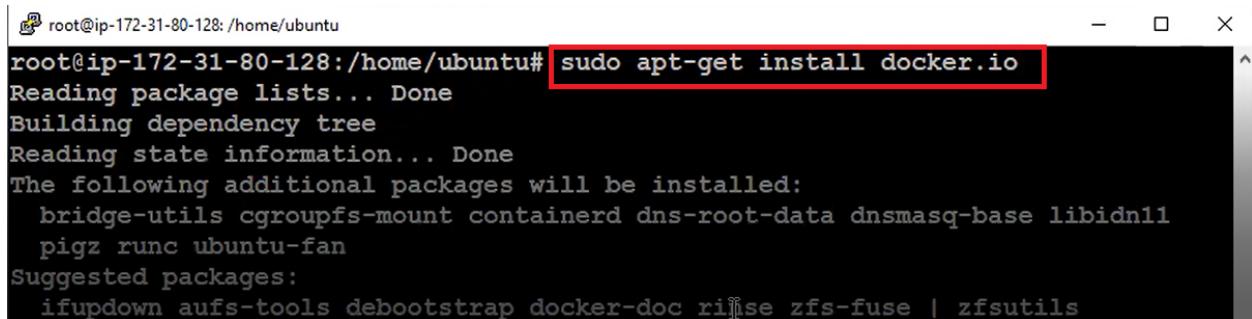


Distributed Architecture in Jenkins

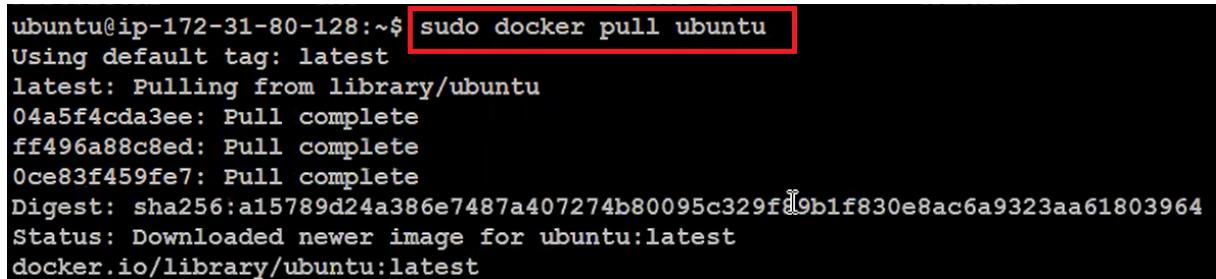
This document covers the instructions on how to set up the master-slave architecture in Jenkins. So, let's take a look at the steps involved in setting up the master-slave architecture, once you have logged into your Ubuntu instance and launched Jenkins on port 8080.

1. In your Ubuntu instance, run the following command to install Docker:
sudo apt-get install docker.io. When prompted for permission during installation, type '**y**' and then press enter.



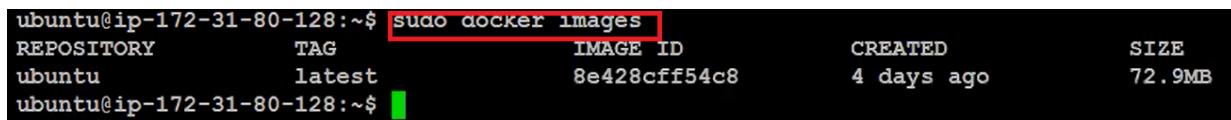
```
root@ip-172-31-80-128:/home/ubuntu# sudo apt-get install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils cgroupfs-mount containerd dns-root-data dnsmasq-base libidn11
    pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools debootstrap docker-doc riilse zfs-fuse | zfsutils
```

2. Next, we need to pull the Ubuntu image. Therefore, run the command **sudo docker pull ubuntu**.



```
ubuntu@ip-172-31-80-128:~$ sudo docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
04a5f4cda3ee: Pull complete
ff496a88c8ed: Pull complete
0ce83f459fe7: Pull complete
Digest: sha256:a15789d24a386e7487a407274b80095c329fd9b1f830e8ac6a9323aa61803964
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest
```

3. We can check if the image is properly installed or not by running the command: **sudo docker images**.



REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	latest	8e428cff54c8	4 days ago	72.9MB

