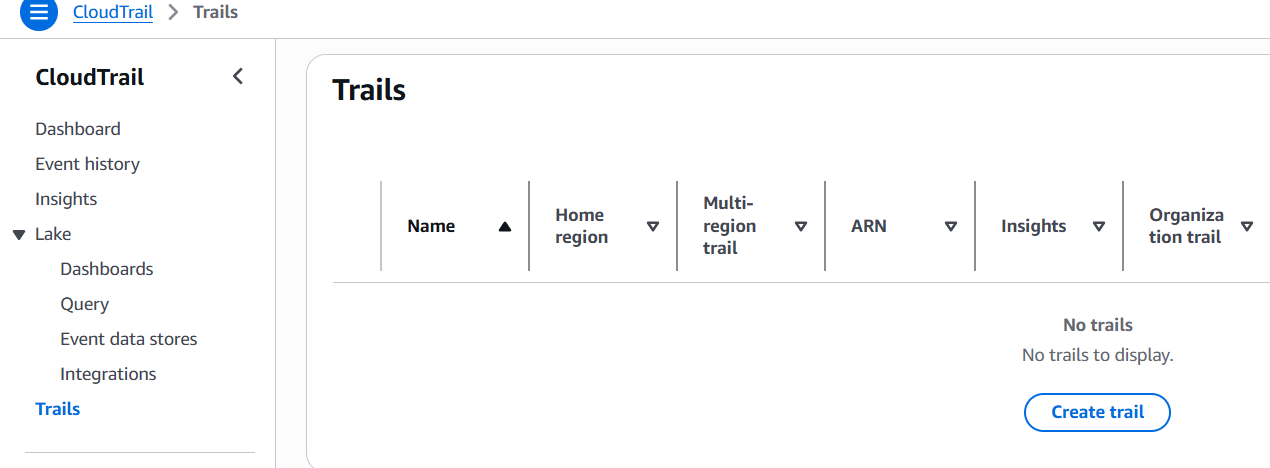
**Cloud Trial And Cloud Watch Tasks**

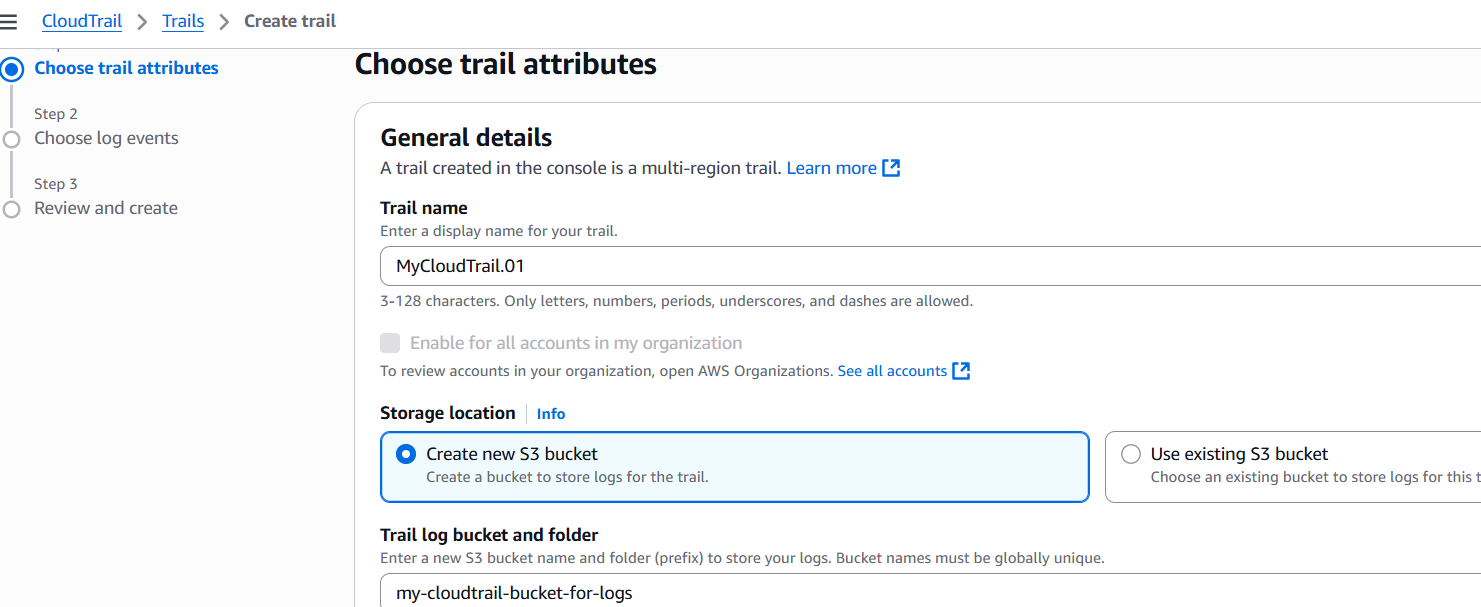
1) Enable cloudtrail monitoring and store the events in s3 and cloudwatch log events.

**CloudTrail** delivers logs to an S3 bucket for long-term storage and to CloudWatch Logs for real-time monitoring and alerting.

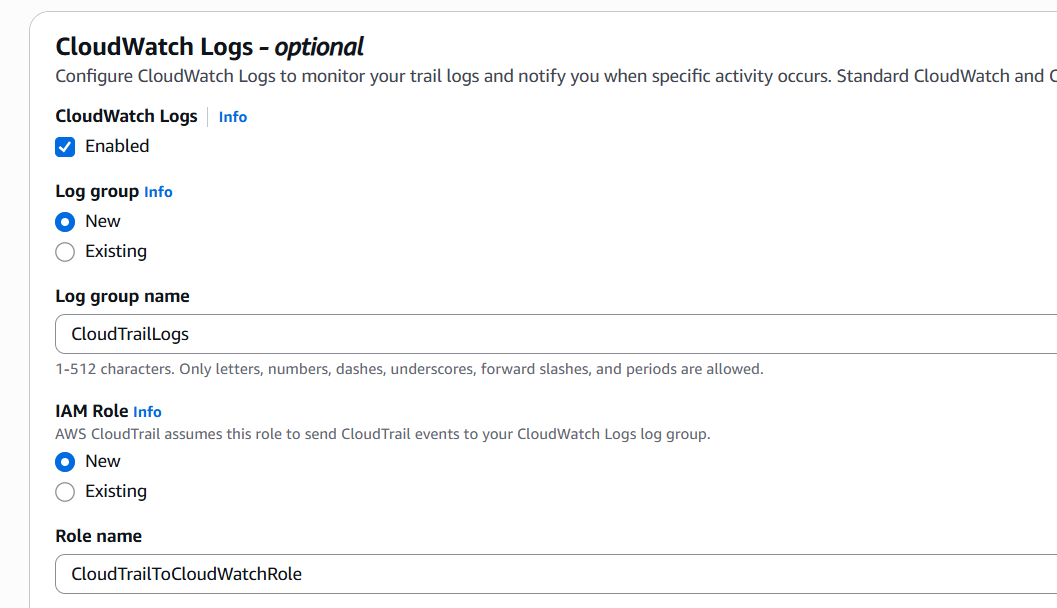
* Search for "CloudTrail". Click on "Create trail."



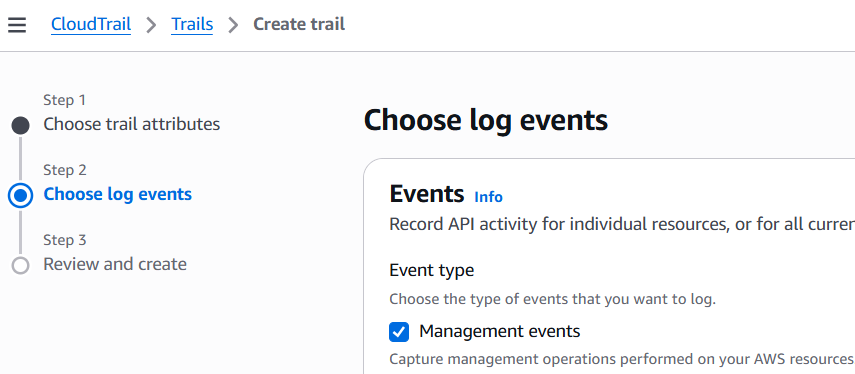
* On the Create Trail page, for Trail name, type a name for your trail.
* For Storage location, choose Create new S3 bucket to create a bucket and name it.

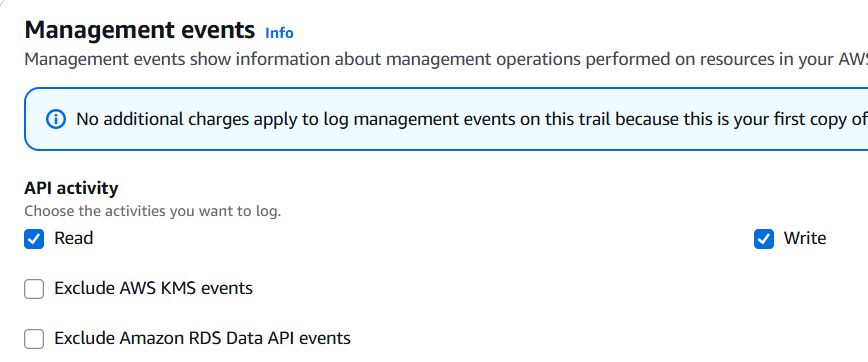


* Under "Additional settings," check "Enable log file validation"
* Under "CloudWatch Logs," check "Enabled"
* Create a new log group or select existing
* Create a new IAM role or use existing with proper permissions
* Click "Next" for event selections

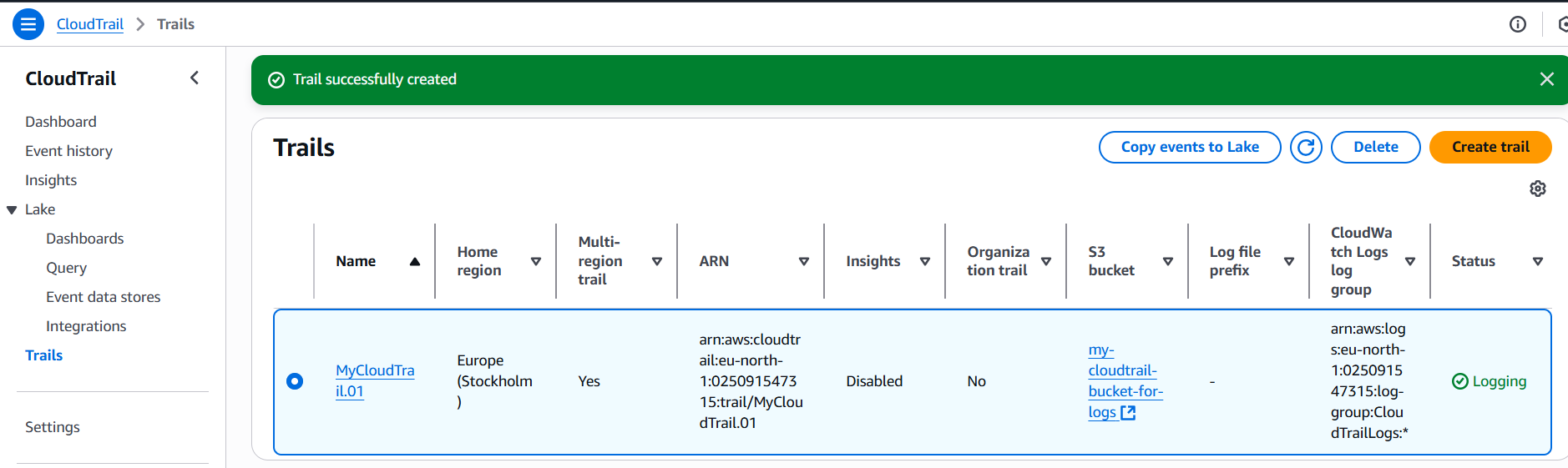


* Select "Management events" and choose "Read" and "Write"

