**Name: Shashwat Kumar**

**Roll Number: R171218093**

**SAP ID: 500068392**

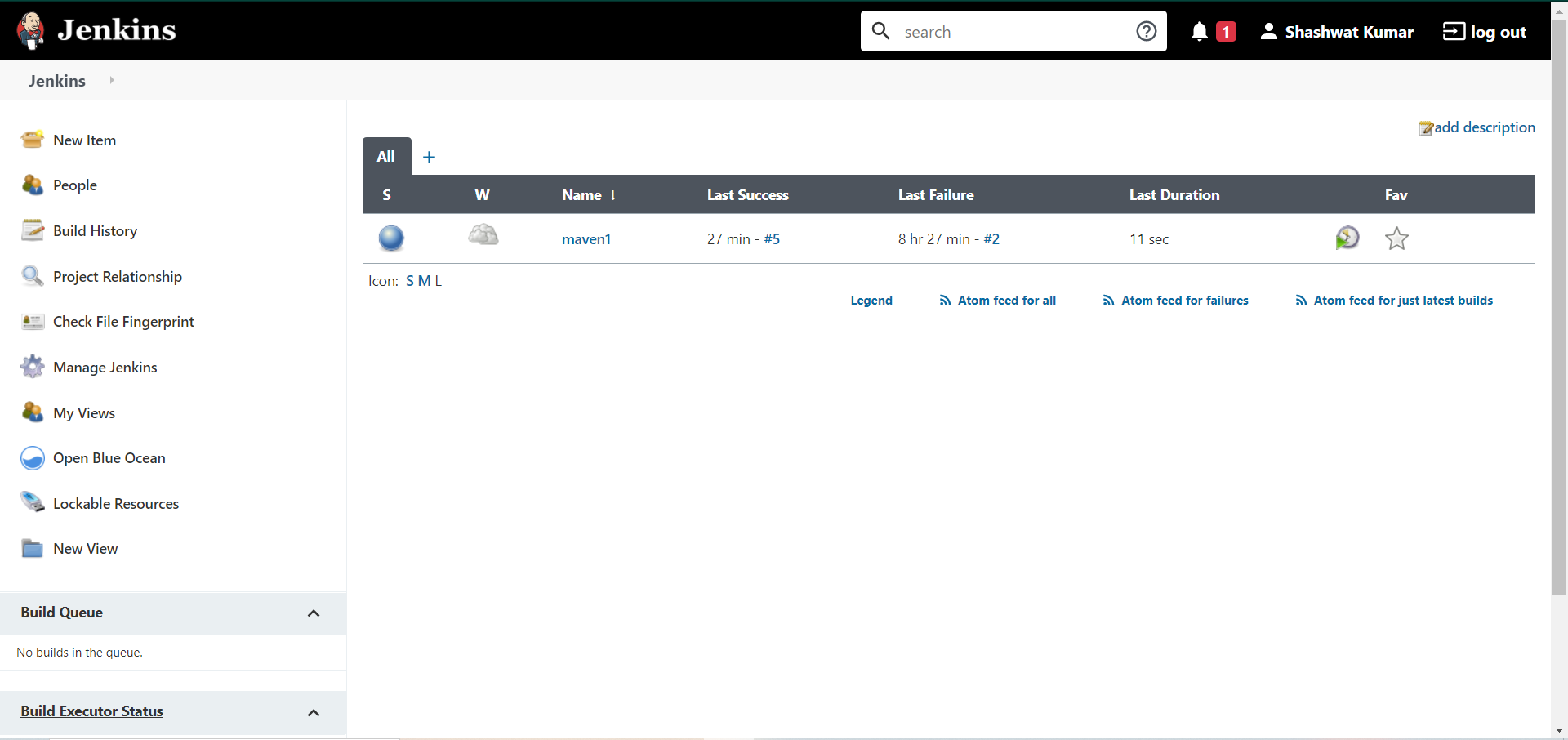
**Lab: 8**

**Implementing Master/Slave architecture in Jenkins**

**Solution:**

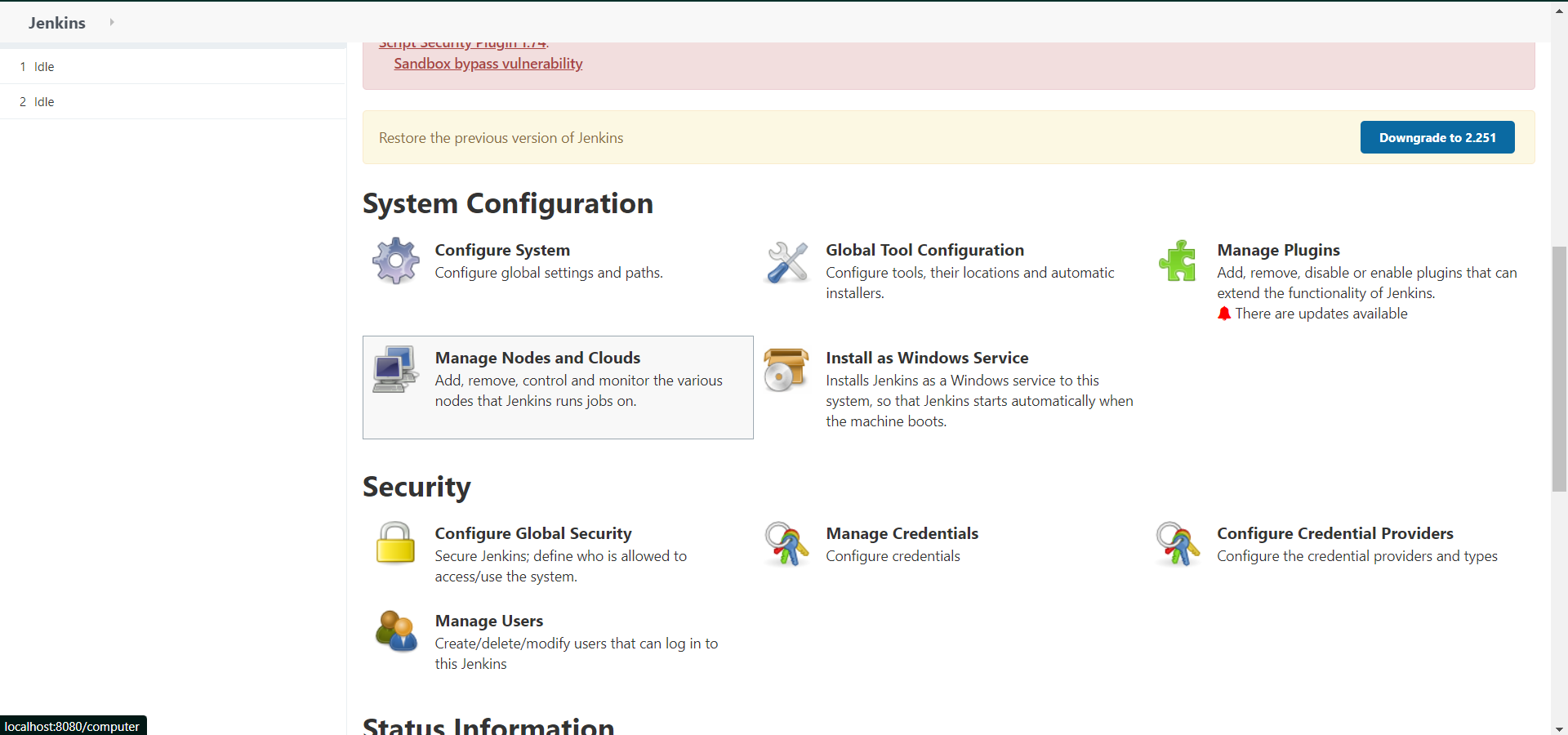