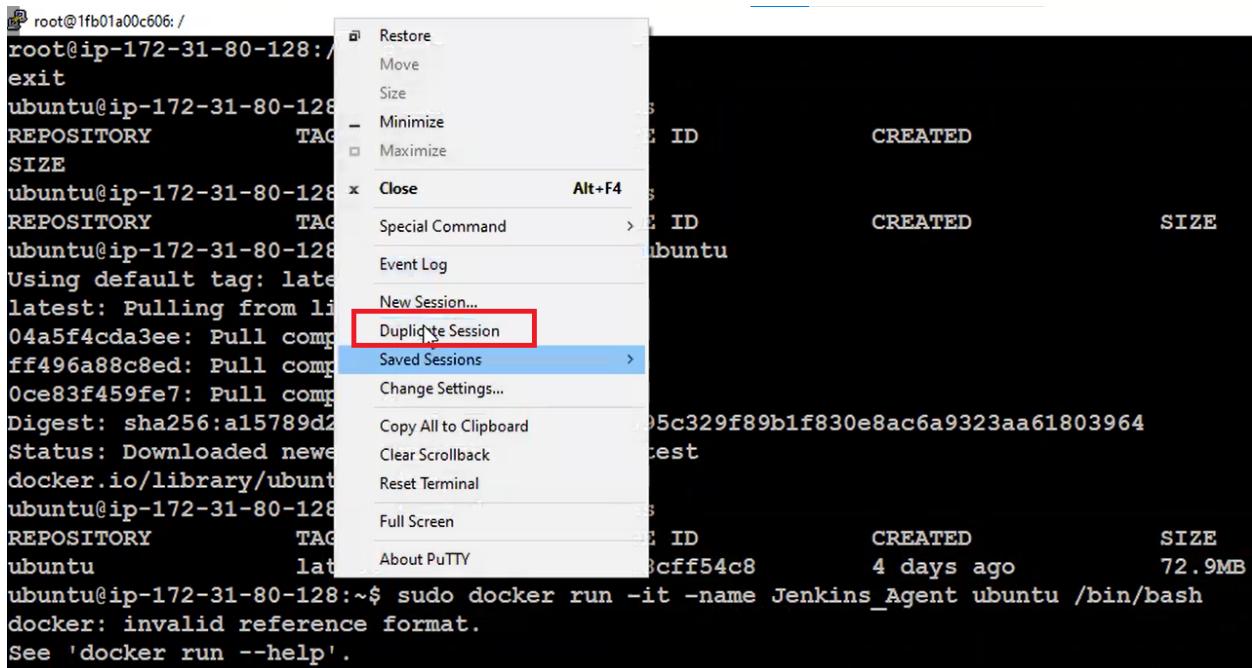
4. Then start the Ubuntu container by running the command:
sudo docker run -it --name Jenkins_Agent ubuntu /bin/bash.

```
ubuntu@ip-172-31-80-128:~$ sudo docker run -it --name Jenkins_Agent ubuntu /bin/bash
root@1fb01a00c606:/#
```

