
🧩 Jenkins Components – Detailed Explanation with Notes

📌 Overview of Jenkins Architecture

Jenkins follows a **master-agent (controller-worker)** architecture. It supports both **standalone** and **distributed** CI/CD systems.

🔧 1. Jenkins Master (Controller)

✅ Role:

- Main server that manages:
    - Build scheduling
    - Dispatching jobs to agents
    - Monitoring executions
    - Managing plugin ecosystem
    - Exposing web UI

📌 Notes:

- Jenkins master alone can execute jobs, but it is **not recommended** for high-load scenarios.
- Default port: 8080

- Jenkins home directory: /var/lib/jenkins

🖥️ 2. Jenkins Agent (Slave / Worker)

✅ Role:

- Executes jobs assigned by the master
- Useful for running builds on different OS, tools, or configurations

🧠 Notes:

- Agent can run on VMs, bare metal, Docker containers, or Kubernetes Pods.
- Connection methods:
  - **SSH**
  - **JNLP (Java Web Start)**
  - **Cloud-based** agents (e.g., EC2, Kubernetes)

📦 3. Jobs / Projects

✅ Types of Jobs:

Type	Description
Freestyle	Basic, GUI-driven configuration

Pipeline	Groovy-scripted CI/CD logic
Multibranch	Auto-discovers Git branches with Jenkinsfile
Matrix	Test across multiple OS/tools
External Job	Monitors jobs outside Jenkins

🧠 **Notes:**

- Pipelines = Code as CI/CD logic (Jenkinsfile)
- Jobs can be triggered manually, periodically, or by GitHub hooks

---

🧪 4. Build Executor

✅ **Role:**

- A slot where a build runs
- Each agent (or master) has a number of executors (parallel job runners)

Example:

- Master has 2 executors → 2 builds can run in parallel

🧩 5. Plugins

✅ **Role:**

- Extend Jenkins functionality:
    - Git integration
    - Docker
    - Slack notifications
    - Kubernetes support

🔥 **Popular Plugins:**

- Git Plugin
- Docker Pipeline
- Blue Ocean (modern UI)
- Credentials Binding
- Pipeline: Groovy

🧠 **Notes:**

- Jenkins has **1700+ plugins**
- Plugin updates should be tested in dev/staging Jenkins before production

---

🔑 6. Credentials Store

✅ **Role:**

- Securely stores:
    - GitHub tokens
    - Docker Hub credentials
    - SSH keys
    - API tokens

📦 Credential Types:

- **Username/Password**
- **Secret Text**
- **SSH Key**
- **Certificate**

📝 7. Jenkinsfile

✅ Role:

- Pipeline-as-code script (written in Groovy)
- Stored in Git repository with the application

Example:

```
pipeline {
  agent any
```

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

📚 8. SCM (Source Code Management)

✅ Role:

- Integration with Git, GitHub, GitLab, Bitbucket
- Monitors code changes
- Triggers builds automatically

🛠️ 9. Build Tools Integration

Common Tools:

- **Maven, Gradle, Ant**
- **npm, yarn**
- **Docker CLI**

✅ **Role:**

- Compile, test, and package code inside Jenkins jobs

🔄 10. Triggers

✅ **Types:**

- **Manual**
- **SCM Polling**
- **Webhooks (GitHub/GitLab)**
- **Scheduled (CRON)**
- **After another job**

📊 11. Console Output & Logs

✅ **Role:**

- View job output (stdout/stderr)
- Troubleshooting failed builds
- Logs stored under job workspace

📁 12. Workspace

✅ Role:

- Directory on agent (or master) where job executes
- Stores:
  - Source code
  - Logs
  - Artifacts

📦 13. Artifacts

✅ Role:

- Files generated by jobs (JARs, reports, logs)
- Can be archived and downloaded from Jenkins UI

🔄 14. Build History

✅ Role:

- Stores metadata about previous builds
- Includes:
  - Status (Success/Failure)
  - Duration
  - Changes
  - Logs

🔒 15. Security Components

🔐 Includes:

- User management (via Jenkins or external providers)
- Role-based access (via RBAC plugin)
- CSRF protection
- Credential masking

🌍 16. Jenkins UI Components

UI Section	Purpose
Dashboard	View all jobs
Job page	View builds, config
Build page	Console, logs, artifacts
Manage Jenkins	Global settings, plugins
Blue Ocean	Modern pipeline UI

📌 Summary Table

Component	Purpose
Master	Central controller
Agent	Executes builds
Jobs	Define CI/CD steps
Pipeline	Code-driven CI/CD
Plugins	Extends Jenkins
Credentials	Secure secrets
Triggers	Automate builds
Jenkinsfile	CI/CD as code
Logs/Console	Debug info
SCM Integration	Connects to Git
Security	AuthN/AuthZ, RBAC

UI	Job & system management