Certification Project

Banking and Finance Domain

FundMe is a Global leading Banking and Financial services provider based out of Germany. The company offers products and services like Banking, Funds Management, Loans, Debit Cards and Credits Cards, Investment Banking etc. Initially the company was using a Monolithic application architecture, As the company grown, It started facing difficulties in managing the application infrastructure and application deployments and Scaling of application when the traffic load increases.

FundMe has decided to opt for microservice architecture for its applications and decided to go DevOps by implementing necessary automations using CICD. **FundMe** has decided to use AWS as primary cloud services provider to create servers, databases and application deployments.

The company's goal is to deliver the product updates frequently to production automatically with High quality & Reliability. They also want to accelerate software delivery speed, quality and reducing feedback time between developers and testers.

Currently, they are facing following problems, because of various technologies involved in the project.

- ✓ Building Complex Monolithic Application is difficult.
- ✓ Manual efforts to test various components/modules of the project
- ✓ Incremental builds are difficult to manage, test and deploy.
- ✓ It was not possible to scale up individual modules independently.
- ✓ Creation of infrastructure and configure it manually is very time consuming
- ✓ Continuous manual monitoring the application is quite challenging.

In order to implement a POC, you are requested to develop a Java maven spring boot microservice using spring boot and in memory h2 database.

1. a microservice which exposes below mentioned endpoints as APIs and uses pre configured AWS RDS – mysql database to store the data.

a. /createAccount (HTTP Method : POST) (Request Body : JSON)

b. /updateAccount/{account no.} (HTTP Method : PUT) (Request Body : JSON)

c. /viewPolicy/{account no.} (HTTP Method : GET) (No Request Body)

d. /deletePolicy/{account no.} (HTTP Method : DELETE) (No Request Body)

2. Write necessary Junit testcase.

3. Generate HTML report using TestNG.

4. Push your code into your GitHub Repository.

Note: Preload some data into the database.

Later, you need to implement Continuous Integration & Continuous Deployment using following tools:

- ✓ Git - For version control for tracking changes in the code files
- ✓ Maven – For Continuous Build
- ✓ Jenkins - For continuous integration and continuous deployment
- ✓ Docker - For deploying containerized applications
- ✓ Kubernetes – for running containerized application in managed cluster.
- ✓ Ansible - Configuration management tools
- ✓ Terraform - For creation of infrastructure.
- ✓ Prometheus and Grafana – For Automated Monitoring and Report Visualization

This project will be about how to test the services and deploy code to dev/stage/prod etc, just on a click of button.

Business challenge/requirement

As soon as the developer pushes the updated code on the GIT master branch, the code should be checked out, compiled, tested, packaged and containerized. A new test-server should be provisioned using terraform and should be automatically configured using Ansible with all the required software's and as soon as the server is available, the application must be deployed to the test-server automatically.

The deployment should then be tested using a test automation tool, and if the build is

Successful, Prod server must be configured with all the software it should be pushed to the prod server. All this should happen automatically and should be triggered from a push to the GitHub master branch. Continuous monitoring server must be configured to monitor the test as well as prod server using Prometheus and Grafana should be configured to display a dashboard with following metrics.

1. CPU utilization

2. Disk Space Utilization

3. Total Available Memory

Link for the Solution of **FundMe** project code is attached below. Use it to validate your solution.

Note: To have a detailed information about running the application and exposed APIs, Input/Output format, Refer to the README.md in the GitHub repository.