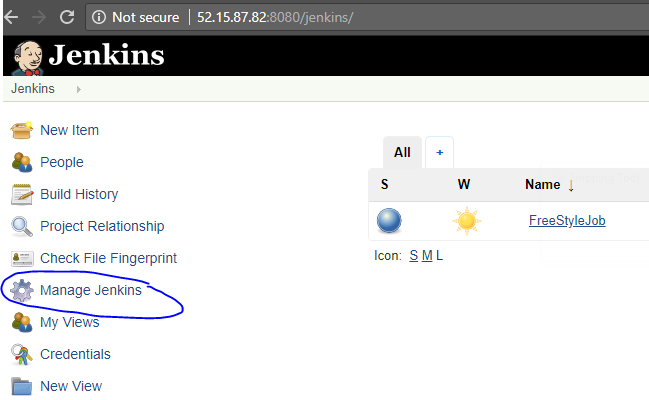
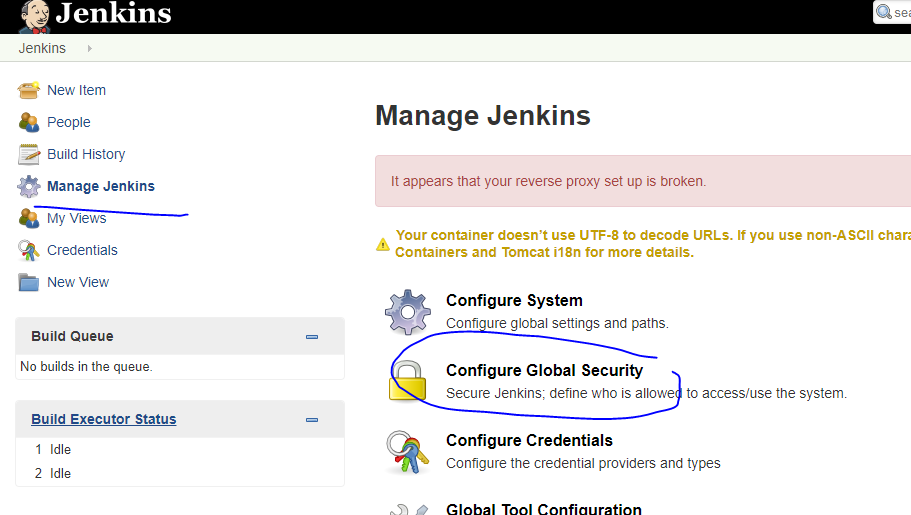
**Jenkins master and slave configuration:**

**Linux and windows Slave configuration**

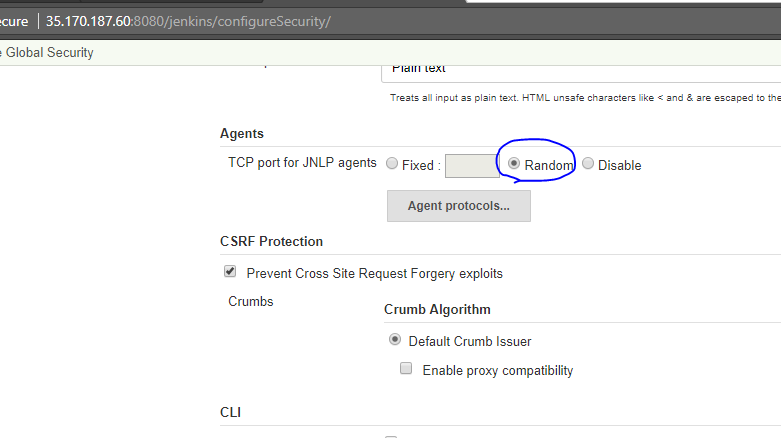
1. **First Enable JNLP (**Java Network Launch Protocol)**:**

Go to “Manage Jenkins”





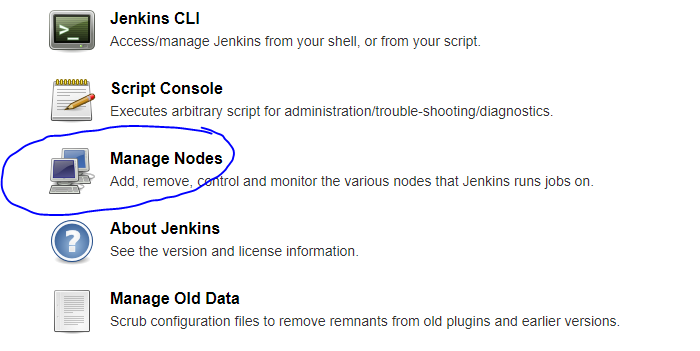
Choose the option Random🡪 Save the config.



