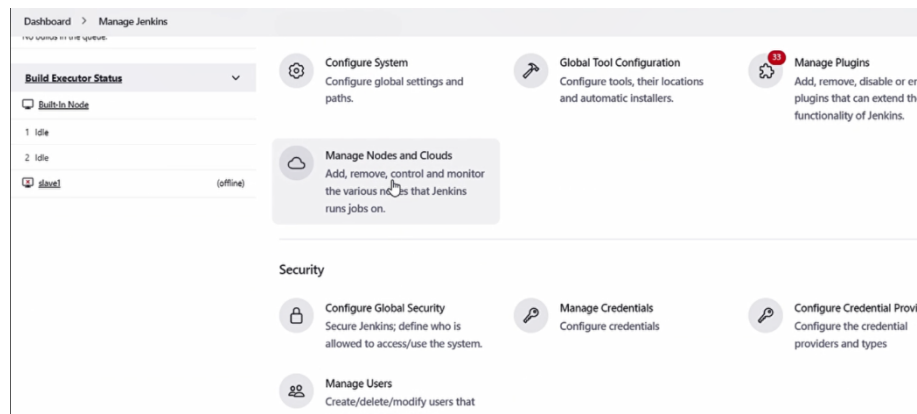
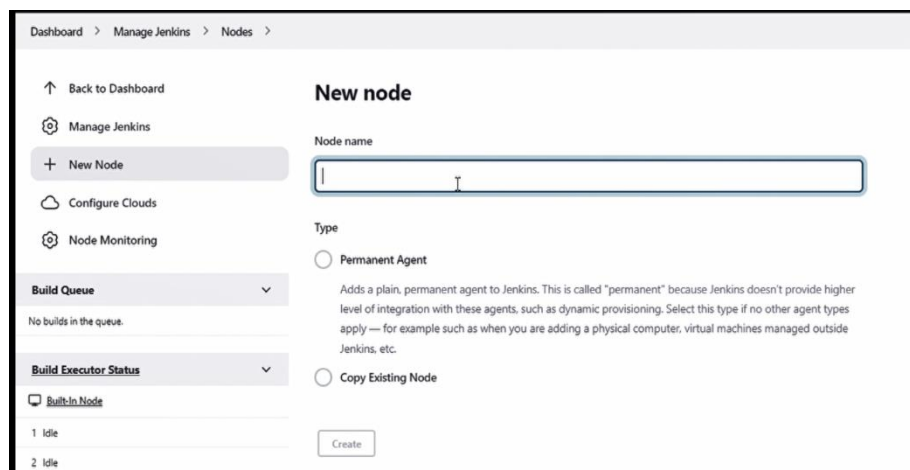


Procedure :

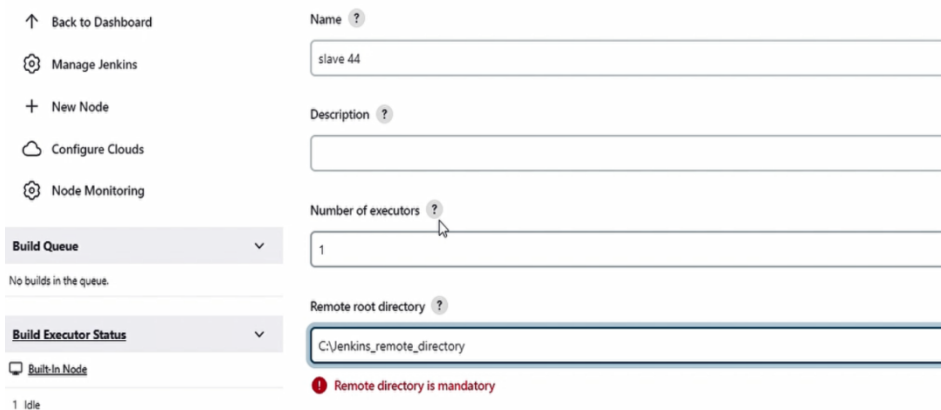
1. Log in to your Jenkins on the dashboard click manage Jenkins and in that section go to Manage nodes and clouds .



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



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



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

Dashboard > Manage Jenkins > Nodes >

Built-In Node

1 Idle

2 Idle

slave1 (offline)

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 .

Dashboard > Nodes > slave 44

Back to List

Status

Delete Agent

Configure

Build History

Load Statistics

Log

Build Executor Status

Agent slave 44 (fun and games) Mark this node temporary

Run from agent command line:

```
curl -sO http://localhost:8080/jnlpJars/agent.jar
java -jar agent.jar -jnlpUrl http://localhost:8080/computer/slaveX2044/jenkins-agent.jnlp -secret
d1b211e7d4dc29f3a561ca956e7795dfff013c4a65b95e7a6d4e89a9ee0e5f8 -workDir "C:\jenkins_remote_directory"
```

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

```
echo d1b211e7d4dc29f3a561ca956e7795dfff013c4a65b95e7a6d4e89a9ee0e5f8 > secret-file
curl -sO http://localhost:8080/jnlpJars/agent.jar
java -jar agent.jar -jnlpUrl http://localhost:8080/computer/slaveX2044/jenkins-agent.jnlp -secret @secret-file -workDir
"C:\jenkins_remote_directory"
```

