Jenkins Master Slave Configuration

Posted on *February 24, 2017*

Introduction :

Jenkins uses a Master-Slave architecture to manage distributed builds. In this architecture, Master and Slave communicate through TCP/IP protocol.

In master-slave architecture of Jenkins, master represents basic installation of Jenkins and it handles all tasks for build system.

Jenkins master is used to handle following things:

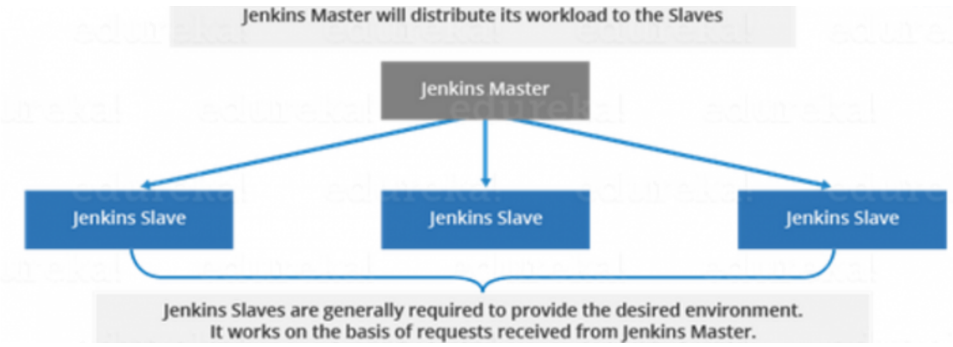
- Scheduling build jobs.
- Dispatching builds to the slaves for the actual execution.
- Monitor the slaves (possibly taking them online and offline as required).
- Recording and presenting the build results.
- A Master instance of Jenkins can also execute build jobs directly.

A slave is a computer that is set up to offload build projects from the master and once connection is established between master and slave , tasks distribution is done automatic. Each slave runs a separate program called a “slave agent” . There is no need to install the full Jenkins on a slave. There are various ways to start slave agents, but in the end the “slave agent” and “Jenkins master” needs to establish a bi-directional communication link (for example a TCP/IP socket.) in order to operate.

Following are the characteristics of Jenkins Slaves:

- It hears requests from the Jenkins Master instance.
- Slaves can run on a variety of operating systems.
- The job of a Slave is to do as they are told to, which involves executing build jobs dispatched by the Master.

Below diagram shows the sample master slave connection. It consists of a Jenkins Master which is managing three Jenkins Slave.


Jenkins Master will distribute its workload to the Slaves
Jenkins Master
Jenkins Slave
Jenkins Slave
Jenkins Slave
Jenkins Slaves are generally required to provide the desired environment.
It works on the basis of requests received from Jenkins Master.

Note : To configure slave machine “slave setup” plugin is use and It also allows master to start and stop slaves on demand from the master node.

How To Setup Slaves On Jenkins :

There are two ways for setting up the slaves.

1. Using username and password
2. Using ssh keys

Explaining slave setup using username and password. Preparing slaves using ssh key follow the same steps only difference is that instead of username and password for slave , ssh key is used.

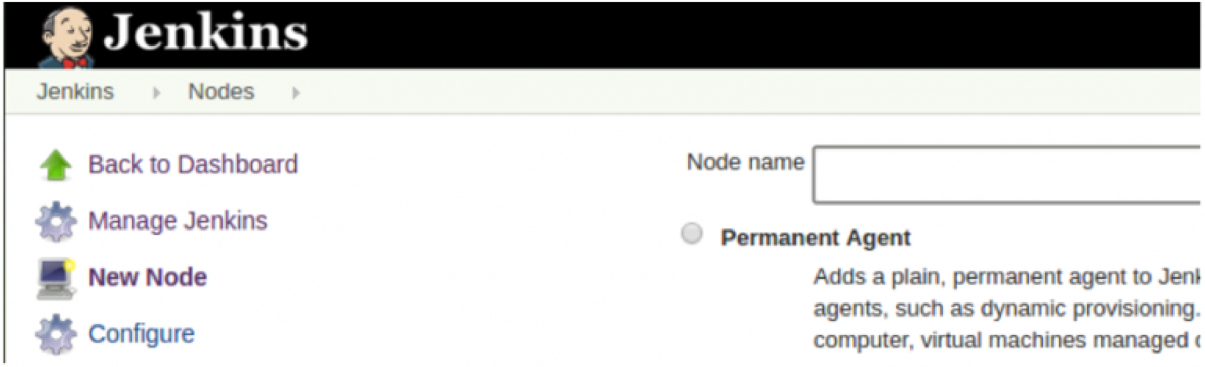
Slave Prerequisites :

1. Create a Jenkins user and a password using the following command
sudo adduser jenkins –shell /bin/bash
2. Now, login as Jenkins user.
su jenkins
3. Create a “jenkins_slave” directory under /home/jenkins
mkdir /home/jenkins/jenkins_slave

Note : Before Executing above steps JRE (Java Run time Environment) need to install on slave machine.

