Jenkins Configure Master and Slave Nodes

Rohit Raut Apr 18 · 5 min read
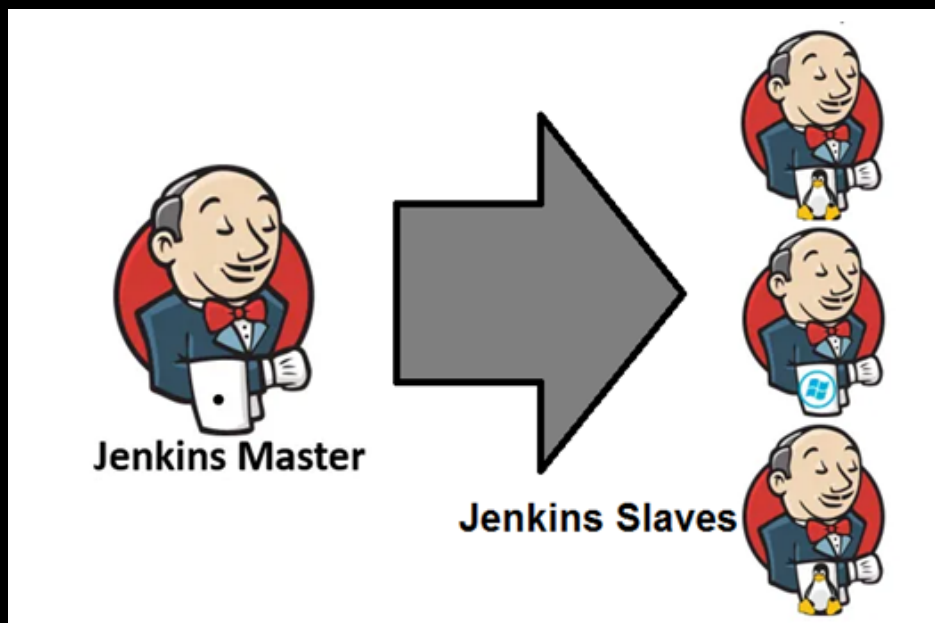
What is Jenkins?

Jenkins is an open-source automation tool written in java that automates some of the software development lifecycle tasks like build, test, deploy, and more. In this article, we will discuss how to configure Jenkins master-slave setup also called master-slave or master-agent architecture.

Need of Jenkins Mater-[Agent-slave] Architecture.

When we build the Jenkins job in a single Jenkins master node then Jenkins uses the resource of the base machine and If no executor is available then the jobs are queued in the Jenkins server. Sometimes you might need several different environments to test your builds. This cannot be done by a single Jenkins server. It is recommended not to run different jobs in the same system that required a different environment. In such scenarios where we need a different machine with a different environment that takes the specific job from the master to build.

On the same Jenkins setup, multiple teams are working with their jobs. All jobs are running on the same base operating system and the base operating system has limited

Jenkins Distributed Architecture

Jenkins uses A Master-Slave architecture to manage distributed builds. The machine where we install Jenkins software will be Jenkins master and that run's on port 8080 by default. On the slave machine, we install a program called Agent. This agent requires JVM. This agent executes the tasks provided by Jenkins master. We can launch n numbers of agents and we can configure which task will be run on which agent server from Jenkins master by assigning the agent to the task.

There are multiple ways to Install Agent in this article we will see the following.

1. **Launching Agent with SSH from Master.**

2. **Launch Agent by connecting it to master.**

Launching Agent with SSH from Master.

For this first we need to install Plugins in Jenkins and Plugin name is

```
SSH Build Agents plugin
```

Install java first in slave Node.

