

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

Build Executor Status	
1	Idle
2	Idle

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.

Build History	
Build Queue	
No builds in the queue.	
Build Executor Status	
1 Idle	
2 Idle	

S	W	Name	Last Success	Last Failure	Last Duration
...	☀	Job A	N/A	N/A	N/A
...	☀	Job B	N/A	N/A	N/A
...	☀	Job C	N/A	N/A	N/A

Icon: S M L Icon legend Atom feed for all Atom feed for failures Atom feed for just latest builds

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.

Build History

Build Queue (1)

Job C

Build Executor Status

1 Job A #1

2 Job B #1

S	W	Name ↓	Last Success	Last Failure	Last Duration	
...	☀	Job A	N/A	N/A	N/A	▶
...	☀	Job B	N/A	N/A	N/A	▶
...	☀	Job C	N/A	N/A	N/A	▶

Icon: S M L

Icon legend

Atom feed for all

Atom feed for failures

Atom feed for just latest builds

Build History

Build Queue

No builds in the queue.

Build Executor Status

1 Job C #1

2 Idle

S	W	Name ↓	Last Success	Last Failure	Last Duration	
...	☀	Job A	N/A	N/A	N/A	▶
...	☀	Job B	N/A	N/A	N/A	▶
...	☀	Job C	N/A	N/A	N/A	▶

Icon: S M L

Icon legend

Atom feed for all

Atom feed for failures

Atom feed for just latest builds

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

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

☐ Discard old builds ?

☐ GitHub project

☐ This project is parameterized ?

☐ Throttle builds ?

☒ Execute concurrent builds if necessary ?

Advanced

▼

Build History

Build Queue (1)

Job A

Build Executor Status

1 Job A #2

2 Job A #3

S	W	Name ↓	Last Success	Last Failure	Last Duration	
✓	☀	Job A	7 min 31 sec #1	N/A	1 min 0 sec	▶
✓	☀	Job B	7 min 31 sec #1	N/A	1 min 0 sec	▶
✓	☀	Job C	6 min 31 sec #1	N/A	1 min 0 sec	▶

