Cloud Monitoring and Alerts: A Comprehensive AWS CloudWatch Implementation

Vishva R

July 12, 2025

Contents

Abstract

This report presents a detailed implementation of a cloud monitoring and alerting system using Amazon Web Services (AWS) CloudWatch. The project focuses on monitoring an Amazon EC2 instance, configuring automated alerts, and visualizing performance metrics through a customized dashboard. By integrating CloudWatch with Amazon Simple Notification Service (SNS), this study demonstrates practical applications of cloud monitoring in optimizing resource utilization, ensuring system reliability, and enhancing operational efficiency. The report covers the methodology, system architecture, configuration processes, results, and key learnings, providing a comprehensive resource for academic and professional exploration of cloud infrastructure management.

1 Introduction

Cloud computing has revolutionized IT infrastructure management, enabling scalable, flexible, and cost-effective solutions for organizations worldwide. Central to maintaining the performance and reliability of cloud-based systems is cloud monitoring, a process that involves reviewing, observing, and managing operational workflows. Amazon Web Services (AWS) CloudWatch is a robust monitoring and observability service that provides actionable insights into application performance, system health, and resource utilization. This report details a project undertaken to implement a monitoring solution for an AWS-based application, specifically focusing on an Amazon EC2 instance, with automated alerts and dashboard visualization.

The objectives of this project include:

- Establishing a monitoring system for an EC2 instance using CloudWatch.
- Configuring automated alerts for performance threshold breaches.
- Creating a dashboard for real-time and historical data visualization.
- Integrating Amazon SNS for notification delivery.

This report is structured to provide a comprehensive overview of the project, including system architecture, implementation steps, results, and reflections on the learning outcomes.

2 Background and Literature Review

Cloud monitoring is a cornerstone of modern IT operations, particularly in the context of DevOps and Site Reliability Engineering (SRE). According to AWS documentation, CloudWatch enables the collection of metrics, logs, and events, offering a unified view of system health [**?**]. Studies in cloud computing emphasize the importance of real-time monitoring for proactive issue detection and resource optimization [**?**]. Amazon SNS complements CloudWatch by providing a scalable messaging service for alert notifications [**?**].

The literature highlights several challenges in cloud monitoring, including the complexity of managing distributed systems, ensuring low-latency alerts, and visualizing high-dimensional data. This project addresses these challenges by leveraging CloudWatchs capabilities to monitor an EC2 instance and integrating SNS for timely notifications.

3 Project Overview and Architecture

The project aimed to create a robust monitoring system for an AWS-based application hosted on an EC2 instance. The system was designed to collect performance metrics, trigger alerts based on predefined thresholds, and provide a visual interface for data analysis.

3.1 System Architecture

The architecture comprises the following components:

- **Amazon EC2 Instance**: A virtual server running Amazon Linux 2, serving as the primary application host.
- **AWS CloudWatch**: Collects metrics such as CPU utilization, disk I/O, and network traffic.
- **Amazon SNS**: Routes alert notifications to subscribed endpoints (e.g., email).
- **CloudWatch Dashboard**: Visualizes metrics for real-time and historical analysis.

3.2 Design Considerations

The system was designed to ensure scalability, reliability, and ease of use. Key considerations included:

- High-frequency metric collection for accurate monitoring.
- Threshold-based alerts to detect performance anomalies.
- User-friendly dashboard for operational insights.

4 Monitoring and Alert Configuration

The implementation process involved configuring the EC2 instance, enabling CloudWatch monitoring, and setting up alerts via SNS. The following subsections detail the methodology.

4.1 Setup Process

1. **EC2 Instance Creation**: An EC2 instance was launched using the AWS Management Console with the Amazon Linux 2 AMI. The instance was configured with a t2.micro instance type for cost efficiency.
2. **Enabling Detailed Monitoring**: Detailed monitoring was activated to collect metrics at 1-minute intervals, providing granular insights into system performance.
3. **CloudWatch Alarms**: Alarms were created to monitor critical metrics, including CPUUtilization, DiskReadBytes, DiskWriteBytes, NetworkIn, and NetworkOut.
4. **SNS Topic Creation**: An SNS topic named "EC2$_Alerts"was created to handle notification delivery.$**Email S**
$An email address was subscribed to the SNS Topic, with confirmation completed via the AWS-provided confirmation link.$

4.2 Alarm Configuration

A primary alarm was configured for CPUUtilization exceeding 70% over a 5-minute period. The alarm triggers a notification to the SNS topic, which sends an email to the subscribed

address. Additional alarms were set for disk and network metrics to ensure comprehensive monitoring.

Table 1: CloudWatch Alarm Configurations

Parameter	**Value**
Metric	CPUUtilization
Threshold	$> 70\%$
Evaluation Period	5 minutes
Action	Notify SNS Topic ($EC2_Alerts$)
Metric (Secondary)	DiskReadBytes
Threshold (Secondary)	$> 1,000,000$ bytes
Metric (Tertiary)	NetworkIn
Threshold (Tertiary)	$> 10,000,000$ bytes

5 CloudWatch Dashboard

A customized CloudWatch dashboard was created to visualize the EC2 instances performance metrics. The dashboard includes the following widgets:

- **CPU Utilization**: A line graph displaying CPU usage as a percentage over time.
- **Disk Read/Write Operations**: Bar charts tracking disk I/O activities.
- **Network Traffic**: Area charts showing incoming and outgoing network data.