Start the Jenkins service using the java -jar jenkins.war command in the command prompt

Go to the Jenkins dashboard at localhost:8080

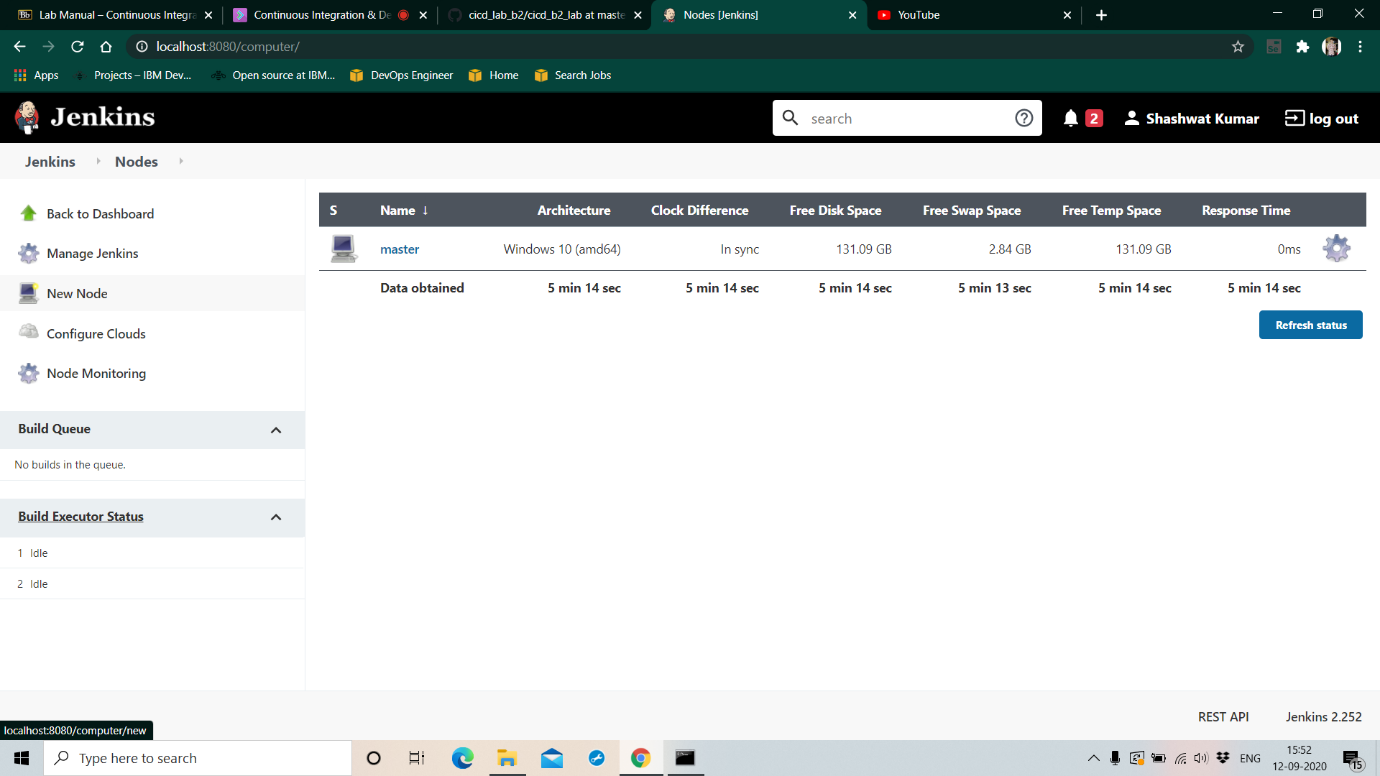


Select the option of Manage Jenkins.