Icon: S M L

Icon legend

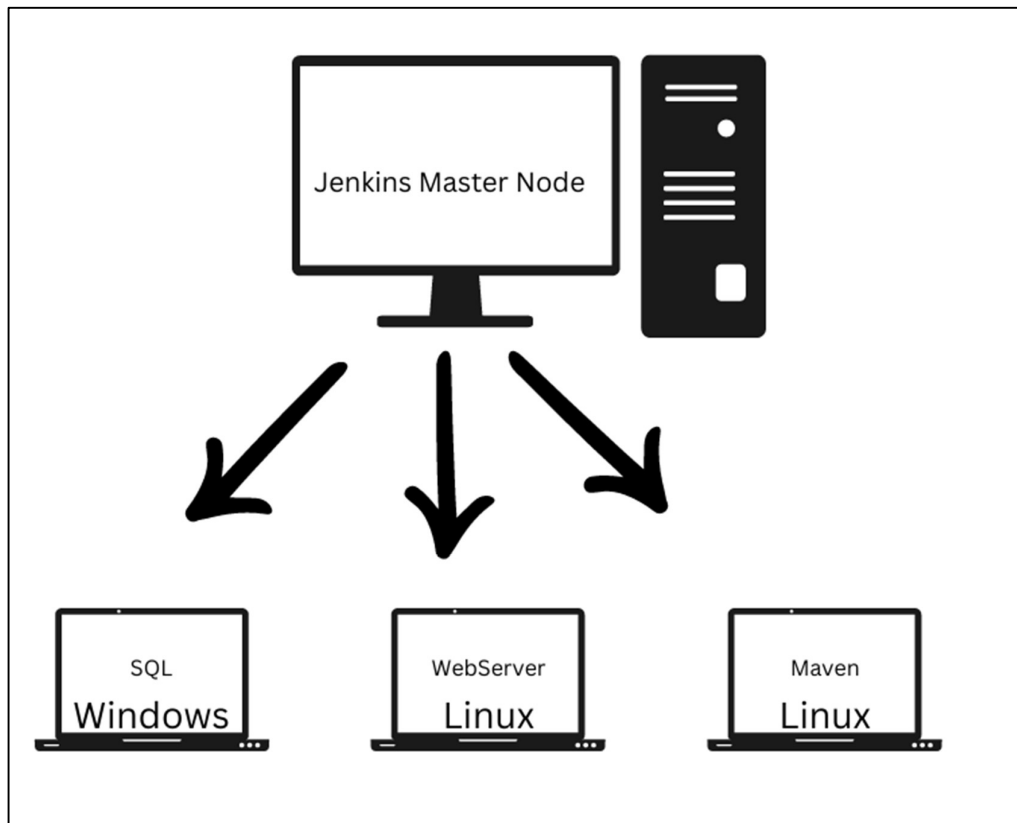
Atom feed for all

Atom feed for failures

Atom feed for just latest builds

Here Job A run parallely 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.

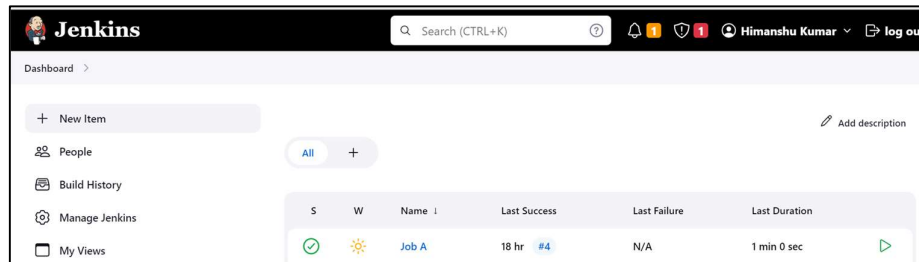
- It requires Internet Connectivity so it can connect to master by using User & Password.
- Require on Agent Program which helps to connect with master.
- Run the Agent Program means register it.

Instances (2) Info									
Find instance by attribute or tag (case-sensitive)									
<input type="checkbox"/>	Name	Instance ID	Instance state	In...	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	
<input type="checkbox"/>	Jenkins Master	i-0ae895f...	Running	t2.micro	ap-south-1a	ec2-15-206-79-198.ap-...	15.206.79.198	-	
<input type="checkbox"/>	Jenkins Linux Slave	i-078740c...	Running	t2.micro	ap-south-1a	ec2-13-127-2-145.ap-s...	13.127.2.145	-	

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.

System Configuration

System
Configure global settings and paths.

Tools
Configure tools, their locations and automatic installers.

Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.

Nodes and Clouds
Add, remove, control and monitor the various nodes that Jenkins runs jobs on.

Click on New Node

Nodes

[+ 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	5.70 GB	0 B	471.63 MB	0ms
Data obtained		26 min	26 min	26 min	26 min	26 min	26 min

New node

Node name

Linux-Slave

Type

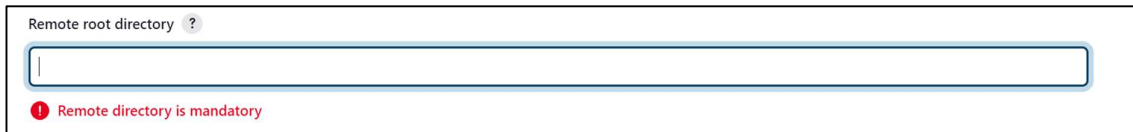
☒ Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.

[Create](#)

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.



Remote root directory ?

