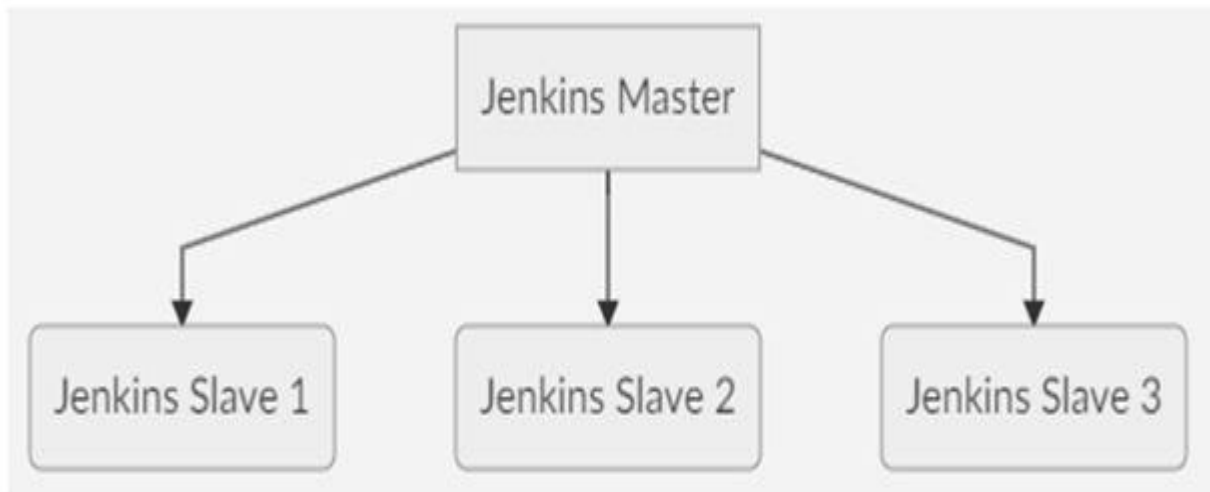


EXPERIMENT NO. 6

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

THEORY :-

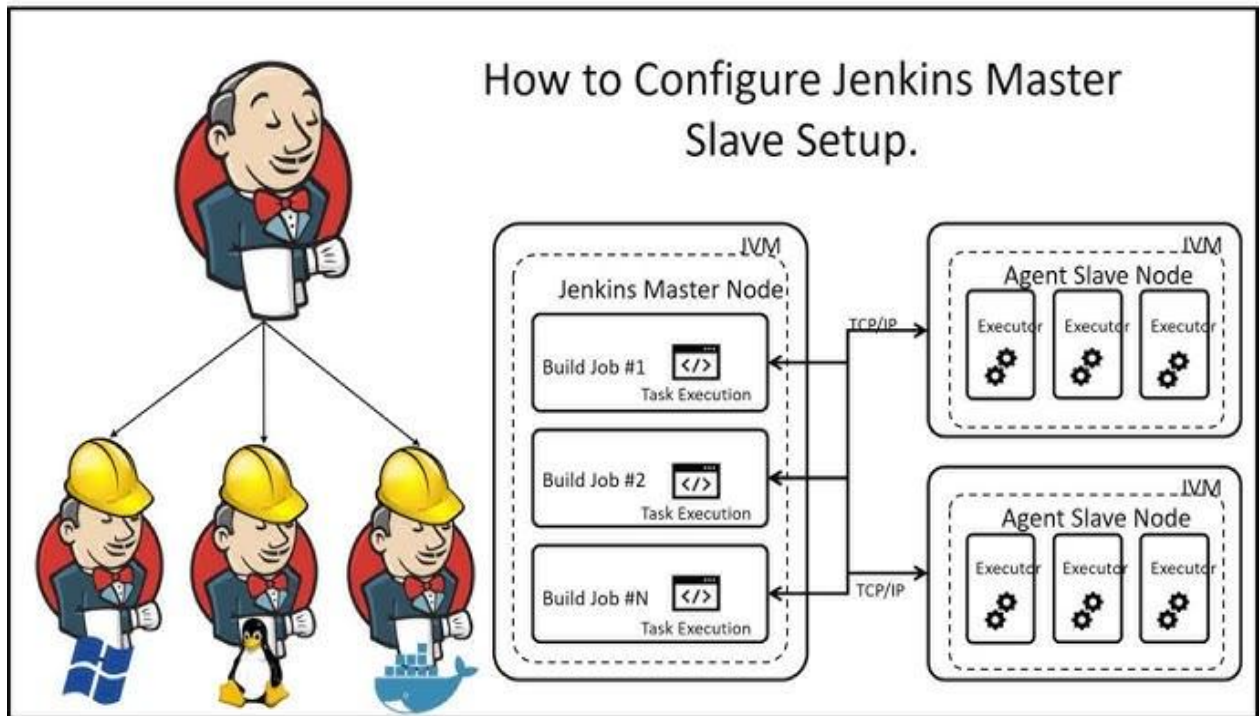


Understanding the Master and Slave Architecture

A standalone Jenkins instance can quickly grow into a resource-intensive application.

