

EXPERIMENT: 6

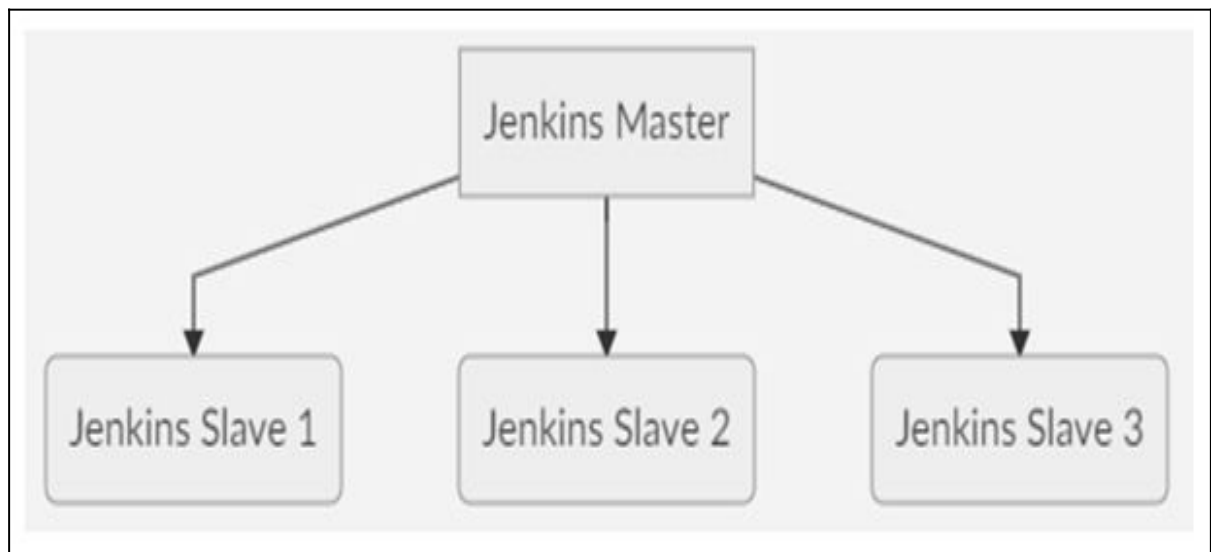
AIM:

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

PROBLEM DEFINITION:

Understand Jenkins Master-Slave architecture and expand a standalone Jenkins setup by adding slave nodes.

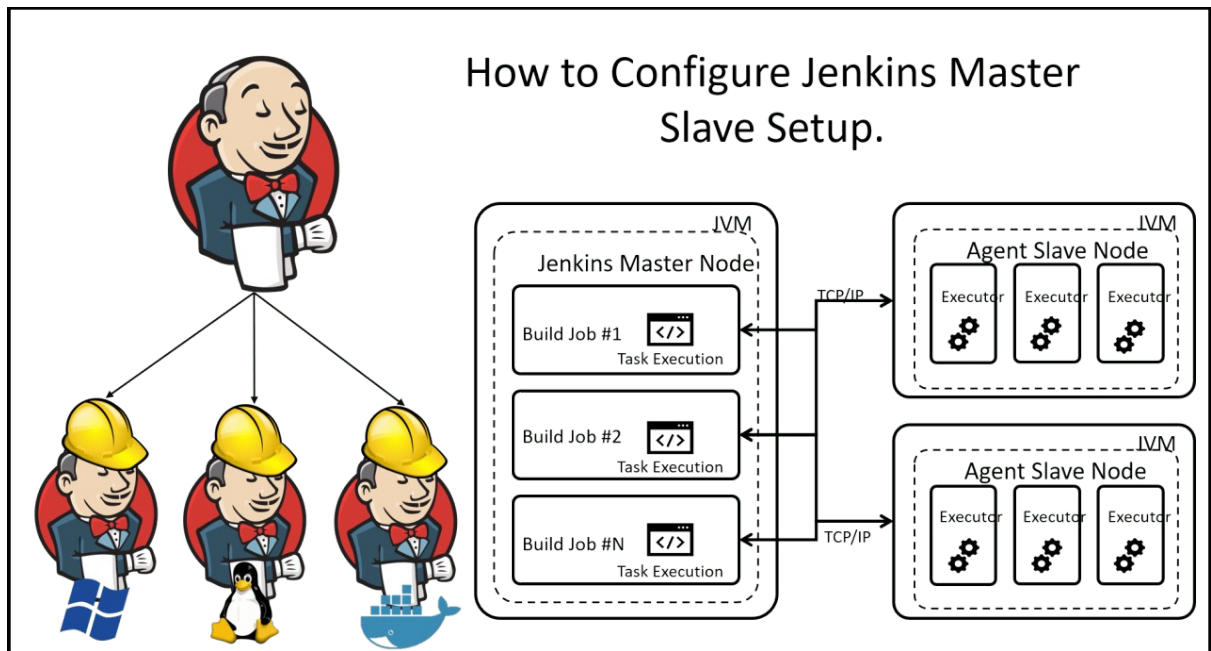
THEORY:



Understanding the Master and Slave Architecture

A standalone Jenkins instance can quickly grow into a resource-intensive application. To manage this, Jenkins can be scaled using a slave node architecture, which offloads some responsibilities from the master Jenkins instance. A Jenkins slave node is a device configured to act as an automation executor on behalf of the master. The Jenkins master represents the base Jenkins installation, performing basic operations and serving the user interface, while the slaves handle the heavy lifting.

This distributed computing model allows the Jenkins master to remain responsive to users while offloading automation execution to connected slave(s). For example, a Jenkins master schedules jobs, assigns them to slaves, sends builds to slaves for execution, monitors slave states (online or offline), and displays build results.



Step to Configure Jenkins Master and Slave Nodes:

1. Click on “Manage Jenkins”

The screenshot shows the Jenkins 'Manage Jenkins' page. At the top, there is a search bar and a user profile icon. Below the search bar, there is a notification about the new version of Jenkins (2.516.2) being available for download. Below this, there is a warning about building on the built-in node being a security issue, with buttons to 'Set up agent', 'Set up cloud', and 'Dismiss'. Below the warning, there is a section titled 'Java 17 end of life in Jenkins' with buttons for 'More Info' and 'Ignore'. At the bottom, there is a section titled 'Warnings have been published for the following currently installed components:' with buttons for 'Go to plugin manager' and 'Configure which of these warnings are shown'. The warnings listed are: 'Git client plugin 6.3.0: File system information disclosure vulnerability' and 'Jakarta Mail API 2.1.3-2: SMTP command injection vulnerability'. Both warnings mention that a fix is available and point to the 'plugin manager' to update the plugin.

2. Click on “Manage Nodes”

Nodes + New Node Configure Monitors Refresh

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Windows 11 (amd64)	In sync	66.50 GiB	12.60 GiB	66.50 GiB	0ms
	slave		N/A	N/A	N/A	N/A	N/A
	slave-1	Windows 11 (amd64)	In sync	66.50 GiB	12.59 GiB	66.50 GiB	100ms
Data obtained		22 min	22 min	22 min	22 min	22 min	22 min

Icon: S M L Legend

3. Select “New Node” and enter the node name in the “Node Name” field.

New node

Node name

slave-1

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.