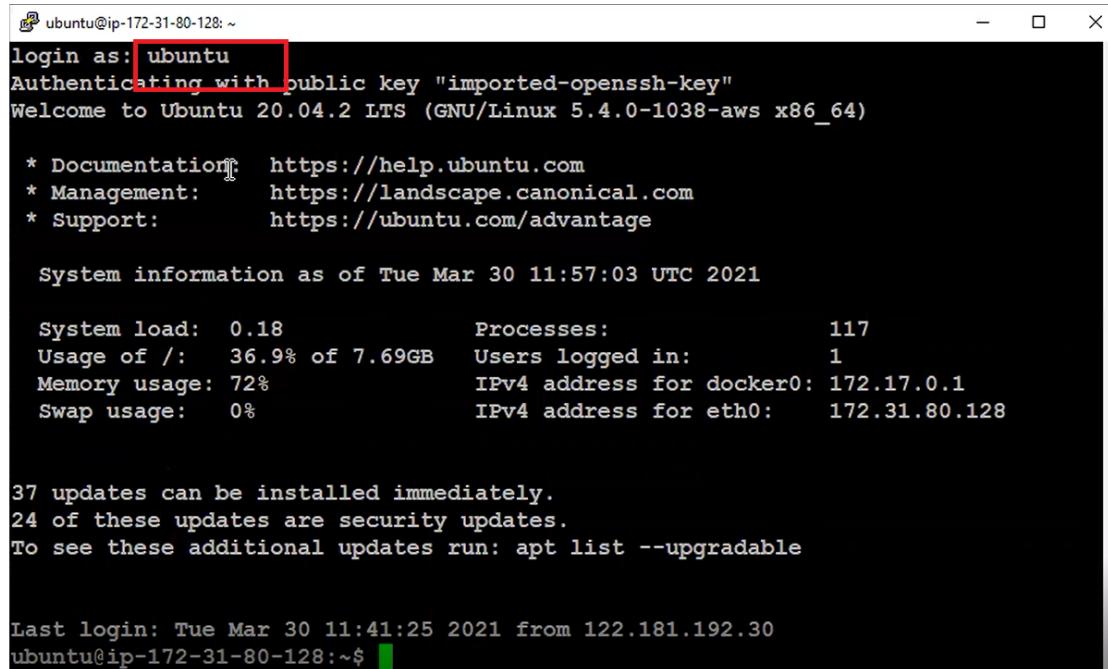
5. We need to know the IP address of the image that is currently being run. Therefore, type the following command: **cat /etc/hosts**.

```
root@1fb01a00c606:/# cat /etc/hosts
127.0.0.1      localhost
:1      localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.17.0.2      1fb01a00c606
root@1fb01a00c606:/#
```

6. We need to duplicate the window to login on the AWS EC2 instance, as the container is running on the current window. Therefore, click on the top and select the **Duplicate Session**, as shown in the image below.



7. Enter the username in the other window as shown in the image below.



```
ubuntu@ip-172-31-80-128: ~
login as: ubuntu
Authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-1038-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

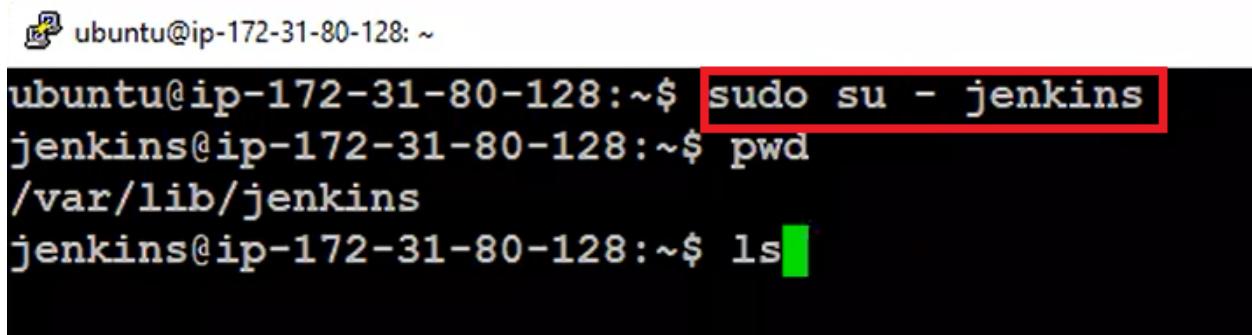
 System information as of Tue Mar 30 11:57:03 UTC 2021

 System load:  0.18          Processes:           117
 Usage of /:   36.9% of 7.69GB  Users logged in:      1
 Memory usage: 72%          IPv4 address for docker0: 172.17.0.1
 Swap usage:   0%            IPv4 address for eth0:   172.31.80.128

37 updates can be installed immediately.
24 of these updates are security updates.
To see these additional updates run: apt list --upgradable

Last login: Tue Mar 30 11:41:25 2021 from 122.181.192.30
ubuntu@ip-172-31-80-128:~$
```

8. Next, switch the user to Jenkins by running the command: **sudo su - jenkins**.



```
ubuntu@ip-172-31-80-128: ~
ubuntu@ip-172-31-80-128:~$ sudo su - jenkins
jenkins@ip-172-31-80-128:~$ pwd
/var/lib/jenkins
jenkins@ip-172-31-80-128:~$ ls
```

