**Project Description:**

* I developed a front end angular web application which has a student form and connects with backend using Rest Web Services.
* I have created the mappings for Rest Api’s using Springboot as a backend. We have used JPA to connect with Database Amazon.
* I established a CI/CD pipeline that include a GitHub as a source code repository and the build automation tool Jenkins, and the Kubernetes platform for the automated build and deployment of our application.

**Instruction Steps:**

**1. Creating a Front End Angular Application**

* Download all the required packages like npm, node to download the Angular CLI.
* Now download angular CLI and make new project using ng new command
* Next create the necessary components and routing files.
* Write a service typescript file which connects the angular application to the backend.
* Use ng serve command to check if angular application works locally.

2. **Creating SpringBoot Backend Application**

* Install Spring tool Service and any version of java JDK.
* Create new springboot project in STS and choose the JPA and MYSQL connector as dependencies.
* Create three files: controller, model, repository along with necessary annotations which make the backend calls.
* Setup tomcat server and make sure application runs locally.

3.**Dockerising application:**

* To dockerize a springboot application use mvn install for the application and then write a docker file that copies .jar file to specified path.
* Use docker build command to make a image of application and then push the image to Docker hub account.
* To dockerize a angular application we need to use ngnix and use –prod to make a dist folder that keeps all the angular application file. Checkout the docker file int the angular folder.
* Use docker build command to make a image of application and then push the image to Docker hub account.

4.**Deploying Images to Kubernetes Cluster:**

* Write deployment.yaml and service.yaml files for both frontend and backend application, so that we can deploy and start service in kubernetes cluster.
* Access the external IP address that we get when we run command “kubectl get svc”
* Make sure both front and backend application are accessible and work together using Rest Api’s.
* Finally Autoscale both of the deployments using a loadbalancer by “kubectl hpa” command

5. **Setting up Jenkins**

We need to set up a pipeline that checks any changes to GitHub and build using the Jenkins configure Build commands

* In the Jenkins UI, click on ‘New Items’, enter a name and then click on Freestyle project
* Set up and copy your repository and GitHub URL here.
* In build Triggers set it to GitHub hook trigger and Poll SCM
* Add \* \* \* \* \* to the Schedule for time.
* In build execute shell commands use mvn clean install and put your URL in docker build and publish.

6.**Github Webhook Trigger**

* Create a repository on Github.
* Install git on your local machine.
* Git commit & push the code to Github.
* Add your Dockerfile to Github.
* Go to repository and click on settings.
* Click on webhook and then Add webhook. 8. Provide your jenkins url and select Just the push event. It will send POST request whenever there is any change in repository.

**References:**

* Angular: <https://angular.io/docs>
* Springboot: https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/
* Ranchers Cluster buildup: https://www.youtube.com/watch?v=jF8jCg1WPwo
* Jenkins Installation: https://www.youtube.com/watch?v=twkRYe6m8DQ
* Github webhook trigger: https://www.youtube.com/watch?v=xlvSjDHvUwU