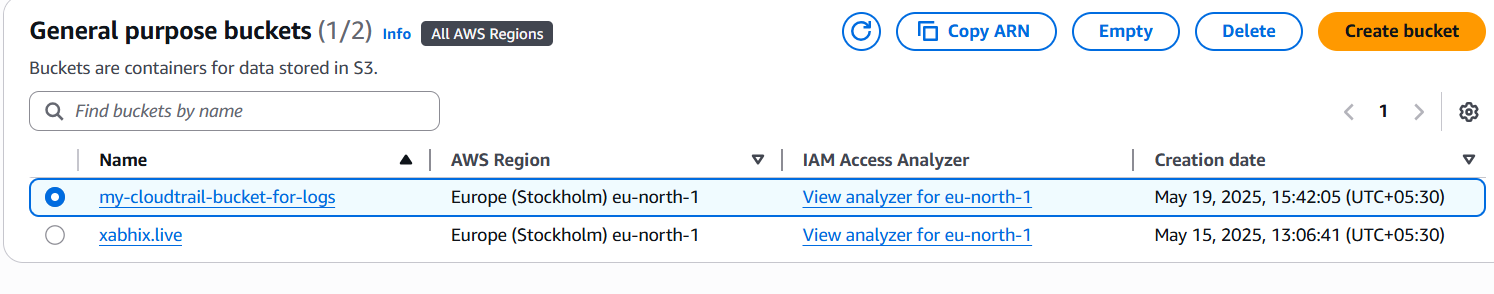




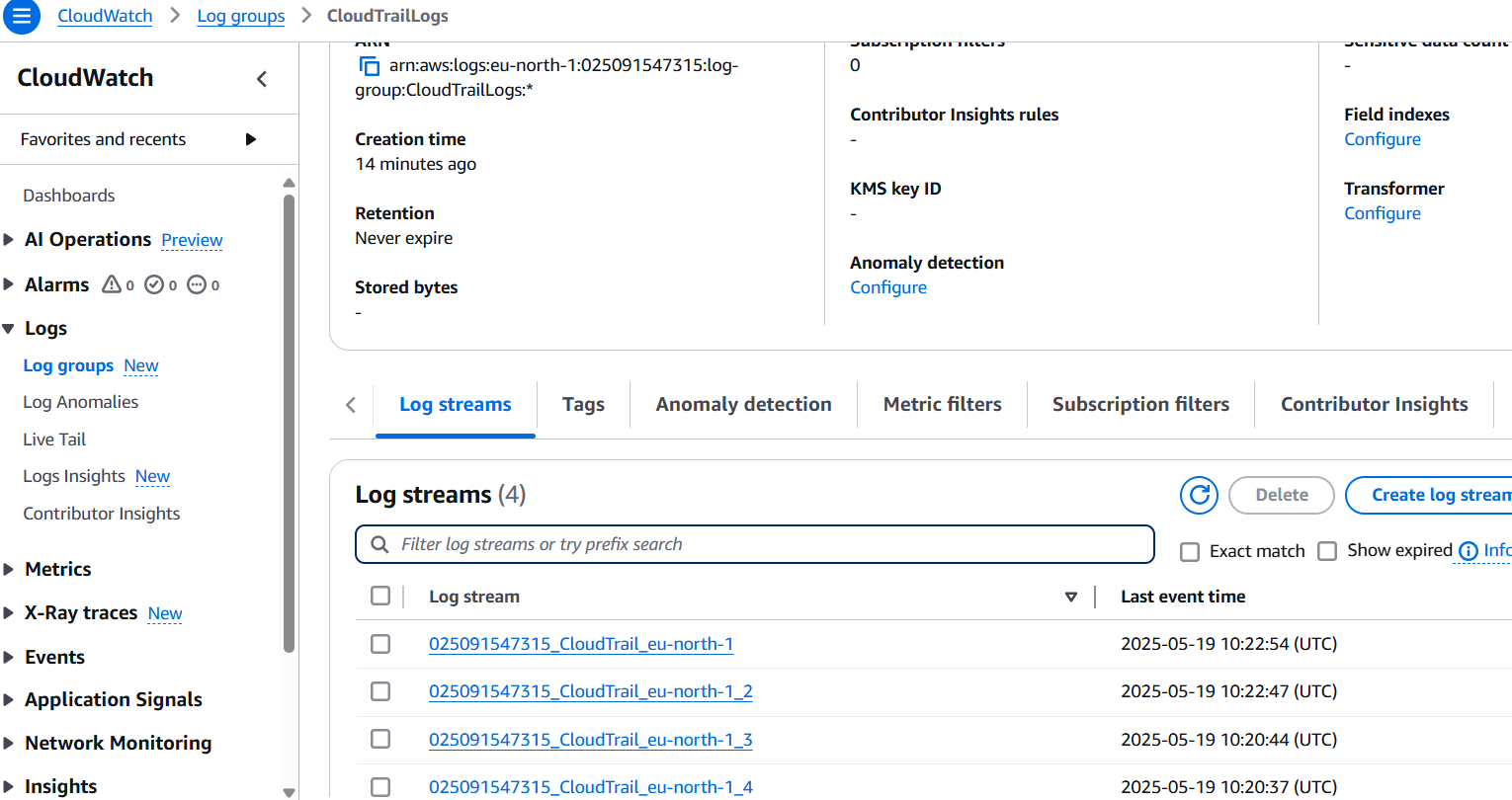
* Click "Next" and then "Create trail"



* To Verify:
  + Go to the S3 console and confirm your bucket has been created.

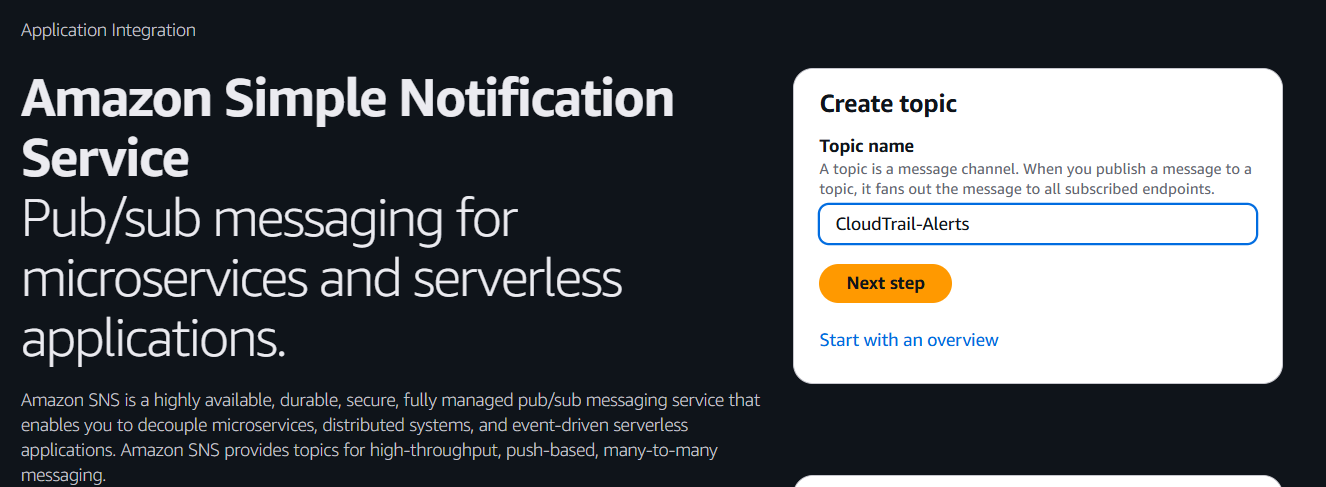


* + Go to the CloudWatch console and confirm your log group has been created.

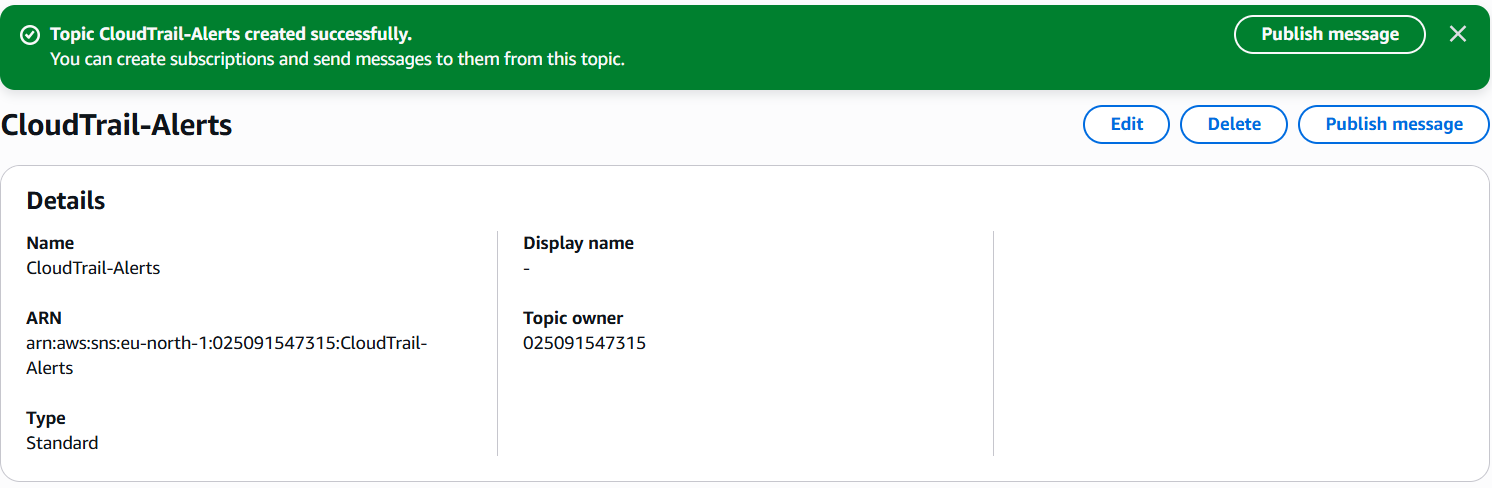


2) Enable SNS for cloudtrial to send alert on email.

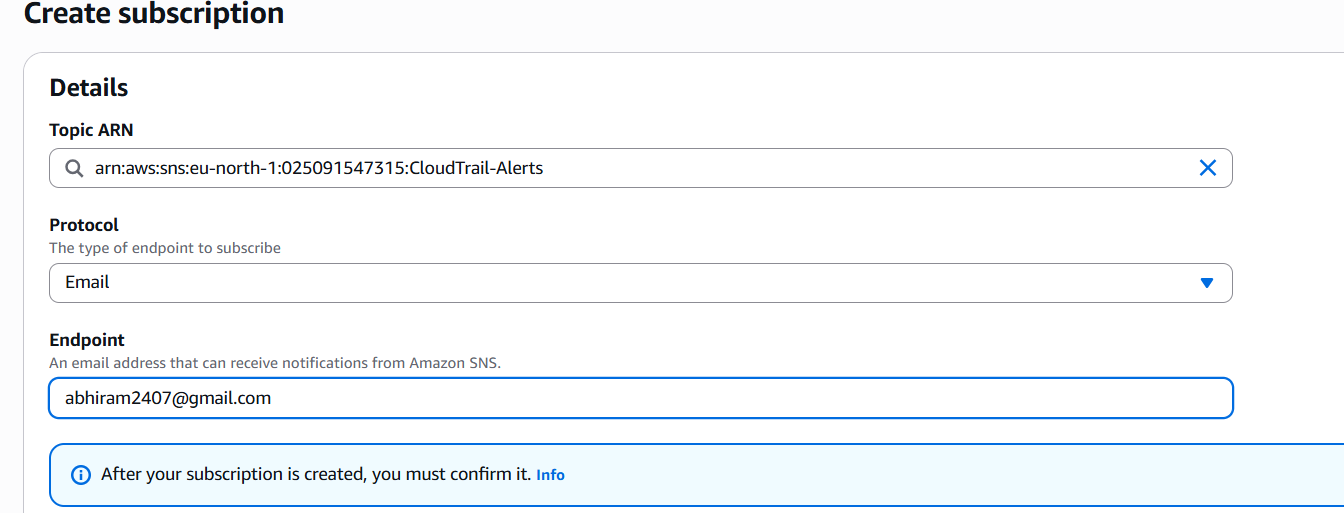
* Go to the SNS (Simple Notification Service) console.
* Click "Create topic."

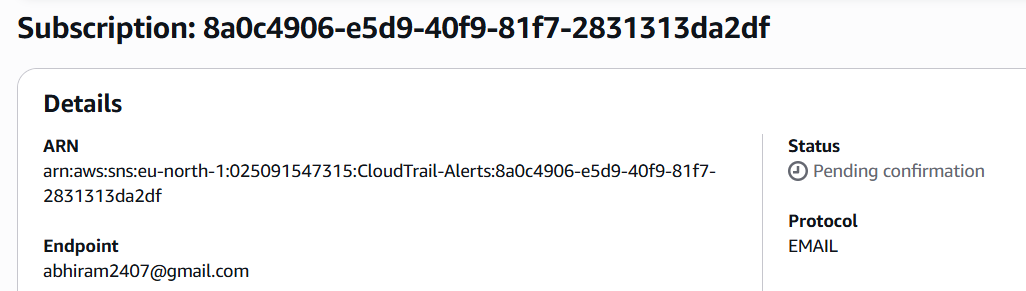


* Type: Choose "Standard". Give your topic a name
* Leave other settings as default for now. Click "Create topic."

