AWS_CloudWatchProject

Amazon **CloudWatch** is a monitoring and observability service for AWS resources and applications. It collects and tracks metrics, monitors log files, sets alarms, and helps gain system-wide visibility.

In this tutorial, we will create a demo to monitor CPU utilization on an EC2 instance. Here's the step-by-step flow:

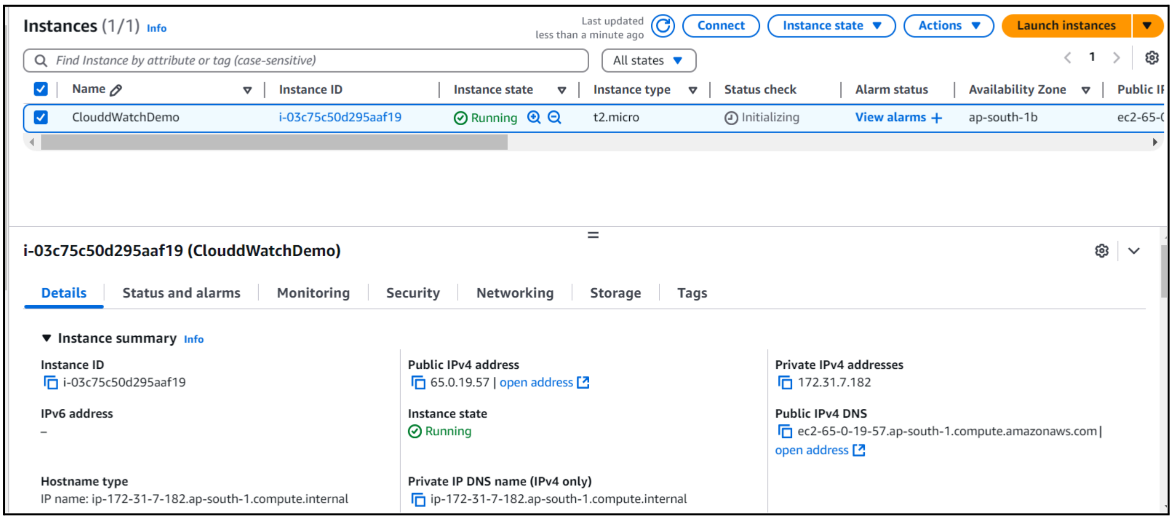
1. **Launch an EC2 instance**.
2. **Run a Python script** to simulate high CPU usage, reaching an 80% threshold.
3. Use **AWS CloudWatch** to monitor the CPU utilization.
4. Set up an **SNS (Simple Notification Service)** to send an email alert when the threshold is crossed.

Let's get started! 🚀

Steps to Simulate CPU Spike and Monitor Utilization

Step 1: Launch an EC2 Instance

1. **Launch an EC2 instance** through the AWS Management Console.
2. **Connect to the instance** via SSH to run the Python script.



Instances (1/1) Info
Last updated less than a minute ago
Connect
Instance state ▼
Actions ▼
Launch instances ▼
Find Instance by attribute or tag (case-sensitive)
All states ▼
1

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IF
☑	ClouddWatchDemo	i-03c75c50d295aaf19	Running	t2.micro	Initializing	View alarms +	ap-south-1b	ec2-65-(

i-03c75c50d295aaf19 (ClouddWatchDemo)