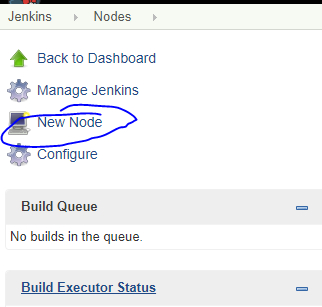
Go back to “Manage Jenkins” page.

1. **Configure node for Linux**:

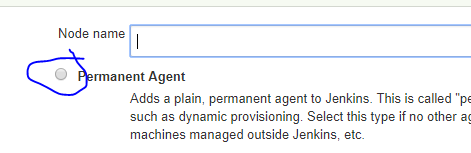
Click on manage nodes:

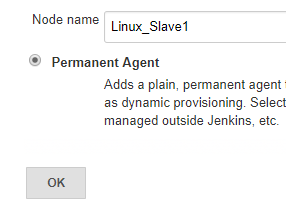


Click on new node:



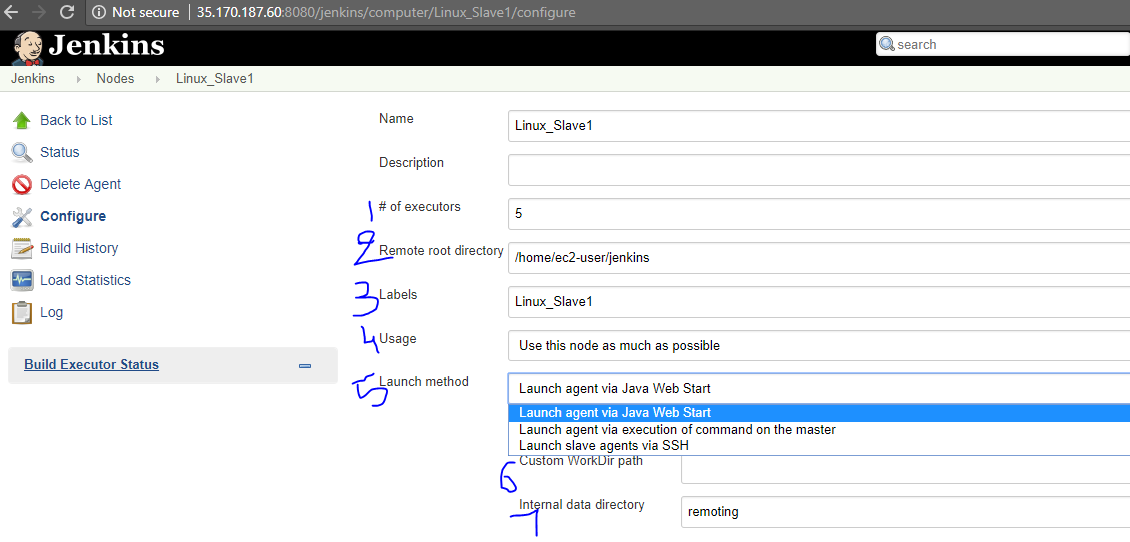
Give the node name: ex: “slave1”





