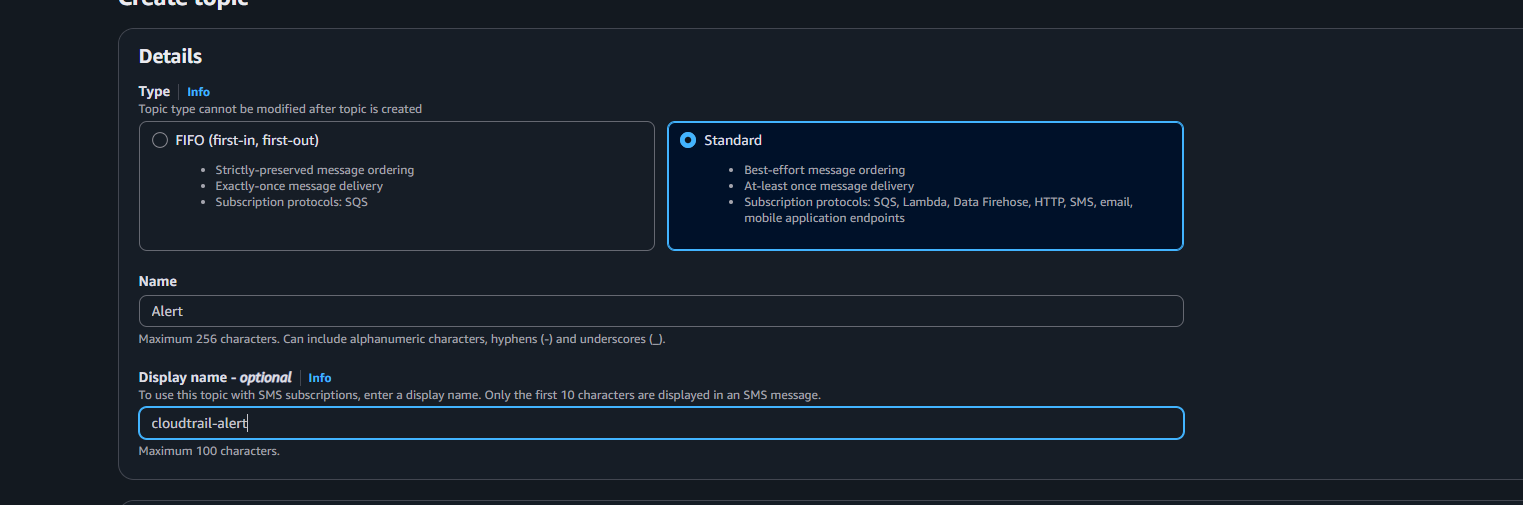


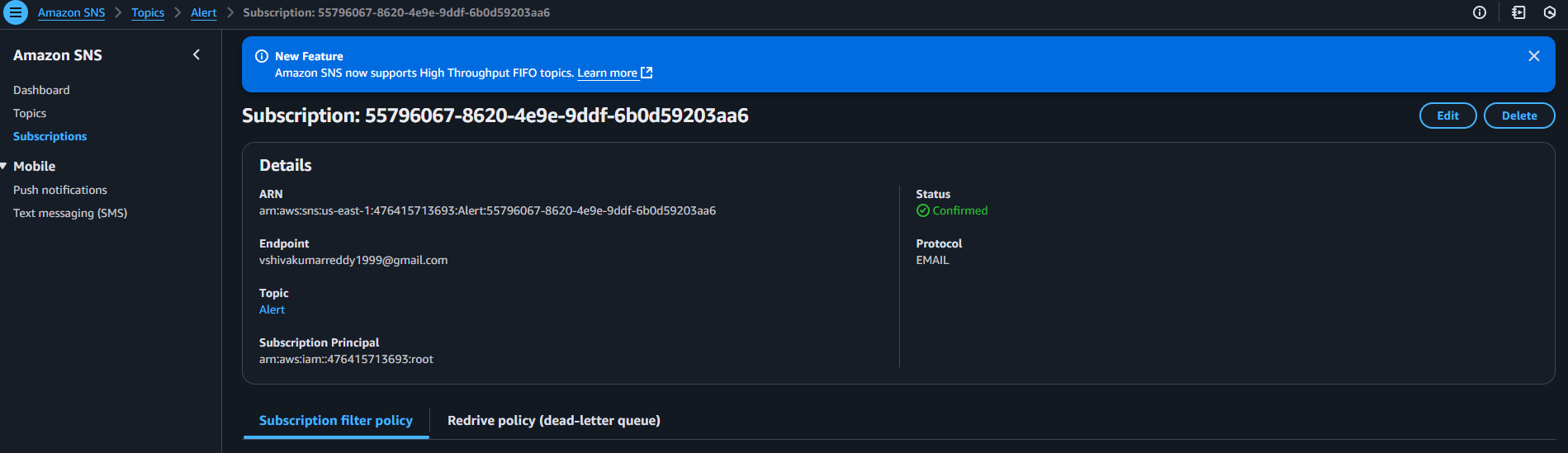


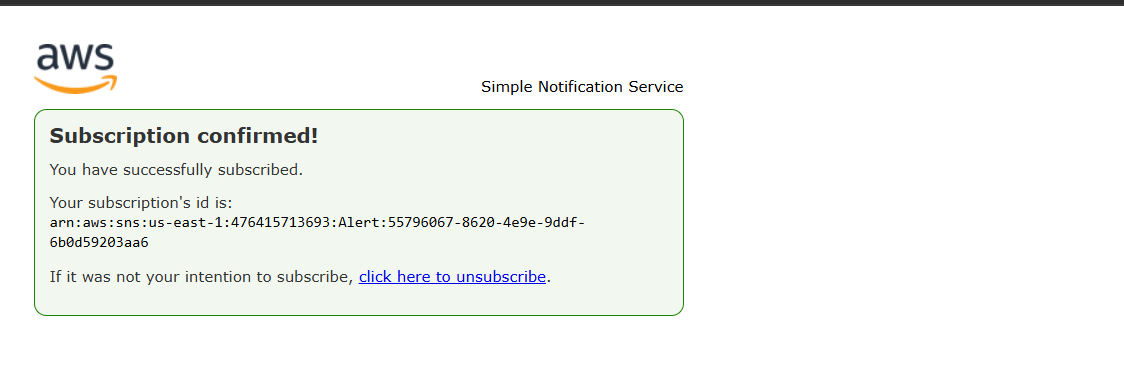
1. Enable SNS for cloudtrial to send alert on email.

