Project 1: AWS CloudWatch Dashboards for Comprehensive Monitoring

Intern: Shyam Champakbhai Chotaliya **Company:** Fortune Cloud Technologies **Date:** [14-07-2025]

Project Overview

This project involved designing and implementing a centralized monitoring solution using Amazon CloudWatch Dashboards. The objective was to gain real-time visibility into key operational and financial metrics across four critical focus areas: Billing & Cost, Application & System Logs, Network Performance, and Security & Compliance. This dashboard serves as a single pane of glass for monitoring the health, performance, and cost of AWS resources.

Description of Dashboard Sections

The final dashboard is logically divided into four sections, each targeting a specific monitoring requirement.

1. Billing and Cost Dashboard

This section provides financial visibility into the AWS account.

- **Widgets:**
  - **Total Estimated Charges (USD):** A number widget showing the current month-to-date estimated cost.
  - **Daily Cost by Service:** A bar chart breaking down daily spending by individual AWS services (e.g., EC2, S3), making it easy to identify cost drivers.
  - **Monthly Cost Breakdown:** A line chart tracking costs over the month.

2. Logs Dashboard

This section centralizes log analysis, enabling quick troubleshooting and operational awareness.

- **Widgets:**
  - **Recent Application Errors:** A logs table widget running a CloudWatch Log Insights query to display recent lines from Nginx error logs containing keywords like "error" or "exception".
  - **Failed Login Attempts:** A number widget that displays a count of failed SSH login attempts, derived from a custom CloudWatch Metric Filter on system security logs.

3. Network Performance Dashboard

This section monitors the flow of data in and out of our key resources.

- **Widgets:**
  - **EC2 Network In/Out:** A line chart showing the bytes transferred in and out of the primary web server (EC2 instance), which is crucial for identifying traffic spikes or potential data exfiltration.
  - **ELB Request Count & Errors (Optional):** A stacked area graph showing the total request count alongside 4xx (client) and 5xx (server) error rates for the load balancer.

4. Security and Compliance Dashboard

This section acts as an early warning system for potential security threats and compliance issues.

- **Widgets:**
  - **Active GuardDuty Findings:** A gauge widget displaying the total number of active security findings from AWS GuardDuty.
  - **Recent Unauthorized API Activity:** A logs table showing API calls from CloudTrail that resulted in "Access Denied" or "Unauthorized" errors.
  - **Non-Compliant Config Resources:** A number widget counting the resources that are currently flagged as non-compliant by AWS Config rules.

Step-by-Step Setup Instructions

The following steps were taken to create the monitoring solution:

1. **Pre-Setup & Configuration:**
   - **Billing:** Enabled **Cost Explorer** and **Billing Alerts** in the AWS Billing console.
   - **Security Services:** Enabled **AWS GuardDuty** (and generated sample findings), **AWS Config** (with rules like `s3-bucket-public-read-prohibited`), and **AWS CloudTrail** (configured to send management events to a new CloudWatch Log Group).
2. **Resource & Agent Setup:**
   - Created an **IAM Role** (`EC2-CloudWatch-Agent-Role`) with the `CloudWatchAgentServerPolicy` permission.
   - Launched a `t2.micro` **EC2 instance** with the IAM role attached.
   - Installed **Nginx** as a sample web server to generate logs.
   - Installed and configured the **CloudWatch Agent** on the EC2 instance to collect Nginx error logs (`/var/log/nginx/error.log`) and system security logs (`/var/log/secure`).
3. **Dashboard and Widget Creation:**
   - Created a new CloudWatch Dashboard named `Comprehensive-Monitoring-Dashboard`.
   - Added widgets for each of the four sections, connecting them to the appropriate data sources:
     - **Billing widgets** were linked to Billing metrics and Cost Explorer data.

        - **Log widgets** were configured to query the CloudWatch Log Groups created by the Agent and CloudTrail.
        - **Network widgets** were based on EC2 and ELB metrics.
        - **Security widgets** pulled data from GuardDuty, CloudTrail, and AWS Config metrics.

4. **Metric Filter :**

    - A **CloudWatch Metric Filter** was created to parse `System-Security-Logs` for failed login attempts and publish a custom metric (`FailedLogins`).

Insights Gained from the Metrics

- **Cost Insight:** The dashboard immediately clarified that EC2 instances were the primary cost driver. This insight can guide future cost-optimization efforts, such as choosing Reserved Instances or Savings Plans.
- **Operational Insight:** The Log Insights widget for application errors is incredibly powerful. It eliminates the need to manually SSH into a server and `tail` log files, reducing the Mean Time to Resolution (MTTR) for application issues.
- **Security Insight:** The security dashboard proved effective as an early warning system. The "Unauthorized API Activity" widget successfully captured test attempts to access resources without permission, demonstrating its value in detecting potential account reconnaissance.
- **Performance Insight:** The network traffic widget provides a clear baseline of normal activity. Any significant deviation from this baseline could indicate a DDoS attack, a viral marketing success, or faulty application behavior, prompting further investigation.

> ***Note:** This document covers only Project 1 out of the 6 assigned internship projects. The remaining projects are in progress and will be documented separately.*