▼ Instance summary Info

Instance ID
i-03c75c50d295aaf19

Public IPv4 address
65.0.19.57 | open address

Private IPv4 addresses
172.31.7.182

IPv6 address
–

Instance state
Running

Public IPv4 DNS
ec2-65-0-19-57.ap-south-1.compute.amazonaws.com | open address

Hostname type
IP name: ip-172-31-7-182.ap-south-1.compute.internal

Private IP DNS name (IPv4 only)
ip-172-31-7-182.ap-south-1.compute.internal

Step 2: Run the Python Script to Simulate CPU Spike

You can use the following Python script, which will help simulate CPU usage:
(https://github.com/deepakkesarkar8809/AWS_CloudWatchProjects.git)

```
==================================================================
=
import time

def simulate_cpu_spike(duration=30, cpu_percent=80):
    print(f"Simulating CPU spike at {cpu_percent}%...")
    start_time = time.time()

    # Calculate the number of iterations needed to achieve the desired CPU utilization
    target_percent = cpu_percent / 100
    total_iterations = int(target_percent * 5_000_000)  # Adjust the number as needed

    # Perform simple arithmetic operations to spike CPU utilization
    for _ in range(total_iterations):
        result = 0
        for i in range(1, 1001):
            result += i

    # Wait for the rest of the time interval
    elapsed_time = time.time() - start_time
    remaining_time = max(0, duration - elapsed_time)
    time.sleep(remaining_time)

    print("CPU spike simulation completed.")

if __name__ == '__main__':
    # Simulate a CPU spike for 30 seconds with 80% CPU utilization
    simulate_cpu_spike(duration=30, cpu_percent=80)
==========================================================
```

Step 3: Check Python3 Installation on EC2 Instance

Before running the script, confirm that **Python3** is installed on your EC2 instance:

```
$ python3
Python 3.12.3 (main, Sep 11 2024, 14:17:37) [GCC 13.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
```

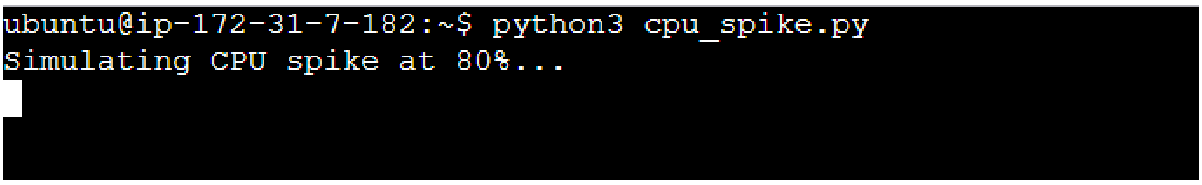
Step 4: Copy the Script to EC2 Instance Using SCP

Use the **SCP command** to transfer the Python script to your EC2 instance. Ensure that the `.pem` file is stored in your working folder:

scp -i "C:/Users/Deepak/CloudWatch_Project/Deepak-key.pem" "C:/Users/Deepak/CloudWatch_Project/cpu_spike.py" ubuntu@65.0.19.57:/home/ubuntu

Step 5: Monitor CPU Utilization in CloudWatch

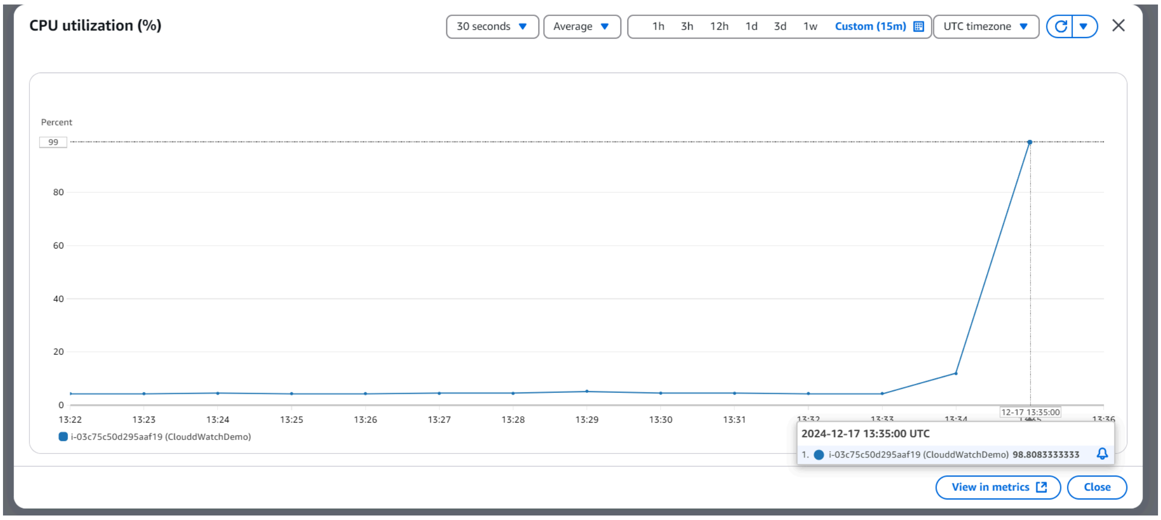
1. **After running the script, go to CloudWatch in the AWS Management Console.**
2. **In the left-hand menu, under CloudWatch, select Metrics.**
3. **Choose EC2 and then select the CPUUtilization metric to view the CPU usage for your instance.**
4. **You will now see the CPU utilization graph for your EC2 instance and can monitor the changes in real-time or over a custom time range.**



```
ubuntu@ip-172-31-7-182:~$ python3 cpu_spike.py
Simulating CPU spike at 80%...
```