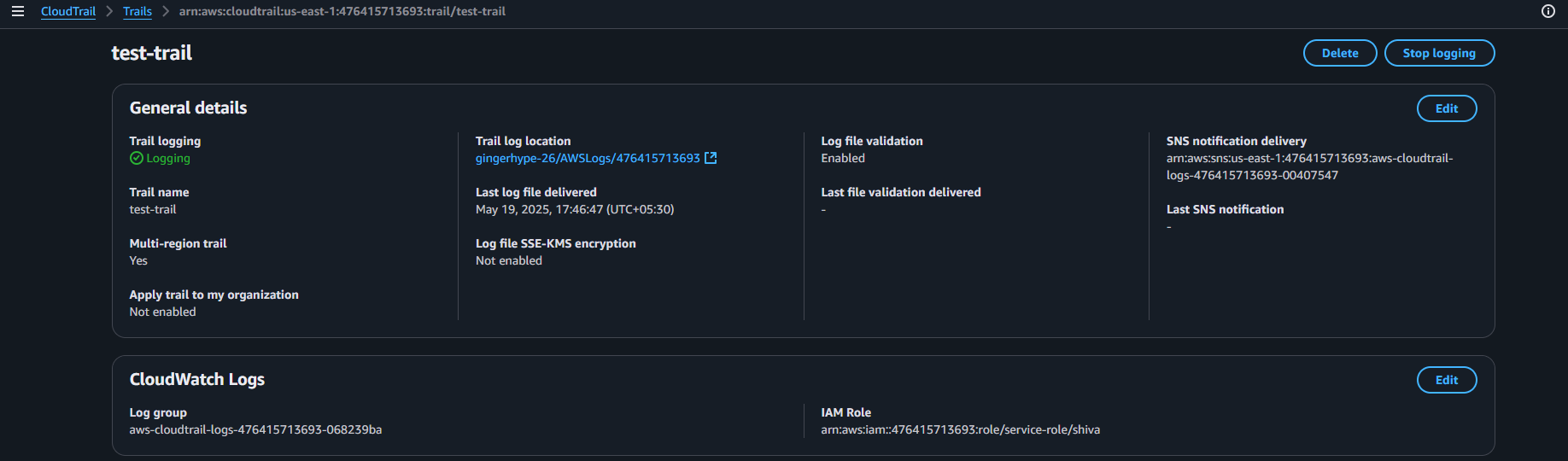
Steps:

1. Go to sns and create a topic
2. Next click on create subscription give name as alert and add to which sns has to be send (Email)
3. Go to mail and click on subscription
4. Next go to cloudtrail and to my trail click on edit and enable sns