Setting Up Slaves Using Username And Password :

1. Go to **Jenkins dashboard** -> **Manage Jenkins** -> **Manage Nodes.**
2. Select **new node** option.

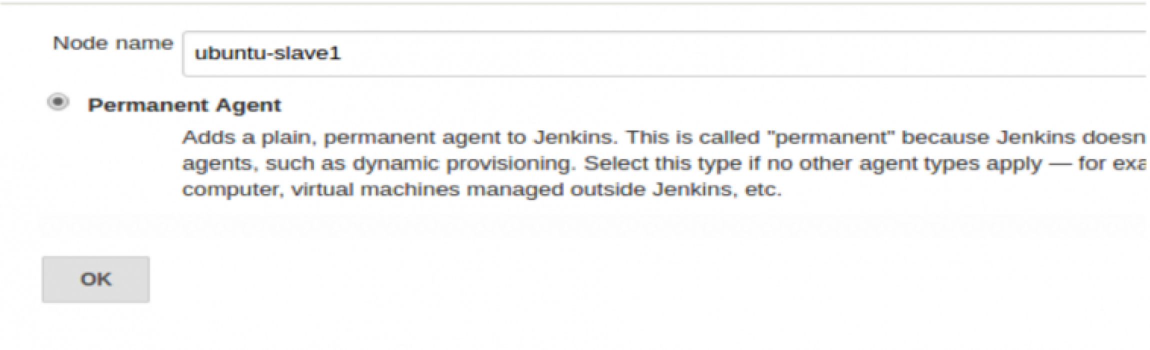


Node name

○ Permanent Agent

Adds a plain, permanent agent to Jenk
agents, such as dynamic provisioning.
computer, virtual machines managed