Then Select Manage Nodes and Cloud



Click on new Node option

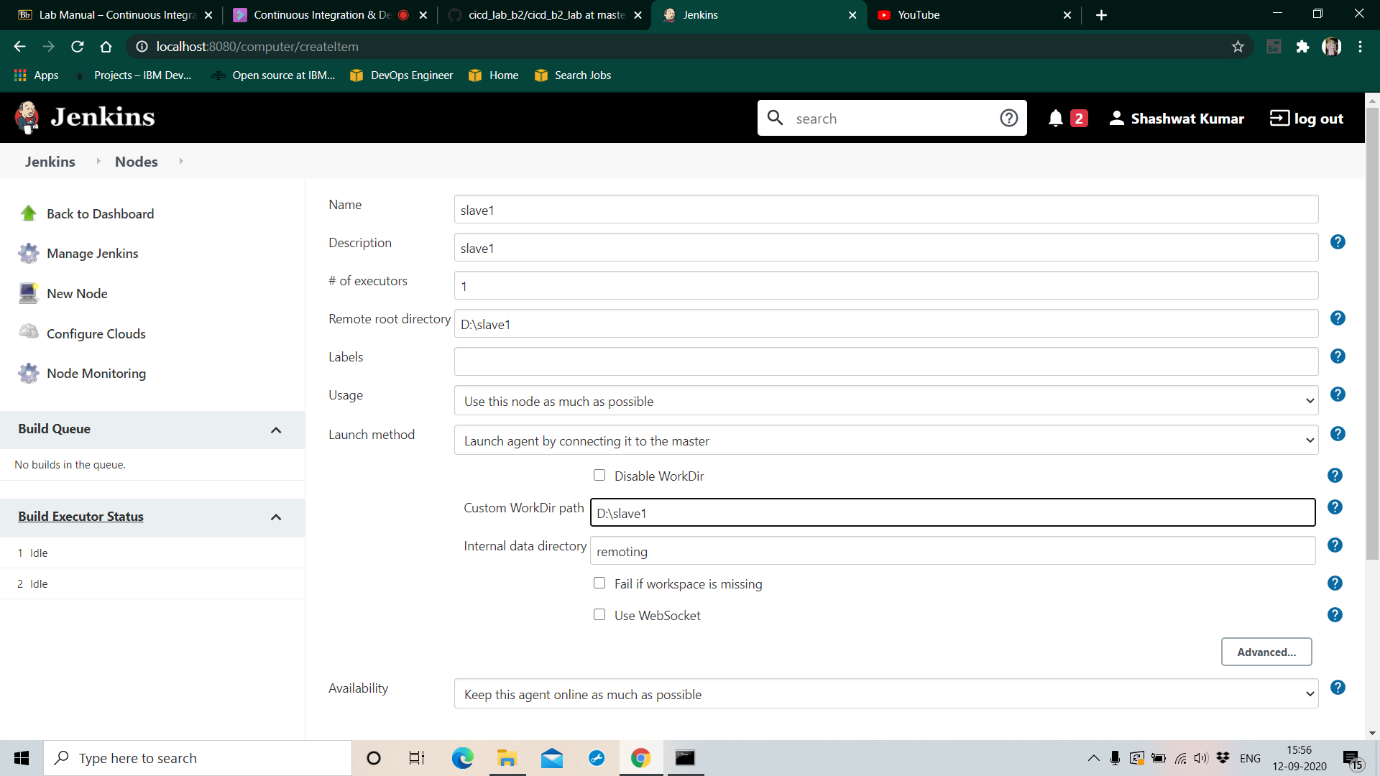


It should land you on this page.

