**What is CI/CD?**

CI/CD is a method to frequently deliver [applications](https://www.redhat.com/en/topics/cloud-native-apps) to customers by introducing [automation](https://www.redhat.com/en/topics/automation) into the stages of [application development](https://www.redhat.com/en/topics/cloud-native-apps/why-choose-red-hat-cloud-native). The main concepts attributed to CI/CD are continuous integration, [continuous delivery](https://www.redhat.com/en/topics/devops/what-is-continuous-delivery), and continuous deployment. It is a solution to the problems [integrating](https://www.redhat.com/en/topics/integration) new code can cause for development and operations teams.

**What are the features of modern CI/CD tools?**

1. Docker-based Architecture Right From the Start
2. Cloud and Version Control Agnosticism
3. Pipeline Creation with Standardized Definitions
4. Graphical Pipeline View
5. Parallel Steps
6. Standardized Plugin Mechanism (Docker-based)
7. Configuration Options Through Both Code and UI
8. Reusable Pipelines for Microservices
9. Live Pipeline Debugging with Breakpoints

10.Native Support for Kubernetes, Helm, and Docker

11.Saas, On-prem, and Hybrid Installation Methods

12.Zero Config, Distributed Caching

13.Monorepo Support

**What is CI/CD pipeline? Explain with block diagram.**

A continuous integration and continuous deployment ([CI/CD](https://www.redhat.com/en/topics/devops/what-is-ci-cd)) pipeline is a series of steps that must be performed in order to deliver a new version of software. CI/CD pipelines are a practice focused on improving software delivery throughout the software development life cycle via automation.

A pipeline is a process that drives software development through a path of building, testing, and deploying code, also known as CI/CD. By automating the process, the objective is to minimize human error and maintain a consistent process for how software is released. Tools that are included in the pipeline could include compiling code, unit tests, code analysis, security, and binaries creation. For containerized environments, this pipeline would also include packaging the code into a container image to be deployed across a hybrid cloud.

By automating CI/CD throughout development, testing, production, and monitoring phases of the software development lifecycle, organizations develop higher quality code, faster. Although it’s possible to manually execute each of the steps of a CI/CD pipeline, the true value of CI/CD pipelines is realized through automation.



