



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., CloudWatch on AWS). View metrics like CPU usage and disk I/O for your cloud VM.

Name: CHANDRU S

Department: INFORMATION TECHNOLOGY



Introduction

Effective monitoring is essential in cloud computing to ensure the performance, reliability, and availability of cloud resources. **AWS CloudWatch** offers a powerful monitoring solution that provides real-time insights into AWS services and infrastructure.

This **Proof of Concept (PoC)** explores how to use **CloudWatch** to monitor an **EC2 instance**, focusing on key performance metrics such as **CPU utilization** and **disk I/O**. The PoC demonstrates how to enable monitoring, access real-time data, and analyze system health, helping users optimize cloud-based virtual machines for better performance and reliability.

Overview

This Proof of Concept (PoC) demonstrates how to set up AWS CloudWatch for monitoring an EC2 instance, ensuring optimal performance and resource management. The key steps include:

1. Enabling basic monitoring for an EC2 instance using CloudWatch.
2. Viewing key performance metrics, such as CPU utilization and disk read/write operations, to assess instance health.
3. Utilizing real-time insights from CloudWatch to detect performance bottlenecks and potential issues before they impact services.

By completing this PoC, users will gain a practical understanding of how to integrate CloudWatch monitoring with EC2 instances, enabling proactive performance management and efficient cloud resource utilization.

Objectives

The primary objective of this Proof of Concept (PoC) is to enable AWS CloudWatch monitoring for an EC2 instance and analyze key performance metrics. Specific goals include:

1. **Activating CloudWatch Monitoring** – Enable monitoring for an EC2 instance to track resource utilization.
2. **Analyzing Key Metrics** – View and interpret **CPU usage** and **disk I/O** data to assess instance performance.
3. **Understanding Real-time Monitoring** – Utilize CloudWatch to gain visibility into cloud resource health and performance trends.

Importance

1. **Performance Monitoring** – CloudWatch provides insights into CPU usage, disk I/O, and network traffic, helping to identify and troubleshoot performance issues.

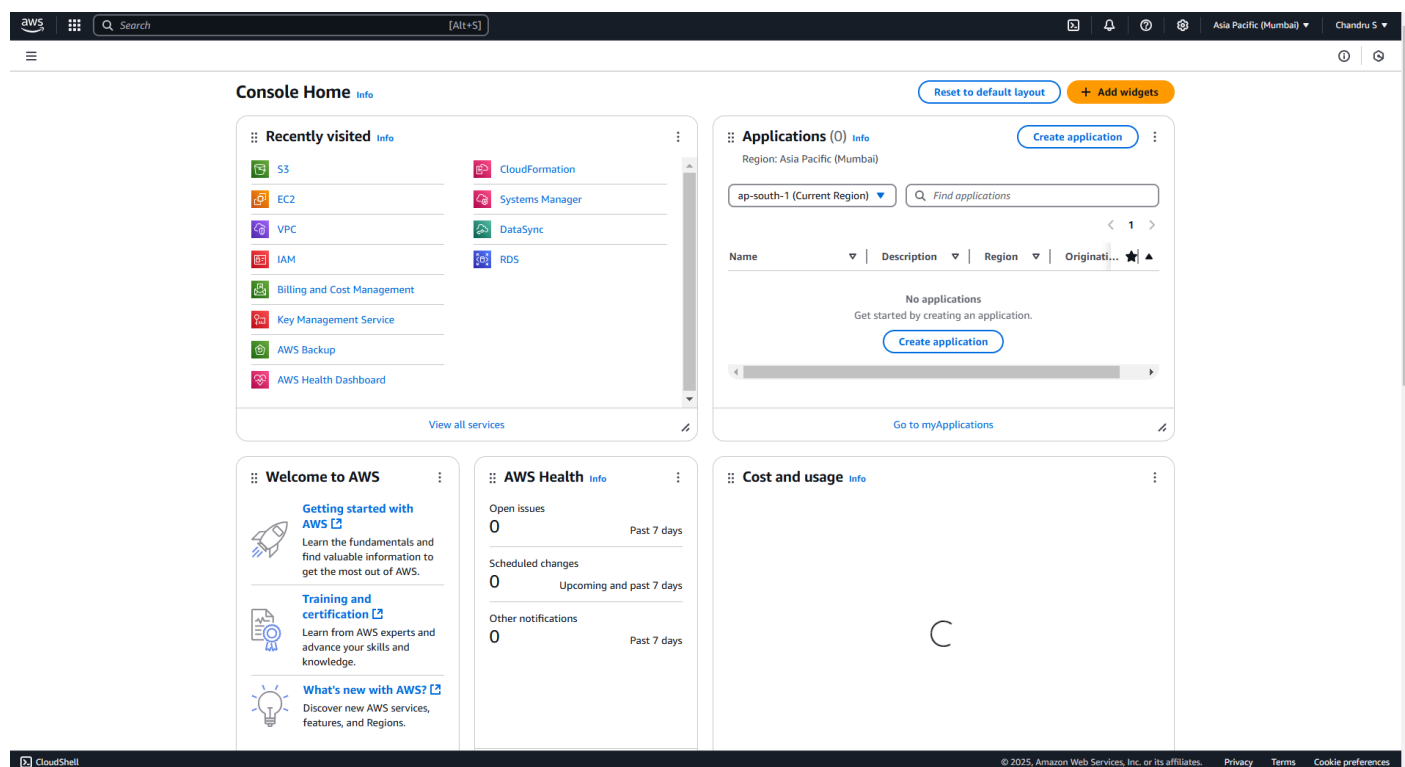
2. **Real-time Visibility** – Enables administrators to access live performance data, ensuring quick responses to fluctuations in resource consumption to prevent downtime.
3. **Efficient Resource Management** – Helps optimize instance capacity and resource allocation, leading to improved efficiency and potential cost savings.
4. **Proactive Issue Detection** – Monitors system behaviour and patterns, allowing early detection of anomalies and performance bottlenecks before they impact operations.