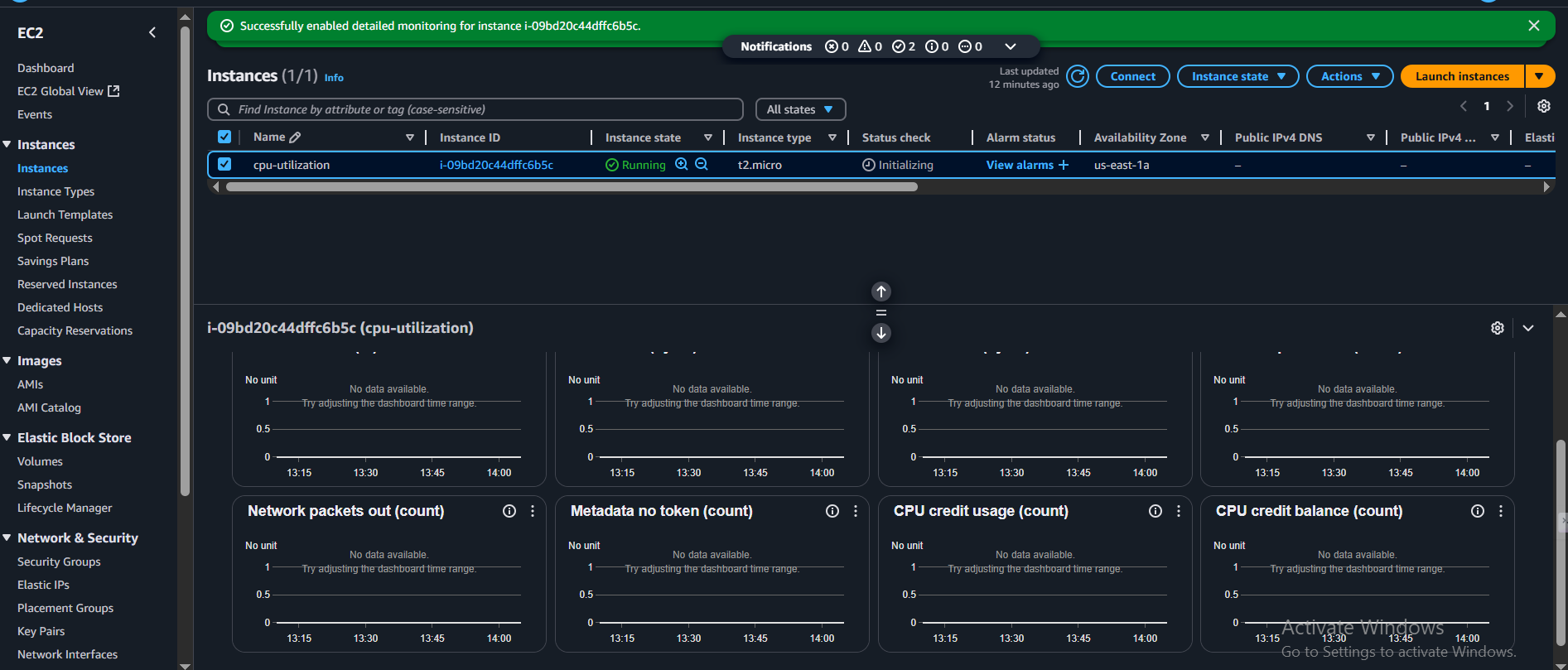




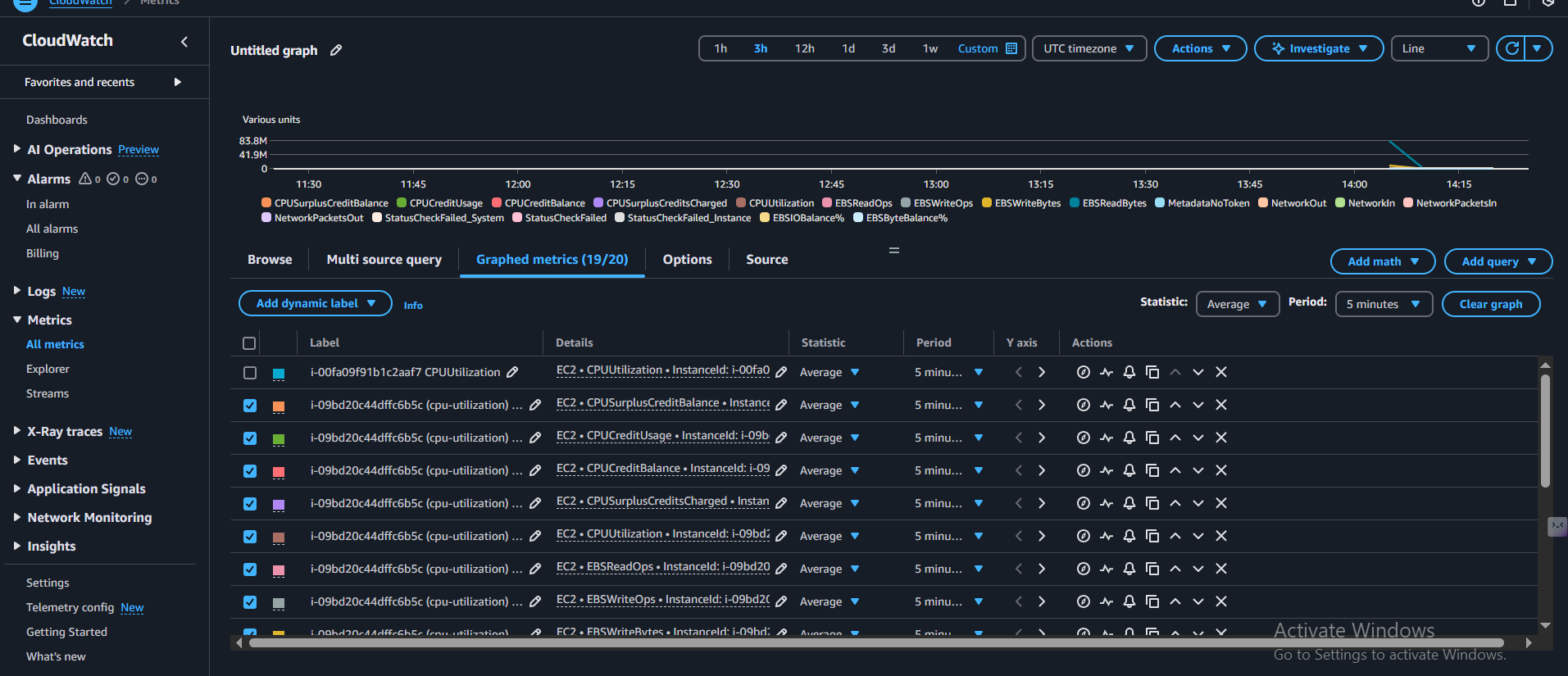


1. Configure cloud watch monitoring and record the cpu utilization and other metrics of ec2.

* **lunch the instance >>create the ec2 >>go to monitoring>clck on mange detailed monitoring option enable**

