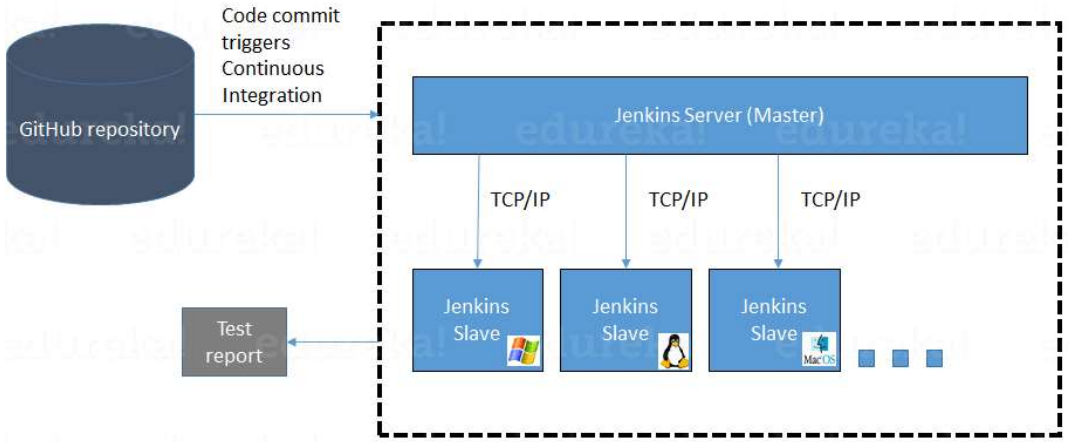
Experiment No : 06

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

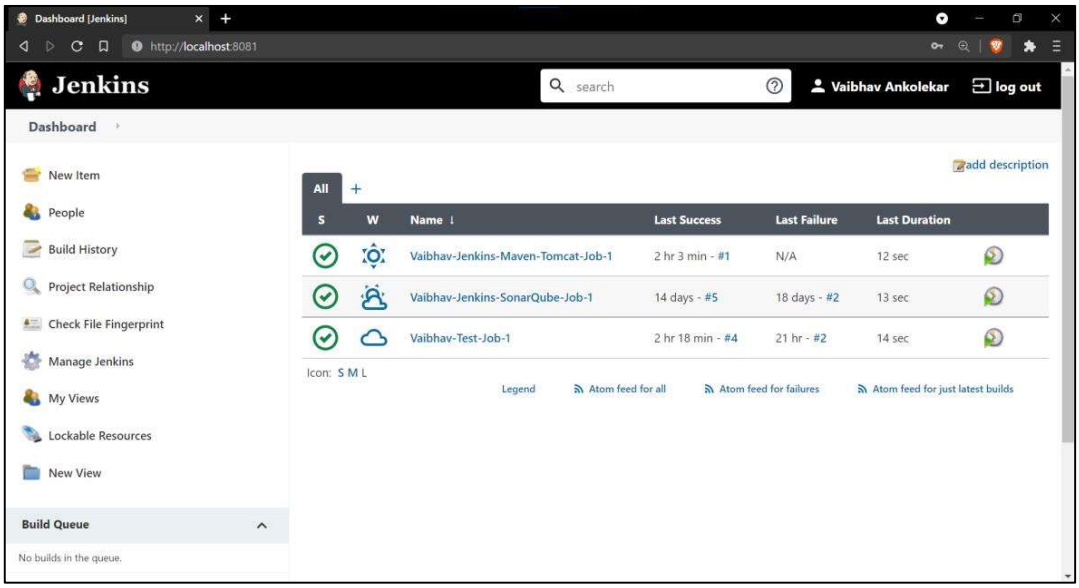
- Scheduling build jobs.
- Dispatching builds to the slaves for the actual execution.
- Monitor the slaves (possibly taking them online and offline as required).
- Recording and presenting the build results.
- A Master instance of Jenkins can also execute build jobs directly.

