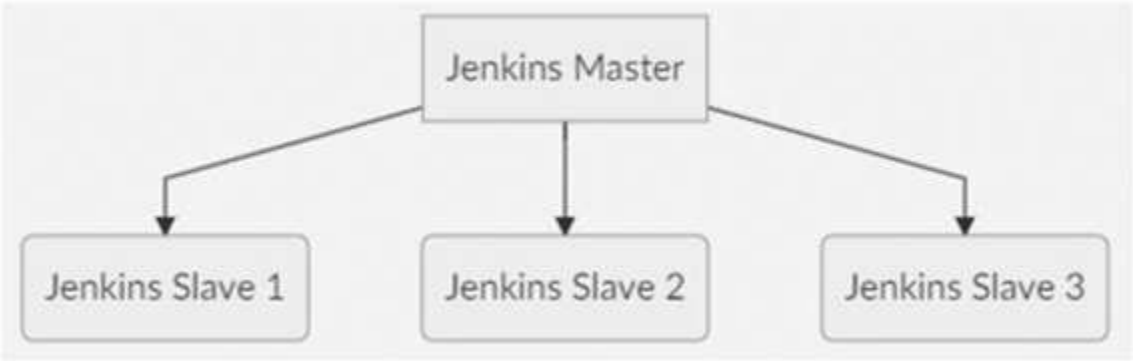
Experiment No.6
To understand Jenkins Master-Slave Architecture and scale your Jenkins standalone implementation by implementing slave nodes
Date of Performance:
Date of Submission:

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

Objective: The objective of understanding Jenkins Master-Slave architecture is to comprehend how to scale Jenkins infrastructure by implementing slave nodes, thereby distributing the workload and enhancing the performance and resilience of the Jenkins CI/CD system

Theory:

A Jenkins master comes with the basic installation of Jenkins, and in this configuration, the master handles all the tasks for your build system. You are working on multiple projects, you may run multiple jobs on each project. Some projects need to run on some nodes, and in this process, we need to configure slaves. Jenkins slaves connect to the Jenkins master using the Java Network Launch Protocol.
Jenkins Master and Slave Architecture



The Jenkins master acts to schedule the jobs, assign slaves, and send builds 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.

Steps:

1. Click on Manage Jenkins in the left corner on the Jenkins dashboard.
2. Scroll down, Click on Manage Nodes and clouds.

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

Select **New Node** and enter the name of the node in the Node Name field.

4. Select Permanent Agent and click the OK button. Initially, you will get only one option, "Permanent Agent." Once you have one or more slaves you will get the "Copy Existing Node" option.

Select **Permanent Agent** and click the **OK** button. Initially, you will get only one option, "Permanent Agent." Once you have one or more slaves you will get the "Copy Existing Node" option.

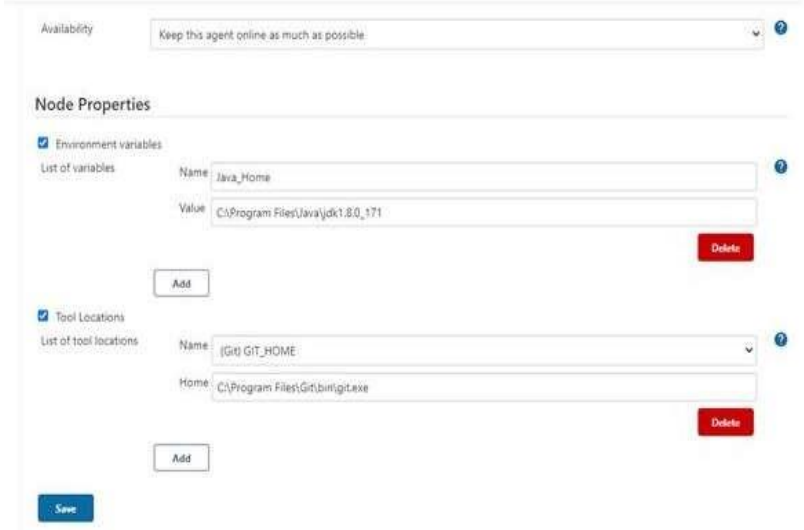
In the above screen shot, Parallel_Agent_01 was Created and currently it is in offline mode.

5. Click on configure, Provide the details.
   1. Name -Parallel_Agent_01.
   2. Number of executors- 5.
   3. Remote root directory-We have to provide a Jenkins path.
   4. Labels-Parallel_Agent.
   5. Launch method-Launch agent by connecting it to the master.

6. Node Properties Tab:
   - Check Environment variables

     Provide the Java path
   - Check Tool Locations

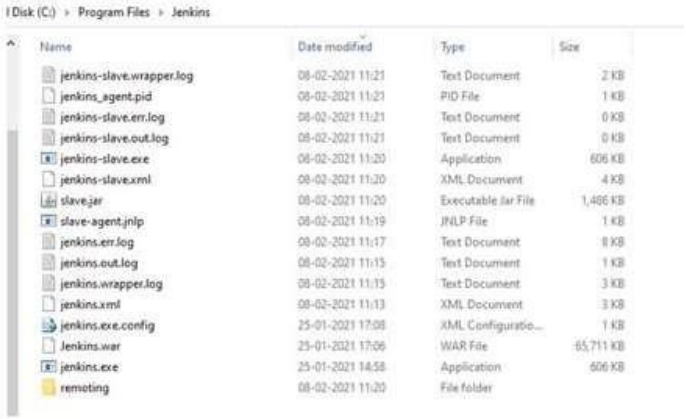
     Provide the Git path and click on save button.



