

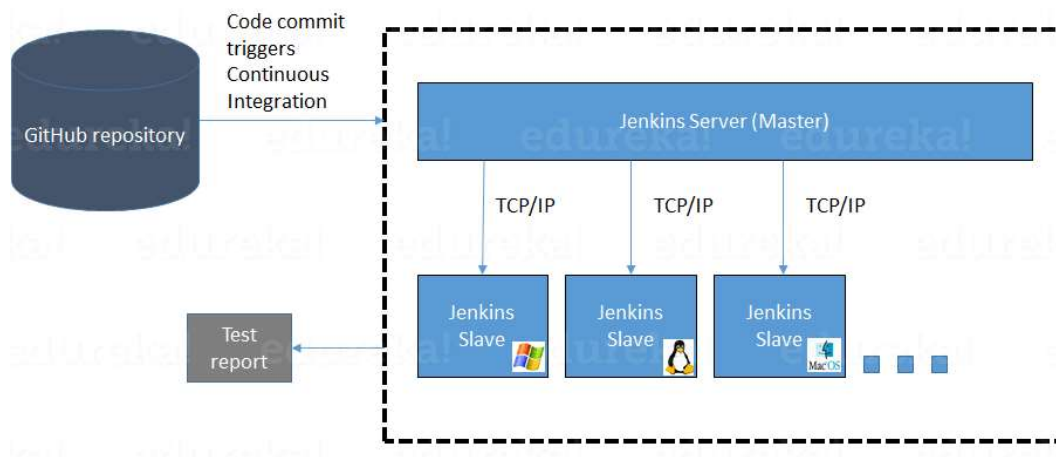
Experiment No : 06

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

Theory :

Jenkins is one of the most important tools in DevOps. Jenkins is used in the Continuous Integration stage of DevOps. In this blog, I am going to talk about the Jenkins Master and Slave architecture.

Jenkins uses a Master-Slave architecture to manage distributed builds. In this architecture, Master and Slave communicate through TCP/IP protocol.

**Jenkins Master**

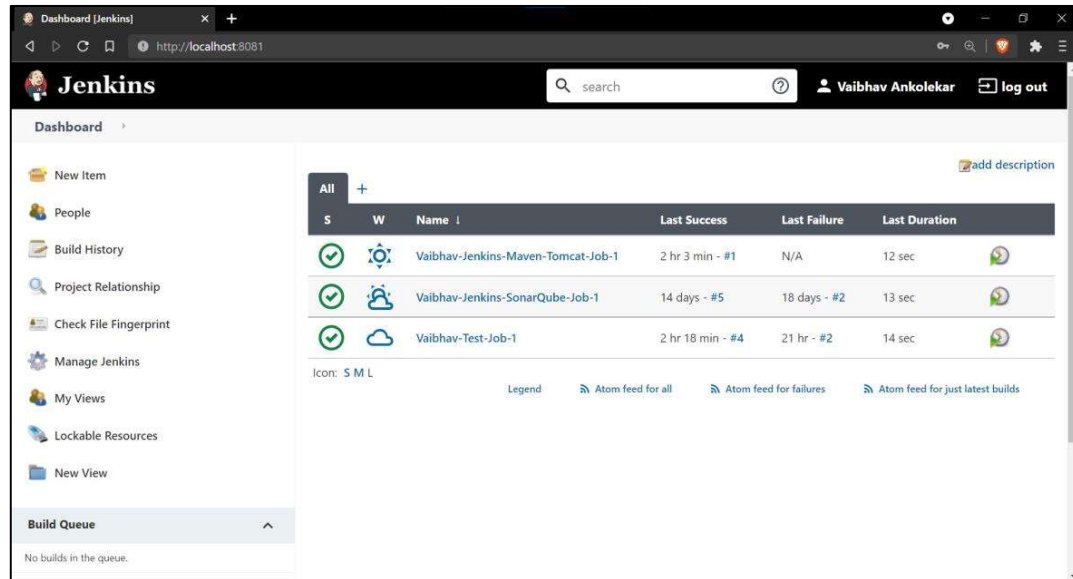
- Scheduling build jobs.
- Dispatching builds to the slaves for the actual execution.
- Monitor the slaves (possibly taking them online and offline as required).
- Recording and presenting the build results.
- A Master instance of Jenkins can also execute build jobs directly.

