

Chapter 4: CI/CD (Continuous Integration / Continuous Deployment)

CI/CD Concepts

Term	Description
Continuous Integration (CI)	Developers frequently merge code changes into a shared repository. Each merge triggers automated builds and tests. Goal: detect integration issues early.
Continuous Delivery (CD)	Builds on CI. Ensures the software is always in a deployable state. Deployment is manual but ready anytime.
Continuous Deployment	Extends Continuous Delivery. Every change that passes the pipeline is automatically deployed to production without manual intervention.
Traditional vs CI/CD	Traditional deployment is manual and error-prone; CI/CD is automated, faster, and reduces risk through quick feedback cycles.

CI/CD Pipeline Stages

Stage	Purpose
1. Code Commit	Developers commit code to version control (e.g., Git).
2. Build	Code is compiled and converted into executable format.
3. Test	Automated unit/integration tests are run to verify changes.
4. Deploy	Code is deployed to staging/production environments.
5. Monitor	Monitoring tools check app performance and error rates post-deployment.

Popular CI/CD Tools

Tool	Features
Jenkins	Open-source automation server with pipeline scripting.
GitHub Actions	GitHub-native CI/CD automation using YAML workflows.
Azure DevOps	Microsoft's CI/CD platform; integrates well with Azure Cloud.
GitLab CI/CD	Built-in CI/CD system within GitLab for complete DevOps automation.

Benefits of CI/CD

- Faster delivery of features
- Early bug detection
- Better collaboration between Dev, QA, and Ops
- Improved software quality
- Consistent deployments

Traditional vs CI/CD Comparison

Feature	Traditional Approach	CI/CD Approach
Deployment Method	Manual	Automated
Integration Testing	Late in process	Frequent
Error Detection	Post-deployment	Pre-deployment
Time to Release	Longer	Shorter
Rollback Strategy	Manual and risky	Automated and easy

 **Part 1: 10 Multiple Choice Questions (MCQs)**

 **Part 2: Real-World CI/CD Pipeline Examples**

- Jenkinsfile
 - GitHub Actions Workflow
-

 **Part 1: MCQs on CI/CD**

1. What does CI stand for in DevOps?

- A. Continuous Improvement
- B. Continuous Integration
- C. Code Injection
- D. Code Integration

 **Answer:** B

2. What is the main goal of Continuous Integration (CI)?

- A. Automate system monitoring
- B. Avoid merging conflicts by working alone
- C. Frequently integrate code and detect issues early
- D. Deploy directly to production

 **Answer:** C

3. Which of the following best describes Continuous Deployment?

- A. Code is tested and manually deployed
- B. Code is automatically deployed after passing all tests
- C. Code is committed without testing
- D. Deployments are done once per month

 **Answer:** B

4. Which stage of the CI/CD pipeline compiles and prepares the code for testing?

- A. Monitor
- B. Build
- C. Deploy
- D. Commit

 **Answer:** B

5. What is the correct order of a CI/CD pipeline?

- A. Build → Commit → Test → Deploy → Monitor
- B. Commit → Build → Deploy → Test → Monitor
- C. Commit → Build → Test → Deploy → Monitor
- D. Monitor → Test → Build → Commit → Deploy

 **Answer:** C

6. Which CI/CD tool is open-source and highly customizable for setting up pipelines with plugins?

- A. Jenkins
- B. Azure DevOps
- C. GitHub Actions
- D. Bamboo

 **Answer:** A

7. In GitHub Actions, workflows are defined using which format?

- A. XML
- B. JSON
- C. YAML
- D. CSV

 **Answer:** C

8. Which of the following is a key benefit of CI/CD?

- A. Manual release cycles
- B. Slower feedback
- C. Frequent and reliable software releases
- D. Static project documentation

Answer: C

9. What does a failed unit test indicate in the CI process?

- A. Code is ready to deploy
- B. Code merge is successful
- C. There is a bug or unexpected behavior
- D. Monitoring tool is offline