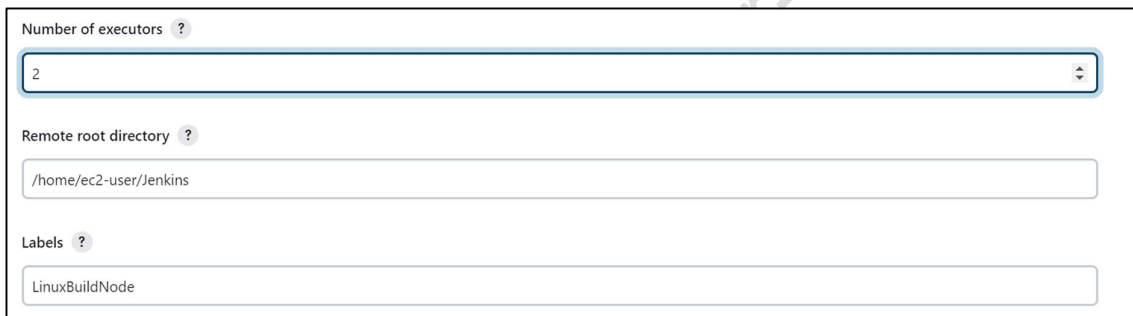
Remote directory is mandatory

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

```
[ec2-user@ip-172-31-45-173 ~]$ mkdir Jenkins; cd Jenkins
[ec2-user@ip-172-31-45-173 Jenkins]$ pwd
/home/ec2-user/Jenkins
[ec2-user@ip-172-31-45-173 Jenkins]$
```

i-0ae895f534a6c3a39 (Jenkins Master)

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.



Number of executors ?

2

Remote root directory ?

/home/ec2-user/Jenkins

Labels ?

LinuxBuildNode

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.



Launch method ?

Launch agents via SSH

Host ?

15.206.79.198

Credentials ?

- none -

Add

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.

Kind

SSH Username with private key

Scope ?

Global (Jenkins, nodes, items, all child items, etc)

ID ?

Linux-Slave-Credentials

Description ?

Username

ec2-user

☐ Treat username as secret ?

Private Key

☒ Enter directly

Key

Enter New Secret Below

```

-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAKcALHh3H7re9zcQf1qP6Yohfra9WFejdhbDhX5NGp1gGGf2
NCa2rvoz6/ETPnwtqzK+RIgtvCzbXA+JtW0pr/7YkTxE17DtIUx8X8ydNAh9HPy+

```

Passphrase

Add

Cancel

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.

Credentials ?

ec2-user

ec2-user

Add

Host Key Verification Strategy ?

Non verifying Verification Strategy

Advanced

Availability ?

Keep this agent online as much as possible

Save

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

Nodes + 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	5.69 GB	0 B	471.47 MB	0ms
	Linux-Slave	Linux (amd64)	In sync	5.69 GB	0 B	471.47 MB	101ms
Data obtained		0.22 sec	0.2 sec	0.2 sec	0.19 sec	0.19 sec	0.16 sec

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.

Enter an item name

Slave Job A

» Required field

Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

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.**

☐ Discard old builds ?

☐ GitHub project

☐ This project is parameterized ?

☐ Throttle builds ?

☐ Execute concurrent builds if necessary ?

☒ Restrict where this project can be run ?

Label Expression ?

L

Linux-Slave

LinuxBuildNode

built-in instead of 'L'?

Advanced

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.

Build Steps

Execute shell ?

Command

See [the list of available environment variables](#)

```
echo hello > /tmp/Hello.txt
ifconfig >> /tmp/Hello.txt
sleep 60
```

Advanced ▾

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.

All +

S	W	Name ↓	Last Success	Last Failure	Last Duration	
✓	☀	Job A	19 hr #4	N/A	1 min 0 sec	▶
✓	☀	Job B	19 hr #1	N/A	1 min 0 sec	▶
Not built	☀	Job C	19 hr #1	N/A	1 min 0 sec	▶
⋮	☀	Slave Job A	N/A	N/A	N/A	▶

Now I am going to run this JOB.