Jenkins Slave

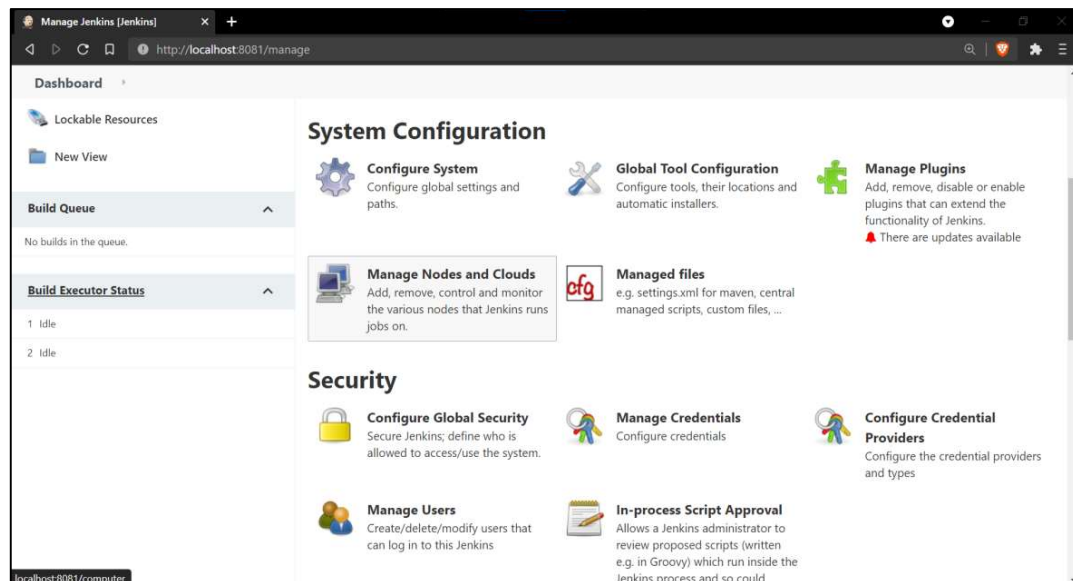
- It hears requests from the Jenkins Master instance.
- Slaves can run on a variety of operating systems.
- The job of a Slave is to do as they are told to, which involves executing build jobs dispatched by the Master.
- You can configure a project to always run on a particular Slave machine or a particular type of Slave machine, or simply let Jenkins pick the next available Slave.

Procedure :

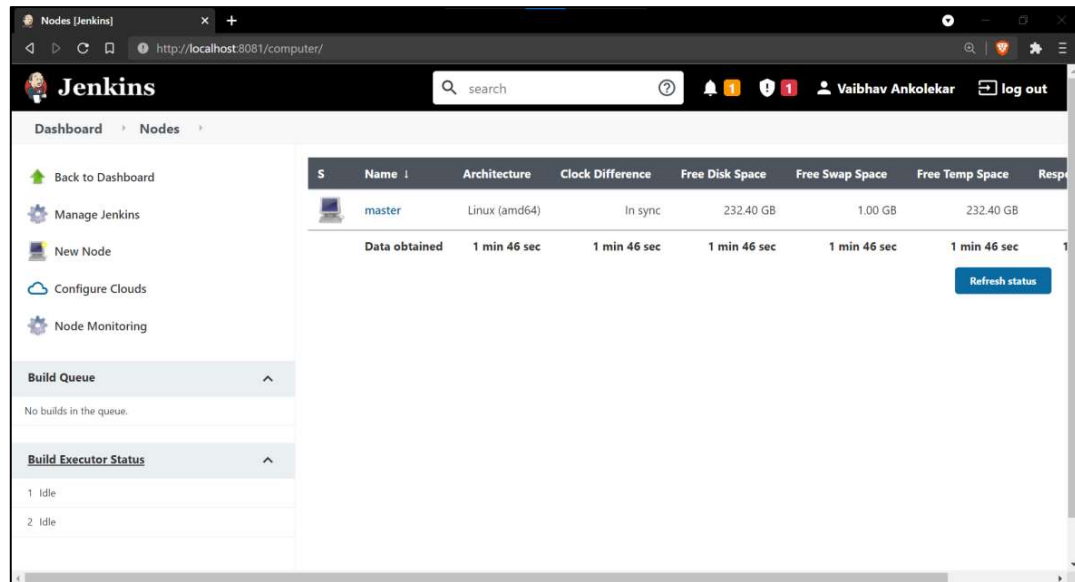
1) Open Jenkins Dashboard and click on Manage Jenkins.



