

Windows Slaves adding:

Step1:

Open jenkins dashboard and select manage jenkins and in that select configure and select manage nodes and add nodes

Give these details:

Name: windows

Description: dfdferf rferfef

No.of executors: 1

Remote Root directory: F:\Jenkins (we will give this path after creating jenkins folder in F drive of remote server)

Labels: windowsslave (using this label we will select our node in pipeline and free style proj)

Usage: only build jobs with label expressions matching this node (select this button)

Launch method: Launch agent by connecting it to controller (select this button)

Select Use websocket yes in button of internal data dir.

Save

← → ↻ Not secure | 192.168.10.51:8080/manage/computer/windows/configure 🔍 📄 ☆ 🗑️ 👤 ⋮

Dashboard > Manage Jenkins > Nodes > windows > Configure

Status

Delete Agent

Configure

Build History

Load Statistics

Script Console

Log

System Information

Disconnect

Build Executor Status

1 Idle

Name ?

windows

Description ?

windows node production server

[Plain text] Preview

Number of executors ?

1

Remote root directory ?

F:\Jenkins

Labels ?

windowslave

Usage ?

Only build jobs with label expressions matching this node

Save

← → ↻ Not secure | 192.168.10.51:8080/manage/computer/windows/configure 🔍 📄 ☆ 🗑️ 👤 ⋮

Dashboard > Manage Jenkins > Nodes > windows > Configure

Usage ?

Only build jobs with label expressions matching this node

Launch method ?

Launch agent by connecting it to the controller

☐ Disable WorkDir ?

Custom WorkDir path ?

Internal data directory ?

remoting

☐ Fail if workspace is missing ?

☒ Use WebSocket ?

Advanced

Availability ?

Keep this agent online as much as possible

Save

We will get X in nodes for the windows. Click on windows in nodes and we will get Commands.

Open Jenkins dashboard in windows server and copy that command and open it in admin mode command line of Windows server. And paste the first command link there. We will connect successfully. check nodes the node is connected

Install same versions of java and git in windows slave. And give paths of both in environmental variables system.

Check git and java versions from command line of windows server and if success it works fine. Give SSL certificate ignore commands in command line of windows server (slave).

Sudo apt-get install fontconfig openjdk-11-jre

=====

[Jenkins Master Slave Configuration on Windows | by Divya Bodduluri | Medium](#)

[] Open Jenkins dashboard in windows server.

[] open the cmd move to jenkins folder past the first line command of windows server

[] Download the latest WinSW-x64.exe from this link [Releases · winsw/winsw \(github.com\)](#), rename it as jenkins-agent.exe(can be any desired name) and place it in the Jenkins folder.

[] Next, download the jenkins.xml file from [winsw/samples/jenkins.xml at v3 · winsw/winsw \(github.com\)](#), rename it as jenkins-agent.xml and place it in the Jenkins folder.

> Documents > Jenkins >

<input type="checkbox"/> Name	Date modified	Type	Size
remoting	8/31/2023 12:14 AM	File folder	
agent.jar	8/31/2023 12:14 AM	Executable Jar File	1,339 KB
jenkins-agent.exe	8/31/2023 12:34 AM	Application	17,859 KB
jenkins-agent.xml	8/31/2023 12:36 AM	XML File	1 KB

Note that the name of the exe and xml file should be the same.