People

Build History

Project Relationship

Check File Fingerprint

Build Queue ▾
No builds in the queue.

Build Executor Status ▾
Built-In Node (offline)
Linux-Slave
1 Idle
2 Slave Job A #4

All +

S	W	Name ↓	Last Success	Last Failure	Last Duration	
✓	☀	Job A	19 hr #4	N/A	1 min 0 sec	▶
✓	☀	Job B	19 hr #1	N/A	1 min 0 sec	▶
✓	☀	Job C	19 hr #1	N/A	1 min 0 sec	▶
✓	☀	Slave Job A	24 min #3	N/A	0.15 sec	▶

Icon: S M L
Icon legend
Atom feed for all
Atom feed for failures
Atom feed for just latest builds

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

 **Console Output**

Progress:

```
Started by user Himanshu Kumar
Running as SYSTEM
Building remotely on Linux-Slave (LinuxBuildNode) in workspace /home/ec2-user/Jenkins/workspace/Slave Job A
[Slave Job A] $ /bin/sh -xe /tmp/jenkins6054008640525272537.sh
+ echo hello
+ ifconfig
+ sleep 60
```

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

 **Console Output**

```
Started by user Himanshu Kumar
Running as SYSTEM
Building remotely on Linux-Slave (LinuxBuildNode) in workspace /home/ec2-user/Jenkins/workspace/Slave Job A
[Slave Job A] $ /bin/sh -xe /tmp/jenkins6054008640525272537.sh
+ echo hello
+ ifconfig
+ sleep 60
Finished: SUCCESS
```

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.

Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

Name
Windows Slave [Add additional tags](#)

▼ Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

[Recents](#) [Quick Start](#)

Number of instances [Info](#)
1

Software Image (AMI)
Microsoft Windows Server 2022 Full Locale English AMI provided by Amazon
ami-05dbd1926bf06723

Virtual server type (instance type)
t2.micro

Firewall (security group)
launch-wizard-3

Storage (volumes)
1 volume(s) - 30 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in)

[Cancel](#) [Launch instance](#) [Review commands](#)

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

Here I got Public DNS, Administrator, And Password.

Session Manager **RDP client** **EC2 serial console**

Instance ID
i-0d6d8978a2995aba7 (Windows Slave)

Connection Type

☒ **Connect using RDP client**
Download a file to use with your RDP client and retrieve your password.

☐ **Connect using Fleet Manager**
To connect to the instance using Fleet Manager Remote Desktop, the SSM Agent must be installed and running on the instance. For more information, see [Working with SSM Agent](#)

You can connect to your Windows instance using a remote desktop client of your choice, and by downloading and running the RDP shortcut file below:

[Download remote desktop file](#)

When prompted, connect to your instance using the following details:

Public DNS
ec2-52-66-141-103.ap-south-1.compute.amazonaws.com

User name
Administrator

Password [Get password](#)

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

Get Windows password [Info](#)

Use your private key to retrieve and decrypt the initial Windows administrator password for this instance.

Instance ID
i-0d6d8978a2995aba7 (Windows Slave)

Key pair associated with this instance
RAHUL-JenkinsAWS

Private key
Either upload your private key file or copy and paste its contents into the field below.

Private key contents - optional

Private key contents

Remote Desktop Connection

Computer: 41-103.ap-south-1.compute.amazonaws.com

User name: None specified

You will be asked for credentials when you connect.

Windows Security

Enter your credentials

These credentials will be used to connect to ec2-52-66-141-103.ap-south-1.compute.amazonaws.com.

Administrator

.....

☐ Remember me

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.

Remote Desktop Connection

Computer: 41-103.ap-south-1.compute.amazonaws.com

User name: Administrator

You will be asked for credentials when you connect.

The identity of the remote computer cannot be verified. Do you want to connect anyway?

The remote computer could not be authenticated due to problems with its security certificate. It may be unsafe to proceed.

