



## **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: KEERTHIKA ANANTHAN

Department: AML



## 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 cloudbased 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.

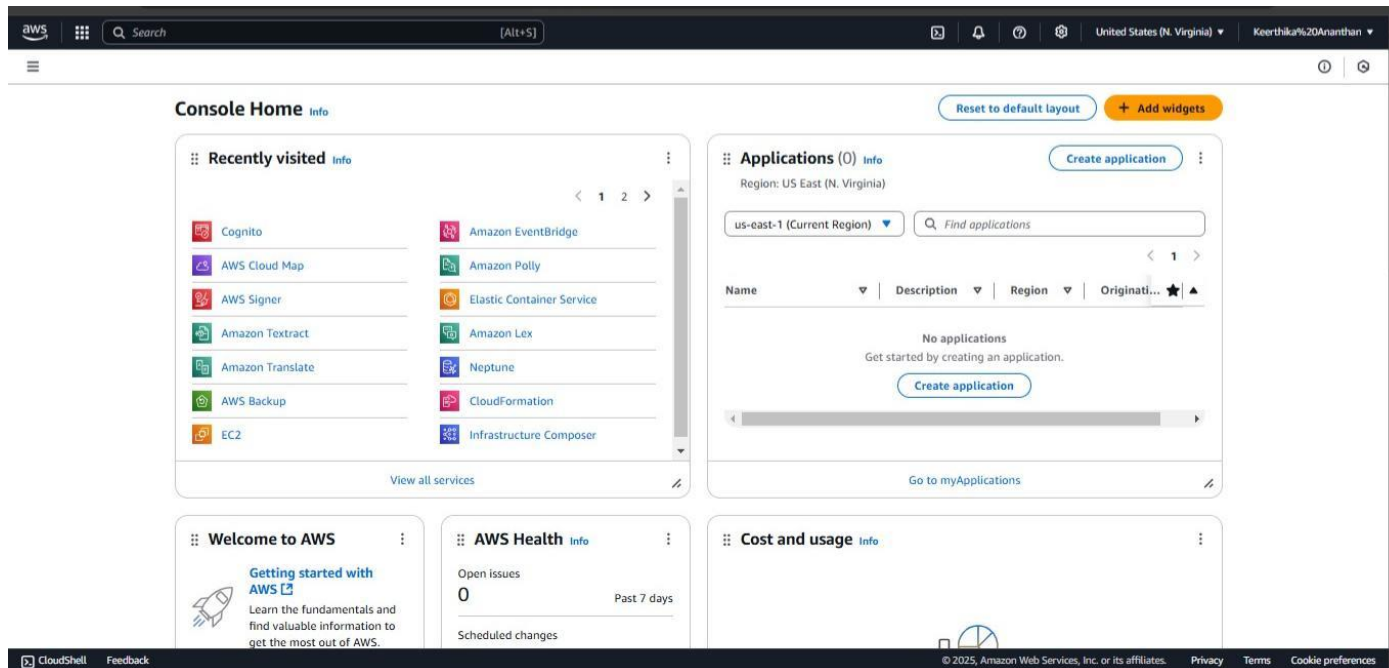
**3. Resource Management:** Understanding the resource consumption of the EC2 instance (such as CPU usage and disk I/O)

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.