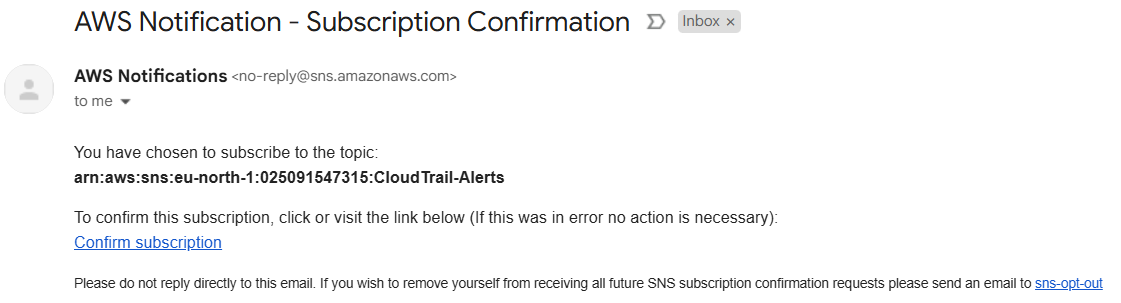


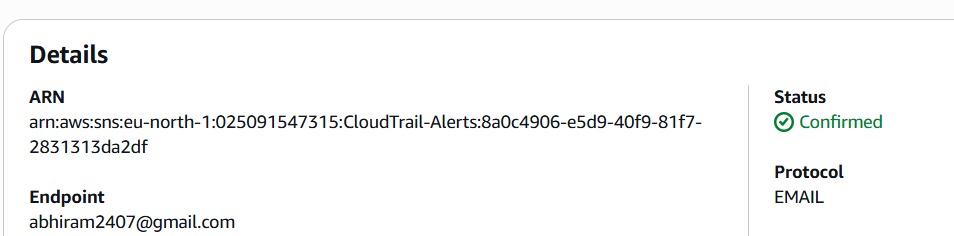
* Once the topic is created
  + Scroll down to the "Subscriptions" section and click "Create subscription".
  + Topic ARN: This should be pre-filled with the ARN of the topic you just created.
  + Protocol: Select "Email".
  + Endpoint: Enter the email address where you want to receive alerts.
  + Click "Create subscription". Currently it will be in the “Pending” status until we confirm the subscription.



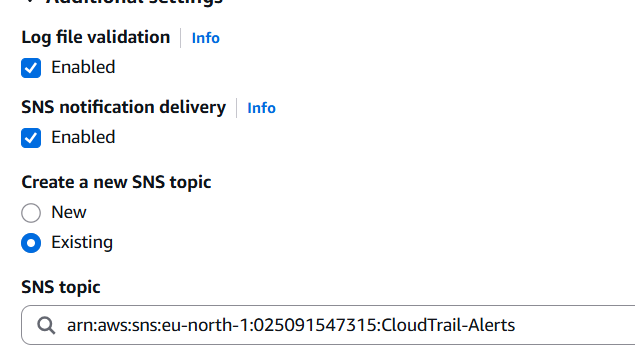


* Confirm Subscription:
  + An email will be sent to the address you provided.
  + Open the email and click the "Confirm subscription" link.
  + You should see a confirmation page in your browser.

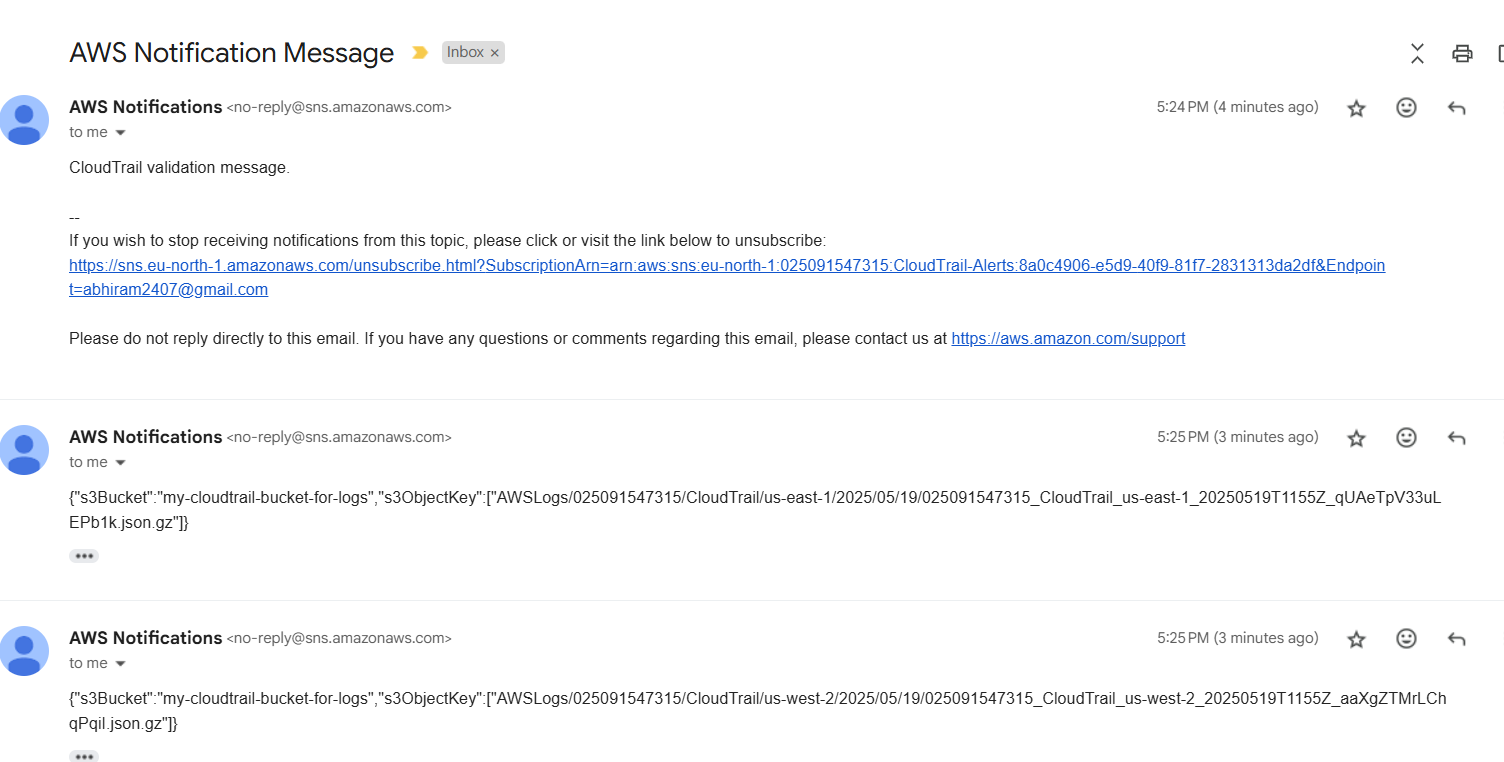




* We can edit the cloudtrail settings and check the “SNS notification delivery” and select our SNS topic to start getting alters on our email.

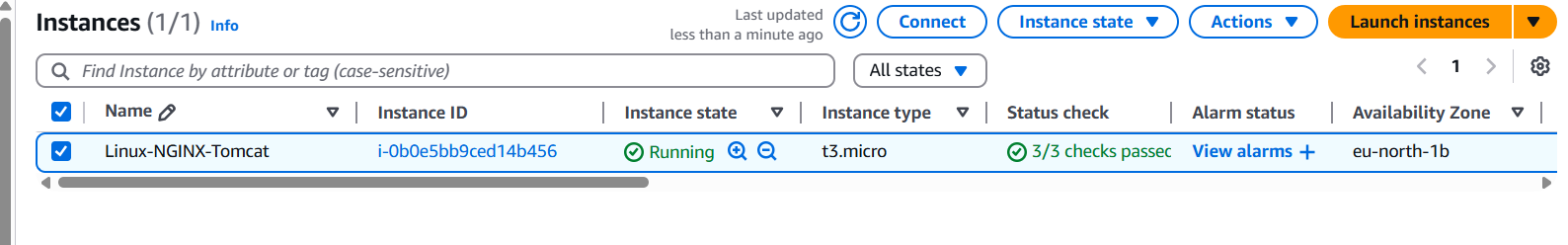


* We will now receive an SNS notification every time a new log file is added to your S3 bucket (not about what’s *in* the logs, just that a log file arrived).

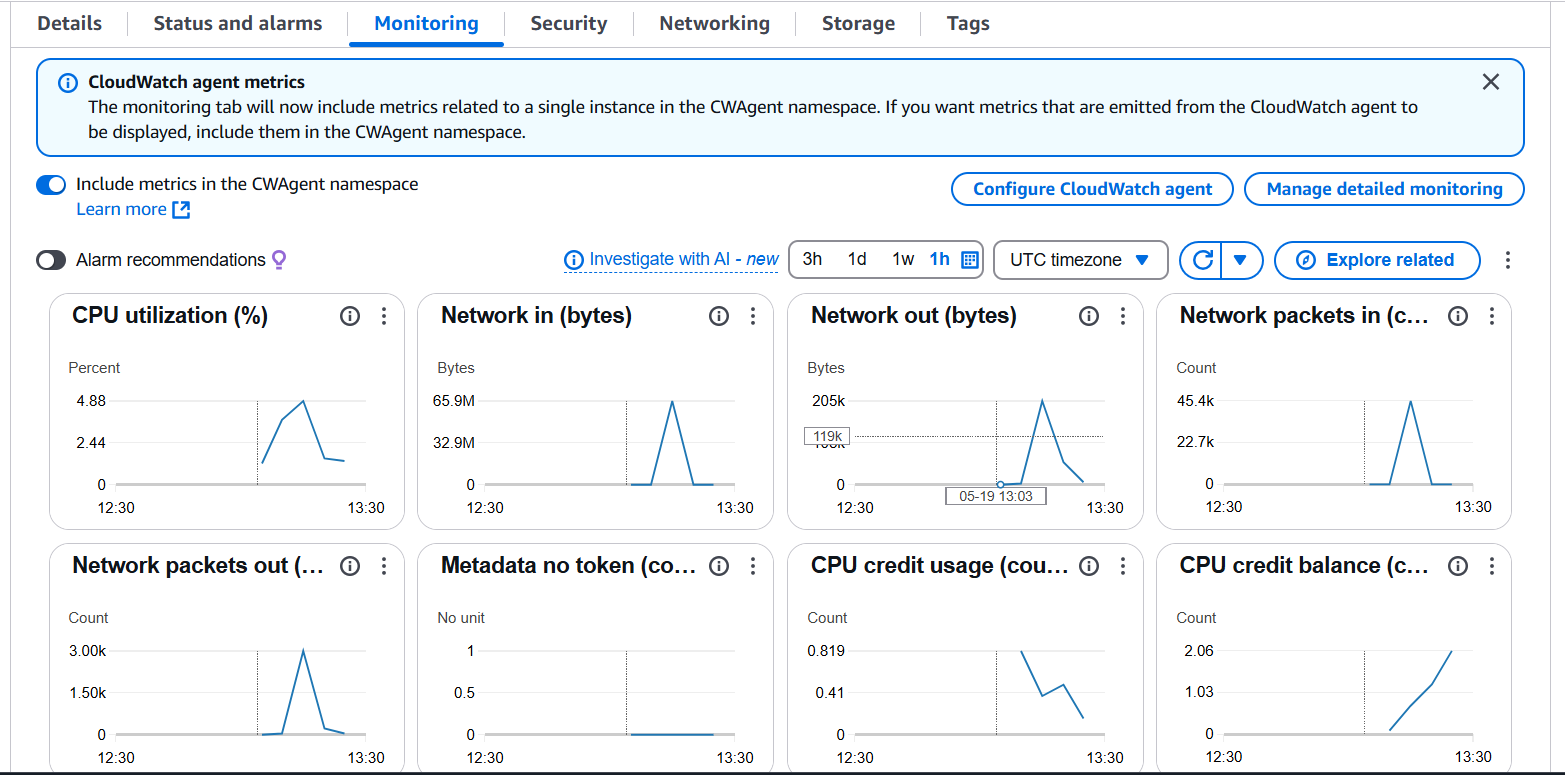


3) Configure cloud watch monitoring and record the cpu utilization and other metrics of ec2.

