USRYA
EMPOWERING DIGITAL TRANSFORMATION

Project Title:

Monitoring and Notification System for Error Logs

Created By:

Deepak Patra

Reference ID:- UA/BLR/000013

Reviewed By:

USRYA ARTIVISION PRIVATE LIMITED

Objective: - The objective of this solution is to implement a detailed monitoring and notification system for Apache error logs. The system will continuously analyse the Apache error logs for errors, and send real-time notifications via AWS SNS (Simple Notification Service) when critical errors are detected. This solution aims to enhance the operational efficiency and reliability of the web application hosted on Apache web servers.

1. Overview:

- Apache web servers play a critical role in hosting web applications and services. Monitoring Apache error logs is essential for identifying and resolving issues that may impact the performance and availability of the application.
- This solution employs AWS services, including CloudWatch Logs for log collection, CloudWatch Metric Filters for log analysis, and SNS for notification delivery.
- This solution used AWS services, including CloudWatch Logs for log collection, CloudWatch Metric Filters for log analysis, and SNS for notification delivery.

2. Architecture:

Log Collection: Apache error logs are collected from multiple web servers using CloudWatch Logs Agent. The agent securely streams log data to CloudWatch Logs in near real-time.

Log Analysis: CloudWatch Metric Filters are used to parse and analyze Apache error log entries. Metric Filters are configured to detect specific error patterns (i.e.- “error” or “ERROR”) indicative of critical issues.

Alerting and Notification: When a critical error is detected, CloudWatch Alarms trigger SNS notifications. SNS delivers notifications via email or other endpoints.

2. Components:

- **Apache web server:** Host the web application and generate error logs, or either you can download the error files from GitHub.
- **CloudWatch Logs:** Collect(go to your local system and fire the cmd:- “scp -i <path of pem file> <path to your transfer file> ec2-user@<ec2-ip>:<path of ec2 store>”,before that make sure that proper permissions are granted) and store Apache error logs in a centralized location.
- **CloudWatch Metric Filters:** Analyze log entries to identify critical errors.
- **CloudWatch Alarms:** Monitor Metric Filter metrics and trigger notifications.
- **SNS (Simple Notification Service):** Deliver real-time notifications to operations team members.

3. **Implementation Steps and Use case :**

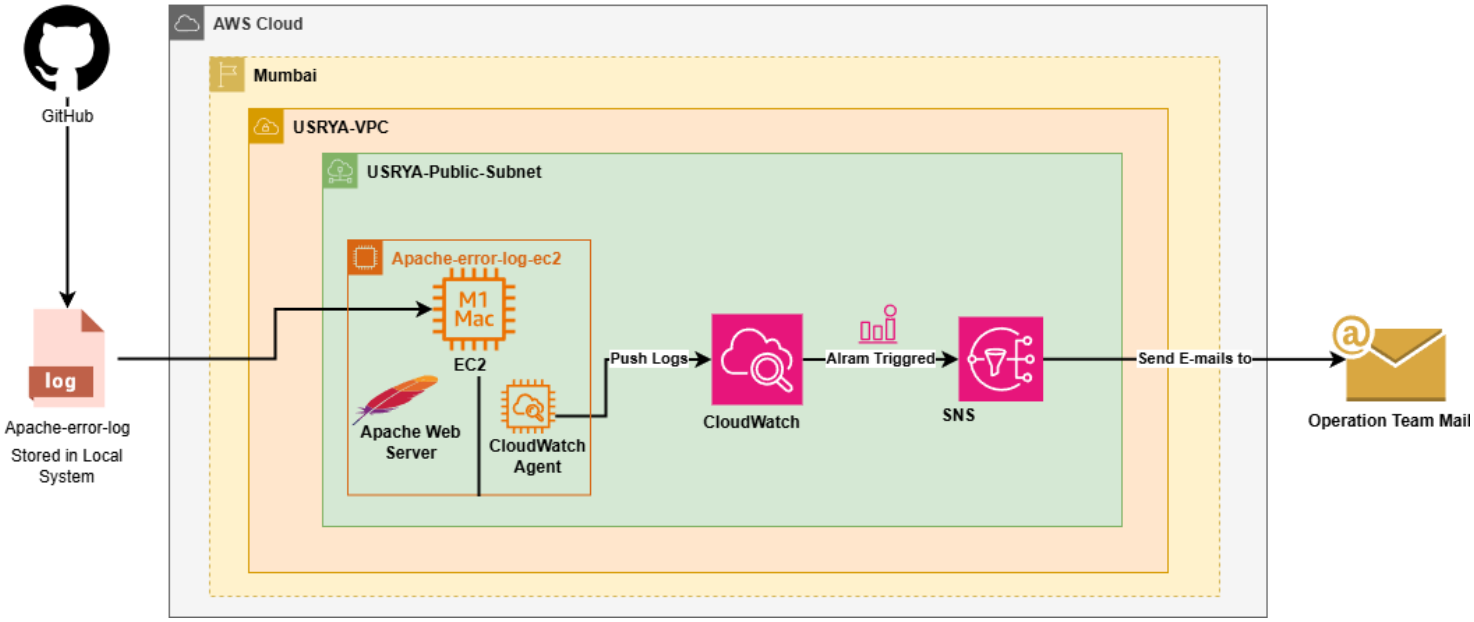
- **Step 1: If you want to Take the Error Logs from GitHub Then You can take the Error from:- loghub/Apache/Apache_2k.log at master · logpai/loghub (github.com).**
- If You want to Take the Log files from EC2 Apache Server Then At first launch Two EC2 Instances and install httpd server.
- **Step 2: Configure CloudWatch Logs Agent inside each EC2:** Install and configure CloudWatch Logs Agent on each Apache web server to stream error logs to CloudWatch Logs. Why Because configuring the CloudWatch Logs Agent to stream error logs from Apache web servers to CloudWatch Logs provides a robust and scalable solution for centralized log management, real-time monitoring, and proactive issue detection in your infrastructure.
- **Step 3: Create Metric Filters:** Define Metric Filters in CloudWatch Logs to parse “error” or “ERROR” log entries and identify critical