Jenkins Slave

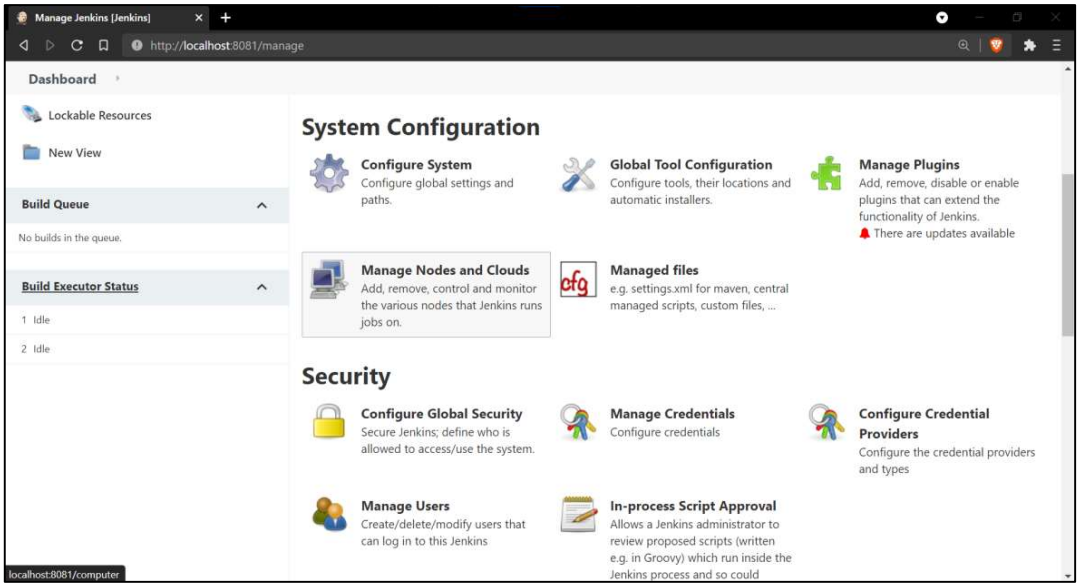
- It hears requests from the Jenkins Master instance.
- Slaves can run on a variety of operating systems.
- The job of a Slave is to do as they are told to, which involves executing build jobs dispatched by the Master.
- You can configure a project to always run on a particular Slave machine or a particular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

Procedure :

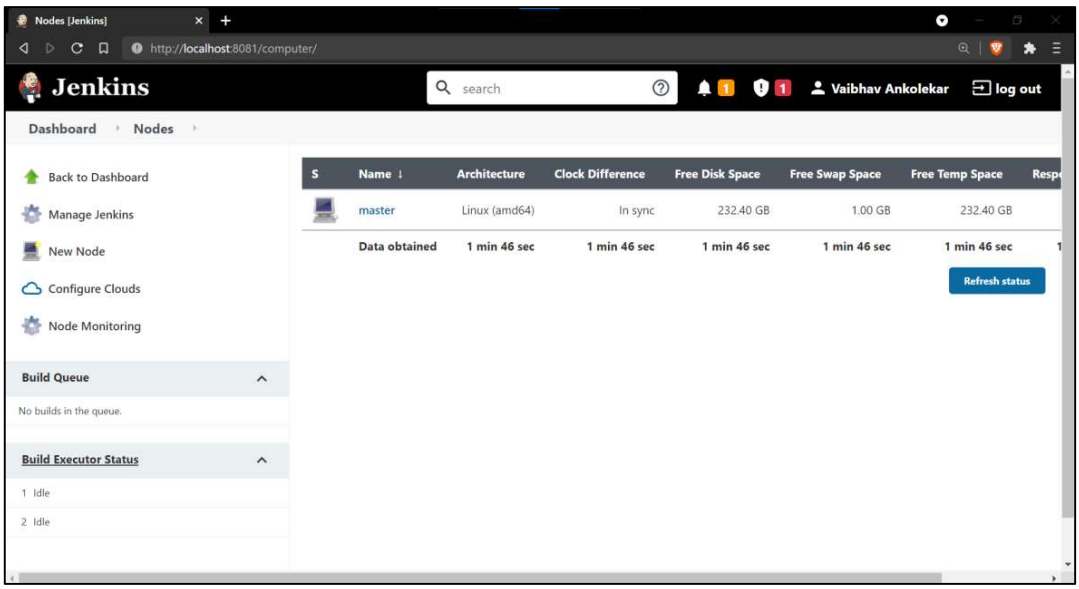
1) Open Jenkins Dashboard and click on Manage Jenkins.



