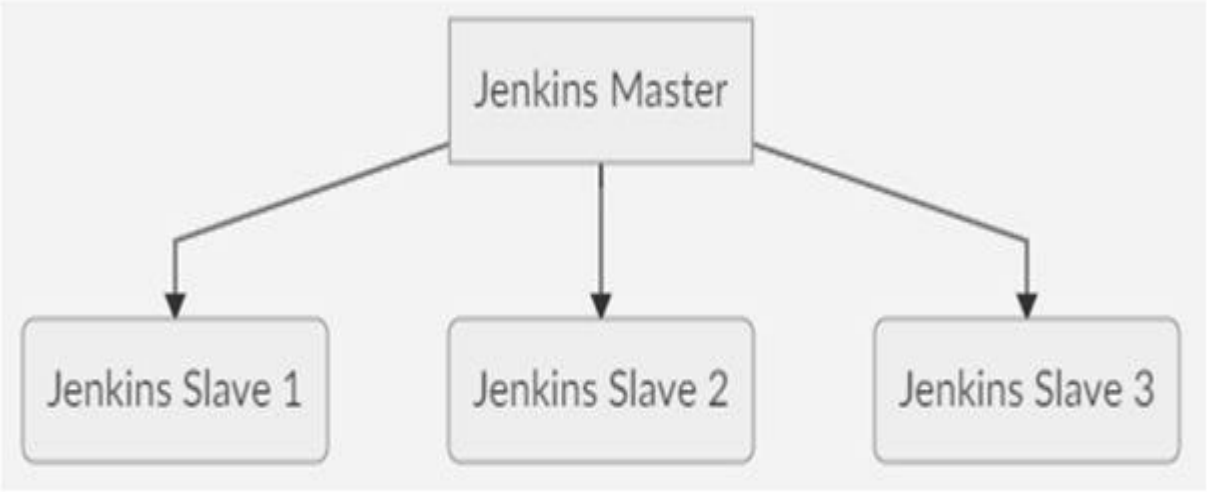
EXPERIMENT NO. 6

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

THEORY :-

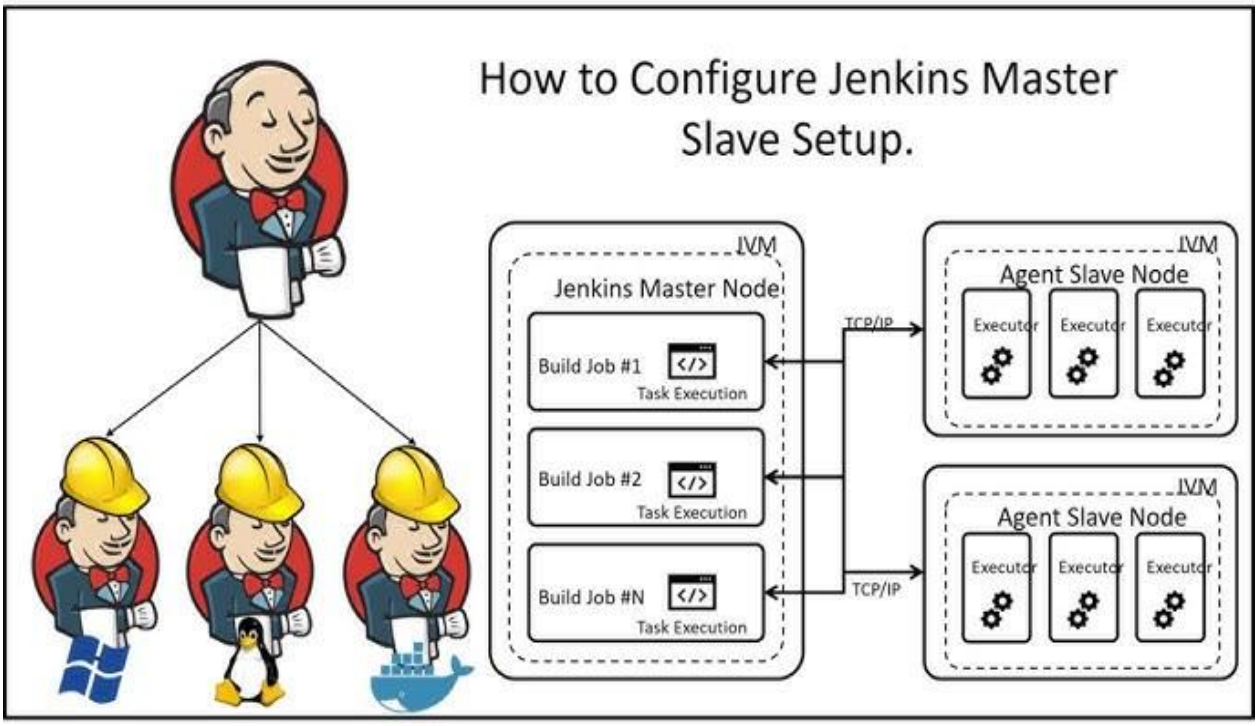


Understanding the Master and Slave Architecture

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

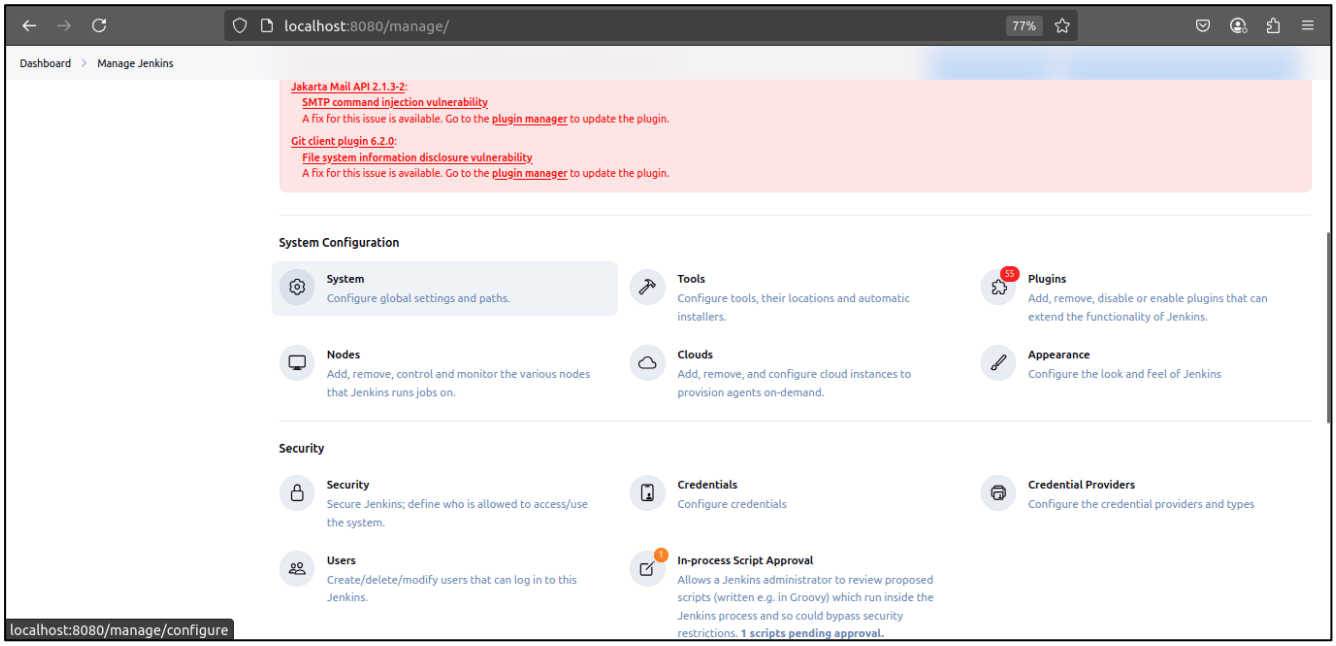
To manage this, Jenkins can be scaled using a slave node architecture, which offloads some responsibilities from the master Jenkins instance. A Jenkins slave node is a device configured to act as an automation executor on behalf of the master. The Jenkins master represents the base Jenkins installation,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