errors. Why Because creating Metric Filters in CloudWatch Logs to parse error log entries is essential for effective error detection, alerting, and monitoring in your Apache web server environment. This step enhances the responsiveness, reliability, and scalability of your monitoring solution, empowering you to maintain the availability and performance of your web applications.

- **Step 4: Configure Alarms:** Set up CloudWatch Alarms to monitor Metric Filter metrics and trigger SNS notifications when critical errors occur. Why Because configuring CloudWatch Alarms to monitor Metric Filter metrics and trigger SNS notifications enhances the responsiveness, reliability, and scalability of your error monitoring and alerting system. This proactive approach enables you to effectively manage incidents, minimize downtime, and maintain the overall health of your Apache web server infrastructure.
- **Step 5: Create SNS Topic:** Create an SNS topic to serve as the notification endpoint for CloudWatch Alarms. Why Because creating an SNS topic to serve as the notification endpoint for CloudWatch Alarms enhances the effectiveness, scalability, and reliability of your error monitoring and alerting system for Apache web servers. This centralized notification hub facilitates flexible, targeted, and timely communication of critical error events to stakeholders and response teams, enabling rapid incident response and resolution.
- **Step 6: Subscribe Operations Team:** subscribe to the SNS topic to receive notifications. Why Because subscribing the operations team to the SNS topic enables them to receive timely, actionable notifications about critical error events detected in the Apache error logs. This facilitates rapid incident response, promotes collaboration and communication among team members, and

supports continuous improvement of operational processes and procedures.

Architecture Diagram when you take error files from GitHub: -



Overview

This architecture involves collecting Apache error logs from a GitHub repository, processing them on an EC2 instance, pushing the logs to CloudWatch, analyzing the logs using CloudWatch, and sending notifications via Amazon SNS when certain conditions are met. The entire setup is hosted in the AWS Mumbai region within a VPC.

Components

1. **GitHub Repository:**
    - Apache-error-log: The source of the Apache error log file, which is stored in a GitHub repository.
2. **AWS Cloud (Mumbai Region):**
    - The architecture is deployed in the Mumbai region within an AWS VPC.
3. **VPC (Virtual Private Cloud):**

- **USRYA-VPC:** A virtual network dedicated to your AWS account.
- **USRYA-Public-Subnet:** A public subnet within the VPC, meaning it is accessible from the internet.

4. **Amazon EC2:**
    - Apache-error-log-ec2: An EC2 instance within the public subnet.
    - Role: This instance pulls the Apache error log from the GitHub repository and processes it.
5. **Amazon CloudWatch:**
    - Logs: After processing the log file, the EC2 instance sends the log data to Amazon CloudWatch Logs for monitoring and further analysis.
6. **AWS Lambda:**
    - Processing and Analysis: AWS Lambda functions are used to further process and analyze the log data streamed from CloudWatch Logs.
    - Trigger: CloudWatch Logs can trigger a Lambda function whenever new log data is available.
7. **Amazon SNS (Simple Notification Service):**
    - Notifications: Once the log data is processed and stored, SNS is used to send notifications.
    - Email Notification: SNS sends an email notification to inform the stakeholders about the log processing status or any significant findings from the logs.

