



## **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: Vishnu K**

**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.

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.

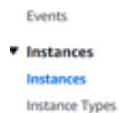
Understanding how CloudWatch helps in real-time monitoring by providing visibility into cloud resource health.

## 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

### Step 1:

3. On the EC2 Dashboard, click on **Launch Instances** and enter a name for your instance. Leave other settings as default and Click **Launch Instance**.

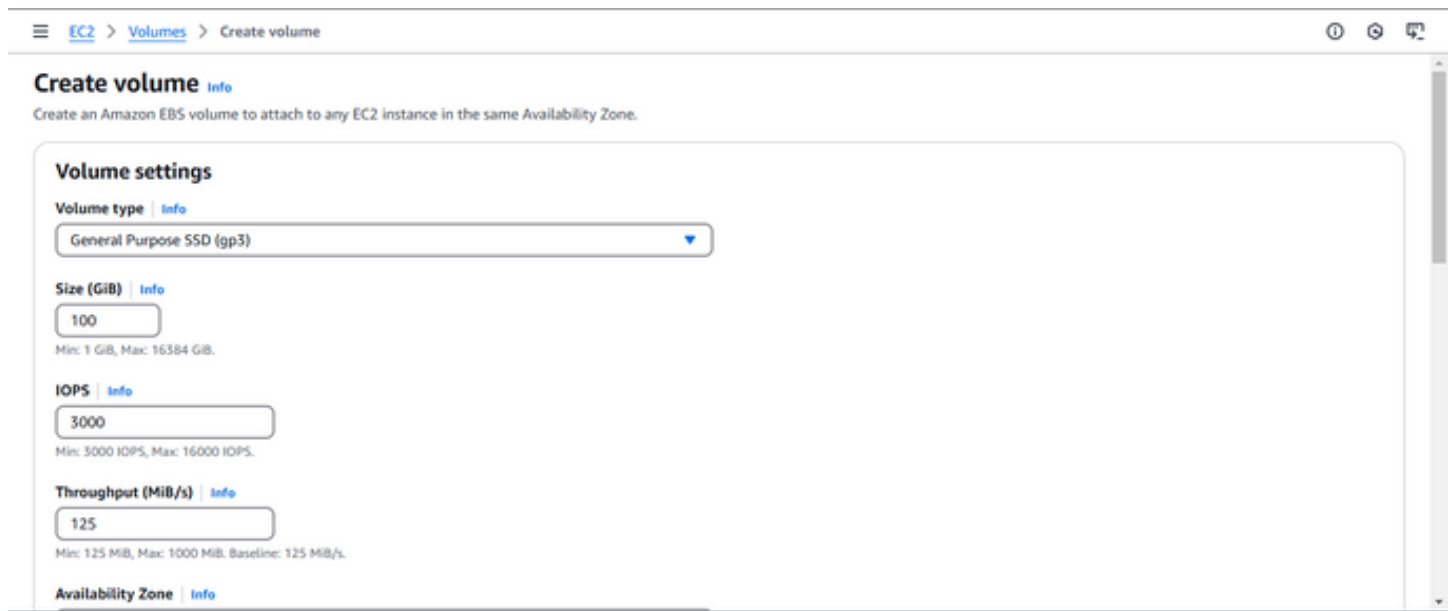


<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
<input type="checkbox"/>	krish ec server	i-017f1f541d6cc12b9	Terminated	t2.micro	-	<a href="#">View alarms +</a>	us-east-1b	-
<input checked="" type="checkbox"/>	sampleinstance	i-07b427e21e865c6f9	Running	t2.micro	2/2 checks passed	<a href="#">View alarms +</a>	us-east-1b	ec2-3-21

## Step 2:

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' page in the AWS Management Console. The breadcrumb navigation at the top reads 'EC2 > Volumes > Create volume'. The page title is 'Create volume' with an 'Info' link. Below the title is a subtitle: 'Create an Amazon EBS volume to attach to any EC2 instance in the same Availability Zone.' The main content area is titled 'Volume settings' and contains four sections: 'Volume type' with a dropdown menu set to 'General Purpose SSD (gp3)', 'Size (GiB)' with a text input set to '100' and a range of 'Min: 1 GiB, Max: 16384 GiB', 'IOPS' with a text input set to '3000' and a range of 'Min: 3000 IOPS, Max: 16000 IOPS', and 'Throughput (MiB/s)' with a text input set to '125' and a range of 'Min: 125 MiB, Max: 1000 MiB. Baseline: 125 MiB/s'. At the bottom, there is an 'Availability Zone' section with an 'Info' link.