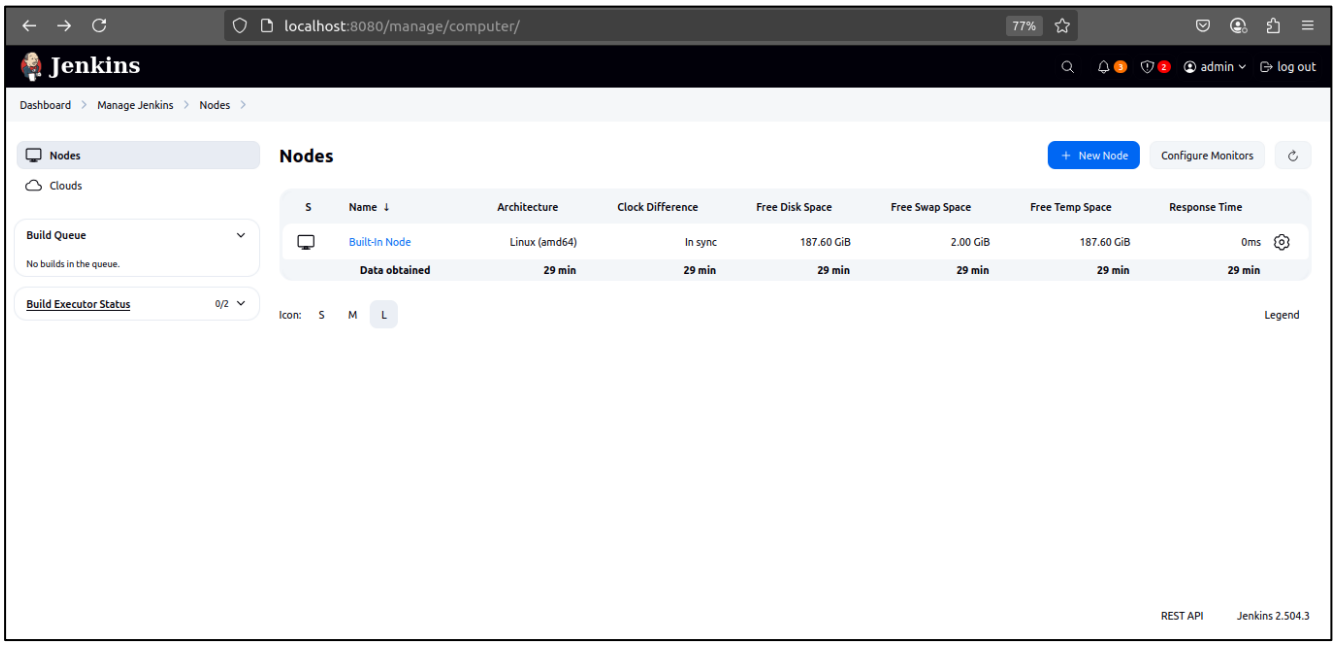


Steps to Configure Jenkins Master and Slave Nodes:

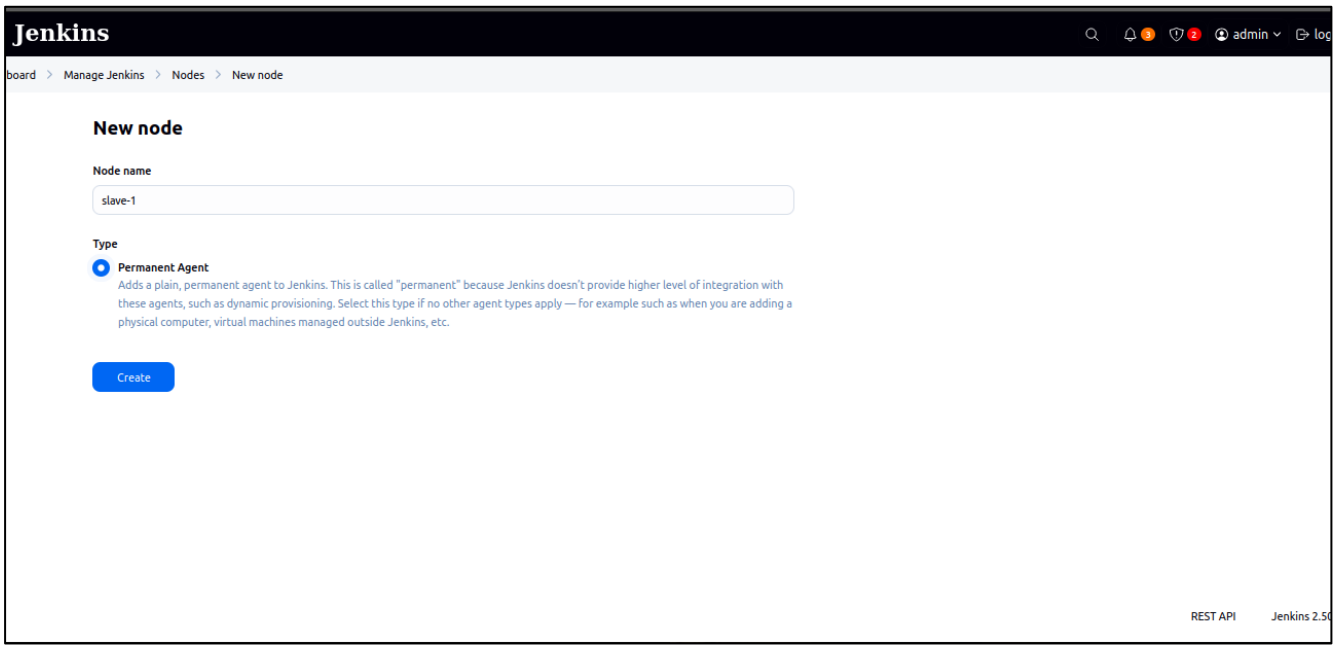
1.Click on “Manage Jenkins”



2.Click On “Manage Nodes”