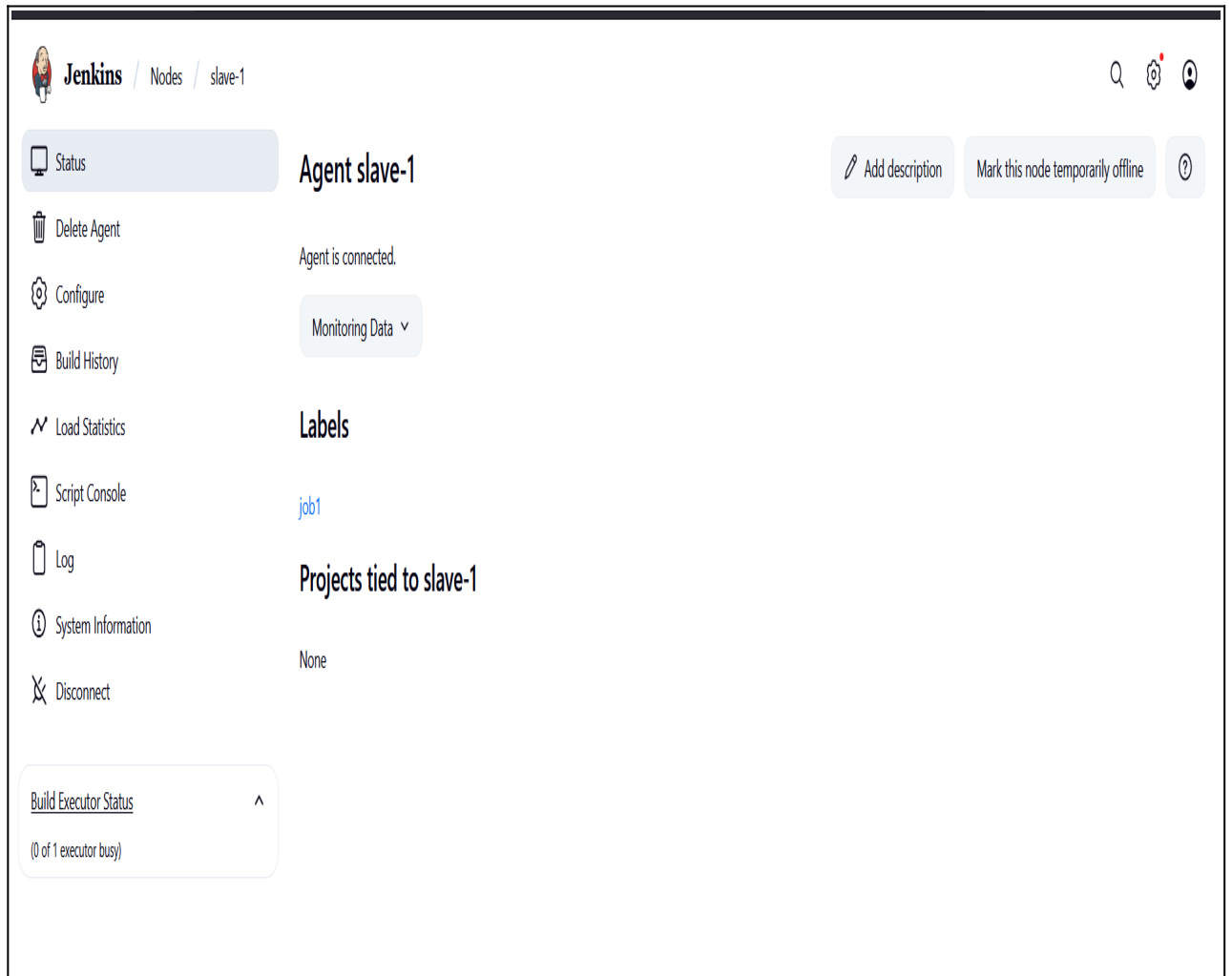
Create

4. Enter the required information.
5. Enter the Hostname in the “Host” field.
6. Click the “Add” button to add credentials and click “Jenkins.”
7. Enter Username, Password, ID, and Description.

The screenshot shows the Jenkins configuration page for a new node. The breadcrumb navigation at the top is 'Jenkins / Manage Jenkins / Nodes'. The page contains several input fields: 'Name' with the value 'slave-2', 'Description' (empty), 'Number of executors' set to '1', 'Remote root directory' set to 'C:\Program Files\Jenkins\remoting', and 'Labels' set to 'job1'. There is a 'Save' button at the bottom left of the form area.

