

Academic Year: 2025-26

Semester: V

Class / Branch: TEIT

Subject: DevOps Lab

Name of Instructor: Prof. Seema Jadhav

Experiment No. 6

Aim: To implement Jenkins Master-Slave Architecture with Scaling.

Theory:

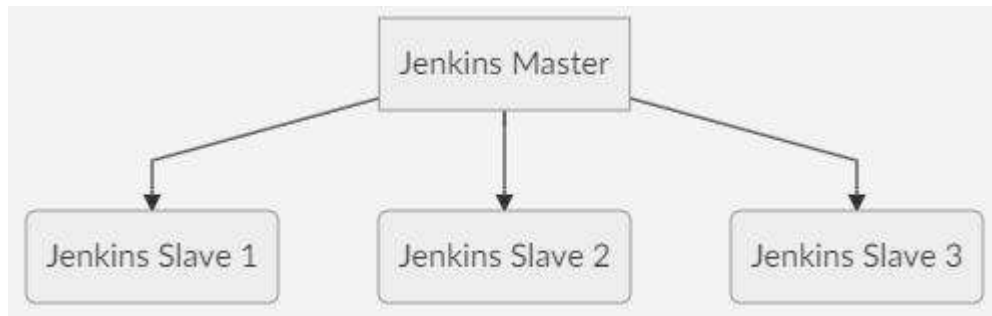
Need of Jenkins Mater-[Agent-slave] Architecture:

When we build the Jenkins job in a single Jenkins master node then Jenkins uses the resource of the base machine and if no executor is available then the jobs are queued in the Jenkins server. Sometimes you might need several different environments to test your builds. This cannot be done by a single Jenkins server. It is recommended not to run different jobs in the same system that required a different environment. In such scenarios where we need a different machine with a different environment that takes the specific job from the master to build.

On the same Jenkins setup, multiple teams are working with their jobs. All jobs are running on the same base operating system and the base operating system has limited resources. Also, we don't want to put our personal data on the same system where other teams can read.

Jenkins Distributed Architecture:

Jenkins uses A Master-Slave architecture to manage distributed builds. The machine where we install Jenkins software will be Jenkins master and that run's on port 8080 by default. On the slave machine, we install a program called Agent. This agent requires JVM. This agent executes the tasks provided by Jenkins master. We can launch n numbers of agents and we can configure which task will be run on which agent server from Jenkins master by assigning the agent to the task.



Jenkins Master and Slave Concept

A Jenkins master comes with the basic installation of Jenkins, and in this configuration, the master handles all the tasks for our build system.

If we are working on multiple projects, we may run multiple jobs on each project. Some projects need to run on some nodes, and in this process, we need to configure slaves. [Jenkins slaves connect to the Jenkins master](#) using the Java Network Launch Protocol(JNLP).

The Jenkins master acts to schedule the jobs, assign slaves, and send builds to slaves to execute the jobs.

It will also monitor the slave state (offline or online) and get back the build result responses from slaves and the display build results on the console output. The workload of building jobs is delegated to multiple **slaves**.

Steps to Configure Jenkins Master and Slave Nodes

STEP1: In Jenkins Dashboard Click on Manage Jenkins -> Manage Nodes

Dashboard > Manage Jenkins

Build History

Project Relationship

Check File Fingerprint

Manage Jenkins

My Views

Build Queue

No builds in the queue.

Build Executor Status

built-in node + 1 agent (0 of 2 executors busy)

New version of Jenkins (2.462.1) is available for download ([changelog](#)).

Building on the built-in node can be a security issue. You should set the number of executors on the built-in node to 0. See [the documentation](#). [Manage](#) [Dismiss](#)

Warnings have been published for the following currently installed components: [Configure which of these warnings are shown](#)

Jenkins 2.452.3 core and libraries:
[Multiple security vulnerabilities in Jenkins 2.470 and earlier, LTS 2.452.3 and earlier](#)
A fix for this issue is available. Update Jenkins now.

System Configuration

- System**
Configure global settings and paths.
- Tools**
Configure tools, their locations and automatic installers.
- Plugins**
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.
- Nodes**
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.
- Clouds**
Add, remove, and configure cloud instances to provision agents on-demand.
- Appearance**
Configure the look and feel of Jenkins.



STEP 2: Select New Node and enter the name of the node in the Node Name field.

Select Permanent Agent and click the OK button. Initially, you will get only one option, “Permanent Agent.” Once we have one or more slaves you will get the “Copy Existing Node” option. Click Create

Jenkins Search (CTRL+K) ? [1] [2] SUJATA OAK

Dashboard > Manage Jenkins > Nodes > New node

New node

Node name

agent2

Type

☒ Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.

☐ Copy Existing Node

Create

STEP3: Configure node with below details:

```
sujata@Ubuntu:~/Desktop/JENKINS_LAB$ pwd  
/home/sujata/Desktop/JENKINS_LAB
```


```
sujata@Ubuntu:~/Desktop/JENKINS_LAB$ su root  
Password:  
root@Ubuntu:/home/sujata/Desktop/JENKINS_LAB# find / -type f -name java
```

```
/usr/lib/jvm/java-11-openjdk-amd64/bin/java
```



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY
Department of Information Technology
(NBA Accredited)



 **Jenkins**

Search (CTRL+K) ?

