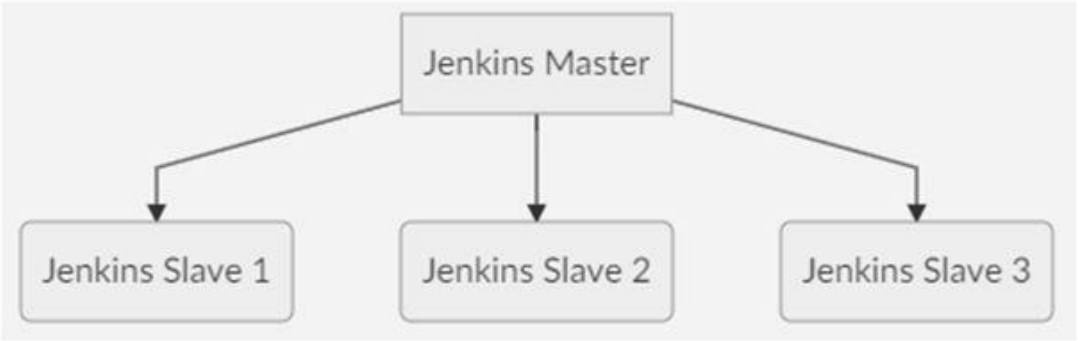
Experiment 06

Aim : To understand Jenkins Master – Slave architecture and scale your Jenkins standalone implementation by implementing slave nodes .

Theory : Master/slave is a model of asymmetric communication or control where one device or process controls one or more other devices or processes and serves as their communication hub. In some systems, a master is selected from a group of eligible devices, with the other devices acting in the role of slaves.

An agent is a computer that is set up to offload build projects from the master and once setup this distribution of tasks is fairly automatic.

Jenkins Master and Slave Architecture

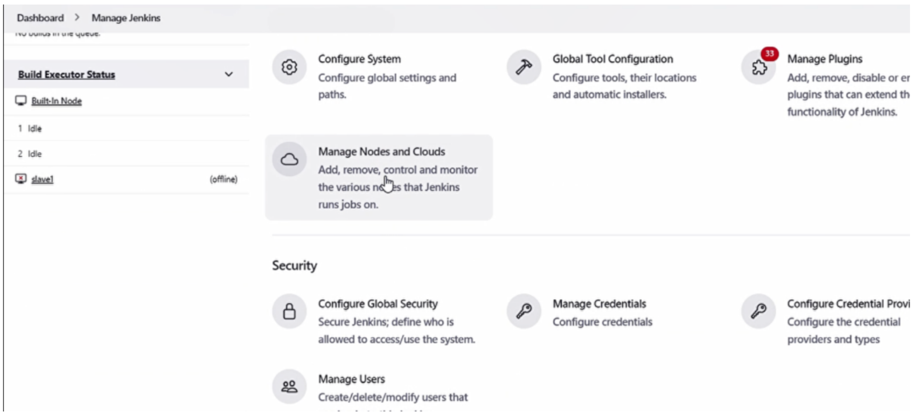


- The Jenkins master acts to schedule the jobs, assign slaves, and send builds to slaves to execute the jobs.
- It will also monitor the slave state (offline or online) and get back the build result responses from slaves and the display build results on the console output. The workload of building jobs is delegated to multiple **slaves**.

Master-slave replication enables data from one database server (the master) to be replicated to one or more other database servers (the slaves). The master logs the updates, which then ripple through to the slaves.

Procedure :

1. Log in to your Jenkins on the dashboard click manage Jenkins and in that section go to Manage nodes and clouds .



Dashboard > Manage Jenkins

No builds in the queue.

Build Executor Status

Built-In Node

1 Idle

2 Idle

slave1 (offline)

Configure System
Configure global settings and paths.

Global Tool Configuration
Configure tools, their locations and automatic installers.

Manage Plugins
Add, remove, disable or en plugins that can extend th functionality of Jenkins.