8. Select the dropdown menu to add credentials into the “Credentials” field.
9. Select the dropdown to add the Host Key Verifications Strategy under “Non-verifying Verification Strategy.”
10. Select “Keep this agent online as much as possible” in the "Availability" field.

```
PS C:\Program Files\Jenkins\remoting> java -jar agent.jar -url http://localhost:8080/ -secret 12f110ef6cb10cf3c23fad6fd227f7717d26894d0d9b014c11425289076ca3
06 -name "slave-1" -webSocket -workDir "C:\Program Files\Jenkins\remoting"
Sep 10, 2025 6:13:02 AM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using C:\Program Files\Jenkins\remoting\remoting as a remoting work directory
Sep 10, 2025 6:13:02 AM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to C:\Program Files\Jenkins\remoting\remoting
Sep 10, 2025 6:13:02 AM hudson.remoting.Launcher createEngine
INFO: Setting up agent: slave-1
Sep 10, 2025 6:13:02 AM hudson.remoting.Engine startEngine
INFO: Using Remoting version: 3309.v27b_9314fd1a_4
Sep 10, 2025 6:13:02 AM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using C:\Program Files\Jenkins\remoting\remoting as a remoting work directory
Sep 10, 2025 6:13:02 AM hudson.remoting.Launcher$CuiListener status
INFO: WebSocket connection open
Sep 10, 2025 6:13:02 AM hudson.remoting.Launcher$CuiListener status
INFO: Connected
```



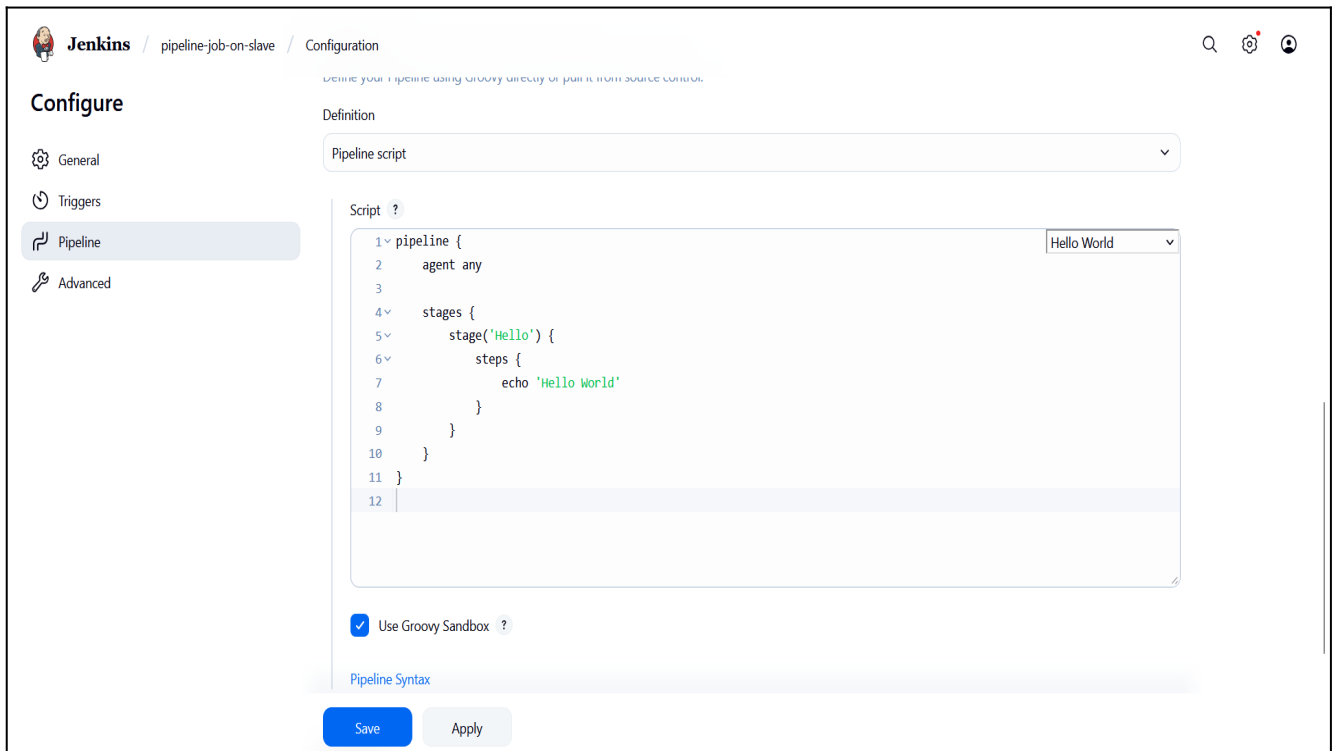
Creating a Freestyle Project and Running on the Slave Machine:

1. Click “New Item”, which will redirect to the job’s new page.
2. Enter the project name, select “Freestyle Project”, and click OK.
3. On the job’s page, click the “Build Now” button to execute your pipeline.
4. Verify the history of the executed build number under “Build History” by checking the build status and output on the remote host.
5. Click on the build number and select “Console Output” to view the executed job and output on the remote host.

Creating a Pipeline and Running on the Slave Machine:

1. Click “New Item” in the top left corner on the dashboard.
2. Enter the project name in the “Enter an item name” field, select the “Pipeline” project, and click OK.
3. Give a description (optional).
4. Go to the “Pipeline” section.
5. Copy and paste the desired Pipeline script into the script field.
6. Save the changes, which will redirect to the Pipeline new page.
7. On the left pane, click the “Build Now” button to execute your pipeline.
8. Verify the history of executed builds under “Build History” by clicking the build status.
9. On each build, the output will display on the results page.

- Click on the build number and select “Console Output” to see that the pipeline ran on a slave machine.



This screenshot shows the Jenkins configuration page for a pipeline named 'pipeline-job-on-slave'. The left sidebar contains navigation links: General, Triggers, Pipeline (selected), and Advanced. The main area is titled 'Configure' and shows the 'Definition' tab. A dropdown menu is set to 'Pipeline script'. Below this is a 'Script' editor with a Groovy pipeline script. The script defines a pipeline with an 'any' agent, a 'Hello' stage, and an 'echo' step that outputs 'Hello World'. A 'Hello World' dropdown menu is visible at the top right of the script editor. Below the script editor, there is a checkbox for 'Use Groovy Sandbox' which is checked. At the bottom, there are 'Save' and 'Apply' buttons.

Jenkins / pipeline-job-on-slave / Configuration

Configure your pipeline using Groovy syntax or pull it from source control.