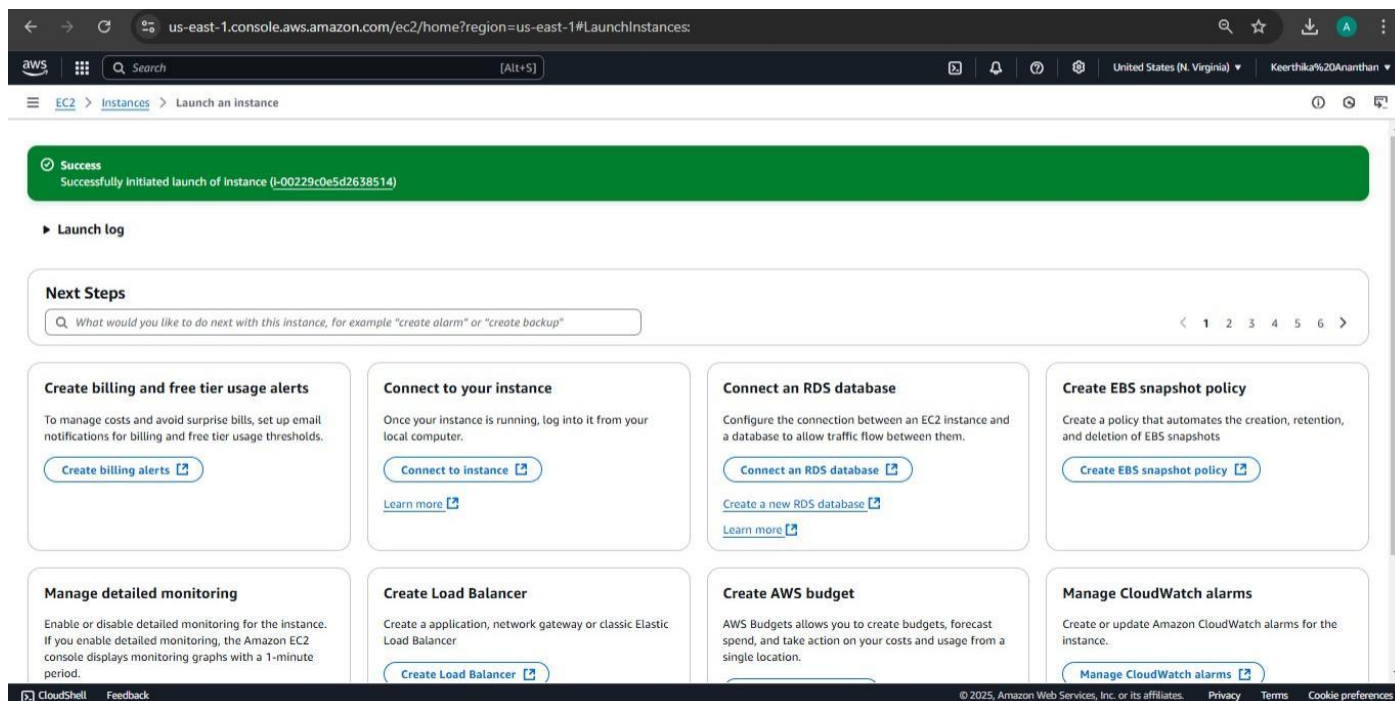
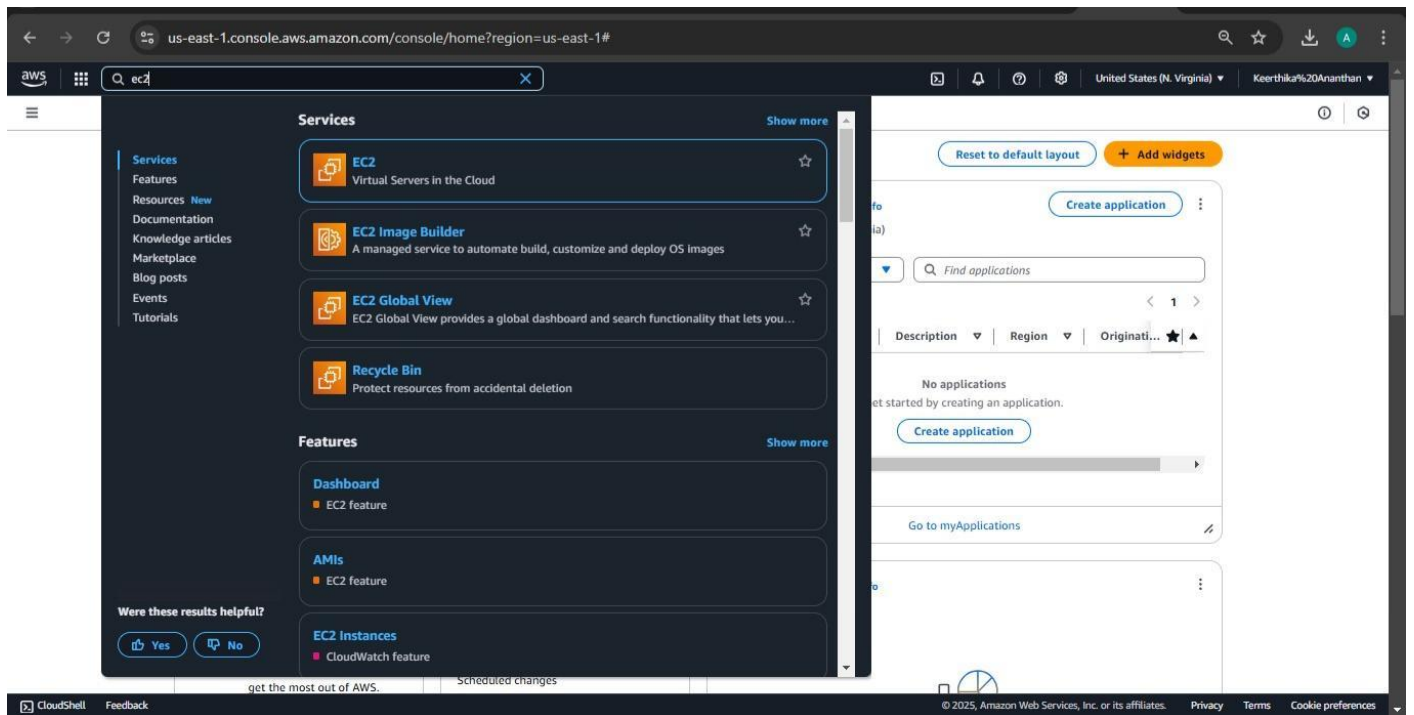
## Step-by-Step Overview Step