Step 6: View CPU Utilization and Simulation Results

- As you run the script, the **CPU utilization will spike** to the maximum threshold you set (e.g., 80%).
  (Example of CPU spike)

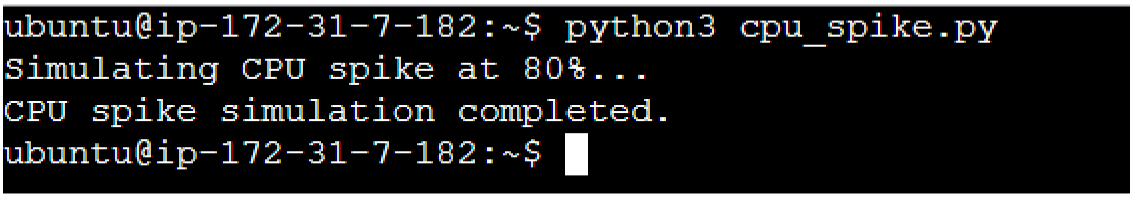


CPU utilization (%)

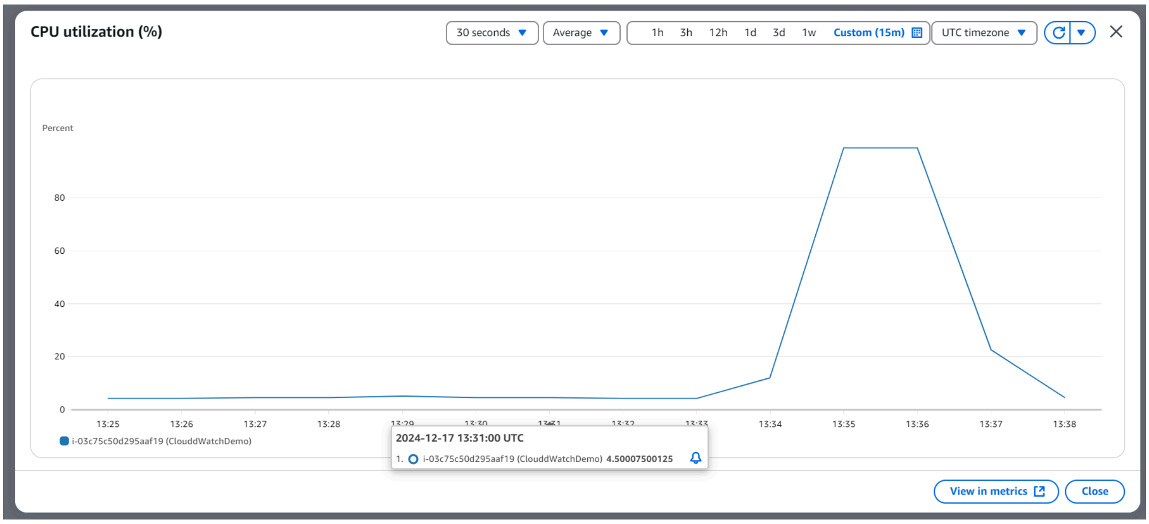
2024-12-17 13:35:00 UTC

1. ● i-03c75c50d295aaf19 (ClouddWatchDemo) 98.8083333333

- Once the simulation completes, **CPU utilization** will return to normal.
  (CPU utilization after simulation)



```
ubuntu@ip-172-31-7-182:~$ python3 cpu_spike.py
Simulating CPU spike at 80%...
CPU spike simulation completed.
ubuntu@ip-172-31-7-182:~$ 
```