The dashboard supports real-time monitoring and allows users to analyze historical data by selecting custom time ranges. Widget configurations were optimized for clarity and ease of interpretation.

6 Alert Notification via SNS

The alert system was integrated with Amazon SNS to ensure timely notifications. When the CPUUtilization exceeds 70% for 5 minutes, the CloudWatch alarm transitions to the "ALARM" state, triggering an SNS notification. The SNS topic sends an email to the subscribed address, including details such as the alarm name, metric value, and timestamp. This setup ensures rapid response to potential performance issues, a critical feature for maintaining system reliability.

7 Results and Analysis

The monitoring system was tested under various conditions, including simulated high CPU usage scenarios. Key findings include:

- **Metric Collection**: CloudWatch successfully collected metrics at 1-minute intervals, providing detailed insights into system performance.
- **Alert Accuracy**: The CPUUtilization alarm triggered reliably when the threshold was exceeded, with notifications delivered within seconds via SNS.
- **Dashboard Usability**: The dashboard provided clear visualizations, enabling quick identification of performance trends and anomalies.

To extend the analysis, additional metrics such as memory utilization (via custom metrics) could enhance monitoring capabilities. The systems scalability was validated by adding secondary alarms for disk and network metrics.

8 Challenges and Solutions

Several challenges were encountered during the implementation:

- **Challenge: SNS Subscription Confirmation**: Initial email subscription attempts failed due to unconfirmed subscriptions.
- **Solution**: Ensured the confirmation link was accessed promptly and verified the email address in the AWS Management Console.
- **Challenge: Dashboard Customization**: Configuring widgets to display meaningful data required multiple iterations.
- **Solution**: Utilized CloudWatch documentation and predefined templates to streamline widget creation.

These challenges provided valuable insights into troubleshooting AWS services and optimizing configurations.

9 Future Enhancements

To further improve the monitoring system, the following enhancements are proposed:

- **Custom Metrics**: Implement custom metrics for memory usage and application-specific performance indicators.
- **Automated Scaling**: Integrate Auto Scaling policies with CloudWatch alarms to dynamically adjust EC2 instance capacity.
- **Log Analysis**: Use CloudWatch Logs Insights to analyze application logs for deeper troubleshooting.
- **Multi-Region Monitoring**: Extend the system to monitor resources across multiple AWS regions.

These enhancements would enhance the systems robustness and applicability to complex, production-grade environments.

10 Conclusion and Learnings

This project successfully demonstrated the implementation of a cloud monitoring and alerting system using AWS CloudWatch and SNS. The system effectively monitored an EC2 instance, triggered alerts for performance issues, and provided a user-friendly dashboard for data visualization. Key learnings include:

- Proficiency in configuring and managing EC2 instances for monitoring.
- Expertise in CloudWatch metrics, alarms, and dashboard customization.
- Practical application of SNS for automated alert notifications.
- Understanding the importance of monitoring in maintaining cloud infrastructure reliability.

These skills are critical for roles in cloud computing, DevOps, and IT infrastructure management. The project also highlighted the importance of iterative testing and documentation in

achieving reliable outcomes.

11 Supplementary Information

To provide a deeper understanding, this section elaborates on the technical and operational aspects of the project.

11.1 Technical Details

The EC2 instance was configured with the following specifications:

- **Instance Type**: t2.micro
- **Operating System**: Amazon Linux 2
- **Storage**: 8 GB General Purpose SSD (gp2)
- **Security Group**: Allow HTTP and SSH traffic

CloudWatch metrics were collected using the default AWS SDK, with no additional agents installed. The SNS topic was configured with a standard delivery policy to ensure reliable notification delivery.

11.2 Operational Considerations

Operationally, the system was designed to minimize latency in alert delivery and ensure high availability. The choice of a 5-minute evaluation period for alarms balanced sensitivity with the need to avoid false positives. The dashboard was optimized for use by both technical and non-technical stakeholders, with clear labels and intuitive layouts.

12 Case Study: Real-World Application

To contextualize the project, consider a real-world scenario where a company deploys a web application on an EC2 instance. The monitoring system described in this report would enable the company to:

- Detect performance bottlenecks, such as high CPU usage during peak traffic.
- Receive immediate alerts for system anomalies, enabling rapid response.
- Analyze historical data to optimize resource allocation and reduce costs.

This case study underscores the practical value of cloud monitoring in enterprise settings, aligning with industry trends toward proactive system management.

References

- AWS CloudWatch Documentation. Available at: `https://aws.amazon.com/cloudwatch/` [**?**]
- AWS Simple Notification Service Documentation. Available at: `https://aws.amazon.com/sns/` [**?**]
- ACM Digital Library. "Cloud Monitoring: A Survey." Available at: `https://dl.acm.org` [**?**]

Prepared by: Vishva R

Date: July 12, 2025