MASTER SLAVE ARCHITECTURE IN JENKINS

Imagine you have a really smart person named Jenkins who helps you with your computer tasks, like building and testing software. Now, sometimes you have a lot of work to do, and it might be too much for Jenkins to handle alone. That's where Master-Slave Architecture comes in.

Master: Jenkins has a boss called the Master. The Master knows what needs to be done and plans out all the tasks. It's like the manager who assigns jobs to the team.

Slaves: The Slaves are like Jenkins' assistants. They do the actual work that the Master tells them to do. These Slaves can work on different computers. So, if you have a lot of tasks, you can ask Jenkins to use multiple Slaves to get things done faster.

How It Helps with Automation:

1. **Faster Work:** With Slaves helping out, tasks like building software or running tests happen quicker because many things can be done at the same time on different computers.

2. **Big Jobs, No Problem:** If you're working on a huge project, the Master can split the work and give parts of it to different Slaves. They work together to finish the job faster.

3. **Different Skills:** Each Slave can be set up with special tools or skills. So, if you need to test your software on different types of computers, Jenkins can use Slaves with those specific skills.

4. **No Downtime:** If one Slave is busy or has a problem, the Master can simply send the work to another Slave. This way, the work keeps going without any interruptions.

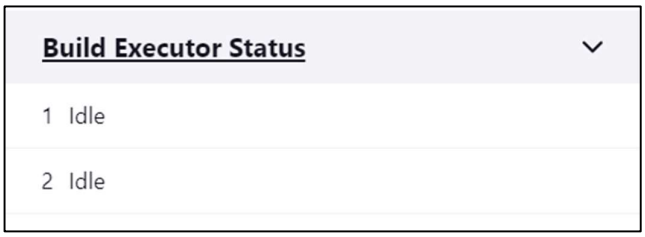
5. **Less Strain:** Since Slaves work on their own computers, your computer doesn't get overloaded. It's like having more people help, so no one gets tired.

6**. Easy Management:** The Master keeps track of everything. It tells you if things are going well or if there are any issues. So, you have better control over your projects.

So, think of Master-Slave Architecture in Jenkins like having a smart manager (Master) and a team of helpers (Slaves) who can get a lot of tasks done quickly and smoothly. It's like teamwork for your computer tasks, making things faster and more organized when you're automating your work.

Before Learning about how we can create the Master Slave Architecture Let’s Understand the work of Build Executor.

Build Executor



In Jenkins, a "build executor" acts like a worker that performs tasks on your behalf. Think of it as a pair of hands that can handle tasks such as building software, running tests, and generating reports. Instead of managing all these tasks manually, you can rely on Jenkins and its "build executors" to help you out.

- Task Management: Imagine you have a list of tasks, like building software projects and conducting tests.

- "Build Executors": Jenkins has these worker units known as "build executors." They're like assistants that can tackle various tasks concurrently.

- Task Allocation: When you request Jenkins to initiate a task, it checks if any "build executor" is available. If one is free, Jenkins assigns the task to that executor.

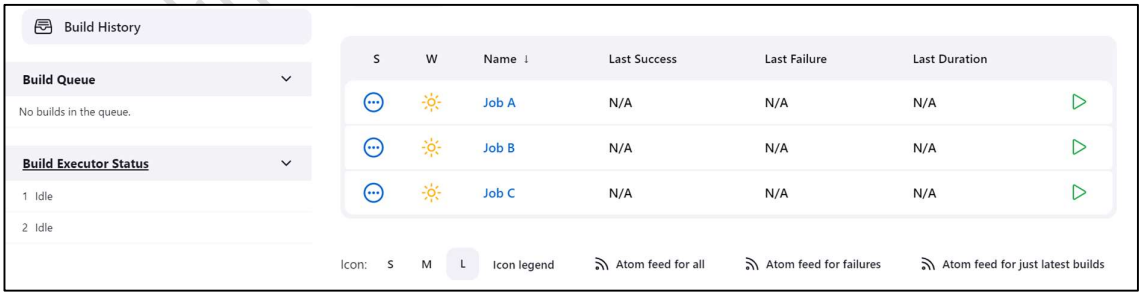
- Task Execution: The chosen "build executor" starts handling the task you assigned. This could involve actions like compiling code, running tests, or generating reports.

- Parallel Efficiency: If you have multiple "build executors," they can concurrently manage distinct tasks. This parallel approach speeds up the process since tasks are handled simultaneously.

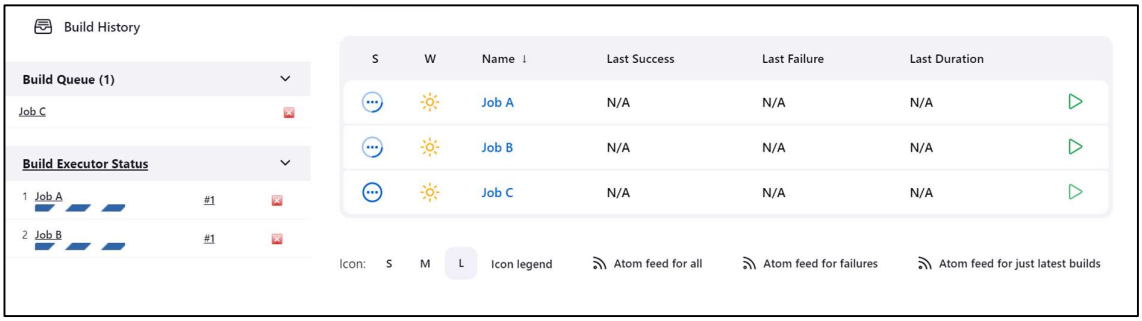
- Feedback Loop: After completing the task, the "build executor" sends feedback to Jenkins regarding the outcome. This could be confirmation of successful execution, or any issues encountered.

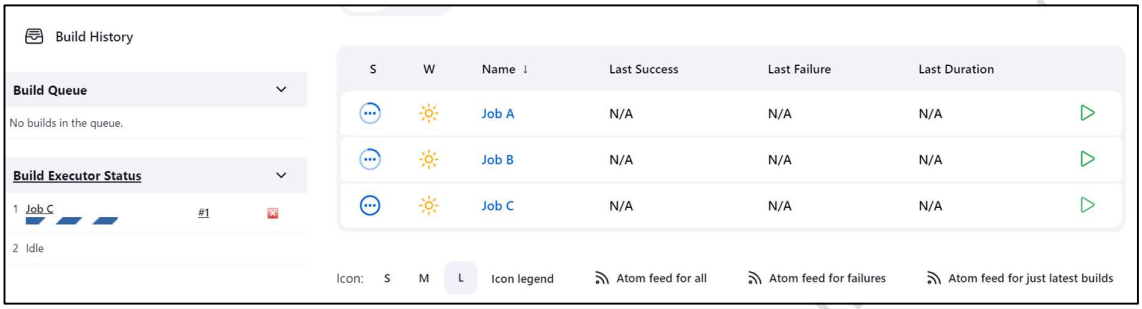
If we have two Build Executor and at that time, if we try to run 3 jobs at a time then, the last job will go into the build Queue.