Manage Nodes and Clouds
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Security

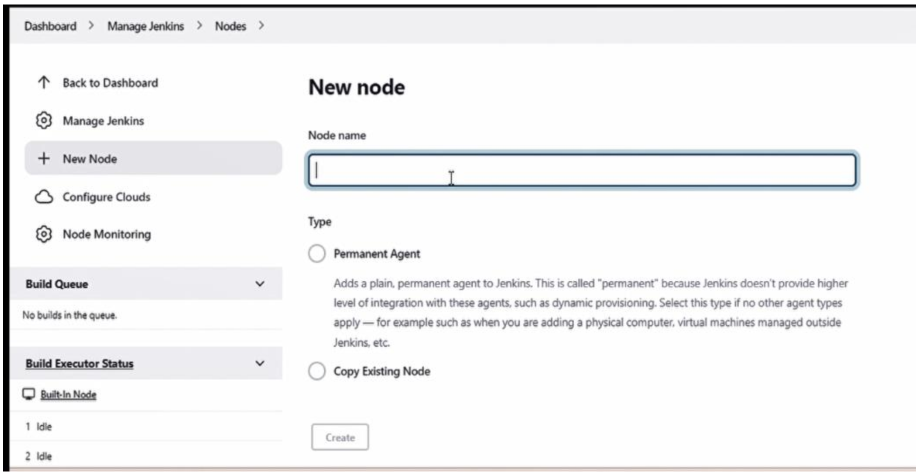
Configure Global Security
Secure Jenkins; define who is allowed to access/use the system.

Manage Credentials
Configure credentials

Configure Credential Provi
Configure the credential providers and types

Manage Users
Create/delete/modify users that

2. After that click on new node and create a new node . and select permanent agent .



Dashboard > Manage Jenkins > Nodes >

Back to Dashboard

Manage Jenkins

New Node

Configure Clouds

Node Monitoring

Build Queue

No builds in the queue.

Build Executor Status

Built-In Node

1 Idle

2 Idle

New node

Node name

Type

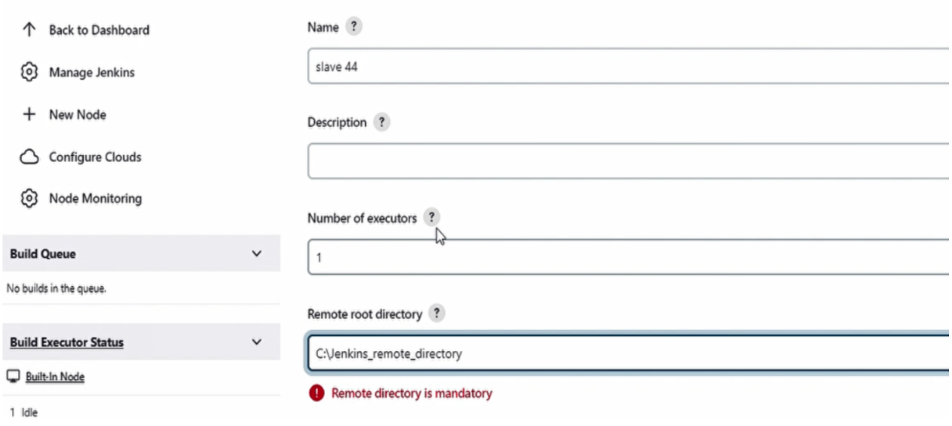
○ Permanent Agent

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.

○ Copy Existing Node

Create

3. Now in remote root directory , add this path wherever your Jenkins is installed .



Back to Dashboard

Manage Jenkins

New Node

Configure Clouds

Node Monitoring

Build Queue

No builds in the queue.

Build Executor Status

Built-In Node

1 Idle

Name ?

slave 44

Description ?

Number of executors ?

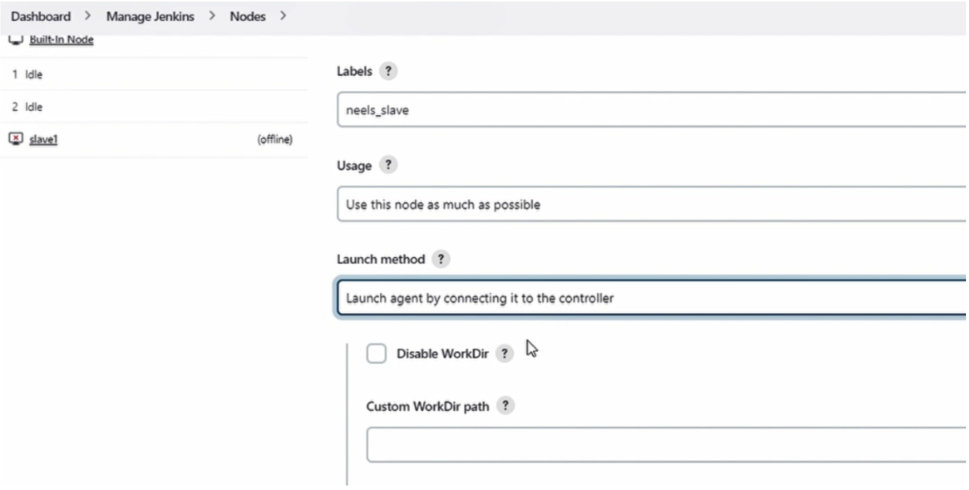
1

Remote root directory ?