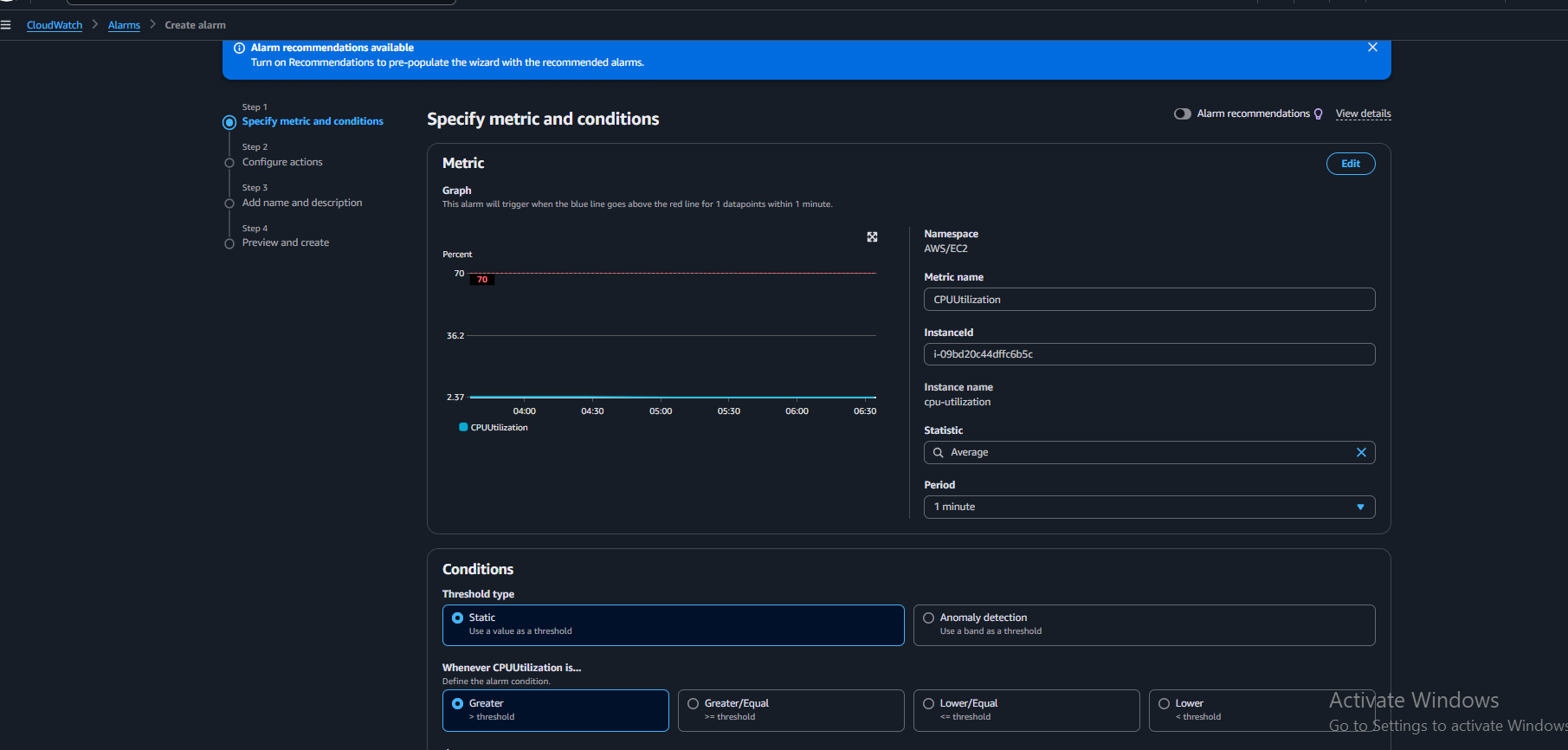


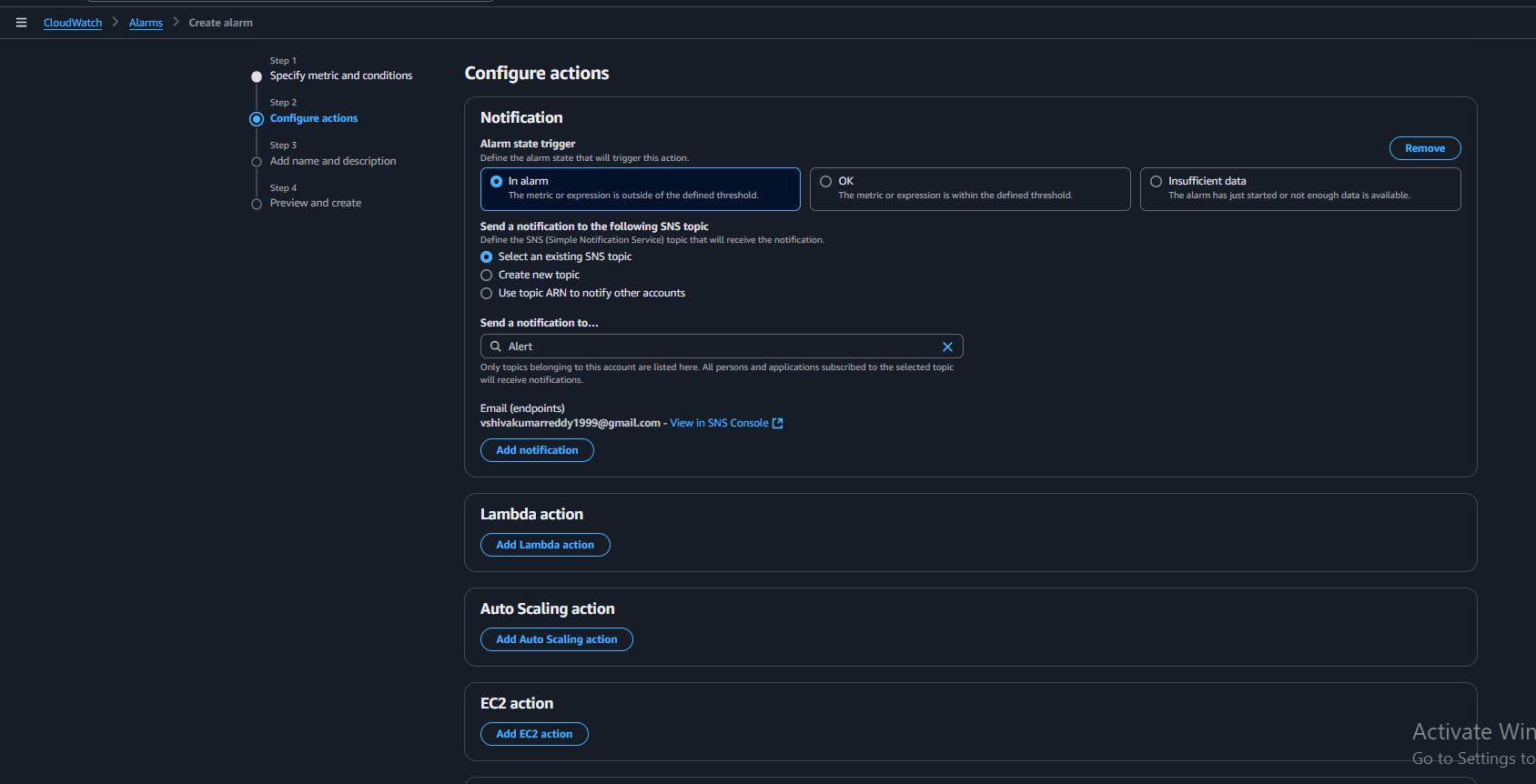
* go to the cloud watch console>>**select all metrics>>click all the metrica >>slect all>>and monitor it**