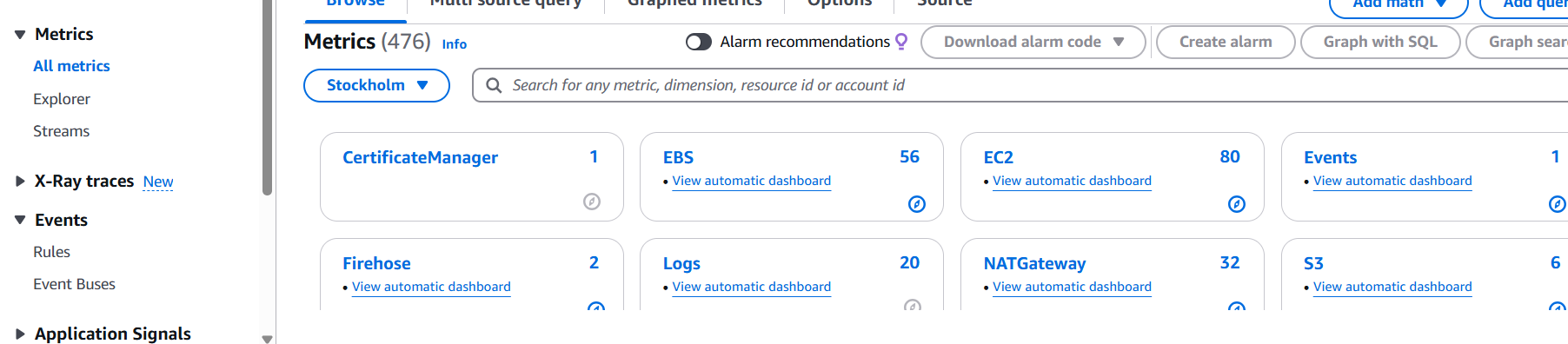
* Select a running instance



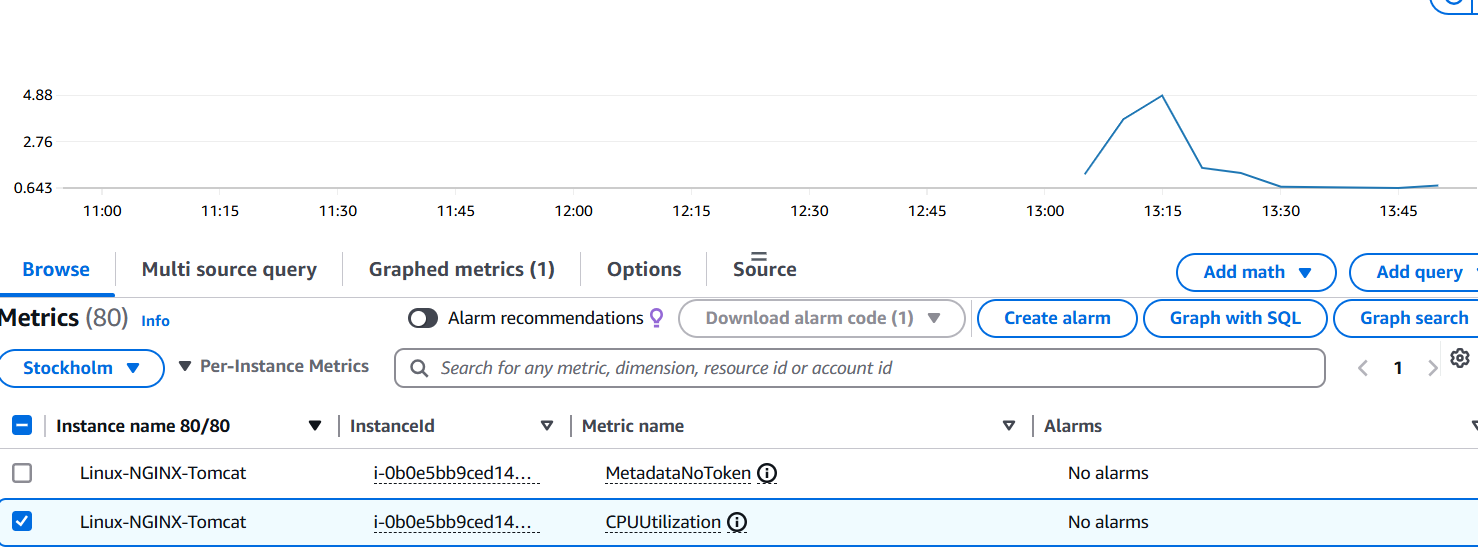
* Go to the "Monitoring" tab
* CloudWatch basic monitoring is enabled by default



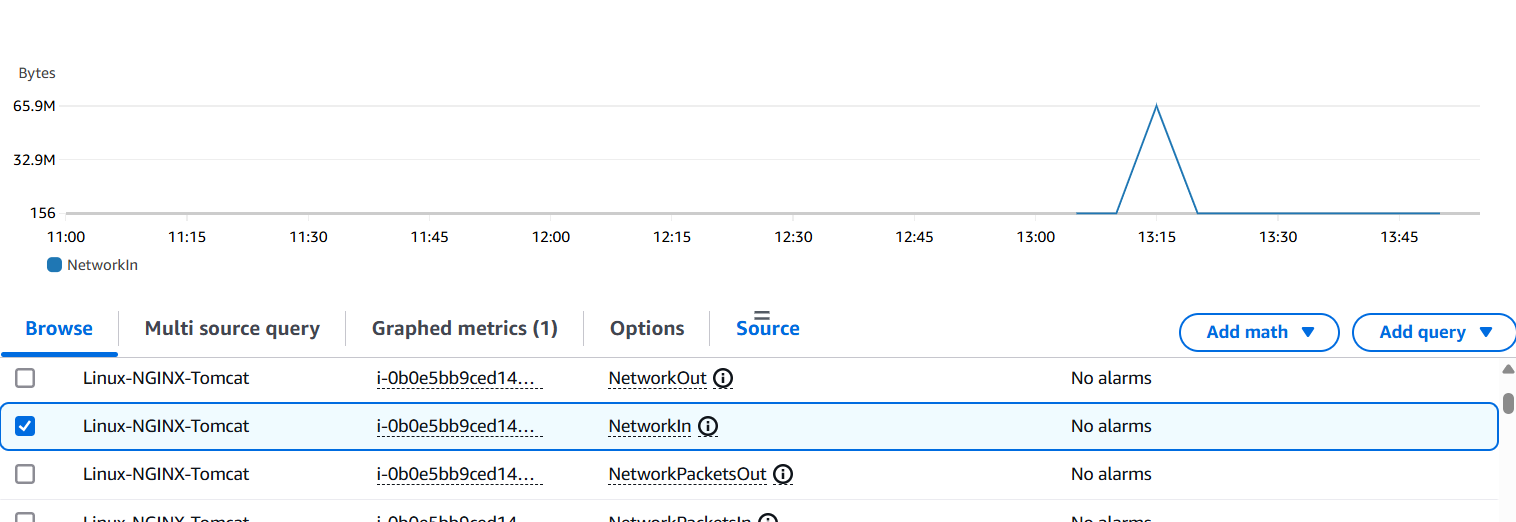
* Go to the CloudWatch console.
* Click "Metrics" -> "All metrics." Click EC2 > Per Instance Metrics.



* Choose metrics like:
  + CPUUtilization-

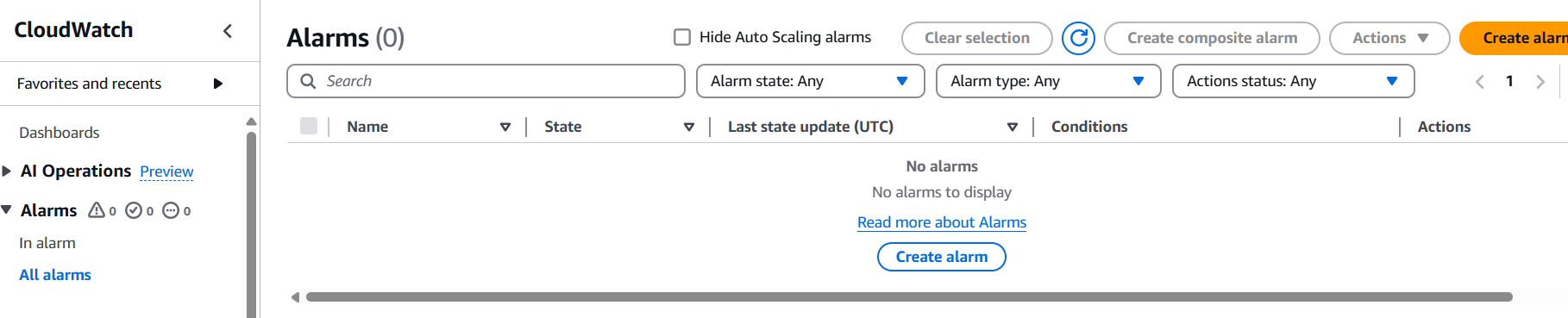


* + NetworkIn-

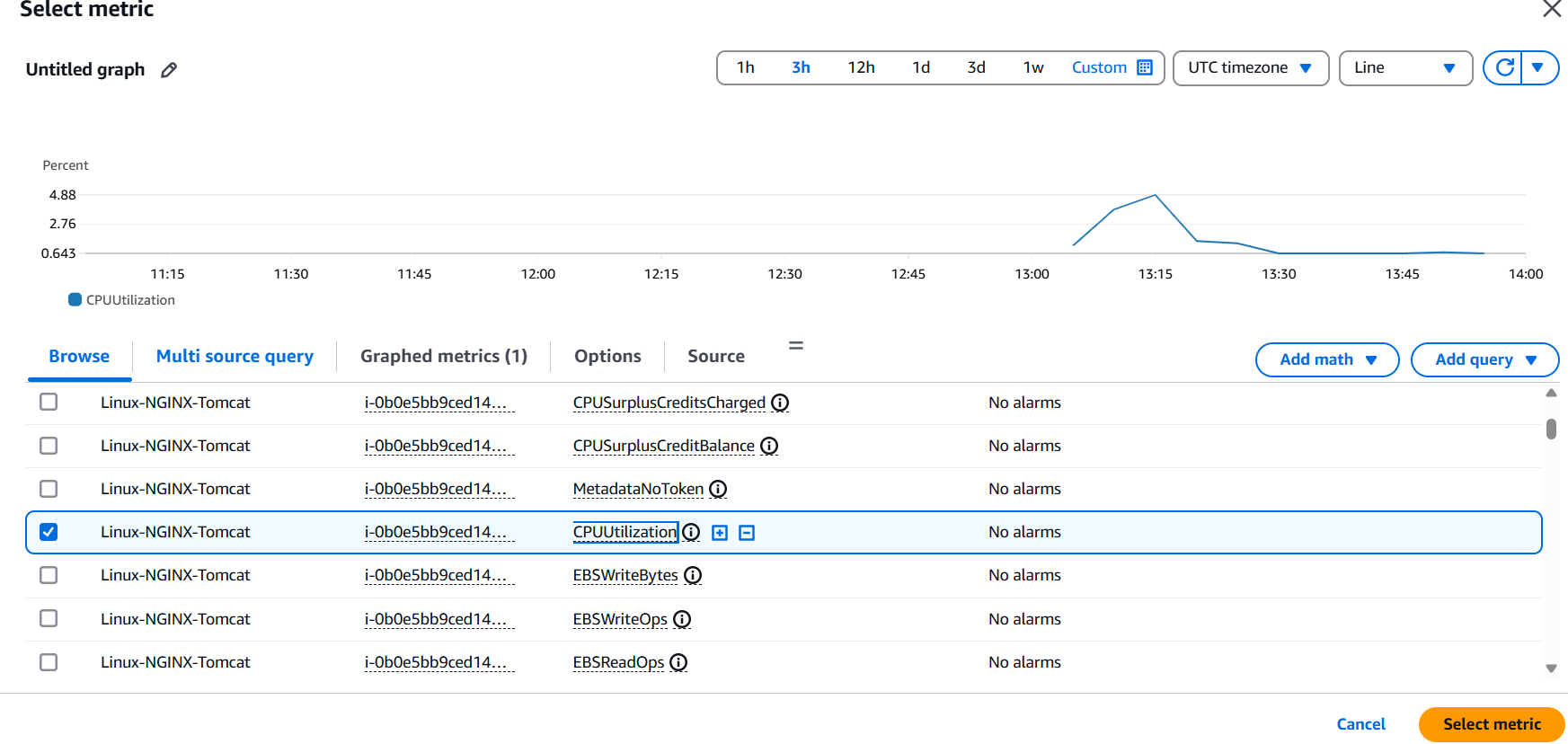


4) Create one alarm to send alert to email if the cpu utilization is more than 70 percent.

