# Identify the **Elements** and **components** of CI/CD pipeline in DevOps

# Elements of CI/CD pipeline in DevOps

1. **Version Control System (VCS):**

   A system that manages and tracks changes in the source code.

   Example: Git,

2. **Code Repository:** A centralized place to store and manage the source codes.

   **Examples:** GitHub, Docker

# Cont.’’

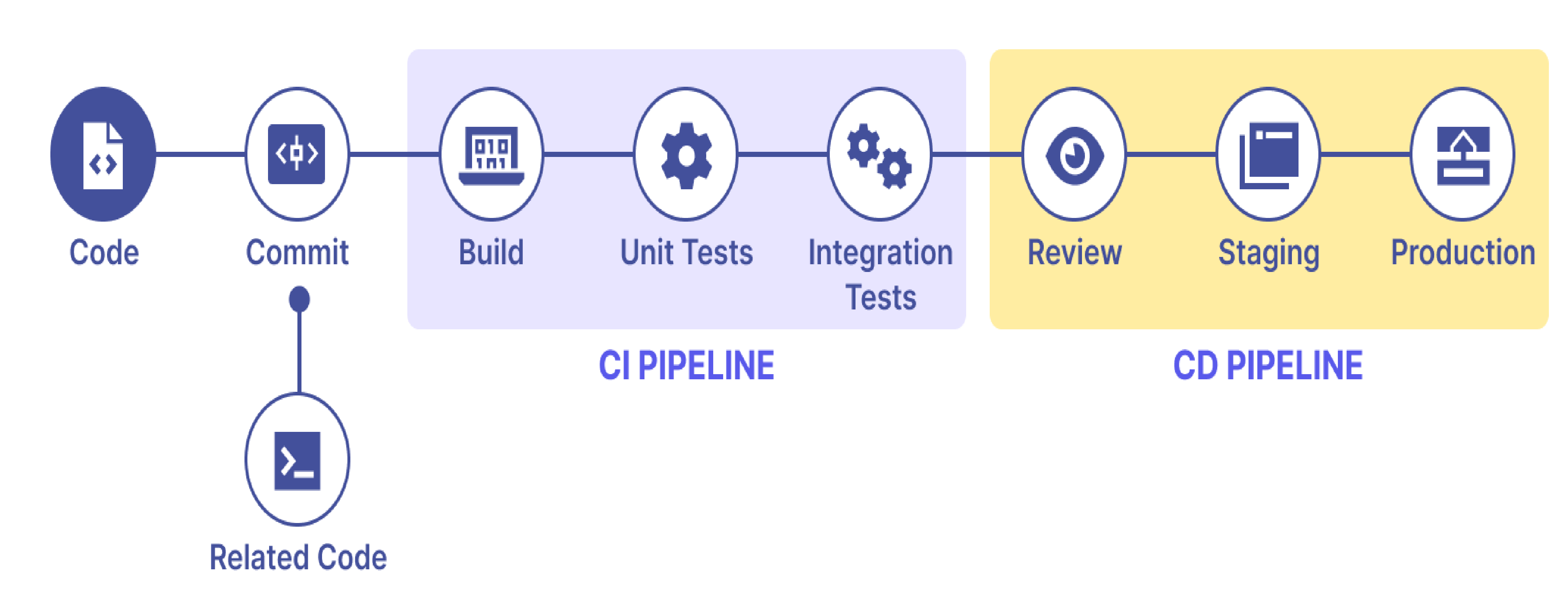
## 3.Continuous Integration (CI):

is a DevOps software development practice where developers regularly merge their code changes into a central repository, after which automated builds and tests are run.

## 4.Continuous Delivery (CD):

is the process of automating build, test, configuration, and deployment from a build to a production environment.

# CI/CD


Code
Commit
Build
Unit Tests
Integration Tests
Review
Staging
Production
Related Code
CI PIPELINE
CD PIPELINE

# Cont’

## 5.Continuous Deployment:

- Automatically deploys code changes to production after passing all tests.

**Purpose:** Accelerates the release process and reduces manual intervention.

## 6. Pipeline Orchestration:

Coordination and management of the CI/CD pipeline steps.

**Examples:** Jenkins Pipeline, GitLab CI/CD.

# Cont’

## 7. Monitoring and Logging:

Tools for real-time tracking of application performance and health.

And also it is used to Identify and resolves issues quickly.

## 8. Feedback Mechanism:

Notifications and feedback loops to inform teams about pipeline status.

## 9. Security Scanning:

Automated security checks for vulnerabilities and compliance.

Examples: SonarQube, OWASP Dependency-Check.

Purpose: Enhances the security of the codebase.

## 10. Collaboration and Communication:

Integration with communication tools (e.g., Slack, Microsoft Teams) to notify team members about build and deployment status.

# Components of CI/CD pipeline in DevOps

## 1. Source Code Repository:

The central location where the source code is stored and version-controlled.

**Example:** Git, Docker.

## 2. Continuous Integration (CI) Server:

A server or service responsible for automating the build and testing processes.

**Examples:** Jenkins, Travis CI, GitLab CI.

# Cont’

## 3. Build Tools:

Software tools responsible for compiling, packaging, and assembling the source code into executable artifacts.

**Examples:** Maven, Gradle.

## 4. Automated Testing Tools:

Tools that execute various types of automated tests (unit, integration, acceptance) to verify the correctness and quality of the code.

**Examples:** JUnit, Selenium, Jest.

# Cont’

## 5. Artifact Repository:

A repository that stores and manages binary artifacts produced during the build process.
**Examples:** Nexus, JFrog Artifactory.

## 6. Continuous Delivery/Deployment Tools:

Tools or scripts responsible for deploying applications to different environments.
**Examples:** Ansible, Puppet, Chef.

# Cont’

## 7.Configuration Management:

Tools that manage and maintain infrastructure and application configuration in a consistent and automated manner.

**Examples:** Terraform, AWS CloudFormation.

## 8. Environment Orchestration:

Tools that manage the deployment and scaling of applications in different environments, especially in containerized environments.

**Examples:** Kubernetes, Docker Swarm.

# Thank You!!