Certificate name

Name in the certificate from the remote computer: EC2AMAZ-SJB1TDC

Certificate errors

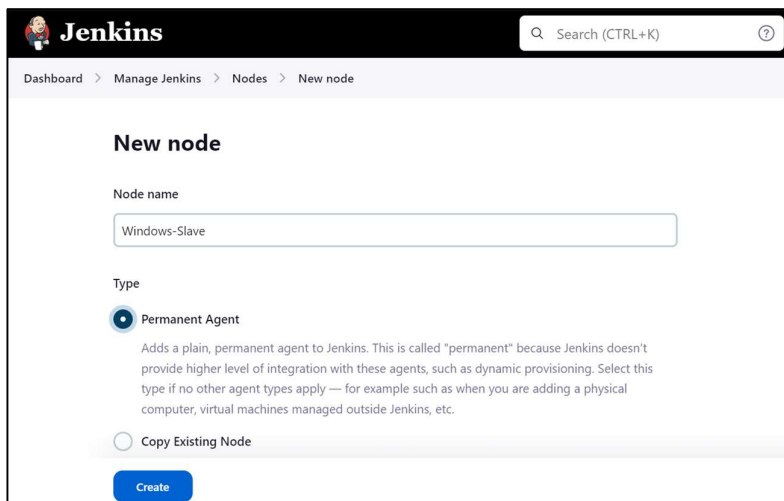
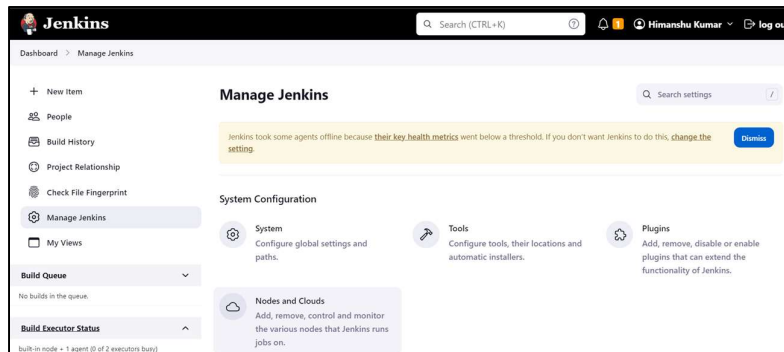
The following errors were encountered while validating the remote computer's certificate:

The certificate is not from a trusted certifying authority.

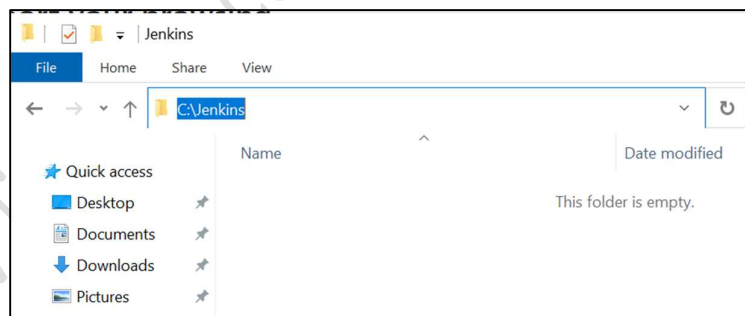
Do you want to connect despite these certificate errors?

☒ Don't ask me again for connections to this computer

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.

Number of executors ?
<input type="text" value="1"/>
Remote root directory ?
<input type="text" value="C:\Jenkins"/>
Labels ?
<input type="text" value="Windows-Slave"/>

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.



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 ?