C:\Jenkins_remote_directory

Remote directory is mandatory

4. Also add a label name and keep the rest unchanged and then apply and save it .



Labels ?
neels_slave

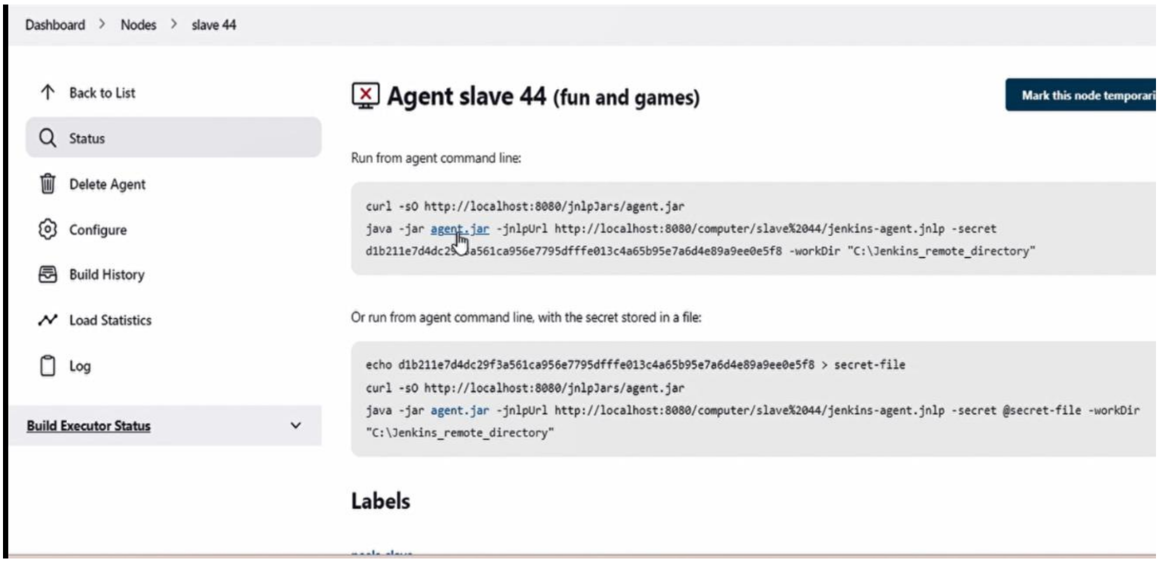
Usage ?
Use this node as much as possible

Launch method ?
Launch agent by connecting it to the controller

□ Disable WorkDir ?

Custom WorkDir path ?

5. Now go the main dashboard and click the node we names (slave) and in status section download the jar file in the dialogue code box .



Agent slave 44 (fun and games)

Run from agent command line:

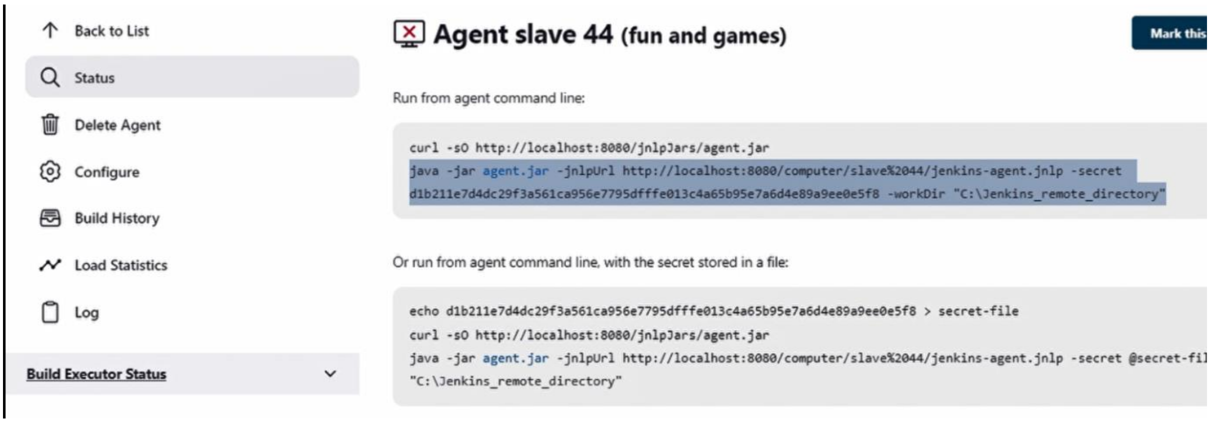
```
curl -sO http://localhost:8080/jnlpJars/agent.jar
java -jar agent.jar -jnlpUrl http://localhost:8080/computer/slave%2044/jenkins-agent.jnlp -secret
d1b211e7d4dc29f3a561ca956e7795dfffe013c4a65b95e7a6d4e89a9ee0e5f8 -workDir "C:\Jenkins_remote_directory"
```