9. Now we need to generate the public and private key pairs. So, type the command **ssh-keygen**.

```

 ubuntu@ip-172-31-80-128: ~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/var/lib/jenkins/.ssh/id_rsa):
Created directory '/var/lib/jenkins/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /var/lib/jenkins/.ssh/id_rsa
Your public key has been saved in /var/lib/jenkins/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:4rQq44jGG0doUQrnUUQPZXUDecJwJMkUN8ctvx2+zrI jenkins@ip-172-31-80-12
The key's randomart image is:
+---[RSA 3072]---+
| . o==+BBB++.. |
| .oo. +o++++.. |
| o. . o o |
| o . . . |
| o . o S + . |
| .. o o . o |
| ... o . . |
| .+= . . . |
| +ooo. E+o |
+---[SHA256]---+
jenkins@ip-172-31-80-128: ~$ ls .s

```

10. After running this command you will see that the key pair is generated, as shown in the image above.

11. Now go to the **.ssh** directory by running the command: **ls .ssh/**, and you will see that the private key **id_rsa** and public key **id_rsa.pub** are generated.

```

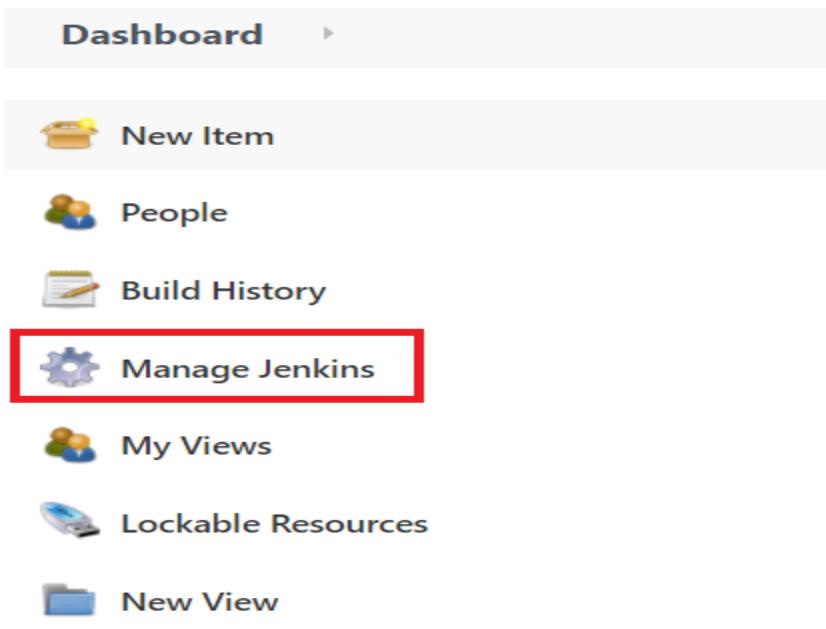
jenkins@ip-172-31-80-128: ~$ ls .ssh/
id_rsa  id_rsa.pub

