Cloud watch

1) It used to monitor our services Cloud watch is used to monitor our infrastructure and set centralised logs for our services
2) Use cloud watch agent we can get logs
3) In alarms we have 3 states 1) ok (it is healthy) 2) insufficient state (we did not get the data) 3) in alarm (it is on alarm)
4) It has to types monitoring 1) basic monitoring (it show every 5 min once) 2) detailed monitoring (it show every minute)
5) Using cloud watch and set SNS(simple notification service) we get mail
6) Using cloud watch by default we cannot monitor memory utilization by using cloud watch agent we can monitor
7) System hardware and boot level issue time we get system check failing

==========
===

1) What is aws cloud watch?

AWS CloudWatch is a monitoring and observability service provided by Amazon Web Services. It allows you to monitor your AWS resources and the applications you run on AWS in real time. Here are some key features and uses of AWS CloudWatch:

Key Features

1. **Metrics Collection**: CloudWatch collects and tracks metrics for AWS services like EC2 instances, RDS databases, and DynamoDB tables. It also supports custom metrics, allowing you to monitor any part of your application.
2. **Logs Monitoring**: CloudWatch Logs lets you monitor, store, and access log files from EC2 instances, AWS CloudTrail, and other sources. You can set up log data retention and query logs using CloudWatch Logs Insights.
3. **Alarms**: You can create CloudWatch Alarms to trigger actions based on metric thresholds. For example, an alarm can notify you if CPU usage on an EC2 instance exceeds a certain percentage.
4. **Dashboards**: CloudWatch Dashboards provide a customizable view of your metrics and logs. You can create visualizations like graphs and charts to monitor application performance and operational health.
5. **Events**: CloudWatch Events (now part of Amazon EventBridge) enables you to respond to changes in your AWS resources. You can set up rules to trigger actions like Lambda functions or SNS notifications when specific events occur.
6. **Insights**: CloudWatch provides insights into your infrastructure and applications through features like CloudWatch Logs Insights and Container Insights. These tools help you analyze and debug performance issues.

Uses

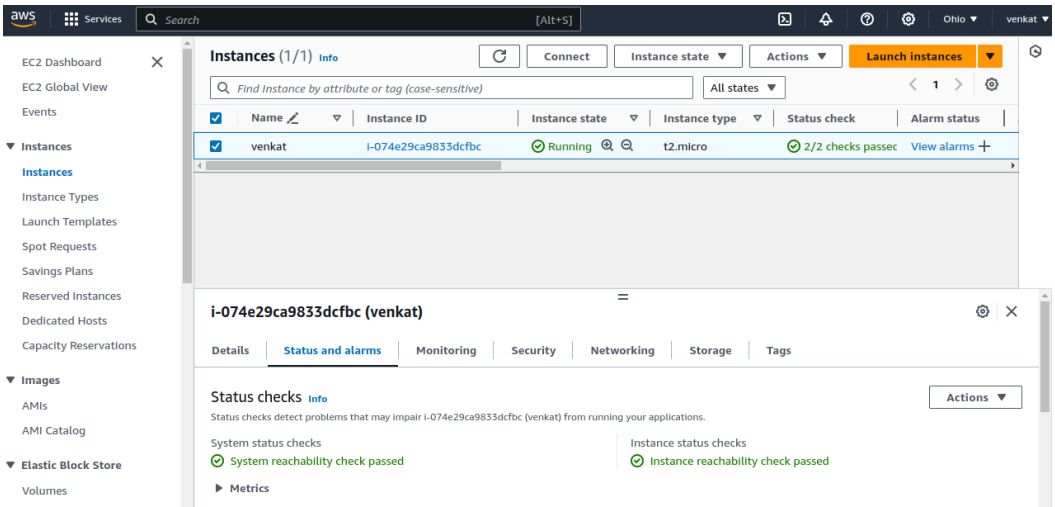
1. **Resource Monitoring**: Keep track of the health and performance of your AWS resources (e.g., EC2 instances, RDS databases) by monitoring metrics like CPU utilization, memory usage, and network traffic.
2. **Application Performance Monitoring**: Monitor the performance of your applications by collecting and analyzing metrics, logs, and traces. This helps in identifying bottlenecks and performance issues.
3. **Incident Response**: Set up alarms and notifications to alert you when something goes wrong. This enables quick responses to incidents, reducing downtime and improving system reliability.
4. **Log Management**: Centralize and manage log data from various sources. Use CloudWatch Logs Insights to query and analyze logs for troubleshooting and debugging.
5. **Automated Actions**: Automate responses to certain events or conditions. For instance, you can automatically scale resources up or down based on usage patterns or trigger Lambda functions for automated remediation.
6. **Compliance and Security**: Monitor and log access and activity within your AWS environment to ensure compliance with security policies and regulations. CloudWatch can be integrated with AWS CloudTrail to provide detailed logging of API calls and other activities.
7. **Cost Management**: Analyze usage patterns and optimize resource utilization to manage costs effectively. CloudWatch provides insights into how your resources are being used, helping you to identify and eliminate waste.