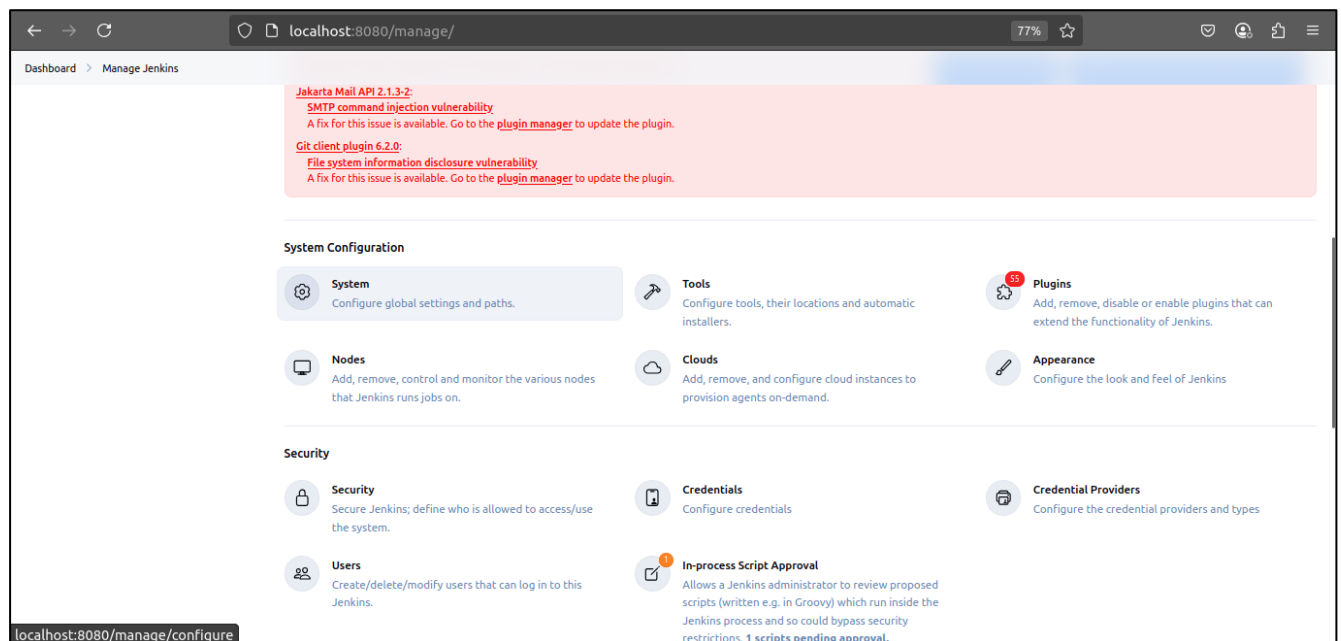
To manage this, Jenkins can be scaled using a slave node architecture, which offloads some responsibilities from the master Jenkins instance. A Jenkins slave node is a device configured to act as an automation executor on behalf of the master. The Jenkins master represents the base Jenkins installation, 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.