This PoC equips users with hands-on experience in integrating CloudWatch monitoring with EC2, fostering better cloud performance management and reliability.

Step-by-Step Overview

Step 1:

- Open the [AWS Management Console](#).
- Enter your username and password to log in.



Step 2:

- Navigate to the EC2 Dashboard.
- Click Launch Instances and enter a name (e.g., "My Monitoring Instance").
- Keep the default settings and click Launch Instance.

The screenshot shows the AWS Management Console interface for EC2 Instances. The left sidebar contains navigation links for Dashboard, EC2 Global View, Events, Instances, Images, Elastic Block Store, Network & Security, and Load Balancing. The main content area displays the 'Instances (1/1) info' page. A table lists the instance 'My Monitoring Instance' with ID 'i-055146e853899ad3b', state 'Running', type 't2.micro', and availability zone 'ap-south-1a'. Below the table, the 'Details' tab is selected, showing instance summary information such as Instance ID, IP addresses, hostname, and VPC ID.

Step 3:

- In the EC2 Dashboard, go to the left menu and click Volumes under Elastic Block Store (EBS).
- Click Create Volume and configure the storage size and type as needed.
- Click Create to finalize the volume.

The screenshot shows the 'Create volume' page in the AWS Management Console. The page title is 'Create volume' with an 'Info' link. Below the title, it says 'Create an Amazon EBS volume to attach to any EC2 instance in the same Availability Zone.' The 'Volume settings' section contains several fields: 'Volume type' is set to 'General Purpose SSD (gp3)', 'Size (GiB)' is '100', 'IOPS' is '3000', 'Throughput (MiB/s)' is '125', 'Availability Zone' is 'ap-south-1a', and 'Snapshot ID - optional' is 'Don't create volume from a snapshot'. There is an 'Encryption' section with a checkbox 'Encrypt this volume' which is currently unchecked. At the bottom, there is a 'Tags - optional' section with an 'Add tag' button.

Step 4:

- Go to the Volumes list.
- Select the newly created EBS Volume.

- Click Actions > Attach Volume.
- Select the EC2 instance to which the volume should be attached.