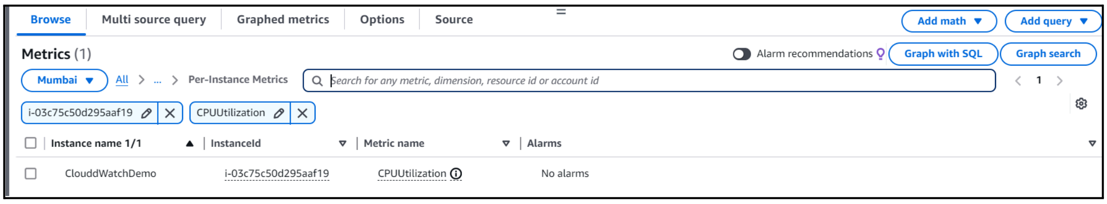
2024-12-17 13:31:00 UTC
1. ○ i-03c75c50d295aaf19 (ClouddWatchDemo) **4.50007500125**

==
====

Step 6: Set Email Alarm for CPU Utilization

Now, let's set up an email alarm to notify you when CPU utilization exceeds a specific threshold (e.g., 80%).

1. **Go to CloudWatch → Alarms → Create Alarm**.
2. **Choose the Metric**: Select **EC2 → Per-Instance Metrics → CPUUtilization**.

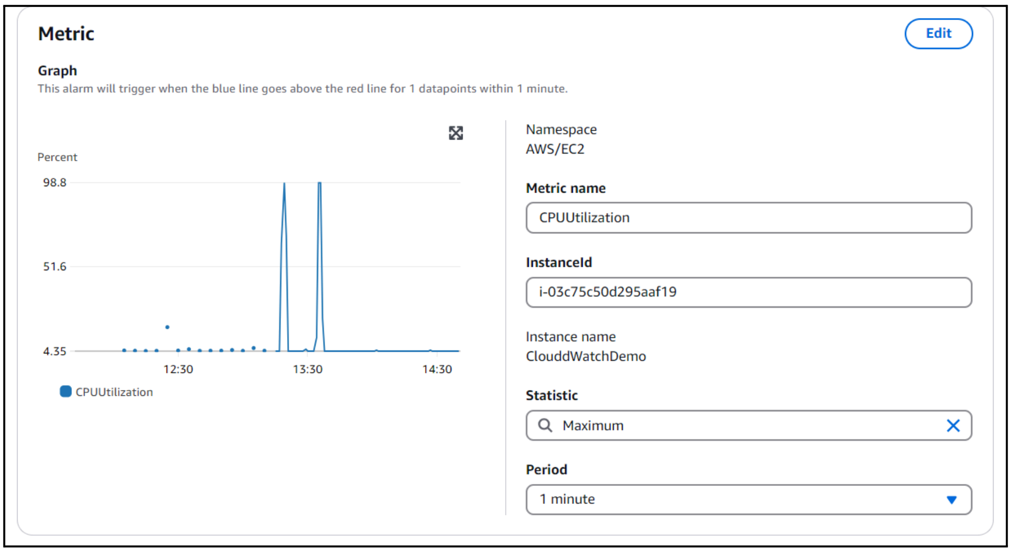


Metrics (1)

i-03c75c50d295aaf19 ✕ CPUUtilization ✕

□	Instance name 1/1	InstanceId	Metric name	Alarms
□	ClouddWatchDemo	i-03c75c50d295aaf19	CPUUtilization ⓘ	No alarms

3. **Set the Threshold**:
   - Set the **CPU utilization** threshold to **80%** or the desired value.
   - Choose the **"Greater than/equal to"** condition and set the threshold value.



Metric

Graph

This alarm will trigger when the blue line goes above the red line for 1 datapoints within 1 minute.