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

```
echo d1b211e7d4dc29f3a561ca956e7795dfffe013c4a65b95e7a6d4e89a9ee0e5f8 > secret-file
curl -sO http://localhost:8080/jnlpJars/agent.jar
java -jar agent.jar -jnlpUrl http://localhost:8080/computer/slave%2044/jenkins-agent.jnlp -secret @secret-file -workDir
"C:\Jenkins_remote_directory"
```

Labels

6. Then also the copy this link in first dialogue box ;



Agent slave 44 (fun and games)

Run from agent command line:

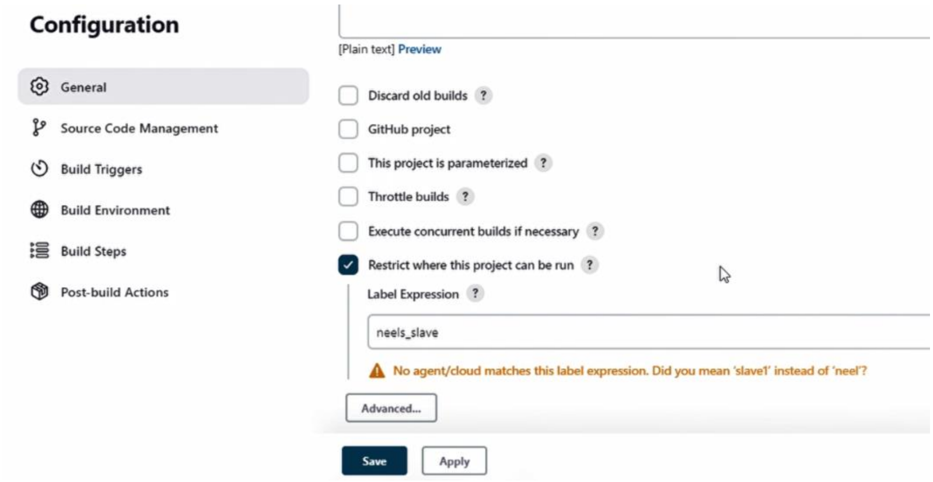
```
curl -sO http://localhost:8080/jnlpJars/agent.jar
java -jar agent.jar -jnlpUrl http://localhost:8080/computer/slave%2044/jenkins-agent.jnlp -secret
d1b211e7d4dc29f3a561ca956e7795dfffe013c4a65b95e7a6d4e89a9ee0e5f8 -workDir "C:\Jenkins_remote_directory"
```

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

```
echo d1b211e7d4dc29f3a561ca956e7795dfffe013c4a65b95e7a6d4e89a9ee0e5f8 > secret-file
curl -sO http://localhost:8080/jnlpJars/agent.jar
java -jar agent.jar -jnlpUrl http://localhost:8080/computer/slave%2044/jenkins-agent.jnlp -secret @secret-fil
"C:\Jenkins_remote_directory"
```

7. After that go to the main jenkins dashboard and refresh the page and select a simple maven project (non-pipelined) , like first maven code we discussed .

8. Now select the option ; Restrict where this project can run .. and in the label input add your project nodes slave node .



Configuration

[Plain text] Preview

□ Discard old builds ?

□ GitHub project

□ This project is parameterized ?

□ Throttle builds ?

□ Execute concurrent builds if necessary ?

