

**University of Central Punjab**

**Faculty of Information Technology**

**Department of Software Engineering**

**Software Construction and Development, Fall 2023**

**Assignment No. 04**

**Topic: Collaborative Software Development using a Remote Repository**

**CLO Covered In This Assignment: CLO: 03**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **01** | **Student Name:** | **M. Ibrahim** | **Roll Number:** | **L1s21bsse0115** |
| **02** | **Student Name:** | **M. Haseeb** | **Roll Number:** | **L1s21bsse0099** |
| **03** | **Student Name:** | **Sarib Butt** | **Roll Number:** | **L1f20bsse0317** |
| **04** | **Student Name:** |  | **Roll Number:** |  |
| **Total Marks** | | **50** | **Obtained Marks** |  |

**Important Instruction**

1. This is **a group assignment**. **Each group will consist of three members**.
2. No assignment will be accepted if work done not in groups.
3. Only the repository owner will submit the assignment on the portal.
4. Each question is from the topic we have discussed during class lecture.
5. **Copying** from any source **shall result in zero marks**.

**Good Luck**

1. Explain what is meant by continuous integration and continuous delivery/deployment. Write the detailed note on CI/CD Pipelines. Explain the workflow of these pipelines. Give a small introduction of the popular tools which are used for setting up these pipelines. **[05+05+05+10=25 Marks]**

Continuous Integration (CI) and Continuous Delivery/Continuous Deployment (CD) are practices in software development aimed at improving the development and release processes by automating and streamlining various stages. These practices are commonly referred to as CI/CD.

**1. Continuous Integration (CI):**

Definition: CI is the practice of frequently integrating code changes from multiple contributors into a shared repository. The primary goal is to detect and address integration issues early in the development process.

Workflow: Developers submit their code changes to a version control system (e.g., Git), triggering an automated build process that compiles the code, runs tests, and produces artifacts.

**2. Continuous Delivery (CD):**

Definition: CD extends CI by automating the delivery process, ensuring that the software is always in a deployable state. This includes activities like automated testing, deployment to staging environments, and potentially manual approval steps before production deployment.

Workflow: After successful CI, the software is automatically deployed to a staging environment. If all tests pass and any required approvals are obtained, it can be further deployed to production.

**3. Continuous Deployment (CD):**

Definition: CD takes automation a step further by automatically deploying code changes to production without manual intervention, once they pass all automated tests and checks.

Workflow: Similar to Continuous Delivery, but the deployment to production is fully automated without requiring manual approval.

**CI/CD Pipelines:**

CI/CD pipelines are sets of automated processes that facilitate the steps from code integration to deployment. The typical workflow of a CI/CD pipeline includes:

**1. Code Commit:**

Developers commit their code changes to a version control system.

**2. Automated Build:**

The CI/CD pipeline triggers an automated build process, compiling the code and generating executable artifacts.

**3. Automated Testing:**

The pipeline runs automated tests (unit tests, integration tests, etc.) to ensure code quality and identify any issues.

**4. Deployment to Staging:**

If all tests pass, the software is deployed to a staging environment for further testing and validation.

**5. Manual Approval (for CD):**

In the case of Continuous Delivery, there might be a manual approval step before deploying to production.

**6. Deployment to Production (for CD):**

For Continuous Deployment, if all criteria are met, the pipeline automatically deploys the code changes to the production environment.

**Popular CI/CD Tools:**

**1. Jenkins:**

An open-source automation server that supports building, deploying, and automating any project.

**2. GitLab CI/CD:**

Integrated CI/CD functionality within the GitLab platform.

**3. Travis CI:**

A cloud-based CI/CD service that integrates with GitHub repositories.

**4. CircleCI:**

A cloud-based CI/CD platform with easy configuration and integration capabilities.

**5. TeamCity:**

A CI/CD server developed by JetBrains with support for various build and deployment scenarios.

**6. GitHub Actions:**

A CI/CD and automation service tightly integrated with GitHub repositories.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Q2).** With the help of examples using git and git hub, explain each concept given below? You need to paste the screen shorts for each example you create.

**[25 Marks]**

1. Pull Requests and Code Reviews
2. Fast Forward Merge
3. 3-Way Merge
4. Merge Squash
5. Rebase