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



## 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**.



**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 AWS Management Console interface for creating a new EBS volume. The top navigation bar includes the AWS logo, a search bar, and user information. The breadcrumb trail indicates the path: EC2 > Volumes > Create volume. The main heading is 'Create volume' with an 'Info' link. Below this is a sub-header 'Volume settings' and a descriptive text: 'Create an Amazon EBS volume to attach to any EC2 instance in the same Availability Zone.' The settings are organized into sections: 'Volume type' (General Purpose SSD (gp3)), 'Size (GiB)' (100), 'IOPS' (3000), 'Throughput (MiB/s)' (125), 'Availability Zone' (us-east-1a), and 'Snapshot ID - optional' (Don't create volume from a snapshot). Each input field has an 'Info' link. At the bottom, there is an 'Encryption' section with a note about using Amazon EBS encryption. The footer contains links for CloudShell, Feedback, and copyright information.

**Create volume** [Info](#)

Create an Amazon EBS volume to attach to any EC2 instance in the same Availability Zone.

**Volume settings**

**Volume type** [Info](#)

General Purpose SSD (gp3)

**Size (GiB)** [Info](#)

100

Min: 1 GiB, Max: 16384 GiB.

**IOPS** [Info](#)

3000

Min: 3000 IOPS, Max: 16000 IOPS.

**Throughput (MiB/s)** [Info](#)

125

Min: 125 MiB, Max: 1000 MiB. Baseline: 125 MiB/s.