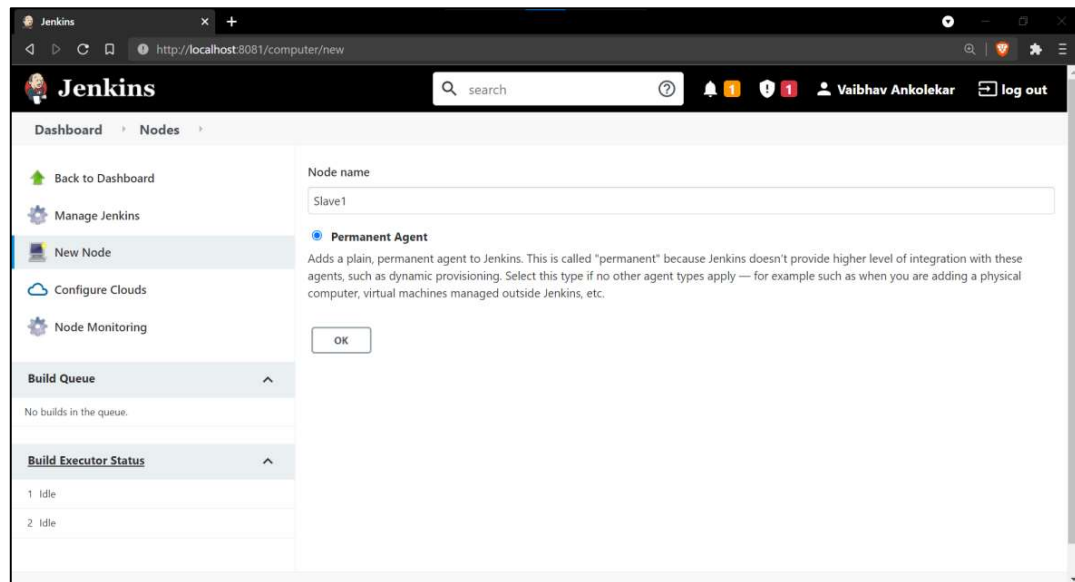
2) Click on Manage Nodes and Clouds



3) Click on New Node.



4) Give a name for your Node, select Permanent Agent and click OK.



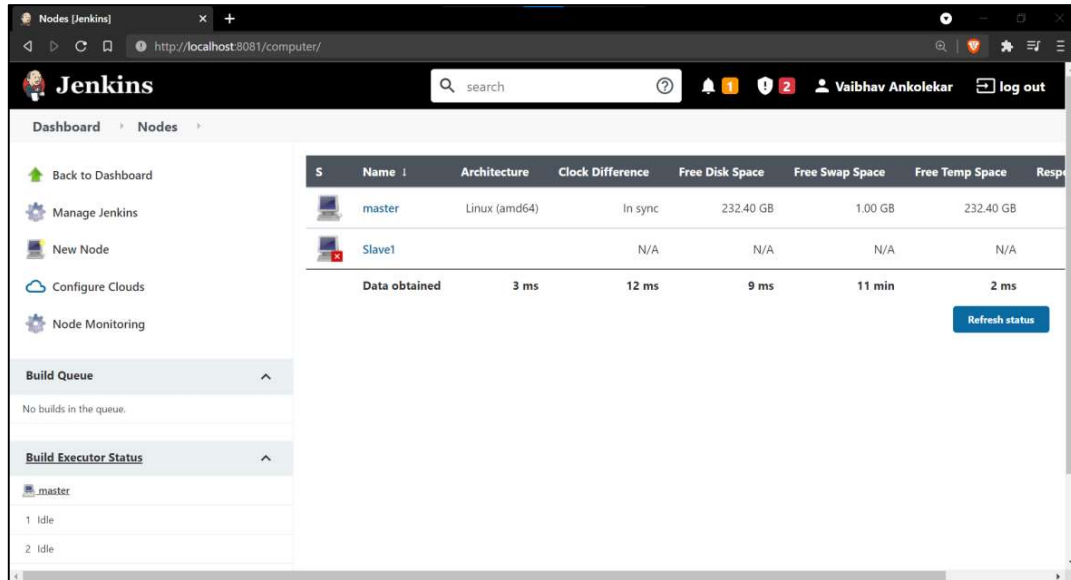
5) Now configure the new node as per your choice.

