

WRT ubuntu worker node: (isnt working for me)

1) Create a new VM

2) Make a Jenkins directory

```
sudo mkdir /var/lib/jenkins
```

```
ls -l /var/lib | grep jenkins
```

You will see the owner of the newly created directory is 'root'

3) Now, we need to create a new user 'jenkins' and specify the above created directory to be used by the newly created user

```
sudo useradd -d /var/lib/jenkins jenkins
```

4) Now, we need to make sure that the 'jenkins' user has the permission to operate on the above created directory

```
sudo chown -R jenkins:jenkins /var/lib/jenkins
```

5) run the below command

```
ls -l /var/lib | grep jenkins
```

You will notice that now the owner of this directory is 'jenkins'

6) Now, we need to generate a key

```
ssh-keygen
```

7) currently the above-generated key will be present in the root at the ~/.ssh

We will copy the currently generated key and then create a .ssh in the jenkins directory we just created and add that key there.

So, first creating a new .ssh directory in the jenkins directory

```
sudo mkdir /var/lib/jenkins
```

Copying the currently generated key from:

```
cat ~/.ssh/id_rsa.pub
```

Create a new authorized_keys file in the jenkins directory

```
sudo vi /var/lib/jenkins/.ssh/authorized_keys
```

And paste the copied key here.

Now, check if keys was copied successfully:

```
cat /var/lib/jenkins/.ssh/authorized_keys
```

8) We know jenkins is a java program, hence we need to install JDK (same version as master)

```
sudo apt-get update
```

```
sudo apt-get install openjdk-11-jre
```

9) We need to copy the private key from:

```
cat ~/.ssh/id_rsa
```

10) Now, we need to go back to our jenkins master, in the jenkins UI and setup a new node where we will be required to paste the above copied private key.

We will also need to give the newly created directory in the working directory for the node.

When we click for it to add the node and connect, and gives it some time it will fail. When we look in logs to see what the error was we will be able to see that it was trying to search for a file at `/var/lib/jenkins/.ssh/known_hosts` which was never created by us.

11) Now, to rectify this, we need to go into our MASTER node and follow the steps given below:

a) Give sudo privileges and also set passwordless, using command:

sudo visudo

Add the below line below the root user entry:

jenkins ALL=(ALL) NOPASSWD:ALL

(here jenkins is the user that we created earlier)

b) now we need to change `/etc/ssh/sshd_config` in both the controller server and the host server, to allow passwordless ssh

```
sudo nano /etc/ssh/sshd_config
```

```
#PubkeyAuthentication yes (uncomment this line)
```

```
PasswordAuthentication no (change 'no' to 'yes')
```

c) Now, restart the ssh server and check the status if it is running on both controller and host

```
sudo systemctl restart ssh
```

```
sudo systemctl status ssh
```

d) Now we need to set password for the 'ubuntu' user in the agent node, because we the password in our master node when we transfer the ssh key to login to the 'ubuntu' user of the agent node.

```
sudo passwd ubuntu
```

('ubuntu' is the user)

12) Go back to the AGENT node and generate SSH key:

```
ssh-keygen
```

13) Transfer the key to the 'ubuntu' user of the agent node

```
ssh-copy-id -i id_rsa.pub ubuntu@<IP-address>
```

It will prompt you to enter the password, enter the password you setup for the 'ubuntu' in the earlier steps

NOTE: If this doesnt work try using 'sudo' with the above command. If it still doesnt work, try step 12 for the other user that we created 'jenkins' and continue again from there.

14) Once we are inside the 'ubuntu' user of our MASTER node, try to change user to 'jenkins'

```
Sudo su - jenkins
```

15) Now exit

exit

16) So, when we logged in to the MASTER node from our agent node using ssh, it would have created a known_hosts file which we can use in our jenkins known_hosts file.

You can see known hosts at:

sudo cat ~/.ssh/known_hosts

Now, we need to copy this to our known hosts file in /var/lib/jenkins/.ssh

sudo cp ~/.ssh/known_hosts /var/lib/jenkins/.ssh

Once all the above steps are done we go back to the jenkins UI where adding agent failed and launch agent again.

Slave:

```
sudo passwd ubuntu
sudo mkdir /var/lib/jenkins
ls -l /var/lib | grep jenkins
sudo useradd -d /var/lib/jenkins jenkins
sudo chown jenkins:jenkins /var/lib/jenkins
ls -l /var/lib | grep jenkins
sudo mkdir /var/lib/jenkins/.ssh
cat ~/.ssh/id_rsa.pub
sudo vi /var/lib/jenkins/.ssh/authorized_keys
sudo cat /var/lib/jenkins/.ssh/authorized_keys
sudo apt-get update
sudo apt-get install fontconfig openjdk-11-jre
java -version
sudo cat ~/.ssh/id_rsa
```

Master:

Paste the private key of slave vm in .pem file on master node and then ssh into slave's ubuntu user

Sudo su - jenkins

Once you log in to slave, exit from there and come back to master. Now master have a known_hosts file (but jenkins directory does not), so we copy that file to jenkins directory.

=====

Slave:

sudo apt-get update

sudo apt-get install openjdk-11-jre -y

sudo mkdir jenkins

sudo chmod 777 -R jenkins

=====

Slave:

sudo apt-get update

sudo apt-get install openjdk-11-jre

cd /opt/

sudo mkdir jenkins

sudo chmod 777 jenkins

cd jenkins

pwd

ssh-keygen

cd ~/.ssh

Private key: sudo cat id_rsa

Public key: sudo cat id_rsa.pub

(very very important, Add the private key in the below file)

sudo nano authorized_keys

Remote root directory: /var/lib/jenkins

Launch method: Launch agent via SSH

Host: Enter the IP address of the slave

Select authentication using SSH.

In the SSH key enter the private key.

Host Key Verification Strategy: Non verifying verification strategy
(This very very important, else this method to add the slave wont work)

=====

Slave node:

```
sudo su
cd /opt/
mkdir jenkins
chmod 777 jenkins
cd jenkins/
pwd
```

(gives out /opt/jenkins, we use this as the Remote root directory while setting slave node in master jenkins UI)

```
ssh-keygen
```

```
cd ~/.ssh
```

```
cat id_rsa
```

(we use this private key while setting SSH key in master jenkins UI)

```
cat id_rsa.pub
```

```
cat authorized_keys
```