3. Give it a name, select the “**permanent agent**” option and click **OK**.

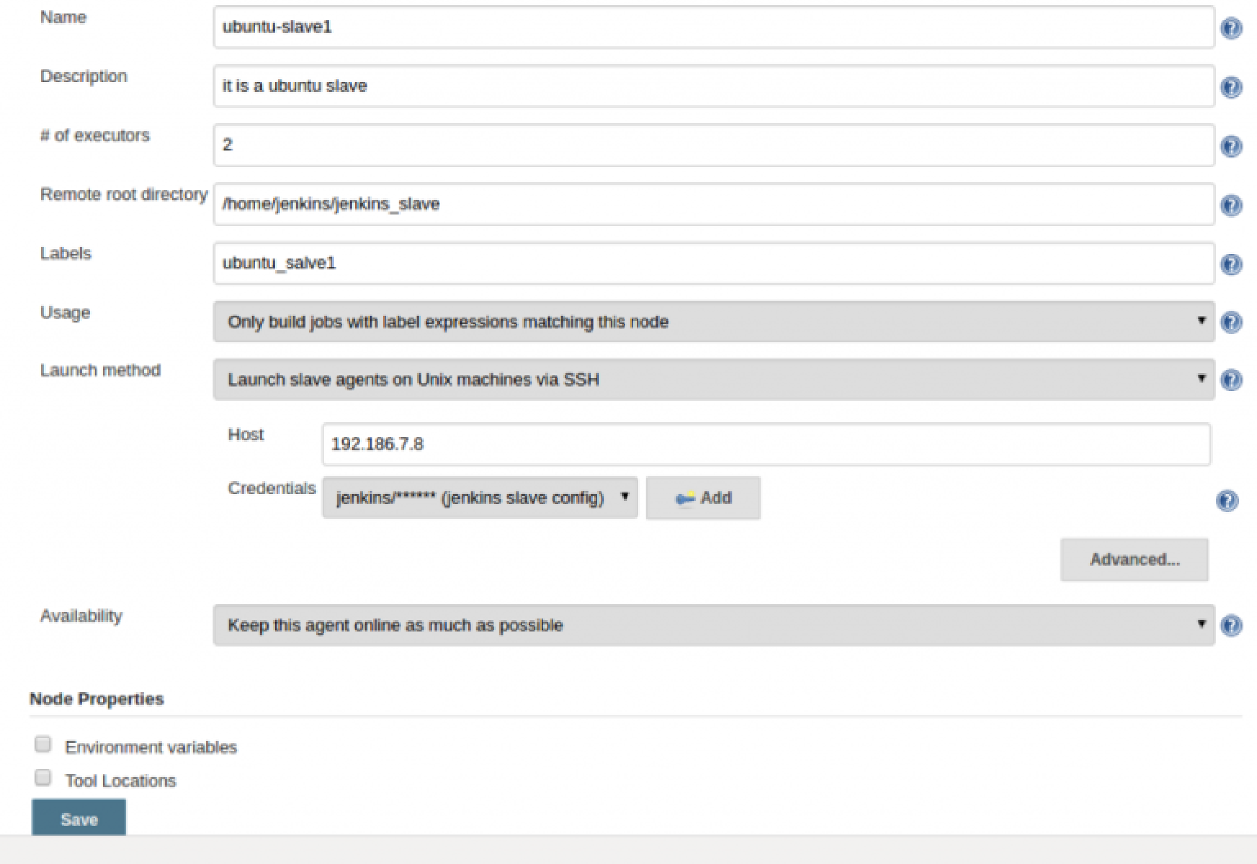


Node name: ubuntu-slave1

● Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn
agents, such as dynamic provisioning. Select this type if no other agent types apply — for exa
computer, virtual machines managed outside Jenkins, etc.

4. Enter the details as shown in the image below and save it. For credential box, click the add button and enter the slaves Jenkins username and password (For logging into the slave machine) that created during the slave configuration. To know what each option means, click the question mark at the right side of each text box.



Name: ubuntu-slave1

Description: it is a ubuntu slave

# of executors: 2

Remote root directory: /home/jenkins/jenkins_slave

Labels: ubuntu_salve1

Usage: Only build jobs with label expressions matching this node

Launch method: Launch slave agents on Unix machines via SSH

Host: 192.186.7.8

Credentials: jenkins/****** (jenkins slave config)

Availability: Keep this agent online as much as possible

Node Properties

□ Environment variables

□ Tool Locations

In above Diagram :**Name** : Name that uniquely identifies an agent within this Jenkins installation.
Description : Optional ,description for this agent.
# of executors : The maximum number of concurrent builds that Jenkins may perform on this agent.
Remote root directory : An agent needs to have a directory dedicated to Jenkins. Specify the path to this directory on the agent. It is best to use an absolute path, such as /var/jenkins or c:\jenkins. This should be a path local to the agent machine. There is no need for this path to be visible from the master.
Labels : Labels (or tags) are used to group multiple agents into one logical group.**Usage** : Controls how Jenkins schedules builds on this node.Use this node as much as possible only build jobs with label expressions matching this node(select if above label field is set.)

Launch method : It Controls how Jenkins starts this agent.
Launch agent via Java Web Start :
Allows an agent to be launched using Java Web Start. In this case, a JNLP file must be opened on the agent machine, which will establish a TCP connection to the Jenkins master. This means that the agent need not be reachable from the master,the agent just needs to be able to reach the master. By default, the JNLP agent will launch a GUI, but it's also possible to run a JNLP agent without a GUI, e.g. as a Window service.
Launch agent via execution of command on the master :
Starts an agent by having Jenkins execute a command from the master. Use this when the master is capable of remotely executing a process on another machine, e.g. via SSH or RSH**.**
Launch slave agents via SSH:
Starts a slave by sending commands over a secure SSH connection. The slave needs to be reachable from the master, and master will have to supply an account that can log in on the target machine. No root privileges are required.
Let Jenkins control this Windows slave as a Windows service:
In case of windows server as slave. Please use this option. Please do it in your own risk as it is not tested in this process.
Start and stop this node on-demand (Need slave setup plugin):
Wrapper for other launch methods: executes a script (i.e. to provision a VM) before attempting to connect to the slave machine then launches the Jenkins slave with a launch method of user choice, and after disconnection executes another script (i.e. for shutting down the VM).
Availability : Controls when Jenkins starts and stops this agent. Keep this agent online as much as possible.
Take this agent online and offline at specific times :

In this we set "Startup Schedule" and "Scheduled Up time"

Startup Schedule : schedule cron job to start and stop node.
Examples:
every fifteen minutes : H/15 * * * *
Scheduled Up time : The number of minutes to keep the node up for. If this is longer than the startup schedule, then the node will remain constantly on-line.Take this agent online when in demand, and offline when idle

