

VIRTUAL BOX UBUNTU – PREMISE

Launch two Ubuntu Servers (Control jenkins & Agent)

```
ssh osboxes@10.0.0.60 in vsc
```

```
password:
```

```
sudo su
```

```
password:
```

You should become the root

Run the Commands Below

```
# apt-get install update
```

```
# sudo apt-get install openssh-server
```

```
# sudo service ssh restart
```

```
# sudo systemctl status sshd
```

```
# sudo apt install default-jre
```

```
# java --version
```

Run the below commands to install jenkins from .warfile on ubuntu premise server

```
# curl -fsSL https://pkg.jenkins.io/debian/jenkins.io.key | sudo tee \  
/usr/share/keyrings/jenkins-keyring.asc > /dev/null
```

Then add a Jenkins apt repository entry:

```
# echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \  
https://pkg.jenkins.io/debian binary/ | sudo tee \  
/etc/apt/sources.list.d/jenkins.list > /dev/null
```

Update your local package index, then finally install Jenkins:

```
# sudo apt-get update
```

```
# ##### sudo apt-get install fontconfig openjdk-11-jre
# sudo apt-get install jenkins
```

Start Jenkins

You can enable the Jenkins service to start at boot with the command:

```
# sudo systemctl enable jenkins
```

You can start the Jenkins service with the command:

```
# sudo systemctl start jenkins
```

You can check the status of the Jenkins service using the command:

```
# sudo systemctl status jenkins
```

Creating the **RSA Key Pair – one public and one private**

Ssh-keygen is a tool for creating **new authentication key pairs for SSH**. Such key pairs are used for **automating logins, single sign-on, and for authenticating hosts (agent) – private key**

Run

```
# ssh-keygen -f ~/.ssh/jenkins_agent_key
```

Provide a passphrase to use with the key [enter]

Confirm the output looks something like this: [enter]

Private Key

Run

```
# cat .ssh/jenkins_agent_key
```

Public key

Run

```
# cat .ssh/jenkins_agent_key.pub
```

Run

```
# sudo lsof -i -P -n | grep LISTEN → check the listening port in the system.
```

Run

```
# sudo apt-get remove openssh-server → Remove existing openssh tcp before running below docker container command
```

Creating your Docker agent On Linux

Run

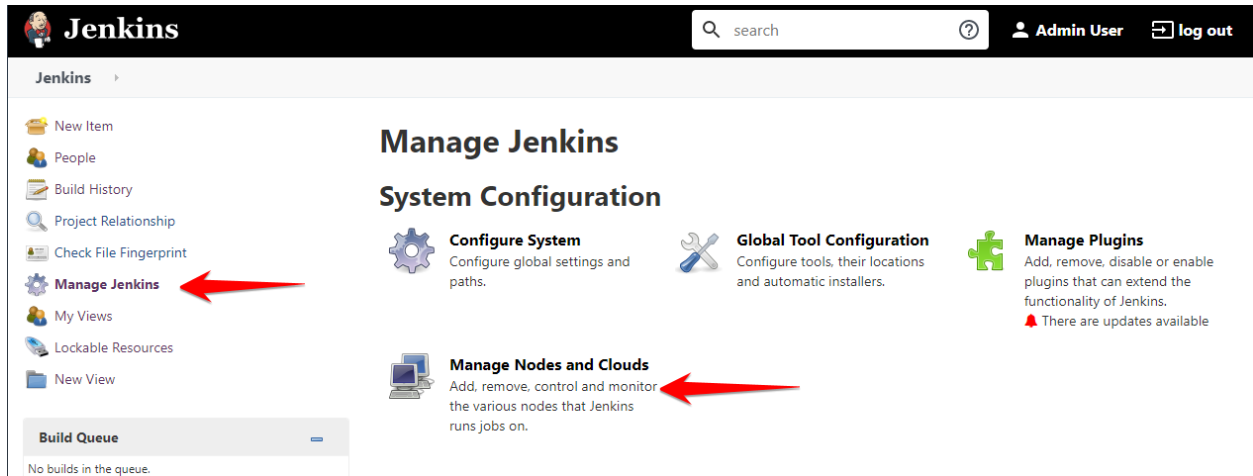
```
docker run -d --rm --name=agent1 -p 22:22 \
-e "JENKINS_AGENT_SSH_PUBKEY=[your-public-key]" \
jenkins/ssh-agent:alpine
```

Run


```
# VARS1="HOME=|USER=|MAIL=|LC_ALL=|LS_COLORS=|LANG="
# VARS2="HOSTNAME=|PWD=|TERM=|SHLVL=|LANGUAGE=|_="
# VARS="${VARS1}|${VARS2}"
# docker exec agent1 sh -c "env | egrep -v '^(${VARS})' >>/etc/environment"
```

Now the docker agent is up and running.