The screenshot shows the Jenkins 'Slave1 Configuration' page. The left sidebar contains links: Back to List, Status, Delete Agent, Configure (highlighted), Build History, Load Statistics, Script Console, Log, System Information, and Disconnect. The main content area has the following fields: Name (Slave1), Description, Number of executors (1), Remote root directory (/home/vaibhav-ank/jenkins-slave1), Labels, and Usage. A 'Save' button is at the bottom right.

6) In Launch method choose: *Launch agent by connecting it to master*. Fill the next details and click Save.

This screenshot shows the 'Launch method' section of the Jenkins 'Slave1 Configuration' page. The 'Launch method' dropdown is set to 'Launch agent by connecting it to the master'. Below it, there are checkboxes for 'Disable WorkDir' and 'Use WebSocket', both of which are unchecked. The 'Custom WorkDir path' is set to '/home/vaibhav-ank/jenkins-slave1' and the 'Internal data directory' is set to 'remoting'. A 'Save' button is at the bottom right.

7) New node will be Created. Click on the name of the node.

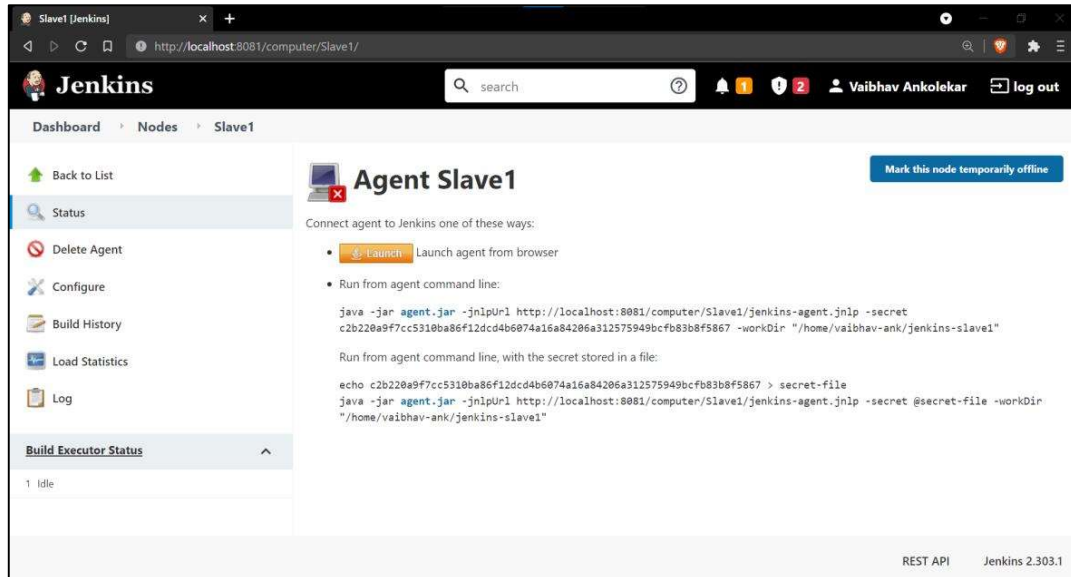


The screenshot shows the Jenkins web interface at `http://localhost:8081/computer/`. The left sidebar contains links: Back to Dashboard, Manage Jenkins, New Node, Configure Clouds, Node Monitoring, Build Queue, and Build Executor Status. The main content area displays a table of nodes:

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
1	master	Linux (amd64)	In sync	232.40 GB	1.00 GB	232.40 GB	
2	Slave1		N/A	N/A	N/A	N/A	

Below the table, there is a section for 'Data obtained' with values: 3 ms, 12 ms, 9 ms, 11 min, and 2 ms. A 'Refresh status' button is located at the bottom right of the table.

8) Click on the blue link `agent.jar` to download it.



