Assignment 04: Serverless Framework

# Team Information

Team Name: Bay Area Rockers

Team Members: Shawn Chumbar, Dhruval Shah, Sajal Agarwal

GitHub Repository Link for this Assignment: <https://github.com/schumbar/CMPE272/tree/main/assignment_04>

# Assignment Description

Deploy a simple serverless application. For example, here is one which generates Doge meme images, randomly choosing colors and locations of text, writing an image into an S3 bucket, as described here: https://github.com/iopipe/lambda-workshop  
Edit the code and do fun things! Submit Word document showing your work, and link to github repo with your code, etc..

# Deliverables:

1. server-application: folder containing our serverless application.  
2. server-application/README.md: documentation for our serverless application.  
3. CMPE272\_assignment04.docx: word document containing our assignment submission.

# Instructions:

## Prerequisites:

All instructions related to this section are assuming that you are using macOS.  
1. You must have docker installed on your system.  
2. You must have the AWS CLI installed on your system.  
3. You must have homebrew installed on your system.  
4. You must have the AWS SAM CLI installed on your system.

## Steps:

Please see below for the steps that we performed to create this serverless application.  
1. Create a new serverless application using the AWS SAM CLI. Please note the following choices for the CLI options provided:  
 1. Choose AWS Quick Start Templates  
 2. Choose Hello World Example  
 3. "Use the most popular runtime and package type?" yes  
 4. Would you like to enable X-Ray tracing on the function(s) in your application? [y/N]: N  
 5. Would you like to enable monitoring using CloudWatch Application Insights?  
 For more info, please view https://docs.aws.amazon.com/AmazonCloudWatch/latest/monitoring/cloudwatch-application-insights.html [y/N]: N  
 6. Project name [sam-app]: server-application  
2. Change directory to the newly created server-application directory.  
 cd server-application  
3. Build the application using the SAM CLI: sam build  
4. Deploy the application locally using the SAM CLI: sam local start-api  
5. Verify that the application is running locally by opening a new browser window and navigating to http://localhost:3000/  
6. Make any necessary edits (i.e. add code to retrieve current time and date and return response to user)  
7. Deploy the application to AWS using the SAM CLI: sam deploy --guided

# References

Please see below for the list of references that were used for this assignment:  
1. ChatGPT-4  
2. https://www.youtube.com/watch?v=MipjLaTp5nA&t=261s  
3. https://www.youtube.com/watch?v=fEZE3rm8Ma8&t=518s  
4. <https://www.youtube.com/watch?v=4bz8Rz2RHws>

# Please see next page for images:

# Please see below for images related to this assignment:

Deployment via AWS SAM CLI:

A screenshot of a computer screen

Description automatically generated

Webpage Output:

A screenshot of a computer

Description automatically generated

AWS Cloud Formation:

A screenshot of a computer

Description automatically generated

AWS Lambda Test Event:

A screenshot of a computer

Description automatically generated

AWS Lambda Test Event Output:

A screenshot of a computer

Description automatically generated