Configure the SSH plugin to connect the jenkins with the docker agent



1. Go to **New Node** option in side menu;
2. Fill the Node/agent name and select the type; (e.g. Name: agent1, Type: Permanent Agent)
3. Now fill the fields:
 - Remote root directory; (e.g.: /home/jenkins)
 - label; (e.g.: agent1)
 - usage; (e.g.: only build jobs with label expression...)
 - Launch method; (e.g.: Launch agents by SSH)
 - Host; (e.g.: localhost or your IP address)
 - Credentials; (e.g.: jenkins)
 - Host Key verification Strategy; (e.g.: Manually trusted key verification ...)



Jenkins

Admin User
log out

Jenkins > Nodes >

[Back to Dashboard](#)
[Manage Jenkins](#)
[New Node](#)
[Configure Clouds](#)
[Node Monitoring](#)

Build Queue

No builds in the queue.

Build Executor Status

1 Idle
2 Idle

Name:

Description:

of executors:

Remote root directory:

Labels:

Usage:

Launch method:

Host:

Credentials:

Host Key Verification Strategy:

☐ Require manual verification of initial connection

Availability:


Node Properties

☐ Disable deferred wipeout on this node

☐ Environment variables

☐ Tool Locations

Press the button save and the agent1 will be registered, but offline. Click on it.



Jenkins

Admin User
log out

Jenkins > Nodes >

[Back to Dashboard](#)
[Manage Jenkins](#)
[New Node](#)
[Configure Clouds](#)
[Node Monitoring](#)



Build Queue

No builds in the queue.

Build Executor Status

master

1 Idle

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	agent1		N/A	N/A	N/A	N/A	N/A
	master	Linux (amd64)	In sync	234.39 GB	4.00 GB	234.39 GB	0ms
Data obtained 2 ms			2 ms	1 ms	0 ms	0 ms	55 min

Now press the button **Launch agent** and wait some seconds, then you should receive the message: **Agent successfully connected and online** on the last log line

Jenkins > Nodes > agent1

```
UPPER=1
OPTIND=1
OSTYPE=linux-gnu
PATH=/usr/local/openjdk-11/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
PIPESTATUS=([0]="0")
PPID=170
PS4='+ '
PWD=/home/jenkins
SHELL=/bin/bash
SHELLOPTS=braceexpand:hashall:interactive-comments
SHLVL=1
SSH_CLIENT='172.17.0.1 40120 22'
SSH_CONNECTION='172.17.0.1 40120 172.17.0.3 22'
TERM=dumb
UID=1000
USER=jenkins
_='']'
[07/07/20 23:18:53] [SSH] Checking java version of /home/jenkins/jdk/bin/java
Couldn't figure out the Java version of /home/jenkins/jdk/bin/java
bash: /home/jenkins/jdk/bin/java: No such file or directory

[07/07/20 23:18:53] [SSH] Checking java version of java
[07/07/20 23:18:53] [SSH] java -version returned 11.0.7.
[07/07/20 23:18:53] [SSH] Starting sftp client.
[07/07/20 23:18:53] [SSH] Copying latest remoting.jar...
[07/07/20 23:18:54] [SSH] Copied 1,522,833 bytes.
Expanded the channel window size to 4MB
[07/07/20 23:18:54] [SSH] Starting agent process: cd "/home/jenkins" && java -jar remoting.jar -workDir /home/jenkins -jar-cache /home/jenkins/remoting/jarCache
Jul 08, 2020 2:18:54 AM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/jenkins/remoting as a remoting work directory
Jul 08, 2020 2:18:54 AM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /home/jenkins/remoting
<===[JENKINS REMOTING CAPACITY]==>channel started
Remoting version: 4.3
This is a Unix agent
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by jenkins.slaves.StandardOutputSwapper$ChannelSwapper to constructor
java.io.FileDescriptor(int)
WARNING: Please consider reporting this to the maintainers of jenkins.slaves.StandardOutputSwapper$ChannelSwapper
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
Evacuated stdout
Agent successfully connected and online
```