By leveraging AWS CloudWatch, you can gain better visibility into your AWS environment, improve operational efficiency, and enhance the performance and reliability of your applications.

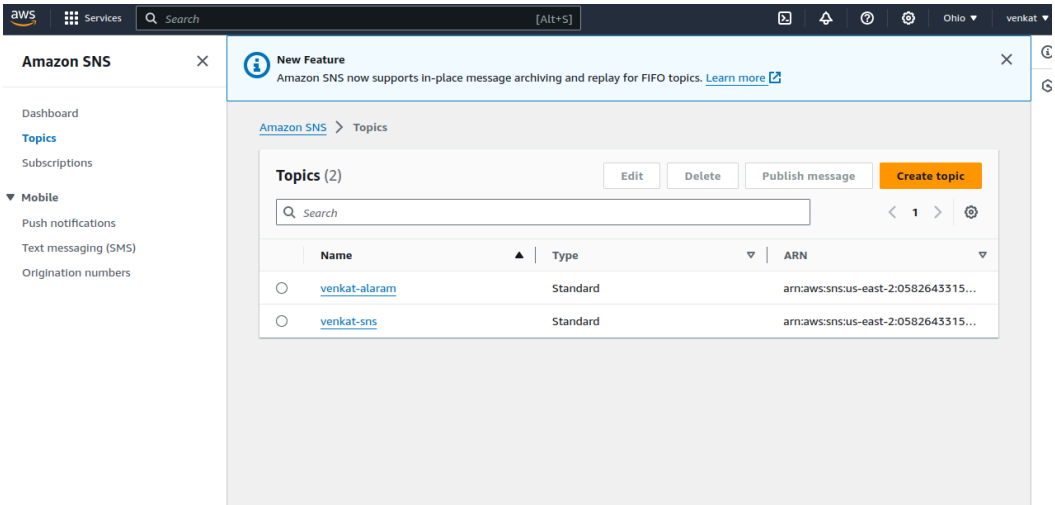
==========
======

⇒ to set 2/2 checks pass alarm set for one server

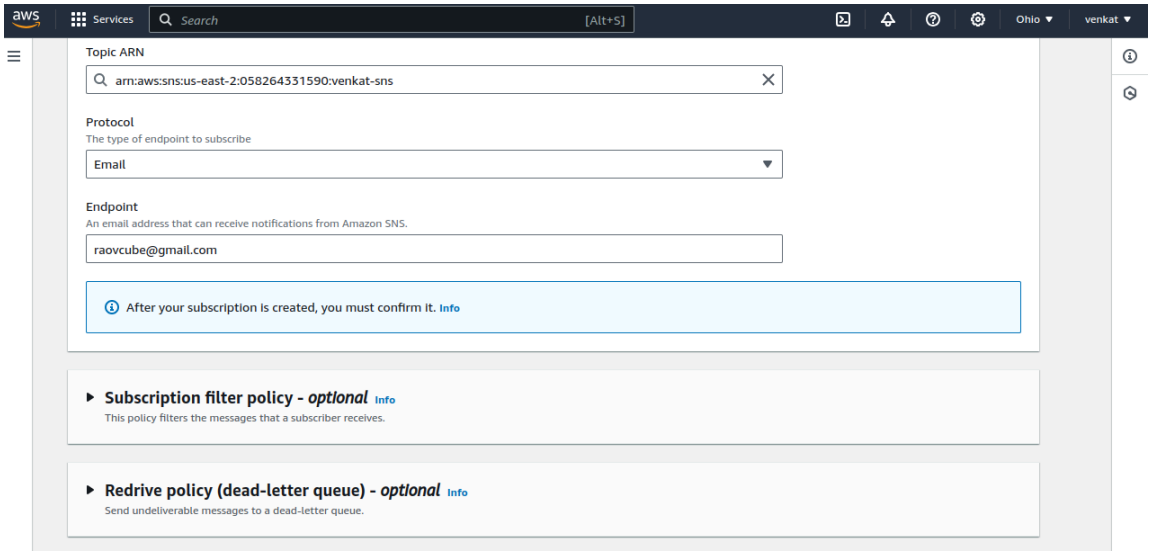
1) Go to ec2 launch ec2 server name= venkat

