

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 two 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

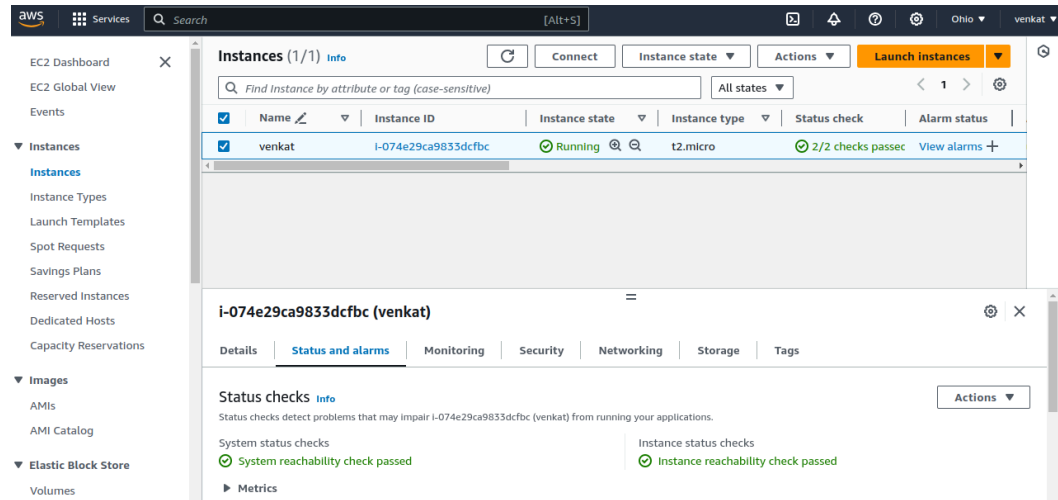
1. **Resource Monitoring:** Keep track of the health and performance of your AWS resources (e.g., EC2 instances, RDS databases) by monitoring metrics like CPU utilization, memory usage, and network traffic.
2. **Application Performance Monitoring:** Monitor the performance of your applications by collecting 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.

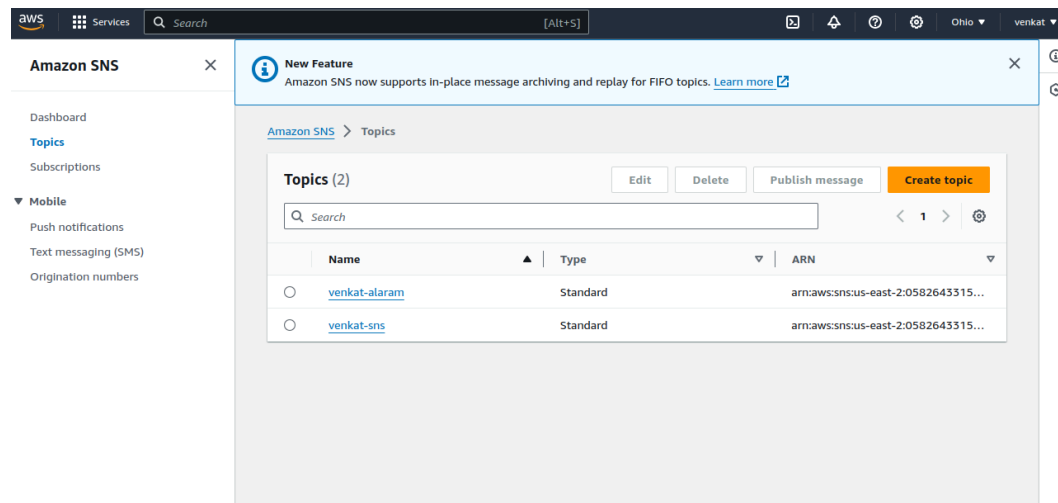
=====

⇒ 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

Topic ARN

Protocol
 The type of endpoint to subscribe

Endpoint
 An email address that can receive notifications from Amazon SNS.

After your subscription is created, you must confirm it. [Info](#)

Subscription filter policy - optional [Info](#)
 This policy filters the messages that a subscriber receives.

Redrive policy (dead-letter queue) - optional [Info](#)
 Send undeliverable messages to a dead-letter queue.

