# **full stack engineering**

# **Couse 5- Agile & DevOps**

# **Assignment - CICD Implementation (pipeline)**

Submission Date: June 30th, 2023, 11.59 PM

Assignment Submission Mode: Canvas LMS- File Upload

Weightage: 40%

**Objective:**

To understand and implement CICD pipeline for VTS / Pharmacy (course -2/3 assignment) using Jenkins.

**Problem Statement:**

Jenkins is a popular open-source tool to perform continuous integration and build automation.

You are expected to continue CICD implementation for same application i.e. VTS / Pharmacy.

As a part of this assignment, you are required to implement CICD pipeline for Front End and Back End Code Base with the help of below instructions.

[\*\*Note: Continue the implementation by same Group, group details are mentioned in the excel sheet]

## **1: Cicd for front-end code BASE Instructions**

1. **Continuous Integration:**

* Create/Use a git repository of an existing application Learning Management System Front End Code Base
* Through Jenkins fetch the code from GitHub and map with the job enabled for a specific task execute the next phase of action

1. **Build Automation:**

* Initiate build phase by compiling code using Jenkins & perform respective unit test
* This phase will generate the required artefact to the Jenkins workspace folder

1. **Continuous Code Analysis:**

* Integrate SonarQube with Jenkins
* Perform continuous code analysis using SonarQube scanner for ReactJS codebase of Learning Management System

1. **Continuous Testing and Continuous Delivery:**

* Deliver the code to the staging environment (either create the environment in your local system or one can use AWS cloud service)
* Perform functional testing using Selenium

1. **Continuous Deployment:**

* After the successful delivery to staging, deploy the same code to the production
* Using plugin integrate the AWS S3 bucket with Jenkins to copy the artefact
* In this case the production environment should be AWS S3 bucket

1. **Continuous Monitoring:**

* After the successful deployment to production, implement the continuous monitoring through Kubernetes.
* Use Metrics Server and Basic Logs to capture the system health and app health and publish them.

## **2: CICD for Back-end code base insturctions:**

1. **Continuous Integration:**

* Create/Use a git repository of an existing application Learning Management System Back End Code Base
* Through Jenkins fetch the code from GitHub and map with the job enabled for a specific task execute the next phase of action

1. **Continuous Deployment:**

* Using Jenkins perform unit test, and under build steps, execute the deployment script
* Deploy the back-end code base to AWS EC-2 instance.

1. **Continuous Monitoring:**

* After the successful deployment to production, implement the continuous monitoring through Kubernetes.
* Use Metrics Server and Basic Logs to capture the system health and app health and publish them.

**Note:**

One can implement the CICD by either creating multiples jobs/project in Jenkins or using Jenkins pipeline.

## **3: Submission Instructions**

1. Upload the complete code to the github repository.
2. Create a word document, with the description CICD implemented, snapshots of Jenkins action performed, Build actions performed and monitoring actions performed.
3. Upload the word document to the Canvas LMS
4. While naming the files use your Group ID and your name for identification purposes
5. Academic Honesty: You are welcome to discuss with peers and refer the internet in order to better understand the concept, but you may not share code or do not do a verbatim copy from the internet.

**Files to Be Submitted**

1. Complete code in Github- Link to be shared
2. Description document to be uploaded to Canvas

It should contain

* Description of implemented CICD for Front End and Back End Code Base
* Snapshots of Jenkins UI and Kubernetes UI where you performed different action to complete the required steps
* Assumptions made
* Github Link
* Execution instructions and deployment script used for back-code deployment

## **4: Weightage:**

**Total: 25 Points**

**1): Implementation of CICD for Front-End code & Demo: 25 %**

**2): Implementation of CICD for Back-End code & Demo: 15%**

## **5: Evaluation**

After the completion of deadline, Assignment discussion will be scheduled, where you can showcase the demonstration of implemented CICD.

## **6: Notes:**

* This is a take-home assignment to be carried out by each group independently
* In case of any further queries, use discussion forums, or reach out to me at [bharani.k@wilp.bits-pilani.ac.in](mailto:bharani.k@wilp.bits-pilani.ac.in?subject=FSE%20C5%20Cohort1%20Assigment)