



**St. JOSEPH'S**  
GROUP OF INSTITUTIONS  
OMR, CHENNAI - 119



## Placement Empowerment Program

*Cloud Computing and DevOps Centre*

Set Up a Cloud-Based Monitoring Service Enable basic cloud monitoring (e.g., Cloud Watch on AWS). View metrics like CPU usage and disk I/O for your cloud VM.

Name: Samiya Afreen J

Department: CSE

# **Introduction:**

In cloud computing, effective monitoring is crucial for ensuring the performance, reliability, and availability of cloud resources. **AWS CloudWatch** provides a comprehensive monitoring solution for AWS resources, enabling users to track various metrics in real-time. This Proof of Concept (PoC) focuses on leveraging **CloudWatch** to monitor the performance of an EC2 instance by enabling basic monitoring for key metrics such as **CPU utilization** and **disk I/O**. This PoC demonstrates how to enable, view, and analyze these metrics, giving insights into the health and performance of cloud-based virtual machines.

# **Overview:**

This PoC will walk through the process of setting up **AWS CloudWatch** to monitor an EC2 instance. The main steps include:

1. Enabling basic cloud monitoring for an EC2 instance.
2. Viewing key metrics such as **CPU utilization** and **disk read/write operations**, to assess the performance of the instance.
3. Exploring how CloudWatch provides real-time insights into the instance's resource usage, allowing administrators to identify performance bottlenecks or issues before they affect the service.

By completing this PoC, users will understand how to integrate CloudWatch monitoring for EC2 instances, enabling effective performance monitoring of virtual machines in the cloud.

# **Objective:**

The primary objective of this PoC is to enable **basic cloud monitoring** using **AWS CloudWatch** and view essential metrics for an EC2 instance. Specific goals include:

**Enabling CloudWatch monitoring** for an EC2 instance.

**Viewing CPU usage and disk I/O** metrics to analyze the instance's performance.

Understanding how CloudWatch helps in real-time monitoring by providing visibility into cloud resource health.

# **Importance of this PoC:**

- 1. Performance Monitoring:** By tracking **CPU usage, disk I/O, and network traffic**, CloudWatch provides crucial insights into the resource utilization of an EC2 instance, which helps in identifying and troubleshooting performance issues.
- 2. Real-time Visibility:** Enabling CloudWatch monitoring ensures that administrators have access to real-time data about the instance's performance. This allows quick reactions to changes in resource consumption, preventing downtime or service degradation.
- 3. Resource Management:** Understanding the resource consumption of the EC2 instance (such as CPU usage and disk I/O) helps in optimizing the instance's capacity and managing resources efficiently, which can also lead to cost savings.
- 4. Proactive Issue Detection:** CloudWatch allows the user to monitor and understand patterns in the system's resource usage, helping detect performance anomalies or bottlenecks before they impact the system.

# Step-by-Step Overview

## Step 1:

1. Go to [AWS Management Console](#).
2. Enter your username and password to log in.

The screenshot shows the AWS Management Console Home page. At the top, there's a navigation bar with the AWS logo, a search bar, and account information for 'Asia Pacific (Mumbai)' and 'Samiya'. Below the navigation is a header bar with links for 'Simple Notification Service' and other services like CloudWatch, Lightsail, Simple Queue Service, Simple Notification Service, and RDS. The main area is divided into several cards:

- Recently visited:** IAM, EC2, S3, Billing and Cost Management, VPC, Amazon Comprehend, CloudFront, CloudFormation.
- Applications:** Shows 0 applications. It includes a 'Create application' button and a note: 'Get started by creating an application.'
- Welcome to AWS:** Includes sections for 'Getting started with AWS' (with a link to the documentation), 'Training and Development', and 'Feedback'.
- AWS Health:** Shows 0 open issues (Past 7 days) and 0 scheduled changes (Upcoming and past 7 days).
- Cost and usage:** Shows current month costs at \$3.16 and forecasted month end costs at \$3.52. A bar chart visualizes the cost distribution.

At the bottom, there are links for 'View all services', 'Go to myApplications', and various legal links like 'CloudShell', 'Feedback', 'Privacy', 'Terms', and 'Cookie preferences'.