In the below example, I am going to run 3 Freestyle jobs where I am going to run the sleep command. If you run the command sleep 5, it means that the execution will pause for 5 seconds before moving on to the next instruction. So, I am running the sleep command for 1 Minute.

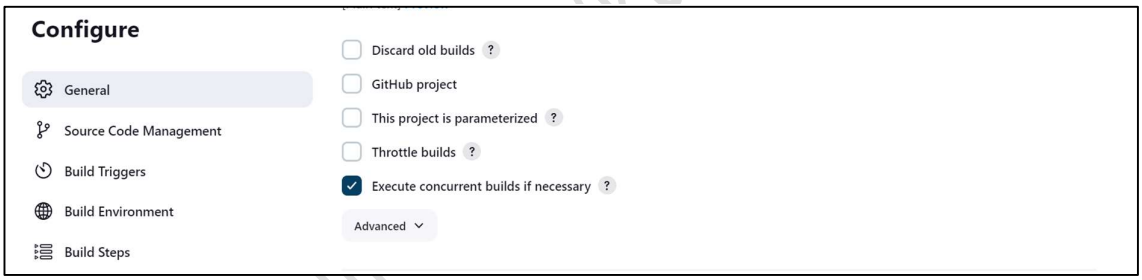


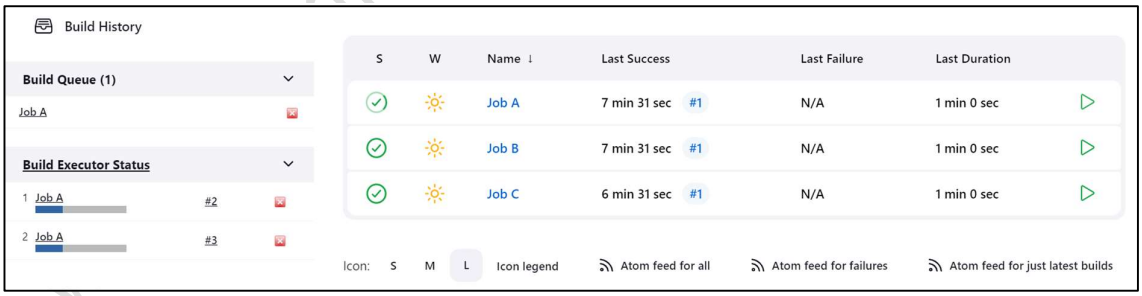
I have created Three Jobs A, B and C. and all Job I am going to build now by using ▶ sign. Here two Job will start running and one Job will go in the Build queue. And as soon as the Job is finished then Job C will start running. You can see the below diagram for that.





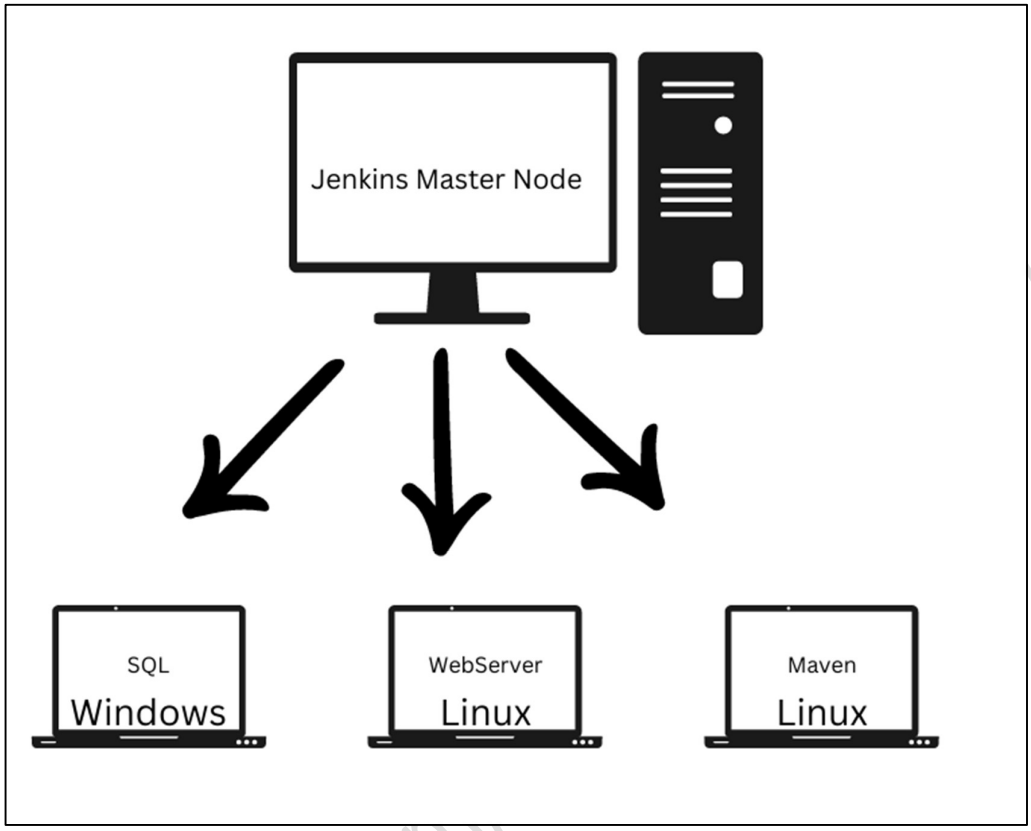
If we have a requirement to run the Job A 2 times which is not allowed by default. For that we need to mark on **Execute concurrent builds if necessary,** from the General Bar. Then click on Save -> Build





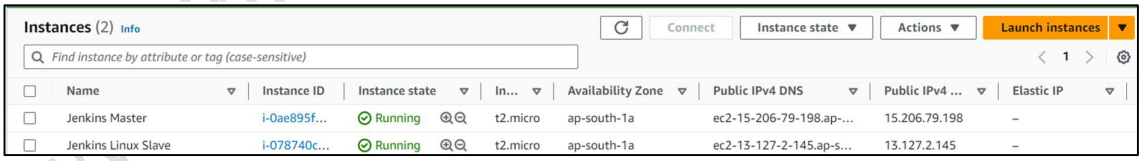
Here Job A run parallelly 3 Times.

Now Let's Move on How we can create the Setup for Master Slave Architecture on Jenkins. In this Practical, I am going to Create Linux OS and windows OS as Slave manually and later I will explain about How to create Dynamic Slave.



For Creating Linux as slave Node, we need to launch 1 OS. Here I am Using AWS Cloud for launching the Linux OS, and For using this OS as slave node, we need perform several steps.

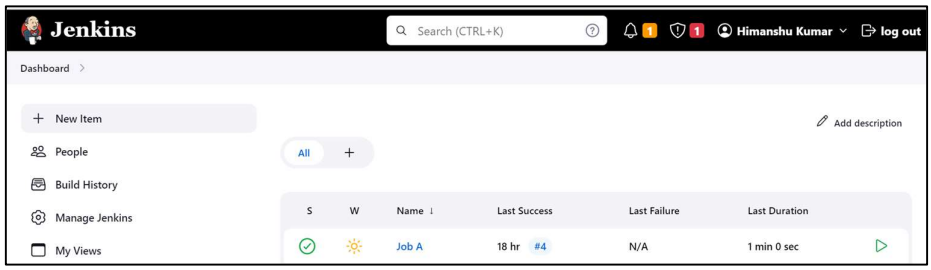
- It requires Internet Connectivity so it can connect to master by using User & Password.
- Require on Agent Program which helps to connect with master.
- Run the Agent Program means register it.