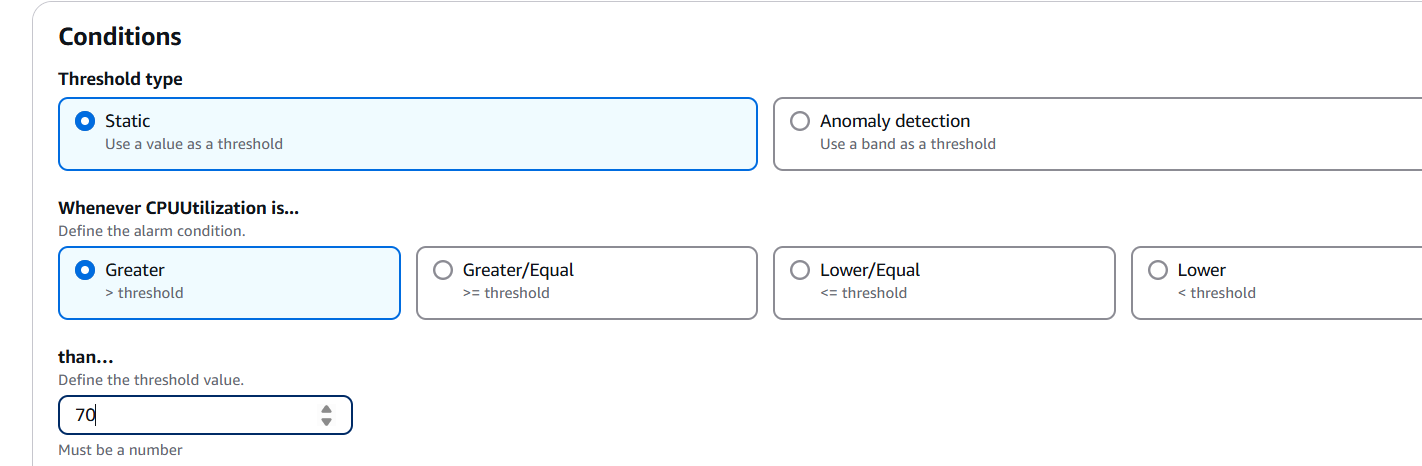
* In the CloudWatch dashboard, click on "Alarms" in the left navigation pane. Then “All Alarms”. Click the "Create alarm" button.



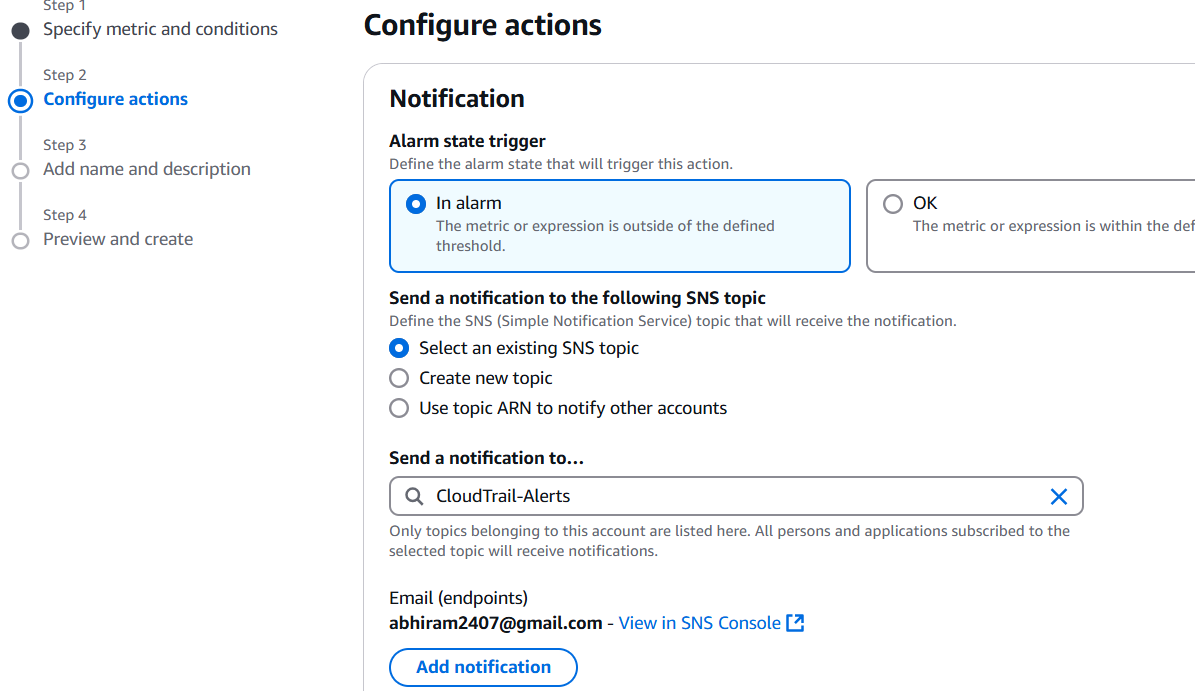
* Click "Select metric"
* Choose "EC2" > "Per-Instance Metrics"
* Find your instance and select "CPUUtilization"
* Click "Select metric"



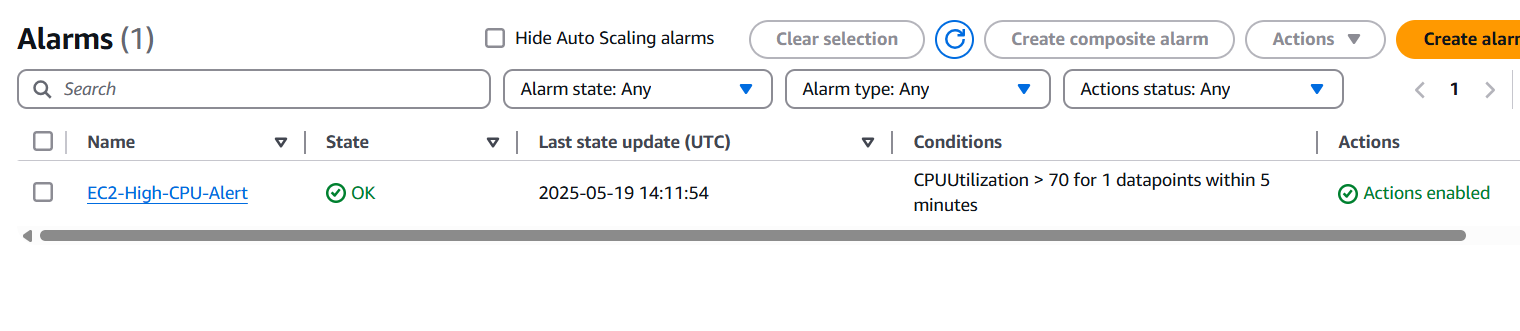
* Set the conditions:
  + Choose "Static"
  + Select "Greater than or equal to" threshold
  + Enter "70" for the threshold value and then Click "Next"



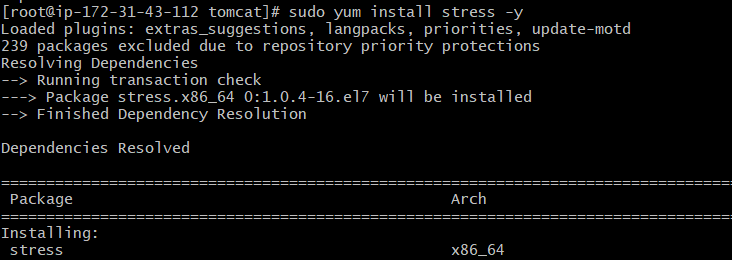
* Configure Actions:
  + Click "Next".
  + Alarm state trigger: Select "In alarm".
  + Select an SNS topic: Choose "Select an existing SNS topic".
  + Send a notification to: Select the SNS topic already created (Task 2).



* Name the alarm and create it. Confirm your email subscription

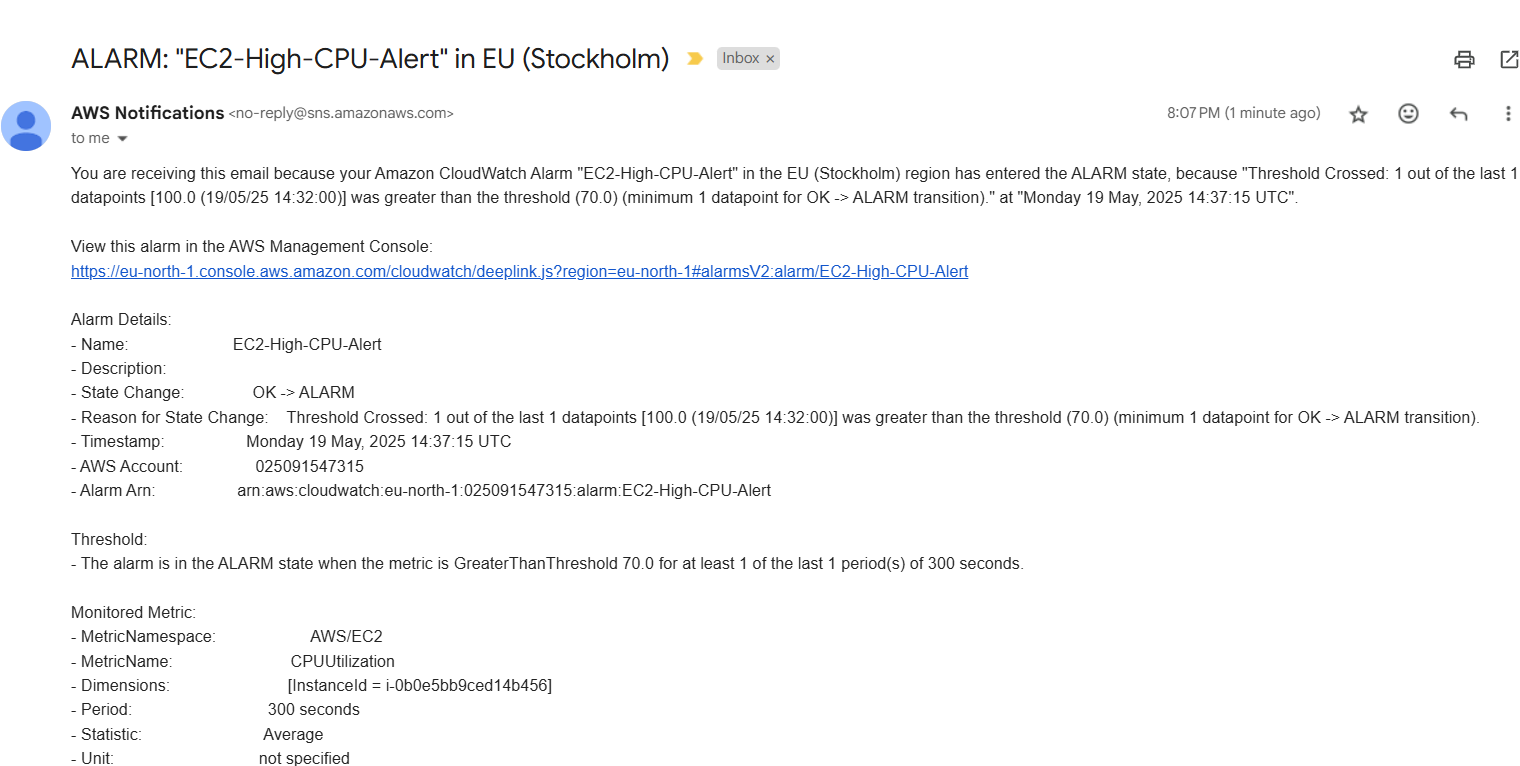


* To verify you'll need to simulate high CPU utilization on your EC2 instance. You can use tools like stress (on Linux).





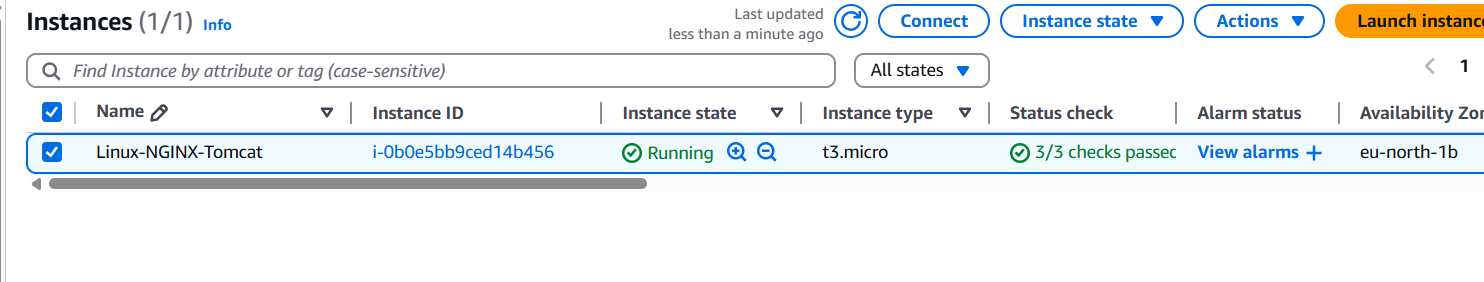
* After sometime , you should receive an email alert.

