

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: DEFIN TN

Department: CSE

Introduction and Overview

Cloud-based monitoring is a critical component of modern infrastructure management, enabling real-time insights into the performance and health of cloud resources. In this PoC, we will set up AWS CloudWatch to monitor key metrics for an EC2 instance, such as CPU utilization, disk I/O, and network traffic. This task demonstrates how to track system performance, identify bottlenecks, and set up alerts for proactive issue resolution.

Objective

The goal of this project is to:

1. Understanding the basics of AWS CloudWatch and its monitoring capabilities.
2. Configuring CloudWatch to monitor essential EC2 metrics.
3. Gaining hands-on experience in proactive cloud resource management

Importance of Cloud-Based Monitoring

Hands-On Learning: Provides practical exposure to cloud-based monitoring tools like AWS CloudWatch, helping you gain essential skills for real-world cloud infrastructure management.

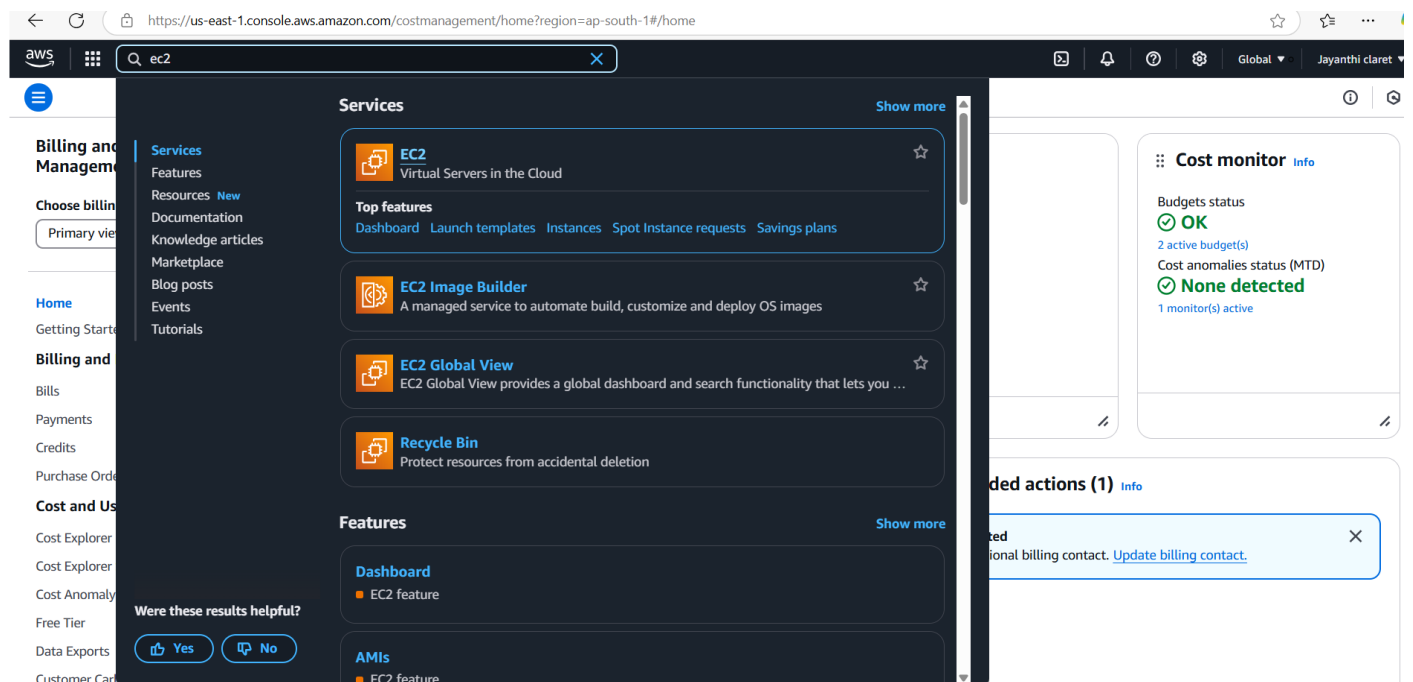
Proactive Resource Management: Enables you to monitor system performance in real-time, identify performance issues, and take corrective actions before they impact end users.