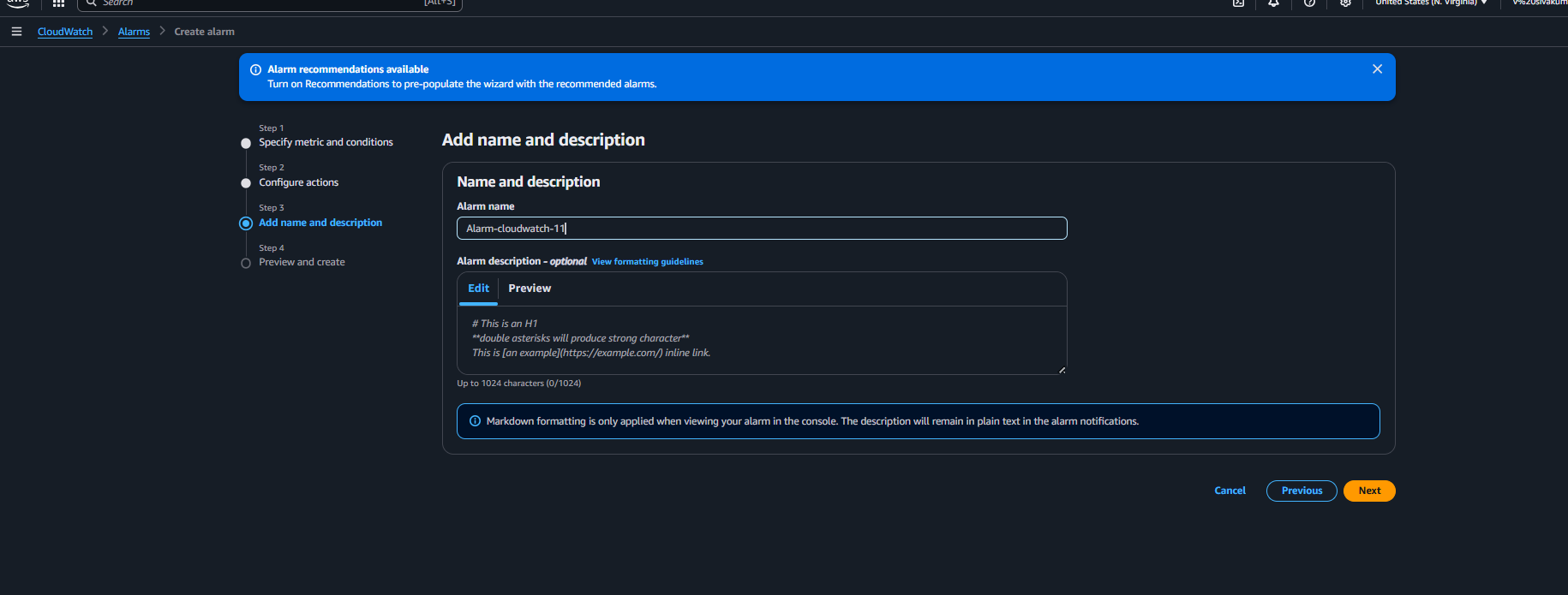


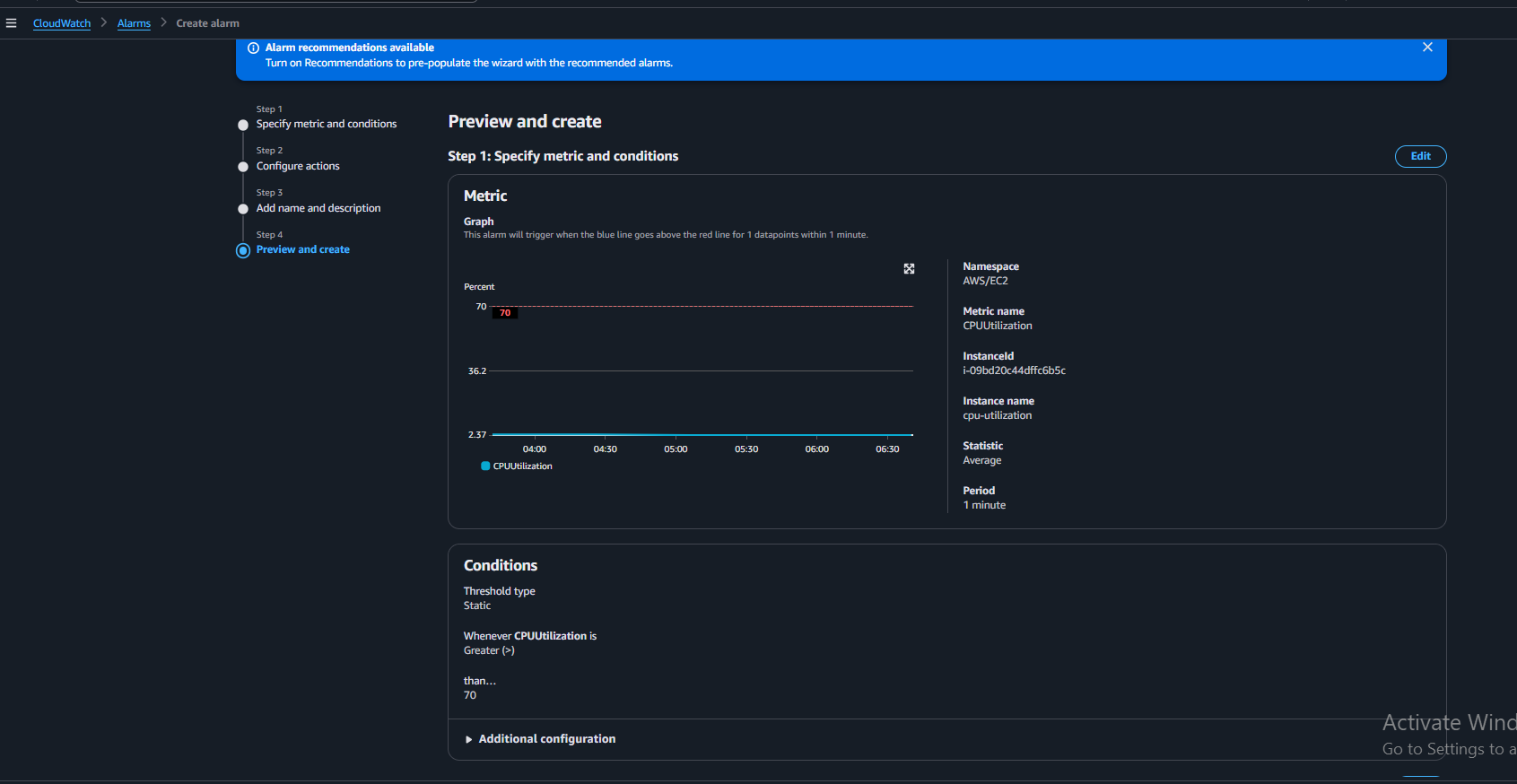
1. Create one alarm to send alert to email if the cpu utilization is more than 70 percent.

Go to the **cloud watch >>slect alarmm>>slect create alaram>>in the metrics select ec2 >>in that select cpu utilization>> slect in alarm>>select sns >>click on next>>give the cpu percentage 70>> select next>>preview>>click next >>give the alarm name>>create alarm**





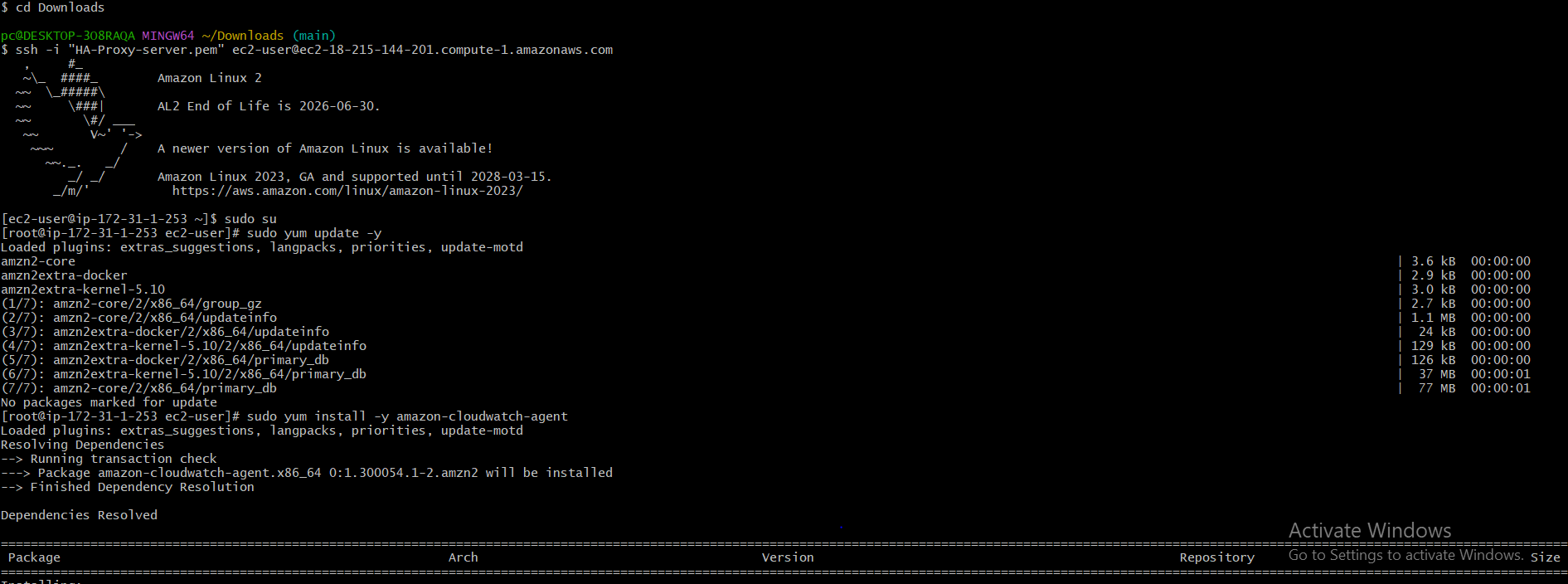




1. Create Dashboard and monitor tomcat service wether it is running or not and send the alert.

Steps:

* First create theinstance>>launch the instance CLI>>install the cloud watch agent >>sudo yum install -y amazon-cloudwatch-agent>>sudo nano /opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json>>sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl \
* -a fetch-config \
* -m ec2 \
* -c file:/opt/aws/amazon-cloudwatch-agent/etc/amazon-cloudwatch-agent.json \
* -s>>sudo /opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -m ec2 -a status
* if you see the status is running>>then the cloudwatch agent is running
* then next install the tomcat and check that the tomacat is running or not



* create abash script to attach the custom/tomcat in cloud watch
* sudo nano /usr/local/bin/monitor\_tomcat.sh
* #!/bin/bash

# Get IMDSv2 token

TOKEN=$(curl -sX PUT "http://169.254.169.254/latest/api/token" \

-H "X-aws-ec2-metadata-token-ttl-seconds: 21600")

# Get instance ID using the token

InstanceId=$(curl -s http://169.254.169.254/latest/meta-data/instance-id \

-H "X-aws-ec2-metadata-token: $TOKEN")

