**What is CI/CD?**

**“CI/CD stands for Continuous Integration and Continuous Delivery (or Continuous Deployment), it is about how an Integrated code on a shared repository is used to release software to production multiple times a day with the help of automation.”**

**Continuous integration**

It is a software development method where members of the team can integrate their work at least once a day. In this method, every integration is checked by an automated build to search the error.

**Continuous delivery**

It is a software engineering method in which a team develops software products in a short cycle. It ensures that software can be easily released at any time.

**Continuous deployment**

It is a software engineering process in which product functionalities are delivered using automatic deployment. It helps testers to validate whether the codebase changes are correct, and it is stable or not.

**What is a CI/CD pipeline?**

A CI/CD pipeline automates the process of software delivery. It builds code, runs tests, and helps you to safely deploy a new version of the software. CI/CD pipeline reduces manual errors, provides feedback to developers, and allows fast product iterations.

## Stages of a CI/CD pipeline

### Source Stage

In the source stage, CI/CD pipeline is triggered by a code repository. Any change in the program triggers a notification to the CI/CD tool that runs an equivalent pipeline. Other common triggers include user-initiated workflows, automated schedules, and the results of other pipelines.

### Build Stage

This is the second stage of the CI/CD Pipeline in which you merge the source code and its dependencies. It is done mainly to build a runnable instance of software that you can potentially ship to the end-user.

Programs that are written in languages like C++, Java, C, or Go language should be compiled. On the other hand, JavaScript, Python, and Ruby programs can work without the build stage.

Failure to pass the build stage means there is a fundamental project misconfiguration, so it is better that you address such issue immediately.

### Test Stage

Test Stage includes the execution of automated tests to validate the correctness of code and the behaviour of the software. This stage prevents easily reproducible bugs from reaching the clients. It is the responsibility of developers to write automated tests.

### Deploy Stage

This is the last stage where your product goes live. Once the build has successfully passed through all the required test scenarios, it is ready to deploy to live server.

**Example of CI/CD Pipeline**

* **Source Code Control:** Host code on GitHub as a private repository. This will help you to integrate your application with major services and software.
* **Continuous integration:** Use continuous integration and delivery platform CircleCI and commit every code. When the changes notify, this tool will pull the code available in GitHub and process to build and run the test.
* **Deploy code to UAT:**Configure CircleCI to deploy your code to AWS UAT server.
* **Deploy to production:**You have to reuse continuous integration steps for deploying code to UAT.

**Advantages of CI/CD pipelines**

* Builds and testing can be easily performed manually.
* It can improve the consistency and quality of code.
* Improves flexibility and has the ability to ship new functionalities.
* CI/CD pipeline can streamline communication.
* It can automate the process of software delivery.
* Helps you to achieve faster customer feedback.
* CI/CD pipeline helps you to increase your product visibility.
* It enables you to remove manual errors.
* Reduces costs and labour.

**CI/CD Tools**

**Jenkin**

Jenkins is an open-source Continuous Integration server that helps to achieve the Continuous Integration process (and not only) in an automated fashion. Jenkins is free and is entirely written in Java. Jenkins is a widely used application around the world that has around 300k installations and growing day by day.

**Features:**

* Jenkin will build and test code many times during the day.
* Automated build and test process, saving timing, and reducing defects.
* The code is deployed after every successful build and test.
* The development cycle is fast.