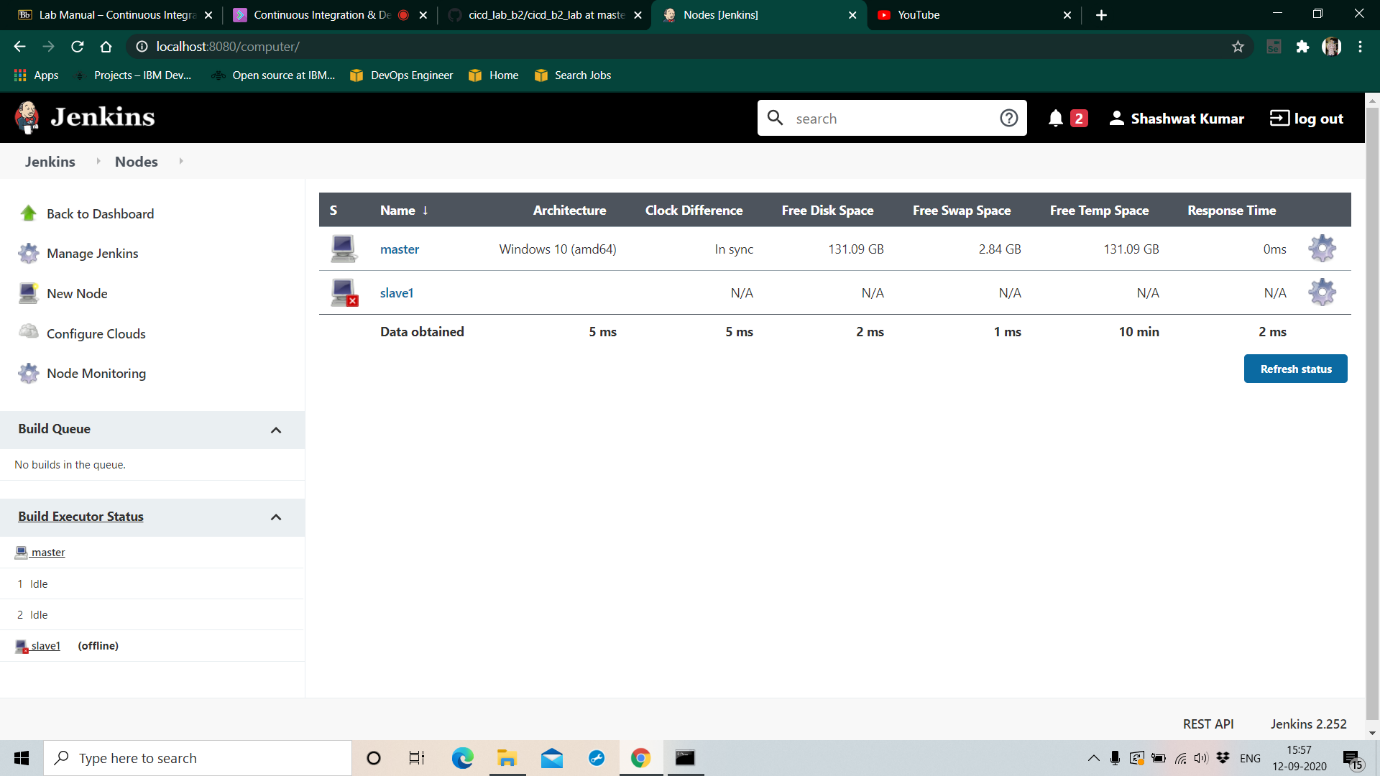
Create a slave node.

Give a name and description to this slave node.

Create a directory for the slave node and provide the location of the directory in the ”Custom WorkDir path”



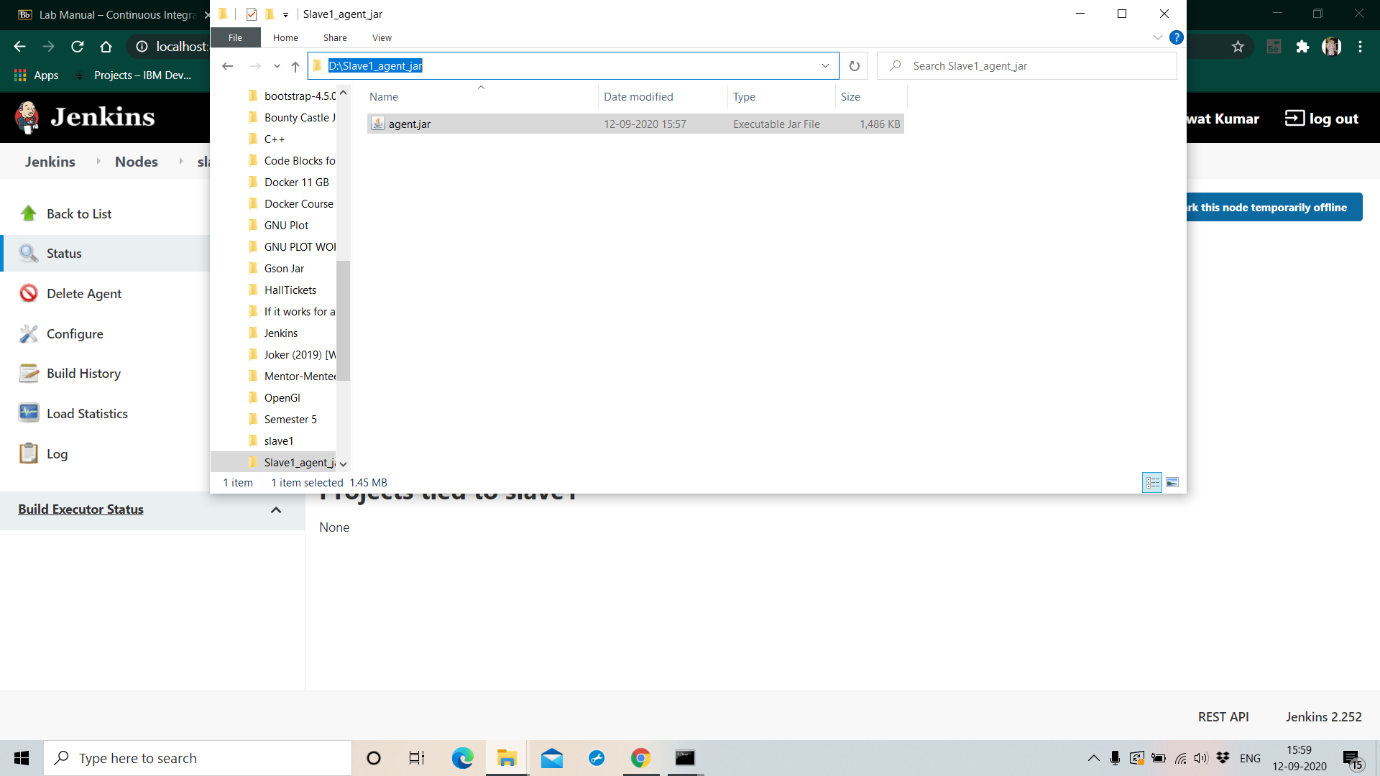
On clicking okay, it should lead you to this page.