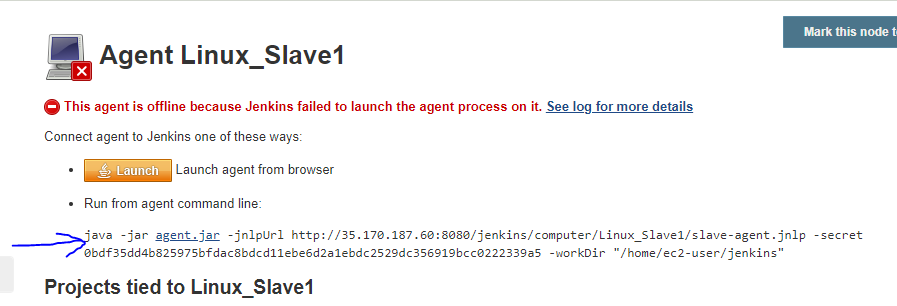
1. Increase the number of executors count to more than one. Default value is 1.
2. Create a folder ‘jenkins’ in Slave machine & give that path here. (You can give any name or any path from the slave machine).
3. Give any name to Labels.
4. Leave with Default option: “Use this node as much as possible”.
5. Choose “Launch agent via Java Web Start” (once you enabled the JNLP then only this option is available).
6. If you want you can give the custome work directory path or give the same as “Remote root director” path as we already given in the section 2.
7. This is default folder ‘remoting’, but you have to create it under the folder “/home/ec2-user/jenkins”.

Save the configuration.



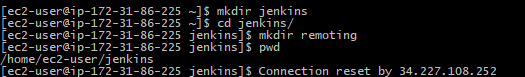
Copy the java command & run this on slave machine.

Before you run this java command you have to create the folder ‘remoting ’under “/home/ec2-user/jenkins” & download the agent.jar under “/home/ec2-user/jenkins”



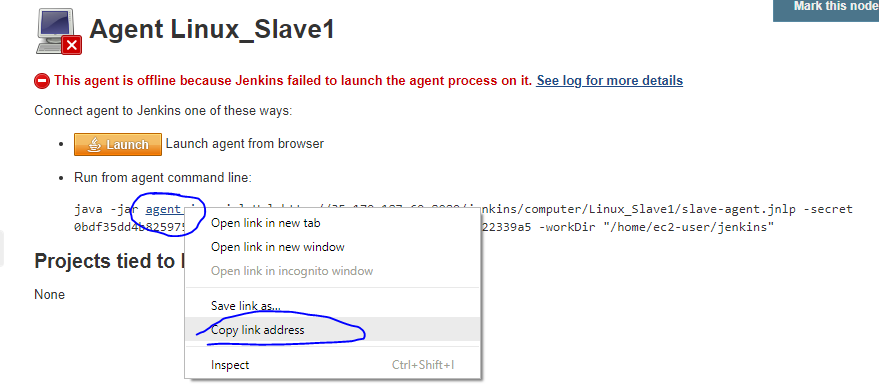
Before you run this java command(as shown in above pic) you have to create the folder ‘remoting ’under “/home/ec2-user/jenkins” & download the agent.jar under “/home/ec2-user/jenkins”. See the below image.

Create the folder ‘remoting ’under “/home/ec2-user/jenkins”



Download the agent.jar under “/home/ec2-user/jenkins”.

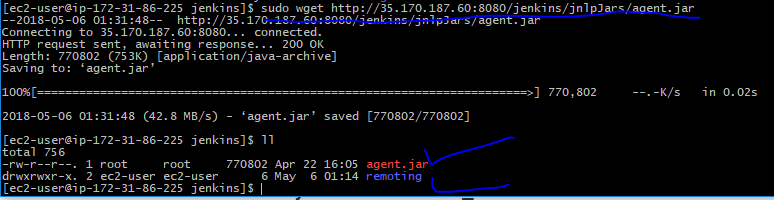
To download, right click on the link agent.jar , it will copy the agent.jar file location which is nothing but out Jenkins master, it will connect the slave to Jenkins master. URL mostly will be http://<IP Address>:8080/jenkins/jnlpJars/agent.jar



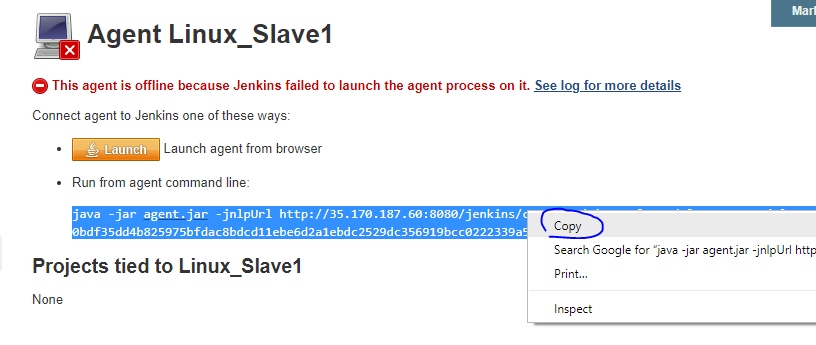
Download it on slave http://<IP Address>:8080/jenkins/jnlpJars/agent.jar