Workflow

1. **Log Collection from GitHub:**

- The Apache error log is stored in a GitHub repository. The EC2 instance (Apache-error-log-ec2) is responsible for fetching this log file.

2. **Processing on EC2 Instance:**
    - The EC2 instance, which runs within the public subnet of the VPC, pulls the Apache error log from the GitHub repository and processes it as necessary. This could involve filtering, formatting, or other pre-processing steps.
3. **Push Logs to CloudWatch:**
    - Once the log is processed on the EC2 instance, the log data is pushed to Amazon CloudWatch Logs. CloudWatch Logs provide centralized logging, making it easier to monitor and analyze the log data.
4. **Log Analysis with AWS Lambda:**
    - AWS Lambda functions are configured to trigger based on new log data in CloudWatch Logs. These Lambda functions can perform further analysis on the logs, such as identifying specific error patterns or extracting key metrics.
5. **Triggering CloudWatch Alarms:**
    - Based on the analysis, CloudWatch Alarms can be set up to monitor specific conditions within the log data. For example, if a certain number of errors are detected within a specific time period, an alarm can be triggered.
6. **Sending Notifications via SNS:**
    - When a CloudWatch Alarm is triggered, it can send a notification to an Amazon SNS topic.

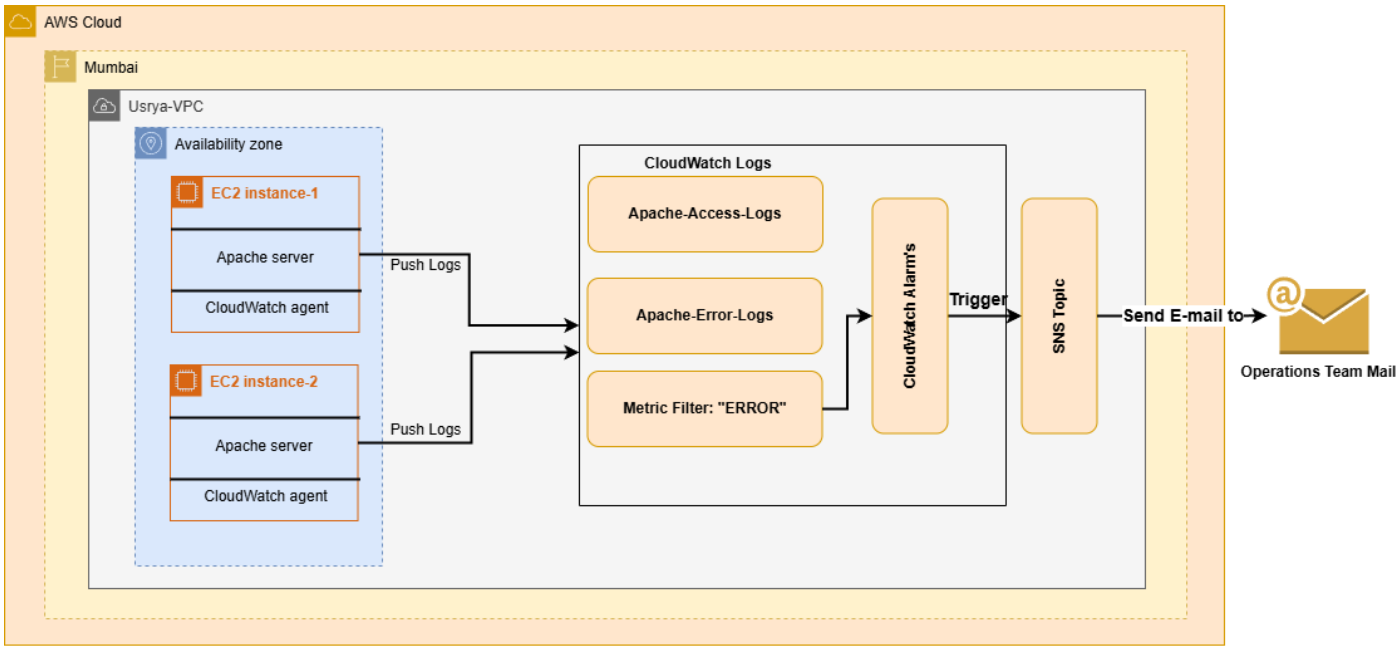
- The SNS topic is configured to send email notifications to designated recipients, alerting them to the issue identified in the log data.

Benefits of this Architecture

- Automation: Automates the process of fetching, processing, analyzing, and notifying based on log data.
- Scalability: Uses AWS services like EC2, CloudWatch, and Lambda, which can scale based on the volume of logs and processing needs.
- Centralized Monitoring: CloudWatch provides a centralized location for log data, making it easier to monitor and respond to issues.
- Real-time Alerts: SNS provides real-time notifications, allowing for quick response to critical log **events.**

This architecture provides a robust solution for handling and monitoring Apache error logs, leveraging AWS services to automate and streamline the process from log collection to notification.