**Availability Zone** [Info](#)

us-east-1a

**Snapshot ID - optional** [Info](#)

Don't create volume from a snapshot

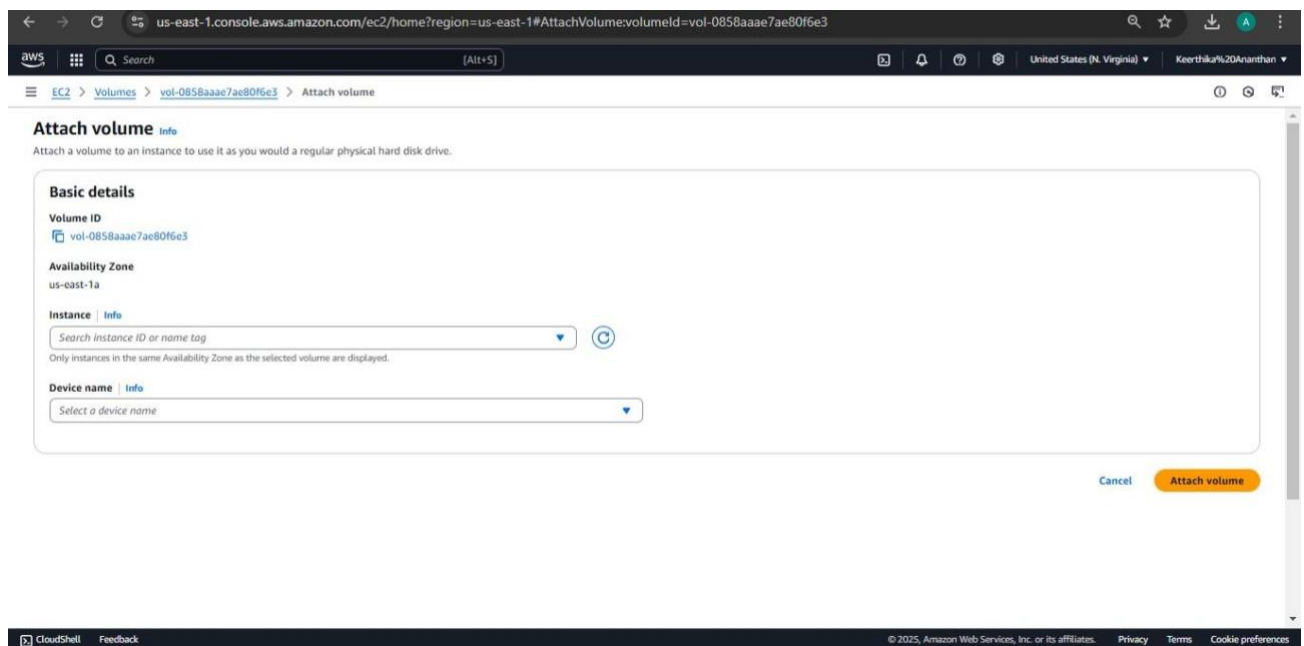
**Encryption** [Info](#)

Use Amazon EBS encryption as an encryption solution for your EBS resources associated with your EC2 instances.

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## Step 4:

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

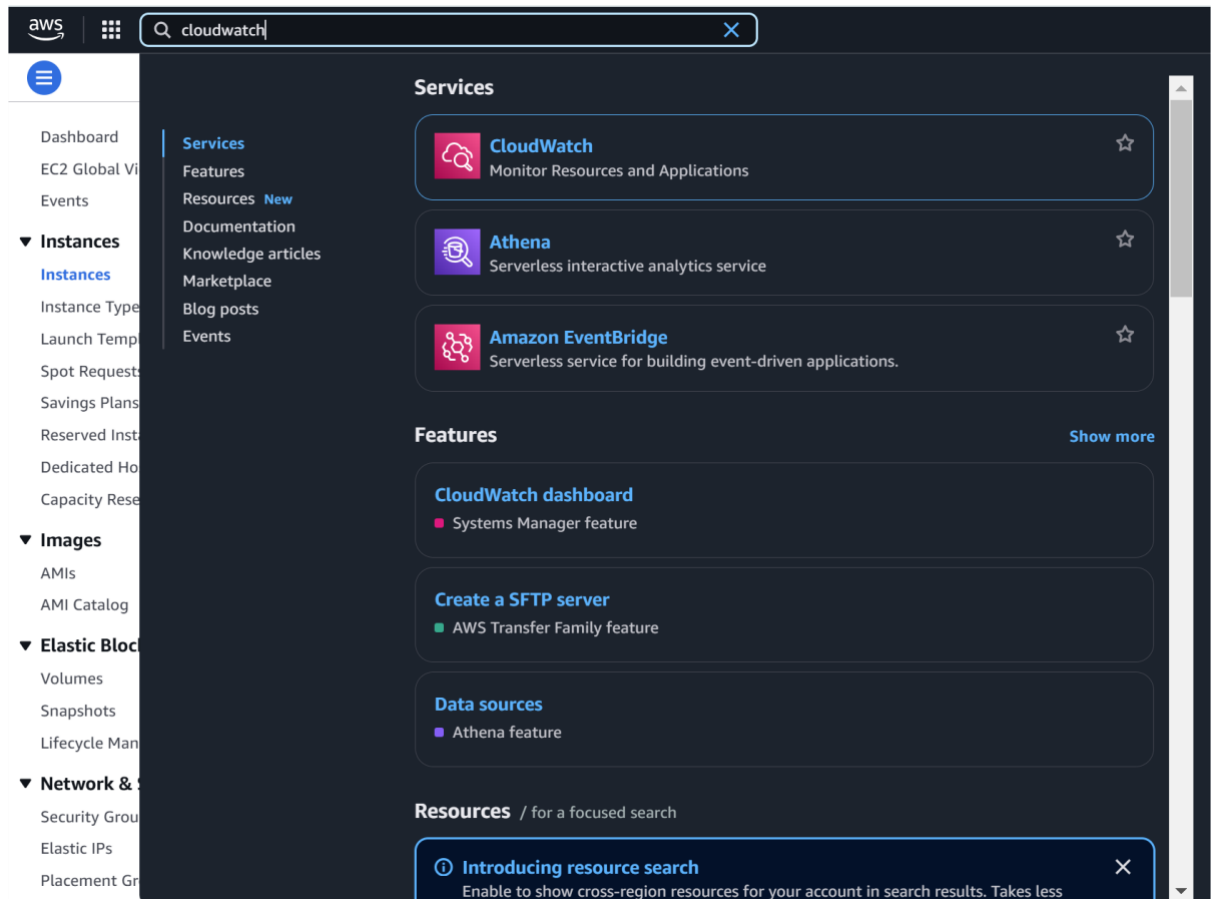


On the AWS Console homepage, look for the search bar at the top.

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

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

## Step 5:



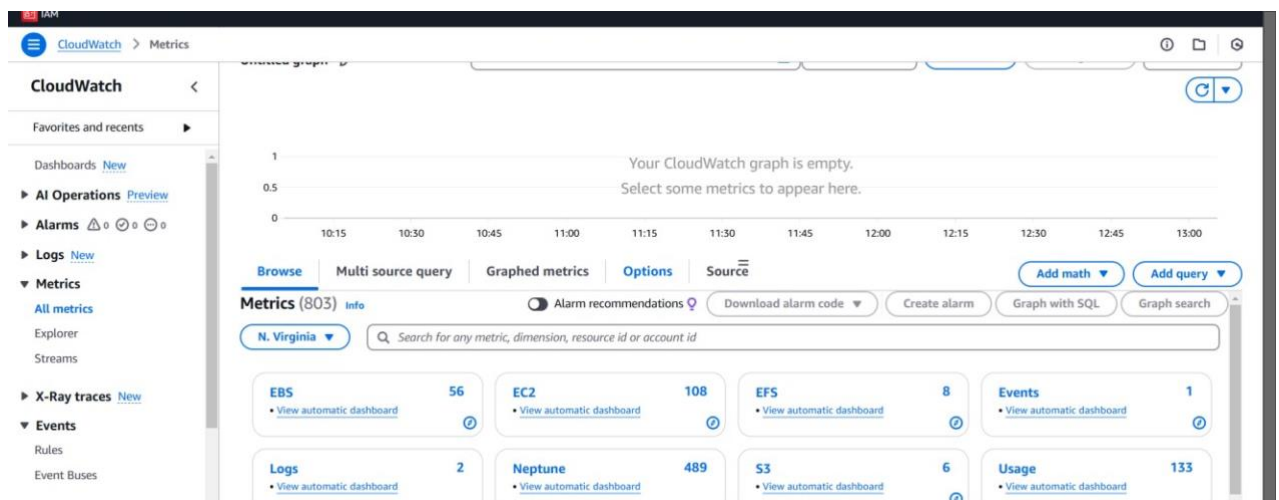
In the CloudWatch dashboard, look at the left-hand menu.