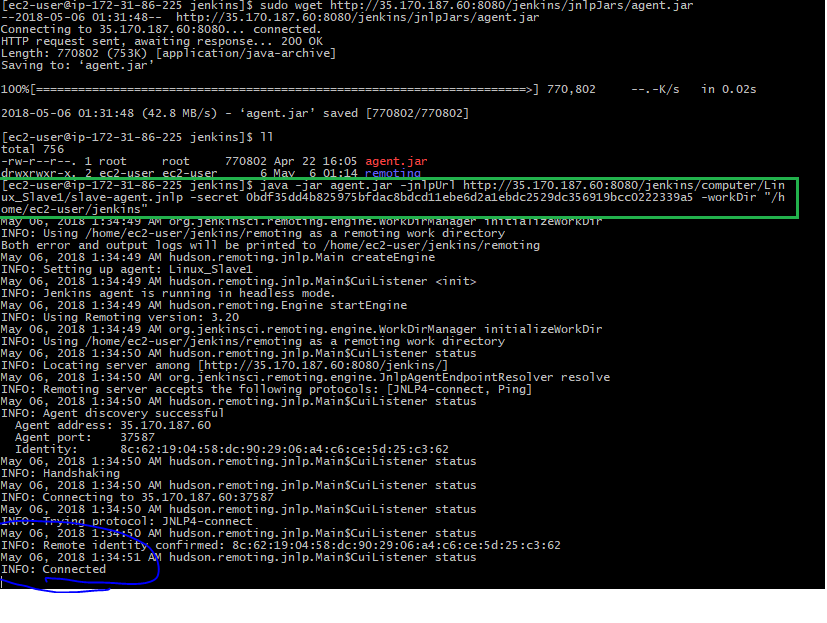
Command: sudo wget http://<IP Address>:8080/jenkins/jnlpJars/agent.jar

If wget package is not installed 🡪 run the command 🡪 sudo yum install wget -y

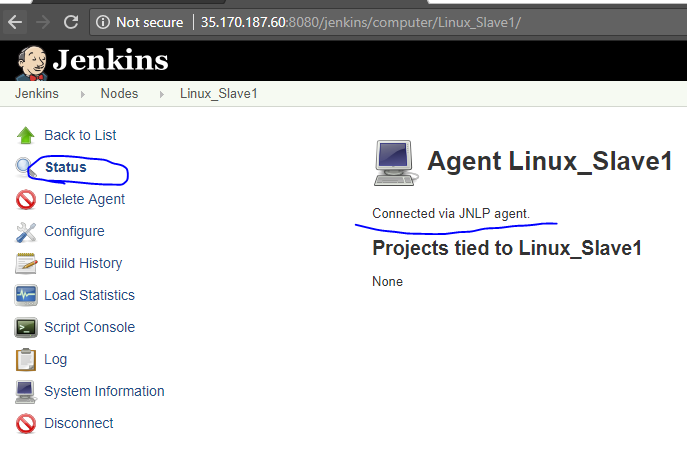


Now copy the java command & run it on the slave machine.

