**Make an E-commerce Website for Sporty Shoes.**

**Developed By: Pai Najranabanu Yakub**

**Project objective**:

* As a Full Stack Developer, you have to build a CI/CD pipeline to demonstrate continuous deployment and host the application on AWS EC2 instance.

**Background of the problem statement:**

* As the project is in the final stage, management has asked you to automate the integration and deployment of the web application. You are required to set up an environment where the application will be hosted and accessed by users. The source code is supposed to be fetched from a GitHub repository.

**You must use the following:**

* Eclipse
* GitHub
* Jenkins
* AWS EC2/ Virtual machine

**Following requirements should be met:**

* A part of the source code should be tracked on the GitHub repository. You need to document the tracked files that are ignored during the final push to the GitHub repository.
* The submission of your GitHub repository link is mandatory. In order to track your task, you need to share the link of the repository in the document
* The step-by-step process involved in completing this task should be documented.

**Important URLs: -**

* Github Link : <https://github.com/NajranaPai/jenkins-test-main.git>
* Jenkins dashboard job url:- http://18.117.254.238:8080/job/simplilearn-phase5/ Credentials for Jenkins view only user: username= viewonly, password: 12345678
* Host server app [URL : -](url:-) <http://18.189.180.153:8080/account> (get)

<http://18.189.180.153:8080/account> (post)

**Steps: -**

**Created an EC2 instance and started Jenkins on it:**

* An ec2 instance which has configured security group that allows users to access it through http protocol on port 80 and then install Java, Jenkins, maven, git and docker on it and make it accessible on port 8080 (default port of Jenkins server).

**Made a springboot application:**

* A springboot application that has two endpoints.

1. Get: - /account to get the list of all the accounts
2. Post:- /account to add an account to the list

* It has Dockerfile with required dependency JDK-11.

**Upload it on Github:**

* Tracked the springboot application with version controlling system i.e. Git and then connected it with remote repository on Source code management system i.e. Github in the main branch.

**Make an host ec2 instance:**

* An ec2 instance which has configured security group that allows users to access it through http protocol on port 80 and install docker on it to run docker container.

**Created a Jenkins Pipeline job:**

Created a pipeline job with steps:

1. SCM pull: pulls the source code from Github of the created springboot app.
2. Maven package: packages the source code pulled by step 1.
3. Docker Build : makes an image of the package created by Maven package with dockerfile in that springboot application.
4. Docker push: push the created image to dockerhub.
5. SSH login to Host ec2 instance through Jenkins credentials that has private key for the SSH.
6. Pull the docker image uploaded by Docker push step and run it as docker container and while running map the internal 8080 port (default springboot app server port) to that ec2 instance’s port 8080.