

Assignment 1

What is CI/CD?

CI/CD stands for "continuous integration/continuous delivery". It is a software development practice that involves continuously integrating and delivering code changes.

In a CI/CD workflow, developers write code and commit their changes to a shared code repository. This triggers an automated process that builds, tests, and deploys the code changes.

The goal of CI/CD is to enable rapid, automated delivery of code changes, while ensuring that the code remains stable and of high quality. By automating the build, test, and deployment process, teams can reduce the time and effort required to release new software updates, and can deploy changes more frequently and with fewer errors.

- Continuous integration (CI) refers to the practice of regularly integrating code changes into a shared code repository. This helps to catch integration issues early, as developers can see how their code changes interact with the rest of the codebase. CI is typically automated, with tools that build, test, and validate code changes as they are committed.
- Continuous delivery (CD) refers to the practice of automatically building, testing, and deploying code changes. With CD, code changes are automatically built and tested, and if they pass the tests, they are automatically deployed to production. CD helps to ensure that code changes can be rapidly and reliably delivered to users, without the need for manual intervention.
- Together, CI and CD form a powerful workflow that helps teams to develop and deliver software more quickly and reliably. By automating the build, test, and deployment process, teams can reduce the time and effort required to release new software updates, and can deploy changes more frequently and with fewer errors. This can help teams to respond more quickly to changing business needs and to deliver new features and improvements to users more rapidly.

What is CI/CD pipeline? With Examples.

In a CI/CD workflow, the process of building, testing, and deploying code changes is typically automated using a "pipeline". A pipeline is a series of automated steps that are executed in sequence, each step building upon the output of the previous step.

The goal of a CI/CD pipeline is to automate the build, test, and deployment process, so that code changes can be rapidly and reliably delivered to users. This helps teams to respond more quickly to changing business needs and to deliver new features and improvements to users more rapidly.

Here is an example of a simple CI/CD pipeline:

1. A developer writes code and commits the changes to a shared code repository.
2. The CI/CD pipeline is triggered, and the code is automatically built.
3. The code is automatically tested to ensure it is functional and does not introduce any new errors.
4. If the tests pass, the code is automatically deployed to a staging environment, where it can be tested further by QA personnel.
5. If the code passes additional testing in the staging environment, it is automatically deployed to the production environment, where it is made available to users.

In this example, the pipeline consists of four automated steps: build, test, deploy to staging, and deploy to production. Each step is executed automatically, based on the output of the previous step. This helps to ensure that code changes are rapidly and reliably delivered to users, while also ensuring that the code is stable and of high quality.

Uses of CI/CD pipelining

- Deploying code changes to web applications: CI/CD pipelines can be used to automatically build, test, and deploy code changes to web applications. This can help teams to quickly and reliably deliver new features and improvements to users.
- Deploying code changes to mobile apps: CI/CD pipelines can be used to automatically build, test, and deploy code changes to mobile apps. This can help teams to quickly and reliably deliver updates to users through app stores.
- Deploying code changes to cloud infrastructure: CI/CD pipelines can be used to automatically build, test, and deploy code changes to cloud infrastructure, such as infrastructure-as-code templates or configuration files for cloud services. This can help teams to quickly and reliably update their cloud environments.
- Deploying code changes to on-premises systems: CI/CD pipelines can be used to automatically build, test, and deploy code changes to on-premises systems, such as servers or networking equipment. This can help teams to quickly and reliably update their on-premises environments.

Diagram of CI/CD Pipelining