Foundation for Automation: Lays the groundwork for automating monitoring processes, such as setting up alerts and scaling actions, which are critical for efficient cloud operations and DevOps practices.

Step-by-Step Overview

Step1:

Open the AWS Management Console. Navigate to the EC2 Dashboard.

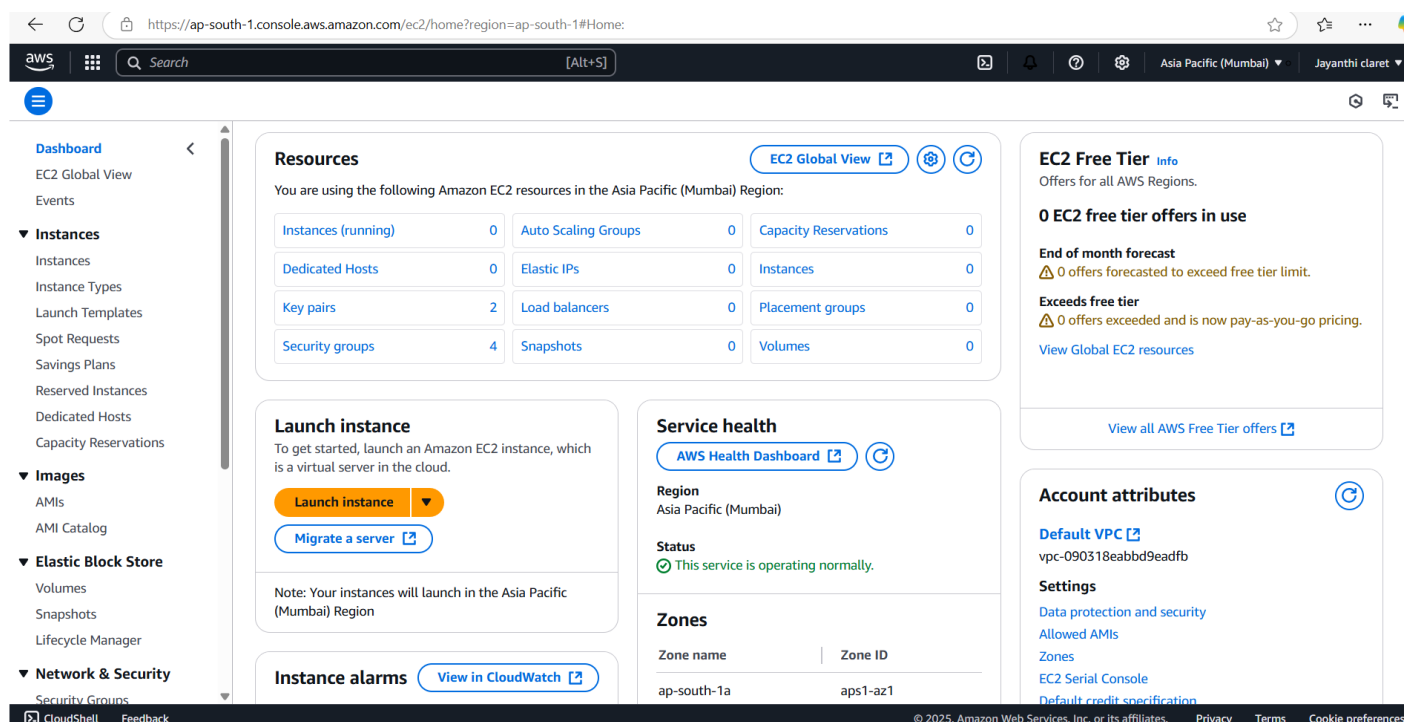


Step 2 :

Click Launch Instance, Configure the instance as needed:

Select an Amazon Machine Image (e.g., Amazon Linux or Ubuntu).