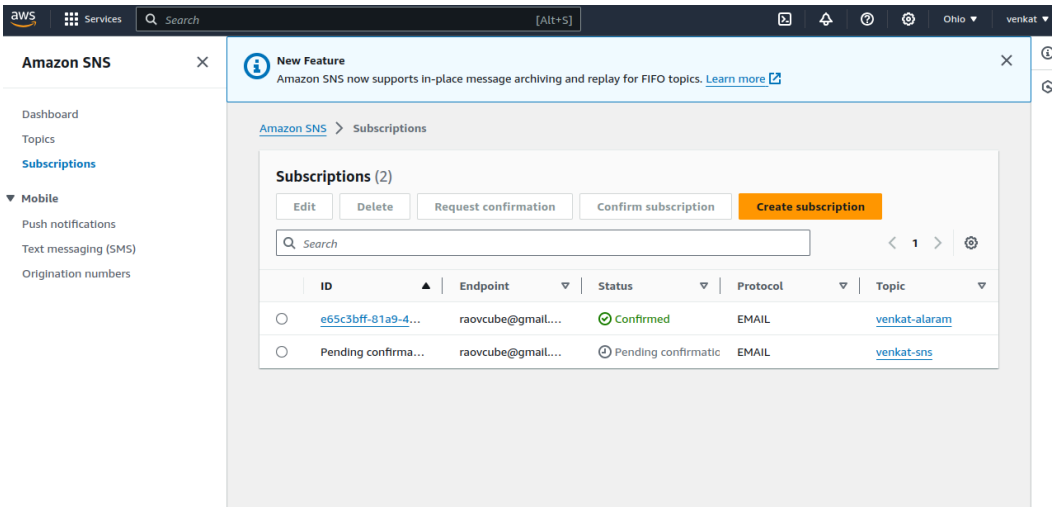


2) Now go SNS click topics click create topic select standard
3) name= venkat-sns click crete topic

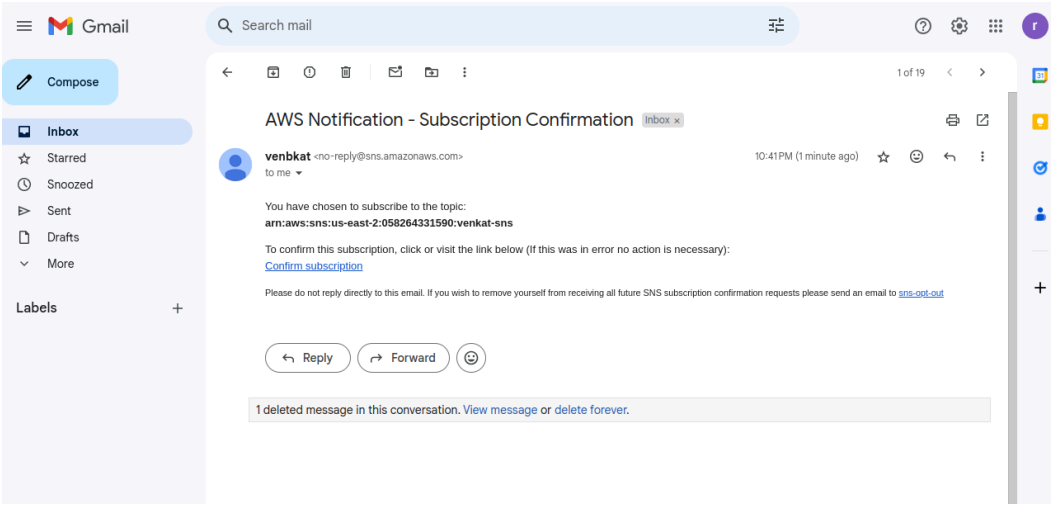


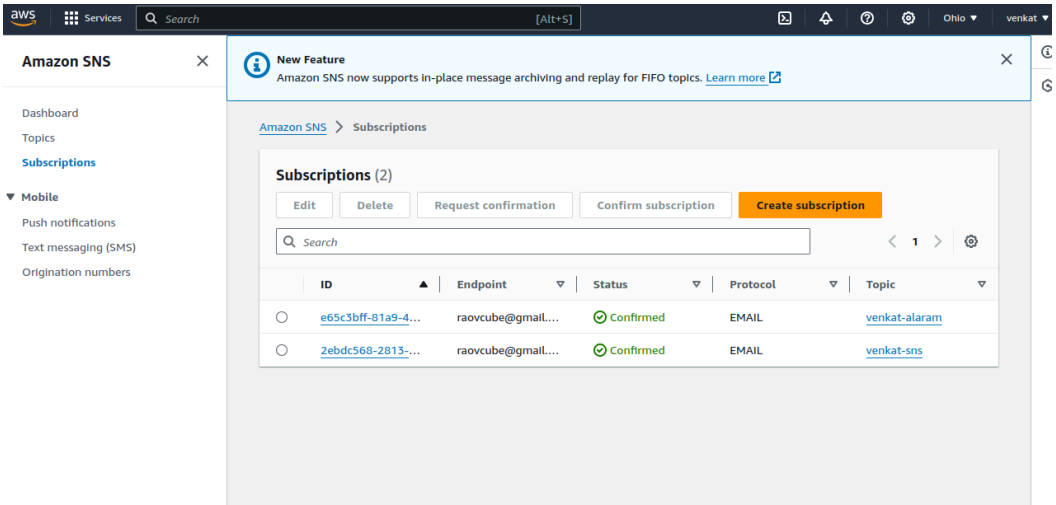
4) Now click subscriptions → create subscription
5) Topic arn= venkat-sns
6) protocol= email
7) Endpoint = raovcube@gmail.com
8) Click create subscription