```

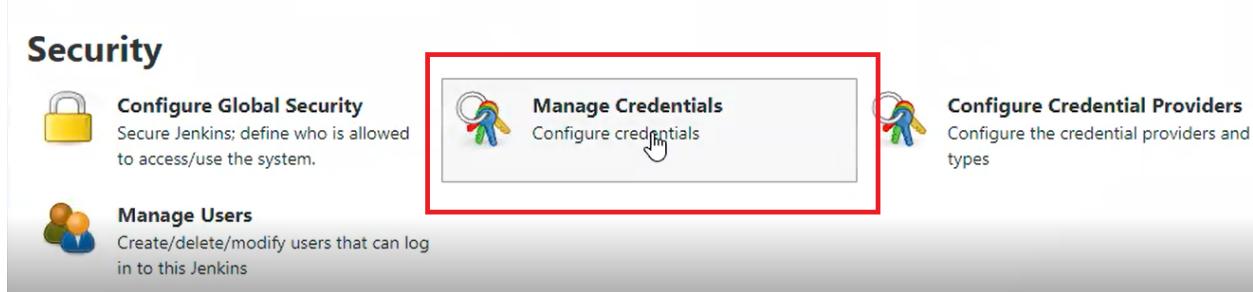
12. Next, go to the **.ssh/** directory by running the command: **cd .ssh/**. Then open the private key **id_rsa**, as shown in the image below, and copy the entire private key.

```
jenkins@ip-172-31-80-128:~/.ssh$ cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzac1rzXktdjEAAAAABG5vbmUAAAAEb9uZQAAAAAAAAAABAAAB1wAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAocRpntmui7HzE11EiaPoXo aTuBDXF69rq0jUQj0fLWQZcNq8
yWn7rolWFYHvSPGGU8weJ77ABLSzXveAR/lpMEAaWfnR8SzODjJgDtmtHwnZWZKK0E1ppL
HIHfjSy4Ocu4YrWKPo8Z1LPQ00UB7NM8HGebsWhTDz7+nSzqZwftwJ/hbnAWi0AOEKRez8
mW9k1V90t9aDQgCLs0Y6xJYMncIDewj+yHf6BuNJIMC+Z9vUng/KH5xKSPDj1NRQfrs23Z
6Fxr1Bo9UKpIaj/Y1xswf8X6Wb4Spq5GwP+vN78dyJUNfENwBvvUvrfNCU4Cu16PvM9iOe
423aUoadpnQBqJZUBgsfs5z2s4xn0JEz2APNkoJEgx/22gPwqgTDNKUfERQBd3Z+CylxiI
a5RMPVSS3q5BhB6gGn1TeMISAzUMPVNb+ZaPnGSrJpu4bAx3UdleT7k8nvKibt/21jo1Hb
O+7Wmx78gBSL20drsbXd9wSvcSNO3M7/fR8Wvrd1AAAFkN7PQ0bez0NGAAAAB3NzaC1yc2
EAAAGBAKHEbaZ7rouxx8xCJRJRImj6F9Gk7gQ1xeva6tI1EI9Hy1kGXDaVmlp+66JVhWB
70jxh1PMHie+wAS0s173gEf6TBAG1n50fEszg4yYA7zrR8J2VmSitBNaaSxyB340suDnL
uGK1ij6PGZsZ0DtFAezTPBxnm0loUw8+/p0s6mcH7cCf4W5wFotADhCkXs/J1vZJFvTrfW
g0IAi7NGOsSWDJ3CA3sI/sh3+gbjSSDAvmfb1J4Pyh+cSkjw49TUUh67Nt2ehca9QaPVCq
SGo/2NcbMH/F+lm+EgauRsD/rze/HciVDXxDcAb71L63zQ1oArpej7zPYjnuNt21KGnaZ0
AYCWVAYLH70c9rOMZ9CRM9gDzZKCRIMf9toD8KoEwzS1HxEUAXd2fgstYssGuUTD1Ukt6u
QYQeoBp9U3jCEgM1DD1TW/mWj5xkqyabuGwMd1HZXk+5PJ7yom7f9pYztR2zvu1pse/IAU
i9tHa7G13fcEr3EjTtzO/30fFlaw9QAAAAMBAEAAAGAFDpex7mYj+F3jsPv4bMDuf5f+U
H3CMY6T0D1QsbZm+rOsjtZHw6LGtiA20NbG1djF4j0FhhpEOk+U5hQ8agbR2Ptmtx5NZUS
lojWv1m3P0Qi7MWrgxLa/s8fTbPZzT4PBdnQSHavhucAIu3bRabn7jjbygQKEH/vbno/lp
lzf dm8KXDUYwx0GG4CTDXxtgS7VWhjeHYxfrv5N1ChJ2q5wIlOzI9Njh5q1aCDCEd1hn3z
```