The Next Steps are-
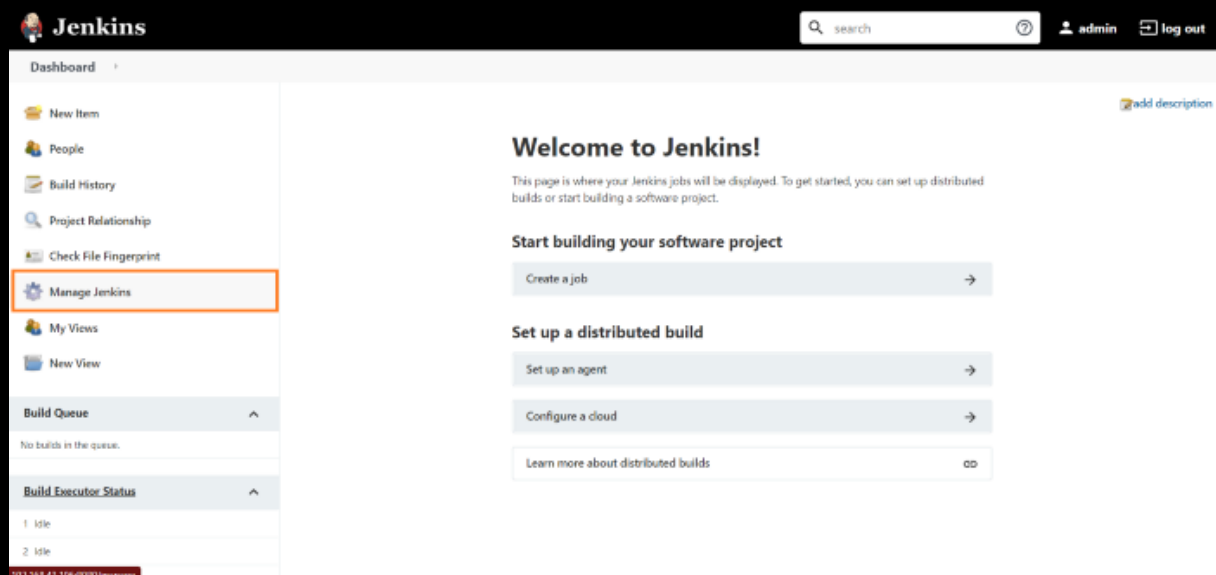
Click on Manage Jenkins
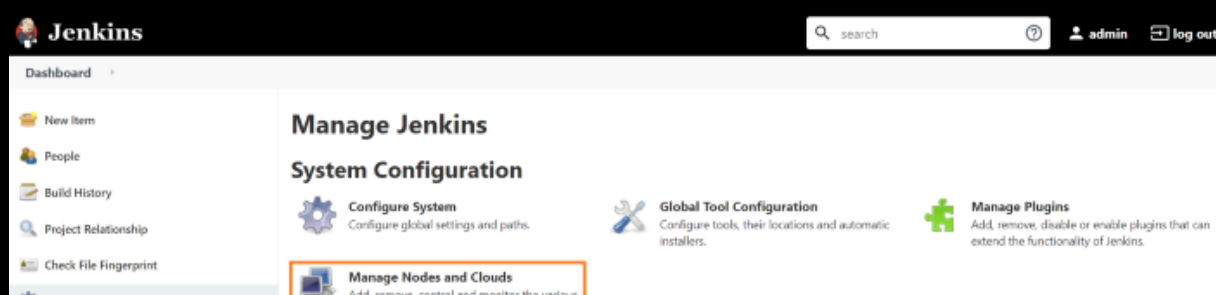
Click On Manage Nodes and Clouds

Click on New Node

Here add your slave name eg. Jenkins slave1

Permanent Agent:

Jenkins has two types of agents *1. On-Demand or Dynamic Agent Also called Dynamic Provisioning* (Automatically launch when new Job is assigned) *2. Always on or Permanent Agent (Static Agent).*

select Permanent Agent and Press OK

Give Name, permanent Agent, and Click OK

Slave Information

Executors: This number represents at a single point in time how many different jobs can run in parallel.

Remote root directory: This is the workspace directory for the agent where it stores data downloaded by the Job.

Label: This label is very important. with the help of this label, we can explicitly tell the job where you have to run.

In the launch method we have to choose "Launch agent via SSH" and to connect via ssh we required an IP address, user name, and password.

Host Key Verification Strategy: If you are connecting the first time via ssh from this system then choose no verification. That's all we have to do.

If you check the logs you will see the below output.

```
Agent successfully connected and online
```

Agent is online

Launch Agent by connecting it to master.

Here we will connect windows as a Jenkins slave. for windows, we have to use the JNLP protocol. JNLP(JAVA NETWORK LAUNCH PROTOCOL) is used to connect to Jenkins from a remote location

Just follow the same steps as above and change is only in Lanch Method. choose "Launch Agent by Connecting it to master"

Click on save

In the same directory, where you have downloaded the agent you have to run the command.

Extra Demo from my side.

Now let's make an AWS EC2 instance as Jenkins Slave.

here I have launched the EC2 instance.

The next step is to log in to VM and install java.

To verify the java version


```
[root@ip-172-31-33-209 ec2-user]# java -version
openjdk version "1.8.0_282"
OpenJDK Runtime Environment (build 1.8.0_282-b08)
OpenJDK 64-Bit Server VM (build 25.282-b08, mixed mode)
[root@ip-172-31-33-209 ec2-user]#
```

Node Config

ADD Private KEY.ppk format

Connected

Build Environment

☐ Use secret text(s) or file(s)

☐ Create Delivery Pipeline version

Build

Execute shell ✕

Command

```
pwd
cat /etc/os-release
```

See the list of available environment variables

Advanced...

Add build step ▾

Post-build Actions

Add post-build action ▾

Save   Apply

Dashboard ▸ EC2 ▸ #3

🔼 Back to Project

🔍 Status

📝 Changes

🖥 Console Output

   📄 View as plain text

📝 Edit Build Information

🚫 Delete build '#3'

◀ Previous Build

🔵 **Console Output**

```
Started by user unknown or anonymous
Running as SYSTEM
Building remotely on AWS_EC2_Instance (AWS) in workspace /home/ec2-user/jenkin-workspace/workspace/EC2
[EC2] $ /bin/sh -xe /tmp/jenkins8938257145252397978.sh
+ pwd
/home/ec2-user/jenkin-workspace/workspace/EC2
+ cat /etc/os-release
NAME="Amazon Linux"
VERSION="2"
ID="amzn"
ID_LIKE="centos rhel fedora"
VERSION_ID="2"
PRETTY_NAME="Amazon Linux 2"
ANSI_COLOR="0;33"
CPE_NAME="cpe:2.3:o:amazon:amazon_linux:2"
HOME_URL="https://amazonlinux.com/"
Finished: SUCCESS
```

Job and Build Console Output

Congratulation, We have successfully build Jenkins Master-Slave Architecture

Follow me for more interesting Blogs.

Thank you for reading!! 😇

Medium

About   Write   Help   Legal

Medium