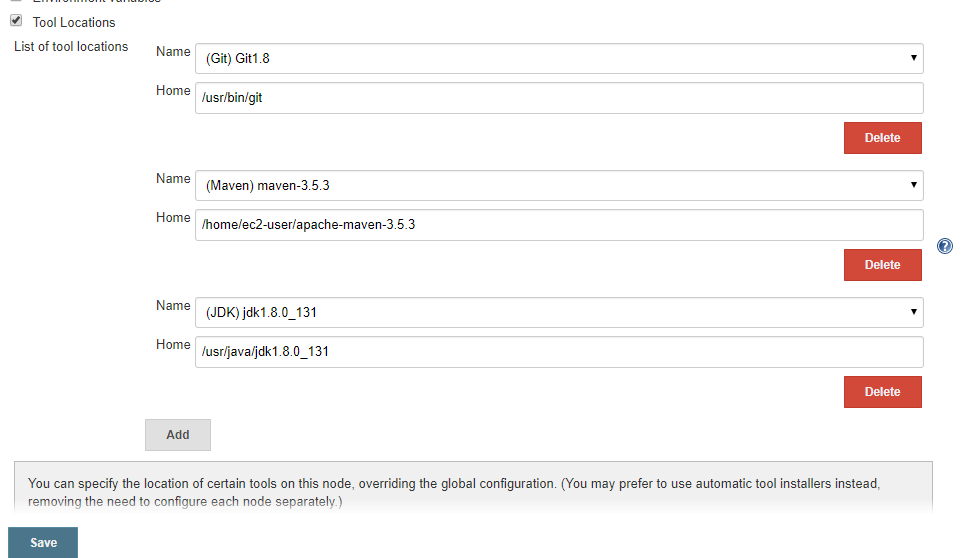




Now, go to Jenkins and see the node status: Click on status or refresh the page.

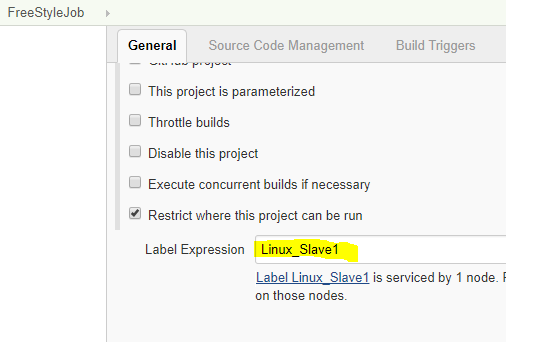


Go to Linux\_Slave1 configuration and add the paths of the tools which are installed on node machine. To work with Jenkins builds, we should to install the required tools on the slave machine.

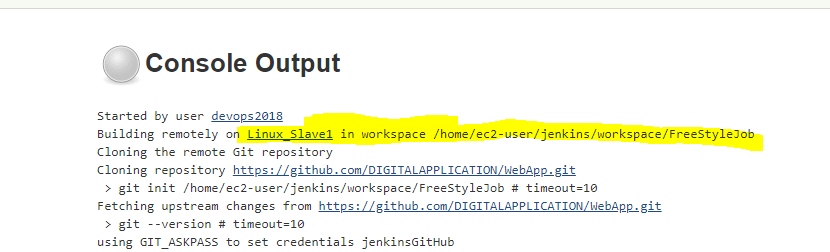


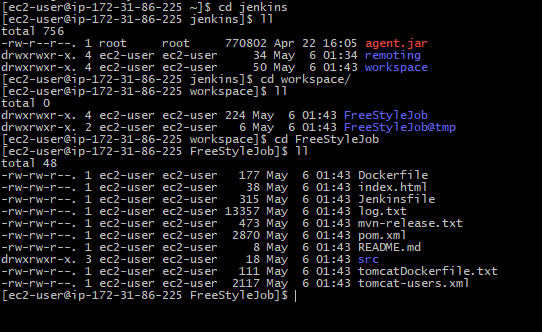
Sample example job which is configured & running on the **Linux\_Slave1** machine.

Go to Jenkins job configuration🡪 select restrict where this project can be run🡪 choose **Linux\_Slave1** **🡪** save the job config & triger the job.



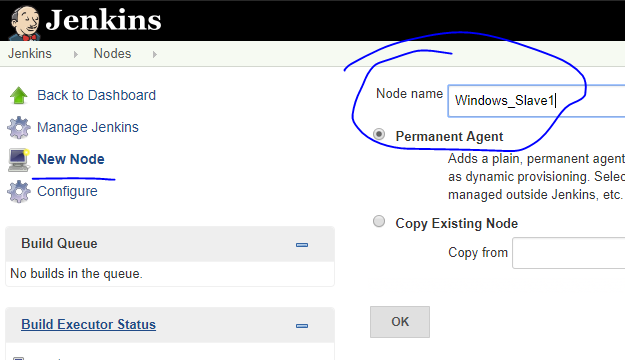
See the build console, its running “**Linux\_Slave1**” machine. Also see the path, job is creating workspace on slave & running the job.