Labels

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

Back to List

Status

Delete Agent

Configure

Build History

Load Statistics

Log

Build Executor Status

Agent slave 44 (fun and games) Mark this node temporary

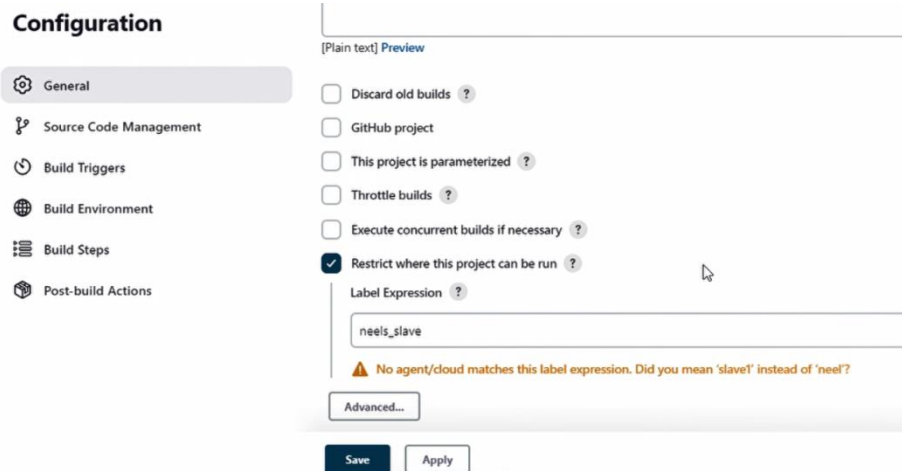
Run from agent command line:

```
curl -sO http://localhost:8080/jnlpJars/agent.jar
java -jar agent.jar -jnlpUrl http://localhost:8080/computer/slaveX2044/jenkins-agent.jnlp -secret
d1b211e7d4dc29f3a561ca956e7795dfff013c4a65b95e7a6d4e89a9ee0e5f8 -workDir "C:\jenkins_remote_directory"
```

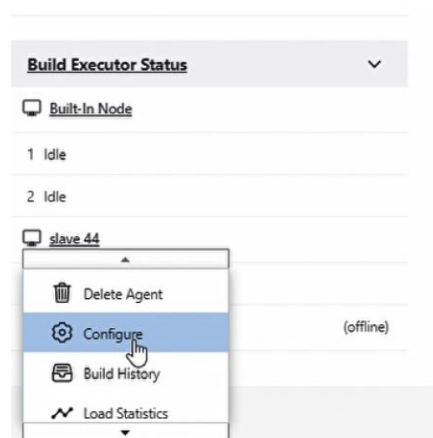
Or run from agent command line, with the secret stored in a file:

```
echo d1b211e7d4dc29f3a561ca956e7795dfff013c4a65b95e7a6d4e89a9ee0e5f8 > secret-file
curl -sO http://localhost:8080/jnlpJars/agent.jar
java -jar agent.jar -jnlpUrl http://localhost:8080/computer/slaveX2044/jenkins-agent.jnlp -secret @secret-fi
"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 .



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



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

Add

Save

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

↑ Back to List

🔍 Status

🗑️ Delete Agent

⚙️ Configure

📁 Build History

📈 Load Statistics

💻 Script Console

🖥️ **Agent slave 44 (fun and games)**

Agent is connected.

Labels

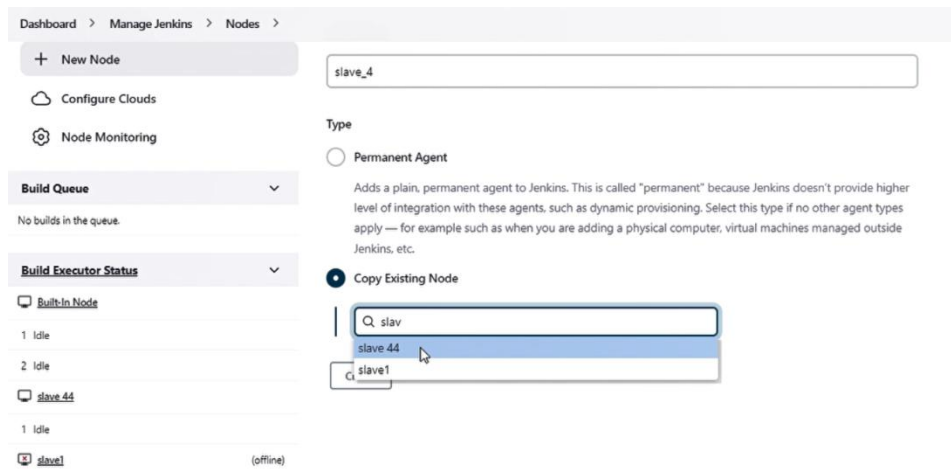
neels_slave

Projects tied to slave 44

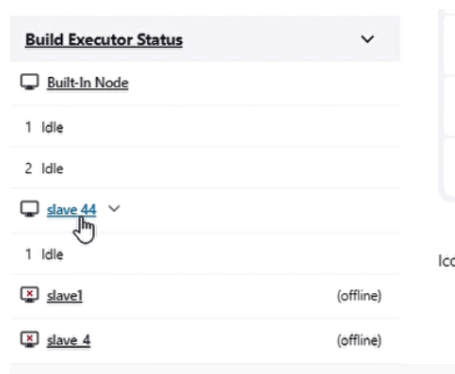
None

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 :



14. So access nodes from the build execution column



15. On success of slave nodes , here's how the overview of slave node (called by label) appears and the project we scaled using the slave node .