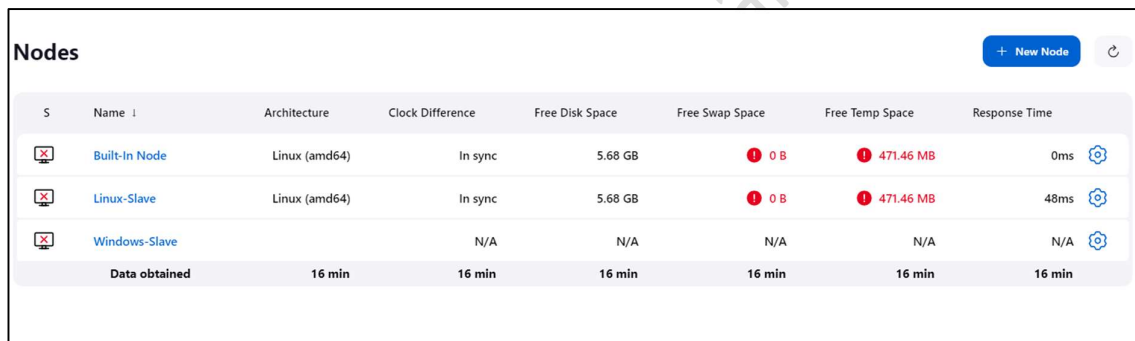
Advanced




Availability ?

Keep this agent online as much as possible

Save

Here the Node is available, but it is offline.



S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	5.68 GB	0 B	471.46 MB	0ms
	Linux-Slave	Linux (amd64)	In sync	5.68 GB	0 B	471.46 MB	48ms
	Windows-Slave		N/A	N/A	N/A	N/A	N/A
Data obtained		16 min	16 min	16 min	16 min	16 min	16 min

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



Run from agent command line: (Windows)

```
curl.exe -sO http://15.206.79.198:8080/jnlpJars/agent.jar
java -jar agent.jar -jnlpUrl http://15.206.79.198:8080/computer/Windows%20Slave/jenkins-agent.jnlp -secret a3fafc49ca5be88dd3ec75b86ebff2821b764920a189607ffa67a7d0e322520e -workDir "C:\Jenkins"
```