The screenshot shows the Jenkins web interface at `http://localhost:8081/computer/Slave1/`. The left sidebar contains links: Back to List, Status, Delete Agent, Configure, Build History, Load Statistics, Log, and Build Executor Status. The main content area displays the 'Agent Slave1' page with a 'Mark this node temporarily offline' button. The page provides instructions on how to connect the agent to Jenkins:

- Launch agent from browser
- Run from agent command line:

```
java -jar agent.jar -jnlpUrl http://localhost:8081/computer/Slave1/jenkins-agent.jnlp -secret c2b220a9f7cc5310ba86f12dcd4b6074a16a84206a312575949bcbf83b8f5867 -workDir "/home/vaibhav-ank/jenkins-slave1"
```

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

```
echo c2b220a9f7cc5310ba86f12dcd4b6074a16a84206a312575949bcbf83b8f5867 > secret-file
java -jar agent.jar -jnlpUrl http://localhost:8081/computer/Slave1/jenkins-agent.jnlp -secret @secret-file -workDir "/home/vaibhav-ank/jenkins-slave1"
```

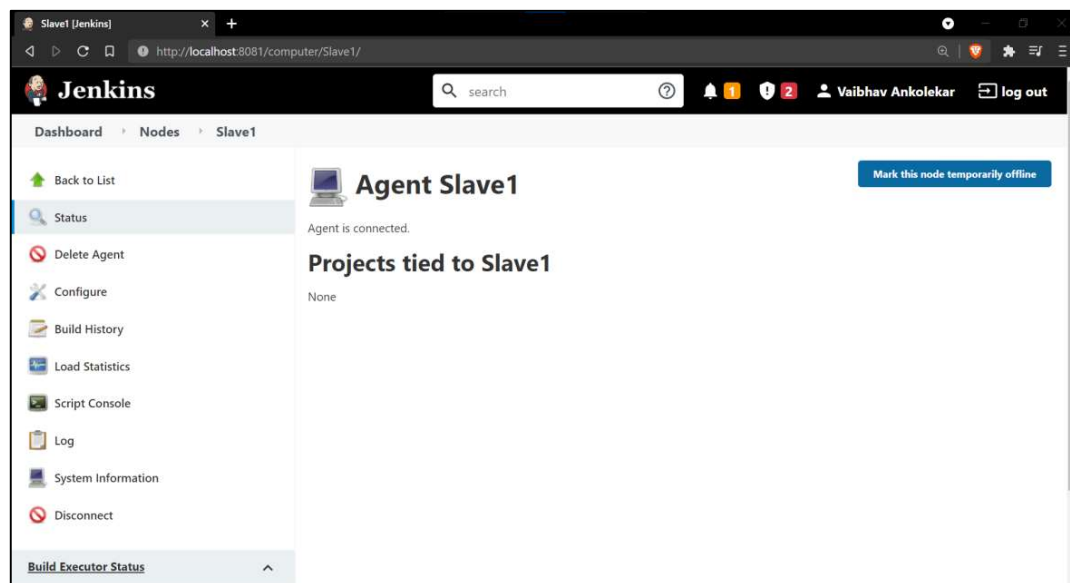
REST API Jenkins 2.303.1

9) In the new node machine, the directory which was stated while configuring it and place the agent.jar file in it.