Open the folder created for the slave node “Slave 1”

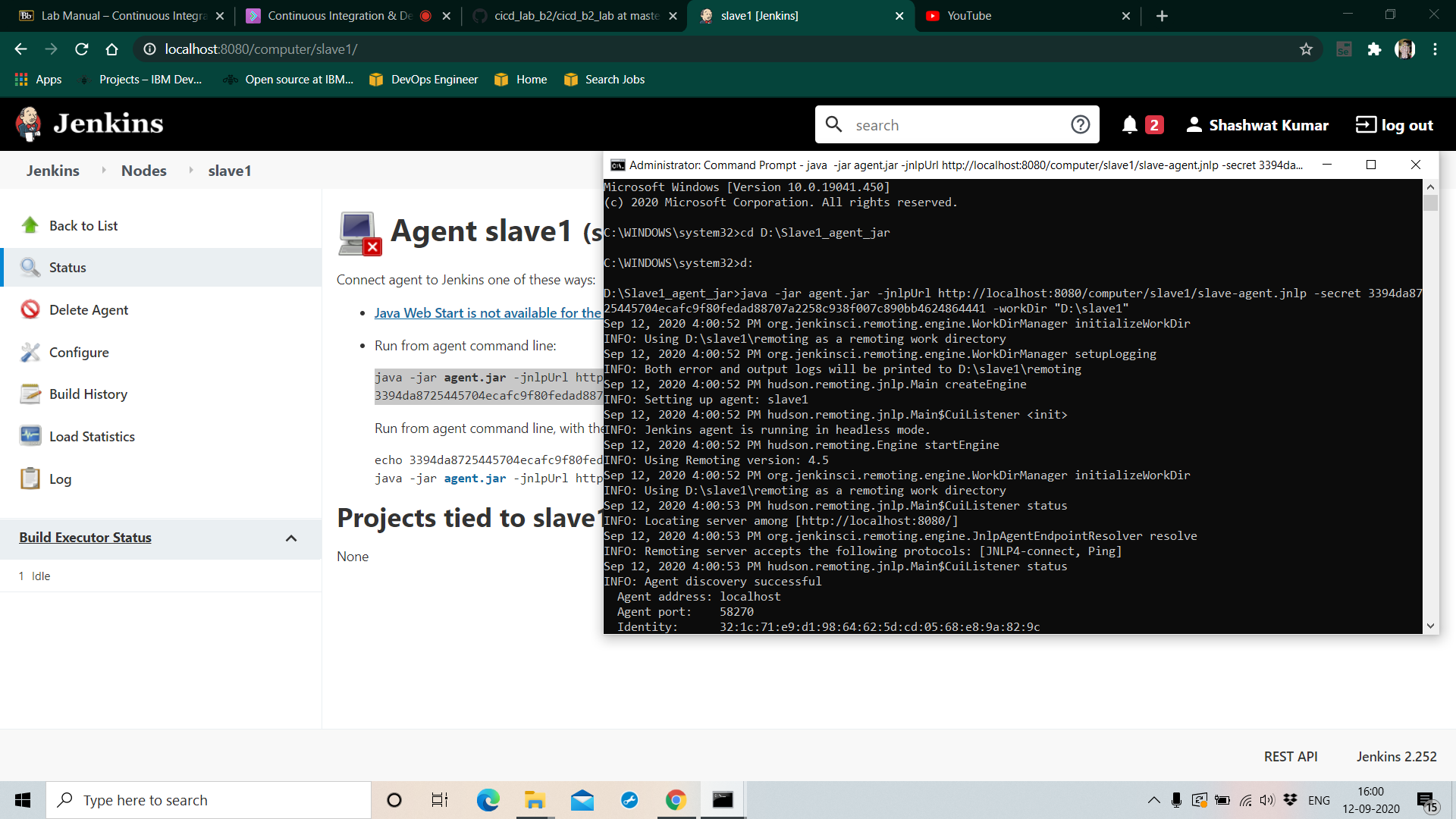
A new jar file “agent.jar” should be created