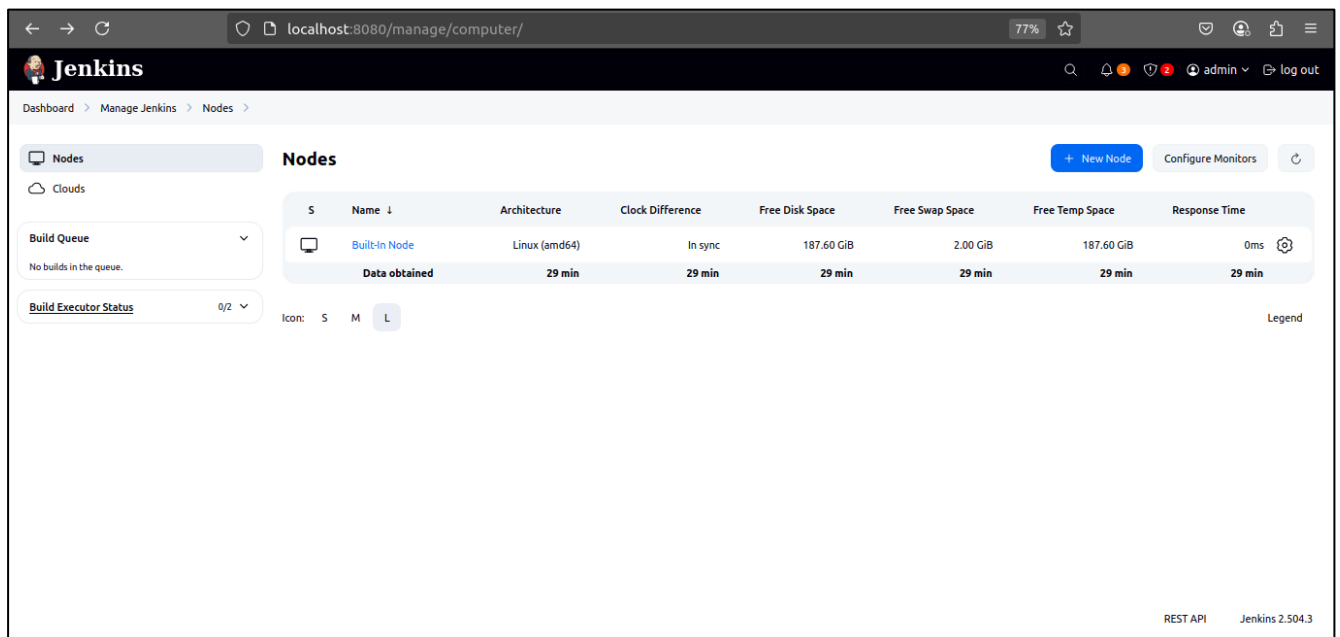


Steps to Configure Jenkins Master and Slave Nodes:

1. Click on “Manage Jenkins”

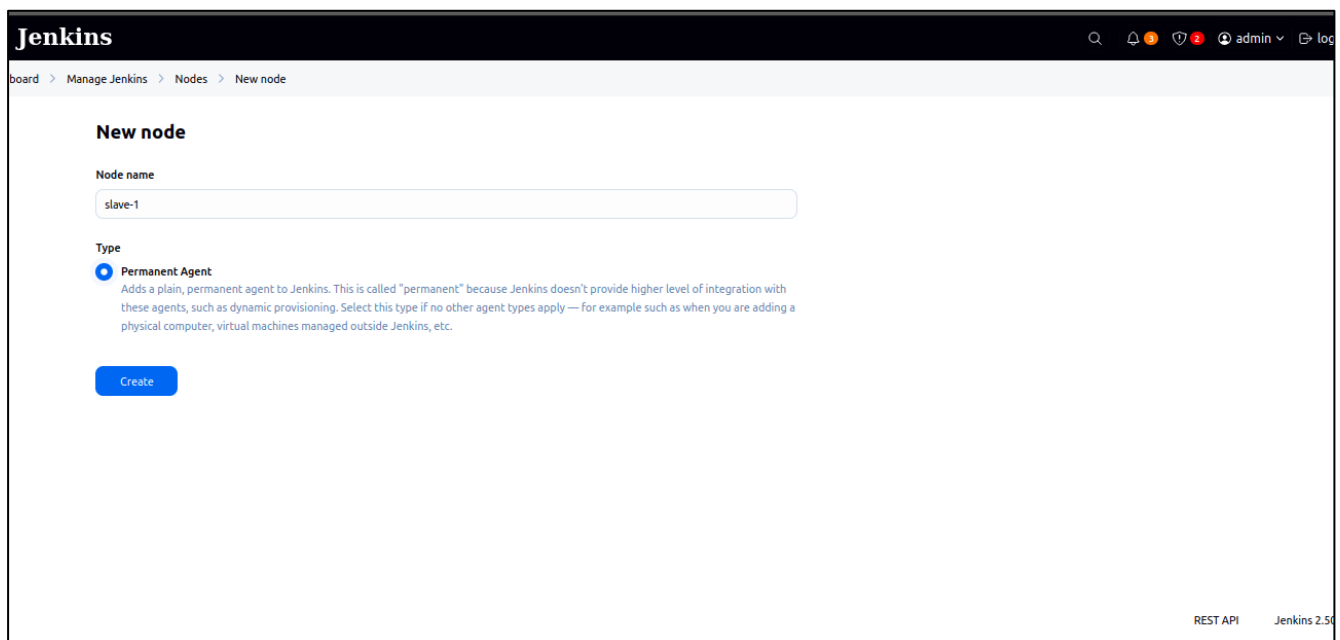


2. Click On “Manage Nodes”



The screenshot shows the Jenkins web interface at `localhost:8080/manage/computer/`. The breadcrumb navigation is `Dashboard > Manage Jenkins > Nodes`. On the left sidebar, the `Nodes` tab is selected. The main content area is titled `Nodes` and features a `+ New Node` button and a `Configure Monitors` button. Below these is a table with the following columns: `S`, `Name`, `Architecture`, `Clock Difference`, `Free Disk Space`, `Free Swap Space`, `Free Temp Space`, and `Response Time`. The table contains one entry, `Built-In Node`, with values: `Linux (amd64)`, `In sync`, `187.60 GiB`, `2.00 GiB`, `187.60 GiB`, and `0ms`. Below the table, there is a `Data obtained` row showing `29 min` for each of the last five metrics. At the bottom right, it says `REST API` and `Jenkins 2.504.3`.