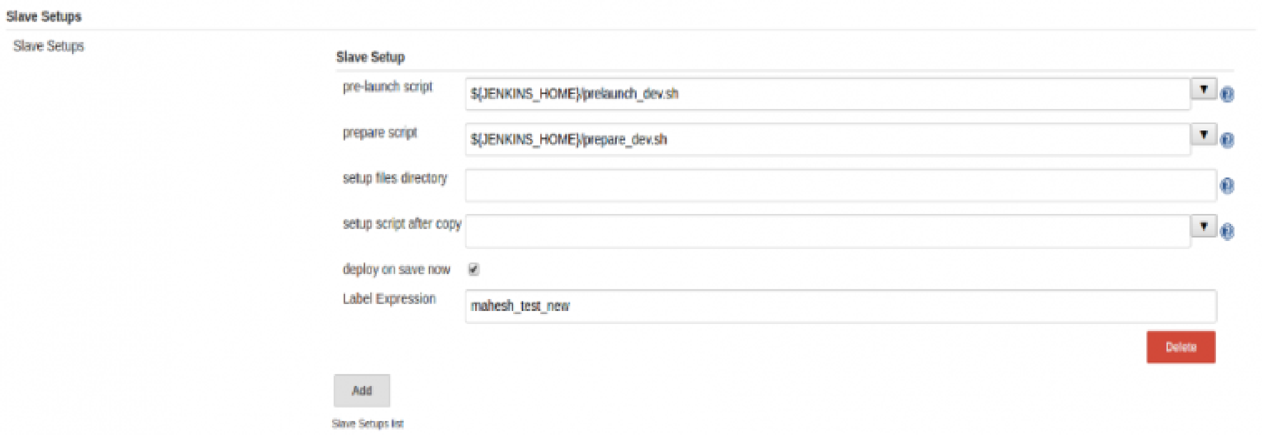
5. Click on save button to create new new node and after that Jenkins will automatically connect to the slave machine

How to prepare slave to execute build?
To prepare slave setup and configuration we should install "Slave Setup" plugin.This plugin prepares slave to execute build . It also allows Jenkins master to start and stop slave on demand from the master node.

Steps to prepare Slave Server :

1. Go to Manage Jenkins->Configure System and check the "Slave Setups" block and click on add button.

2. **pre-launch script** :
   After configuring the slave on Jenkins master ,slave should be launch from Jenkins master to execute build on slave . Pre launch script is executed before Jenkins-master launch the slave.
   Steps to install dependencies on slave using pre-launch script :
   Create shell script on $JENKINS_HOME(i.e. /var/lib/jenkins/).
   Set that shell script path in pre-launch script. **Example** :
   ${JENKINS_HOME}/prelaunch_dev.sh .
   Go to Manage Jenkins->Manage Nodes->Click on the node name-> Click on "Launch Agent" button.
   After clicking on "Launch Agent" Button ,Pre-launch script will be executed
3. **prepare script:**
   Specify the command to be executed on the master before Jenkins starts using the slave. Follow the steps same as "pre-launch script" and specify the prepare script path in front of "prepare script" in below image.
4. **Label Expression :**
   In master slave configuration label is used to prepare build on slave and Configure the slave.In slave setup "label expression" is used to identify the slave Node to install dependencies.

