TECHNOLOGY

Post Graduate Program in DevOps

ASI Insurance

Objectives

To create a microservice application architecture for an Insurance company through DevOps pipeline and deployment on Docker.

Problem Statement and Motivation

ASI Insurance is facing challenges in improving the SLA to its customers due to its organizational growth and existing monolithic application architecture. It requires transformation of the existing architecture to a microservice application architecture, while also implementing DevOps pipeline and automations.

The successful completion of the project will enable ASI Insurance to improve its overall application deployment process, enhance system scalability, and deliver better products and services to its customers.

Industry Relevance

Skills used in the project and their usage in the industry are as below:

- **Jenkins:** It is used for continuous integration and continuous delivery. In this project, Jenkins can be used to automate the entire software delivery process.
- **GitHub:** It is a web-based hosting service that provides a Git repository management system. In this project, it is used to manage the source code of the microservices and facilitating the development process.
- **Docker Hub:** It is a cloud-based repository for storing, managing, and sharing Docker container images. In this project, it is used to store the Docker image and use to deploy the same on cloud.
- **Amazon Web Service:** It is a cloud platform that provide necessary computation and storage resources to host an application. In this project, we will deploy our application on an EC2 instance.

Task (Activities)

1. Create the Dockerfile, Jenkinsfile, Ansible playbook, and the source file of the static website
2. Upload all the created files to GitHub
3. Go to the terminal and install NodeJS 16
4. Open the browser and access the Jenkins application
5. Create Jenkins pipeline to perform CI/CD for a Docker container
6. Create Docker Hub Credentials and other necessary pre-requisites before running build
7. Set up Docker remote host on AWS and configure deploy stage in pipeline
8. Execute Jenkins Build
9. Access deployed application on Docker container

Project Reference

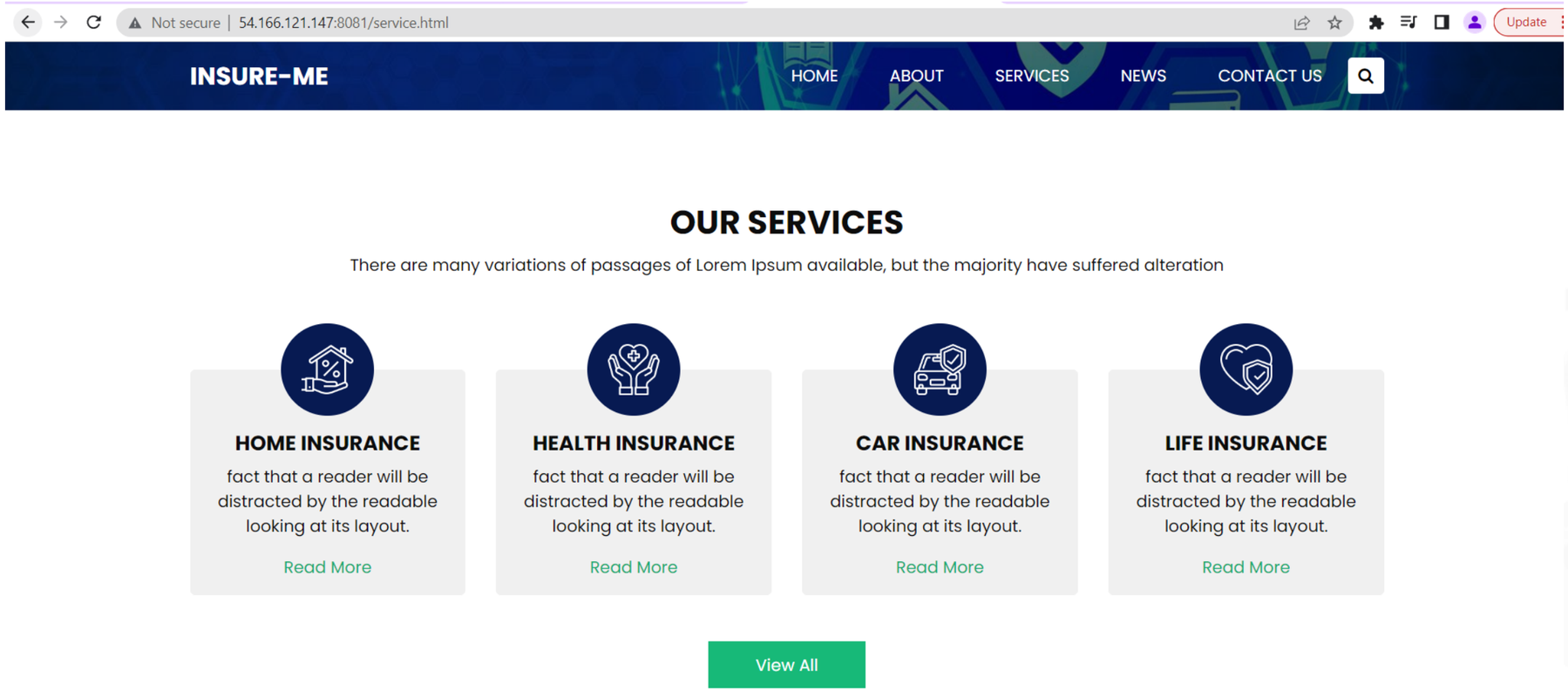
- **Task 1 and 2:** Course 1, Version Control System, Lesson-end project
- **Task 3 and 4:** Course 1, CI/CD with Jenkins, Lesson-end project
- **Task 4, 5, and 6:** Course 4, Docker Certified Associate (DCA) Training, Assisted Practices
- **Task 7:** Course 2, Terraform Loops Built in Functions Provisioners, Assisted Practices
- **Task 8:** Course 1, CI/CD with Jenkins, Assisted Practices
- **Task 9:** Course 4, Docker Certified Associate (DCA) Training, Assisted Practices

Output Screenshot

Main page:

Output Screenshot

Service section page:



TECHNOLOGY

Thank You

simplilearn