

# Deployment Pipeline

The primary goal of the Development Pipeline (DP) are used to streamline software development and release process while ensuring that code changes are thoroughly tested and verified before seeing the real-world. Bellow is a typical DP:

1. **Version Control System (VCS):**
  - Use of a version control system (like Git or CVS) where developers store and manage the source code for the application. Changes are made in feature branches or development branches.
2. **Continuous Integration (CI):**
  - Developers commit their code changes to the VCS, triggering a continuous integration process.
  - In CI, automated build and test processes are initiated to verify that the codebase is still functional after new changes.
  - Common CI tools include Jenkins, Travis CI, CircleCI, and GitLab CI/CD.
3. **Build Stage:**
  - The CI system compiles, builds, and packages the code into a deployable artifact (like binaries, container images, or deployment packages).
4. **Automated Testing:**
  - Various types of tests are performed and may include unit tests, integration tests, regression tests, security scans, and performance tests.
5. **Quality Assurance (QA) Environment:**
  - After successful testing in the CI environment, code is deployed to a QA environment that closely resembles the production environment.
  - Additional testing, such as user acceptance testing (UAT), is carried out in this environment to catch issues that may not be apparent in the CI environment.
6. **Staging Environment:**
  - If code passes QA, it's deployed to a staging environment that more closely related to the real-world. This is the final testing round for validation.
7. **Manual Testing:**
  - In some cases, manual testing may be required for specific features or aspects of the application that cannot be fully automated.
8. **Approval and Release Candidate:**
  - Stakeholders include: product owners, project managers. They review the changes in the staging environment and if the code base is deemed satisfactory, a release candidate is created for deployment to production.
9. **Production Deployment:**
  - The deployment pipeline may have manual intervention points for final approval before production deployment, or it can be fully automated.
  - New code is deployed in controlled manner (sections) to minimize the impact of any potential issues.
10. **Monitoring and Observability:**
  - After deployment, the application is continuously monitored for performance, errors, and other issues.
  - Monitoring tools, such as Prometheus, Grafana, and ELK Stack, provide insights into the system's health and performance.
11. **Rollback:**
  - If issues are detected in the production environment, a rollback plan is executed to revert to the previous stable version.
12. **Continuous Feedback and Improvement:**
  - Data and feedback from monitoring and user interactions are used to improve the codebase and the deployment process. The process is an ongoing cycle.