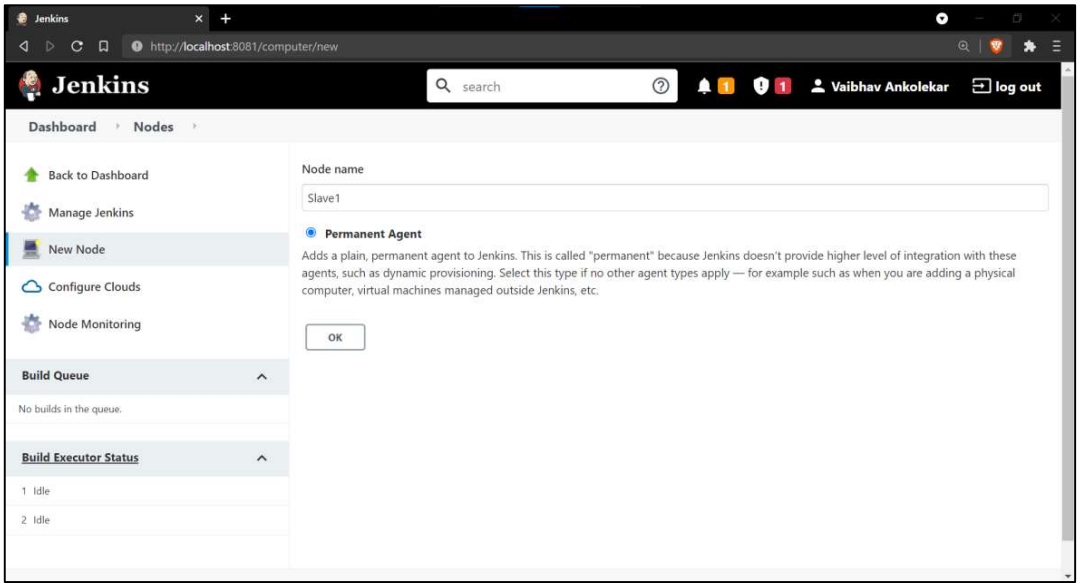
2) Click on Manage Nodes and Clouds



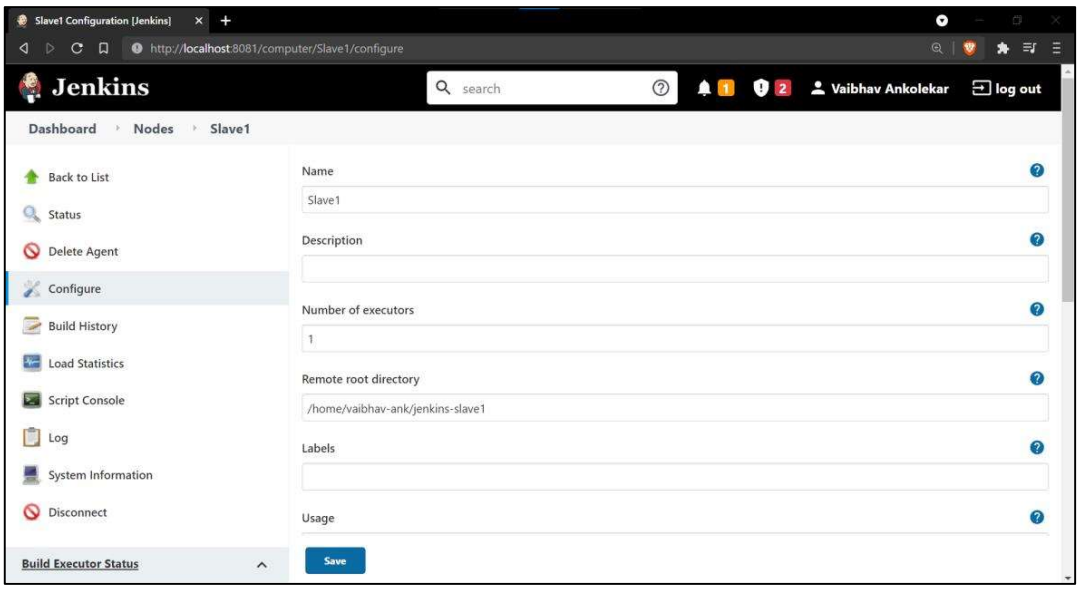
3) Click on New Node.