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

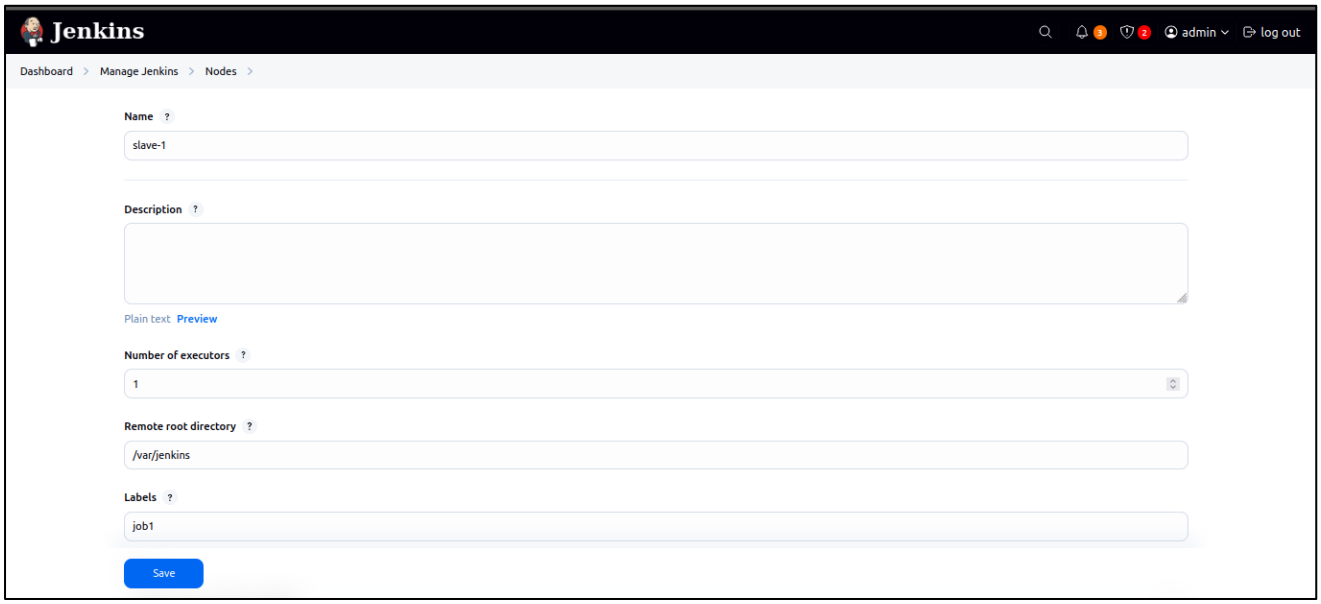


4. Select "Permanent Agent" and click "OK." Initially, only "Permanent Agent" will be available. After adding one or more slaves, the "Copy Existing Node" option will appear.