## Step 3:

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

EC2 > Volumes > vol-0f1ed2f8d493ef4aa

Reserved Instances  
Dedicated Hosts  
Capacity Reservations

▼ Images  
AMIs  
AMI Catalog

▼ Elastic Block Store  
Volumes  
Snapshots  
Lifecycle Manager

▼ Network & Security  
Security Groups  
Elastic IPs  
Placement Groups  
Key Pairs  
Network Interfaces

▼ Load Balancing

**vol-0f1ed2f8d493ef4aa**

Volume ID  
vol-0f1ed2f8d493ef4aa

Size  
100 GiB

Type  
gp3

IOPS  
3000

Volume state  
Available

Fast snapshot restored  
No

Availability Zone  
us-east-1a

Created  
Mon Feb 03 2025 17:55:49 GMT+0530 (India Standard Time)

Managed  
false

Multi-Attach enabled  
No

Operator  
-

Actions  
Delete  
Modify

Create snapshot  
Attach volume  
Detach volume  
Force detach volume  
Manage auto-enabled I/O

Attached resources  
-

Outposts ARN  
-

▼ Source  
Snapshot ID  
-

▼ Encryption  
Encryption  
KMS key ID  
KMS key alias  
KMS key ARN

CloudShell Feedback

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

EC2 > Volumes > vol-01ac5577ef749d37b > Attach volume

**Attach volume** Info

Attach a volume to an instance to use it as you would a regular physical hard disk drive.

**Basic details**

Volume ID  
vol-01ac5577ef749d37b

Availability Zone  
us-east-1b

Instance Info  
i-07b427e21e865c6f9 (sampleinstance) (running)

Only instances in the same Availability Zone as the selected volume are displayed.

Device name Info  
/dev/sdb

Recommended device names for Linux: /dev/xvda for root volume, /dev/sd[f-p] for data volumes.

ⓘ 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.