Answer: C

10. In which phase of CI/CD are logs and metrics actively collected?

- A. Build
- B. Deploy
- C. Monitor
- D. Test

Answer: C

✳️ Part 2: Real-World Pipeline Examples

🚀 Example 1: Jenkinsfile

```
groovy
Copy code
pipeline {
    agent any
    stages {
        stage('Clone') {
            steps {
                git 'https://github.com/your-repo/project.git'
            }
        }
        stage('Build') {
            steps {
                sh './gradlew build'
            }
        }
        stage('Test') {
            steps {
                sh './gradlew test'
            }
        }
        stage('Deploy') {
            steps {
                echo 'Deploying to Production...'
                // Add your deploy script or command here
            }
        }
    }
}
```

```
}
```

Example 2: GitHub Actions Workflow (.github/workflows/main.yml)

yaml

Copy code

name: CI/CD Pipeline

on:

push:
 branches: [main]

jobs:

build-test-deploy:

 runs-on: ubuntu-latest

 steps:

- name: Checkout Code
 uses: actions/checkout@v3

- name: Setup Java

- uses: actions/setup-java@v3

- with:

- java-version: '17'

- name: Build

- run: ./gradlew build

- name: Test

- run: ./gradlew test

- name: Deploy

- run: echo "Deploying to Production..."

MCQs on CI/CD – Set 2

1. Which of the following is *not* part of the CI/CD pipeline?

- A. Build
- B. Test
- C. Analyze
- D. Monitor

 Answer: C

2. What triggers a CI pipeline in most systems?

- A. A server reboot
- B. A code commit or push to repository
- C. Manual approval
- D. Compilation error

 Answer: B

3. What is a major risk in traditional deployment practices?

- A. Frequent code merging
- B. Fast feedback loop
- C. Manual errors and delayed releases
- D. Test automation

 **Answer:** C

4. Which phase ensures that all integrated code functions correctly?

- A. Build
- B. Test
- C. Deploy
- D. Monitor

 **Answer:** B

5. In Continuous Deployment, which step is skipped compared to Continuous Delivery?

- A. Deployment to staging
- B. Human approval before production deployment
- C. Testing phase
- D. Code compilation

 **Answer:** B

6. Jenkins is written in which language?

- A. Java
- B. Python
- C. C++
- D. Ruby

 **Answer:** A

7. GitLab CI/CD uses which file for pipeline configuration?

- A. pipeline.yaml
- B. gitlab-pipeline.yml
- C. .gitlab-ci.yml
- D. ci-config.yaml

 **Answer:** C

8. In Azure DevOps, pipelines can be written in:

- A. Python only
- B. YAML and Classic UI editor
- C. Java only
- D. Bash only

 **Answer:** B

9. Which GitHub Actions keyword defines pipeline trigger events?

- A. run
- B. on
- C. job
- D. event

 **Answer:** B

10. Which of the following tools is a cloud-native CI/CD platform?

- A. Jenkins
- B. GitHub Actions
- C. Ant
- D. Maven

 **Answer:** B

11. What is the main benefit of Continuous Delivery (CD)?

- A. Developers can work in isolation
- B. Product is always in a deployable state

- C. No need for testing
- D. Avoids version control

Answer: B

12. What does a blue/green deployment strategy do?

- A. Rolls back to a previous version
- B. Uses two identical environments for zero-downtime
- C. Deploys only at night
- D. Deletes old features before new ones

Answer: B

13. What is the purpose of using webhooks in CI/CD tools?

- A. Add security to commits
- B. Trigger builds automatically on events like push
- C. Encrypt your pipeline
- D. Send logs to cloud

Answer: B

14. What is 'Artifact' in CI/CD pipeline?

- A. Source code file
- B. Final compiled product after build (e.g., .jar, .exe)
- C. Git repository
- D. Database entry

Answer: B

15. What's the common format for defining GitHub Actions workflows?

- A. INI
- B. JSON
- C. YAML
- D. XML

Answer: C

16. Which action best describes what Jenkins does in CI/CD?

- A. Code review
- B. Test case writing
- C. Automates build, test, and deploy tasks
- D. Deploy code manually