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



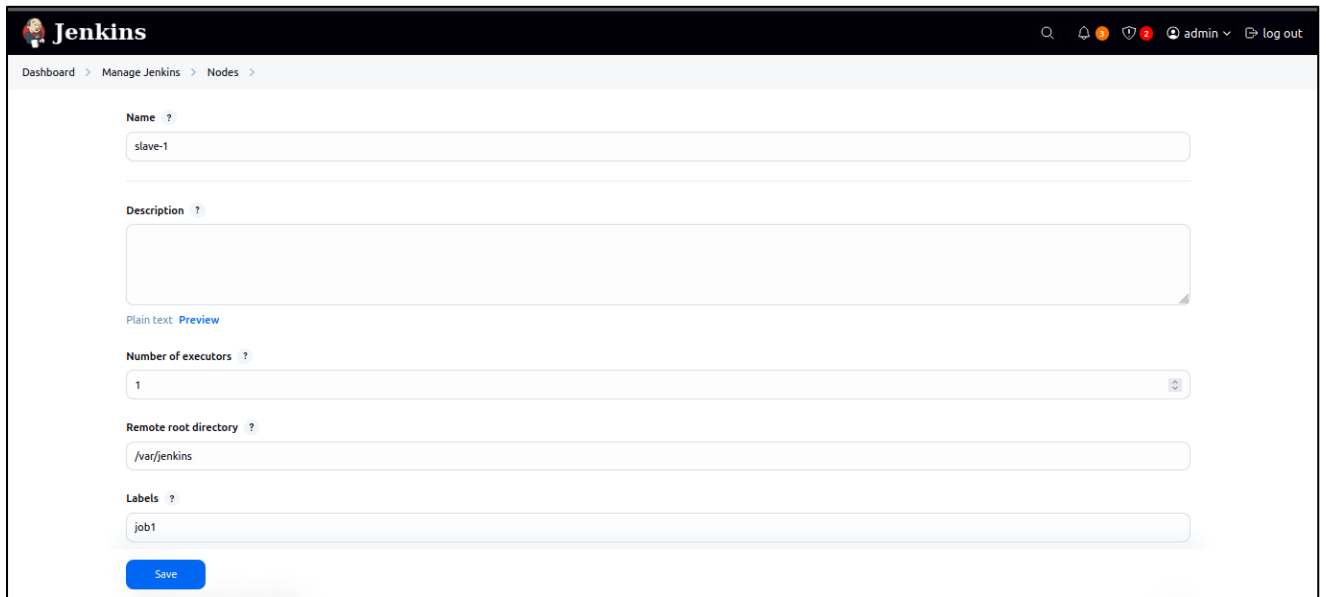
The screenshot shows the Jenkins web interface at `localhost:8080/manage/computer/new`. The breadcrumb navigation is `Dashboard > Manage Jenkins > Nodes > New node`. The main content area is titled `New node`. It has a `Node name` field with the value `slave-1`. Below this is a `Type` section with a radio button selected for `Permanent Agent`. A description for `Permanent Agent` is provided: `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.` At the bottom, there is a `Create` button. At the bottom right, it says `REST API` and `Jenkins 2.504.3`.

4. Select "Permanent Agent" and click "OK." Initially, only "Permanent Agent" will be available. After adding one or more slaves, the "Copy Existing Node" option will appear.

5. Enter the required information.

6. Enter the Hostname in the "Host" field.

7. Click the "Add" button to add credentials and click "Jenkins."



The screenshot shows the Jenkins 'Add New Node' configuration page. The breadcrumb trail is 'Dashboard > Manage Jenkins > Nodes >'. The form contains the following fields:

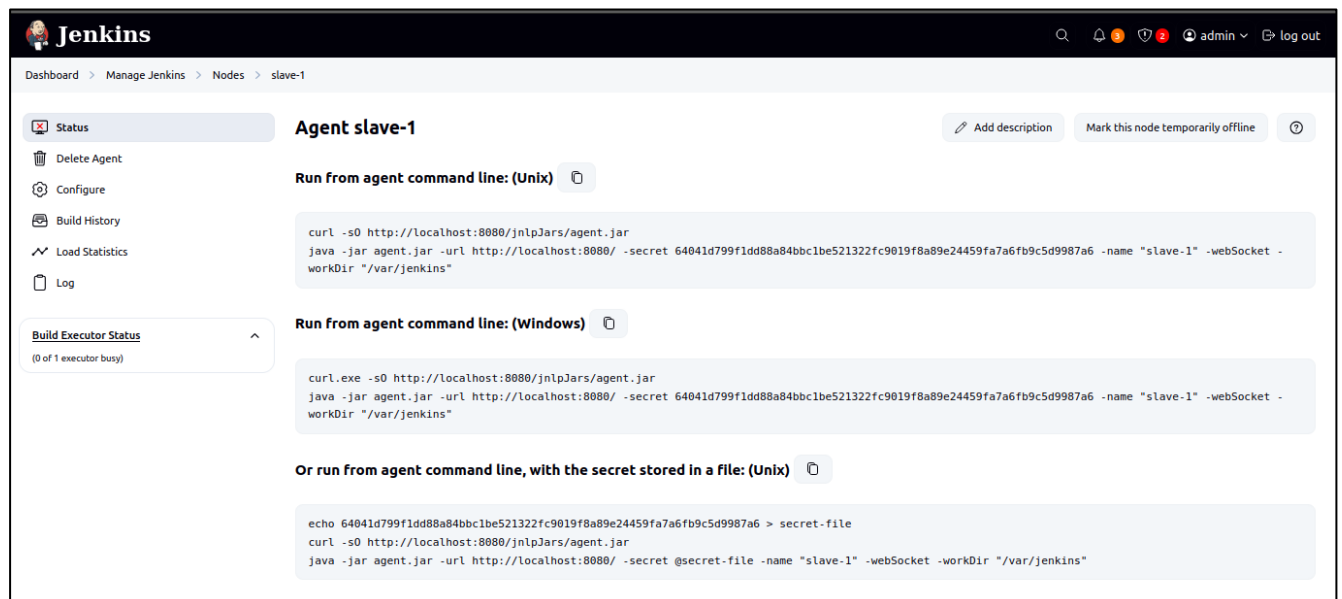
- Name**: A text input field containing 'slave-1'.
- Description**: A large text area for a description.
- Plain text**: A link labeled 'Preview'.
- Number of executors**: A dropdown menu set to '1'.
- Remote root directory**: A text input field containing '/var/jenkins'.
- Labels**: A text input field containing 'job1'.
- Save**: A blue button at the bottom left.

8. Enter Username, Password, ID, and Description.

9. Select the dropdown menu to add credentials in the "Credentials" field.

10. Select the dropdown to add the Host Key Verification Strategy under "Nonverifying Verification Strategy."

11. Select "Keep this agent online as much as possible" in the "Availability" field.



The screenshot shows the Jenkins 'Agent slave-1' configuration page. The breadcrumb trail is 'Dashboard > Manage Jenkins > Nodes > slave-1'. The page has a sidebar on the left with links: Status, Delete Agent, Configure, Build History, Load Statistics, and Log. The main content area is titled 'Agent slave-1' and includes the following sections:

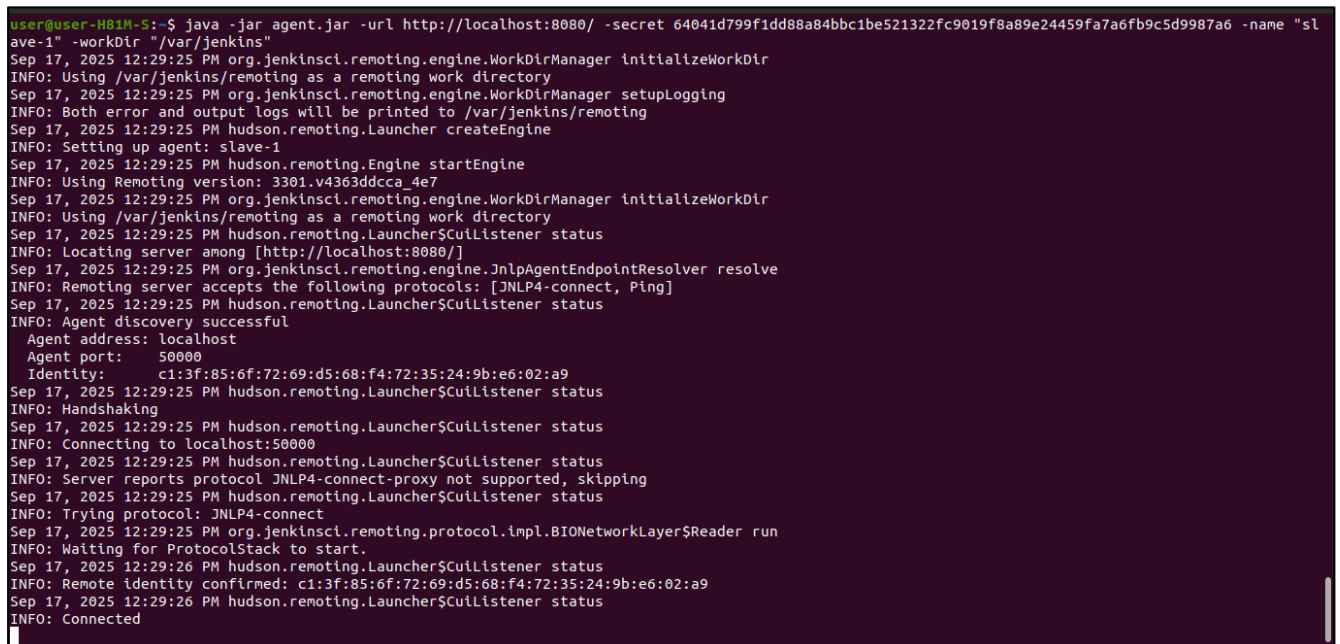
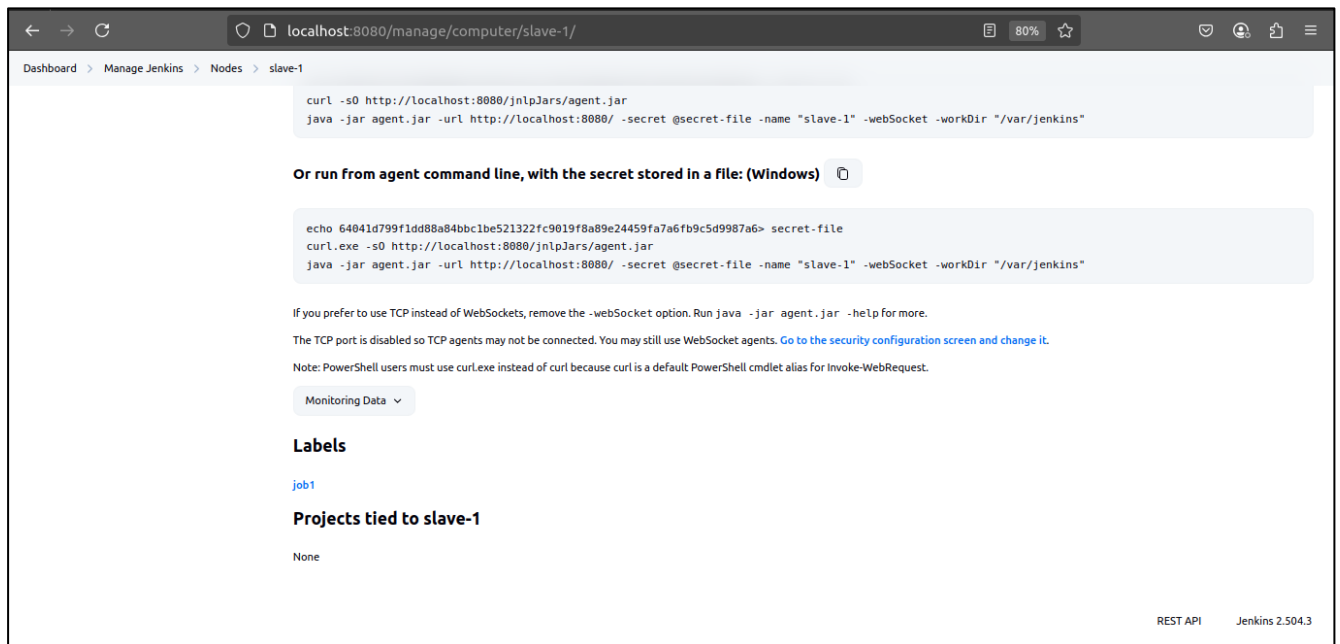
- Run from agent command line: (Unix)**: A code block containing:

```
curl -s0 http://localhost:8080/jnlpJars/agent.jar
java -jar agent.jar -url http://localhost:8080/ -secret 64041d799f1dd88a84bbc1be521322fc9b19f8a89e24459fa7a6fb9c5d9987a6 -name "slave-1" -webSocket -
workDir "/var/jenkins"
```
- Run from agent command line: (Windows)**: A code block containing:

```
curl.exe -s0 http://localhost:8080/jnlpJars/agent.jar
java -jar agent.jar -url http://localhost:8080/ -secret 64041d799f1dd88a84bbc1be521322fc9b19f8a89e24459fa7a6fb9c5d9987a6 -name "slave-1" -webSocket -
workDir "/var/jenkins"
```
- Or run from agent command line, with the secret stored in a file: (Unix)**: A code block containing:

```
echo 64041d799f1dd88a84bbc1be521322fc9b19f8a89e24459fa7a6fb9c5d9987a6 > secret-file
curl -s0 http://localhost:8080/jnlpJars/agent.jar
java -jar agent.jar -url http://localhost:8080/ -secret @secret-file -name "slave-1" -webSocket -workDir "/var/jenkins"
```

At the top right of the main content area, there are buttons: 'Add description', 'Mark this node temporarily offline', and a circular icon.

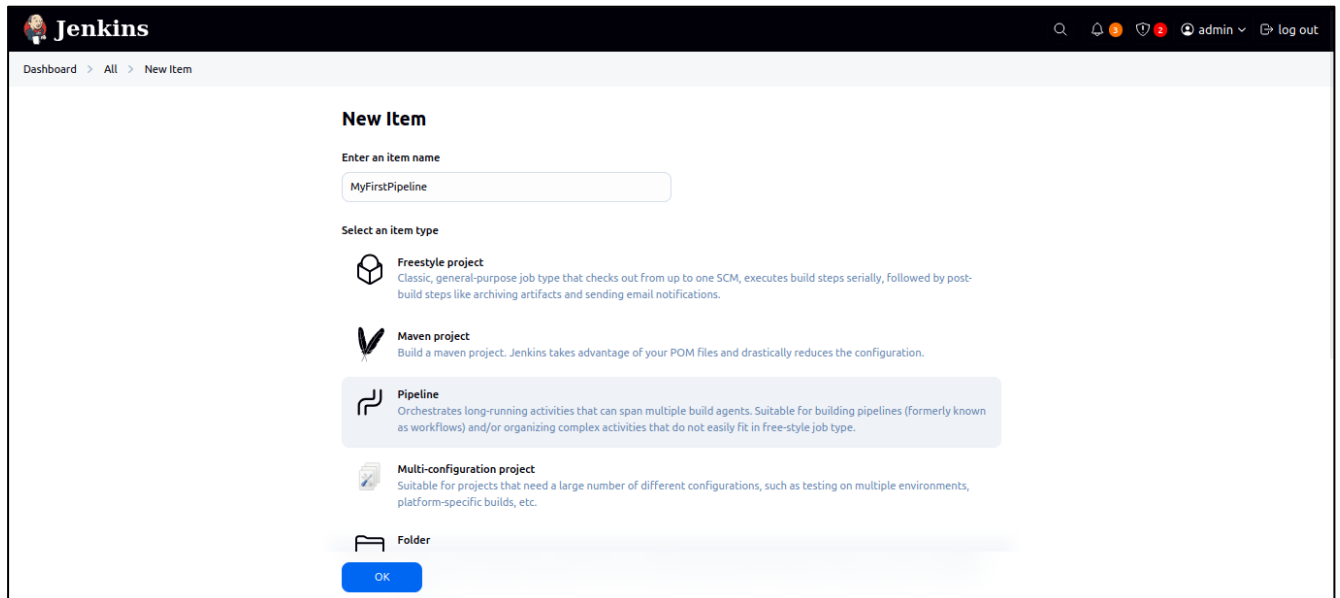


Creating a Freestyle Project and Running on the Slave Machine:

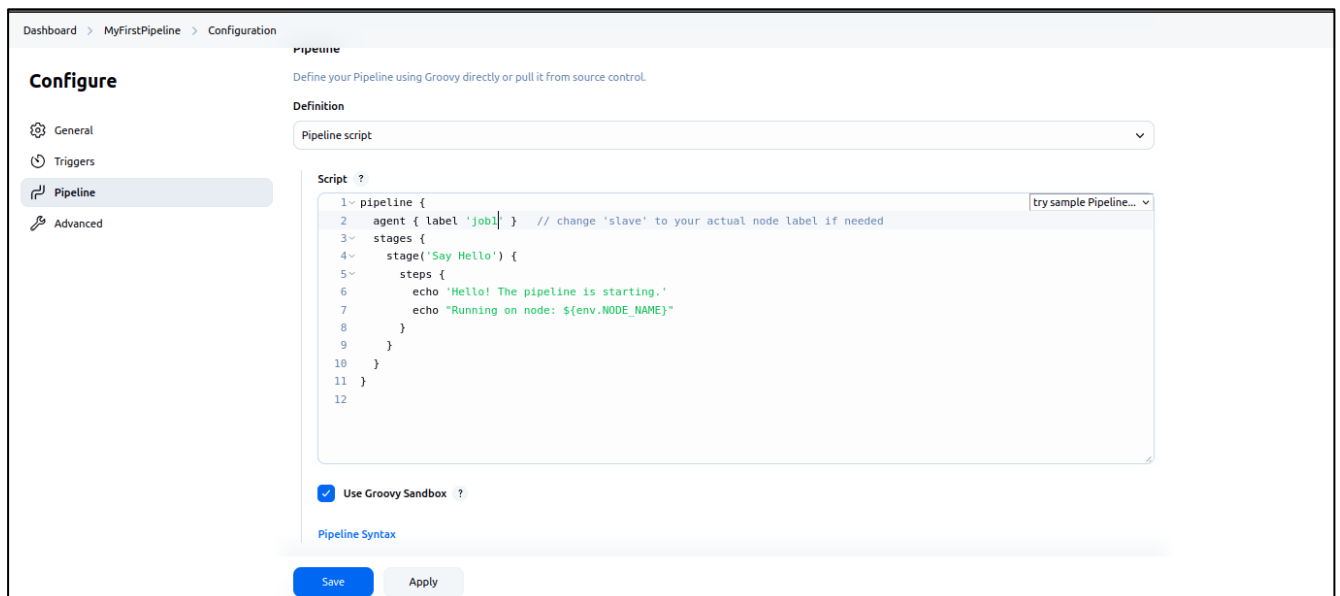
- 1.Click "Save," which will redirect to the job's view page.
- 2.On the left pane, click the "Build Now" button to execute your pipeline.
- 3.Verify the history of the executed build under "Build History" by clicking the build number.
- 4.Click on the build number and select "Console Output" to view 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 name of your project in the "Enter an item name" field, select the "Pipeline" project, and click "OK."