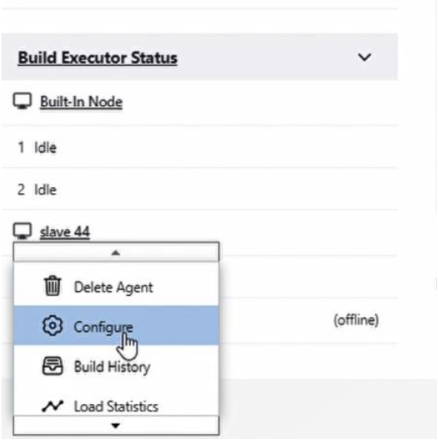
☑ Restrict where this project can be run ?

Label Expression ?

neels_slave

⚠ No agent/cloud matches this label expression. Did you mean 'slave1' instead of 'neel'?

9. Click on apply and save .   then go the dashboard , refresh the page and access the slave on cloud .



Build Executor Status

Built-In Node

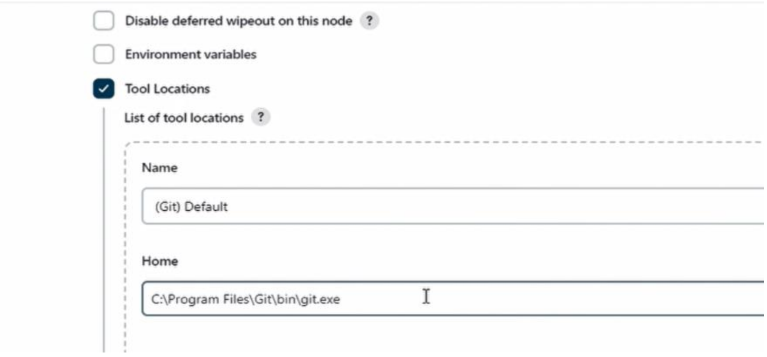
1 Idle

2 Idle

slave 44

(offline)

10. Now without changing other vital tools go to the Add tools locations and select Git(default) option , and enter the link of installed Git on your PC . Also add path to java Apply and Save .



□ Disable deferred wipeout on this node ?

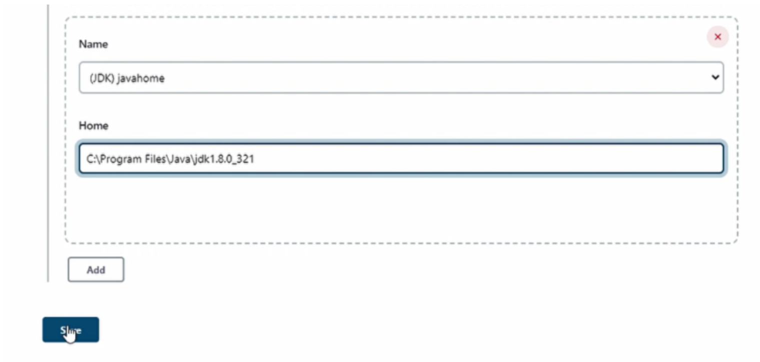
□ Environment variables

☑ Tool Locations

List of tool locations ?

Name: (Git) Default

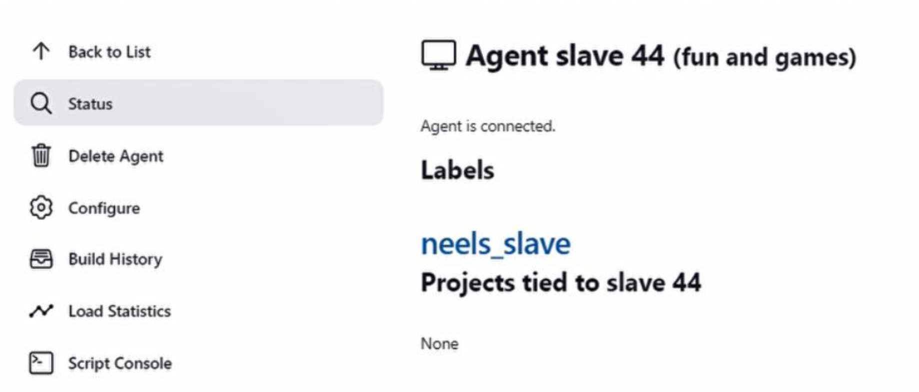
Home: C:\Program Files\Git\bin\git.exe



Name: (JDK) javahome

Home: C:\Program Files\Java\jdk1.8.0_321

11. This is the new status change in slave node :



Agent slave 44 (fun and games)

Agent is connected.

