CI/CD Pipeline Stages Documentation Example

This sample document describes the typical stages of a Continuous Integration and Continuous Deployment (CI/CD) pipeline. Each stage, its purpose, and expected activities are listed below for reference.

Pipeline Stages

1. Source

The pipeline is triggered by changes in the version control system, such as a push or merge to a specific branch.

- Source repository: Git, SVN, etc.
- Trigger: Commit, pull request, or tag

---

2. Build

Source code is compiled and transformed into deliverable artifacts.

- Dependency installation
- Compilation/transpilation
- Build artifact generation

---

3. Test

Automated tests are run to validate code functionality and quality.

- Unit tests
- Integration tests
- Code linting and static analysis

---

4. Package

The build output is packaged for deployment and distribution.

- Archiving artifacts
- Containerization (Docker, etc.)
- Versioning

---

5. Deploy

The packaged application is deployed to staging, testing, or production environments.

- Deployment to servers or cloud
- Infrastructure automation (IaC)
- Configuration management

---

6. Release

The application is made available to end-users or stakeholders.

- Production rollout

- Notification and changelog generation
- Monitoring activation

---

7. Monitor & Feedback

Ongoing monitoring and collection of feedback from the deployed application.

- Application and infrastructure monitoring
- Error and anomaly detection
- Receiving and analyzing user feedback

Example End-to-End Flow

1. Developer pushes code to the repository.
2. CI/CD pipeline is triggered automatically.
3. The code is built and tested.
4. Successful builds are packaged and deployed to a staging environment.
5. After approval, the code is released to production.
6. Monitoring tools provide feedback for continuous improvement.