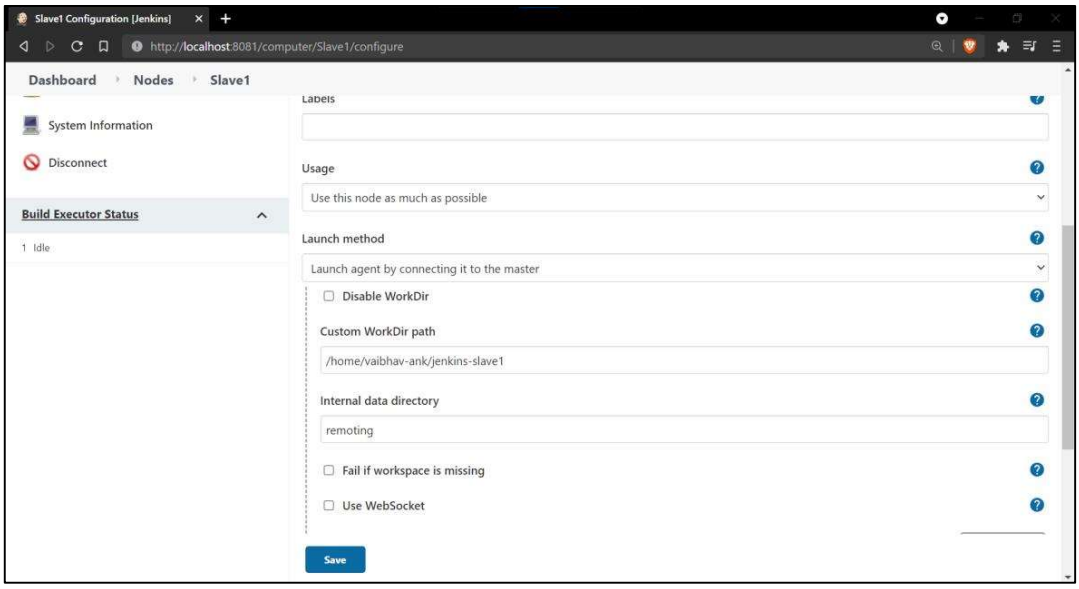
4) Give a name for your Node, select Permanent Agent and click OK.



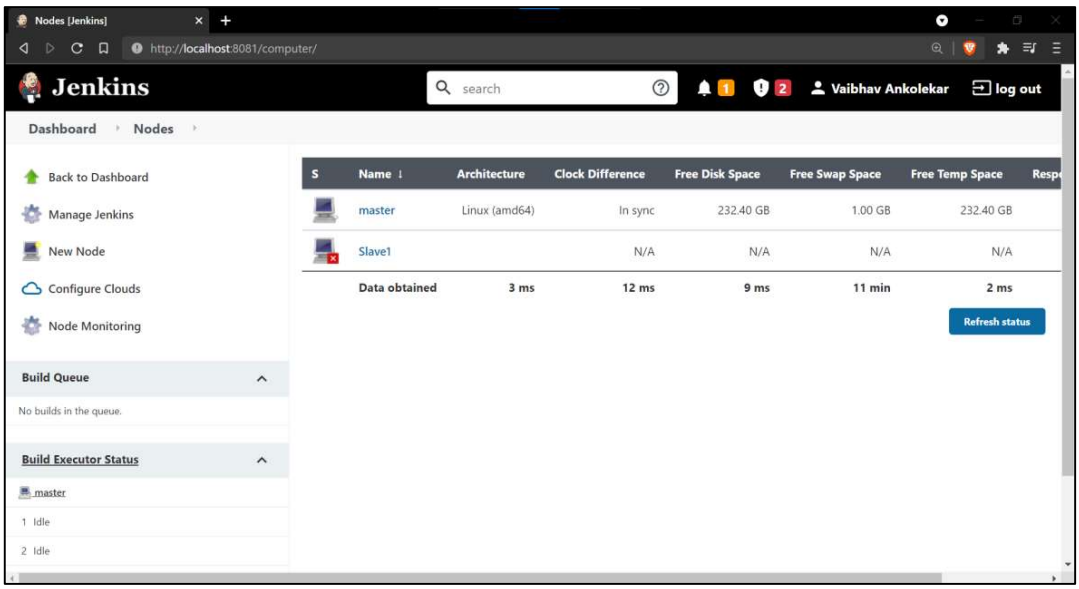
5) Now configure the new node as per your choice.