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
```

Nodes								Refresh status
S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time	
	Built-In Node	Linux (amd64)	In sync	5.68 GB	0 B	471.46 MB	0ms	
	Linux-Slave	Linux (amd64)	In sync	5.68 GB	0 B	471.46 MB	3ms	
	Windows-Slave	Windows Server 2022 (amd64)	In sync	14.20 GB	552.11 MB	14.20 GB	47ms	
Data obtained		1 min 16 sec	1 min 16 sec	1 min 16 sec	1 min 16 sec	1 min 16 sec	1 min 16 sec	

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.

Enter an item name

» Required field

Freestyle project

This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

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

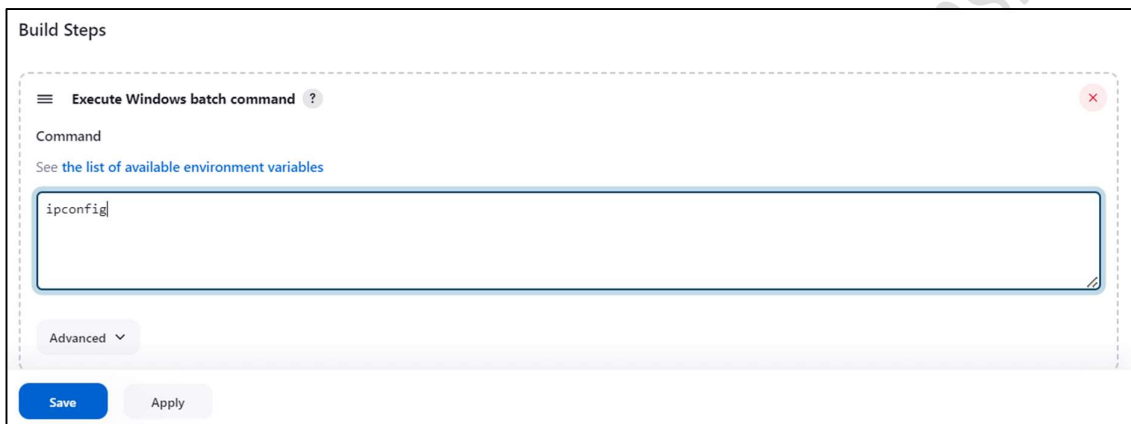
☒ Restrict where this project can be run

Label Expression

Label **Windows-Slave** matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

Advanced

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

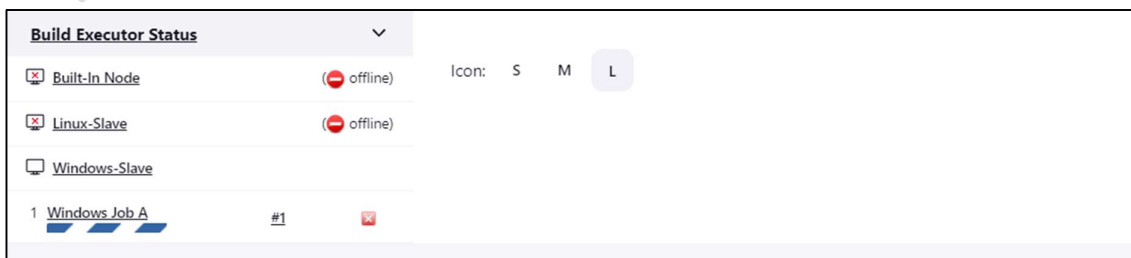


After writing the command I saved my Job. Now I am going to build this Job.

The screenshot displays a table of build jobs. The table has columns for status (S), worker (W), name, last success, last failure, and last duration. There are five rows of jobs. The first four jobs (Job A, Job B, Job C, and Slave Job A) are in a 'Success' state with a green checkmark icon. The fifth job, 'Windows Job A', is in a 'Pending' state with a blue circle icon. A 'Schedule a Build for Windows Job A' button is visible next to the last row.

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	Job A	20 hr #4	N/A	1 min 0 sec
✓	☀	Job B	21 hr #1	N/A	1 min 0 sec
✓	☀	Job C	21 hr #1	N/A	1 min 0 sec
✓	☀	Slave Job A	1 hr 22 min #4	N/A	1 min 0 sec
⋯	☀	Windows Job A	N/A	N/A	N/A

Job Started in Windows Slave



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

Status

Changes

Console Output

View as plain text

Edit Build Information

Delete build '#1'

Console Output

Started by user Himanshu Kumar

Running as SYSTEM

Building remotely on Windows-Slave in workspace C:\Jenkins\workspace\Windows_Job_A

[Windows_Job_A] \$ cmd /c call C:\Users\ADMINI~1\AppData\Local\Temp\2\jenkins7694190579504407257.bat

C:\Jenkins\workspace\Windows_Job_A>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet 2:

Connection-specific DNS Suffix . : ap-south-1.compute.internal

Link-local IPv6 Address : fe80::dff1:f318:ed90:cb4257

IPv4 Address. : 172.31.43.241

Subnet Mask : 255.255.240.0

Default Gateway : 172.31.32.1

C:\Jenkins\workspace\Windows_Job_A>exit 0

Finished: SUCCESS

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

Dashboard > Manage Jenkins > Nodes >

Clouds
Node Monitoring

Build Queue

No builds in the queue.

Build Executor Status

Built-In Node

Linux-Slave

Windows-Slave

Nodes

+ New Node

Refresh

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	5.68 GB	0 B	471.46 MB	0ms
	Linux-Slave	Linux (amd64)	In sync	5.68 GB	0 B	471.46 MB	1ms
	Windows-Slave	N/A	N/A	N/A	N/A	N/A	N/A
	Data obtained	1 min 0 sec	1 min 0 sec	1 min 0 sec	1 min 0 sec	1 min 0 sec	1 min 0 sec

Click on setting Icon for the Linux System.

Launch method ?

Launch agents via SSH

Host ?

15.206.79.198

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.