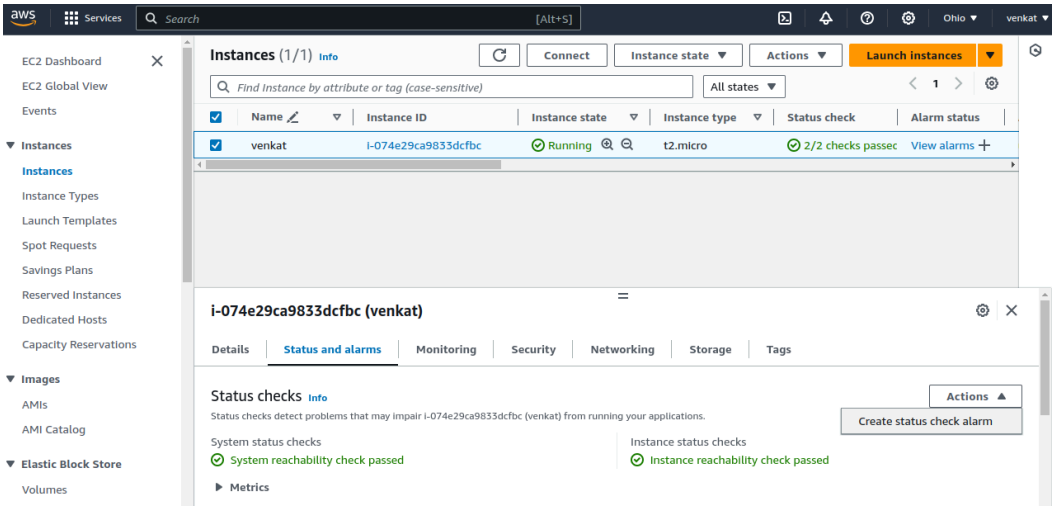


9) Now go to gmail you see subscription mail click subscription and click confirm subscription