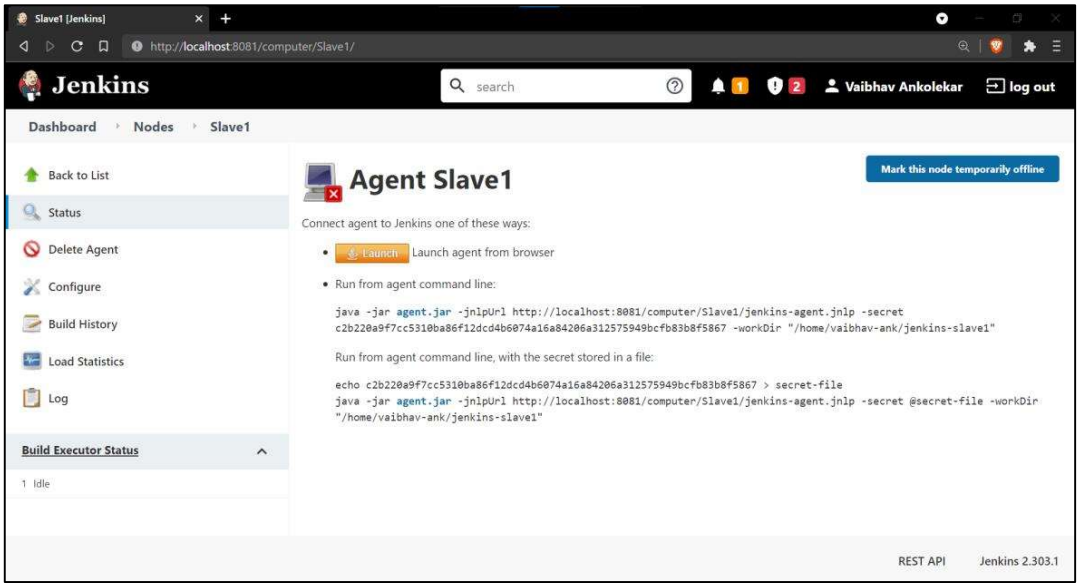
6) In Launch method choose: *Launch agent by connecting it to master*. Fill the next details and click Save.



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

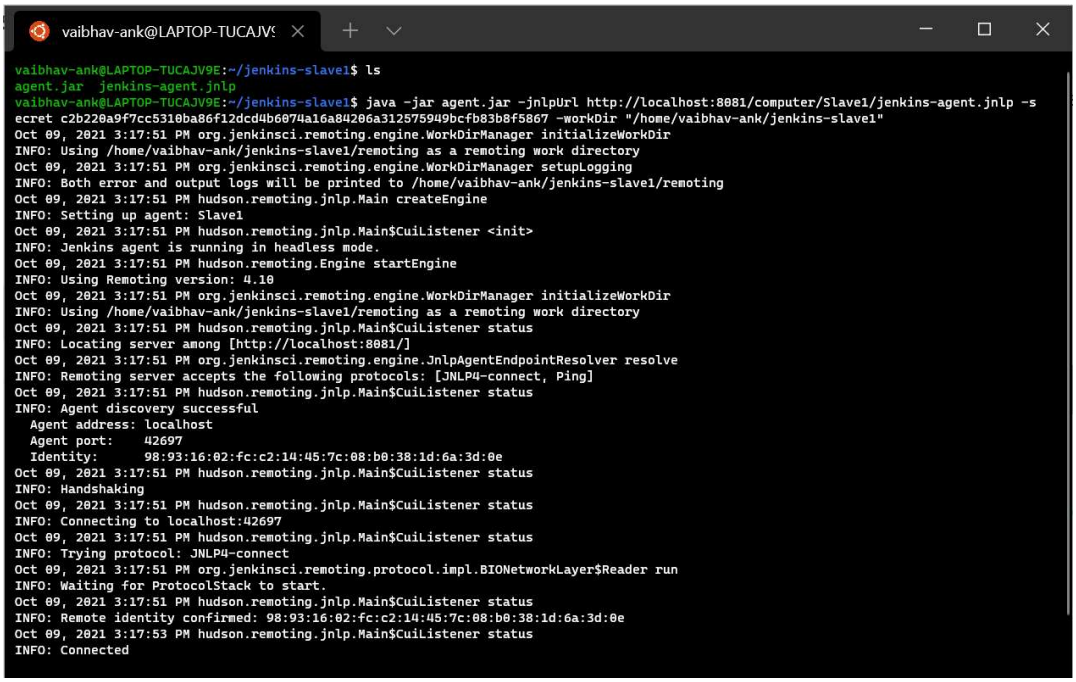


8) Click on the blue link agent.jar to download it.