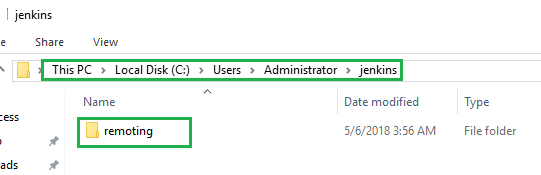


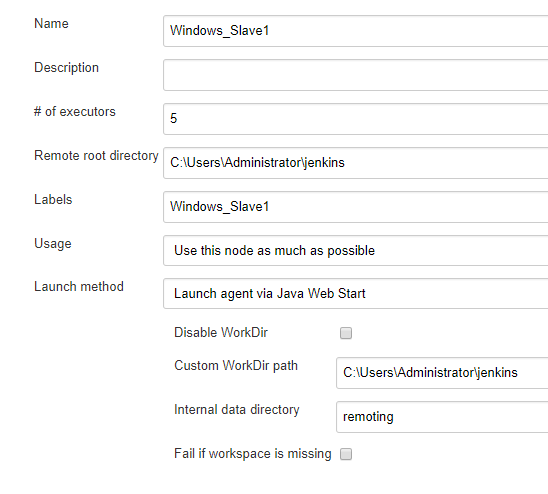


1. **Configure Node for Windows**:

Go to Manage Nodes:

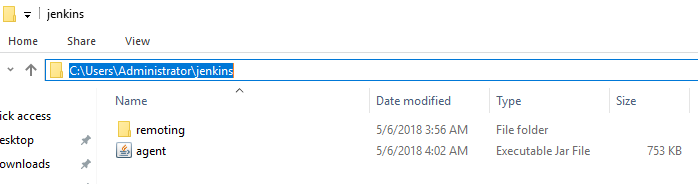




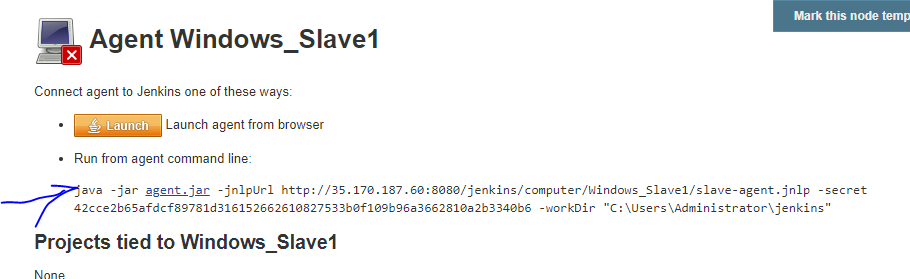


Download the agent.jar in windows slave machine. Just run this URL in any browser http://<Jenkins IP Address>:8080/jenkins/jnlpJars/agent.jar

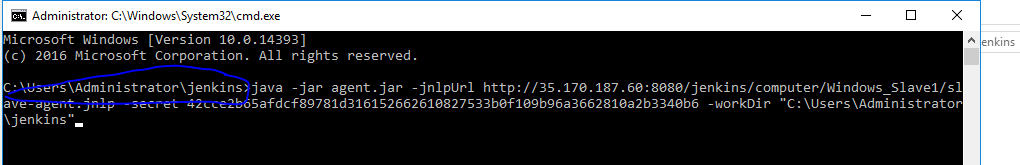
Keep the agent.jar file under Jenkins folder.