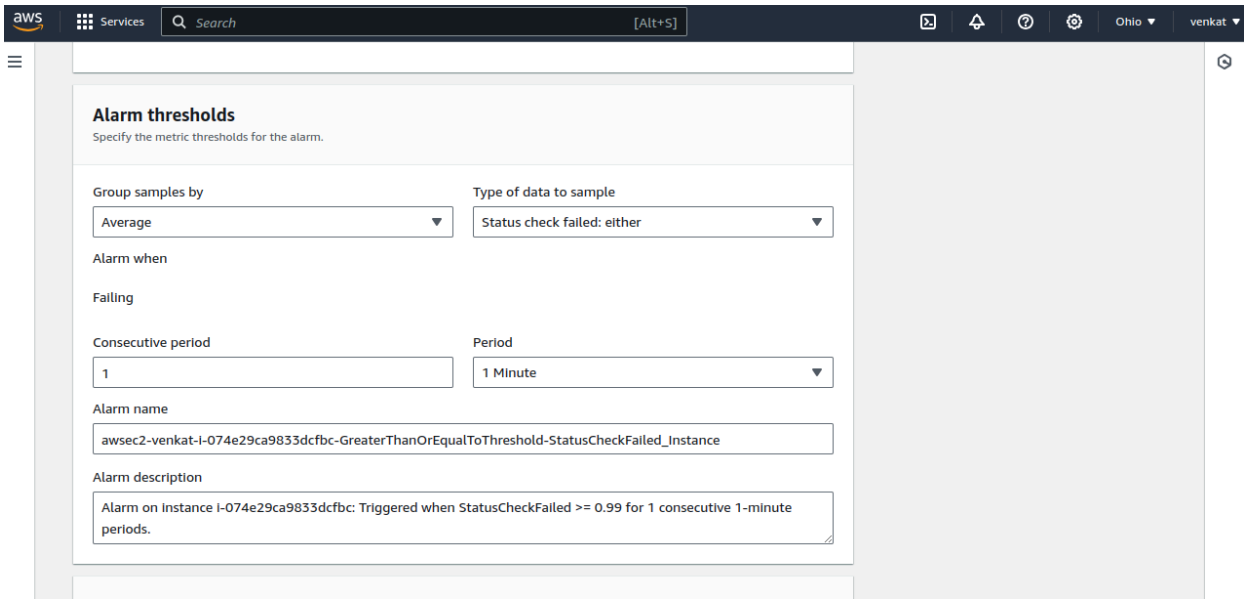




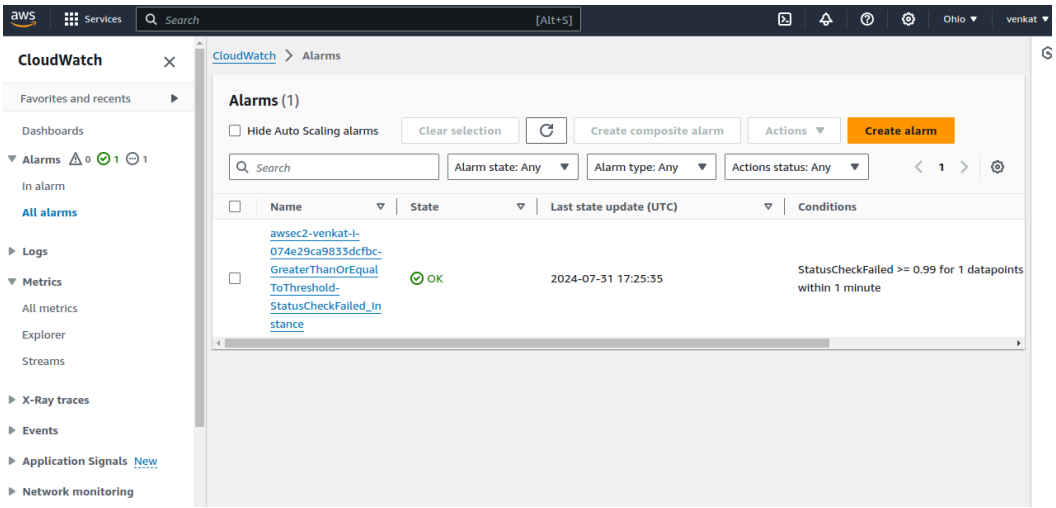
10) Now go to venkat ec2 select venkat server click status and alarm click actions click create status check alarm



11) Click create an alarm
12) Alarm notification = venkat-sns



13) Group sample by = average
14) Type of data to sample = status checkfailed: either
15) Alarm name set your wish ex: venkat-server-status-check-fail
16) Click crete
17) Go to cloud watch go to all alarms check you find it

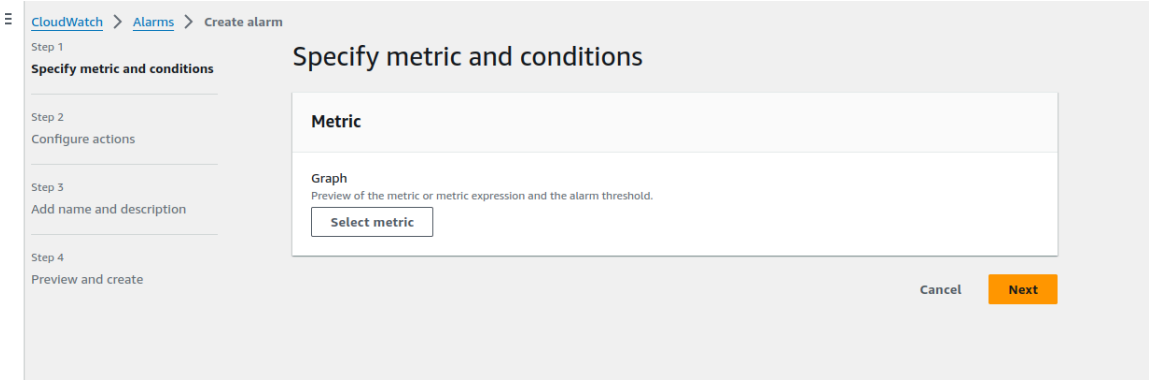


18) If status check fail you get mail

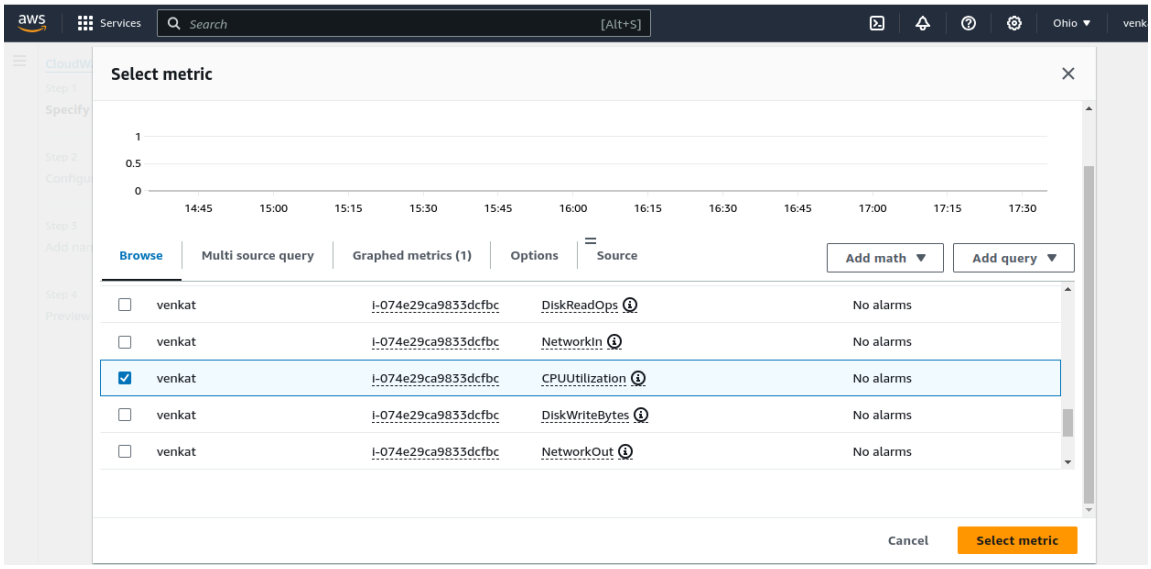
==========

⇒ crete one cloud watch alert when server cpu high get alert:

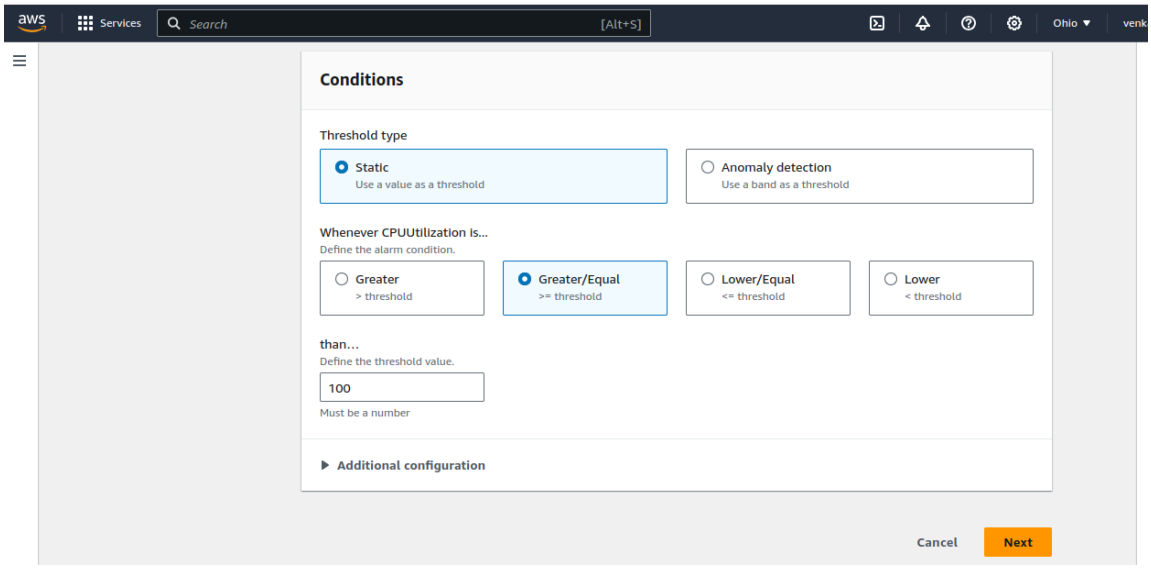
1) Launch one server name = venkat
2) Go to cloud watch and go to alarms click crete alarm



