EXPERIMENT: 6

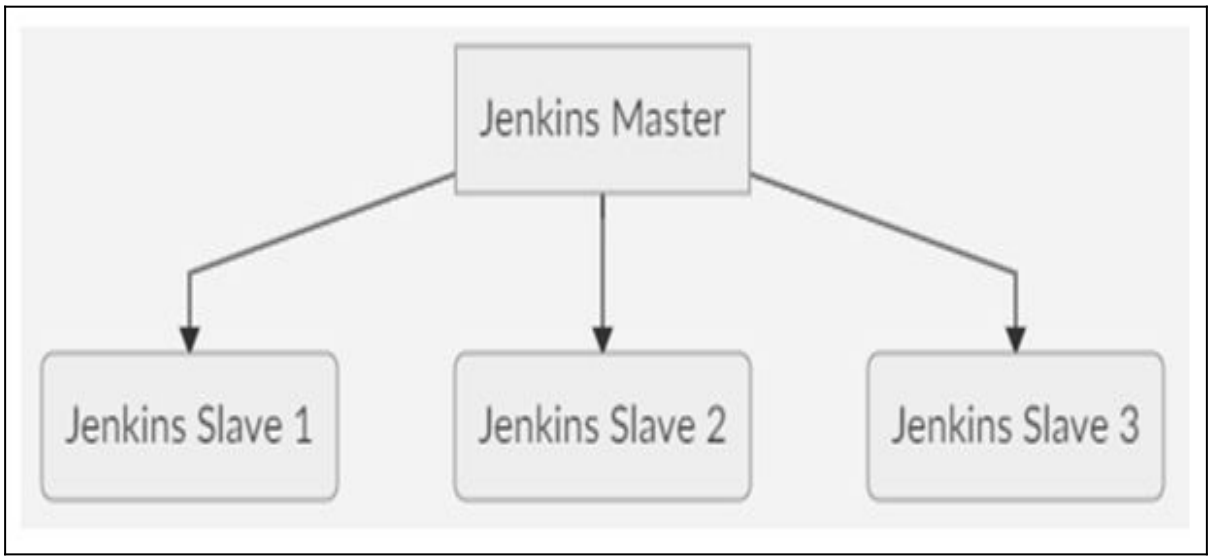
AIM:

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.

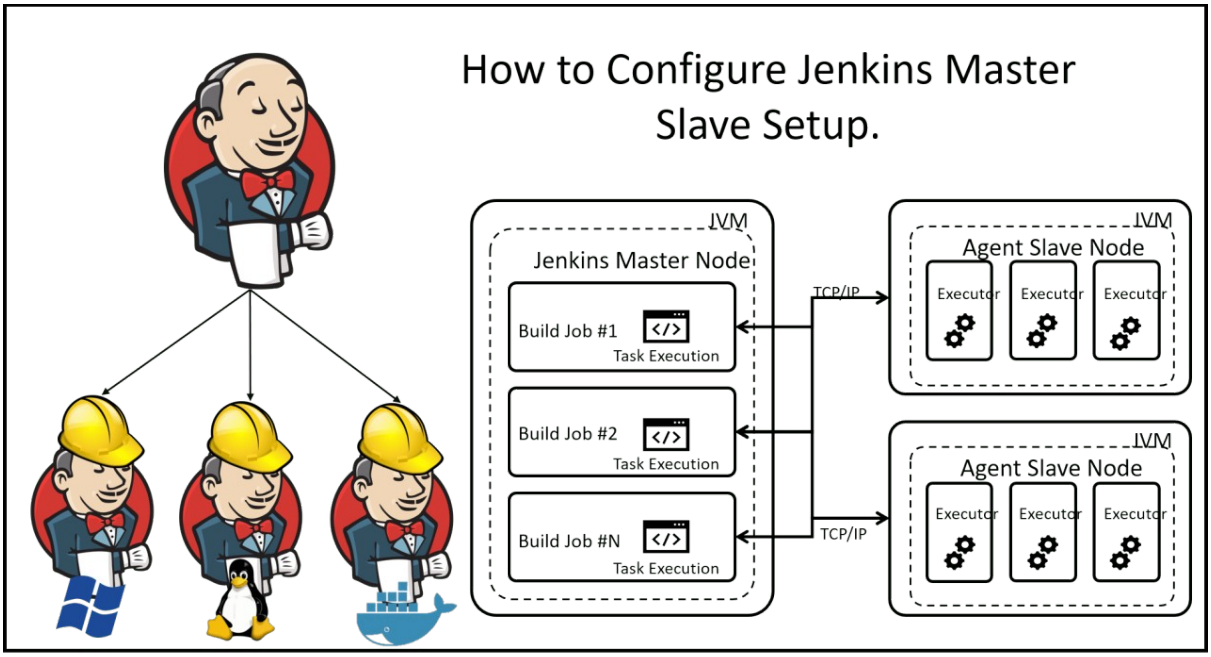
THEORY:



Understanding the Master and Slave Architecture

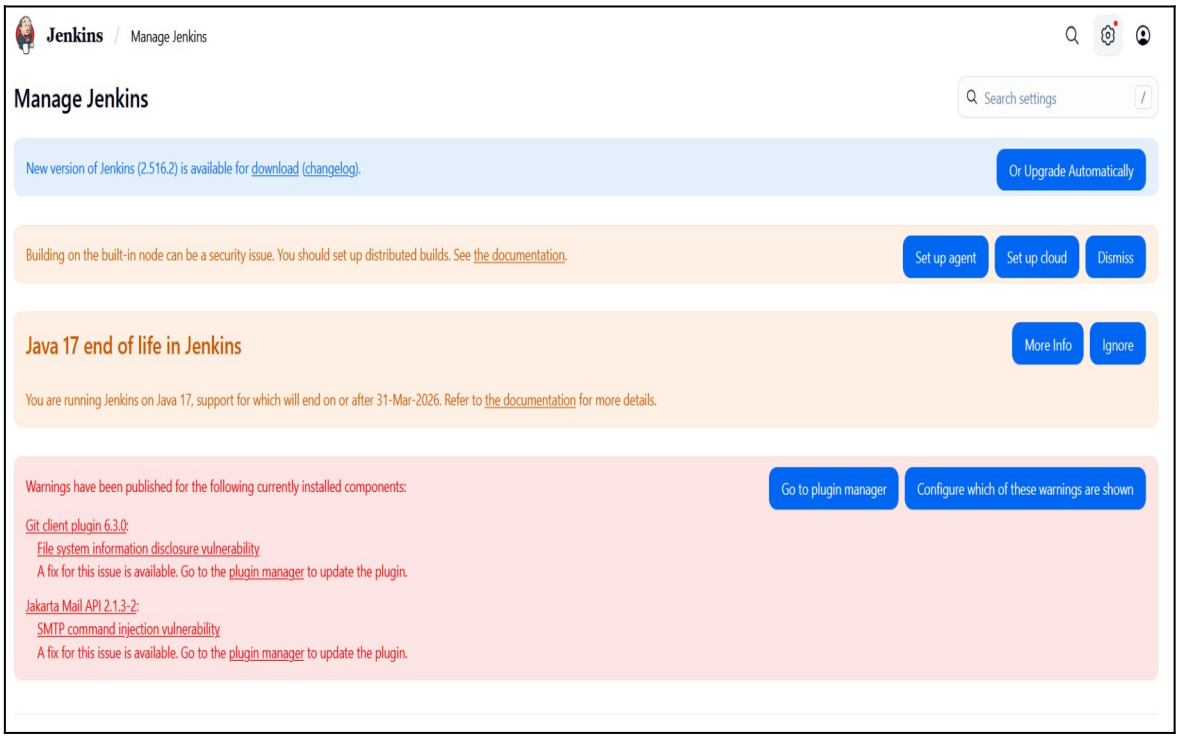
A standalone Jenkins instance can quickly grow into a resource-intensive application. To manage this, Jenkins can be scaled using a slave node architecture, which offloads some responsibilities from the master Jenkins instance. A Jenkins slave node is a device configured to act as an automation executor 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.