Definition

Pipeline script

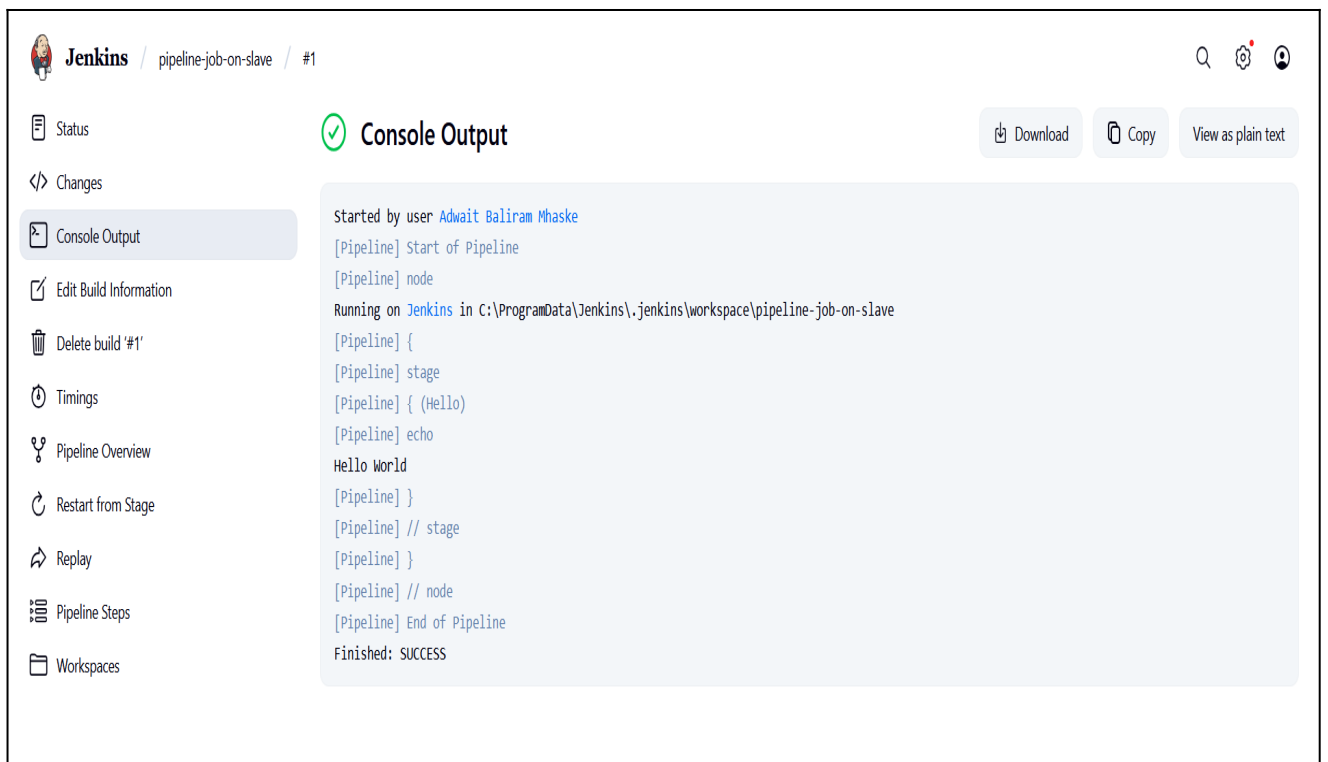
Script ?

```
1 pipeline {
2   agent any
3
4   stages {
5     stage('Hello') {
6       steps {
7         echo 'Hello World'
8       }
9     }
10  }
11 }
12
```

☒ Use Groovy Sandbox ?

Pipeline Syntax

Save Apply



This screenshot shows the Jenkins console output for build #1 of the 'pipeline-job-on-slave' pipeline. The left sidebar contains navigation links: Status, Changes, Console Output (selected), Edit Build Information, Delete build '#1', Timings, Pipeline Overview, Restart from Stage, Replay, Pipeline Steps, and Workspaces. The main area is titled 'Console Output' and shows the output of the pipeline. The output starts with 'Started by user Adwait Baliram Mhaske' and ends with 'Finished: SUCCESS'. The output also shows the pipeline stages and steps, including 'Start of Pipeline', 'node', 'Running on Jenkins in C:\ProgramData\Jenkins\jenkins\workspace\pipeline-job-on-slave', 'stage', 'Hello', 'echo', 'Hello World', and 'End of Pipeline'.

Jenkins / pipeline-job-on-slave / #1

Console Output

Started by user [Adwait Baliram Mhaske](#)

[Pipeline] Start of Pipeline

[Pipeline] node

Running on [Jenkins](#) in C:\ProgramData\Jenkins\jenkins\workspace\pipeline-job-on-slave

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Hello)

[Pipeline] echo

Hello World

[Pipeline] }

[Pipeline] // stage

[Pipeline] }

[Pipeline] // node

[Pipeline] End of Pipeline

Finished: SUCCESS