1 ? 2 ? SUJATA OAK ▾

Dashboard > Manage Jenkins > Nodes >

Name ?

agent2

Description ?

This is a demo on Master-Slave Jenkins

Plain text: [Preview](#)

Number of executors ?

1

Remote root directory ?

/home/sujata/Desktop/JENKINS_LAB

Labels ?

agent2

Usage ?

Only build jobs with label expressions matching this node ▾

Launch method ?

Launch agent by connecting it to the controller ▾

Availability ?

Keep this agent online as much as possible ▾ ?

Under ‘Node Properties’, provide jdk path.



Node Properties

☐ Disable deferred wipeout on this node ?

☐ Disk Space Monitoring Thresholds

☒ Environment variables

List of variables ?

Name

java_home

Value

/usr/lib/jvm/java-11-openjdk-amd64/bin/java

Add

☐ Tool Locations

Save

Jenkins

Search (CTRL+K) ?

1

2

SUJATA OAK

log out

Dashboard > Manage Jenkins > Nodes >

Nodes

Clouds

Build Queue

No builds in the queue.

Build Executor Status

built-in node + 2 agents (0 of 2 executors busy)

Nodes

+ New Node

Configure Monitors

?

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	agent1	Linux (amd64)	In sync	5.88 GiB	923.26 MiB	5.88 GiB	189ms
	agent2		N/A	N/A	N/A	N/A	N/A
	Built-in Node	Linux (amd64)	In sync	5.88 GiB	923.26 MiB	5.88 GiB	0ms
Data obtained		21 sec	21 sec	21 sec	21 sec	21 sec	21 sec

Icon: S M L

Legend



STEP4: On click of 'Save' will display the below page with error message. Here Jenkins connect with Slave node using Java Web Start and it needs a port to establish the connection.

To configure JNLP port in global security. Now goto Manage Jenkins -> Security

Agents

TCP port for inbound agents ?

☒ Fixed

50000

☐ Random

☐ Disable

This port has to be allowed to access across firewall, so from Master terminal run the below command,

```
sudo ufw allow 50000/tcp
```

This command will allow port 50000 to listen for request.

```
root@Ubuntu:/home/sujata/Desktop/JENKINS_LAB# sudo ufw allow 50000/tcp
Rule added
Rule added (v6)
```

STEP5: Again coming back to Jenkins and navigate to Nodes -> agent2 which will display two ways to connect with Agent node.



PARSHVANATH CHARITABLE TRUST'S

A. P. SHAH INSTITUTE OF TECHNOLOGY

Department of Information Technology

(NBA Accredited)



To establish connection, run the below command

```
root@Ubuntu:/home/sujata/Desktop/JENKINS_LAB# curl -sO http://127.0.0.1:8080/jnlpJars/agent.jar
```

```
root@Ubuntu:/home/sujata/Desktop/JENKINS_LAB# java -jar agent.jar -url http://127.0.0.1:8080/ -secret cacd8d769874ea4f1a2a28392ffe62d08add0eeb0ea463cced99fa1f707fad0 -name agent2 -workDir "/home/sujata/Desktop/JENKINS_LAB"
```

OUTPUT:

```
INFO: Both error and output logs will be printed to /home/sujata/Desktop/JENKINS_LAB/remoting
Aug 20, 2024 10:24:53 AM hudson.remoting.Launcher createEngine
INFO: Setting up agent: agent2
Aug 20, 2024 10:24:53 AM hudson.remoting.Engine startEngine
INFO: Using Remoting version: 3206.vb_15dcf73f6a_9
Aug 20, 2024 10:24:53 AM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/sujata/Desktop/JENKINS_LAB/remoting as a remoting work directory
Aug 20, 2024 10:24:54 AM hudson.remoting.Launcher$CuiListener status
INFO: Locating server among [http://127.0.0.1:8080/]
Aug 20, 2024 10:24:54 AM org.jenkinsci.remoting.engine.JnlpAgentEndpointResolver resolve
INFO: Remoting server accepts the following protocols: [JNLP4-connect, Ping]
Aug 20, 2024 10:24:54 AM hudson.remoting.Launcher$CuiListener status
INFO: Agent discovery successful
  Agent address: 127.0.0.1
  Agent port:    50000
  Identity:      80:21:52:35:ca:60:ed:97:f1:2a:65:7a:50:b9:27:77
Aug 20, 2024 10:24:54 AM hudson.remoting.Launcher$CuiListener status
INFO: Handshaking
Aug 20, 2024 10:24:54 AM hudson.remoting.Launcher$CuiListener status
INFO: Connecting to 127.0.0.1:50000
Aug 20, 2024 10:24:54 AM hudson.remoting.Launcher$CuiListener status
INFO: Server reports protocol JNLP4-connect-proxy not supported, skipping
Aug 20, 2024 10:24:54 AM hudson.remoting.Launcher$CuiListener status
INFO: Trying protocol: JNLP4-connect
Aug 20, 2024 10:24:54 AM org.jenkinsci.remoting.protocol.impl.BIONetworkLayer$Reader run
INFO: Waiting for ProtocolStack to start.
Aug 20, 2024 10:24:54 AM hudson.remoting.Launcher$CuiListener status
INFO: Remote identity confirmed: 80:21:52:35:ca:60:ed:97:f1:2a:65:7a:50:b9:27:77
Aug 20, 2024 10:24:54 AM hudson.remoting.Launcher$CuiListener status
INFO: Connected
```

This will establish connection with the configured Slave node.