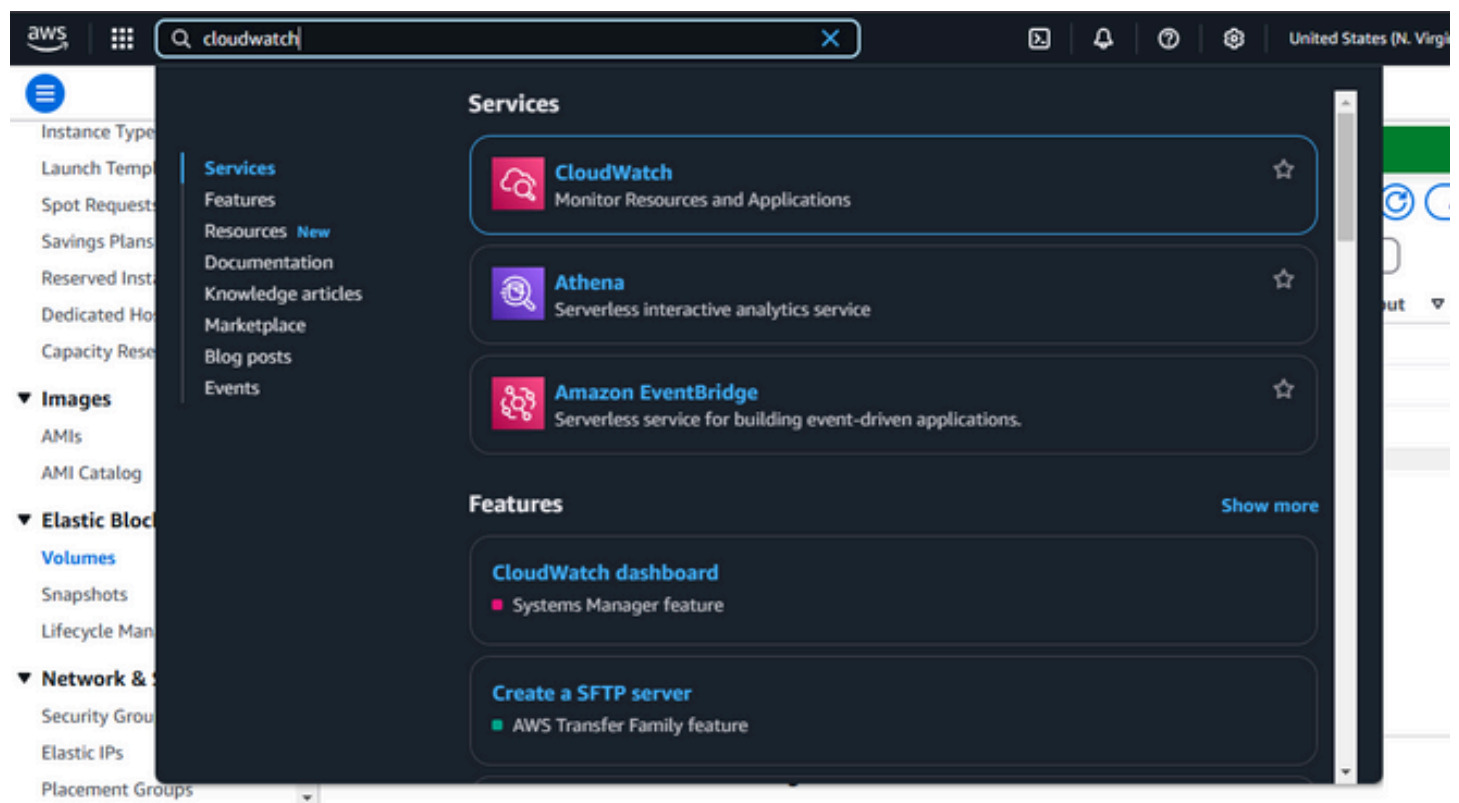
CloudShell Feedback

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

## Step 4:

On the AWS Console homepage, look for the search bar at the top. Type **CloudWatch** in the search bar and press **Enter**.

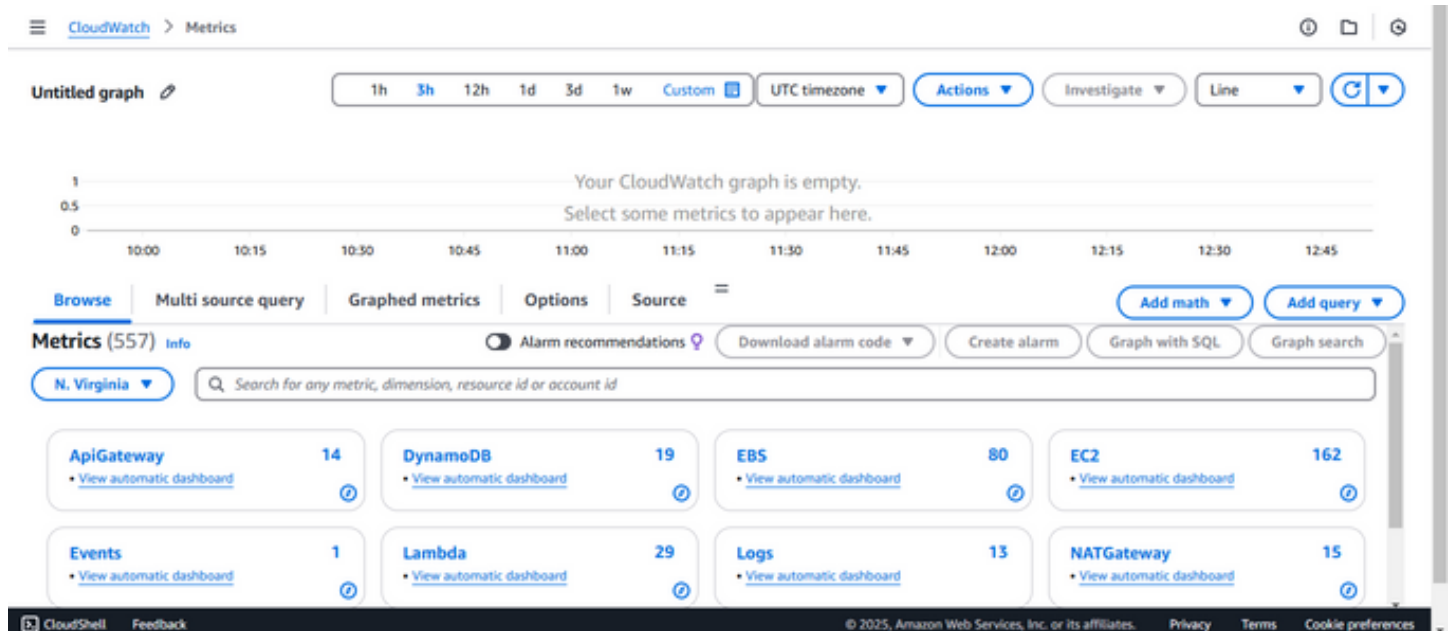
From the search results, click on **CloudWatch**.