13. Then go to the **Jenkins Dashboard** and click on **Manage Jenkins**.

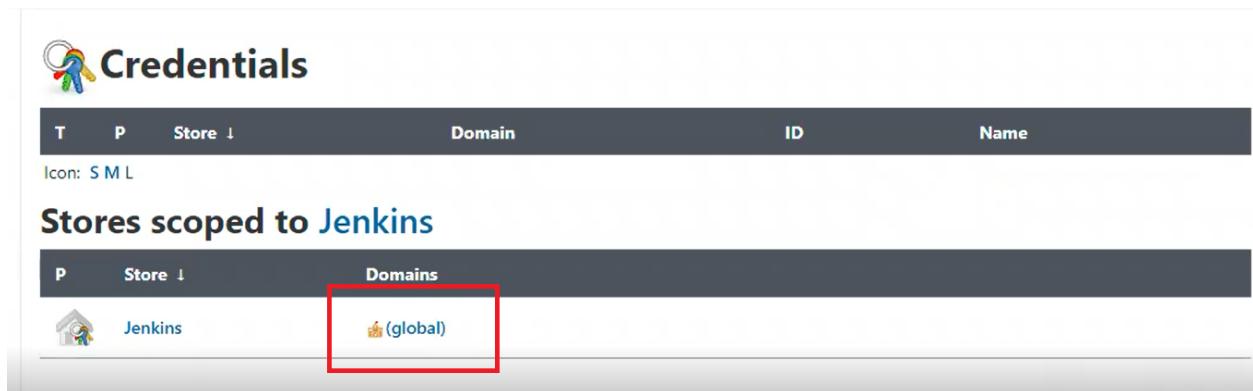


14. Click on **Manage Credentials** under Security.



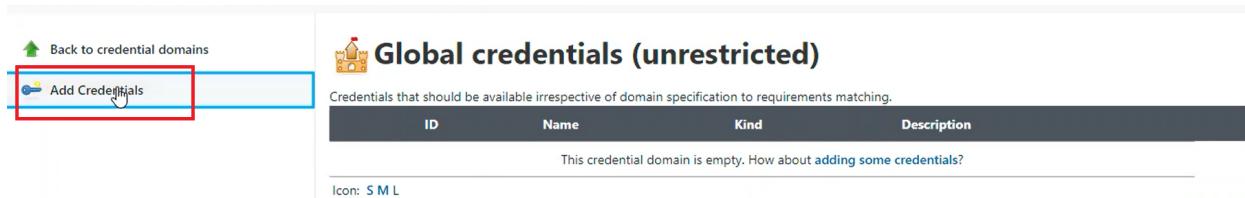
The screenshot shows the Jenkins Security interface. It includes three main options: 'Configure Global Security' (with a lock icon), 'Manage Credentials' (with a key icon), and 'Configure Credential Providers' (with an elephant icon). The 'Manage Credentials' option is highlighted with a red box and a mouse cursor is hovering over it.

15. Click on **global** under the Domains section, as shown in the image below.



The screenshot shows the Jenkins Credentials page. It has a header with tabs 'T', 'P', 'Store ↓', 'Domain', 'ID', and 'Name'. Below the header, it says 'Icon: S M L'. The main section is titled 'Stores scoped to Jenkins' and shows a table with columns 'P', 'Store ↓', and 'Domains'. A row for 'Jenkins' has a 'Domains' field containing '(global)', which is highlighted with a red box.

16. Then click on the **Add Credentials** option.



The screenshot shows the 'Global credentials (unrestricted)' page. At the top left, there's a link 'Back to credential domains' and a button 'Add Credentials' (highlighted with a red box). The main area displays a table with columns 'ID', 'Name', 'Kind', and 'Description'. A message at the bottom says 'This credential domain is empty. How about adding some credentials?' and 'Icon: S M L'.

17. Select the type as **Kind** and choose **SSH Username with private key** from the drop-down menu.



18. Provide the necessary details for **Scope**, **ID**, **Description** and **Username**, as shown in the image below.



The screenshot shows the Jenkins configuration form for 'SSH Username with private key'. The fields are as follows:

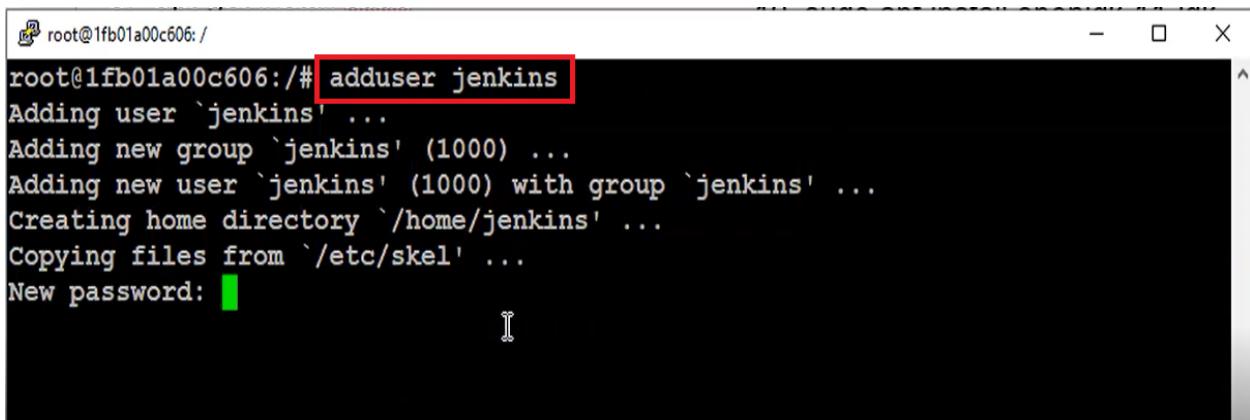
- Scope:** Global (Jenkins, nodes, items, all child items, etc)
- ID:** Master_Jenkins_Private_key
- Description:** Jenkins Master Private Key to Add Multiple Agents
- Username:** jenkins