Choose an instance type (e.g., t2.Micro for free-tier eligibility)



Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name

 [Add additional tags](#)

Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Recents

Quick Start

Amazon Linux

aws

macOS

Mac

Ubuntu

ubuntu

Windows

Microsoft

Red Hat

Red Hat

SUSE Linux

SUSE

Debian

debian

[Browse more AMIs](#)
Including AMIs from AWS, Marketplace and the Community

Summary

Number of instances [Info](#)

Software Image (AMI)
Amazon Linux 2023 AMI 2023.6.2...[read more](#)
ami-0b7207e48d1b6c06f

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4

[Cancel](#) [Launch instance](#) [Preview code](#)

Step 3:

Configure the security group to allow necessary ports (e.g., SSH, HTTP, etc.).

Launch an instance [Info](#)

Network [Info](#)

vpc-090318eabbd9eadfb

Subnet [Info](#)

No preference (Default subnet in any availability zone)

Auto-assign public IP [Info](#)

Enable

Additional charges apply when outside of free tier allowance

Firewall (security groups) [Info](#)

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group ☐ Select existing security group

We'll create a new security group called 'launch-wizard-4' with the following rules:

☒ Allow SSH traffic from
Helps you connect to your instance

☐ Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server

☒ Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server

Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Summary

Number of instances [Info](#)

Software Image (AMI)
Amazon Linux 2023 AMI 2023.6.2...[read more](#)
ami-0b7207e48d1b6c06f

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4

[Cancel](#) [Launch instance](#) [Preview code](#)

Step 4:

Launch the instance, While launching the EC2 instance:

Under the "Advanced Details" section, ensure that the CloudWatch monitoring option is enabled.

The screenshot displays the AWS Management Console interface. The top navigation bar shows the AWS logo, a search bar, and the region 'Asia Pacific (Mumbai)'. The breadcrumb trail indicates the path: EC2 > Instances > Launch an instance.

A green success banner at the top states: "Success Successfully initiated launch of instance (i-06beb0e36b1cf3cc2)".

Below the banner is a "Launch log" section. The "Next Steps" section provides a search bar and a list of recommended actions:

- Create billing and free tier usage alerts**: To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds. [Create billing alerts](#)
- Connect to your instance**: Once your instance is running, log into it from your local computer. [Connect to instance](#) [Learn more](#)
- Connect an RDS database**: Configure the connection between an EC2 instance and a database to allow traffic flow between them. [Connect an RDS database](#) [Create a new RDS database](#) [Learn more](#)
- Create EBS snapshot policy**: Create a policy that automates the creation, retention, and deletion of EBS snapshots. [Create EBS snapshot policy](#)

At the bottom of the "Next Steps" section are four more options: [Manage detailed monitoring](#), [Create Load Balancer](#), [Create AWS budget](#), and [Manage CloudWatch alarms](#).