Click on **Metrics**.

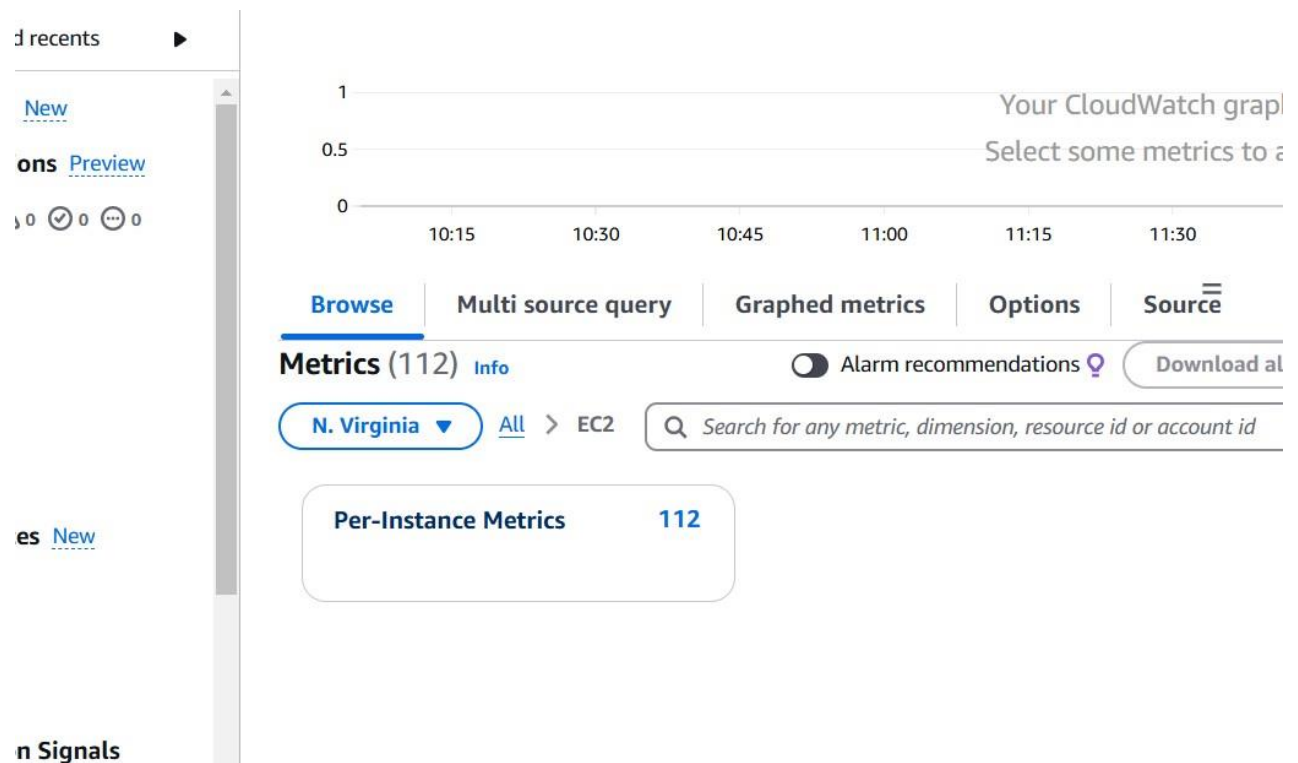
Under **Browse**, click on **EC2**.



## Step 6:



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).

