Student Handout: Jenkins Server Setup

# 1. Jenkins Architecture – Master-Agent Model

Master Node:

- Coordinates jobs, schedules builds, manages web UI.

Agents (formerly called slaves):

- Execute actual build tasks.

Advantages:

- Distributed builds across multiple machines.

- Parallel execution for faster pipelines.

- Specialized environments (e.g., Windows builds on Windows agents, Linux builds on Linux agents).

# 2. Prerequisites for Installation

Java Requirements:

- Java 11 or Java 17 (LTS recommended).

- Either OpenJDK or Oracle JDK.

- Set JAVA\_HOME environment variable.

Version Control:

- Git client installed.

- Access to project repositories.

System Requirements:

- Minimum: 256 MB RAM.

- Recommended: 1 GB+ RAM, 10 GB+ disk space.

- Stable network connection.

- Requirements scale with workload (# of builds, plugins, agents).

# 3. Installation Options

Option 1: Local Installation:

- Linux: Install via apt or yum.

- Windows: Use MSI installer or install as a service.

- macOS: Homebrew or installer package.

Best for: Small teams, development setups.

Option 2: Dockerized Jenkins:

Advantages:

- Same environment across platforms.

- Easy upgrades & rollbacks.

- Isolated from host system.

Command:

docker run -p 8080:8080 -p 50000:50000 \

-v jenkins\_home:/var/jenkins\_home \

jenkins/jenkins:lts

Option 3: Cloud-Based Jenkins:

- AWS: EC2 with Jenkins AMI, Elastic Beanstalk, AWS integrations.

- Azure: VM deployment, Azure DevOps integration.

- GCP: Compute Engine, Kubernetes Engine, Cloud Build integration.

Best for: Large, distributed teams needing scalability and HA.

# 4. Hands-On Demo: Linux Installation

1. Update repos:

sudo apt update

2. Install Java:

sudo apt install openjdk-11-jdk

3. Add Jenkins repository key & source.

4. Install Jenkins:

sudo apt install jenkins

5. Start Jenkins:

sudo systemctl start jenkins

6. Enable auto-start:

sudo systemctl enable jenkins

# 5. Initial Setup Wizard

- Open Jenkins: http://localhost:8080

- Unlock using admin password (from /var/lib/jenkins/secrets/initialAdminPassword).

- Choose plugin setup:

- Install suggested plugins (easy start).

- Select specific plugins (custom setup).

- Create first admin user.

- Configure Jenkins URL.

# 6. Jenkins Dashboard Tour

Main Dashboard:

- Displays job status, build history, system health.

Sidebar Navigation:

- Access jobs, credentials, configuration.

Status Icons:

- Success (✔), Failure (✖), In progress (⟳).

# 7. Security Basics

Admin Account:

- Use strong, unique password.

- Enable 2FA (with plugin).

- Audit accounts regularly.

Role-Based Access Control:

- Install 'Role-Based Authorization Strategy' plugin.

- Create roles with specific permissions.

- Assign users to roles.

Security protects your pipeline, code, and credentials.

# 8. Next Steps

- Install essential plugins (Git, Docker, Pipeline, etc.).

- Configure your first pipeline to validate setup.

- Add agents for distributed builds.

- Implement CI/CD best practices:

- Automated testing.

- Notifications (Slack, email).

- Deployment strategies (Blue/Green, Canary).

# 9. Post-Installation Checks

Environment Variables:

- Ensure JAVA\_HOME is set.

- Add $JAVA\_HOME/bin (Linux/macOS) or %JAVA\_HOME%\bin (Windows) to PATH.

Verification:

- Check Java: java -version

- Verify Jenkins service: sudo systemctl status jenkins

- Open Jenkins UI: http://localhost:8080

# Key Takeaways

- Jenkins can run locally, in Docker, or on the cloud.

- Master-Agent model enables scalability.

- Secure your instance with RBAC + strong admin practices.

- Start simple → scale with plugins, agents, and best practices.