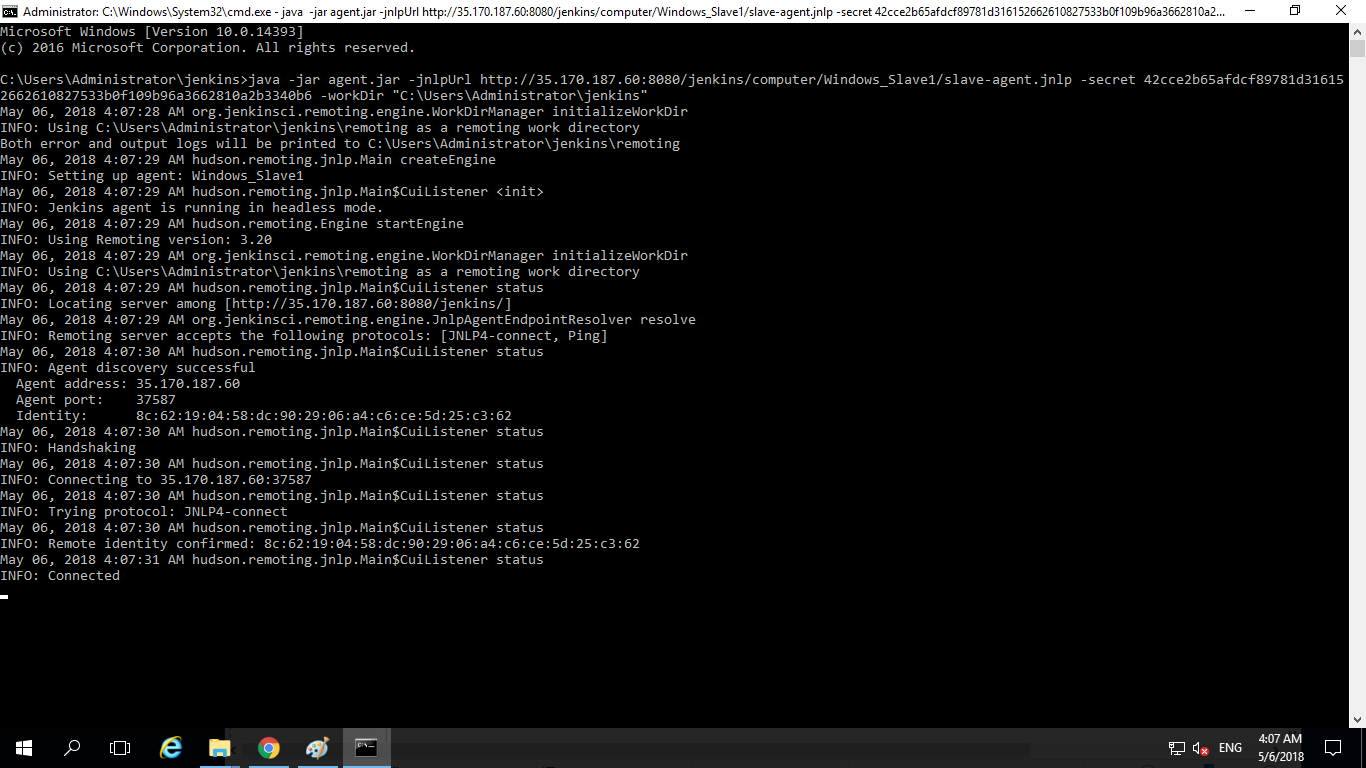


Copy the command & run the commands from windows machine command prompt

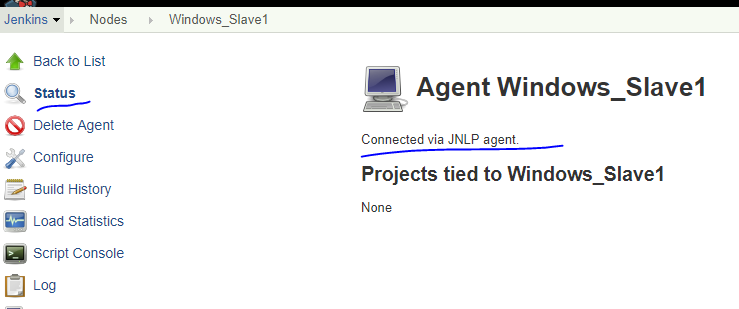


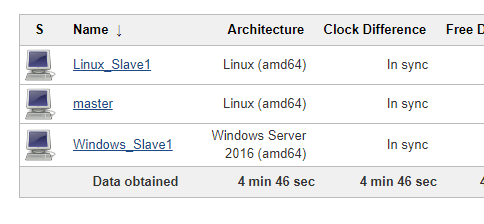




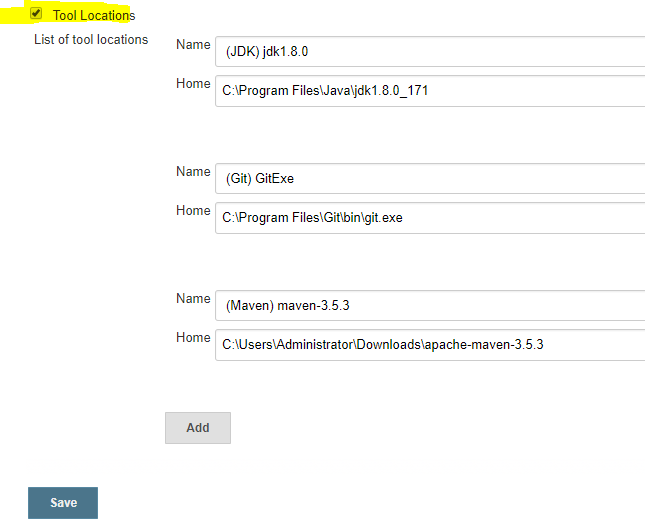


Go to Nodes and check whether the Wi8ndlows slave is online or not. Click on status or refresh the page.