Namespace
AWS/EC2

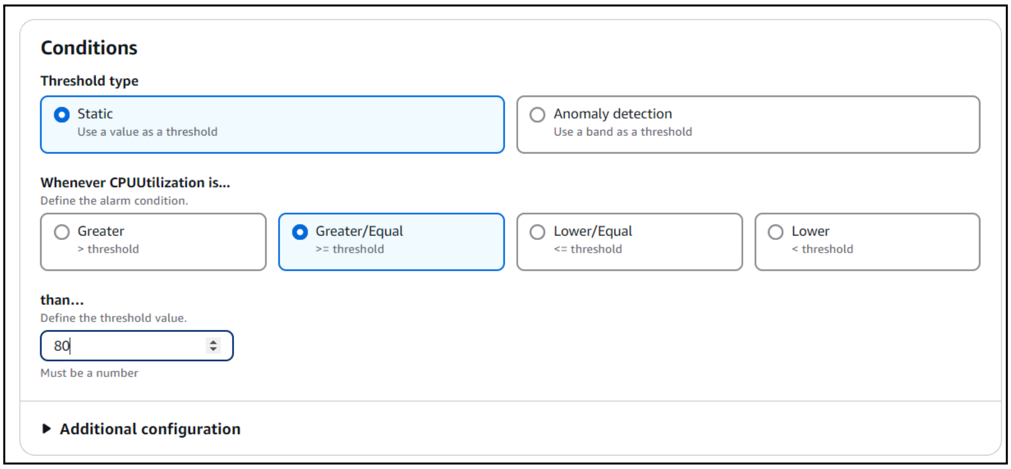
Metric name
CPUUtilization

InstanceId
i-03c75c50d295af19

Instance name
ClouddWatchDemo

Statistic
Maximum

Period
1 minute



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

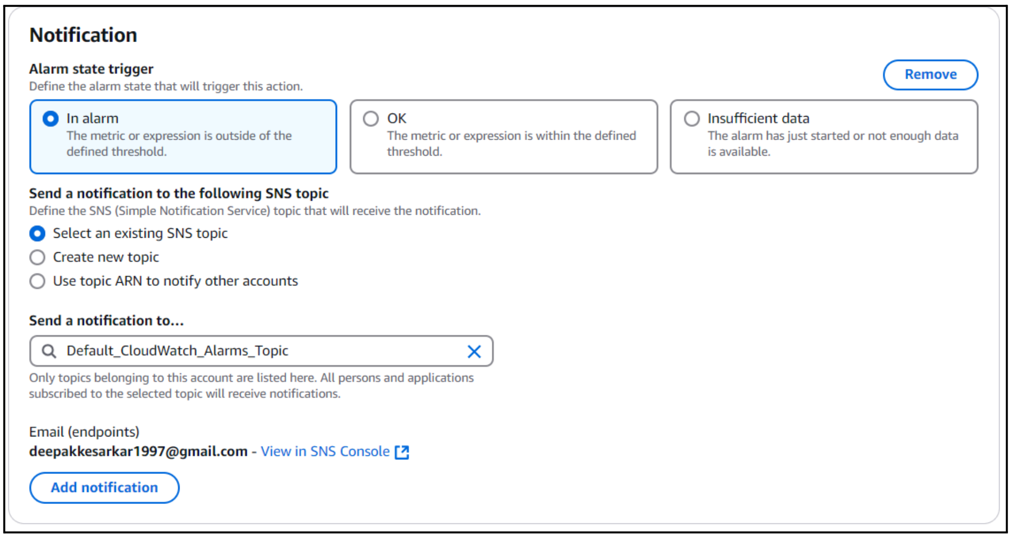
than...
Define the threshold value.

80

Must be a number

4. **Set Actions**:
   - Under **Actions**, choose **New list** and then select **Send a notification**.
   - Choose an SNS **topic** or create a new one to send the email.

5. **Create SNS Topic**:
   - In **SNS**, subscribe to the topic with your email address to receive alerts.
   - Once confirmed, AWS will send a notification email to you when the CPU utilization crosses the threshold, just confirm the over mail so you will get mail from AWS.



Notification

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.

Send a notification to the following SNS topic

Define the SNS (Simple Notification Service) topic that will receive the notification.