# If the instance ID is empty, exit with error

if [[ -z "$InstanceId" ]]; then

echo "ERROR: Instance ID not retrieved. Check metadata settings."

exit 1

fi

# Check if Tomcat is running

if ps aux | grep -v grep | grep tomcat > /dev/null; then

/usr/bin/aws cloudwatch put-metric-data \

--metric-name TomcatStatus \

--namespace "Custom/Tomcat" \

--value 1 \

--dimensions "InstanceId=$InstanceId"

else

/usr/bin/aws cloudwatch put-metric-data \

--metric-name TomcatStatus \

--namespace "Custom/Tomcat" \

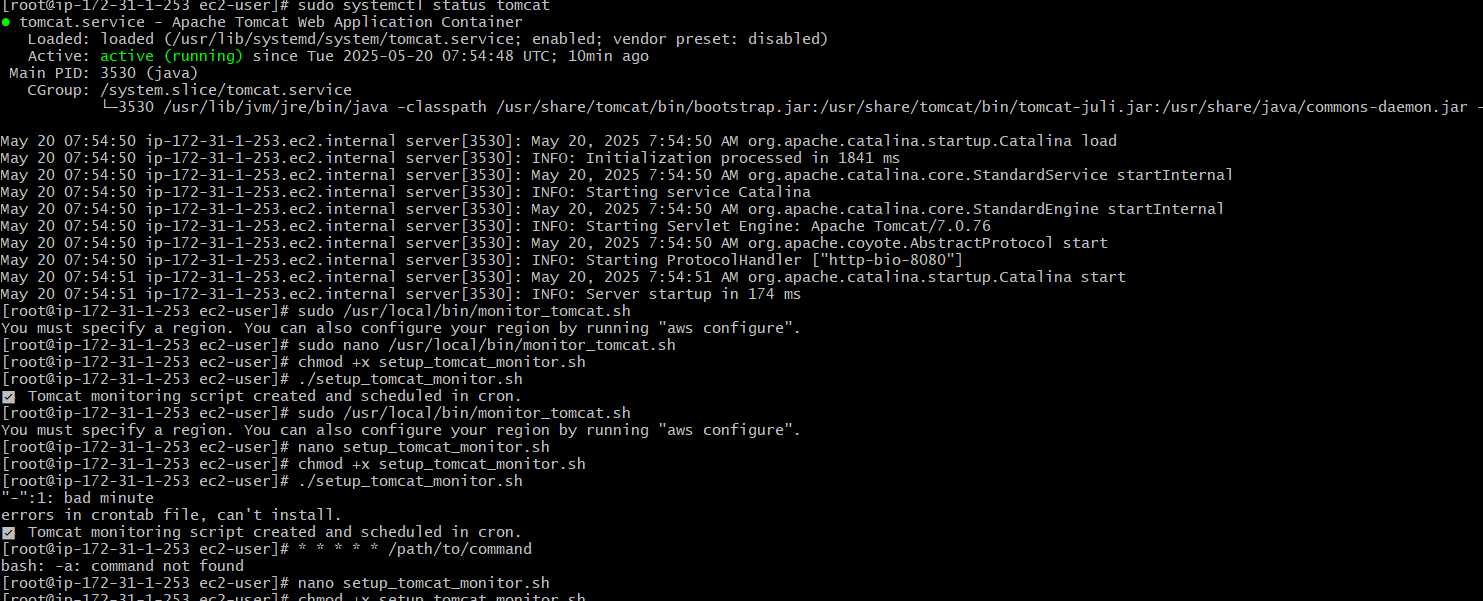
--value 0 \

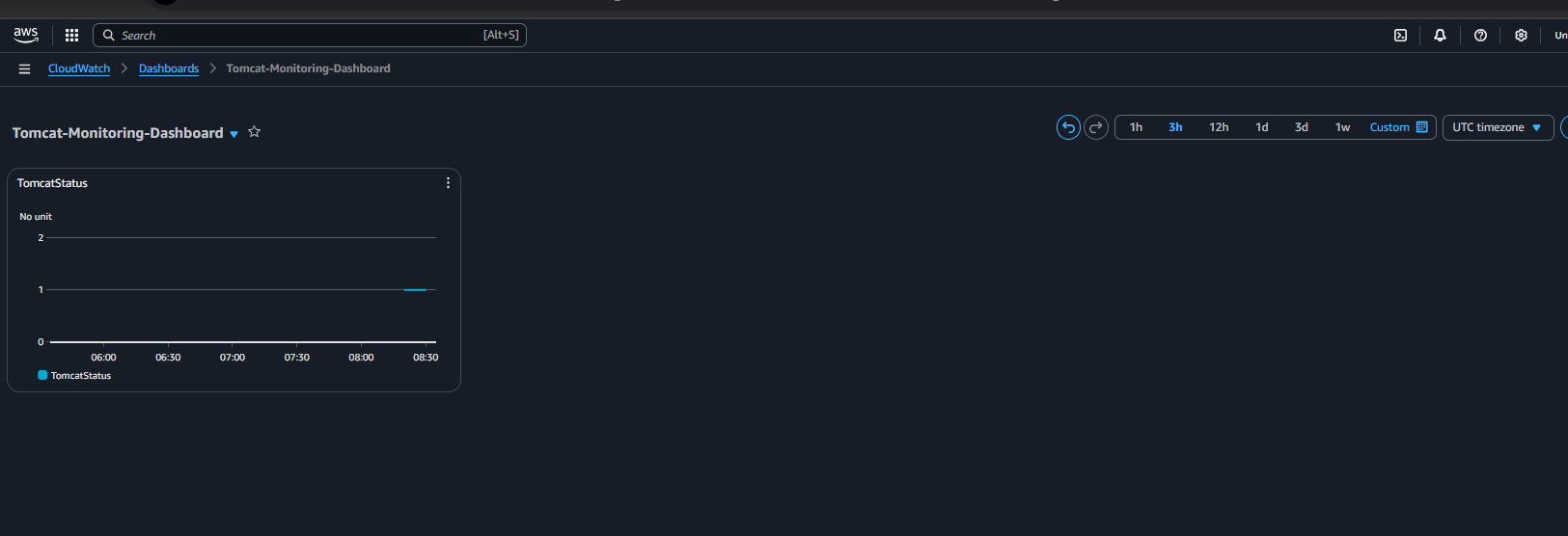
--dimensions "InstanceId=$InstanceId"