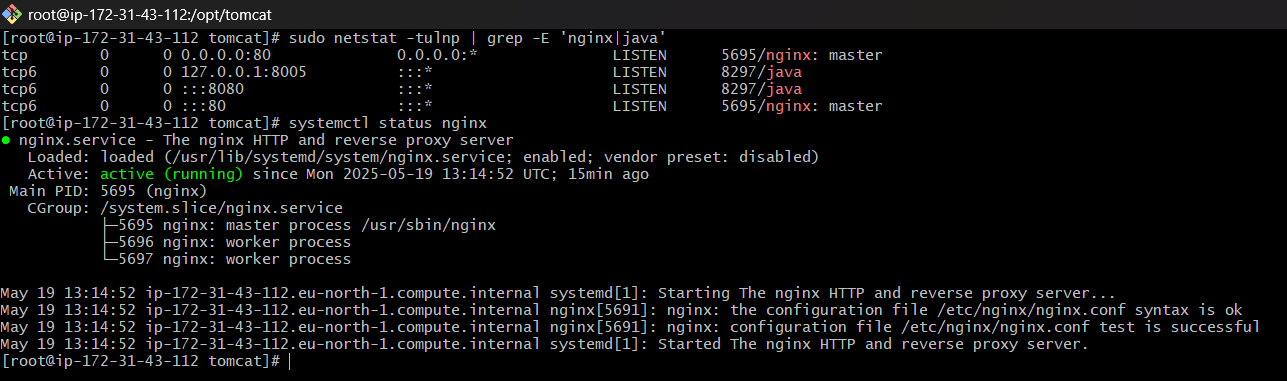


5) Create Dashboard and monitor tomcat service wether it is running or not and send the alert.

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

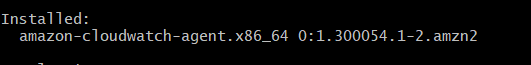
* First have Tomcat and NGIX installed in your EC2 instance.



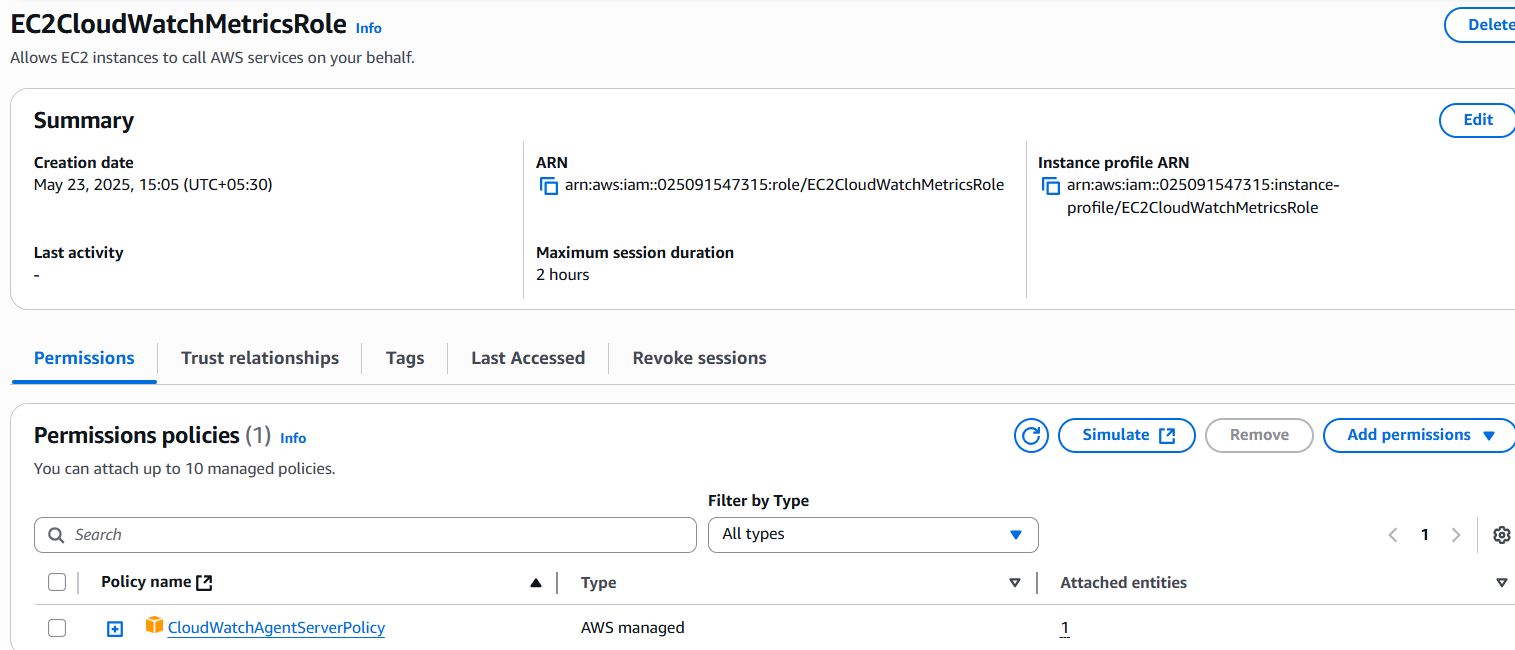


* Install Amazon Cloudwatch Agent

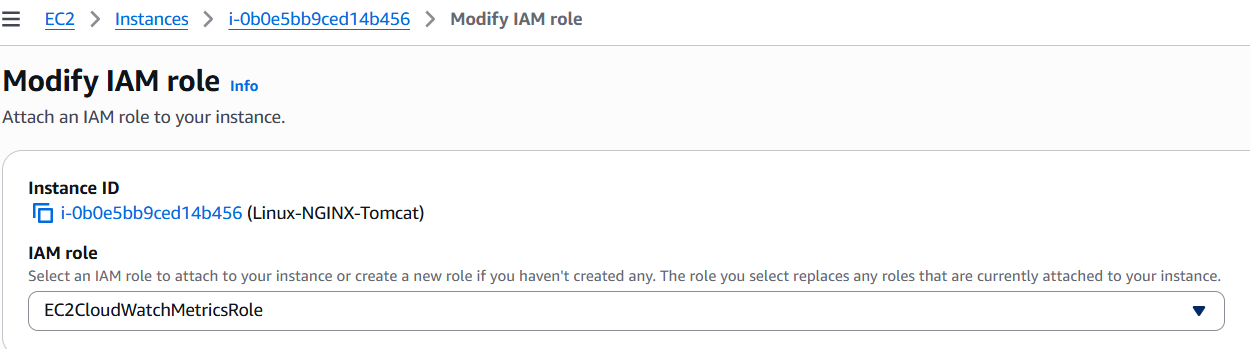




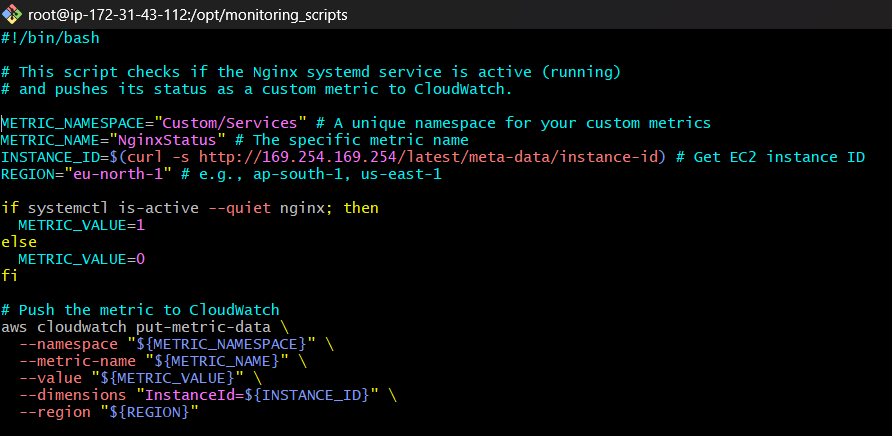
* Ensure the AWS CLI is installed on your EC2 instance.
* Create an IAM Role
  + Go to the AWS Console -> IAM -> Roles.
  + Click "Create role".
  + For "Use case", choose "EC2" and click "Next".
  + In the "Add permissions" section, search for CloudWatchAgentServerPolicy. This managed policy typically includes cloudwatch:PutMetricData and other necessary permissions for sending metrics to CloudWatch. Select it and create the role.



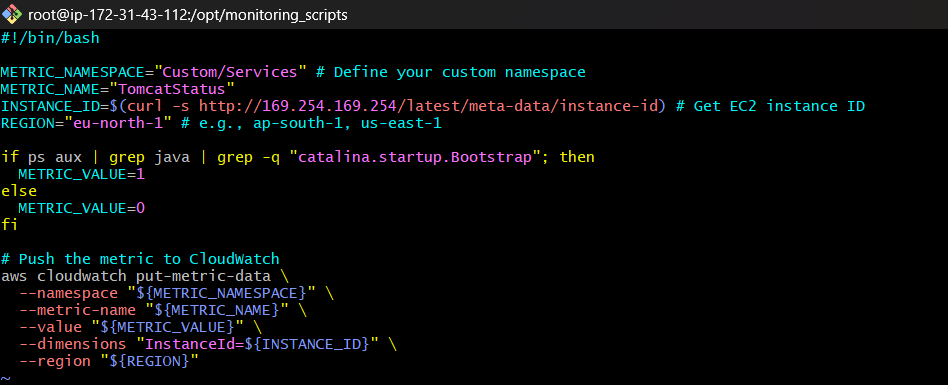
* Attach the IAM Role to Your EC2 Instance
  + Go to the AWS Console -> EC2 -> Instances.
  + Select your EC2 instance.
  + Under "Actions" -> "Security" -> "Modify IAM role".
  + From the "IAM role" dropdown, select the role you just created (e.g., EC2CloudWatchMetricsRole).
  + Click "Update IAM role".



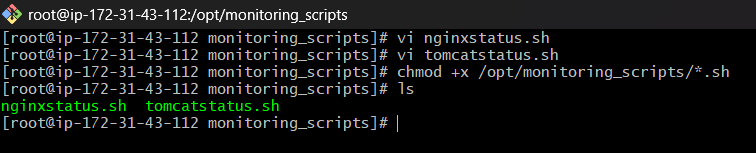
* Make scripts to check the running status of both NGINX and Tomcat and to also send custom metrics to CloudWatch, so you can build dashboards and alarms from them.
* Script for NGINX



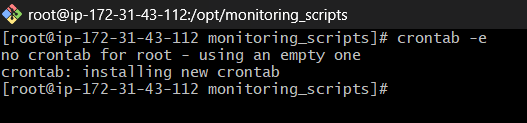
* Script for Tomcat

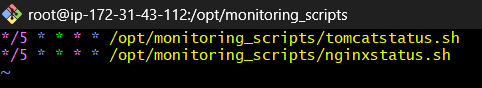


* Give them the permissions to be executable .

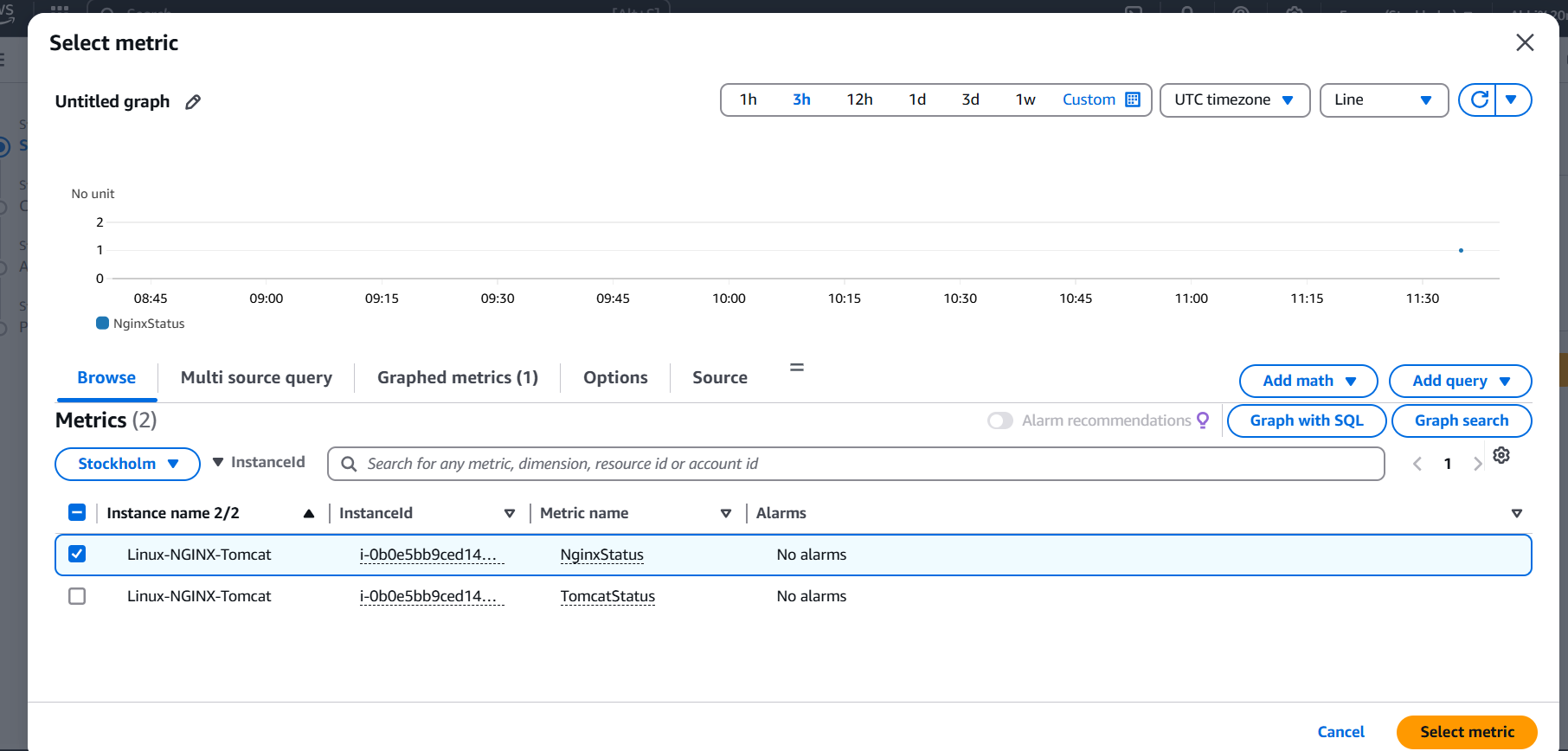


* Creating a **cronjob -**
  + Without a cron job: Your script will run once when you execute it manually, send one data point, and then stop. Your service will not be continuously monitored, and you will not receive alerts if it stops later.
  + With a cron job: The script will run every minute (or whatever interval you configure), continuously checking the service status, sending the 0 or 1 value to CloudWatch, and allowing your alarms to function correctly.
* If you were using the CloudWatch Agent's **exec** plugin, you would not need a cron job for your status script.