● Select an existing SNS topic
○ Create new topic
○ Use topic ARN to notify other accounts

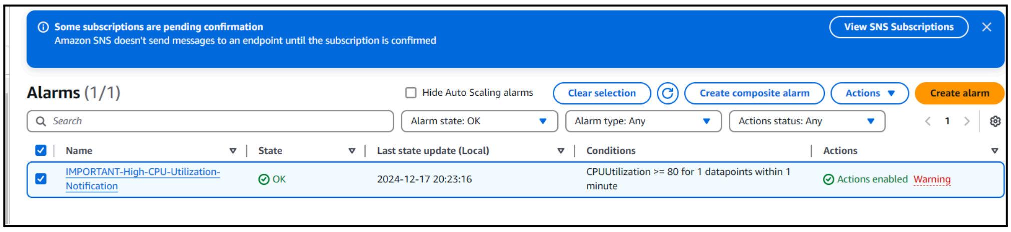
Send a notification to...

Default_CloudWatch_Alarms_Topic

Only topics belonging to this account are listed here. All persons and applications subscribed to the selected topic will receive notifications.

Email (endpoints)
deepakkesarkar1997@gmail.com - View in SNS Console

6. **Create Alarm**:
   - Review the settings and click **Create Alarm**.
   - Alarm now activated by AWS.



Some subscriptions are pending confirmation
Amazon SNS doesn't send messages to an endpoint until the subscription is confirmed

Alarms (1/1)

□ Hide Auto Scaling alarms

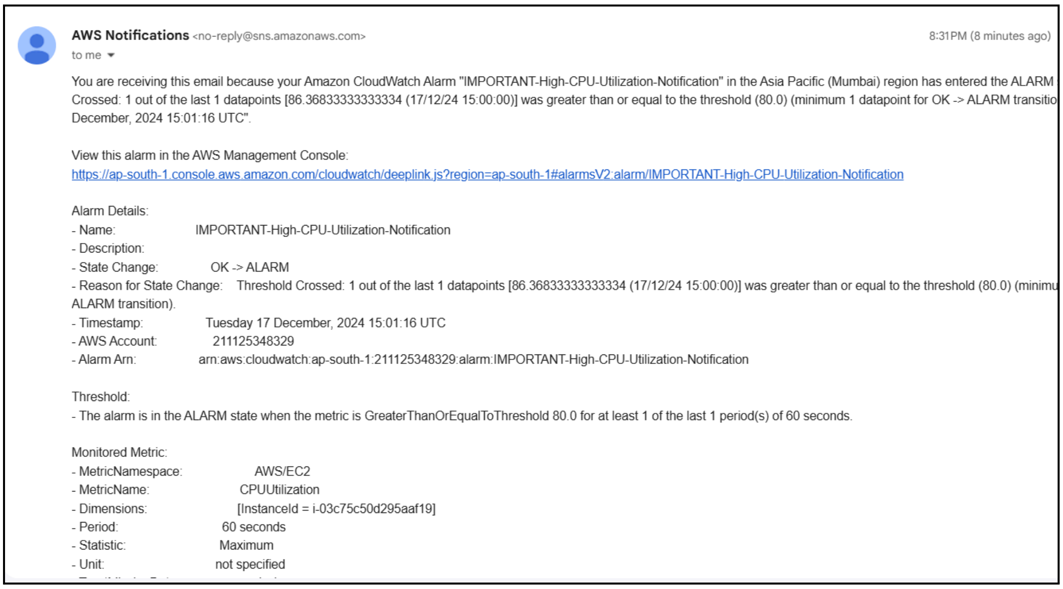
Alarm state: OK
Alarm type: Any
Actions status: Any

☑	Name	State	Last state update (Local)	Conditions	Actions
☑	IMPORTANT-High-CPU-Utilization-Notification	✓ OK	2024-12-17 20:23:16	CPUUtilization >= 80 for 1 datapoints within 1 minute	✓ Actions enabled Warning

---

Step 7: View CPU Utilization and Email Notification

- As you run the script, the **CPU utilization will spike** to 80% (or your chosen threshold).
  (CPU spike example)
- Once the CPU utilization exceeds the set threshold, the **alarm will trigger**, and you'll receive an **email notification**.
- Once the simulation completes, **CPU utilization** will return to normal.
  (Normal CPU utilization after simulation).