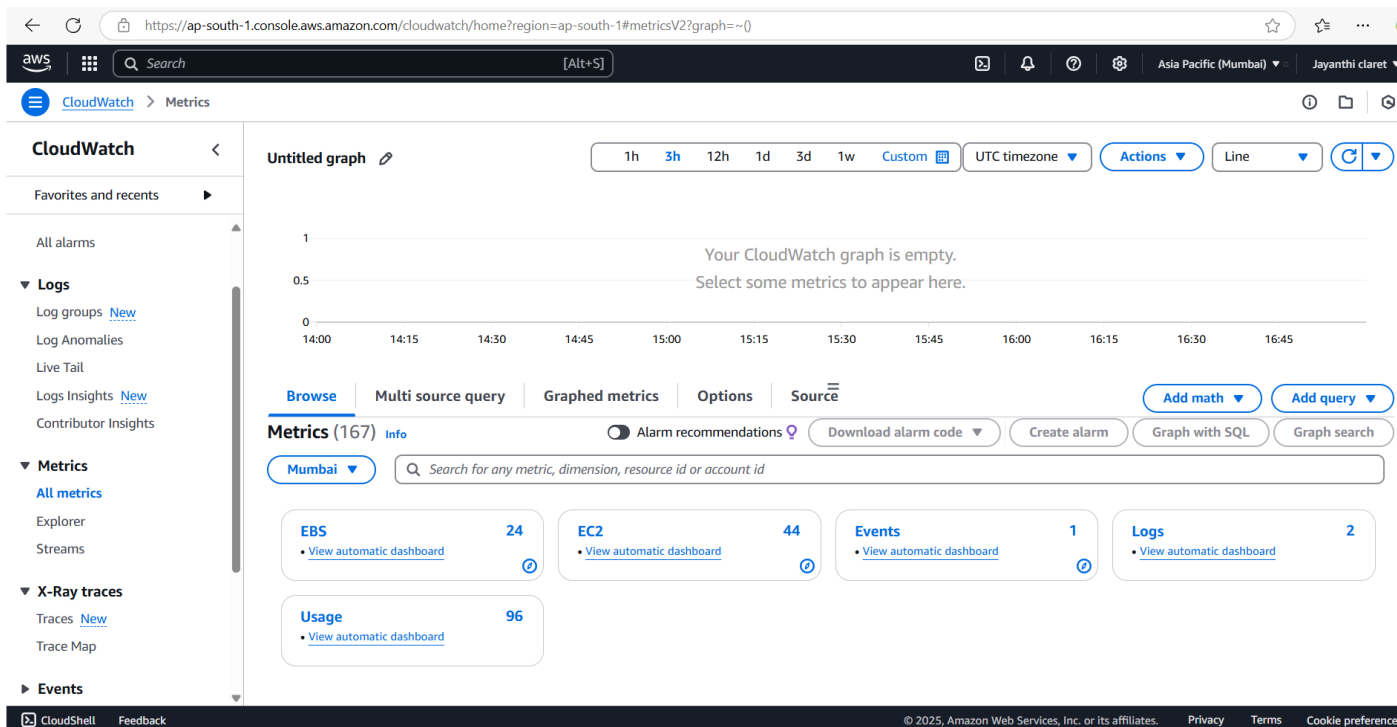
The bottom section of the image shows the "Instances (1/1) Info" page. The left-hand navigation menu includes: Dashboard, EC2 Global View, Events, **Instances** (selected), Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations, **Images**, AMIs, AMI Catalog, **Elastic Block Store**, Volumes, Snapshots, Lifecycle Manager, and **Network & Security**.

The main content area shows the instance details for "task13" (ID: i-06beb0e36b1cf3cc2). The instance is in the "Running" state, using the "t2.micro" instance type, and is located in the "ap-south-1b" Availability Zone. The "Monitoring" tab is selected, showing a notification about "CloudWatch agent metrics" and a toggle to "Include metrics in the CWAgent namespace". The "Tags" tab is also visible.

Step 5:

Open the CloudWatch Dashboard, On the CloudWatch Dashboard, navigate to Metrics on the left-hand menu.

Click All Metrics and choose the EC2 namespace.