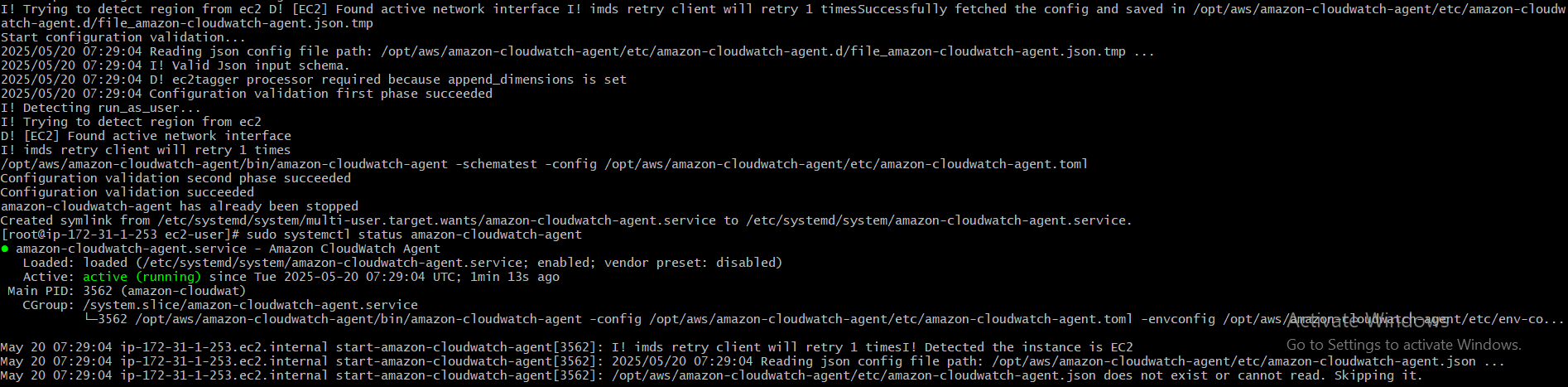
Fi

sudo chmod +x /usr/local/bin/monitor\_tomcat.sh

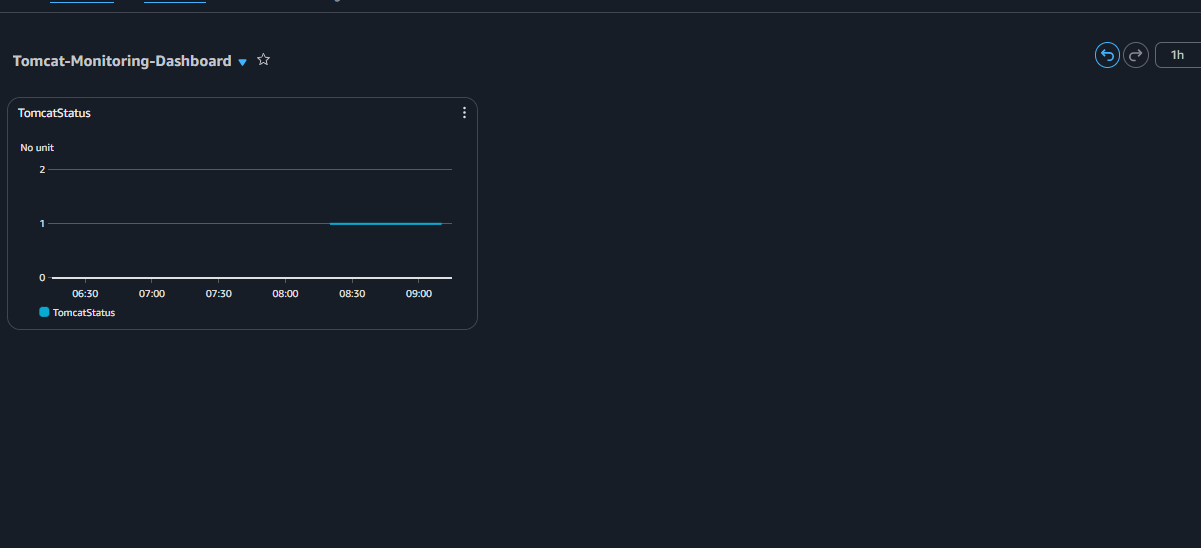
sudo /usr/local/bin/monitor\_tomcat.sh







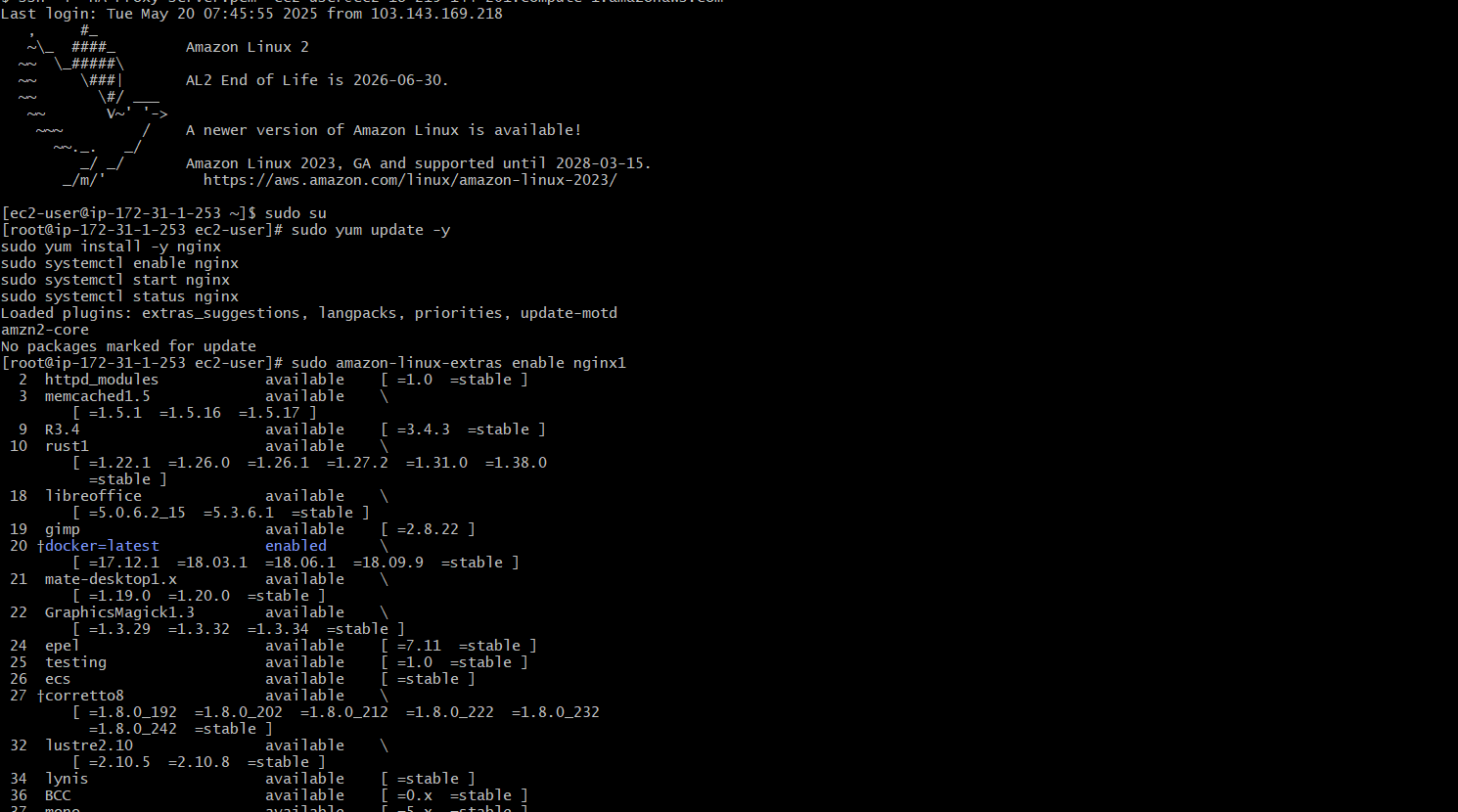
* create the cloud watch dash board and add the alaram to the dashboard