AWS Notifications <no-reply@sns.amazonaws.com> 8:31PM (8 minutes ago)

to me ▾

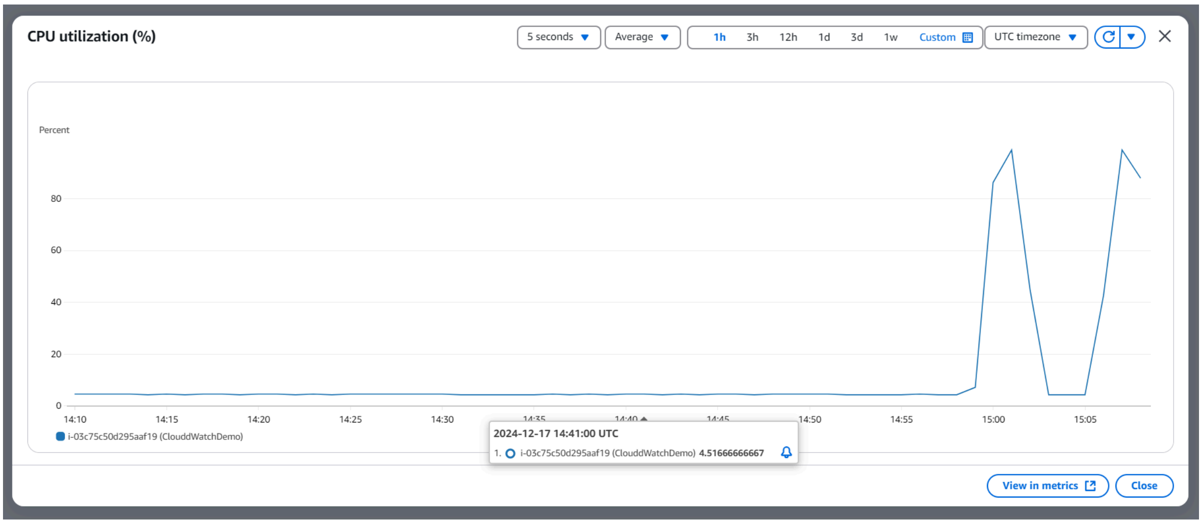
You are receiving this email because your Amazon CloudWatch Alarm "IMPORTANT-High-CPU-Utilization-Notification" in the Asia Pacific (Mumbai) region has entered the ALARM
Crossed: 1 out of the last 1 datapoints [86.36833333333334 (17/12/24 15:00:00)] was greater than or equal to the threshold (80.0) (minimum 1 datapoint for OK -> ALARM transitio
December, 2024 15:01:16 UTC".

View this alarm in the AWS Management Console:
https://ap-south-1.console.aws.amazon.com/cloudwatch/deeplink.js?region=ap-south-1#alarmsV2:alarm/IMPORTANT-High-CPU-Utilization-Notification

Alarm Details:
- Name: IMPORTANT-High-CPU-Utilization-Notification
- Description:
- State Change: OK -> ALARM
- Reason for State Change: Threshold Crossed: 1 out of the last 1 datapoints [86.36833333333334 (17/12/24 15:00:00)] was greater than or equal to the threshold (80.0) (minimu
ALARM transition).
- Timestamp: Tuesday 17 December, 2024 15:01:16 UTC
- AWS Account: 211125348329
- Alarm Arn: arn:aws:cloudwatch:ap-south-1:211125348329:alarm:IMPORTANT-High-CPU-Utilization-Notification

Threshold:
- The alarm is in the ALARM state when the metric is GreaterThanOrEqualToThreshold 80.0 for at least 1 of the last 1 period(s) of 60 seconds.

Monitored Metric:
- MetricNamespace: AWS/EC2
- MetricName: CPUUtilization
- Dimensions: [InstanceId = i-03c75c50d295aaf19]
- Period: 60 seconds
- Statistic: Maximum
- Unit: not specified



CPU utilization (%)

5 seconds ▾ | Average ▾ | 1h | 3h | 12h | 1d | 3d | 1w | Custom | UTC timezone ▾

2024-12-17 14:41:00 UTC
1. ○ i-03c75c50d295aaf19 (ClouddWatchDemo) 4.51666666667

Source for this Tutorial

Day-16 | AWS CLOUD WATCH DEEP DIVE | DEMO - LIVE EC2 CPU ALERTI…