3) Click on select metric
4) Click ec2 click on per-instance metric

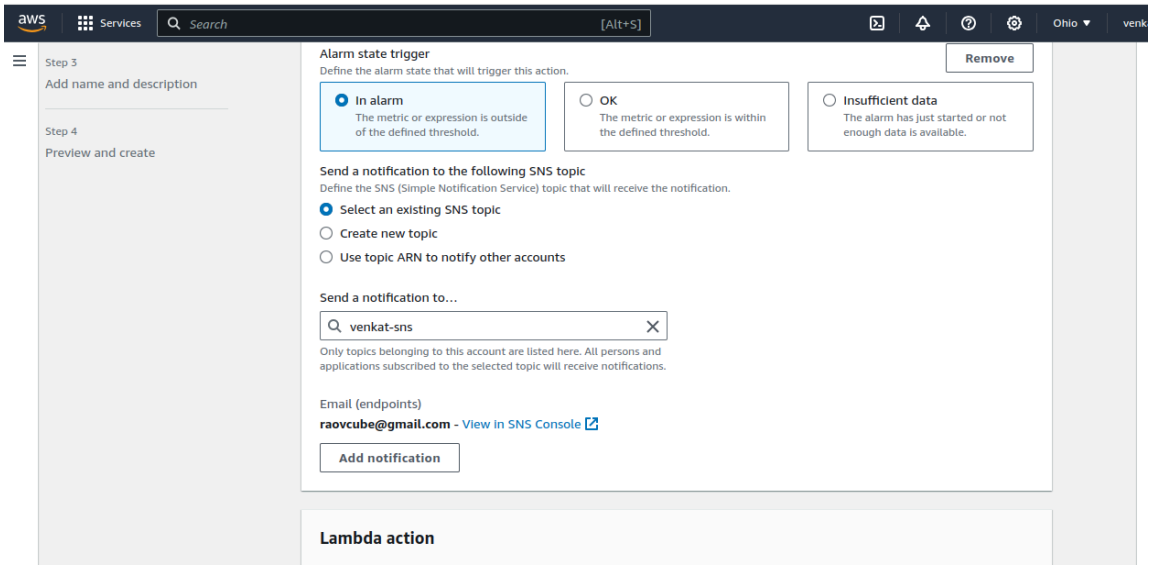


5) Select in venkat server cpu utilization and click on select metric

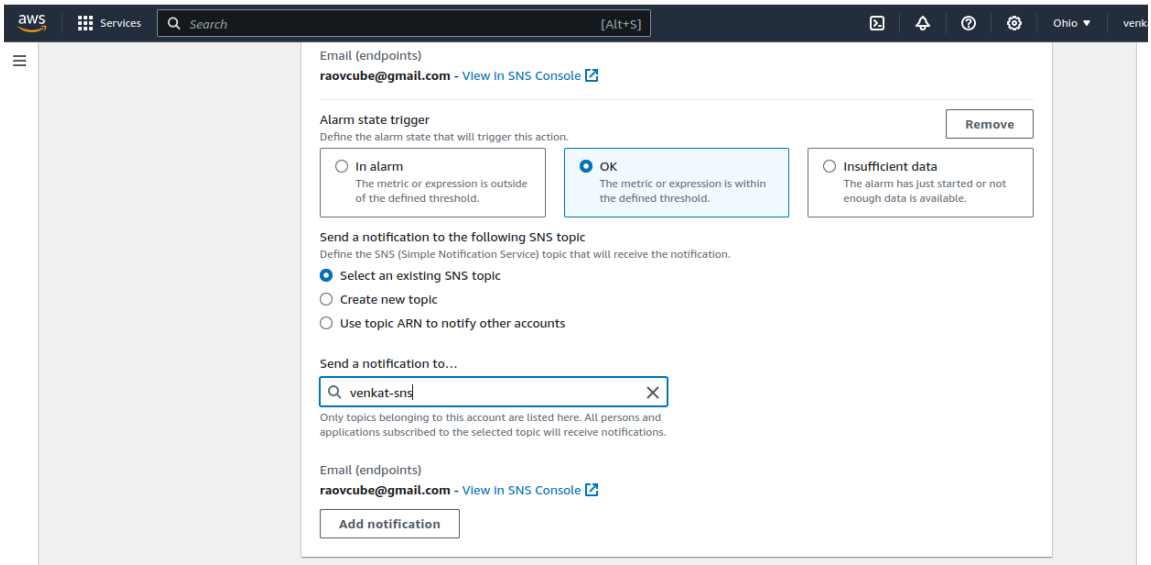


6) Threshold type = static

7) Whenever cpu utilization is = grater/equal
8) Define the threshold value = 100
9) Click next

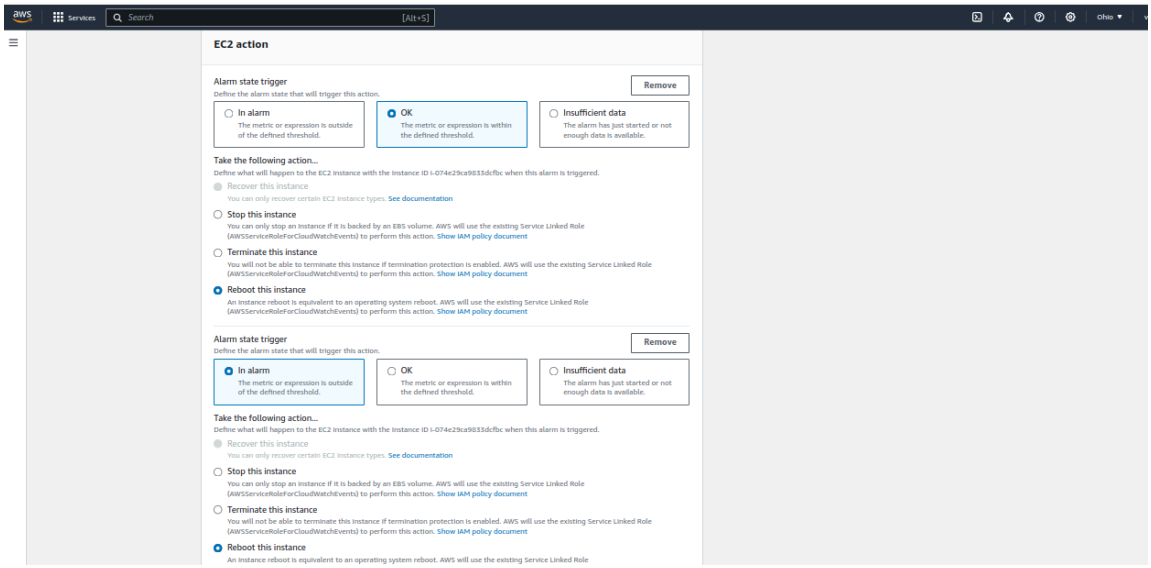


10) Alarm state trigger = in alaram
11) Click select an existing sns topic (you don't have select create new topic and create)
12) Send notification = venakt-sns (your sns select)
13) Click on add notification