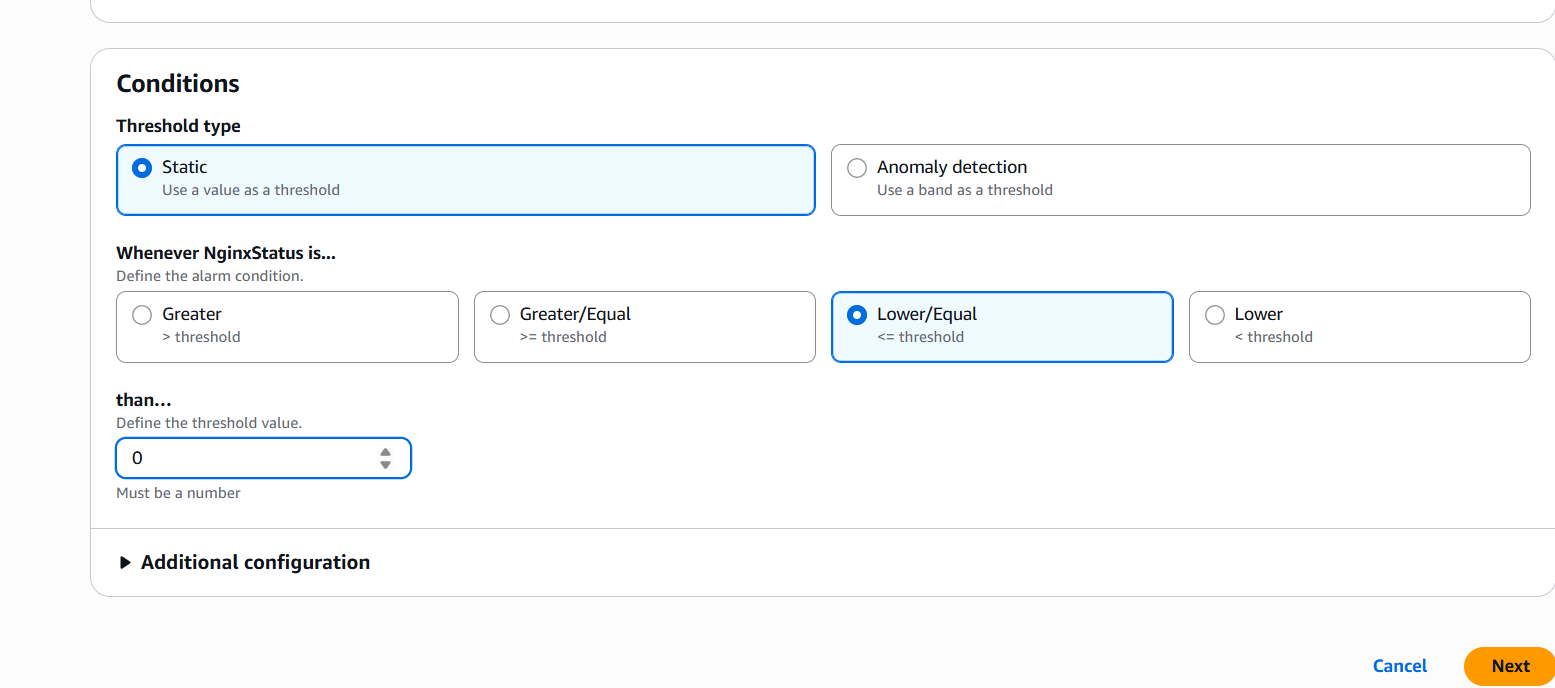
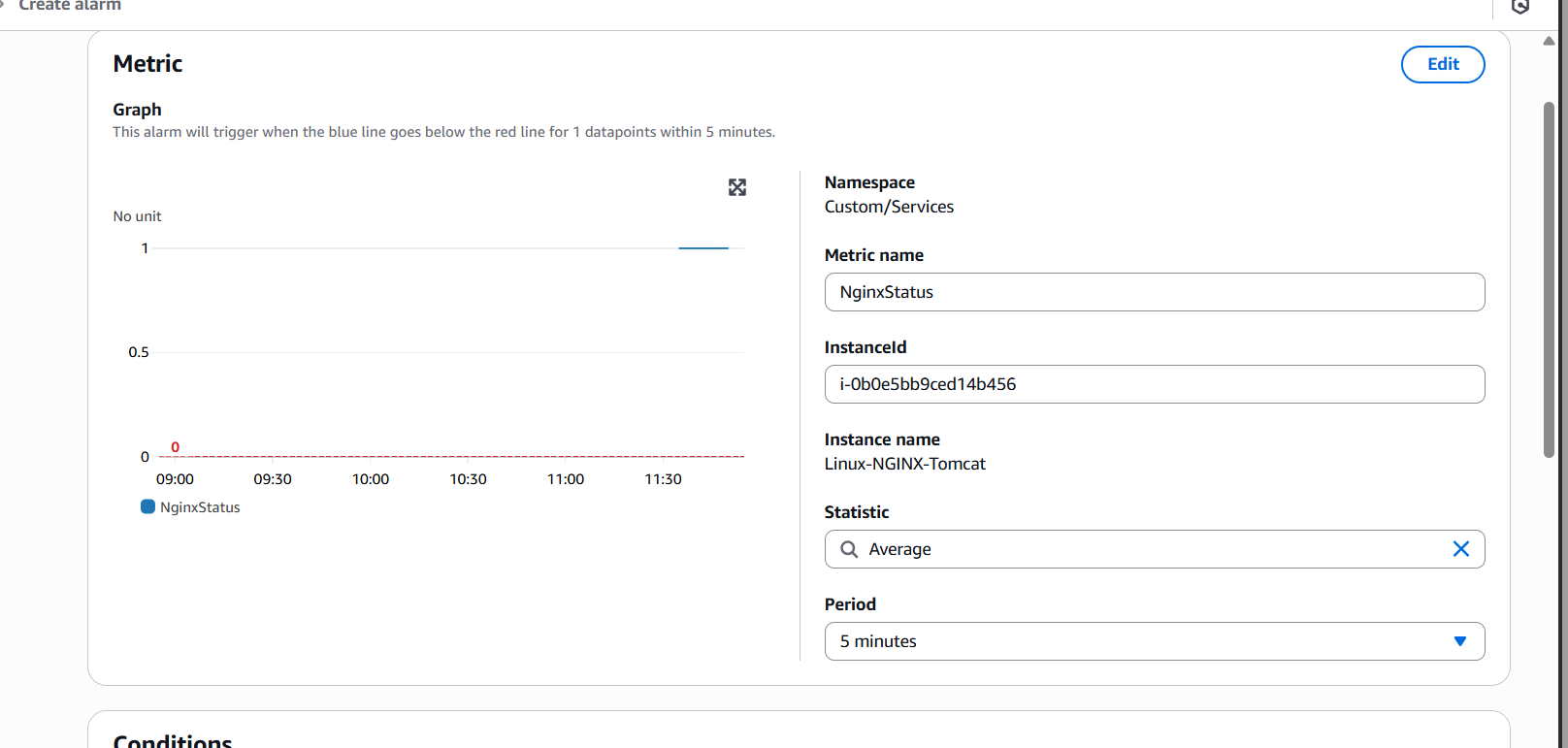




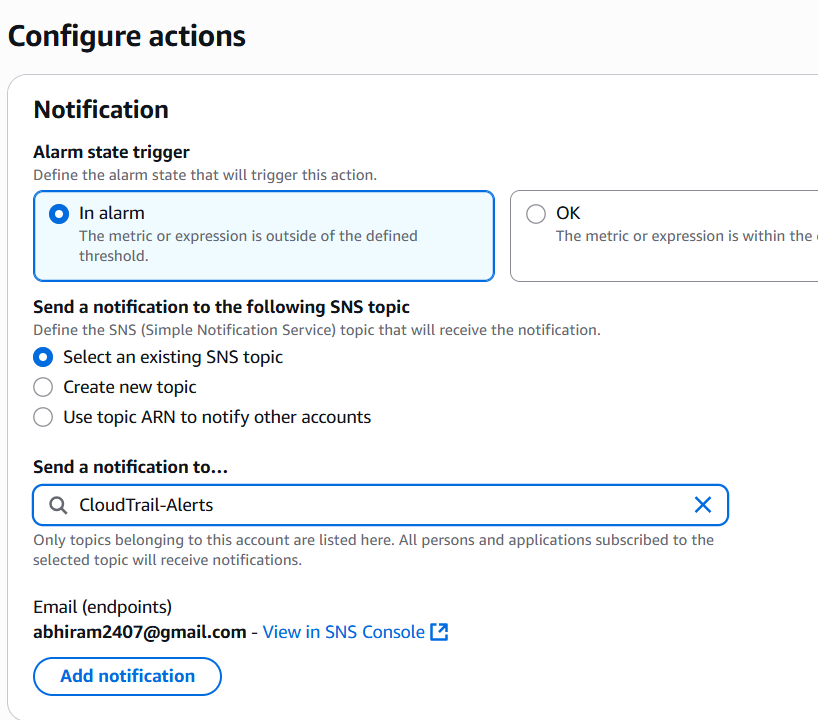
* Create CloudWatch Alarm
  + Go to AWS Console -> CloudWatch.
  + In the left navigation pane, click "Alarms" -> "Create alarm".
  + Click "Select metric".
  + Under "Custom Namespaces",you should see your given name.
  + Select InstanceId -> Find your instance ID.
  + Select the TomcatStatus metric.Click "Select metric".



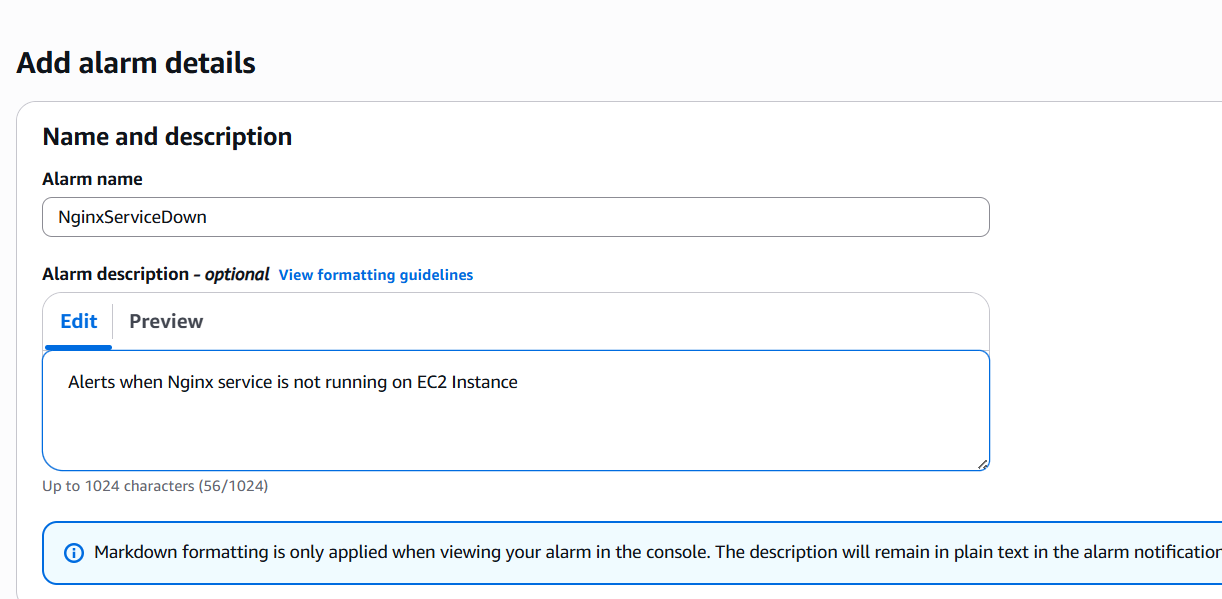
* + Specify metric and conditions:
  + Statistic: Average
  + Period: 5 minute
  + Threshold type: Static
  + Whenever TomcatStatus is: Lower/Equal
  + than: 0 (This means if the script reports 0, it triggers).