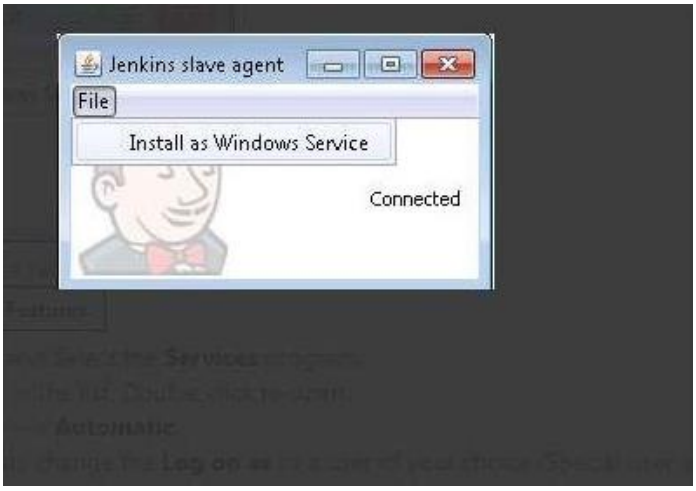
7. Click on Go to the security configuration screen and change it. It will redirect to Configure Global Security → Agents > click on Fixed radio button port: 49187 and click on save Button. Go back to Nodes settings.

8. We can see the screen,
   1. Click on Launch button, it will download the launch agent in your system.

   2. Jenkins-slave.exe file should copy in the Jenkins folder which you installed in your system.
   3. Double Click on jenkins - slave.exe.



4. Run the launch agent, click on the run button and it will show connected.

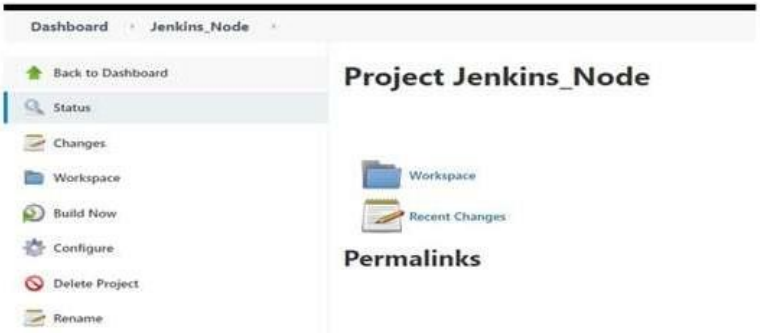


5. In the screenshot below, we can see the connected popup, click on the file menu, select the install as service and click yes button. Once it is done, refresh the page.

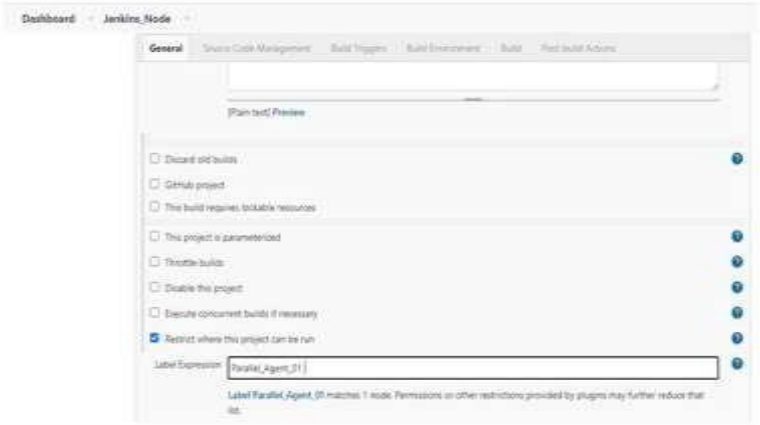
9. we can see the Build executors. One is master and other is Parallel_agent_01

In Master node, we can see the number of executors as 2.

- In Parallel_agent_01, we can see the number of executors as 5.
- Go to build job -> configure.



- In the General tab, check on Restrict where this project can be run.
- In Label Expression, we have to select the node name where we need to execute the build job.



- We can create more nodes as well.

Conclusion:

Q1. How does Jenkins communicate between master and slave?
Jenkins communicates between its master and slave nodes primarily through a combination of protocols, mainly SSH and Java Network Launch Protocol (JNLP). When a job is triggered on the Jenkins master, it connects to the slave using SSH, which allows it to securely execute commands and transfer files over the network. For Jenkins agents launched using the JNLP protocol, the master initiates a connection to the agent, enabling bidirectional communication for job execution and data transfer. These communication methods facilitate the distribution of workload from the master to the slave nodes, allowing for parallel and efficient execution of Jenkins jobs across the connected environment.

Q2. How many slaves can be connected to Jenkins master?
The number of slaves that can be connected to a Jenkins master largely depends on the hardware resources of the Jenkins master and the workload of the jobs being executed. Jenkins itself doesn't have a strict limit on the number of slaves that can be connected, as it's more about the capacity of the master to manage and coordinate the workload. However, practical limitations such as network bandwidth, disk I/O, and CPU/RAM capacity can influence this. Some installations have hundreds or even thousands of slaves connected to a single Jenkins master, especially in large-scale continuous integration (CI) and continuous delivery (CD) environments. Administrators often need to carefully monitor performance metrics to ensure optimal functioning as the number of slaves increases.