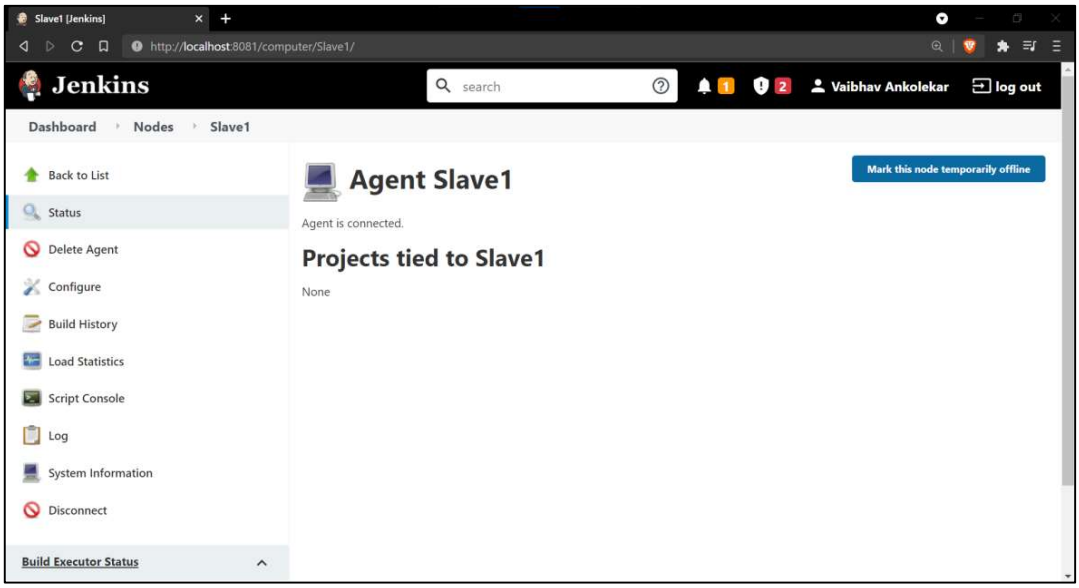


9) In the new node machine, the directory which was stated while configuring it and place the agent.jar file in it.

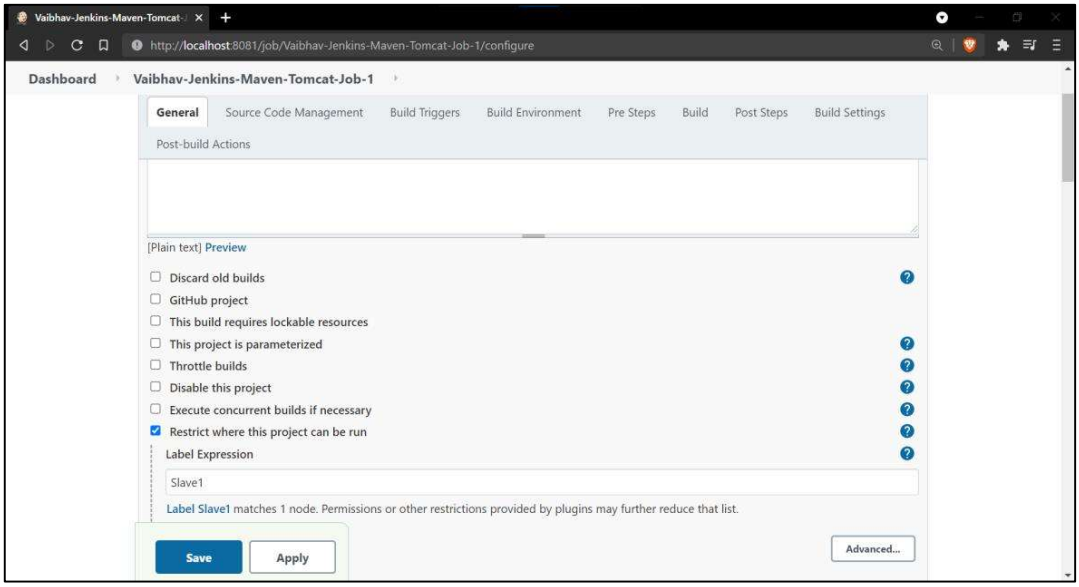
10) Now run the command given in the Node's Jenkins dashboard.