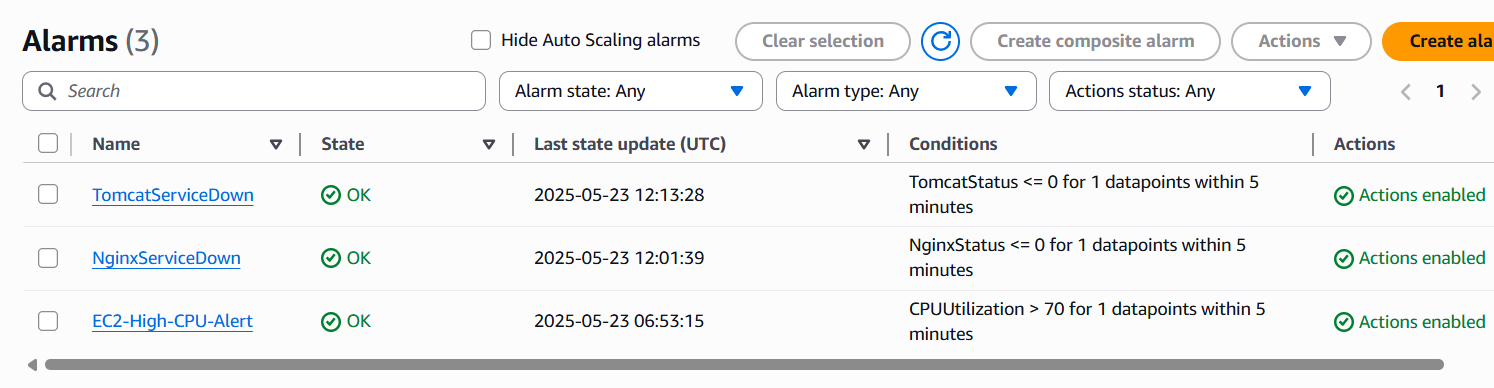
* + Click "Next".
  + Configure actions:
  + In alarm: Select an existing SNS topic
  + Send a notification to: Select your topic
  + Click "Next".



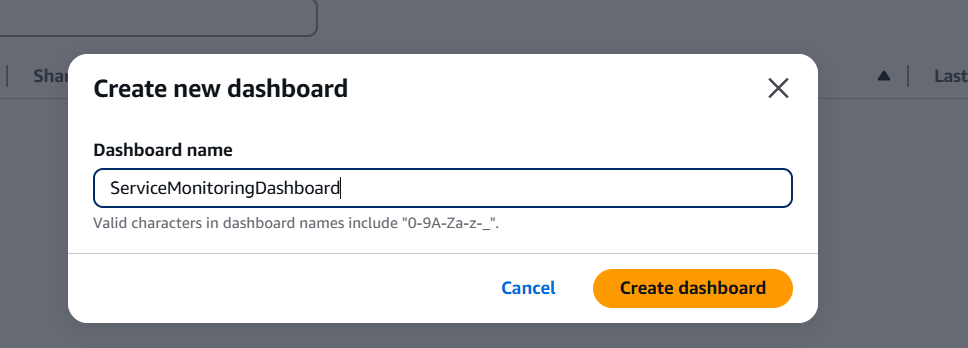
* + Add name and description then Click "Next" and then "Create alarm".

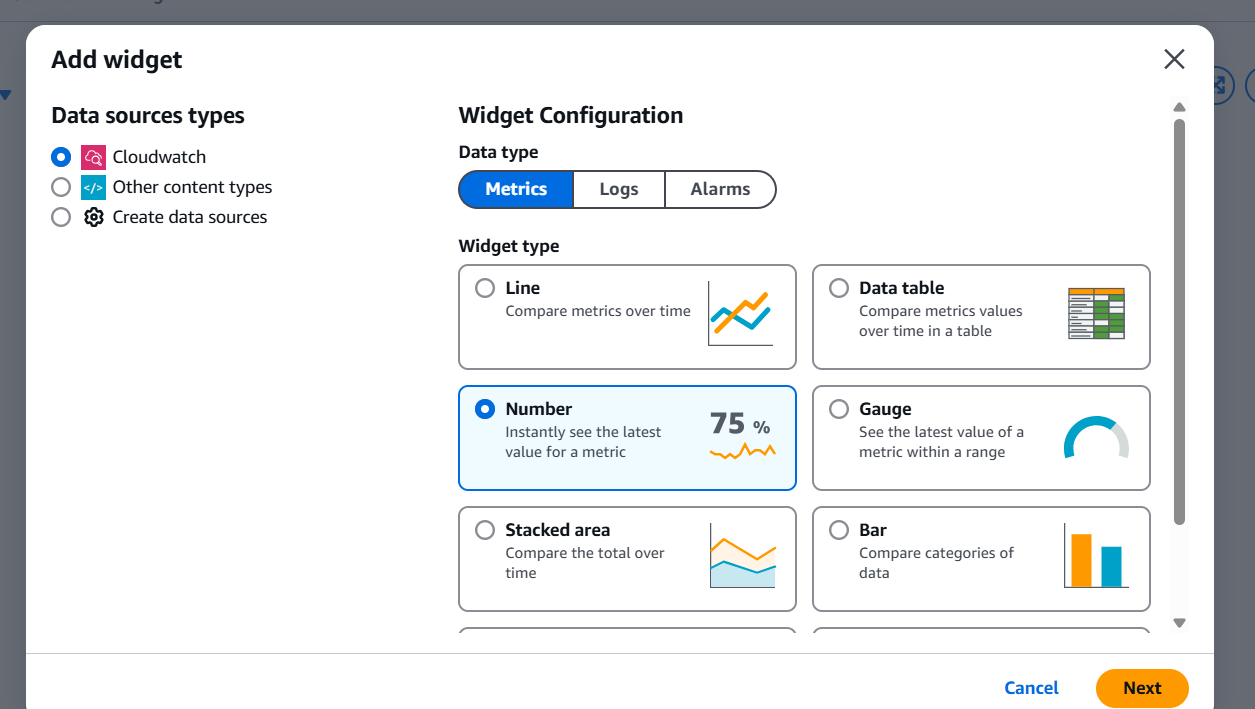


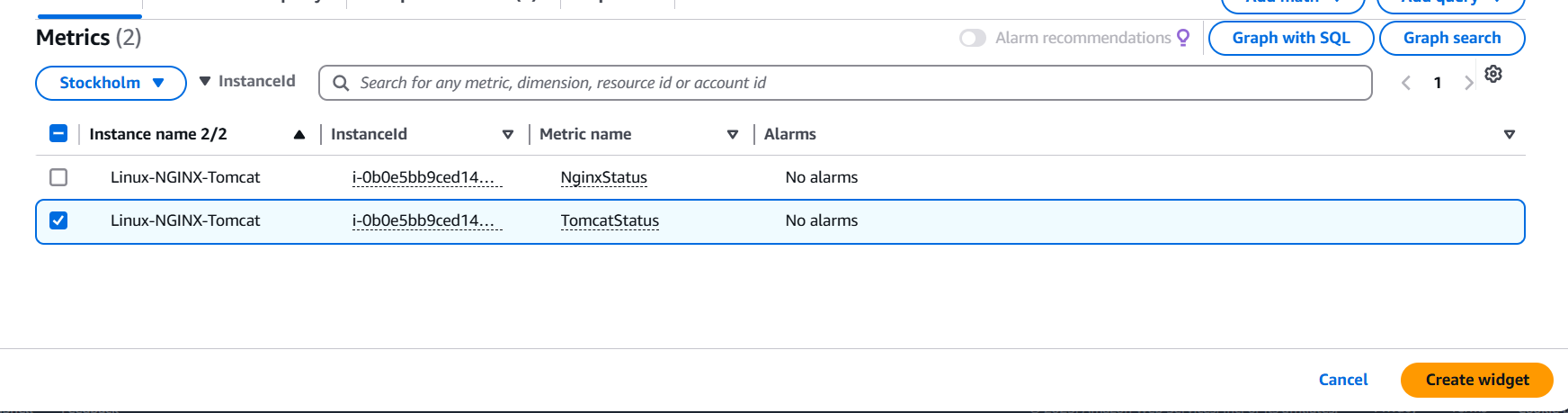
* Repeat the same process for NGINX too.

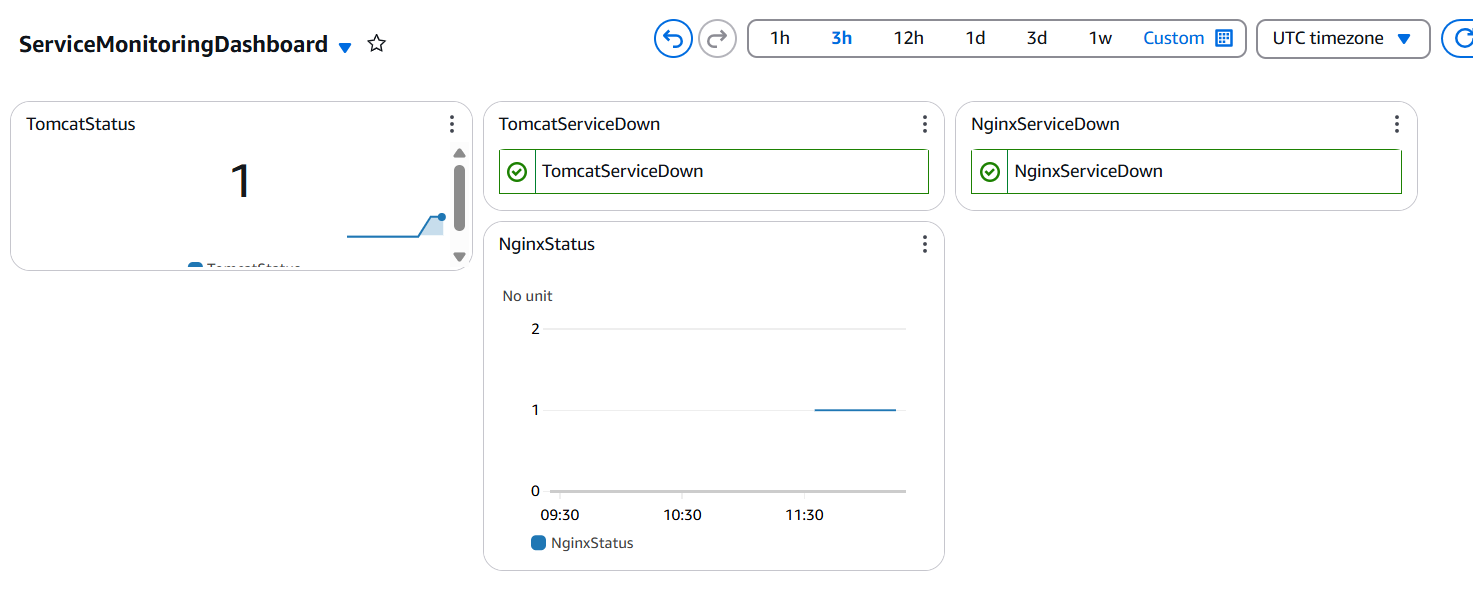


* Create CloudWatch Dashboard
  + Go to AWS Console -> CloudWatch.
  + In the left navigation pane, click "Dashboards" -> "Create dashboard".
  + Give it a name (e.g., ServiceMonitoringDashboard).
  + Add a Widget for Tomcat Status:
  + Choose "Line" (or "Number") and click "Next".
  + Click "Select metric".
  + Under "Custom Namespaces", choose yours -> InstanceId -> your instance ID.
  + Select the TomcatStatus metric.
  + Click "Create widget".









* Testing
  + Go to your EC2 instance via SSH.
  + Stop the Tomcat service and NGINX service
  + Wait for a minute or two.
  + You should receive an email from SNS indicating the TomcatServiceDown alarm is in ALARM state and NginxServiceDown alarm is in ALARM state.

