ASSIGNMENT-1

Eshaan Bagaria

4NI19IS030

ISE “A” 7 SEM

1. What is CI/CD?

CI/CD stands for Continuous Integration / Continuous Deployment. It is a software engineering practice that aims to minimize the time between writing code and actually releasing it, by replacing the traditional practice of integrating code at the end of a development cycle with smaller, more frequent integrations. This allows for more rapid delivery of features, higher quality software, and better collaboration among team members.

In a CI/CD workflow, code changes are automatically built, tested, and deployed to production, following pre-defined rules and procedures. This typically involves using a version control system (such as Git) to track code changes, and automated build and deployment tools to build and deploy code changes. The end result is a more streamlined software development process, with less manual effort required to build, test, and release code.

2. What are feature flags and how it is used?

Feature flags (also called feature toggles, feature switches, or feature flippers) are a way to enable or disable specific features in a software application, without requiring a full deploy. This can be useful in a variety of situations, such as:

* Testing new features: Feature flags can be used to roll out new features to a subset of users, allowing you to test the feature in a production environment before making it available to everyone.
* A/B testing: Feature flags can be used to run A/B tests on different features, allowing you to compare the performance of two different versions of a feature.
* Toggling features on or off: Feature flags can be used to quickly turn a feature on or off, without requiring a deploy. This can be useful if you need to disable a feature due to a bug or other issue.

Feature flags are typically implemented using a configuration file or a database, which specifies which features are enabled or disabled. The code for the feature checks the flag before executing, so that the feature is only run if the flag is set to "enabled." This allows you to easily control which features are available to users, without requiring a full deploy of the codebase.

3. Explain CI/CD pipeline with block diagram? Here is a simple CI/CD pipeline that shows the major stages involved in a typical CI/CD workflow:

1. **Code changes:** Developers write code and commit their changes to a version control system (such as Git).
2. **Build:** The code changes are automatically built by a build server (such as Jenkins or Azure DevOps). The build process typically includes tasks such as compiling the code, running tests, and packaging the code into a deployable artifact (such as a Docker container).
3. **Test:** The code changes are automatically tested by a testing server (such as Jenkins or Azure DevOps). This can include unit tests, integration tests, and other types of tests to ensure the code is working correctly.
4. **Deploy:** If the build and tests are successful, the code changes are automatically deployed to a staging environment (such as a test or staging server). This allows the code changes to be reviewed and tested in a production-like environment before being released to users.
5. **Release:** If the code changes pass testing in the staging environment, they can be released to production. This typically involves deploying the code changes to a production server and making them available to users.

Here is a simple block diagram that illustrates this process:

Copy code

Code changes --------------------------> Build --------------------------> Test --------------------------> Deploy --------------------------> Release version control system build server testing server staging environment production environment