Amazon SNS

Dashboard
 Topics
Subscriptions
 Mobile
 Push notifications
 Text messaging (SMS)
 Origination numbers

New Feature
 Amazon SNS now supports in-place message archiving and replay for FIFO topics. [Learn more](#)

Amazon SNS > Subscriptions

Subscriptions (2)

Edit Delete Request confirmation Confirm subscription Create subscription

Search

ID	Endpoint	Status	Protocol	Topic
e65c3bff-81a9-4...	raovcube@gmail...	Confirmed	EMAIL	venkat-alarum
Pending confirma...	raovcube@gmail...	Pending confirmatic	EMAIL	venkat-sns

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

Gmail

Compose

Inbox

Starred
 Snoozed
 Sent
 Drafts
 More

Labels

Search mail

1 of 19

AWS Notification - Subscription Confirmation

venbkat <no-reply@sns.amazonaws.com> to me

10:41 PM (1 minute ago)

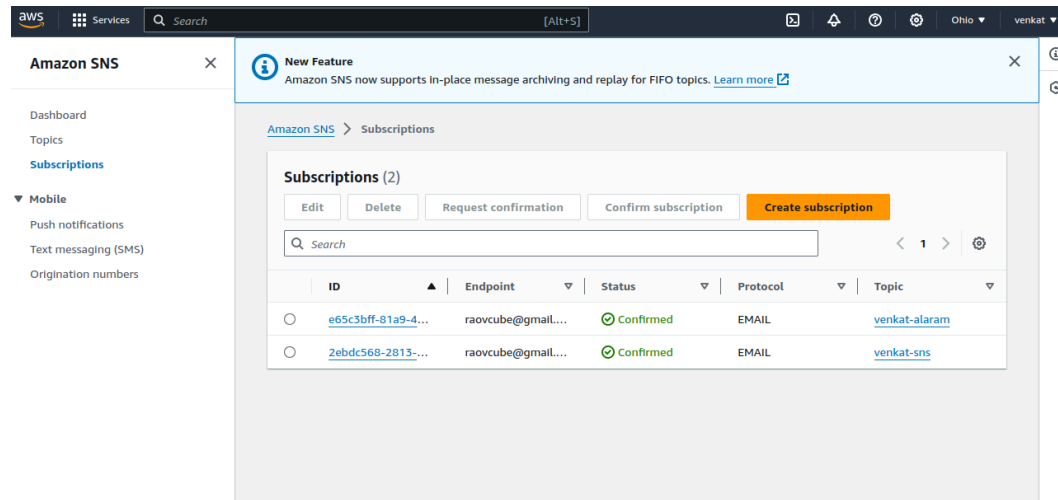
You have chosen to subscribe to the topic:
 arn:aws:sns:us-east-2:058264331590:venkat-sns

To confirm this subscription, click or visit the link below (If this was in error no action is necessary):
[Confirm subscription](#)

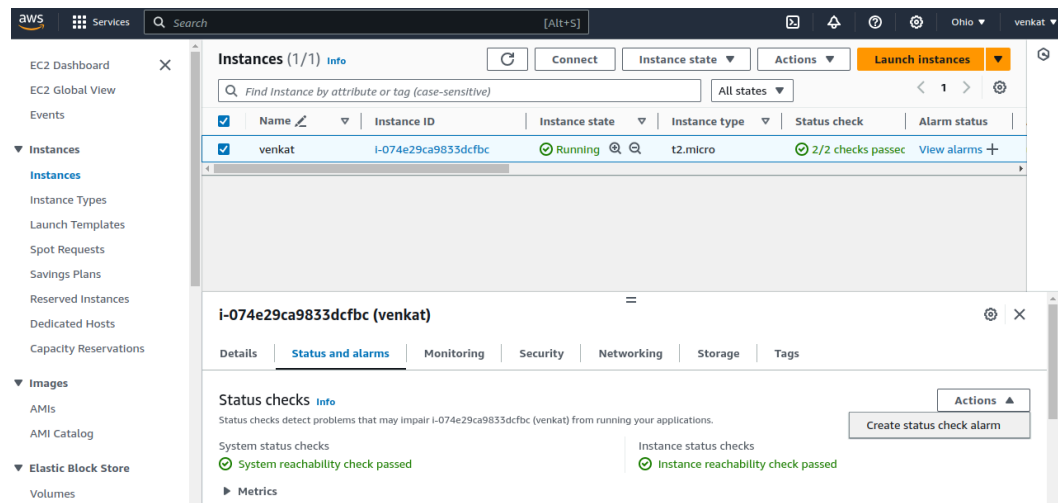
Please do not reply directly to this email. If you wish to remove yourself from receiving all future SNS subscription confirmation requests please send an email to [sns-opt-out](#)

Reply Forward

1 deleted message in this conversation. [View message](#) or [delete forever](#).