Copy the location of the directory where the agent.jar file is created

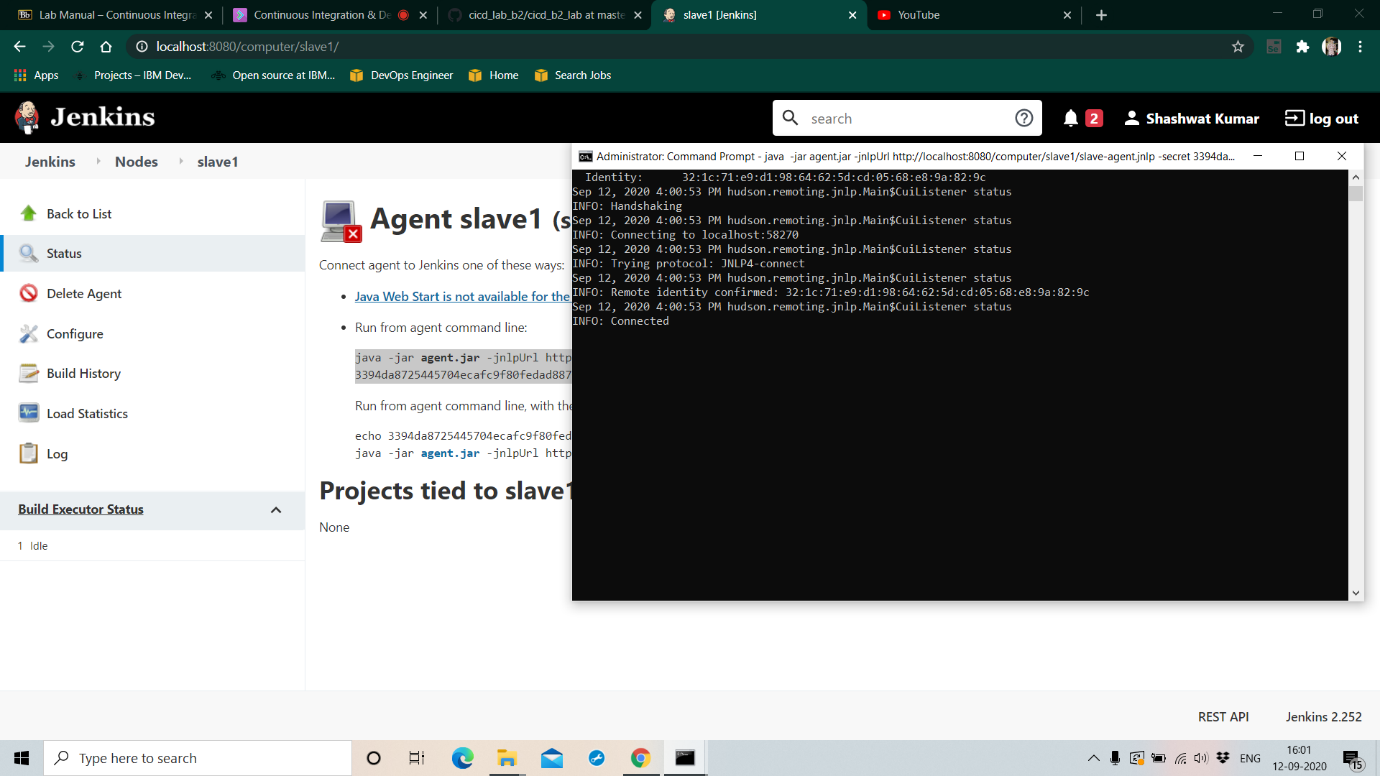


Traverse to the location of agent.jar

Run the command provided for “slave 1” in its configuration settings



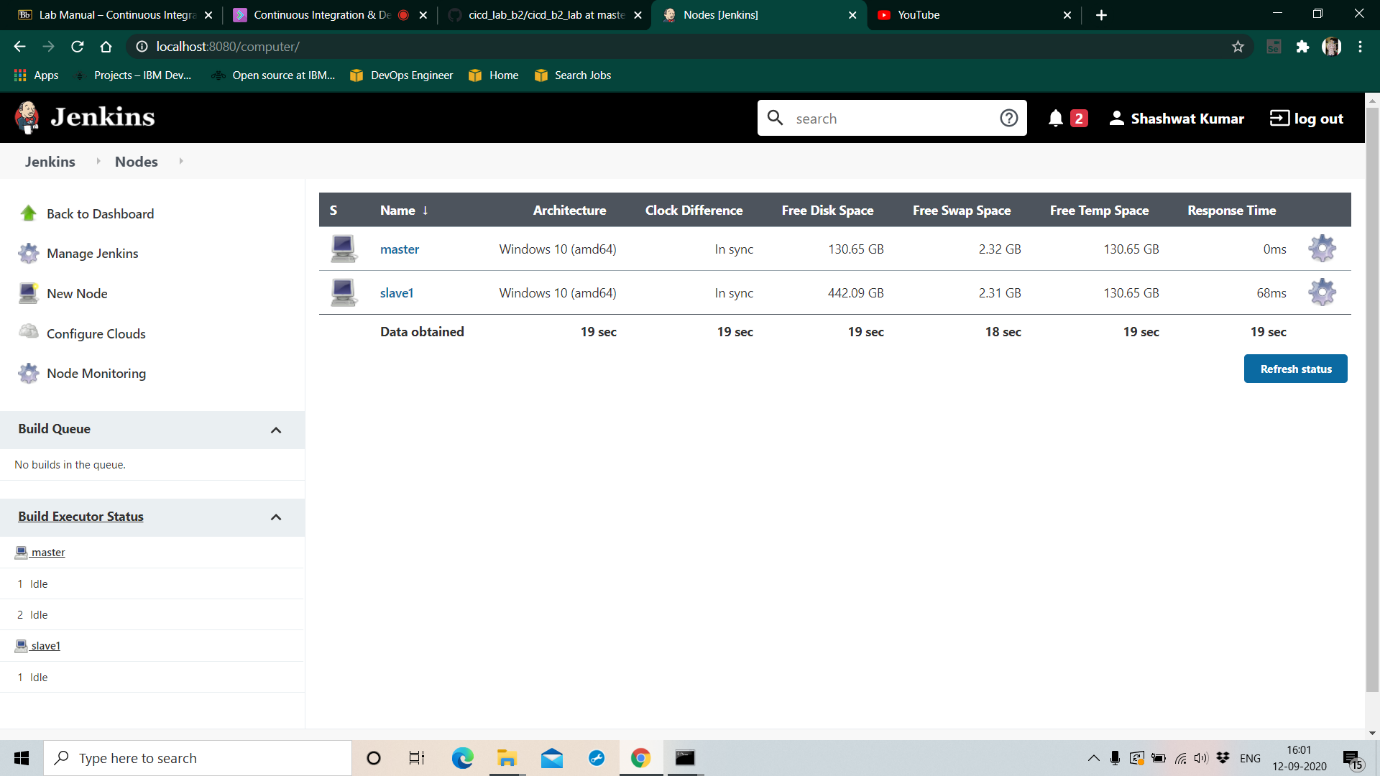
The agent “Slave 1” starts to run.