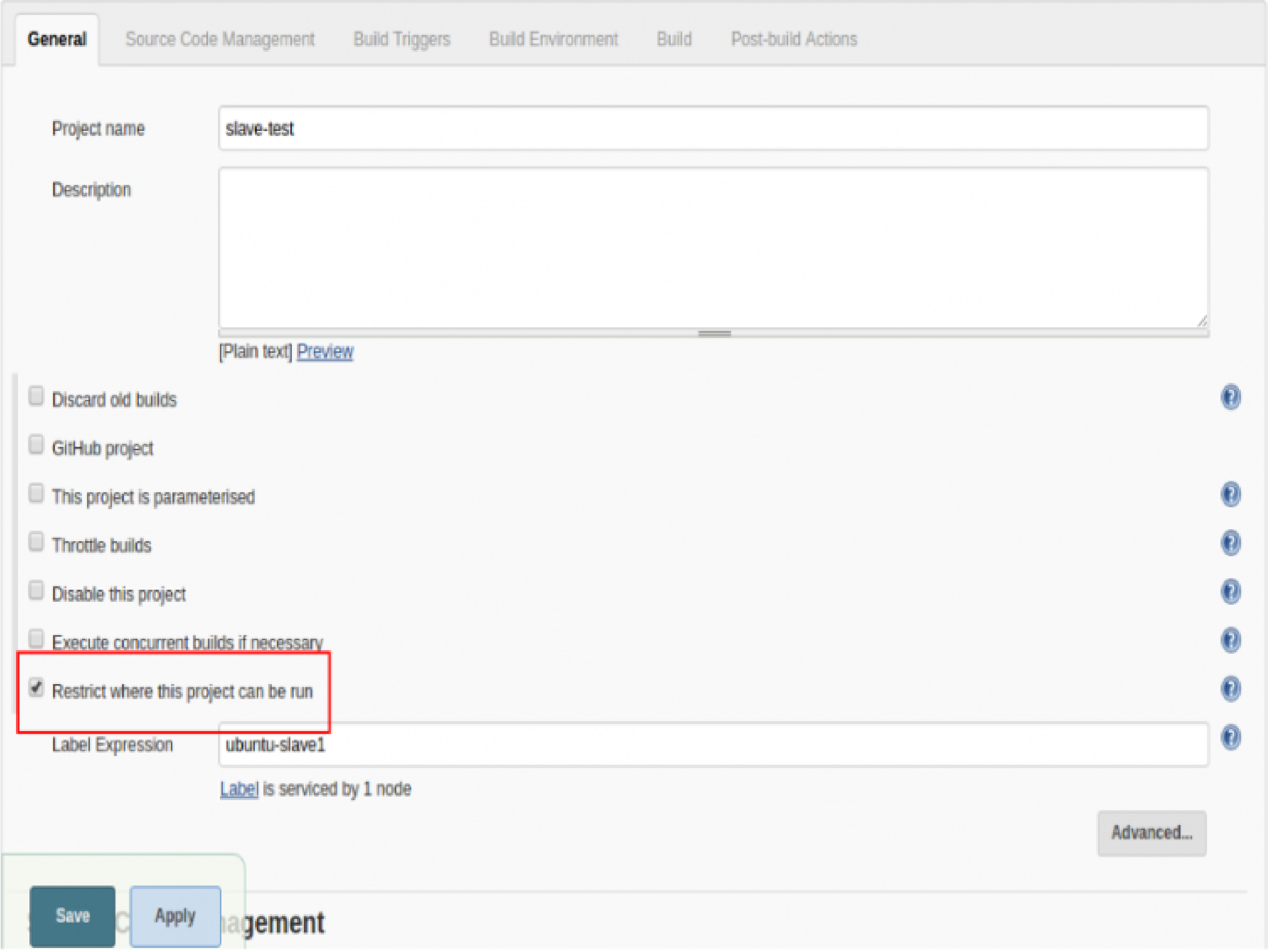


pre-launch script: ${JENKINS_HOME}/prelaunch_dev.sh
prepare script: ${JENKINS_HOME}/prepare_dev.sh
setup files directory:
setup script after copy:
deploy on save now ☑
Label Expression: mahesh_test_new

Test The Slave :

To test the slave, create a sample project and select the option as shown below.In below image add Node label name in "label Expression " to provide connection between jobs and slave node.

Note : Label name in the below image should be same as the Slave label name.



Project name: slave-test

Description:

□ Discard old builds

□ GitHub project

□ This project is parameterised

□ Throttle builds

□ Disable this project

□ Execute concurrent builds if necessary

☑ Restrict where this project can be run

Label Expression: ubuntu-slave1

Label is serviced by 1 node

1

This entry was posted in *Miscellaneous* and tagged *distributed builds*, *jenkins*, *master-slave*. Bookmark the *permalink*.

← How to apply Tablerate shipping on discounted subtotal in Magento 1?

PHPUnit – Power of Data Provider →

1 thought on "Jenkins Master Slave Configuration"

Harshal Sarode says:

April 11, 2017 at 2:46 pm

Reply

thanks so much.can you pls tell how to attach windows slave to jenkins Linux server with Launch method ” Let Jenkins Control this Windows slave as windows service” i screwed up with this issue.
thanks in advance.

Leave a Reply

Your email address will not be published. Required fields are marked *

Comment

Name *

Email *

Website

I'm not a robot

reCAPTCHA
Privacy - Terms

POST COMMENT

Search …

EMAIL SUBSCRIPTION

Enter your email address:

SUBSCRIBE

Delivered by FeedBurner

LOGIN

Username

Password

□ Remember Me

LOG IN

Forgot Password?

RECENT POSTS

- Docker Compose
- WordPress Plugin – WP Simple FAQs (v1.0)
- An Introduction on Chef
- WordPress Plugin – WP Survey Manager (v1.0)
- Laravel – The PHP Framework For Web Artisans