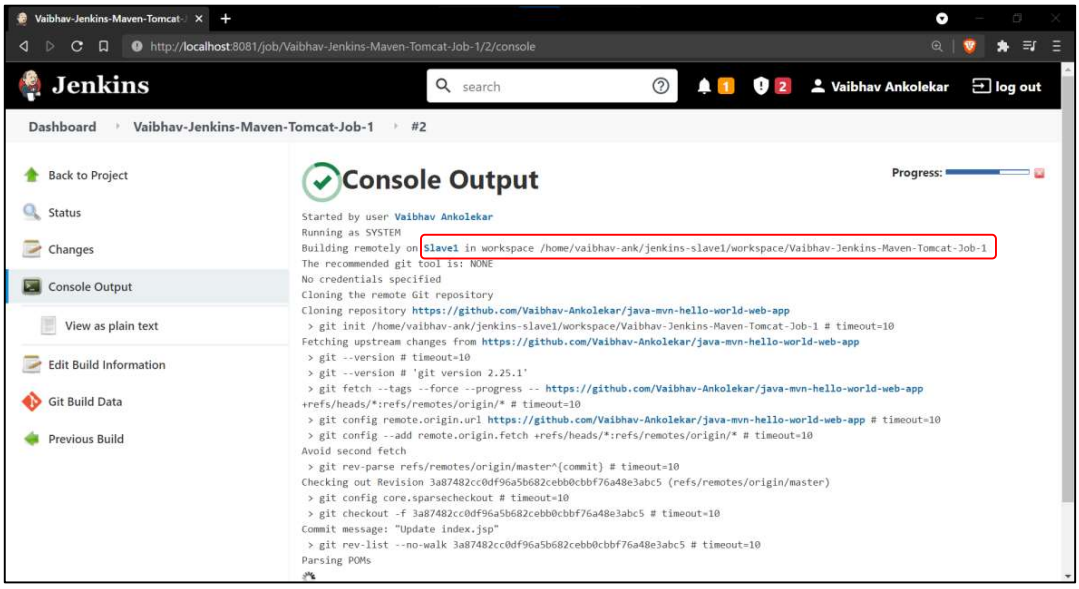
11) If you refresh the Jenkins tab, it will show the node in now connected.



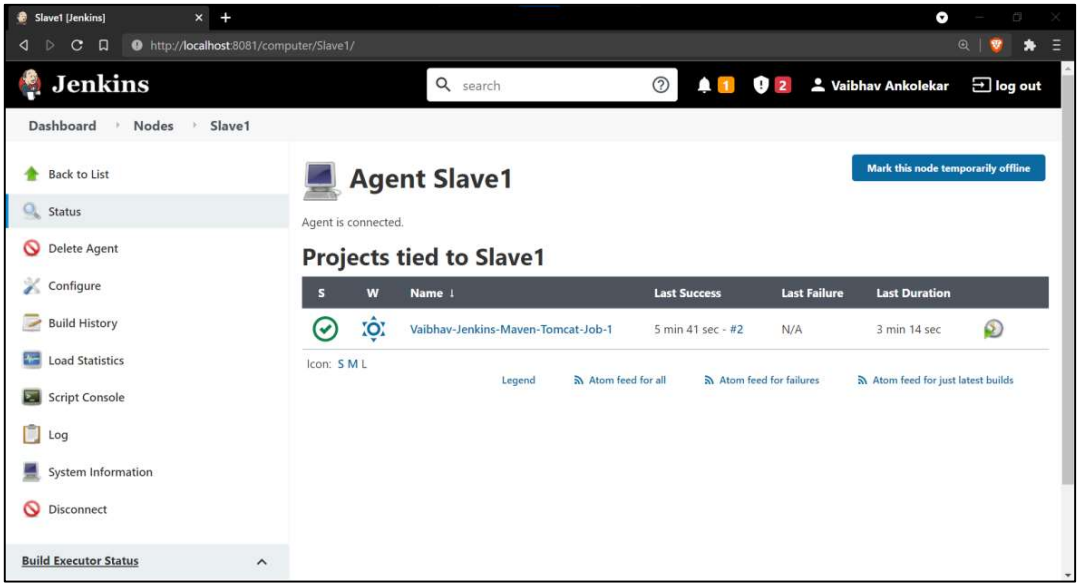
12) Configure an existing job. In General, select 'Restrict where this project can be run' and give the node's name we just created.



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.