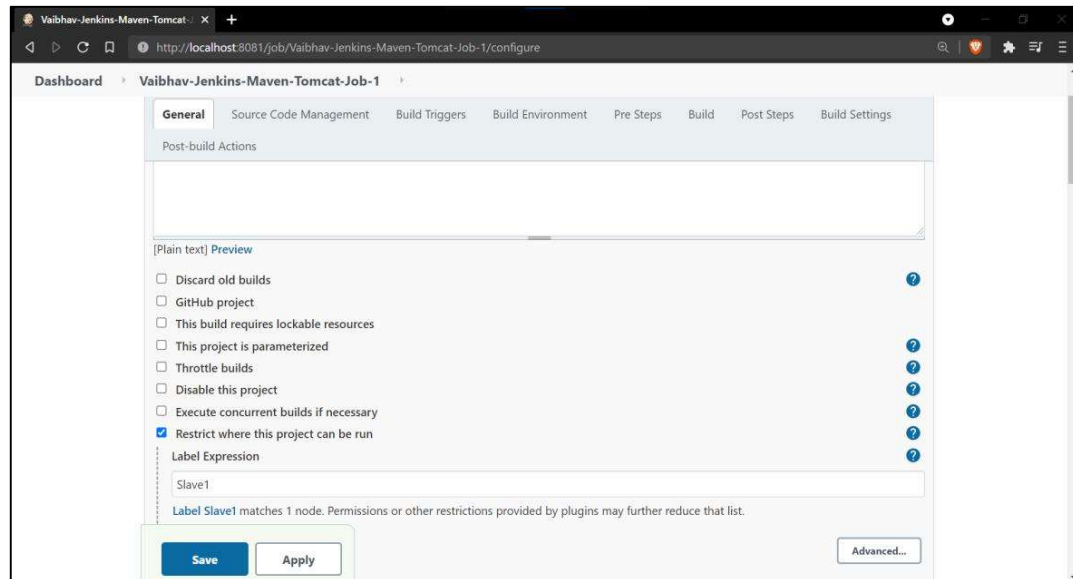
10) Now run the command given in the Node's Jenkins dashboard.

```
vaibhav-ank@LAPTOP-TUCAJV9E:~/jenkins-slave1$ ls
agent.jar jenkins-agent.jnlp
vaibhav-ank@LAPTOP-TUCAJV9E:~/jenkins-slave1$ java -jar agent.jar -jnlpUrl http://localhost:8081/computer/Slave1/jenkins-agent.jnlp -s
ecret c2b228a9f7cc5310ba86f12dcd4b6074a16a84206a312575949bcb83b8f5867 -workDir "/home/vaibhav-ank/jenkins-slave1"
Oct 09, 2021 3:17:51 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/vaibhav-ank/jenkins-slave1/remoting as a remoting work directory
Oct 09, 2021 3:17:51 PM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /home/vaibhav-ank/jenkins-slave1/remoting
Oct 09, 2021 3:17:51 PM hudson.remoting.jnlp.Main createEngine
INFO: Setting up agent: Slave1
Oct 09, 2021 3:17:51 PM hudson.remoting.jnlp.Main$CuiListener <init>
INFO: Jenkins agent is running in headless mode.
Oct 09, 2021 3:17:51 PM hudson.remoting.Engine startEngine
INFO: Using Remoting version: 4.10
Oct 09, 2021 3:17:51 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/vaibhav-ank/jenkins-slave1/remoting as a remoting work directory
Oct 09, 2021 3:17:51 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Locating server among [http://localhost:8081/]
Oct 09, 2021 3:17:51 PM org.jenkinsci.remoting.engine.JnlpAgentEndpointResolver resolve
INFO: Remoting server accepts the following protocols: [JNLP4-connect, Ping]
Oct 09, 2021 3:17:51 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Agent discovery successful
Agent address: localhost
Agent port: 42697
Identity: 98:93:16:02:fc:c2:14:45:7c:08:b0:38:1d:6a:3d:0e
Oct 09, 2021 3:17:51 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Handshaking
Oct 09, 2021 3:17:51 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Connecting to localhost:42697
Oct 09, 2021 3:17:51 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Trying protocol: JNLP4-connect
Oct 09, 2021 3:17:51 PM org.jenkinsci.remoting.protocol.impl.BIONetworkLayer$Reader run
INFO: Waiting for ProtocolStack to start.
Oct 09, 2021 3:17:51 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Remote identity confirmed: 98:93:16:02:fc:c2:14:45:7c:08:b0:38:1d:6a:3d:0e
Oct 09, 2021 3:17:53 PM hudson.remoting.jnlp.Main$CuiListener status
INFO: Connected
```