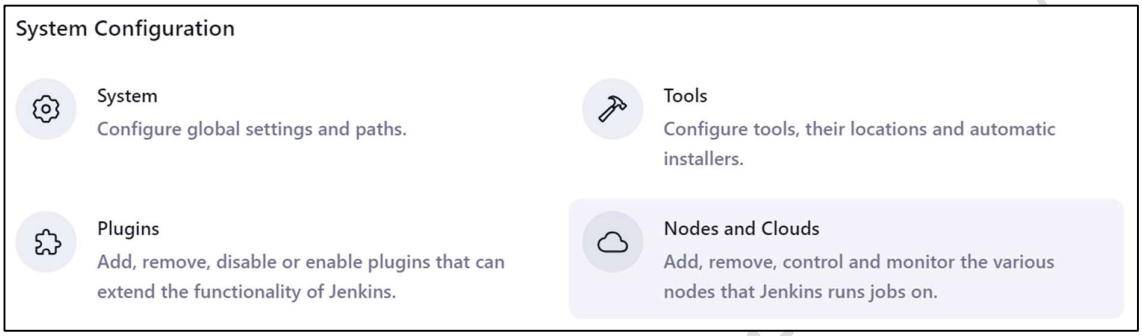
After Creating Linux Slave, In Jenkins Host, we need to Install the Java Software in the Slave node.

Login inside the Slave OS: Run Command: **sudo yum install java-11 -y**

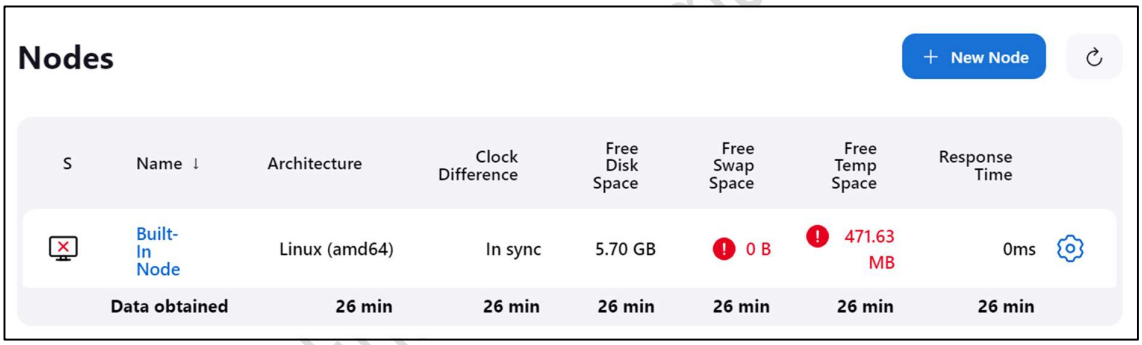
That's all. Now I am going to Enter the Information about the Slave in my master Node.

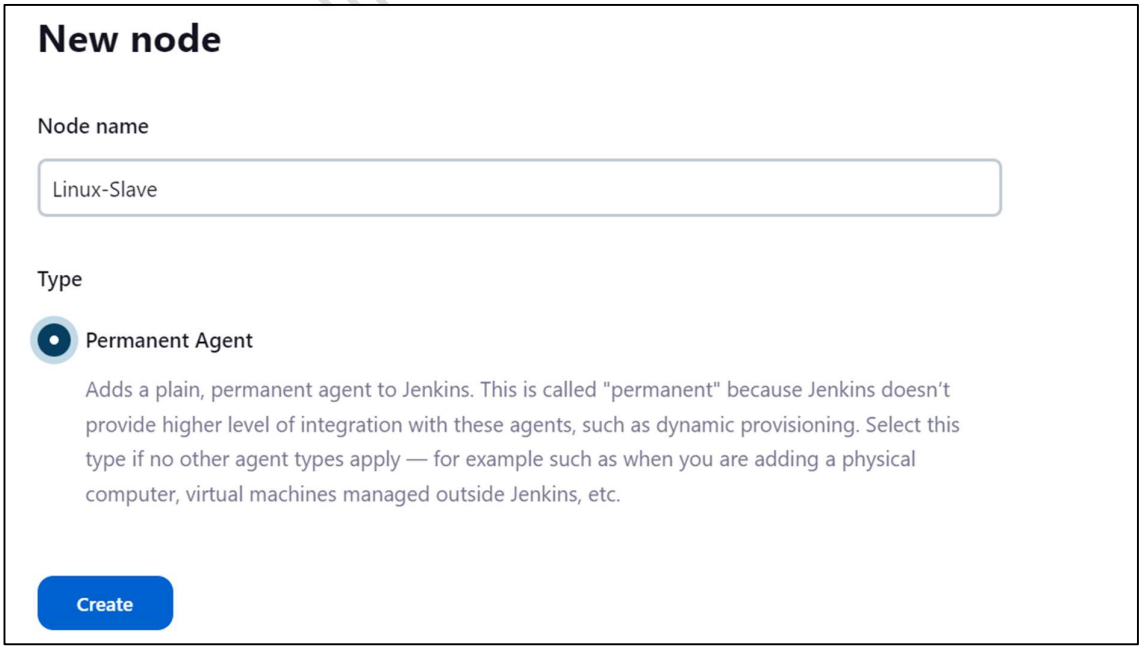


Click on Manage Jenkins, then click on Nodes and Cloud.



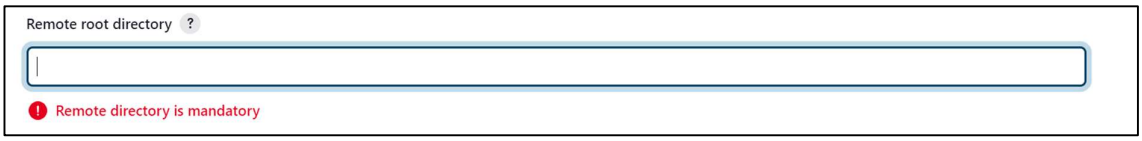
Click on New Node



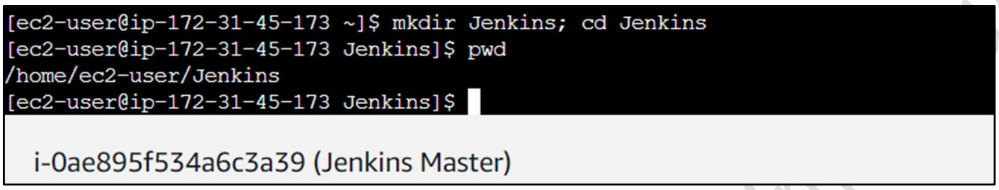


Enter the Slave Node Name and I am going to make this slave as permanent, so I am selecting as Permanent Agent.