Initially the node “Slave 1” showed it was inactive.

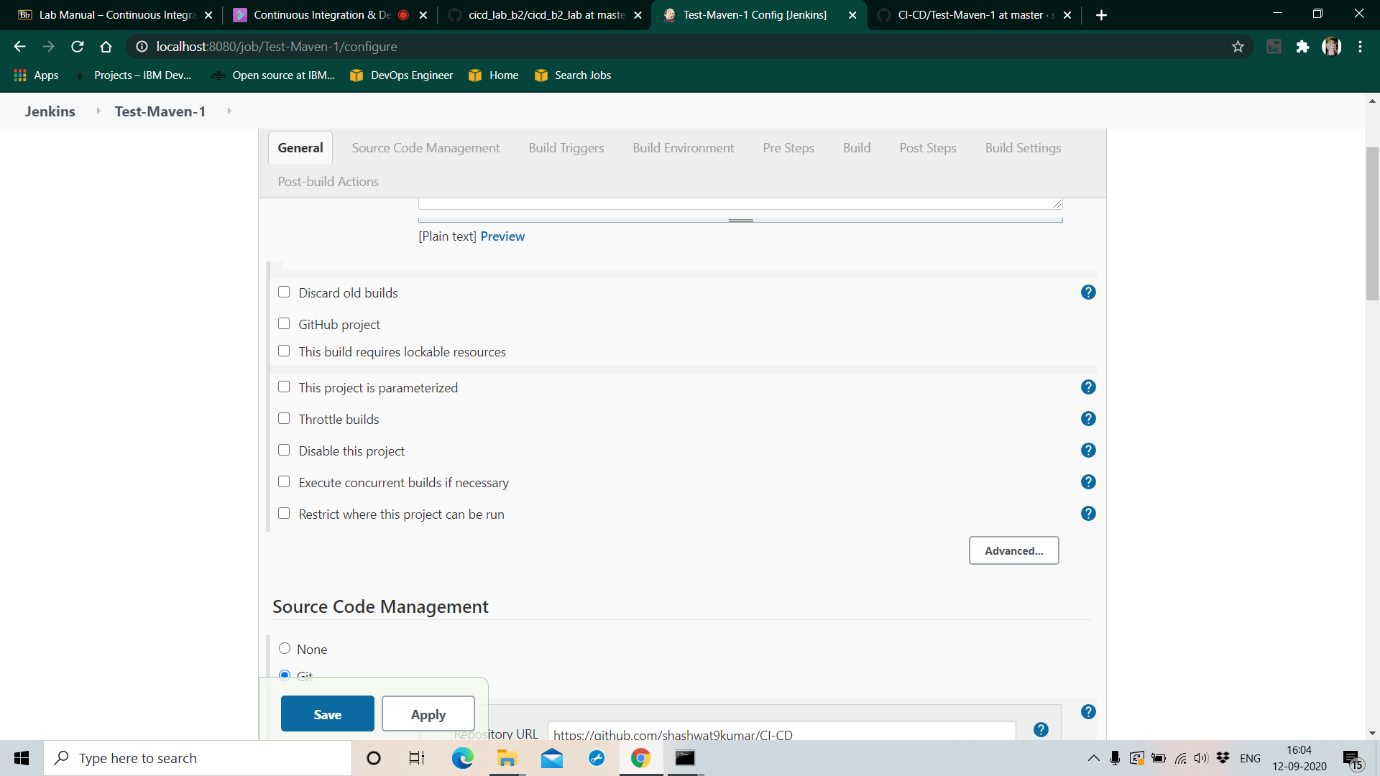
It was marked by the red cross accompanying it

