Jenkins CI/CD – 1 Page Notes

🚀 Introduction

- **Jenkins is an open-source automation server used to build, test, and deploy software automatically.**
- **It enables Continuous Integration (CI) and Continuous Deployment (CD) — making software delivery faster and more reliable.**

🔄 CI/CD Concepts

Concept	Description
Continuous Integration (CI)	Developers frequently merge code into a shared repository, and Jenkins automatically builds/tests it.
Continuous Deployment (CD)	Automatically deploys tested code to production or staging environments.

🧱 Jenkins Architecture

1. **Jenkins Master (Controller): Manages jobs, schedules builds, and monitors results.**
2. **Jenkins Agent (Node): Executes build jobs assigned by the master.**
3. **Plugins: Extend Jenkins features (e.g., Git, Docker, Maven, Pipeline).**
4. **Jenkinsfile: Script that defines the CI/CD pipeline stages.**

⚙️ Installation Steps

1. **Install Java (JDK 11+)**
2. **Download Jenkins .war file or use Docker:**
3. **docker run -p 8080:8080 jenkins/jenkins:lts**
4. **Access Jenkins at http://localhost:8080**
5. **Install suggested plugins and create an admin user.**

📜 Common Jenkins Commands

Command	Description
systemctl start jenkins	Start Jenkins service
systemctl stop jenkins	Stop Jenkins service
cat /var/lib/jenkins/secrets/initialAdminPassword	Get admin password
jenkins --version	Check Jenkins version

🧩 **Pipeline Example (Jenkinsfile)**

```
pipeline {
 agent any
 stages {
  stage('Build') {
   steps {
    echo 'Building...'
   }
  }
  stage('Test') {
   steps {
    echo 'Testing...'
   }
  }
  stage('Deploy') {
   steps {
    echo 'Deploying...'
   }
  }
 }
}
```

🔗 **Common Integrations**

- **Version Control: Git, GitHub, GitLab**
- **Build Tools: Maven, Gradle, npm**
- **Containers: Docker, Kubernetes**
- **Notification: Slack, Email**

✅ **Advantages**

- **Automates build, test, and deployment**
- **Supports 1000+ plugins**
- **Easy to integrate with any tool**
- **Improves development speed and code quality**

⚡ **Key Difference**

Manual Deployment	Jenkins CI/CD
Time-consuming	Automated process
Error-prone	Consistent builds
No visibility	Real-time monitoring