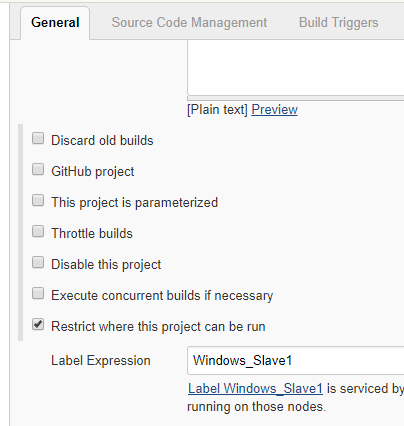




Setup the JDK, Git, Maven paths in the slave configuration.

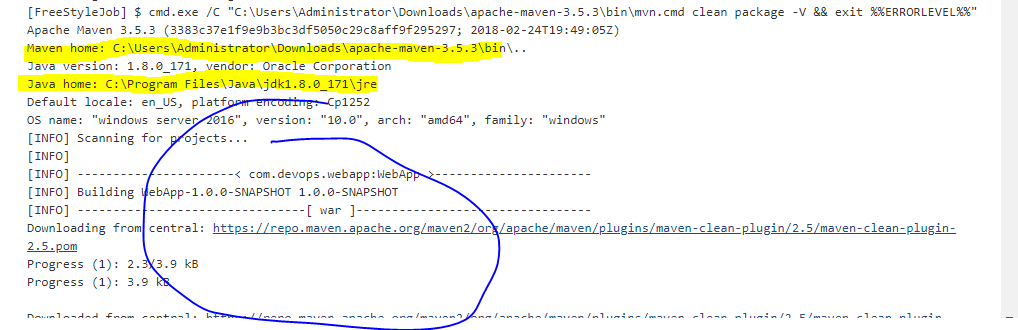


Configure a Jenkins job with win slave:

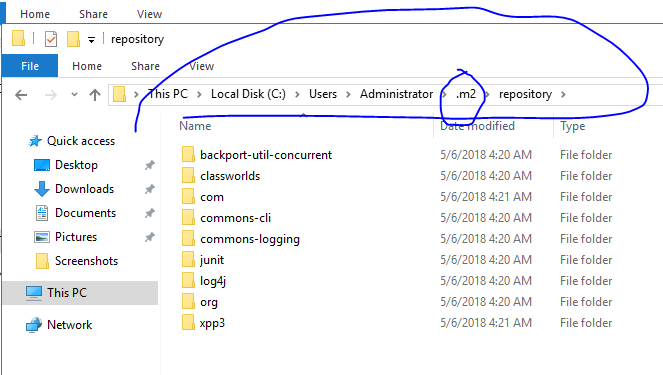




Maven build is running on windows:



Maven created a local repo on win slave & downloaded required artefacts.



Created a job workspace in the windows slave:

