

Experiment No. 6

Aim: To understand Jenkins Master-Slave Architecture and Scale your Jenkins standalone implementation by implementing slave nodes.

LO No. & Statement: (LO4): Examine the importance of Selenium and Jenkins to test software Applications.

Theory:

Jenkins is one of the most important tools in DevOps. Jenkins is used in the Continuous Integration stage of DevOps. In this blog, I am going to talk about Jenkins's Master and Slave architecture. The pointers that I will cover are as follows:

- What is Jenkins?
- Jenkins Architecture
- How does Jenkins Master-Slave architecture work?
- Setting up Slaves with Jenkins Master

What is Jenkins?

- Jenkins is an open-source automation tool written in Java with plugins built for Continuous Integration purposes. Jenkins is used to building and testing your software projects continuously making it easier for developers to integrate changes to the project, and making it easier for users to obtain a fresh build. It also allows you to continuously deliver your software by integrating with a large number of testing and deployment technologies.
- With Jenkins, organizations can accelerate the software development process through automation. Jenkins integrates development life-cycle processes of all kinds, including build, document, test, package, stage, deploy, static analysis, and much more.

Advantages of Jenkins include:

- It is an open-source tool with great community support.
- Too easy to install.
- It has 1000+ plugins to ease your work. If a plugin does not exist, you can code it and share it with the community.
- It is free of cost.
- It is built with Java and hence, it is portable to all the major platforms.

Jenkins Master:

Your main Jenkins server is the Master. The Master's job is to handle:

- Scheduling build jobs.
- Dispatching builds to the slaves for the actual execution.
- Monitor the slaves (possibly taking them online and offline as required).
- Recording and presenting the build results.
- A Master instance of Jenkins can also execute build jobs directly.

Jenkins Slave:

A Slave is a Java executable that runs on a remote machine. The following are the characteristics of Jenkins's Slaves:

- It hears requests from the Jenkins Master instance.
- Slaves can run on a variety of operating systems.
- The job of a Slave is to do as they are told to, which involves executing build jobs dispatched by the Master.
- You can configure a project to always run on a particular Slave machine or a particular type of Slave machine, or simply let Jenkins pick the next available Slave.

Output:

Step 1: Create a node by clicking onto manage nodes and clouds

The screenshot shows the Jenkins 'Manage Jenkins' page. The left sidebar contains navigation links: 'New Item', 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins' (highlighted), and 'My Views'. Below these are sections for 'Build Queue' (showing 1/2 builds in the queue) and 'Build Executor Status' (showing 1/2 executors busy). The main content area is titled 'Manage Jenkins' and includes a 'System Configuration' section with three options: 'Configure System' (Configure global settings and paths), 'Global Tool Configuration' (Configure tools, their locations and automatic installers), and 'Manage Plugins' (Add, remove, disable or enable plugins that can extend the functionality of Jenkins). The 'Manage Nodes and Clouds' option is also visible, described as 'Add, remove, control and monitor the various nodes that Jenkins runs jobs on.' Below this is a 'Security' section.

The second screenshot shows the 'Configure Global Security' page. It features a breadcrumb trail: 'Dashboard > Manage Jenkins > Configure Global Security'. At the top, there is a dropdown menu set to 'Plain text' with a note: 'Treats all input as plain text. HTML unsafe characters like < and & are escaped to their respective character entities.' Below this is an 'Agents' section with a 'TCP port for inbound agents' field and three radio button options: 'Fixed', 'Random' (selected), and 'Disable'. There is also a text input field for 'Agent protocols...'. At the bottom, there are 'Save' and 'Apply' buttons.

- ↑ Back to Dashboard
- ⚙️ Manage Jenkins
- + New Node
- ☁️ Configure Clouds
- 📊 Node Monitoring

Build Queue

No builds in the queue.

Build Executor Status

built in node 0 of 2 executors busy

Manage nodes and clouds

Refresh status

🔖	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
🖥️	Built-In Node	Windows 10 (amd64)	In sync	113.80 GB	5.20 GB	113.80 GB	0ms
	Data obtained	1 min 54 sec	1 min 54 sec	1 min 54 sec	1 min 54 sec	1 min 54 sec	1 min 54 sec

REST API Jenkins 2.361.2

- ↑ Back to Dashboard
- ⚙️ Manage Jenkins
- + New Node
- ☁️ Configure Clouds
- 📊 Node Monitoring

Build Queue

No builds in the queue.

Build Executor Status

built in node 0 of 2 executors busy

New node

Node name

slaveA

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.

Create

REST API Jenkins 2.361.2



🔍 Search (CTRL+K)



Krishna Garg



log out

- ↑ Back to Dashboard
- ⚙️ Manage Jenkins
- + New Node
- ☁️ Configure Clouds
- 📊 Node Monitoring

Build Queue

No builds in the queue.

Build Executor Status

built in node 0 of 2 executors busy

Name

slaveA

Description

jenkins slave

Number of executors

2

Remote root directory

C:\Users\lacer\Desktop\BatchSlave

Labels

jenkinsSlaveA

Dashboard > Manage Jenkins > Nodes >

Usage ⓘ

Use this node as much as possible

Launch method ⓘ

Launch agent by connecting it to the controller

☐ Disable WorkDir ⓘ

Custom WorkDir path ⓘ

C:\Users\jace\Desktop\BatchBulave

Internal data directory ⓘ

remoting

☐ Fail if workspace is missing ⓘ

☐ Use WebSocket ⓘ

Advanced...

Dashboard > Manage Jenkins > Nodes >

Node Properties

☐ Disable deferred wipeout on this node ⓘ

☐ Environment variables

☒ Tool Locations

List of tool locations ⓘ

Name

jdk

Home

C:\Program Files\Java\jdk-11.0.18

Add

Jenkins

Krishna Garg
 log out

Dashboard > Manage Jenkins > Nodes >

↑ Back to Dashboard

⚙ Manage Jenkins

+ New Node

☁ Configure Clouds

📊 Node Monitoring

Build Queue

10 builds in the queue

Build Executor Status

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

Manage nodes and clouds

Refresh status

\$	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
🖥	Built-In Node	Windows 10 (amd64)	In sync	113.80 GB	5.20 GB	113.80 GB	0ms
❌	slavaA		N/A	N/A	N/A	N/A	N/A
	Data obtained	8 ms	7 ms	6 ms	9 min 48 sec	9 min 48 sec	9 min 48 sec

REST API Jenkins 2.361.2

Connected to the Slave

Dashboard > Nodes > slaveA

Back to List

Status

Delete Agent

Configure

Build History

Load Statistics

Log

Build Executor Status

built-in node (1) of 2 executors busy

Agent slaveA (Jenkins Slave)

Mark this node temporarily offline

Run from agent command line:

```
curl -sO http://localhost:8080/jnlpJars/agent.jar
java -jar agent.jar -jnlpurl http://localhost:8080/computer/slaveA/jenkins-agent.jnlp -secret 40892cfa0492d7b408ff8a9e023c7c3d2f8a6f70c0bed28f61f28acc006c30 -workDir "C:\Users\laser\Desktop\BatchSlave"
```

Or run from agent command line, with the secret stored in a file:

```
echo 40892cfa0492d7b408ff8a9e023c7c3d2f8a6f70c0bed28f61f28acc006c30 > secret-file
curl -sO http://localhost:8080/jnlpJars/agent.jar
java -jar agent.jar -jnlpurl http://localhost:8080/computer/slaveA/jenkins-agent.jnlp -secret @secret-file -workDir "C:\Users\laser\Desktop\BatchSlave"
```

Labels

JenkinsSlaveA

Projects tied to slaveA

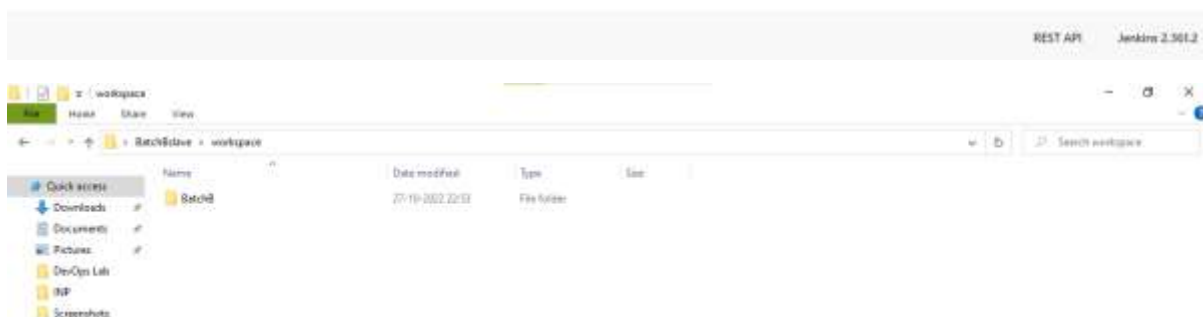
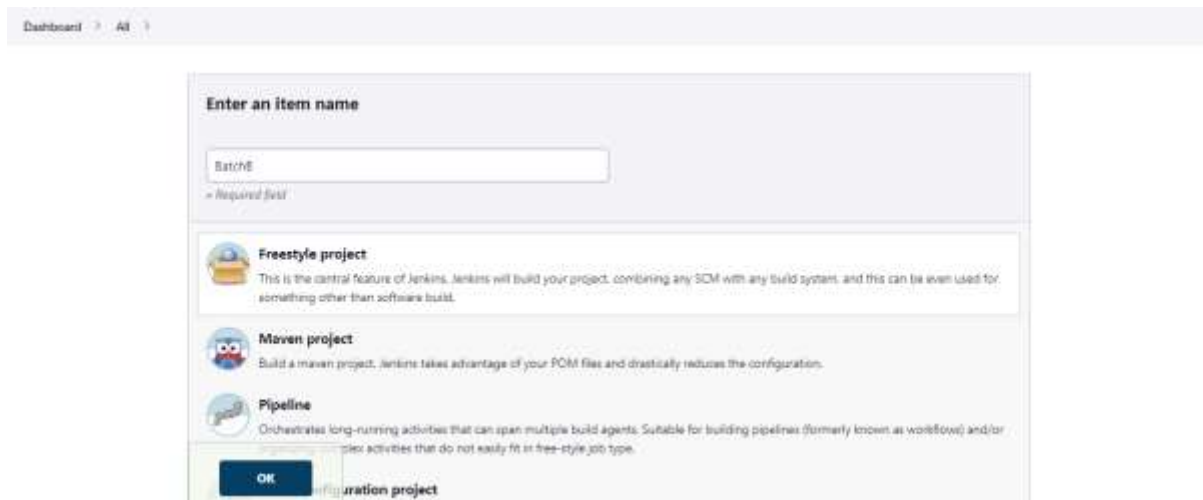
None

Now you can assign the task to the slave

```
C:\Users\acer>curl -G0 http://localhost:8080/jenkins/agent.jar

C:\Users\acer>java -jar agent.jar -jnlpUrl http://localhost:8080/computer/slaveA/jenkins-agent.jnlp -secret 40892cfa0402db490ff8a9023c7c5d2bffa47fa
bc58 -workDir "C:\Users\acer\Desktop\BatchSlave"
Oct 27, 2022 10:49:22 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using C:\Users\acer\Desktop\BatchSlave\remoting as a remoting work directory
Oct 27, 2022 10:49:22 PM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to C:\Users\acer\Desktop\BatchSlave\remoting
Oct 27, 2022 10:49:22 PM hudson.remoting.jnlp.Main createEngine
INFO: Setting up agent: slaveA
Oct 27, 2022 10:49:22 PM hudson.remoting.Engine startEngine
INFO: Using Remoting version: 3844.vb_946a_e4f72e
Oct 27, 2022 10:49:22 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using C:\Users\acer\Desktop\BatchSlave\remoting as a remoting work directory
Oct 27, 2022 10:49:22 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: locating server among [http://localhost:8080/]
Oct 27, 2022 10:49:22 PM org.jenkinsci.remoting.engine.JnlpAgentEndpointResolver resolve
INFO: Remoting server accepts the following protocols: [JNLPA-connect, Ping]
Oct 27, 2022 10:49:22 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Agent discovery successful
    Agent address: localhost
    Agent port: 50005
    Identity: ae:23:3a:9b:19:31:9f:1c:d4:0e:41:ea:42:53:73:6d
Oct 27, 2022 10:49:22 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Handshaking
Oct 27, 2022 10:49:22 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Connecting to localhost:80805
Oct 27, 2022 10:49:22 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Trying protocol: JNLPA-connect
Oct 27, 2022 10:49:22 PM org.jenkinsci.remoting.protocol.impl.BIONetworkLayer$Reader run
INFO: Waiting for protocol stack to start
Oct 27, 2022 10:49:23 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Remote identity confirmed: ae:23:3a:9b:19:31:9f:1c:d4:0e:41:ea:42:53:73:6d
Oct 27, 2022 10:49:23 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Connected
```

The screenshot displays the Jenkins web interface. At the top, the Jenkins logo is on the left, and a search bar and user profile (Krishna Garg) are on the right. Below the header, a breadcrumb trail shows 'Dashboard' > 'Nodes' > 'slaveA'. The left sidebar contains a list of actions: 'Back to List', 'Status' (highlighted), 'Delete Agent', 'Configure', 'Build History', 'Load Statistics', 'Script Console', 'Log', 'System Information', and 'Disconnect'. The main content area is titled 'Agent slaveA (Jenkins Slave)' and includes a button to 'Mark this node temporarily offline'. Below this, it states 'Agent is connected.' and shows a 'Labels' section with the label 'JenkinsSlaveA'. The 'Projects tied to slaveA' section is currently empty, displaying 'None'.



Conclusion:

From this experiment, it is concluded that we have successfully understood the Jenkins Master – Slave Architecture and scaled our Jenkins standalone implementation by implementing slave nodes.

Hence, we have successfully achieved Lab Outcome (LO4).

Also, we have achieved PO1, PO2, PO3, PO4, PO5, and PO12 from this experiment.