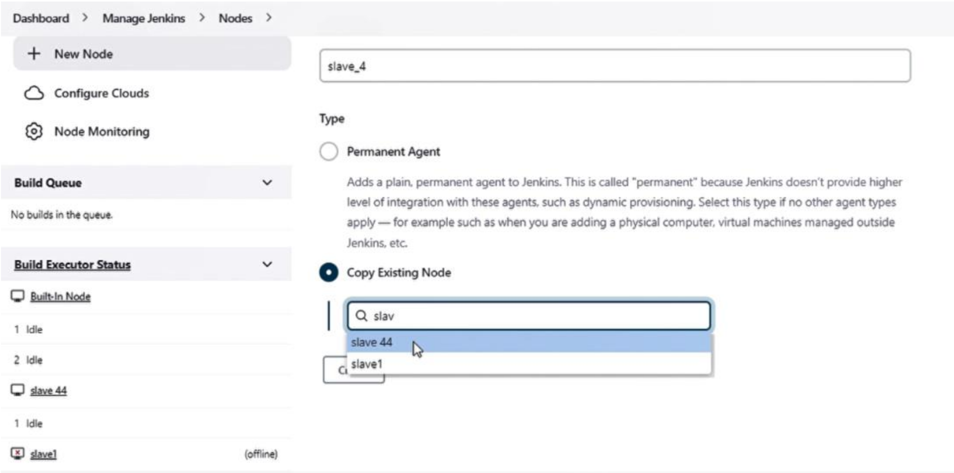
Labels

neels_slave

Projects tied to slave 44

None

12. Now again go the the dashboard , refresh the page go in manage Jenkins and click nodes .
13. You can also create such slaves by migrating previously created data of a slave :



slave_4

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.

● Copy Existing Node

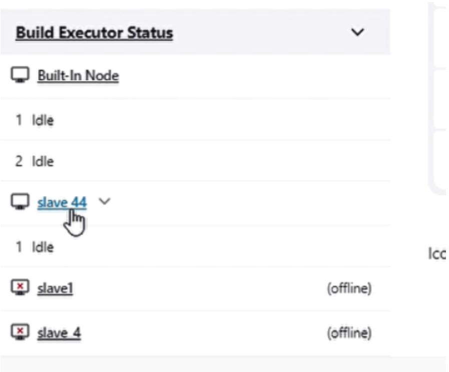
slav

slave 44
slave1

No builds in the queue.

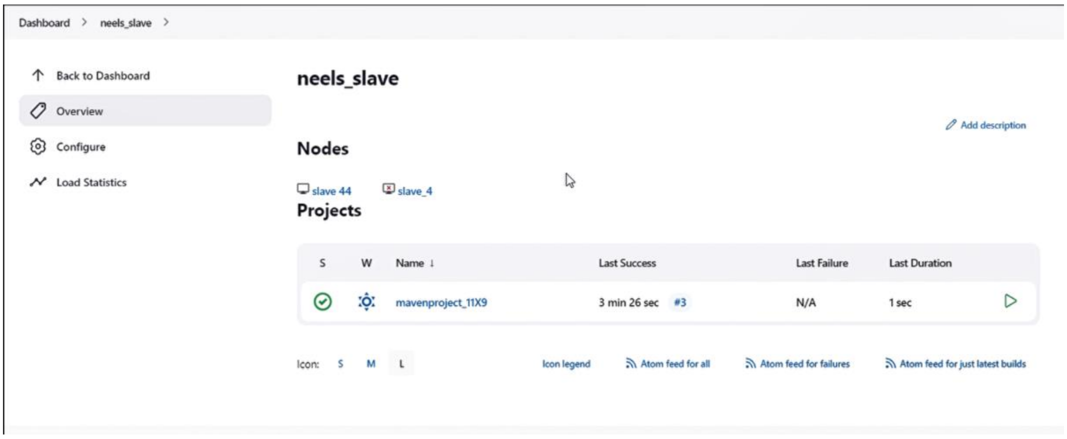
Built-In Node
1 Idle
2 Idle
slave 44
1 Idle
slave1 (offline)

14. So access nodes from the build execution column



Built-In Node
1 Idle
2 Idle
slave 44
1 Idle
slave1 (offline)
slave 4 (offline)

15. On success of slave nodes , here’s how the overview of slave node (called by label)



neels_slave

Nodes

slave 44 slave_4

Projects

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☼	mavenproject_11X9	3 min 26 sec #3	N/A	1 sec

appears and the project we scaled using the slave node .

Conclusion : Hence we understood the master-slave architecture in Jenkins and scaled a Jenkins project successfully using slave nodes – master architecture .