5. Enter the required information.

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

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

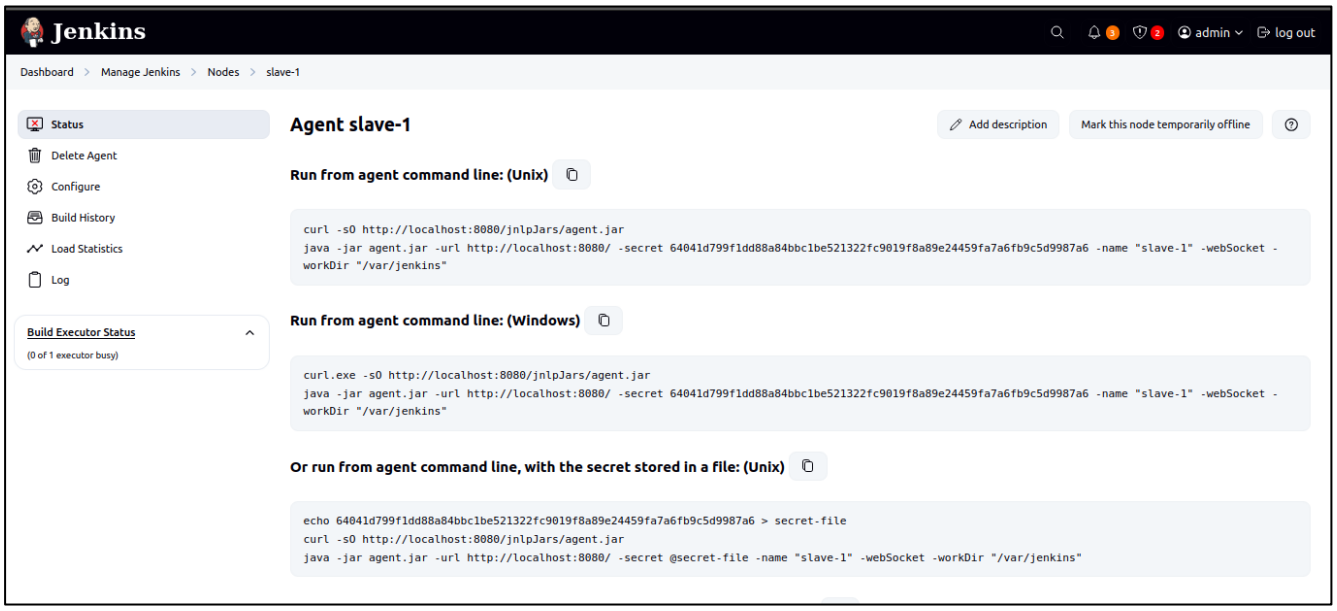


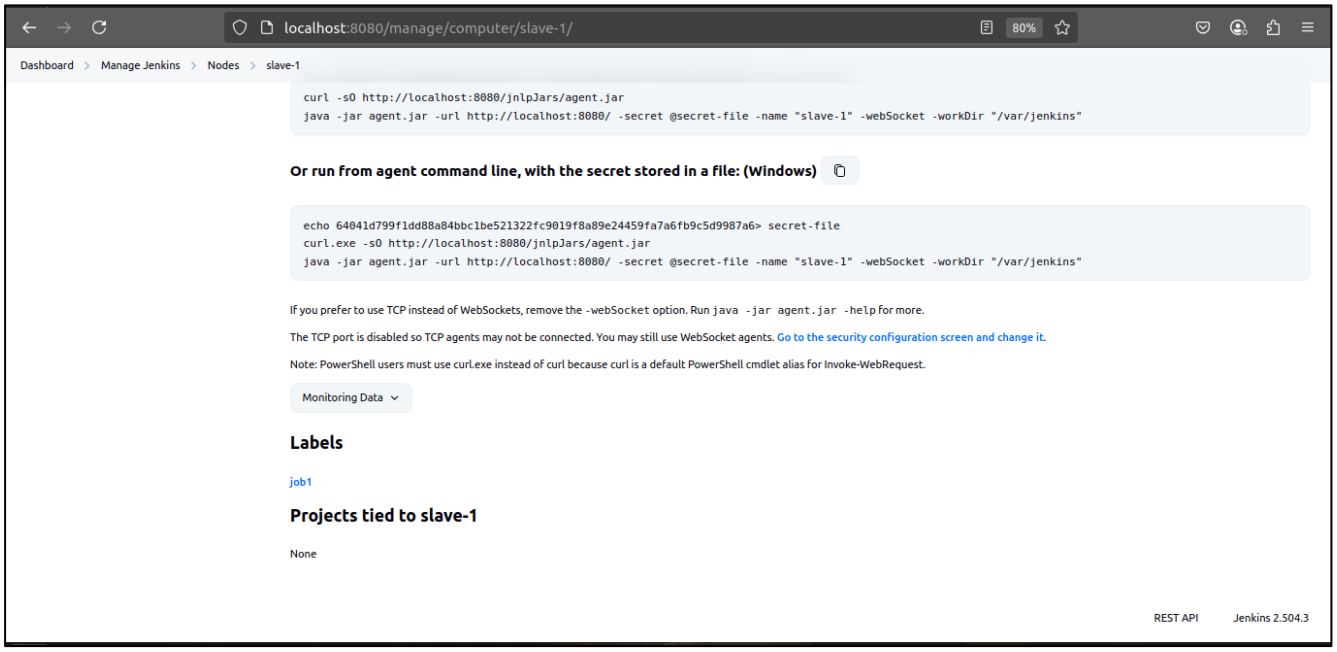
8. Enter Username, Password, ID, and Description.

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.






```
user@user-H81M-S:~$ java -jar agent.jar -url http://localhost:8080/ -secret 64041d799f1dd88a84bbc1be521322fc9019f8a89e24459fa7a6fb9c5d9987a6 -name "sl
ave-1" -workDir "/var/jenkins"
Sep 17, 2025 12:29:25 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /var/jenkins/remoting as a remoting work directory
Sep 17, 2025 12:29:25 PM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /var/jenkins/remoting
Sep 17, 2025 12:29:25 PM hudson.remoting.Launcher createEngine
INFO: Setting up agent: slave-1
Sep 17, 2025 12:29:25 PM hudson.remoting.Engine startEngine
INFO: Using Remoting version: 3301.v4363ddcca_4e7
Sep 17, 2025 12:29:25 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /var/jenkins/remoting as a remoting work directory
Sep 17, 2025 12:29:25 PM hudson.remoting.Launcher$CuiListener status
INFO: Locating server among [http://localhost:8080/]
Sep 17, 2025 12:29:25 PM org.jenkinsci.remoting.engine.JnlpAgentEndpointResolver resolve
INFO: Remoting server accepts the following protocols: [JNLP4-connect, Ping]
Sep 17, 2025 12:29:25 PM hudson.remoting.Launcher$CuiListener status
INFO: Agent discovery successful
  Agent address: localhost
  Agent port:    50000
  Identity:      c1:3f:85:6f:72:69:d5:68:f4:72:35:24:9b:e6:02:a9
Sep 17, 2025 12:29:25 PM hudson.remoting.Launcher$CuiListener status
INFO: Handshaking
Sep 17, 2025 12:29:25 PM hudson.remoting.Launcher$CuiListener status
INFO: Connecting to localhost:50000
Sep 17, 2025 12:29:25 PM hudson.remoting.Launcher$CuiListener status
INFO: Server reports protocol JNLP4-connect-proxy not supported, skipping
Sep 17, 2025 12:29:25 PM hudson.remoting.Launcher$CuiListener status
INFO: Trying protocol: JNLP4-connect
Sep 17, 2025 12:29:25 PM org.jenkinsci.remoting.protocol.impl.BIONetworkLayer$Reader run
INFO: Waiting for ProtocolStack to start.
Sep 17, 2025 12:29:26 PM hudson.remoting.Launcher$CuiListener status
INFO: Remote identity confirmed: c1:3f:85:6f:72:69:d5:68:f4:72:35:24:9b:e6:02:a9
Sep 17, 2025 12:29:26 PM hudson.remoting.Launcher$CuiListener status
INFO: Connected
```

Creating a Freestyle Project and Running on the Slave Machine:

1.Click "Save," which will redirect to the job's view page.

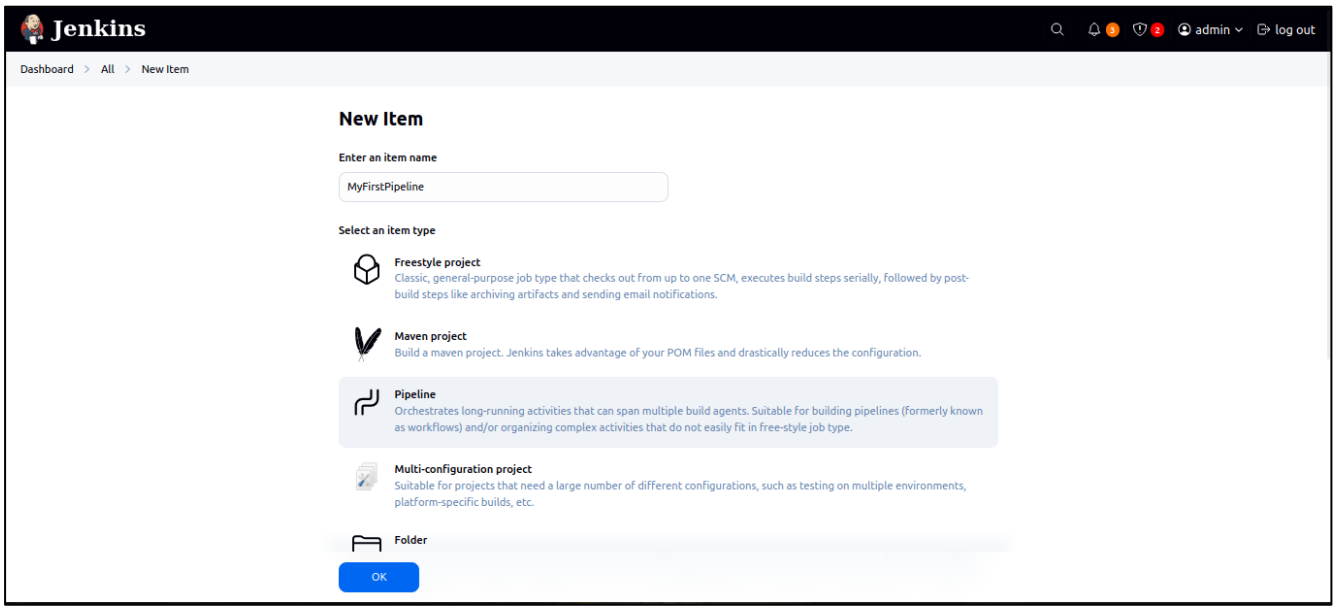
2.On the left pane, click the "Build Now" button to execute your pipeline.