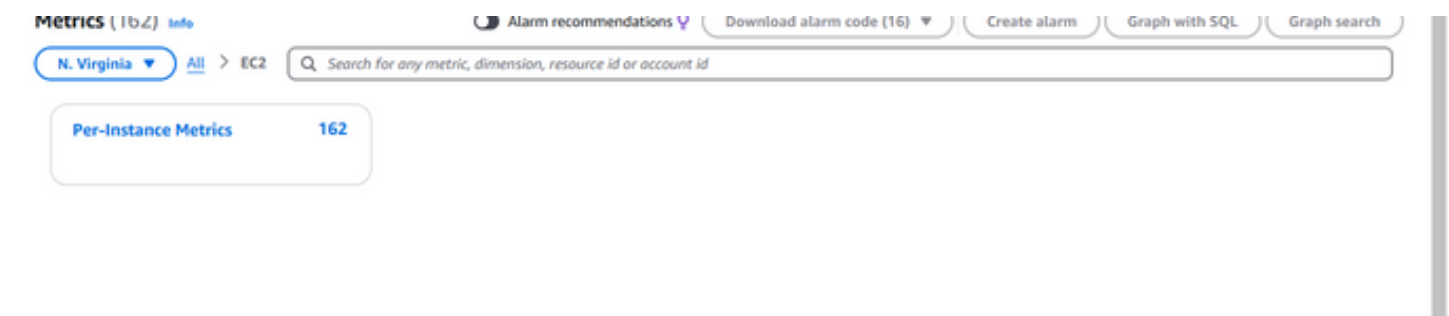
Step 5:

In the CloudWatch dashboard, look at the left-hand menu. Click on **Metrics**.

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



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



## Step 6:

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

Metrics (165) Info

Alarm recommendations Download alarm code (1) Create alarm Graph with SQL Graph search

N. Virginia All > > Per-Instance Metrics Search for any metric, dimension, resource id or account id < 1 2 >

	Instance name 100/165	InstanceId	Metric name	Alarms
<input type="checkbox"/>	sampleinstance	i-07b427e21e865...	MetadataToken	No alarms
<input checked="" type="checkbox"/>	sampleinstance	i-07b427e21e865...	CPUUtilization	No alarms
<input type="checkbox"/>	sampleinstance	i-07b427e21e865...	EBSIOPBalance%	No alarms
<input checked="" type="checkbox"/>	sampleinstance	i-07b427e21e865...	EBSReadOps	No alarms
<input type="checkbox"/>	sampleinstance	i-07b427e21e865...	EBSWriteBytes	No alarms
<input checked="" type="checkbox"/>	sampleinstance	i-07b427e21e865...	EBSWriteOps	No alarms
<input type="checkbox"/>	sampleinstance	i-07b427e21e865...	EBSByteBalance%	No alarms
<input type="checkbox"/>	sampleinstance	i-07b427e21e865...	EBSReadBytes	No alarms

Add dynamic label Info

Statistic: Average Period: 5 minutes Clear graph

<input checked="" type="checkbox"/>	Label	Details	Statistic	Period	Y axis	Actions
<input checked="" type="checkbox"/>	CPUUtilization	EC2 • CPUUtilization • InstanceId: i-07b427e21e865...	Average	5 minutes	< >	🕒 🔔 📄 ⬆ ⬇ ✕
<input checked="" type="checkbox"/>	EBSReadOps	EC2 • EBSReadOps • InstanceId: i-07b427e21e865...	Average	5 minutes	< >	🕒 🔔 📄 ⬆ ⬇ ✕
<input checked="" type="checkbox"/>	EBSWriteOps	EC2 • EBSWriteOps • InstanceId: i-07b427e21e865...	Average	5 minutes	< >	🕒 🔔 📄 ⬆ ⬇ ✕

# 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).
1. **Disk I/O Monitoring:** Added an **EBS volume** to the EC2 instance to track **DiskReadOps** and **DiskWriteOps** metrics, which were visualized in CloudWatch.
2. **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.