Each of the four fields (Scope, ID, Description, and Username) is highlighted with a red rectangular box.

19. Now scroll down and in the **Private Key** option, select **Enter directly**.
Here, paste the private key that we had copied earlier.



20. Click on **OK** to confirm.
21. Now we need to build a **Jenkins** user in the Ubuntu container, which is the slave machine running in the first window.
22. Type the following command to create the user: **adduser jenkins**



```
root@1fb01a00c606:/# adduser jenkins
Adding user `jenkins' ...
Adding new group `jenkins' (1000) ...
Adding new user `jenkins' (1000) with group `jenkins' ...
Creating home directory `/home/jenkins' ...
Copying files from `/etc/skel' ...
New password:
```

23. You will be prompted for a password; enter a password of your choice.
24. Next, we must add this user to the sudo users list. So, run the following command: **usermod -aG sudo jenkins**.



```
jenkins@1fb01a00c606: ~
root@1fb01a00c606:/# usermod -aG sudo jenkins
```

25. Type in the command **apt-get update**.

```
root@1fb01a00c606:/# apt-get update
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease
Get:2 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
Get:3 http://security.ubuntu.com/ubuntu focal-security/multiver:
[21.6 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security/universe
682 kB]
Get:5 http://security.ubuntu.com/ubuntu focal-security/main amd64
kB]
Get:6 http://security.ubuntu.com/ubuntu focal-security/restricted
[107 kB]
```

26. Then run the command **apt-get install sudo**.

```
root@1fb01a00c606:/# apt-get install sudo
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
  sudo
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 514 kB of archives.
After this operation, 2257 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 s
1-lubuntu1.2 [514 kB]
0% [1 sudo 0 B/514 kB 0%]
```

27. Now, switch to Jenkins user by running the command **su - jenkins**.

```
root@1fb01a00c606:/# su - jenkins
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.
```

28. Next, we need to install the openSSH server. Type in **sudo apt-get**

install openssh-server to install this server. After that, you will be prompted for a password; enter the password that you created earlier.

```
jenkins@1fb01a00c606:~$ sudo apt-get install openssh-server
[sudo] password for jenkins:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
```

29. You will be asked to provide a **geographical area** and **time zone**.

Provide the inputs similar to those shown in the image below.

```
scd:
1. Africa   3. Antarctica  5. Arctic   7. Atlantic  9. Indian    11. SystemV  1
3. Etc
2. America  4. Australia   6. Asia     8. Europe    10. Pacific   12. US
Geographic area: 9

Please select the city or region corresponding to your time zone.

1. Antananarivo  3. Christmas  5. Comoro     7. Mahe       9. Mauritius 11. Reunion
2. Chagos        4. Cocos      6. Kerguelen  8. Maldives   10. Mayotte
Time zone: 8
```

30. Now you need to restart the SSH service. So, type the following

command: **sudo service ssh restart**.

```
jenkins@1fb01a00c606:~$ sudo service ssh restart
 * Restarting OpenBSD Secure Shell server sshd
jenkins@1fb01a00c606:~$ sudo service ssh status
 * sshd is running
jenkins@1fb01a00c606:~$ █
```

31. You can also check the status by typing in the command **sudo service ssh status**.

32. Next, we need to install Java. Type the command: **sudo apt install openjdk-11-jdk** to install Java.

```
jenkins@1fb01a00c606:~$ sudo apt install openjdk-11-jdk
Reading package lists... Done
Building dependency tree
Reading state information... Done
```