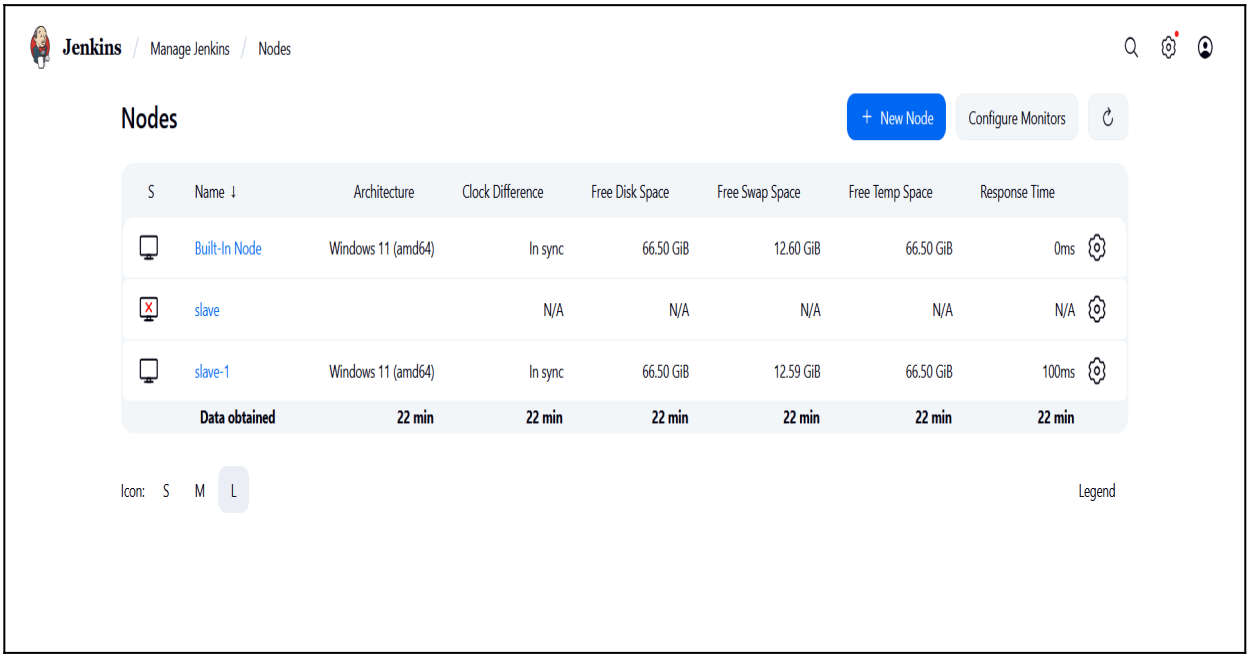


Step to Configure Jenkins Master and Slave Nodes:

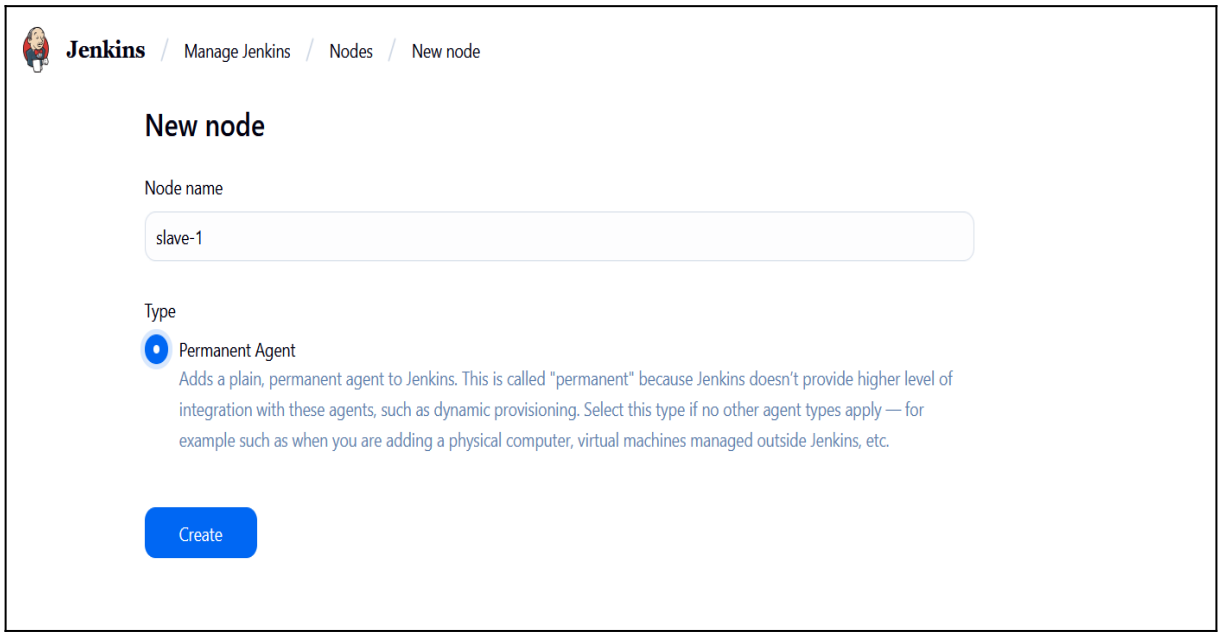
1. Click on “Manage Jenkins”



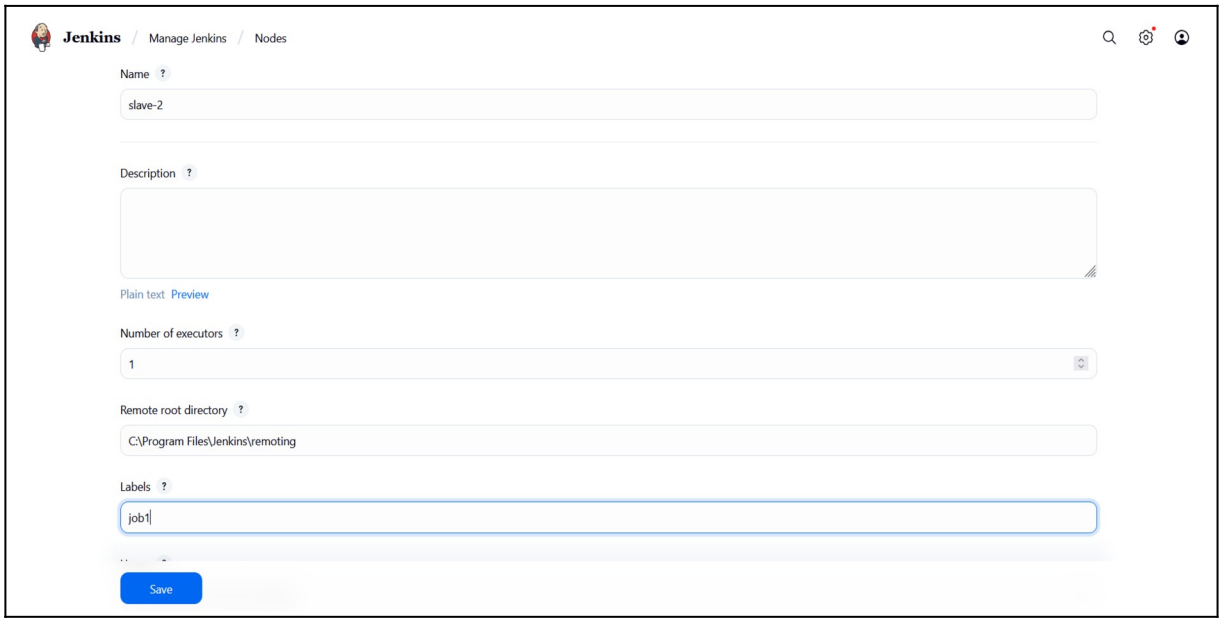
2. Click on “Manage Nodes”