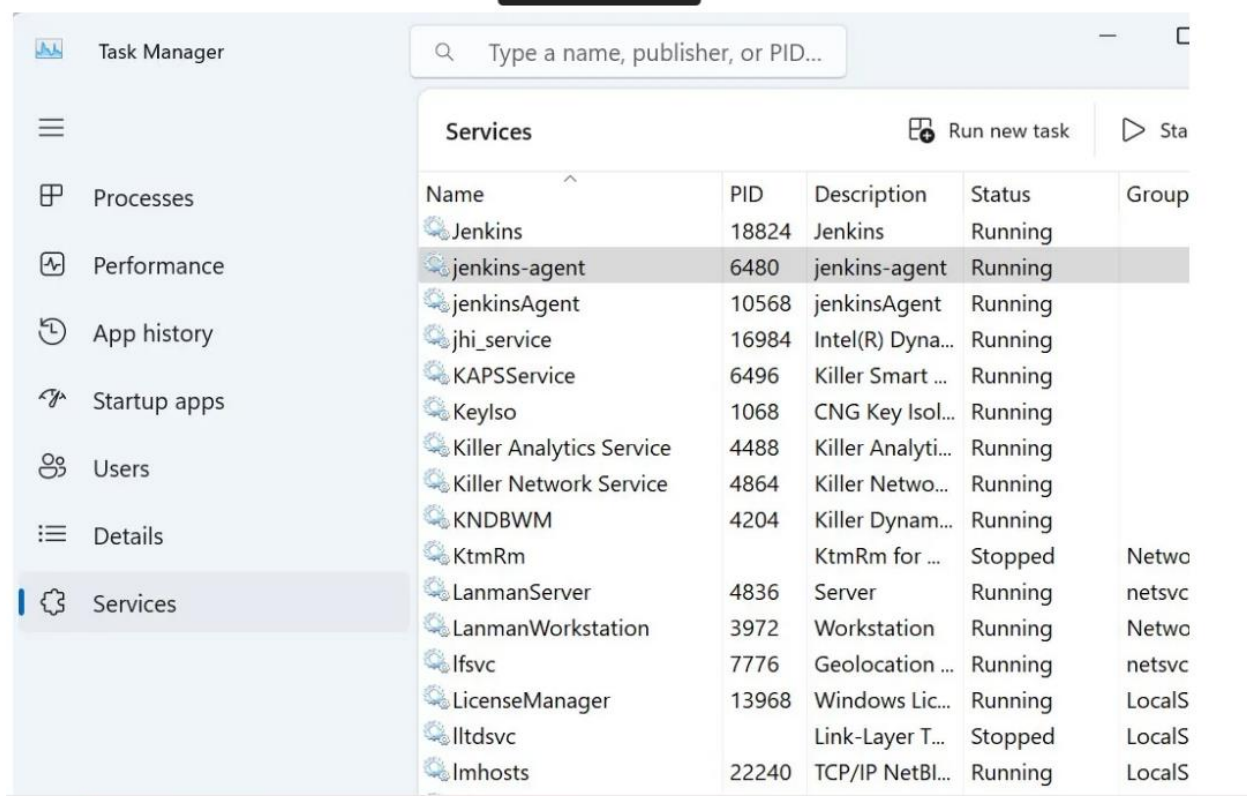
Now open the jenkins-agent.xml in edit mode. Change the value of id and name tag to jenkins-agent. Delete the value of arguments tag and replace the it with the command that is displayed in your node page(the second command that was run in [Picture 3](#), just remove the first word java and paste everything else as it is)

```
jenkins-agent.xml
File Edit View

<service>
  <id>jenkins-agent</id>
  <name>Jenkins-agent</name>
  <description>This service runs Jenkins continuous integration system.</description>
  <env name="JENKINS_HOME" value="%BASE%" />
  <executable>java</executable>
  <arguments>-jar agent.jar -jnlpUrl http://[redacted]:8080/computer/XPS/jenkins-agent.jnlp -secret
2fc81b[redacted]7a55a32bc6cd9fdaf69d6304d0c818f5f0 -workDir "C:\Users\de11\Documents\Jenkins"</arguments>
  <log mode="roll" />
  <onfailure action="restart" />
</service>
```

Now open cmd prompt as an admin, cd to your Jenkins folder path and enter the command: `jenkins-agent.exe install`. Now the Jenkins-agent is installed as a service successfully.

Open services tab in task manager, search for the jenkins-agent service, right click on it and click start.



Once this agent is up and running, you can refresh the Jenkins UI and tadaaaaaa! the slave is connected.

Status

Delete Agent

Configure

Build History

Load Statistics

Script Console

Log

System Information

Disconnect

Build Executor Status

1 Idle

Agent windows

windows node production server

Agent is connected.


Labels

windowsslave

Projects tied to windows

None

← → ↺ Not secure | 192.168.10.51:8080/manage/computer/

 **Jenkins**

Search (CTRL+K)

🔔 🚫 🛑 Pradeep Nadimpalli log out

Dashboard > Manage Jenkins > Nodes >

Configure Clouds

Node Monitoring

Build Queue

Build Executor Status

17 min

Manage nodes and clouds

New Node

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-in Node	Linux (amd64)	In sync	435.55 GB	2.00 GB	435.55 GB	0ms
	windows	Windows Server 2019 (amd64)	In sync	1692.06 GB	180.10 GB	352.44 GB	494ms
	Data obtained	17 min	17 min	17 min	17 min	17 min	17 min

1 Idle

2 Idle

windows

1 Idle

192.168.10.51:8080/manage/computer/#

REST API Jenkins 2.387.1

Create Freestyle Project:

Select git and give git repo url and credentials

Select Restrict where this project can be run

Labelname : windowsslave

This is the labelname we have given to our windows slave

Save

Build job

Job build successfully to slave (windowsserver)

Check F drive /jenkins /workspace/abc

Successfully clone git to slave server

Finished: SUCCESS

Slave server repo path where we clone git files



Pipeline-code:

Select pipeline while creating the job

Pipeline code:

```
pipeline {  
  
    agent {label 'windowsslave'}  
  
    stages {  
  
        stage('Git checkout') {  
  
            steps {  
  
                git branch: 'main', credentialsId: 'PAT', url: 'https://192.168.10.58/pradeepkumar.n/testing'  
  
            }  
  
        }  
  
    }  
  
}
```

Save

Build job and we will get success and check in node server for output.

/var/lib/jenkins/workspace/Window_Slave_pipeline@tmp/jfrog/2/.jfrog deleted
Git checkout: SUCCESS

Windows slave to custom workspace declarative pipeline:

```
pipeline {  
  
    agent {  
        node{  
            label 'windowsslave'  
            customWorkspace 'F:/new11/new22'  
        }  
    }  
  
    stages {  
  
        stage('Git checkout') {  
  
            steps {  
  
                git branch: 'main', credentialsId: 'PAT', url: 'https://192.168.10.58/pradeepkumar.n/testing'  
  
            }  
  
        }  
  
    }  
}
```

}