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:

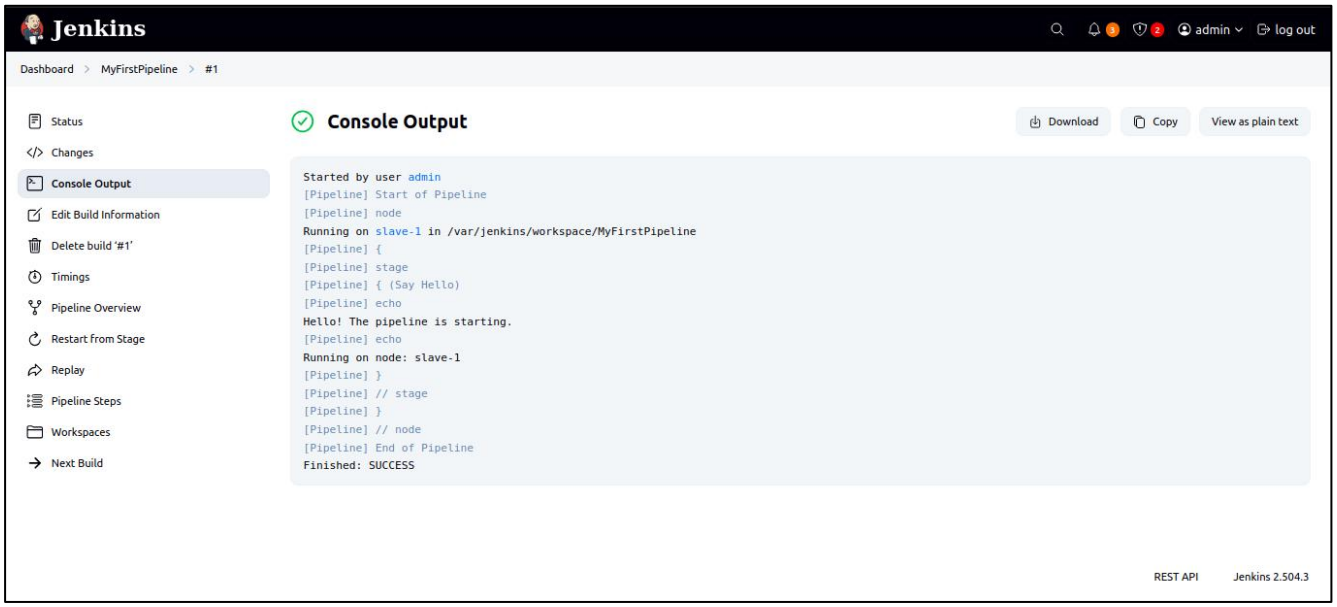
1. Click "New Item" in the top left corner on the dashboard.
2. Enter the name of your project in the "Enter an item name" field, select the "Pipeline" project, and click "OK."



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.