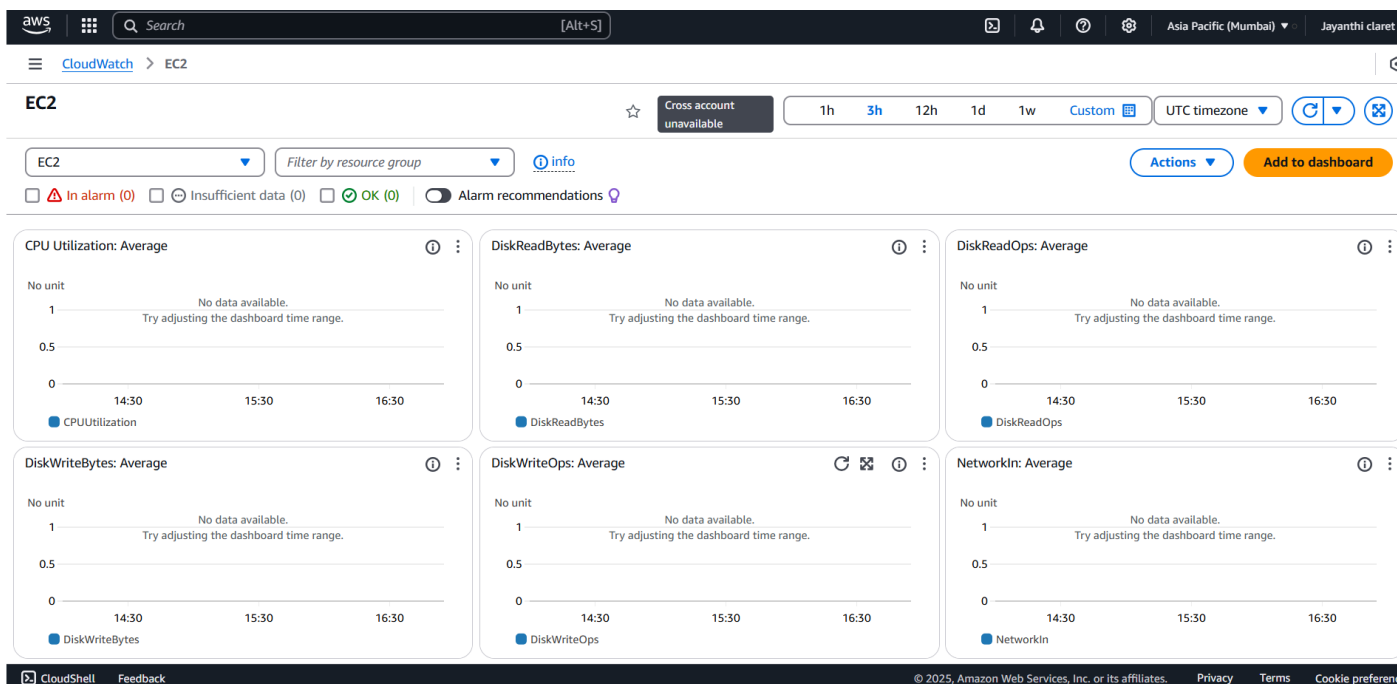
Step 6:

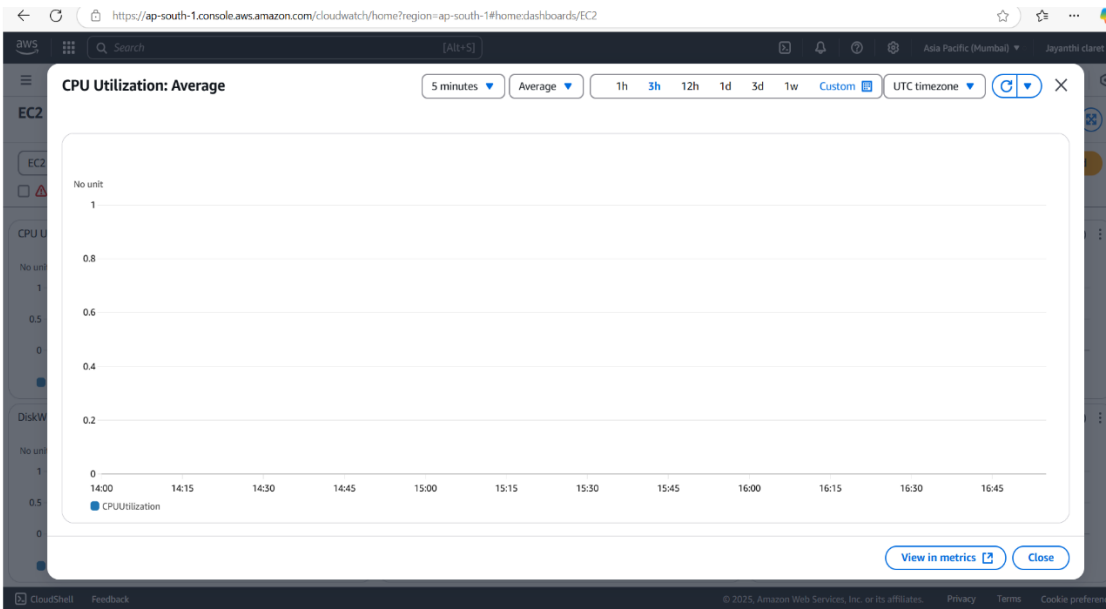
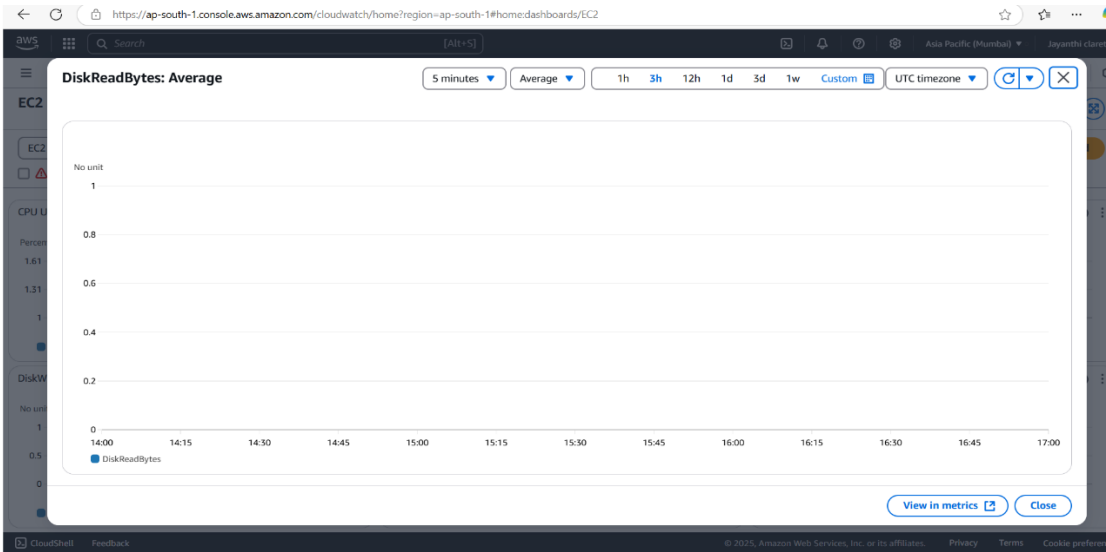
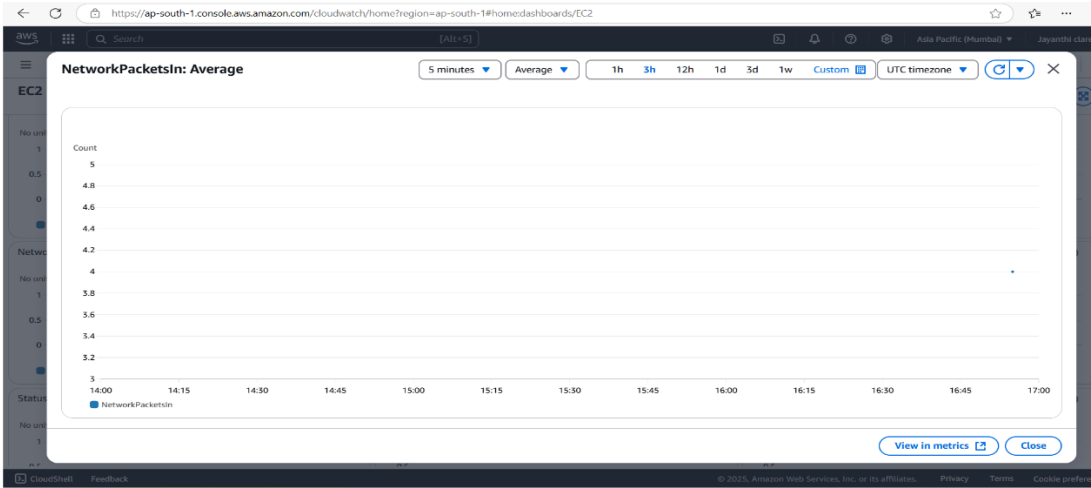
Select metrics like:

CPUUtilization (CPU usage in percentage).

DiskReadBytes and DiskWriteBytes (disk I/O activity).

Network In and Network Out (network data transfer).





Expected Outcome

By completing this POC, you will:

1. Successful setup of AWS CloudWatch to monitor key metrics like CPU usage, disk I/O, and network traffic for an EC2 instance.
2. Creation of a custom CloudWatch dashboard for real-time performance tracking.
3. Improved understanding of cloud monitoring and proactive resource management.