## Step 2:

On the EC2 Dashboard, click on **Launch Instances** and enter a name for your instance (e.g., "My Monitoring Instance"). Leave other settings as default and Click **Launch Instance**.

The screenshot shows the EC2 Dashboard. On the left, there's a sidebar with links for Services, Features, Resources, Documentation, Knowledge articles, Marketplace, Blog posts, Events, and Tutorials. The main content area is titled 'Services' and lists three items:

- EC2**: Virtual Servers in the Cloud
- EC2 Image Builder**: A managed service to automate build, customize and deploy OS images
- EC2 Global View**: EC2 Global View provides a global dashboard and search functionality that lets you ...

Below this, there's a section titled 'Features' with three items:

- Dashboard**: EC2 feature
- AMIs**: EC2 feature
- EC2 Instances**: CloudWatch feature

At the bottom, there's a section titled 'Resources' with a note: 'Introducing resource search' and a sub-note: 'Enable to show cross-region resources for your account in search results. Takes less'. There's also a 'Feedback' link at the bottom right.

The screenshot shows the AWS EC2 Instances page. On the left, there's a navigation menu with sections like EC2, Instances, Images, Elastic Block Store, Network & Security, and more. The main area displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
my web server	i-005c3524seer4e888u	Running	t3.micro	3/3 checks passed	View alarms	ap-south-1c
Virtual-Machine	i-0f7f9c91bdbbc7150	Running	t3.micro	3/3 checks passed	View alarms	ap-south-1c
My Monitoring...	i-0c8b29b2f4eb58880	Running	t3.micro	Initializing	View alarms	ap-south-1c

## Step 3:

**Go to the EC2 Dashboard in the AWS Console.**

In the left menu, click **Volumes** under **Elastic Block Store (EBS)**.

Click **Create Volume**.

The screenshot shows the 'Create volume' wizard in the AWS EBS Volumes page. It has several steps:

- Volume settings**:
  - Volume type**: General Purpose SSD (gp3)
  - Size (GiB)**: 100
  - IOPS**: 3000
  - Throughput (MiB/s)**: 125
  - Availability Zone**: ap-south-1a
  - Snapshot ID - optional**: Don't create volume from a snapshot
  - Encryption**: Use Amazon EBS encryption as an encryption solution for your EBS resources associated with your EC2 instances. (checkbox checked)
  - Encrypt this volume**: (checkbox checked)

# Step 4:

Once created, go to your **Volumes** list, select the newly created volume, and click **Actions > Attach Volume**.

The screenshot shows the AWS EC2 Volumes details page for a volume named 'vol-0dc568333342ddfed'. The volume has a size of 100 GB, type gp3, and IOPS of 3000. It was created on Wednesday, Feb 26 2025 at 10:48:44 GMT+0530 (India Standard Time). The volume is attached to an instance and is managed by an operator. The Actions menu is open, showing options such as Create snapshot, Attach volume, Detach volume, Force detach volume, and Manage auto-enabled I/O.

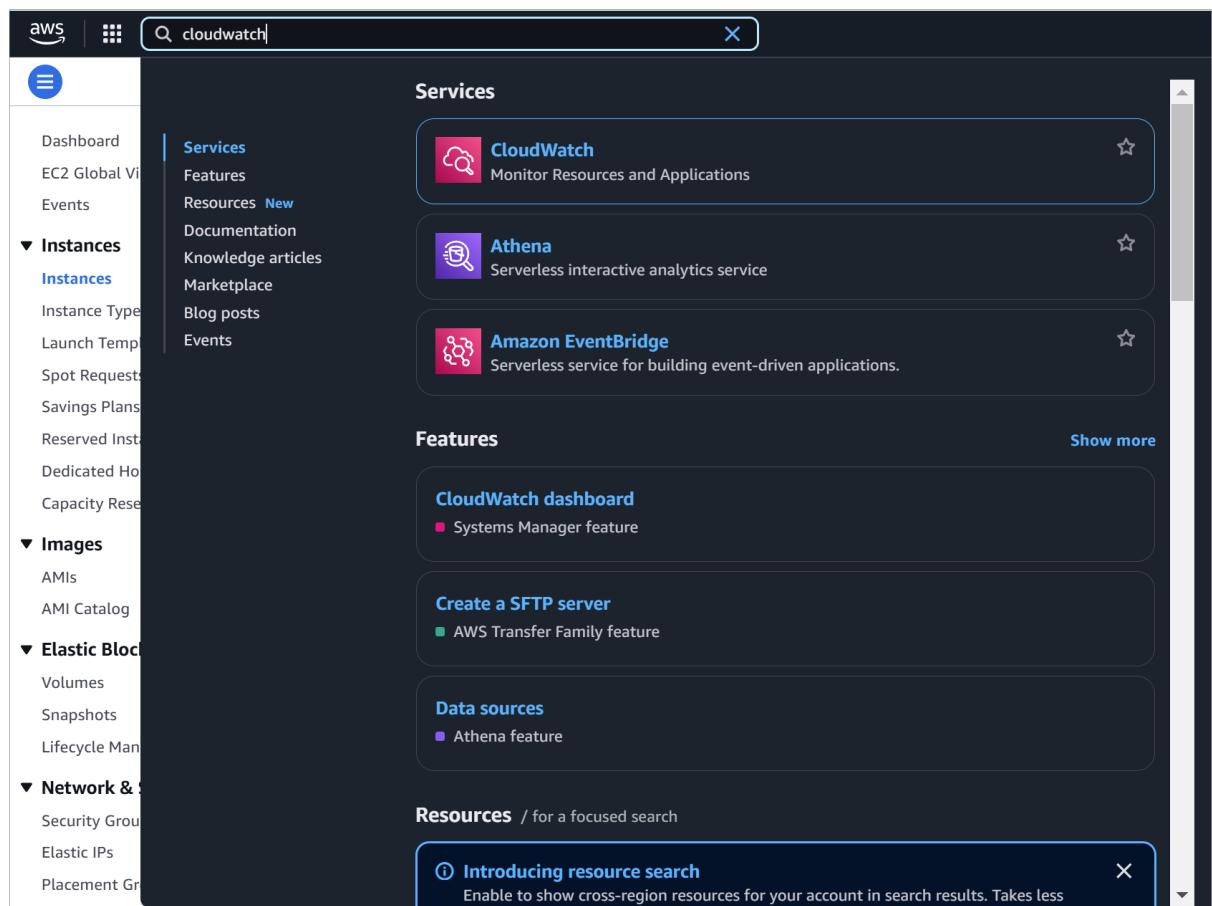
The screenshot shows the 'Attach volume' dialog box. It requires selecting an instance and a device name. The selected instance is 'i-0c8b29b2f4eb5880' (My Monitoring Instance) and the device name is '/dev/sdb'. A note states: 'Newer Linux kernels may rename your devices to /dev/xvdf through /dev/xvdp internally, even when the device name entered here (and shown in the details) is /dev/sdf through /dev/sdp.' There are 'Cancel' and 'Attach volume' buttons at the bottom.

## Step 5:

On the AWS Console homepage, look for the search bar at the top.

Type **CloudWatch** in the search bar and press **Enter**.

From the search results, click on **CloudWatch**.

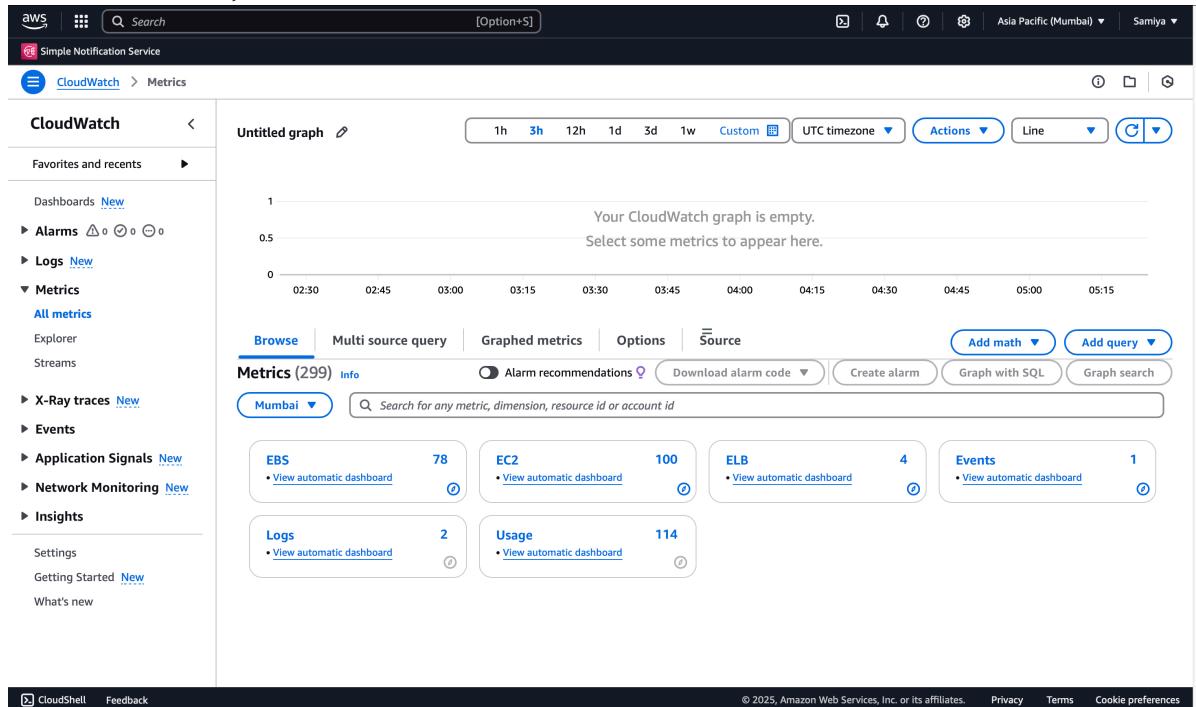


# Step 6:

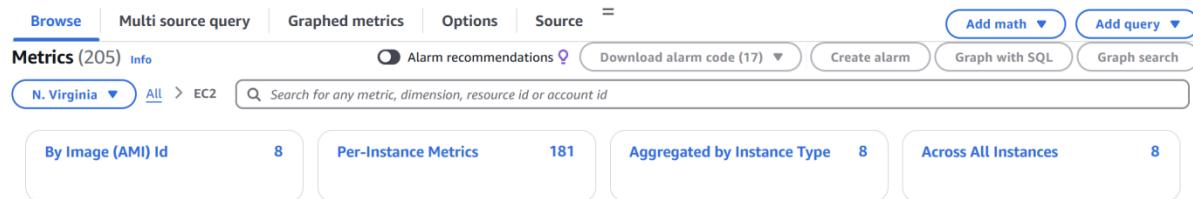
In the CloudWatch dashboard, look at the left-hand menu.

Click on **Metrics**.

Under **Browse**, click on **EC2**.



Then click on the **Per-Instance Metrics**.



# Step 7:

You should now see a list of metrics for all your EC2 instances, such as:

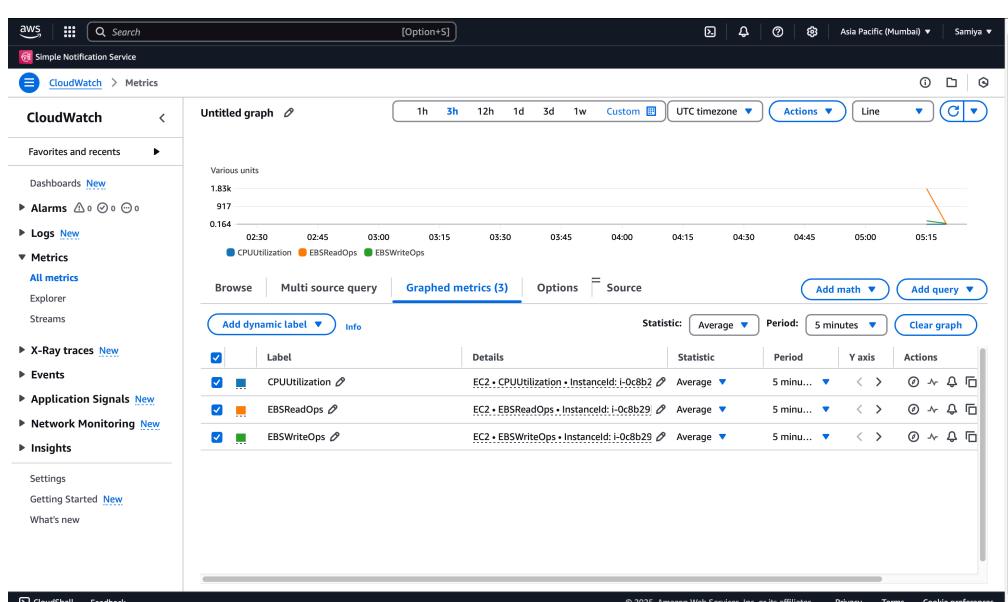
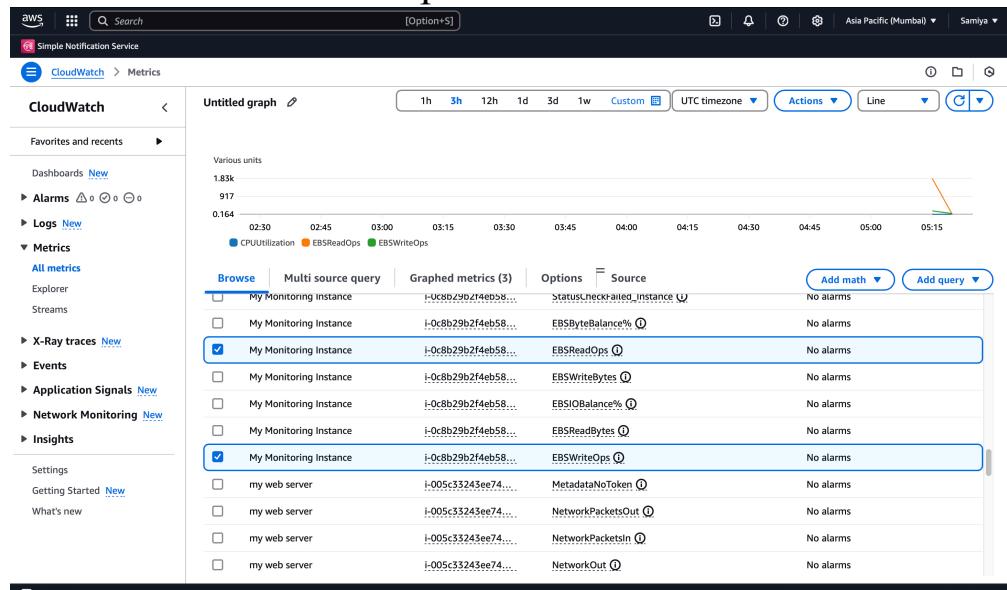
**CPUUtilization** (CPU usage)

**DiskReadOps / DiskWriteOps** (Disk I/O)

Identify the specific EC2 instance you want to monitor (it will be listed by its instance ID).

Click on the metrics associated with your instance

To view detail click Graphed metrics



# Outcome

This Proof of Concept (PoC) aimed to establish a **cloud-based monitoring service** using **AWS CloudWatch** to track key performance metrics for an EC2 instance, specifically focusing on **CPU utilization** and **Disk I/O** (DiskReadOps and DiskWriteOps).

Here's the outcome of the PoC:

1. **CloudWatch Setup:** Successfully configured AWS **CloudWatch** to monitor EC2 instance metrics like **CPU utilization** and **Disk I/O** (DiskReadOps, DiskWriteOps).
2. **Disk I/O Monitoring:** Added an **EBS volume** to the EC2 instance to track **DiskReadOps** and **DiskWriteOps** metrics, which were visualized in CloudWatch.
3. **Cost Efficiency:** The EBS volume was within the **AWS Free Tier** limits (30 GB), and all metrics stayed within **Free Tier** usage, incurring no additional cost.