



**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: Vishali V

Department: CSE



## Set Up a Cloud-Based Monitoring Service

### Introduction

Cloud-based monitoring services are essential for managing the performance and health of virtual machines and applications in a cloud environment. Tools like Amazon CloudWatch enable you to monitor metrics such as CPU usage, disk I/O, and network traffic. By enabling these services, you can gain insights into system performance and identify potential bottlenecks or failures before they impact your operations.

### Objectives

1. Learn how to enable basic cloud monitoring services for a virtual machine.
2. Understand how to view and interpret key performance metrics, including CPU usage and disk I/O.
3. Analyze system performance using the monitoring dashboard in the cloud console.

### Steps and Detailed Procedure

#### 1. Enable Monitoring for Your Virtual Machine:

- o **Step 1.1:** Log in to your cloud provider's console (e.g., AWS Management Console).
- o **Step 1.2:** Navigate to the "Instances" or "Virtual Machines" section, depending on the platform.
- o **Step 1.3:** Select the instance or virtual machine you want to monitor.
- o **Step 1.4:** Enable the monitoring service:

- + For AWS: Go to the "Monitoring" tab of the instance and enable detailed monitoring (if not already enabled).
- + For Azure: Enable Azure Monitor by linking your VM to an Azure Log Analytics workspace.
- + For GCP: Enable "Cloud Monitoring" under the "Operations" section of your VM.

- o **Step 1.5:** Save the changes to ensure monitoring is activated.

#### 2. View Metrics in the Monitoring Dashboard:

- **Step 2.1:** Open the monitoring dashboard in the cloud console.
  - + AWS: Navigate to the Amazon CloudWatch dashboard.
  - + Azure: Open the Azure Monitor service from the Azure portal.
  - + GCP: Access "Cloud Monitoring" from the GCP console.
- **Step 2.2:** Select the instance or resource you wish to monitor.
- **Step 2.3:** View real-time and historical metrics such as:
  - + **CPU Utilization:** Tracks the percentage of CPU resources being used. Look for sustained high usage as a potential bottleneck.
  - + **Disk I/O:** Monitors read and write operations. High I/O could indicate heavy disk usage.
  - + **Network Traffic:** Displays data transfer rates to and from the instance.
- **Step 2.4:** Use the graphical interface to customize charts or add widgets for frequently monitored metrics.

### 3. **Set Up Alarms:**

- **Step 3.1:** In the monitoring dashboard, locate the "Alarms" or "Alerts" section.
- **Step 3.2:** Create a new alarm:
  - + Define the metric to monitor (e.g., CPU utilization above 80%).
  - + Set the threshold value and duration to trigger the alarm.
- **Step 3.3:** Configure the notification settings:
  - + Add an email address, SMS number, or other notification channels.
  - + For AWS, create an SNS (Simple Notification Service) topic and subscribe to it.
- **Step 3.4:** Save and activate the alarm.

### 4. **Analyze Performance Trends:**

- **Step 4.1:** Review collected metrics over time to identify trends or anomalies.
- **Step 4.2:** Export logs or reports for deeper analysis (if supported by the platform).
- **Step 4.3:** Use the insights to:
  - + Optimize resource allocation (e.g., scale up or down).
  - + Schedule maintenance during low-usage periods.
  - + Identify potential issues, such as resource contention or misconfigurations.

## 5. **Optional Advanced Steps:**

- **Step 5.1:** Integrate monitoring tools with third-party analytics platforms (e.g., Datadog, Grafana).
- **Step 5.2:** Set up dashboards to monitor multiple resources simultaneously.
- **Step 5.3:** Automate responses to alarms using scripts or cloud functions (e.g., restarting a VM when memory usage exceeds a threshold).

## **Key Learnings**

- Basics of enabling and using cloud-based monitoring tools.
- How to interpret performance metrics like CPU usage and disk I/O.
- Proactive system performance analysis to ensure operational efficiency.
- Setting up alerts for critical conditions to minimize downtime.