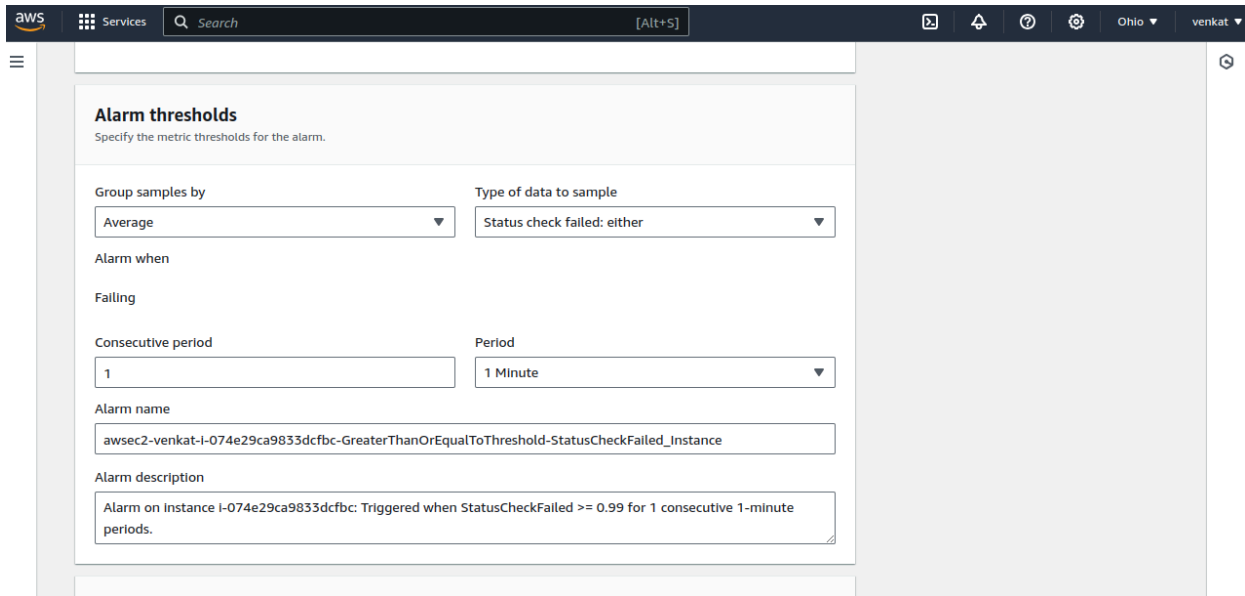


10) Now go to venkat ec2 select venkat server click status and alarm click actions click create status check alarm



11) Click create an alarm

12) Alarm notification = venkat-sns



Alarm thresholds
Specify the metric thresholds for the alarm.

Group samples by: Average Type of data to sample: Status check failed: either

Alarm when: Failing

Consecutive period: 1 Period: 1 Minute

Alarm name: awsec2-venkat-i-074e29ca9833dcfbc-GreaterThanOrEqualToThreshold-StatusCheckFailed_Instance

Alarm description: Alarm on Instance i-074e29ca9833dcfbc: Triggered when StatusCheckFailed >= 0.99 for 1 consecutive 1-minute periods.