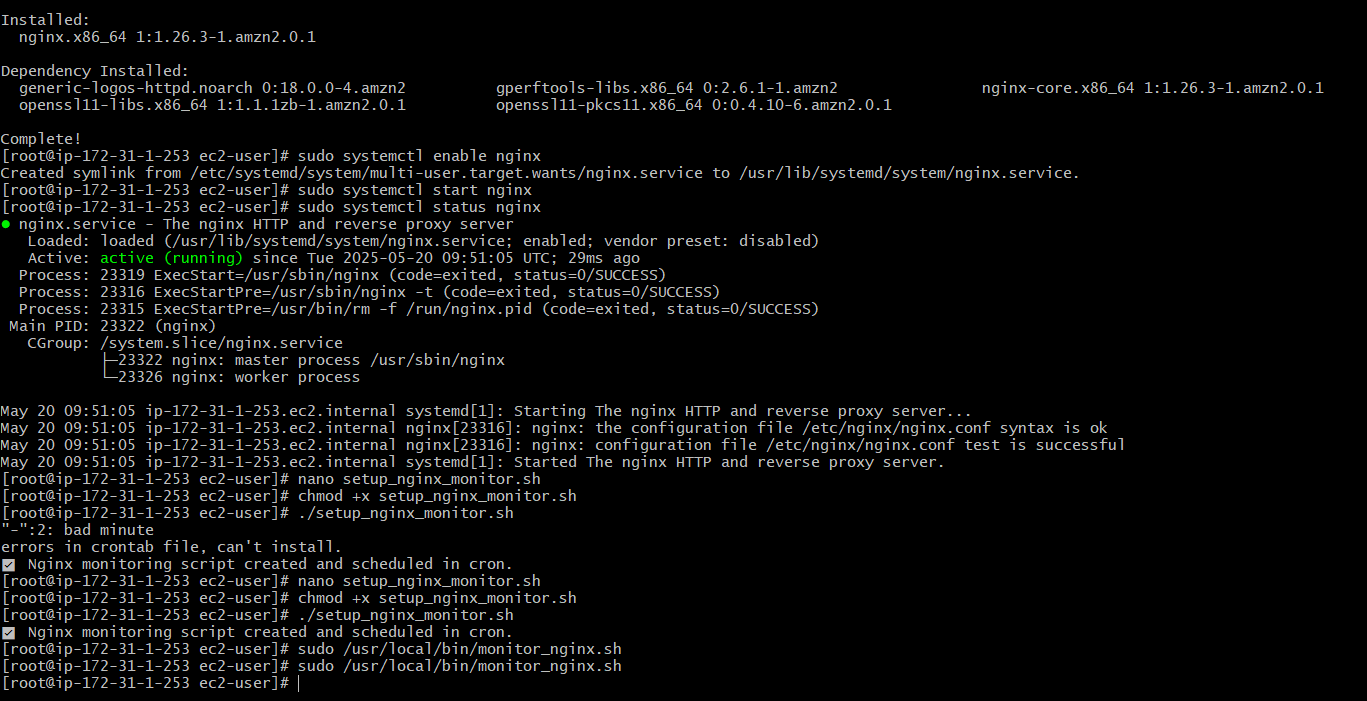


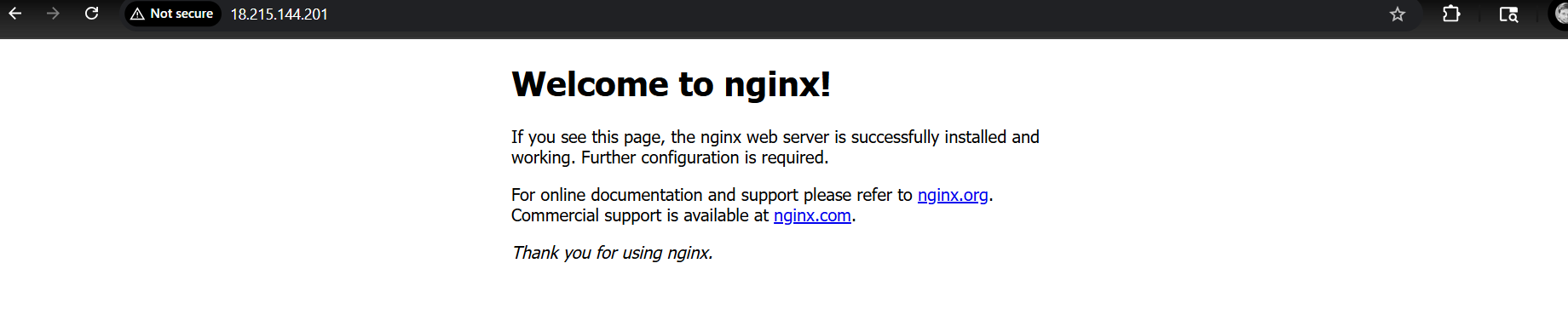
1. Create Dashboard and monitor nginx service to send the alert if nginx is not running.

Steps:

* first mainly launch the instance and create the instance and launch the instance in CLI
* now install the nginx in ec2
* sudsudo systemctl start nginx
* sudo yum update -y
* sudo systemctl enable nginx
* sudo systemctl status nginx







* then create a nano foe bash script to update that nginx in cloud watch

sudo nano /usr/local/bin/monitor\_nginx.sh

#!/bin/bash

# Get IMDSv2 token

TOKEN=$(curl -X PUT "http://169.254.169.254/latest/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600" -s)

# Get instance ID using token

INSTANCE\_ID=$(curl -H "X-aws-ec2-metadata-token: $TOKEN" -s http://169.254.169.254/latest/meta-data/instance-id)

# Check if nginx process is running

if ps aux | grep -v grep | grep nginx > /dev/null

then

# Nginx is running, send value 1

aws cloudwatch put-metric-data --metric-name NginxStatus --namespace "Custom/Nginx" --value 1 --dimensions Name=InstanceId,Value=$INSTANCE\_ID

else

# Nginx is NOT running, send value 0

aws cloudwatch put-metric-data --metric-name NginxStatus --namespace "Custom/Nginx" --value 0 --dimensions Name=InstanceId,Value=$INSTANCE\_ID

fi

* sudo chmod +x /usr/local/bin/monitor\_nginx.sh
* sudo /usr/local/bin/monitor\_nginx.sh

now you will see in cloudwatch metric >>custom/nginx >>then add this nginx to the alarm

* go to the alarm >>add the custom/nginx to alram >>set the %0 >>time shouls be 5 mins >> create the alarm
* next add this alrm to te dash board