PARSHVANATH CHARITABLE TRUST'S
A. P. SHAH INSTITUTE OF TECHNOLOGY
Department of Information Technology
(NBA Accredited)



The screenshot shows the Jenkins 'Nodes' page. On the left, there's a sidebar with 'Nodes' and 'Clouds' tabs. Below them are sections for 'Build Queue' (showing no builds) and 'Build Executor Status' (showing 2 agents, 0 busy). The main area is titled 'Nodes' and contains a table with columns: S, Name, Architecture, Clock Difference, Free Disk Space, Free Swap Space, Free Temp Space, and Response Time. The table lists three nodes: 'agent1' (timed out), 'agent2' (14 sec behind), and 'Built-In Node' (in sync). A '+ New Node' button and 'Configure Monitors' link are at the top right.

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	agent1		N/A	5.88 GiB	N/A	5.88 GiB	Timed out for last 1 attempts
	agent2	Linux (amd64)	14 sec behind	5.88 GiB	923.26 MiB	5.88 GiB	30003ms
	Built-In Node	Linux (amd64)	In sync	5.88 GiB	923.26 MiB	5.88 GiB	0ms
Data obtained		1 min 13 sec	1 min 13 sec	1 min 13 sec	1 min 13 sec	1 min 13 sec	1 min 13 sec

Now Jenkins Slave node is ready to run any job. This node's label name should be mentioned in the corresponding Job configuration as below:

STEP 6: Create a New Job in Jenkins dashboard

The screenshot shows the 'Enter an item name' dialog in Jenkins. The input field contains 'master_slave_jenkins_demo20082024'. Below the input field is a 'Required field' message. The dialog lists four job types: 'Freestyle project' (Classic, general-purpose), 'Pipeline' (Orchestrates long-running activities), 'Multi-configuration project' (Suitable for projects that need a large number of different configurations), and 'Folder' (Creates a container that stores nested items). An 'OK' button is at the bottom left.



STEP 7: Configure the page with following:

Dashboard > master_slave_jenkins_demo20082024 > Configuration

Configure

- General**
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

Description

Demo On Jenkins Master Slave architecture

Plain text: [Preview](#)

- ☐ Discard old builds ?
- ☐ GitHub project
- ☐ This project is parameterized ?
- ☐ Throttle builds ?
- ☐ Execute concurrent builds if necessary ?
- ☒ Restrict where this project can be run ?

Label Expression ?

agent2

Build Steps

≡ **Execute shell** ?

Command

See [the list of available environment variables](#)

```
echo " Hello Students, Welcome to session on MASTER SLAVE ARCHITECTURE IN JENKINS!!!"
```

[Save](#) [Apply](#)

Click on Build-Now, Console Output



The screenshot shows the Jenkins web interface. The top navigation bar includes the Jenkins logo, a search bar, and user information for 'SUJATA OAK'. The breadcrumb trail is 'Dashboard > master_slave_jenkins_demo20082024 > #1 > Console Output'. On the left sidebar, 'Console Output' is selected. The main area displays the console output for build #1, which was started by user 'SUJATA OAK' and finished successfully. The output text is as follows:

```
Started by user SUJATA OAK
Running as SYSTEM
Building remotely on agent2 in workspace /home/sujata/Desktop/JENKINS_LAB/workspace/master_slave_jenkins_demo20082024
[master_slave_jenkins_demo20082024] $ /bin/sh -xe /tmp/jenkins8463428541727822031.sh
+ echo Hello Students, Welcome to session on MASTER SLAVE ARCHITECTURE IN JENKINS!!!
Hello Students, Welcome to session on MASTER SLAVE ARCHITECTURE IN JENKINS!!!
Finished: SUCCESS
```

STEP 8: Goto Jenkins Dashboard->Manage Jenkins->Nodes->agent2

The screenshot shows the Jenkins 'Agent agent2' page. The breadcrumb trail is 'Dashboard > Manage Jenkins > Nodes > agent2'. The left sidebar contains various options for managing the agent. The main content area shows that the agent is connected and provides monitoring data. Below this, a table lists the projects tied to the agent.

Agent agent2 Mark this node temporarily offline ?

This is a demo on Master-Slave Jenkins

Agent is connected.

Monitoring Data ▾

Projects tied to agent2

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☀	master_slave_jenkins_demo20082024	3 min 53 sec #1	N/A	0.74 sec

Conclusion: This way we could connect with many machines as Slave nodes with different environment and execute our Jenkins jobs.