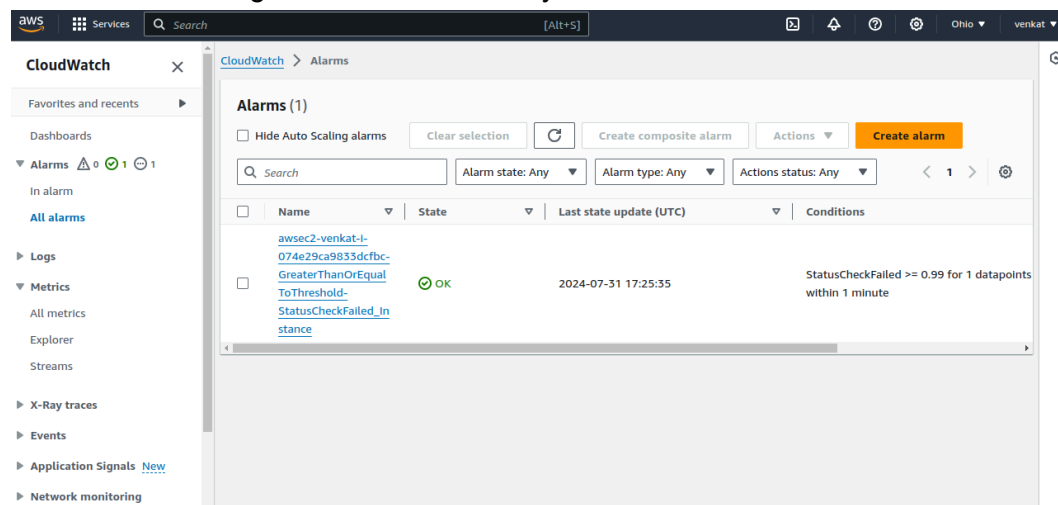
13) Group sample by = average

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



CloudWatch × CloudWatch > Alarms

Alarms (1)

☐ Hide Auto Scaling alarms Clear selection ↻ Create composite alarm Actions Create alarm

Alarm state: Any Alarm type: Any Actions status: Any < 1 > ⚙

<input type="checkbox"/>	Name	State	Last state update (UTC)	Conditions
<input type="checkbox"/>	awsec2-venkat-i-074e29ca9833dcfbc-GreaterThanOrEqualToThreshold-StatusCheckFailed_Instance	OK	2024-07-31 17:25:35	StatusCheckFailed >= 0.99 for 1 datapoints within 1 minute

▶ Log
 ▶ Metrics
 All metrics
 Explorer
 Streams
 ▶ X-Ray traces
 ▶ Events
 ▶ Application Signals New
 ▶ Network monitoring

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

CloudWatch > Alarms > Create alarm

Step 1
Specify metric and conditions

Step 2
Configure actions

Step 3
Add name and description

Step 4
Preview and create

Specify metric and conditions

Metric

Graph
Preview of the metric or metric expression and the alarm threshold.

Select metric

Cancel Next

3) Click on select metric

4) Click ec2 click on per-instance metric

aws Services Search [Alt+S]

CloudWatch

Step 1
Specify

Step 2
Configure

Step 3
Add name

Step 4
Preview

Select metric

1
0.5
0

14:45 15:00 15:15 15:30 15:45 16:00 16:15 16:30 16:45 17:00 17:15 17:30

Browse Multi source query Graphed metrics (1) Options Source Add math Add query

<input type="checkbox"/>	venkat	i-074e29ca9833dcfbc	DiskReadOps	No alarms
<input type="checkbox"/>	venkat	i-074e29ca9833dcfbc	NetworkIn	No alarms
<input checked="" type="checkbox"/>	venkat	i-074e29ca9833dcfbc	CPUUtilization	No alarms
<input type="checkbox"/>	venkat	i-074e29ca9833dcfbc	DiskWriteBytes	No alarms
<input type="checkbox"/>	venkat	i-074e29ca9833dcfbc	NetworkOut	No alarms

Cancel Select metric

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

aws Services Search [Alt+S]

Conditions

Threshold type

Static Use a value as a threshold

Anomaly detection Use a band as a threshold

Whenever CPUUtilization is...
Define the alarm condition.

Greater > threshold

Greater/Equal >= threshold

Lower/Equal <= threshold

Lower < threshold

than...
Define the threshold value.

100

Must be a number

Additional configuration

Cancel Next

6) Threshold type = static

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

The screenshot shows the AWS CloudWatch console interface. On the left, a sidebar indicates the current step is 'Step 3: Add name and description'. The main content area is titled 'Alarm state trigger' and includes a 'Remove' button. Under 'Define the alarm state that will trigger this action.', three radio buttons are present: 'In alarm' (selected), 'OK', and 'Insufficient data'. Below this, the 'Send a notification to the following SNS topic' section is active, showing 'Select an existing SNS topic' as the chosen option. A search box contains 'venkat-sns'. At the bottom, an email endpoint 'raovcube@gmail.com' is listed with an 'Add notification' button.

- 10) Alarm state trigger = in alarm
- 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

This screenshot shows the same AWS CloudWatch console interface, but now the 'Add notification' button is highlighted. The 'Alarm state trigger' section now has 'OK' selected. The 'Send a notification to the following SNS topic' section remains the same, with 'Select an existing SNS topic' chosen and 'venkat-sns' in the search box. The email endpoint 'raovcube@gmail.com' and the 'Add notification' button are still visible at the bottom.