The screenshot shows the AWS Management Console interface for the volume 'vol-06080a16b9c3dfd05'. The left sidebar contains navigation links for EC2, Volumes, and other services. The main content area displays the volume's details, including its ID, size, type, IOPS, status check, throughput, multi-attach status, operator, and encryption settings. The 'Actions' dropdown menu is open, showing options like 'Create snapshot', 'Attach volume', 'Detach volume', 'Force detach volume', and 'Manage auto-enabled I/O'. The 'Attach volume' option is highlighted.

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-06080a16b9c3dfd05

Availability Zone

ap-south-1a

Instance [Info](#)

i-055146e853899ad3b

(My Monitoring Instance) (running)

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

Device name [Info](#)

/dev/sdc

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.

[Cancel](#)

[Attach volume](#)

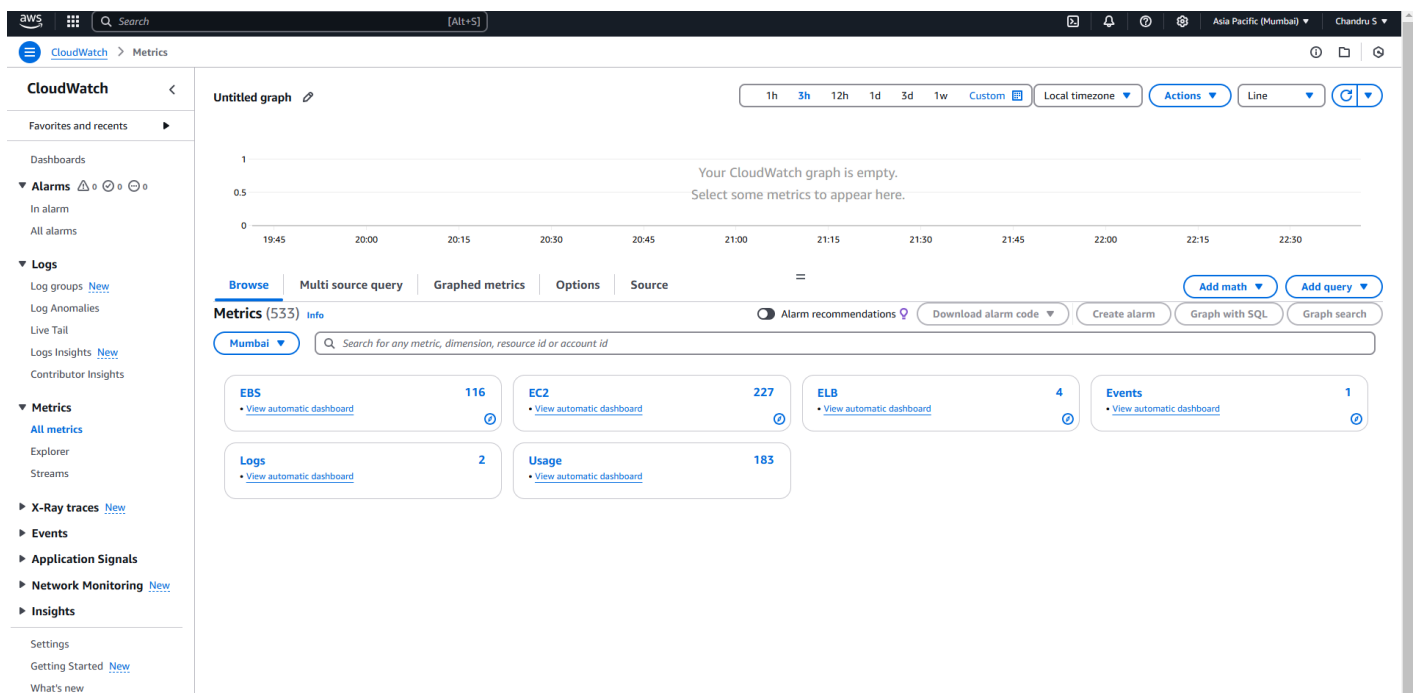
Step 5:

- Go to the AWS Console homepage.
- In the search bar at the top, type **CloudWatch** and press Enter.
- From the search results, click CloudWatch to open the monitoring dashboard.