Answer: C

17. Which of the following is *not* a benefit of CI/CD?

- A. Faster feedback
- B. Reduced deployment risks
- C. Higher cost of development
- D. Shorter release cycles

Answer: C

18. Canary deployment means:

- A. Deploying to a single server
- B. Deploying only on weekends
- C. Gradually releasing the update to a small subset of users
- D. Manually switching all users to new version

Answer: C

19. What does the run keyword do in a GitHub Actions workflow?

- A. Declares a new branch
- B. Triggers the workflow
- C. Executes shell commands
- D. Sends reports

Answer: C

20. Which pipeline stage ensures end-user availability and health checks after deployment?

- A. Build
- B. Monitor
- C. Commit
- D. Code Review

Answer: B

SET 3:

1. What does CI stand for?

- A. Continuous Initialization
- B. Continuous Integration
- C. Central Interface
- D. Combined Infrastructure

Answer: B

2. What is the main benefit of Continuous Deployment over Continuous Delivery?

- A. Manual approvals
- B. Frequent testing
- C. Fully automated deployment
- D. Reduced feedback

Answer: C

3. In a CI/CD pipeline, which step runs after code is committed?

- A. Monitor
- B. Build
- C. Feedback
- D. Documentation

Answer: B

4. Which tool is a popular open-source CI server?

- A. Postman
- B. Jenkins
- C. MongoDB
- D. Maven

Answer: B

5. GitHub Actions pipelines are defined in which directory?

- A. .jenkins/
- B. .github/workflows/
- C. .gitlab/ci/
- D. src/actions/

Answer: B

6. In Azure DevOps, pipelines are written using which format?

- A. Python
- B. JSON
- C. YAML
- D. Java

Answer: C

7. What is the first step in a typical CI/CD pipeline?

- A. Deployment
- B. Testing
- C. Code Commit
- D. Monitoring

Answer: C

8. What triggers a Jenkins pipeline automatically?

- A. Code merge
- B. Code commit/push
- C. Test failure
- D. System restart

Answer: B

9. Which CI/CD concept ensures software is always in a deployable state but deployed manually?

- A. Continuous Deployment
- B. Continuous Integration
- C. Continuous Delivery
- D. Static Deployment

Answer: C

10. Which CI/CD phase focuses on ensuring application availability post-deployment?

- A. Build
- B. Monitor
- C. Test
- D. Commit

Answer: B

11. What keyword is used in GitHub Actions to define trigger events?

- A. start
- B. job
- C. on
- D. events

Answer: C

12. What file is used to define GitLab CI/CD pipeline?

- A. .gitlab-pipeline.yml
- B. .gitlab-ci.yml
- C. .pipeline.yml
- D. .gitlab/config.yml

Answer: B

13. Which CI/CD tool is *integrated* into GitHub?

- A. Travis CI
- B. GitHub Actions
- C. Jenkins
- D. Azure DevOps

Answer: B

14. What does a "green build" mean in CI/CD?

- A. Build was skipped
- B. Build failed

- C. Build succeeded with all tests passing
- D. Code is not integrated

Answer: C

15. Which deployment strategy sends updates to a small set of users first?

- A. Rolling
- B. Blue/Green
- C. Canary
- D. Parallel

Answer: C

16. Which of these is NOT a valid benefit of CI/CD?

- A. Faster delivery
- B. Early bug detection
- C. Expensive integration
- D. Reduced manual effort

Answer: C

17. Jenkins uses which file to define pipeline as code?

- A. Jenkinsfile
- B. build.gradle
- C. pipeline.yaml
- D. run.sh

Answer: A

18. The purpose of automated testing in CI/CD is to:

- A. Replace developers
- B. Check hardware compatibility
- C. Verify code changes automatically
- D. Merge branches

Answer: C

19. Continuous Integration helps in:

- A. Delayed integration
- B. Frequent integration of code with tests
- C. Rare deployments
- D. Manual deployments

Answer: B

20. Which is a monitoring tool often integrated in CI/CD?

- A. JUnit
- B. Selenium
- C. Prometheus
- D. Postman

Answer: C