FinOps Dashboard for Cloud Cost Visibility (Free Tier Usage Tracker)

Abstract

The FinOps Dashboard is a cloud cost monitoring tool designed to track AWS Free Tier service usage and identify potential overages. By automating the collection of cost and usage data from AWS Cost Explorer and storing it in a local SQLite database, this project provides a real-time, visual overview of cloud expenses using Grafana dashboards. The dashboard allows users to detect services approaching billing limits, facilitating informed cost optimization and resource management.

Introduction

Cloud services can easily incur unexpected costs if Free Tier limits are exceeded. Monitoring daily usage manually is time-consuming and prone to errors. This project provides an automated solution that:

- Tracks AWS service usage and costs daily.
- Visualizes trends to detect potential overages.
- Stores historical data for reporting and analysis.

The FinOps Dashboard enables cloud users to manage resources effectively, avoid accidental billing, and maintain cost-efficient operations.

Tools Used

Tool	Purpose
AWS EC2 (Amazon Linux 2023)	Hosting Python scripts and Grafana dashboards
AWS Cost Explorer API	Fetching daily AWS cost and usage data
Python 3 + Boto3	Automating data fetch from AWS
SQLite	Storing cost and usage data locally
Grafana	Dashboard visualization of usage and costs
Cron (cronie)	Scheduling automated daily data collection

Steps Involved in Building the Project

1. AWS Setup

- Created IAM user with programmatic access.
- Assigned AmazonCostExplorerReadOnlyAccess policy.
- Enabled AWS Cost Explorer to access daily usage and cost data.

2. EC2 Instance Setup

- Launched Amazon Linux 2023 EC2 instance (Free Tier).
- Installed Python 3, pip, SQLite, and Grafana.
- Created project folder and virtual environment.

3. Fetching AWS Cost Data

- Developed fetch_cost_data.py script to retrieve cost data using Boto3.
- AWS credentials (Access Key and Secret Key) are securely configured in the script or via IAM roles.
- Stored fetched data into SQLite database for persistence.

4. Automating Daily Data Collection

- Installed cronie for scheduling.
- Added cron job to execute fetch_cost_data.py daily.

5. Grafana Dashboard Setup

- Installed Grafana and SQLite plugin.
- Configured SQLite data source pointing to finops_data.db.
- Created table and bar chart panels to visualize service-wise costs.
- Added color-coded alerts for services nearing Free Tier limits.

Conclusion

The FinOps Dashboard project demonstrates an effective, automated approach to cloud cost management for AWS Free Tier users. By integrating data collection, storage, and visualization:

- Users can monitor costs in real-time.
- Historical data is preserved for analysis.
- Alerts help prevent accidental billing due to Free Tier overages.

The dashboard combines **automation, visualization, and cost control**, providing a practical solution for cloud cost optimization and FinOps practices.

- ✅ Fully deployable on AWS EC2
- ✅ AWS credentials securely configured
- ✅ Provides actionable insights into cloud spending

T