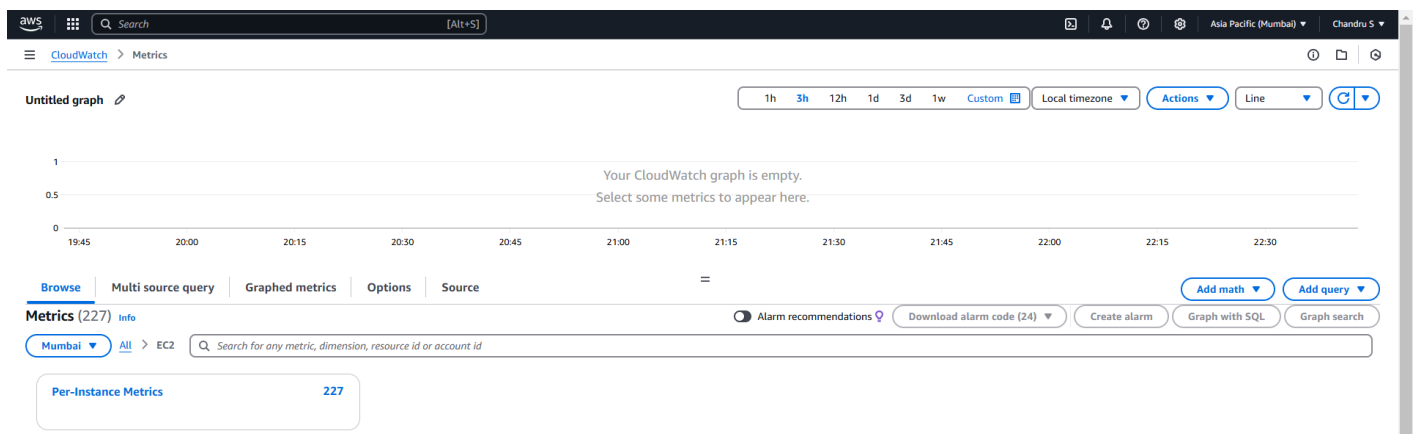
The screenshot shows the AWS Console search results for 'cloudwatch'. The search bar at the top contains the text 'cloudwatch'. Below the search bar, a list of services is displayed. The 'CloudWatch' service is the first result, with a blue border and a star icon. Other services listed include 'Athena' and 'Amazon EventBridge'. The left sidebar shows the navigation menu with 'EC2' selected.

Step 6:

- In the CloudWatch Dashboard, look at the left-hand menu.
- Click **Metrics**.
- Under **Browse**, select **EC2**.