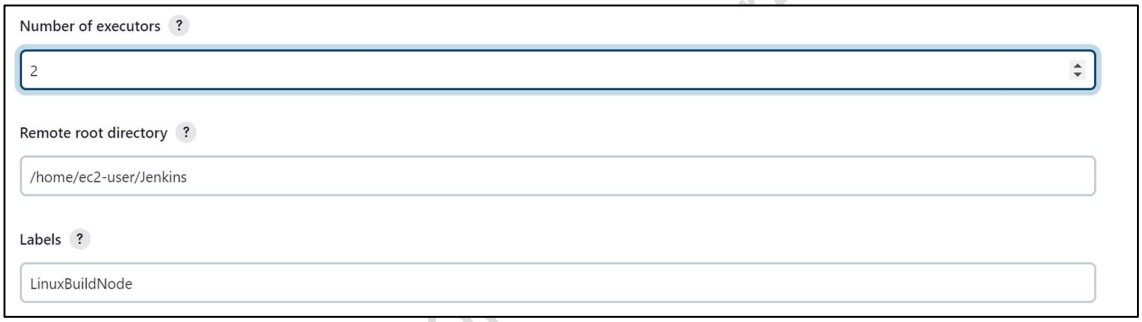
Here In the Configuration, it will ask for root Directory, which we need to configure from Master node.



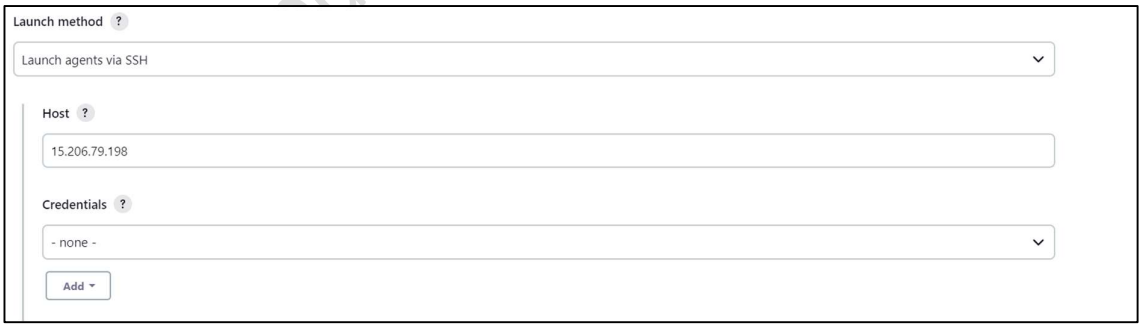
Here I am going to create the Empty directory in master Node with the name of Jenkins.



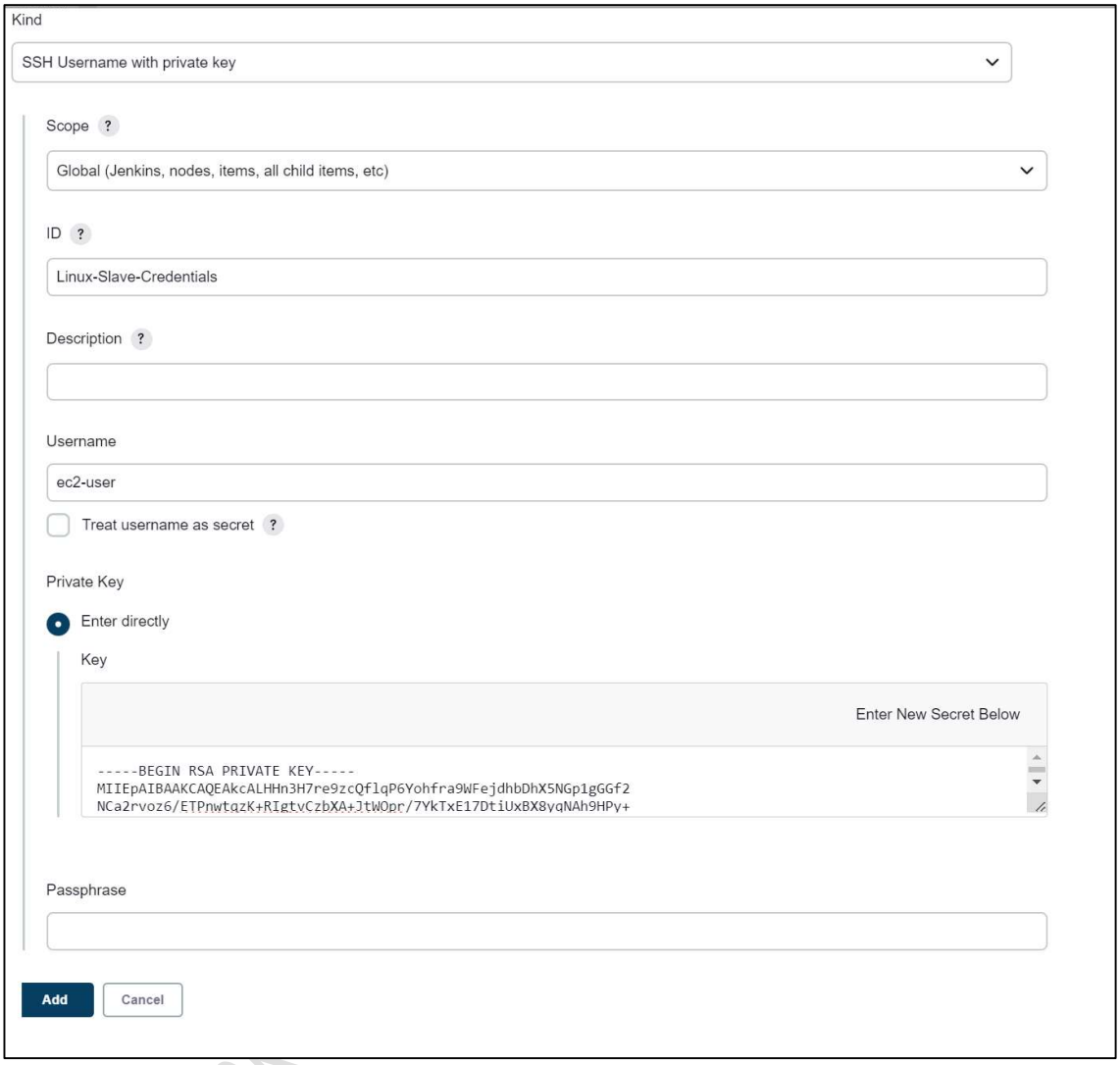
Then Copy the path of Jenkins in the Remote root directory. And I have also set the total executor I need as 2, and the Label name for the current Node as JenkinsBuildNode.



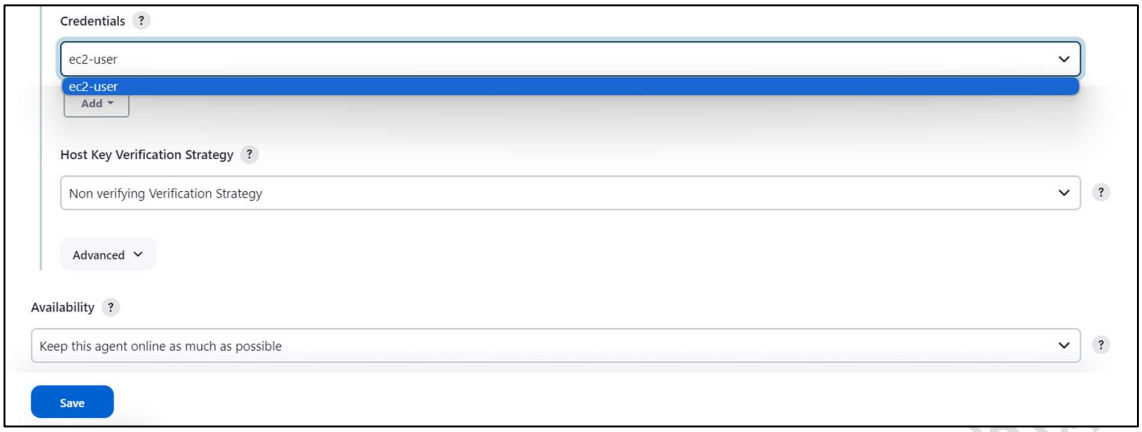
In the Launch Method, Select Launch Agents via SSH then enter the IP address of the agent in HOST. In the Credentials we need to add the username and password for the agent, But Currently I Don't have that. For that I am going to click on Add Button, then Selecting the Jenkins.



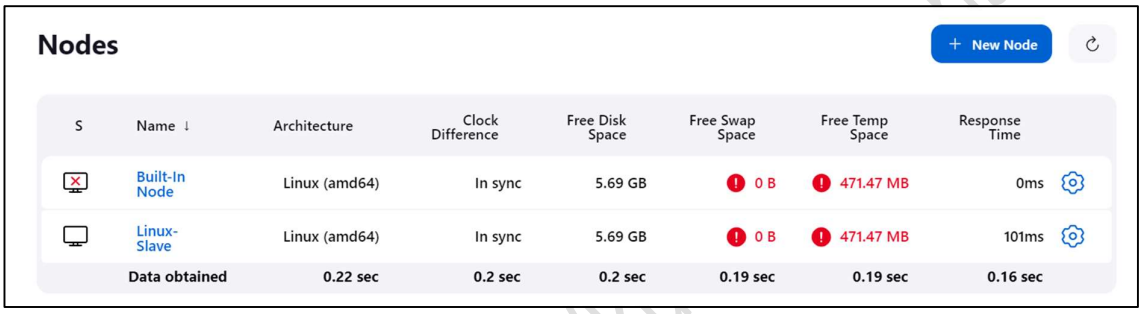
1. Select from Kind: SSH Username and private key. In AWS, by default we are connecting with the Username and private key for that reason here I have selected this option.
2. I provided the ID as Linux-Slave-Credentials.
3. I mentioned the username as ec2-user.
4. I provided the Secret Key as a text format.
5. Click on Add Button.



After Saving the Credentials, click on the Credentials then you can see the credentials that I have created right now. And in Host Key Verification Strategy I have chosen Non-Verifying Verification Strategy. Because while connecting via SSH using username and secret-key. The user will get a prompt do you want to proceed with the connection, that time we need to enter Yes and no. as a user we can interact with the CLI, but Jenkins have no capability to enter the Yes and No from the keyboard. So, for this I have selected this option.

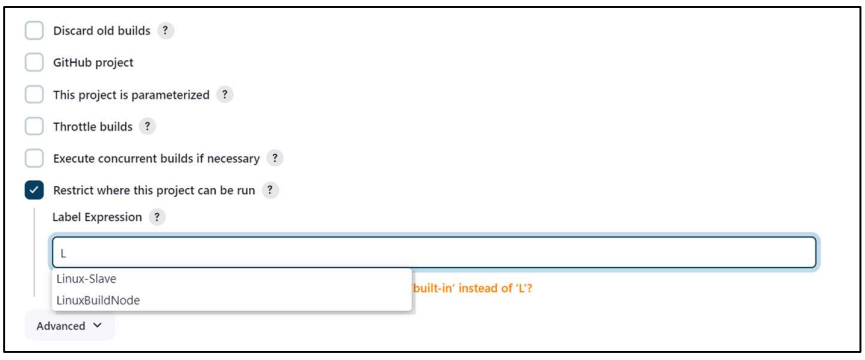


After that Click on Save Button, then we can see that Agent is Configured.

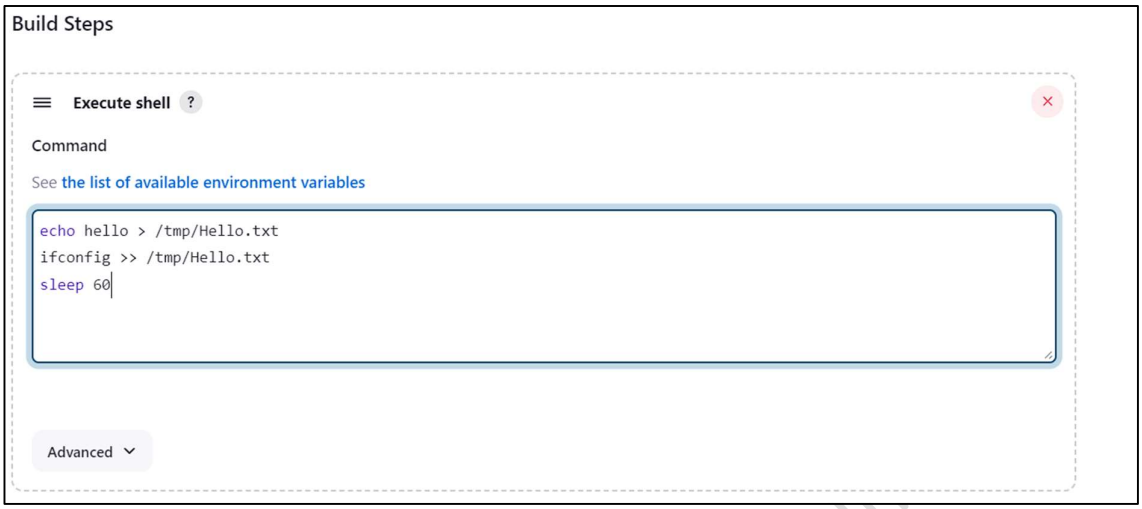


Now, I am going to run 1 Simple Job in Linux Slave, for that I created 1 Job with the Name of Slave Job A and this job is a Freestyle Project and then Click on OK.

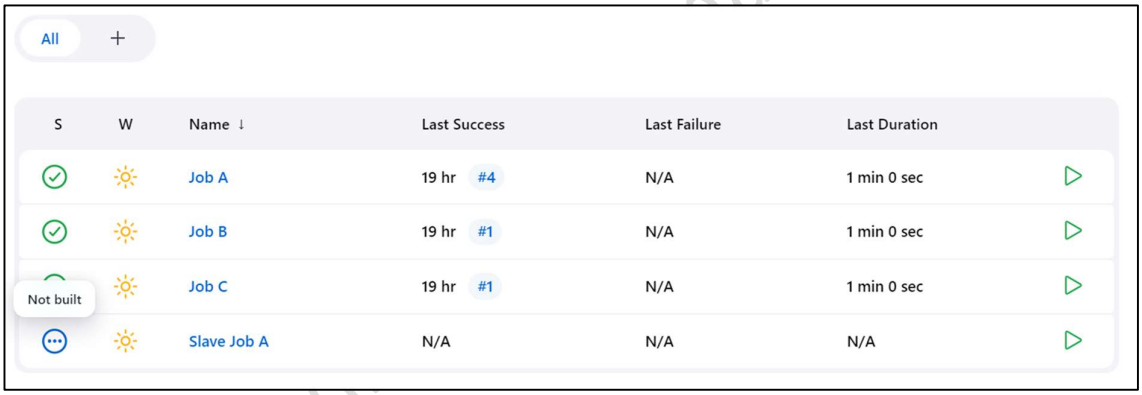
While Creating a Job we have a option where we can decide, where we would like to run our Job. For that we need to enable the option. **Restrict where this project can be run**.



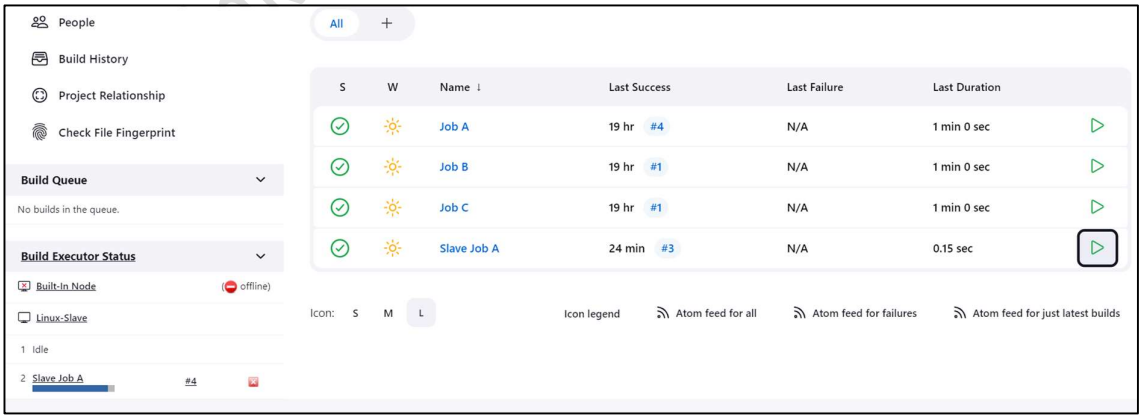
And while Creating the Slave as agent, we have given the slave a label name. so, we need to mention the Label name here and then It is good to go.



And in Build Steps I have mentioned some commands. Where I have written that I would like to print the output of hello and store in the /tmp/Hello.txt file and same I am doing with the ifconfig command and then I just run sleep command for 1 Minute.



Now I am going to run this JOB.



And we can see the progress of the job in left bottom corner.

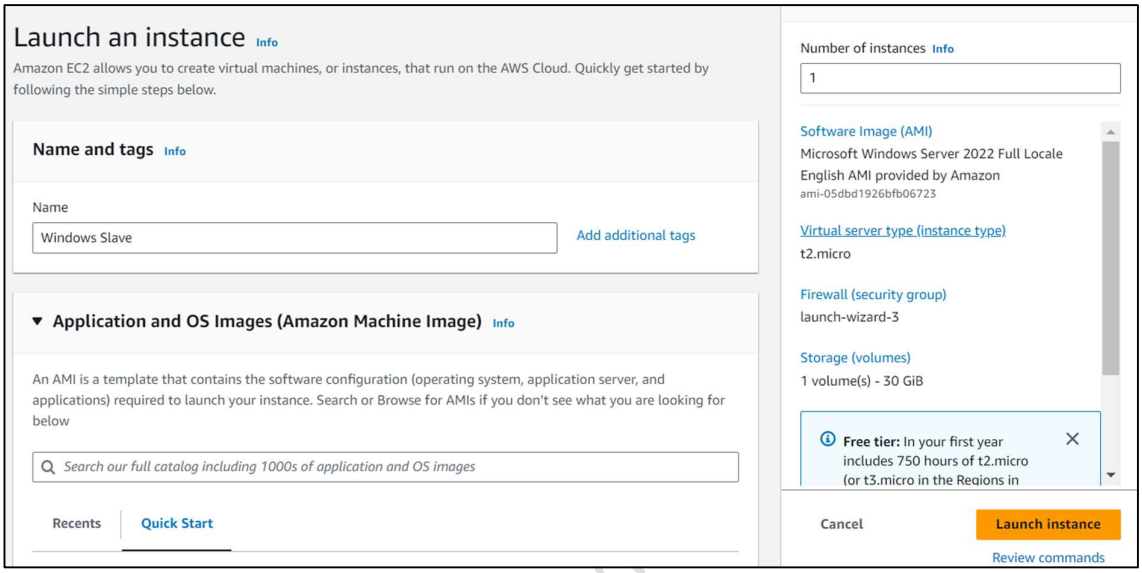
And the Job is successfully Run and we can see the console output.

Now I am going to Show about how we can Create Windows as Slave Node.

In Windows, SSH is not enabled. And we need to provide an agent in Windows OS. Then Windows OS will come to Jenkins and then Connect, and In Windows OS we need to install the Java Development Kit (JDK).

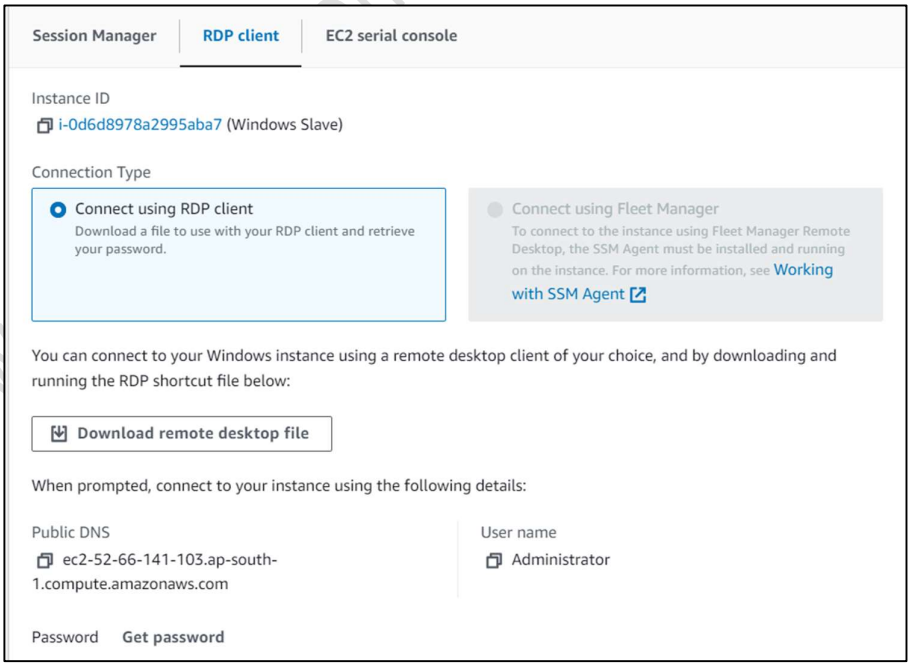
Windows Can Connect by Remote Desktop Protocol (RDP).

For this I am going to Create Microsoft Windows Server 2022 Locale English AMI provided.

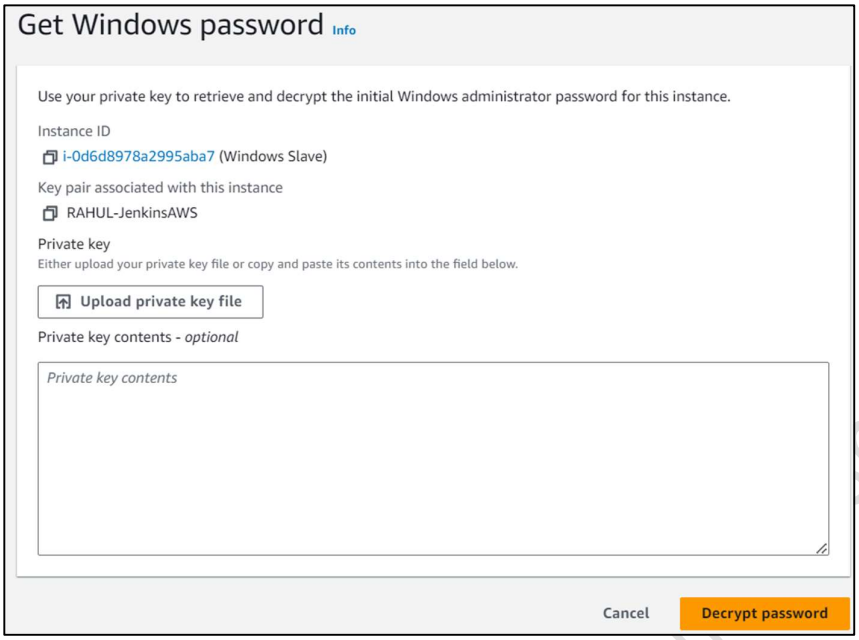


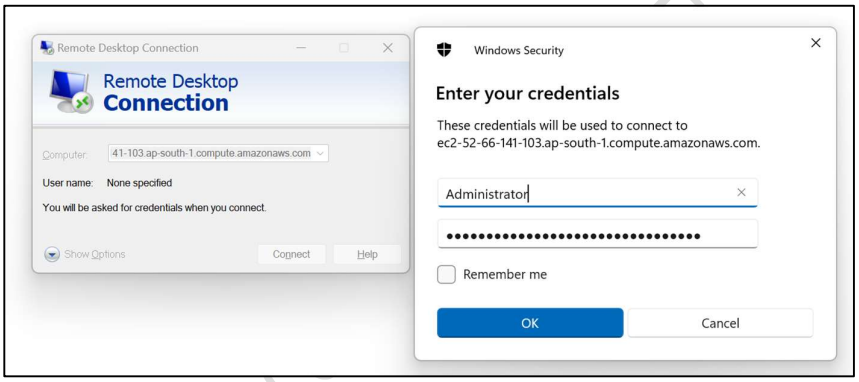
After Click on Launch Instance. Click on Connect to instance. Then click on RDP Client Tab.

Here I got Public DNS, Administrator, And Password.

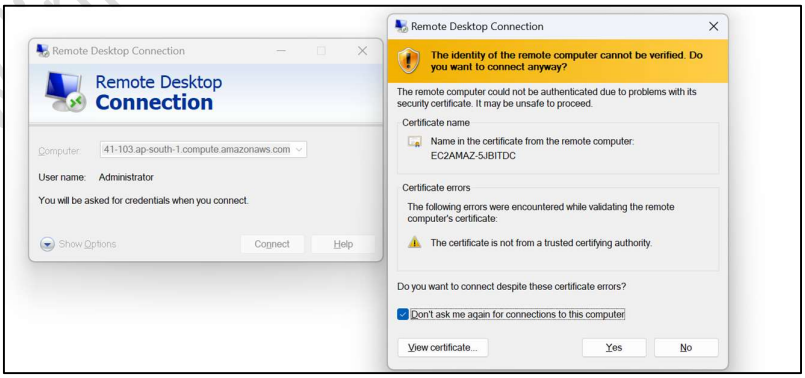


For Getting the Password we need to upload the our private Key file then we need to click on Decrypt password.

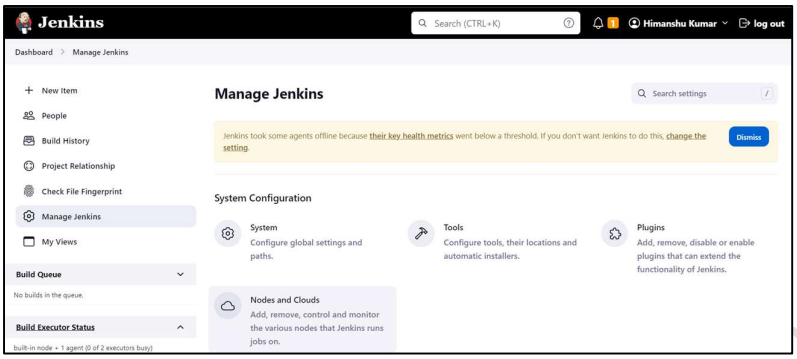


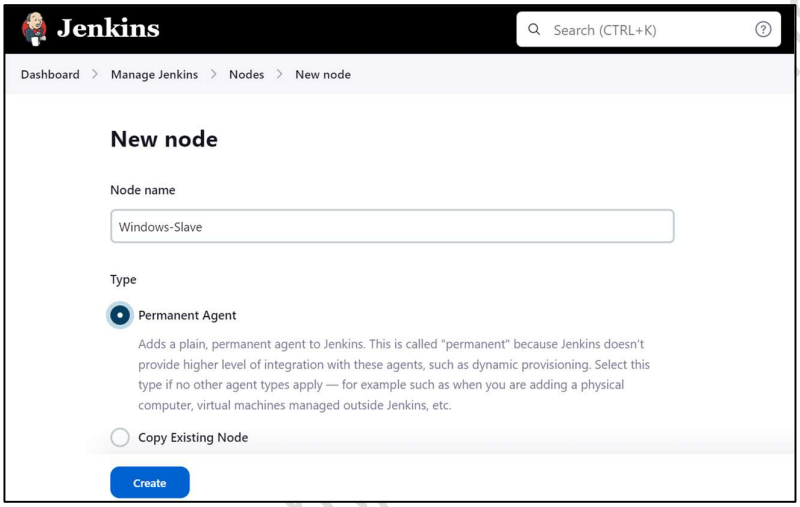


Then we need to Open the RDP. It is available on windows. Just search for RDP and on the computer. I provided my Public DNS in the Computer and then click on Connect and then Mention the Username as Administrator and enter the password and click on Ok, and then I allowed for the Connection. By clicking on that check box and press ok.

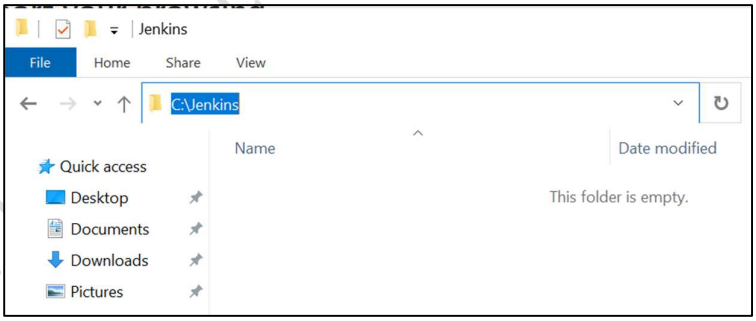


Now In Jenkins Master, I am going to Create the Slave Node. From Jenkins Dashboard. Click on Manage Jenkins, Go to Node and Clouds. Then click on New Node.

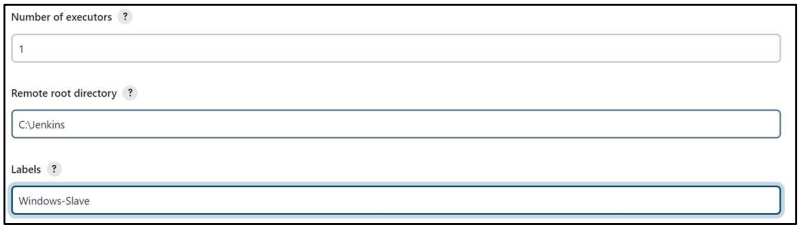




Enter the Node Name and Agent Type. Then click on Create. And we need to Remote root directory.



For this I have created an Empty directory with the name of Jenkins in my remote windows OS. and copied the path.

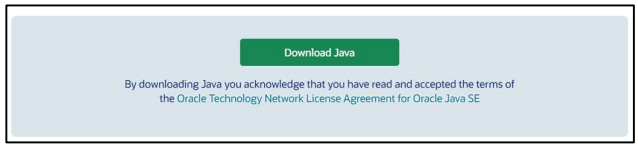


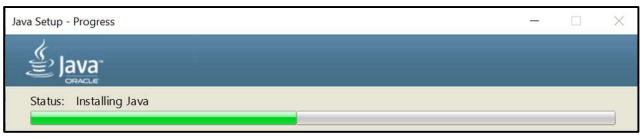
After that I mentioned the Label name as Windows-Slave. And for Launch Method I have selected **Launch agent by connecting it to the controller.** And Enabled Use WebSocket. Then Save it.

Here the Node is available, but it is offline.

Now click on Windows-Slave. Copy the Command and paste it into the CMD terminal in remote Windows OS.

For running that command, I need to install Java in my OS**. If this will not work, then try to install Java 11.**





```
C:\Users\Administrator\Downloads>java -version
java version "1.8.0_381"
Java(TM) SE Runtime Environment (build 1.8.0_381-b09)
Java HotSpot(TM) Client VM (build 25.381-b09, mixed mode, sharing)
```

Now JAVA is installed in my OS. Now I run this command and my agent is online.

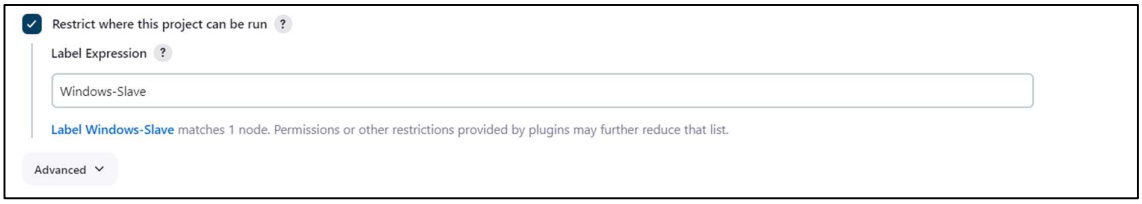
```
C:\Users\Administrator\Downloads>curl.exe -sO http://15.206.79.198:8080/jnlpJars/agent.jar

C:\Users\Administrator\Downloads>java -jar agent.jar -jnlpUrl http://15.206.79.198:8080/computer/Windows%2DSlave/jenkins
-agent.jnlp -secret a3fafc49ca5be88dd3ec75b86ebff2821b764920a189607ffa67a7dbe322520e -workDir "C:\Jenkins"
Aug 22, 2023 8:03:51 AM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using C:\Jenkins\remoting as a remoting work directory
Aug 22, 2023 8:03:51 AM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to C:\Jenkins\remoting
Aug 22, 2023 8:03:51 AM hudson.remoting.jnlp.Main createEngine
INFO: Setting up agent: Windows-Slave
Aug 22, 2023 8:03:51 AM hudson.remoting.Engine startEngine
INFO: Using Remoting version: 3107.v665000b_51092
Aug 22, 2023 8:03:51 AM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using C:\Jenkins\remoting as a remoting work directory
Aug 22, 2023 8:03:52 AM hudson.remoting.jnlp.Main$CuiListener status
INFO: WebSocket connection open
Aug 22, 2023 8:03:52 AM hudson.remoting.jnlp.Main$CuiListener status
INFO: Connected
```

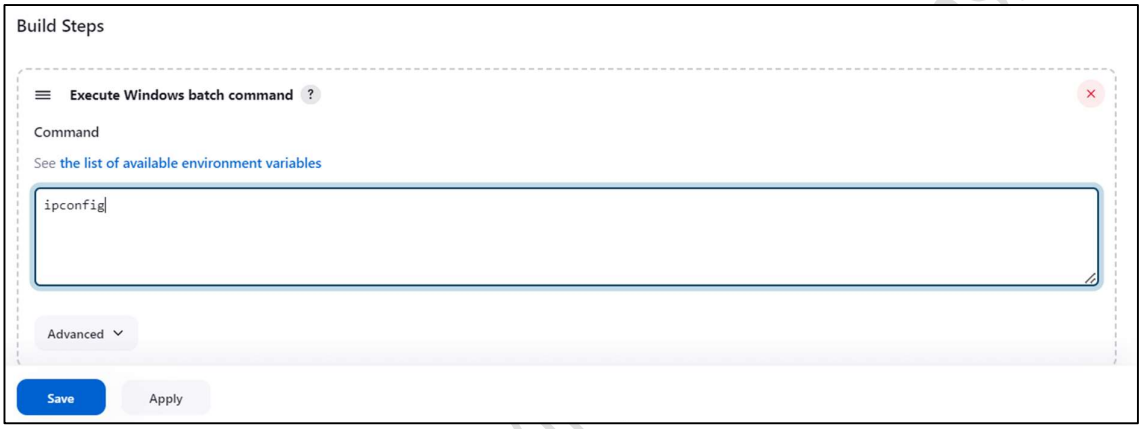
Click on Refresh Icon and we can see that agent is up. Now I am going to Execute any windows command like ipconfig.

Create a New Job and Give the Job Name and select the Freestyle project. And click on OK.

Restrict this project so it can only run in Windows Slave.



For Executing Windows command, we need to select Execute Windows batch Command.



After writing the command I saved my Job. Now I am going to build this Job.

Job Started in Windows Slave

Now We, can see the Console Output By clicking on that Job.

So, by using this way we can Create the Master Slave Architecture in Jenkins.

For Linux System Only: If you are using any Cloud and for some reason if you shut down your Machine, then From the Next time Your Public IP will Change. And you need to replace the new IP with the previous one, otherwise the things will not work.

Go to Manage Jenkins > Nodes and Clouds

Click on setting Icon for the Linux System.

Change the Public IP. Then All things will work. Otherwise, if you don't do this then Jenkins will try to connect with the Previous IP and that system will not exist. So, every time this will fail.