Architecture Diagram when you take error files from Apache: -



Overview

This architecture involves multiple EC2 instances running Apache servers, pushing logs to CloudWatch Logs, which then triggers alarms based on specific metrics, and finally sends notifications via SNS (Simple Notification Service) to email recipients. This setup is useful for monitoring Apache server logs, particularly for tracking and alerting on error logs.

Components

1. **Amazon EC2 Instances:**
    - EC2 instance-1 and EC2 instance-2: These instances run Apache servers and have the CloudWatch agent installed.
    - Apache Server: Handles web traffic and generates logs (both access logs and error logs).
    - CloudWatch Agent: Collects and pushes the logs from the EC2 instances to CloudWatch Logs.
2. **Amazon CloudWatch Logs:**
    - Apache-Access-Logs: A log stream where access logs from the Apache servers are stored.
    - Apache-Error-Logs: A log stream where error logs from the Apache servers are stored.
    - Metric Filter: A filter applied to the error logs to search for specific keywords, in this case, "ERROR".
    - CloudWatch Alarms: Alarms that are triggered based on the metrics derived from the log data. For example, an alarm could be set to trigger if the number of "ERROR" entries in the logs exceeds a certain threshold.
3. **Amazon SNS (Simple Notification Service):**
    - SNS Topic: A topic configured to send notifications when a CloudWatch alarm is triggered.

- Email Notifications: When an alarm is triggered, the SNS topic sends an email to the specified recipients.

Workflow

1. **Log Collection:**
    - Both EC2 instances (instance-1 and instance-2) run Apache servers that generate access logs and error logs.
    - The CloudWatch agent on each instance collects these logs and pushes them to CloudWatch Logs.
2. **Log Storage:**
    - In CloudWatch Logs, there are separate log streams for Apache access logs and error logs.
    - Logs are continuously pushed and stored in these log streams.
3. **Metric Filtering and Alarming:**
    - A metric filter is applied to the Apache error logs to identify occurrences of the word "ERROR".
    - CloudWatch uses this metric to monitor the error logs.
    - When the number of "ERROR" entries exceeds a predefined threshold, a CloudWatch alarm is triggered.
4. **Notification:**
    - The CloudWatch alarm triggers an SNS topic.
    - The SNS topic is configured to send an email notification to the specified recipients, alerting them about the error condition.

Detailed Explanation

1. EC2 Instances and Log Collection:

- EC2 instance-1 and EC2 instance-2 run Apache servers, which handle incoming HTTP requests and generate logs.
- The CloudWatch agent installed on these instances is configured to collect both access and error logs and push them to CloudWatch Logs.

2. CloudWatch Logs:

- Logs from both instances are pushed to CloudWatch Logs, where they are stored in separate log streams: Apache-Access-Logs and Apache-Error-Logs.
- Within CloudWatch Logs, a metric filter is configured to search for the keyword "ERROR" within the Apache-Error-Logs.

3. CloudWatch Alarms:

- The metric filter identifies occurrences of "ERROR" in the logs and generates a corresponding metric.
- A CloudWatch Alarm is set to monitor this metric. If the number of "ERROR" occurrences exceeds a predefined threshold, the alarm is triggered.

4. SNS Notifications:

- When the CloudWatch Alarm is triggered, it sends a notification to an SNS Topic.
- The SNS topic then sends an email notification to the designated recipients, alerting them about the error condition in the Apache servers.

Benefits of this Architecture

- Automated Monitoring: Automatically collects and monitors Apache logs for errors.
- Real-time Alerts: Provides real-time alerts through SNS, allowing for prompt action on issues.

- Scalable: Can be easily scaled to include more instances or additional log streams as needed.
- Centralized Log Management: Uses CloudWatch Logs for centralized log management and monitoring.

This architecture provides a robust solution for monitoring Apache server logs, identifying and alerting on error conditions, and ensuring timely notifications to maintain the health and performance of the web servers.

Total cost:

Sl. No	Component	Type	Count	Per Count monthly Cost	Total Cost Yearly	Remarks
1	EC2	t2.nano	2	2×2.19 USD	26.28 USD	
2	VPC	Public	1	1×13.24 USD	158.88 USD	
3	CloudWatch		1	1×4.85 USD	58.2 USD	
4	SNS		1	1×0.18 USD	2.16 USD	
Total Yearly Cost is = 245.52 USD						

You can find the price at:

https://calculator.aws/#/estimate?id=7b958c52a37e2d5a5a3e86c2b0d595a75c9ac244