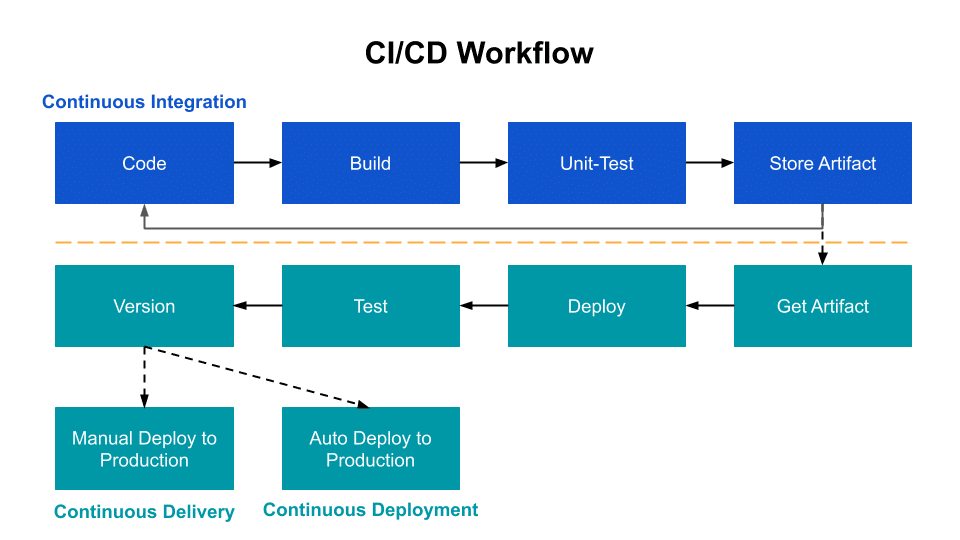
CI/CD PIPELINE

The CI/CD pipeline is one of the best practices software development and devops teams can implement, as it allows them to deliver code changes more frequently and reliably.

The CI/CD pipeline uses test automation to identify potential issues earlier, push code changes to different environments and deliver applications to production environments. Test automation assesses anything from performance to API and security, and it is an essential element for quality control of the pipeline.



1) Use cases of CI/CD

* Push small code changes, frequently
* Build tests
* Fix the build before pushing new code
* Use both unit and integration tests
* Continuously improve the process
* Automate all the things
* Continuously improve the process

2) Advantages of using CI/CD

* Automated testing enables continuous delivery, which ensures software quality and security and increases the profitability of code in production.
* CI/CD pipelines enable a much shorter time to market for new product features, creating happier customers and lowering strain on development.
* The great increase in overall speed of delivery enabled by CI/CD pipelines improves an organization’s competitive edge.
* Automation frees team members to focus on what they do best, yielding the best end products.
* Organizations with a successful CI/CD pipeline can attract great talent. By moving away from traditional waterfall methods, engineers and developers are no longer bogged down with repetitive activities that are often highly dependent on the completion of other tasks.

3) Name all major tools available for CI/CD

* Jenkins
* CircleCI
* TeamCity
* Bamboo
* GitLab
* Buddy
* Travis CI
* Codeship

4) Compare all tools which you mentioned in point 3

**Jenkins key features:**

* Easy installation and upgrade on various OSs
* Simple and user-friendly interface
* Extensible with huge community-contributed plugin resource

**CircleCI key features:**

* Integrates with Bitbucket, GitHub, and GitHub Enterprise
* Runs builds using a container or virtual machine
* Easy debugging

**TeamCity key features:**

* Provides multiple ways to reuse settings and configurations of the parent project to the subproject
* Runs parallel builds simultaneously on different environments
* Enables running history builds, viewing test history reports, pinning, tagging, and adding builds to favorites

**Bamboo key features:**

* Supports up to 100 remote build agents
* Run batches of tests in parallel and get feedback quickly
* Creates images and pushes into a registry

**GitLab key features:**

* View, create and manage codes and project data through branching tools
* Design, develop, and manage codes and project data from a single distributed version control system, enabling rapid iteration and delivery of business values
* Provides a single source of truth and scalability for collaborating on projects and code

**Buddy key features:**

* Easy to customize Docker-based images as a test environment
* Smart change detection, state-of-the-art caching, parallelism, and all-around optimizations
* Create, customize, and reuse builds and test environments

**Travis CI key features:**

* Quick setup
* Live build views for GitHub projects monitoring
* Pull request support

**Codeship key features:**

* Integrates with any tools, services, and cloud environments of choice
* Easy to use. Provides fast and thorough developer support.
* Gets the builds and deployments work faster with CodeShip’s turnkey environment and simple UI