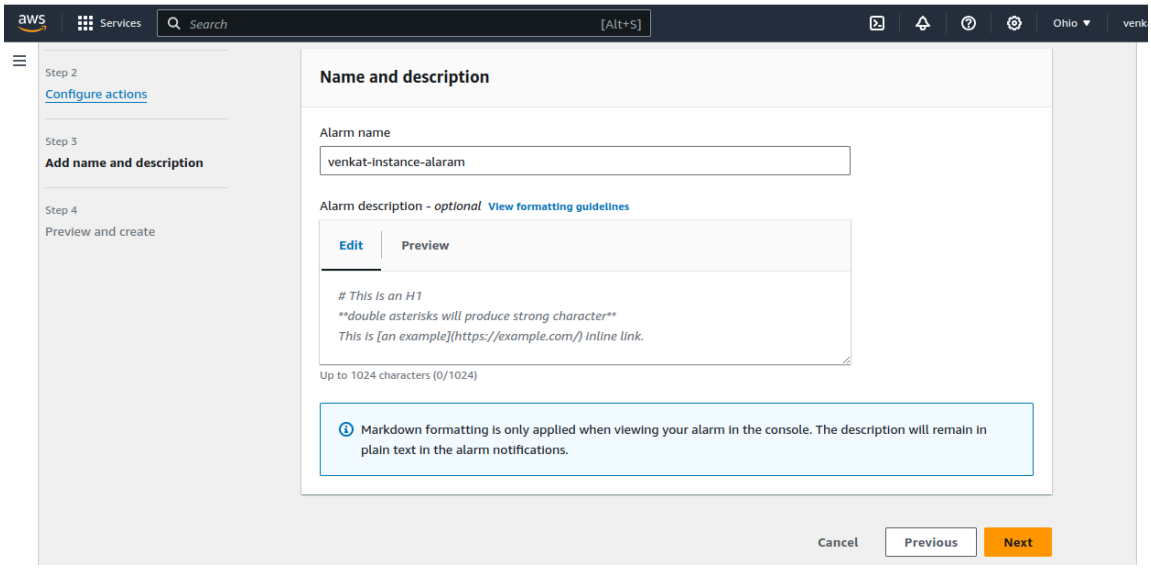
14) Alarm state trigger = ok
15) Click select an existing sns topic
16) Send notification = venakt-sns (your sns select)
17) If you want add more action ex ec2 stop or reboot or terminate time use below steps (optional) (you dont want just click next)

18)Ec2 action



19) Select once ok and again once in alarm select reboot this instance click next

20)Alarm name= venkat-instance-alarm



21)Click crete alarm

22) now connect the venkat ec2 using bash or vmware

23)Set as host name= venkat

24)Use coomnds #hostnamectl set-hostname venkat

25)#bash exec

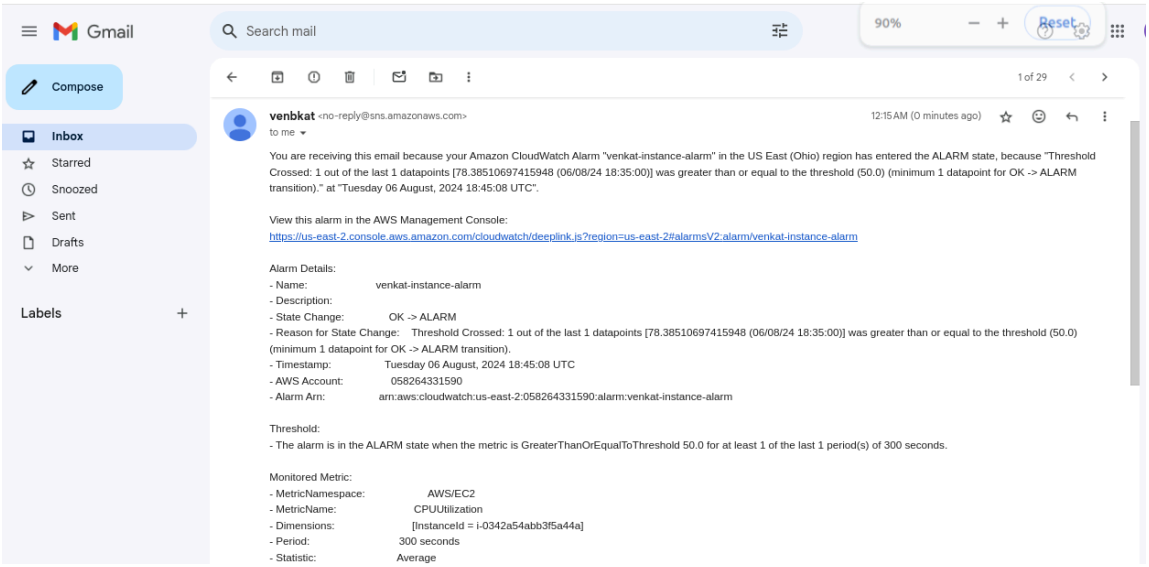
26)Now increase the cpu utilization using stress command

27)#sudo apt update

28)#sudo apt install stress -y

29)#sudo stress --cpu 12 --timeout 500

30)Now go to gmail your sns mail check you get mails cpu alerts

31) 

32)Now once reboot your server this time also you get mails

==========