Now it is active



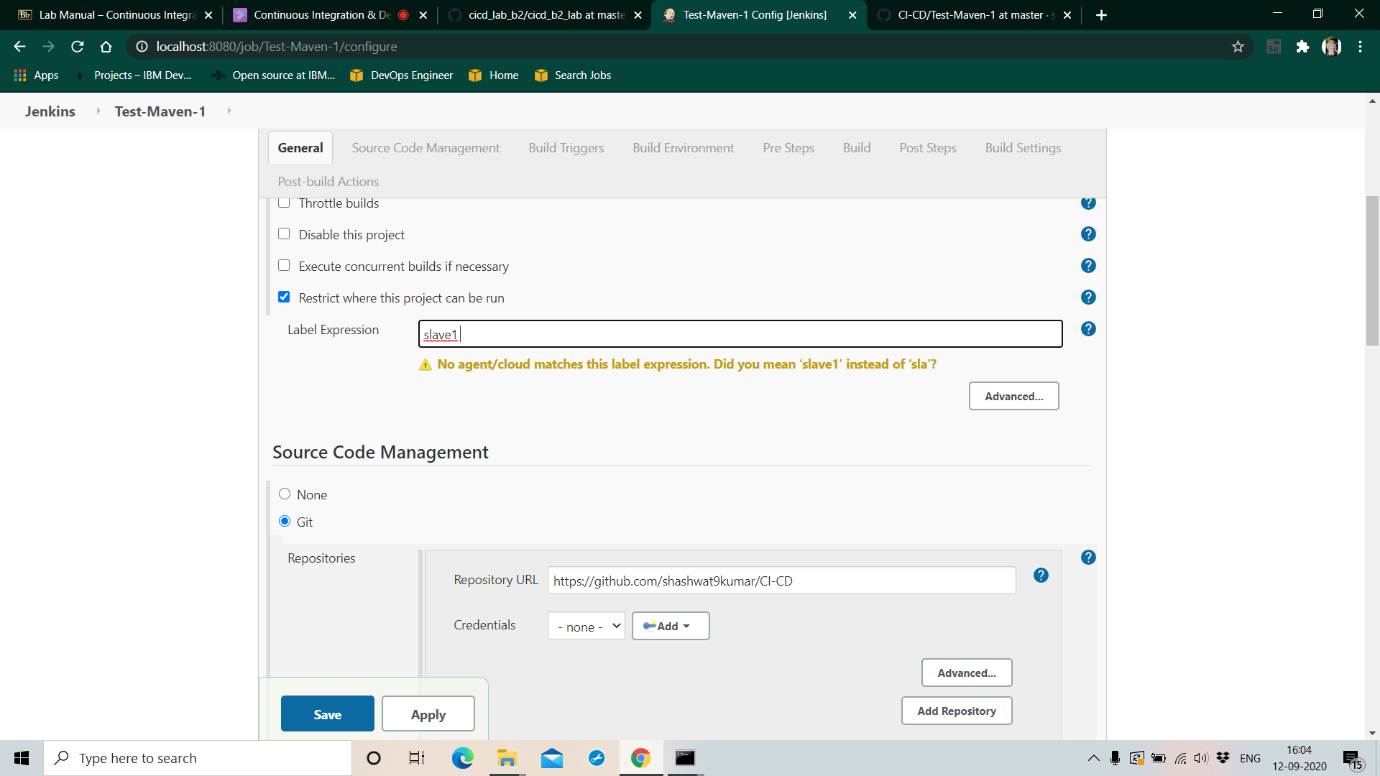
Go to the configurations of an existing maven project.

Go its “General” configurations



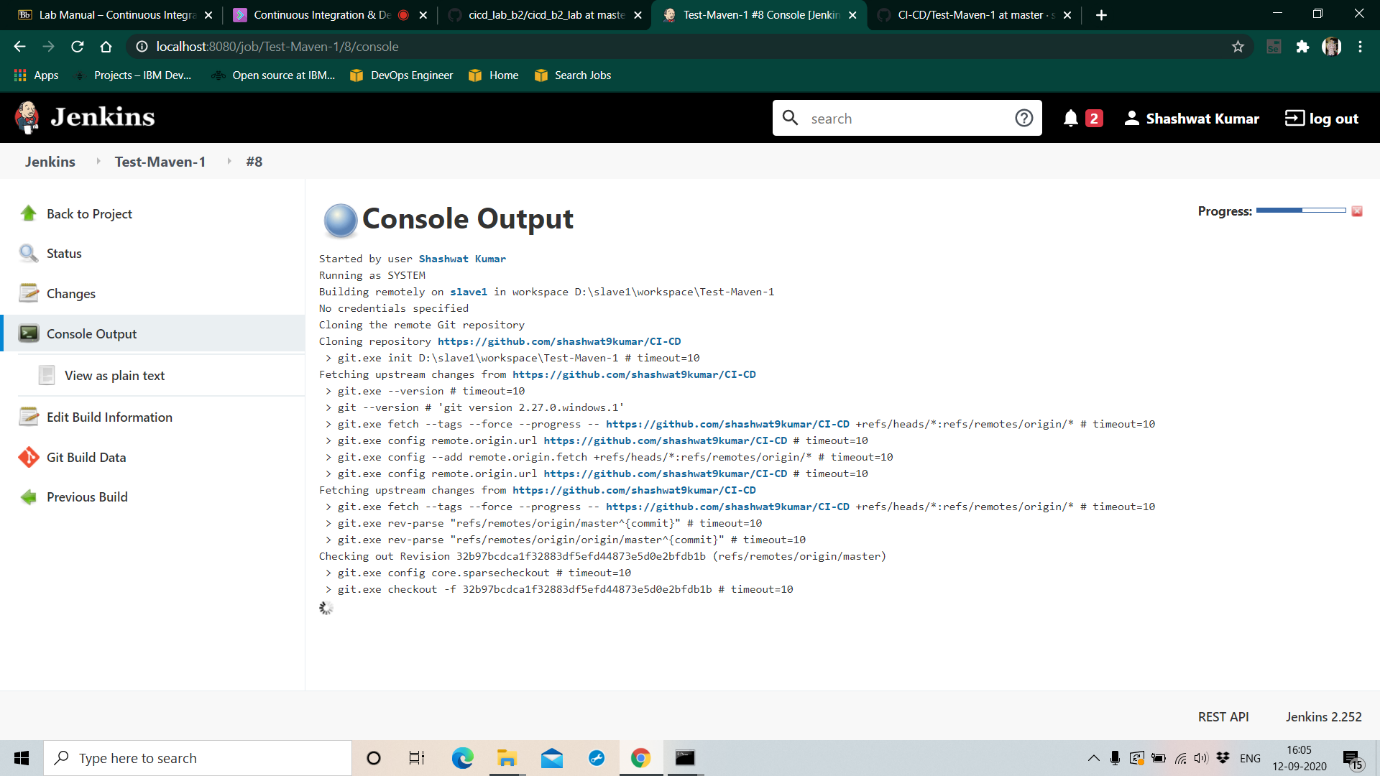
Check the option “Restrict where this project can run”

Provide the name of the slave node “Slave 1” in the label expression



Click on the “Build Now” option for the project

Select the console output to see the build process



The maven project successfully builds and it now runs on a slave node instead of the master node