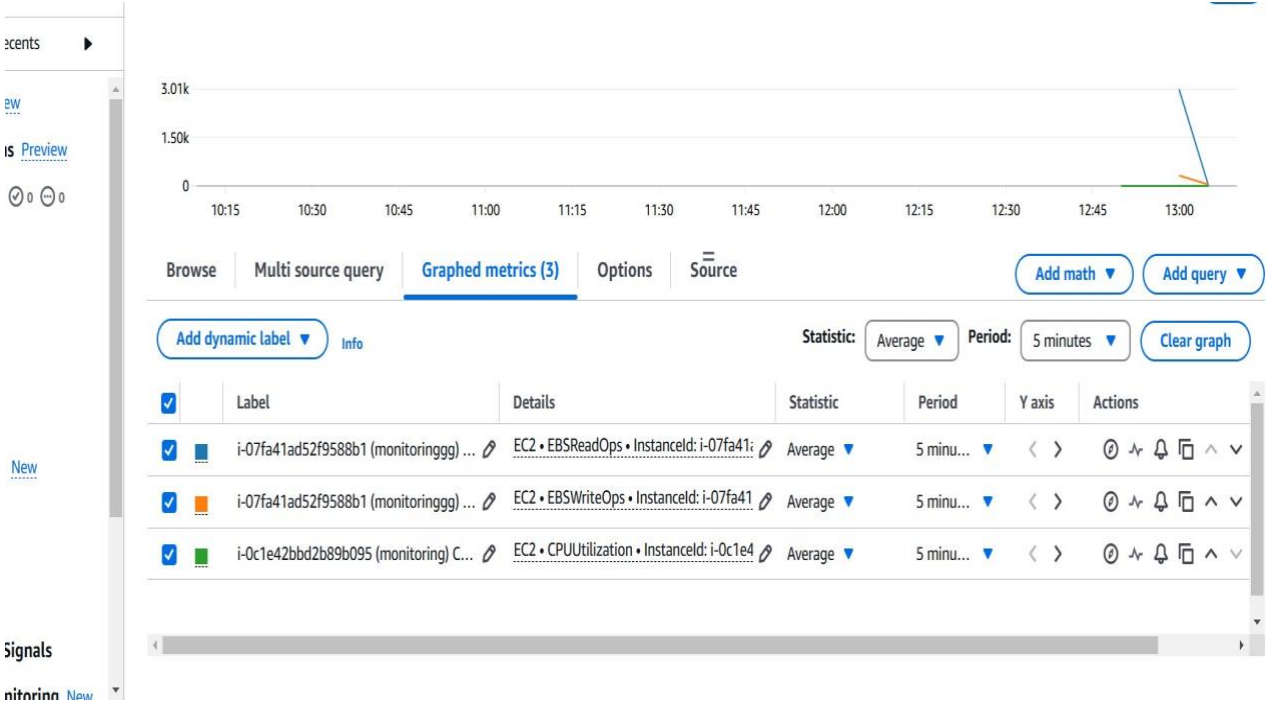
Click on the metrics associated with your instance

To view detail click Graphed metrics

|                                     |              |                     |                            |           |
|-------------------------------------|--------------|---------------------|----------------------------|-----------|
| <input type="checkbox"/>            | monitoringgg | i-U7fa41ad52f958... | statusCheckFailed_Instance | No alarms |
| <input checked="" type="checkbox"/> | monitoringgg | i-07fa41ad52f958... | EBSWriteOps ⓘ              | No alarms |
| <input checked="" type="checkbox"/> | monitoringgg | i-07fa41ad52f958... | EBSReadOps ⓘ               | No alarms |
| <input type="checkbox"/>            | monitoringgg | i-07fa41ad52f958... | EBSIOBalance% ⓘ            | No alarms |

|                                     |            |                    |                  |      |
|-------------------------------------|------------|--------------------|------------------|------|
| <input checked="" type="checkbox"/> | monitoring | i-0c1e42bbd2b89... | CPUUtilization ⓘ | No a |
|-------------------------------------|------------|--------------------|------------------|------|

# Step 8:



# 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.