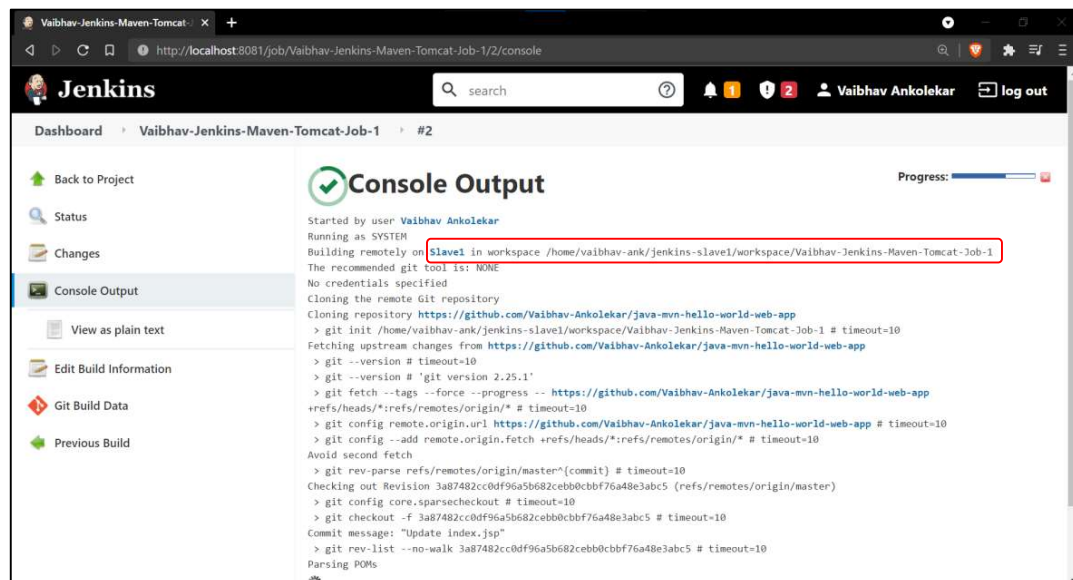
11) If you refresh the Jenkins tab, it will show the node in now connected.



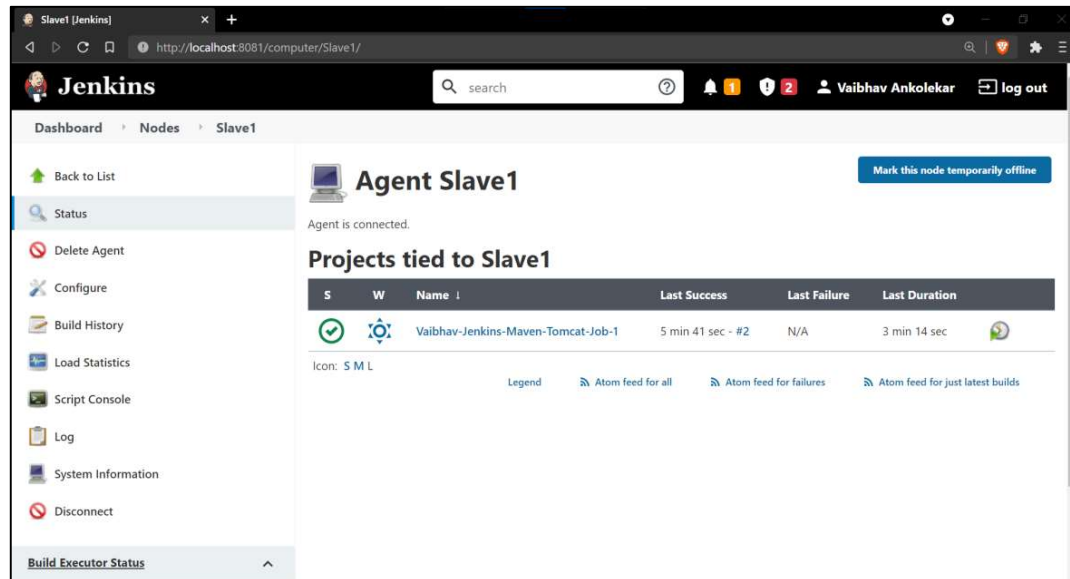
12) Configure an existing job. In General, select 'Restrict where this project can be run' and give the node's name we just created.



13) Run the Job. You can see that the job is now running in the new node's machine.



14) If you now go to the Node's dashboard, you can see a project is added to the list which was not there before.



Conclusion :

If you are working on multiple projects, you may run multiple jobs on each and every project. Some projects need to run on some particular nodes, and in this process, we need to configure slaves. Jenkins slaves connect to the Jenkins master using the Java Network Launch Protocol. It makes easier for a DevOps developer to maintain and test various application on different environment. Thus, we have successfully created a Jenkins Slave and scaled our Jenkins Architecture, which we can use to run jobs on slave rather than on master.