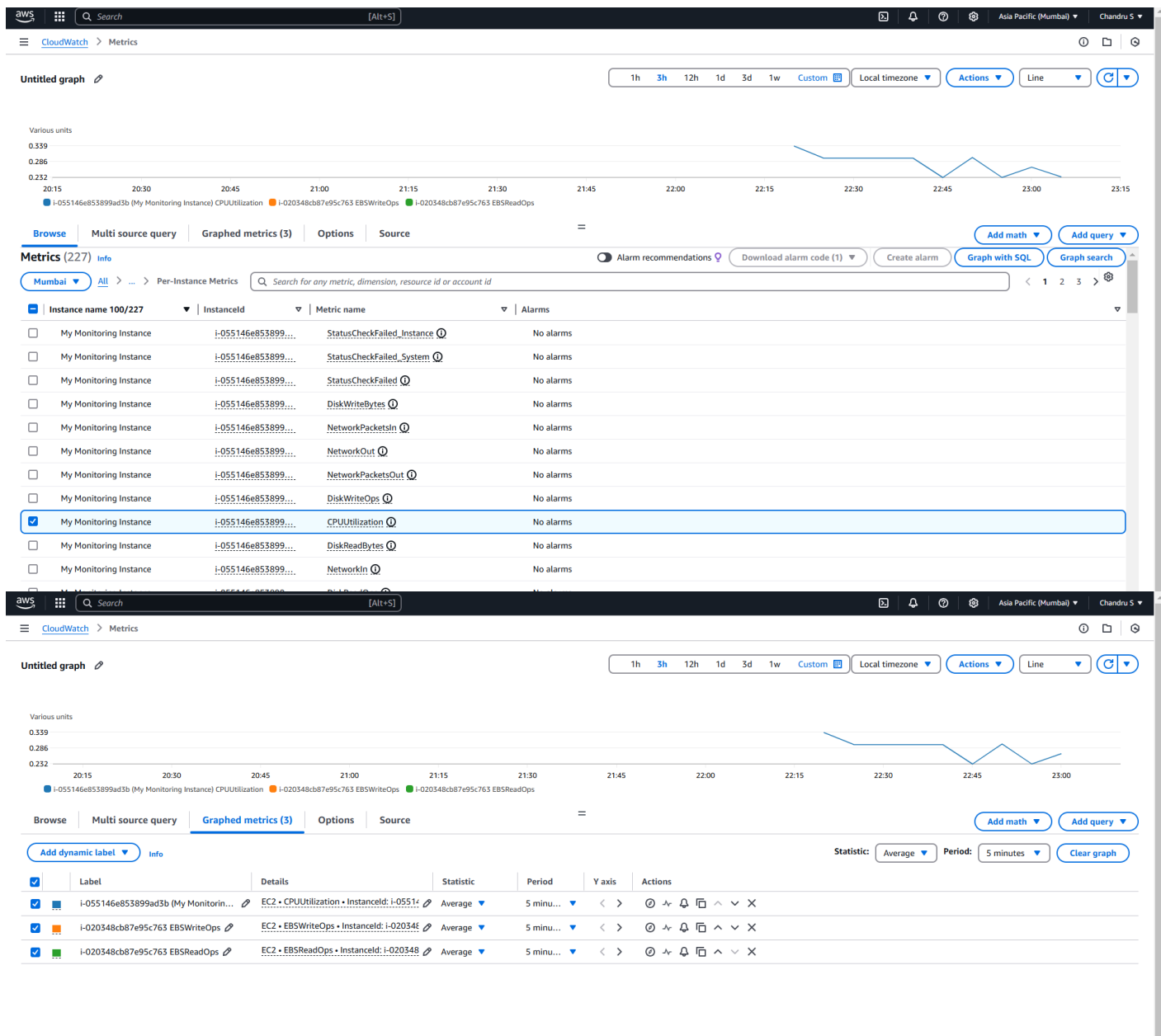


- Click on **Per-Instance Metrics** to view instance-specific data.



Step 7:

- A list of metrics for all EC2 instances will appear, including:
 - **CPUUtilization** (CPU usage)
 - **DiskReadOps / DiskWriteOps** (Disk I/O operations)
- Locate the specific EC2 instance by Instance ID.
- Click on the relevant metrics for your instance.
- To analyze the data, go to the Graphed metrics section for detailed visualization.



By following these steps, you can successfully enable AWS CloudWatch monitoring for your EC2 instance, track key performance metrics, and gain real-time insights into system health and resource usage.

Outcome

This Proof of Concept (PoC) successfully implemented AWS CloudWatch to monitor key performance metrics for an EC2 instance, focusing on **CPU utilization** and **Disk I/O** (DiskReadOps and DiskWriteOps).

Key Outcomes:

- CloudWatch Setup** – Successfully configured AWS CloudWatch to track EC2 instance metrics, including CPU utilization and Disk I/O.
- Disk I/O Monitoring** – Integrated an EBS volume with the EC2 instance to monitor DiskReadOps and DiskWriteOps, visualizing these metrics in CloudWatch.

3. **Cost Efficiency** – Ensured that the EBS volume usage remained within AWS Free Tier limits (30 GB), with all monitored metrics incurring no additional costs.

This PoC provided hands-on experience in configuring CloudWatch monitoring, optimizing cloud resource tracking, and ensuring cost-effective performance monitoring.