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

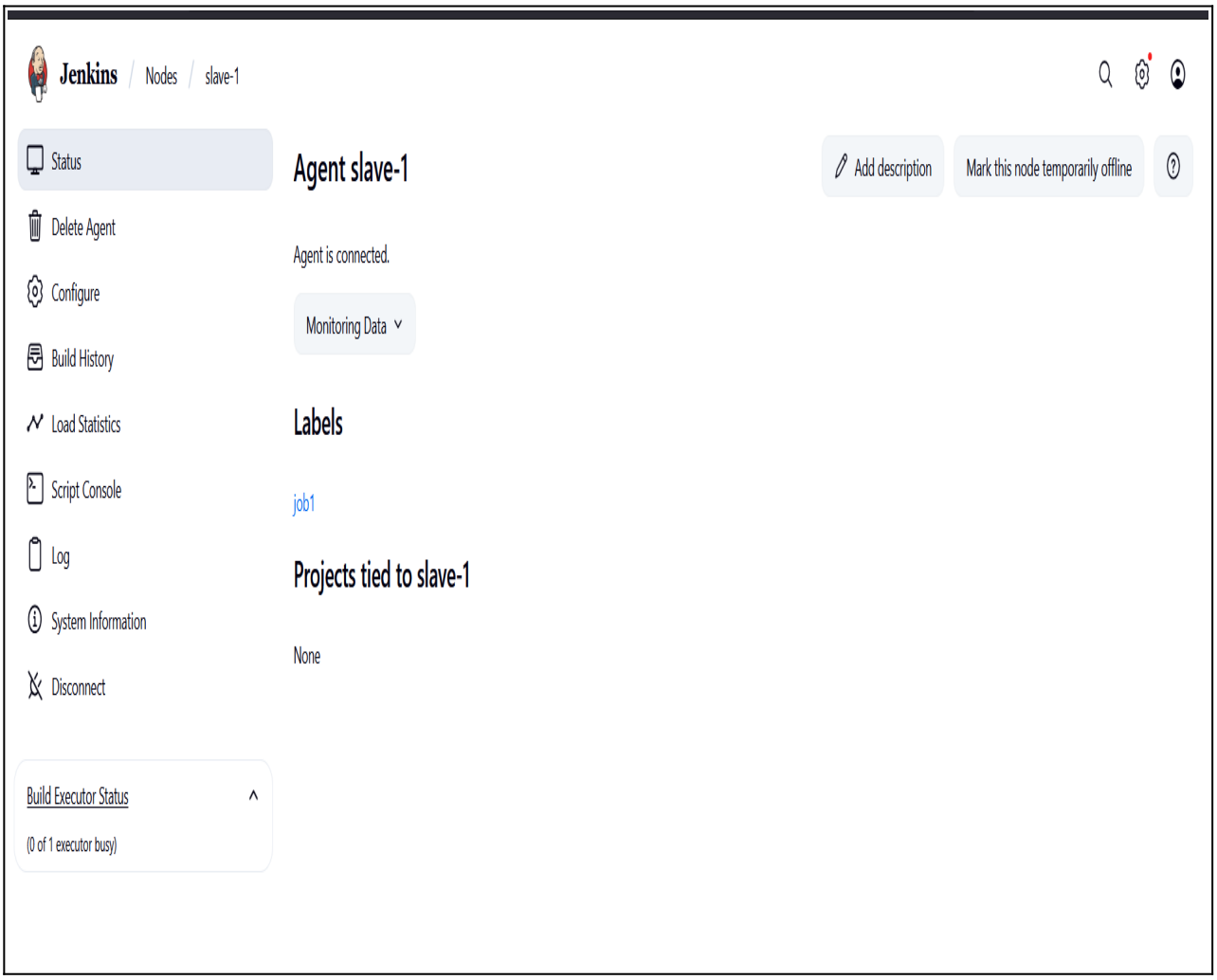


4. Enter the required information.
5. Enter the Hostname in the “Host” field.
6. Click the “Add” button to add credentials and click “Jenkins.”
7. Enter Username, Password, ID, and Description.



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 12f110ef6cb10cf3c23fad6fd227f7717d26894d0d9b014c11425289076ca306 -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.

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