- 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

The screenshot shows the 'EC2 action' configuration page in the AWS IAM console. It is divided into two identical sections, each with a 'Remove' button in the top right corner.

Alarm state trigger
Define the alarm state that will trigger this action.

☐ In alarm
The metric or expression is outside of the defined threshold.

☒ OK
The metric or expression is within the defined threshold.

☐ Insufficient data
The alarm has just started or not enough data is available.

Take the following action...
Define what will happen to the EC2 instance with the Instance ID i-074c29ca9833d6fc when this alarm is triggered.

☐ Recover this instance
You can only recover certain EC2 instance types. [See documentation](#)

☐ Stop this instance
You can only stop an instance if it is backed by an EBS volume. AWS will use the existing Service Linked Role (AWSServiceRoleForCloudWatchEvents) to perform this action. [Show IAM policy document](#)

☐ Terminate this instance
You will not be able to terminate this instance if termination protection is enabled. AWS will use the existing Service Linked Role (AWSServiceRoleForCloudWatchEvents) to perform this action. [Show IAM policy document](#)

☒ Reboot this instance
An instance reboot is equivalent to an operating system reboot. AWS will use the existing Service Linked Role (AWSServiceRoleForCloudWatchEvents) to perform this action. [Show IAM policy document](#)

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

20) Alarm name= venkat-instance-alarm

The screenshot shows the 'Name and description' step of the alarm creation process in the AWS IAM console. The left sidebar shows the progress: Step 2 (Configure actions), Step 3 (Add name and description), and Step 4 (Preview and create).

Name and description

Alarm name
venkat-instance-alarm

Alarm description - optional [View formatting guidelines](#)

Edit | **Preview**

This is an H1
double asterisks will produce strong character
This is [an example](https://example.com/) inline link.

Up to 1024 characters (0/1024)

Markdown formatting is only applied when viewing your alarm in the console. The description will remain in plain text in the alarm notifications.

Cancel Previous **Next**

21) Click create alarm

22) now connect the venkat ec2 using bash or vmware

23) Set as host name= venkat

24) Use commands #hostnamectl set-hostname venkat

25) #bash exec

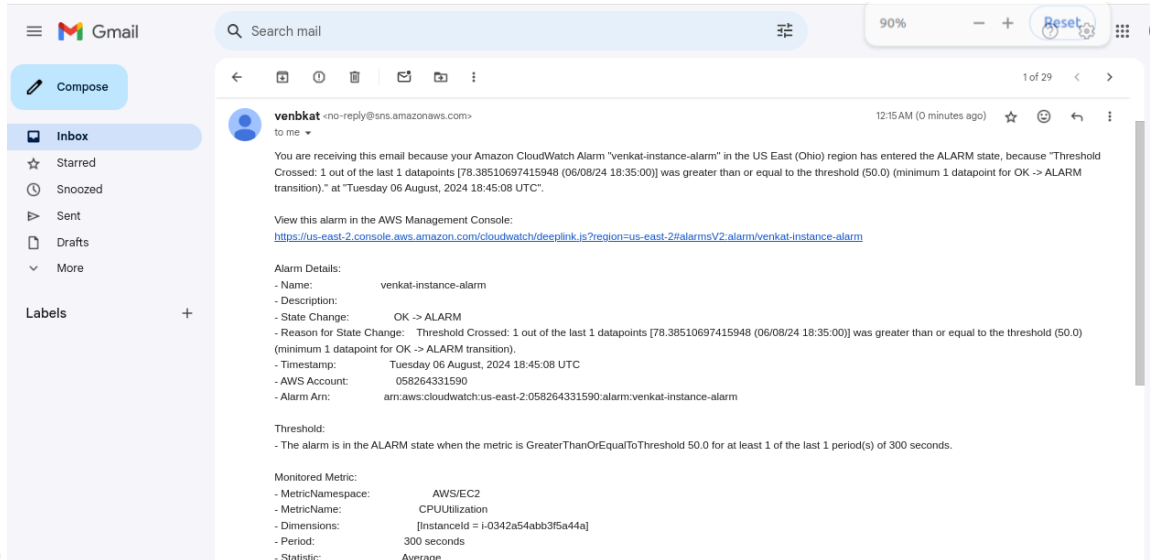
26) Now increase the cpu utilization using stress command

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

=====

=====