A close up of a logo

Description automatically generated

Angela Jonyl Reyes  
100989583

COSC 1104 – Assignment 3

1. **Identify the problem**

When you are managing a fleet of EC2 instances hosting a web application. The application experiences fluctuating traffic, and occasionally, some EC2 instances become overwhelmed, leading to slower response times or downtime.

Manually checking the CPU, memory, and network activity for each instance can sometimes be too time-consuming, especially when the architecture consists of dozens of instances.

By using CloudWatch metrics and a script to monitor CPU utilization, we can automate the detection of instances with high CPU usage (e.g., above 80% for 5 minutes). This allows you to quickly identify and scale up resources or investigate performance issues without manually checking each instance.

1. **Solve the Problem Using Python**

By using boto3 to fetch the CPU utilization data for the EC2 instances, we no longer need to check each instances’ performance in the AWS console. It connects to AWS CloudWatch, which is Amazon's monitoring service for AWS resources. The script fetches CPU utilization data for the last hour, with data points every 5 minutes. Once it retrieves the data, it displays the CPU utilization percentage for each time point over the last hour. Instead of checking each instance’s metrics individually, the script queries CloudWatch for the metrics of multiple EC2 instances (if extended) and outputs the results centrally. In the future, we can expand the script to trigger notifications based on thresholds, where we would be able to arrive to a faster resolution.

References:

* https://fivetran.com/docs/logs/external-logs/cloudwatch/setup-guide
* <https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/GettingSetup.html>
* <https://boto3.amazonaws.com/v1/documentation/api/latest/guide/quickstart.html#installation>
* https://boto3.amazonaws.com/v1/documentation/api/1.35.6/reference/services/cloudwatch.html