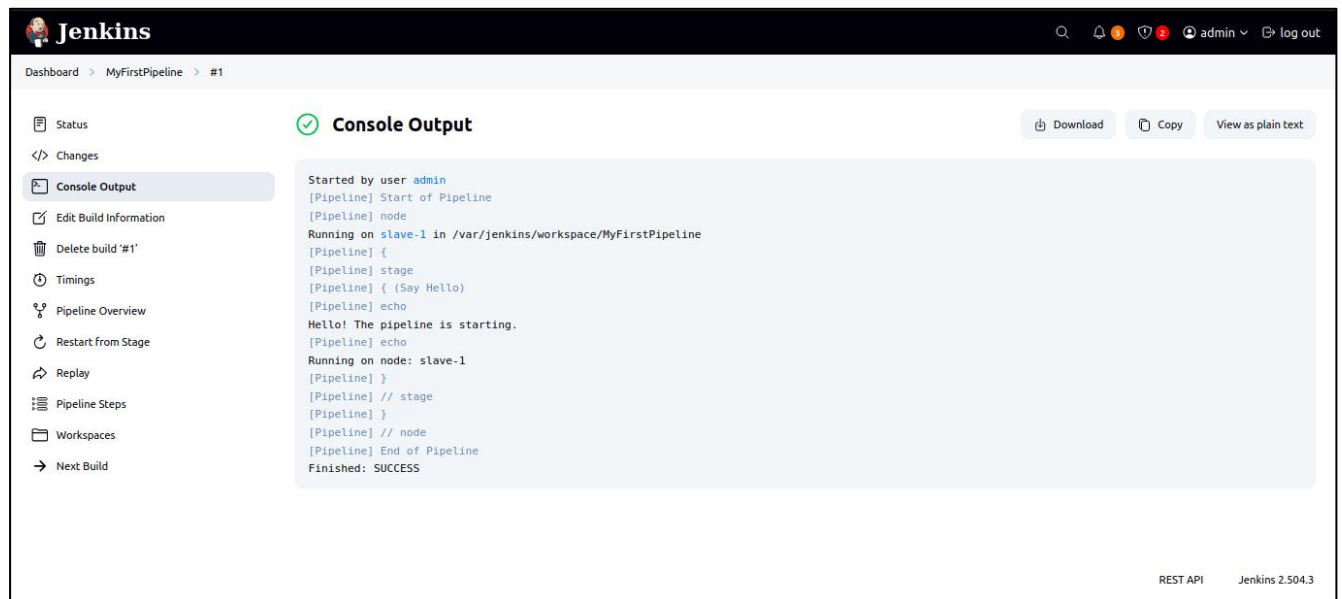


3. Enter a description (optional).
4. Go to the "Pipeline" section, ensure the "Definition" field is set to "Pipeline script."
5. Copy and paste the declarative Pipeline script into the script field.
6. Click "Save," which will redirect to the Pipeline view page.



7. On the left pane, click the "Build Now" button to execute your pipeline.
8. After execution, the Pipeline view will display the results.
9. Verify the history of executed builds under "Build History" by clicking the build number.

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



CONCLUSION

Thus, understanding Jenkins Master-Slave architecture and implementing slave nodes allowed us to scale Jenkins installations and improve continuous integration efficiency.

