

Jenkins agents

Jenkins agents (also known as slaves) are machines or environments that run tasks assigned by the Jenkins master. In a Jenkins architecture, the master manages the scheduling of jobs, dispatches builds to agents, and monitors the results. The agents perform the actual work, such as building, testing, and deploying applications.

Key Functions of Jenkins Agents:

Run Build Jobs: Execute tasks like compiling code, running tests, and deploying builds.

Distribute Workload: Handle builds in parallel or distribute them across multiple environments (e.g., different operating systems or configurations).

Environment Flexibility: Allow Jenkins to run builds in specialized environments with required tools and dependencies.

Types of Jenkins Agents

Jenkins supports multiple types of agents:

- **Permanent Agent** (Static node)
- **Dynamic Agent** (Created on demand)
- **Docker Agent**
- **Cloud-Based Agents** (Kubernetes, AWS, etc.)

Setting Up a Permanent Agent (Static Node)

This is a **manual setup** where an agent runs continuously.

◆ Steps to Create a Jenkins Agent

◆ Step 1: Configure the Agent Node in Jenkins

1. Go to **Jenkins Dashboard** → **Manage Jenkins** → **Manage Nodes and Clouds**.
2. Click **New Node**.
3. Enter a **Node Name** (e.g., `linux-agent`).
4. Select **"Permanent Agent"** → Click **OK**.
5. Configure the node settings:
 - **# of Executors**: Number of concurrent builds (e.g., `2`).
 - **Remote root directory**: Path on the agent machine where Jenkins will store data (e.g., `/home/jenkins`).
 - **Labels**: Tags to identify the agent (e.g., `linux`, `build`).
 - **Launch Method**: Choose **"Launch agent via Java Web Start"** or **SSH**.

◆ Step 2: Connect the Agent to Jenkins

On the agent machine, install **Java 8+**:

bash

```
java -version
```

If Java is not installed:

bash

```
sudo apt install openjdk-11-jdk # Ubuntu/Debian
```

```
sudo yum install java-11-openjdk # RHEL/CentOS
```

- 1.
2. **Download the Agent JAR** from the Jenkins UI:
 - Go to **Jenkins Dashboard** → **Manage Nodes and Clouds**.
 - Click on the newly created agent.
 - Copy the **"Agent.jar"** download link.

Run the following command on the agent machine:

bash

```
java -jar agent.jar -jnlpUrl
```

```
http://<JENKINS_URL>/computer/linux-agent/slave-agent.jnlp -secret
```

```
<SECRET_KEY> -workDir "/home/jenkins"
```

3. Replace `<JENKINS_URL>` with your Jenkins server URL.
 4. The agent should now appear as **Online** in Jenkins.
-

Using an Agent in a Jenkins Pipeline

Once the agent is set up, you can assign jobs to it using the `agent` directive in your pipeline.

♦ Example: Run a Job on a Specific Agent

groovy

```
pipeline {
    agent { label 'linux-agent' } // Uses the agent with this label

    stages {
        stage('Build') {
            steps {
                sh 'echo Running on Linux Agent'
            }
        }
    }
}
```

Setting Up an SSH Agent

If you prefer **SSH-based connection**, follow these steps:

Install OpenSSH Server on the Agent Machine

bash

```
sudo apt install openssh-server # Ubuntu/Debian
sudo systemctl start sshd # Start SSH service
```

Create a Jenkins User on the Agent Machine

```
sudo adduser jenkins
sudo passwd jenkins
```

1. Enable SSH Authentication

- Copy the **Jenkins master's public SSH key** to the agent's `/home/jenkins/.ssh/authorized_keys` file.

2. Configure the Agent in Jenkins

- Set **Launch method** to **"Launch agent via SSH"**.

- Enter the **agent's IP address** and **Jenkins username**.
 - Save & Restart Jenkins.
-

5 Using a Docker Agent

If you want a **temporary agent** that runs inside a Docker container, use:

groovy

```
pipeline {
    agent {
        docker {
            image 'maven:3.8.1'
        }
    }
    stages {
        stage('Build') {
            steps {
                sh 'mvn clean package'
            }
        }
    }
}
```