33. Now return back to the second window, which is the master instance.

34. Type **ssh-copy-id jenkins@ip_address** in the command prompt and run the command.

```
jenkins@ip-172-31-80-128:~/.ssh$ ssh-copy-id jenkins@172.17.0.2
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/var/lib/jenkins/.ssh/id_rsa.pub"
The authenticity of host '172.17.0.2 (172.17.0.2)' can't be established.
ECDSA key fingerprint is SHA256:ASdj4ORo3CToI96CgkS94VnId6tYrwNB/MNCKh8hQoI.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
```

35. This command will copy the public key in the Ubuntu container, which is running on the first window.

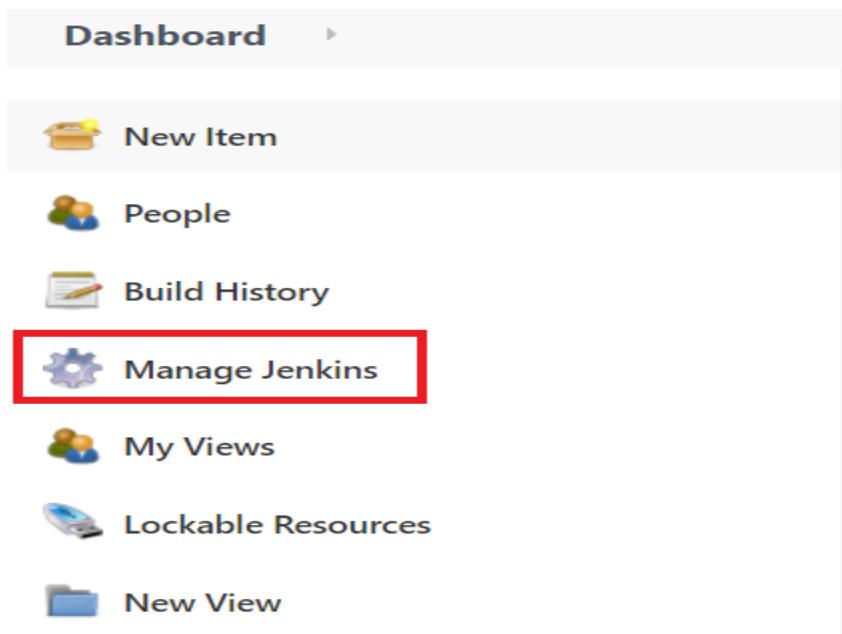
36. Go back to the Ubuntu container and examine the contents of the **.ssh** directory.

37. You will see the **authorized_keys** file, which is the same as the public key on the host machine. This is depicted in the image below.

```

jenkins@1fb01a00c606:~/.ssh
jenkins@1fb01a00c606:~$ ls -a
. . . .bash_history .bash_logout .bashrc .cache .profile .ssh .sudo_as_admin_successful
jenkins@1fb01a00c606:~$ cd .ssh
jenkins@1fb01a00c606:~/ssh$ ls
authorized_keys
jenkins@1fb01a00c606:~/ssh$ cat authorized_keys
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQChxGzme2aoLscfMQiUSUSJo+hfRpO4ENcXr2urSNRCPR8tZBlw2rzJafuu
BH8ukwQBpZ+dHxLM4OMmA02a0fCd1ZkorQTWmkscgd+NLLg5y7hitYo+jxmUs9A7RQHs0zwc25tJaFMPPv6dLOpnB+3An+Fu
AIuzRjrElgydwgN7CP7Id/oG40kgwI5n29SeD8ofnEpI8OPU1FB+uzbdnoXGvUGj1QqkhqP9jXGzB/xfpZvhKmrkbA/683vx
z2I57jbdpShp2mdAGA11QGCx+znPazjGfQkTPYA82SgksDH/baA/CqBMM0pR8RFAF3dn4LLWEhrlEw9VJLerkGEHqAafVN4
dR2V5PuTye8qJu3/aWM7Uds77tabHvyAFIvbR2uxtd33BK9xi07czv99HxZWsPU= jenkins@ip-172-31-80-128
jenkins@1fb01a00c606:~/ssh$ 
  
```

38. Now, return back to the **Jenkins Dashboard** and click on **Manage Jenkins**.



39. Then click on **Manage Nodes and Clouds** under System Configuration.

System Configuration



Configure System
Configure global settings and paths.



Global Tool Configuration
Configure tools, their locations and automatic installers.

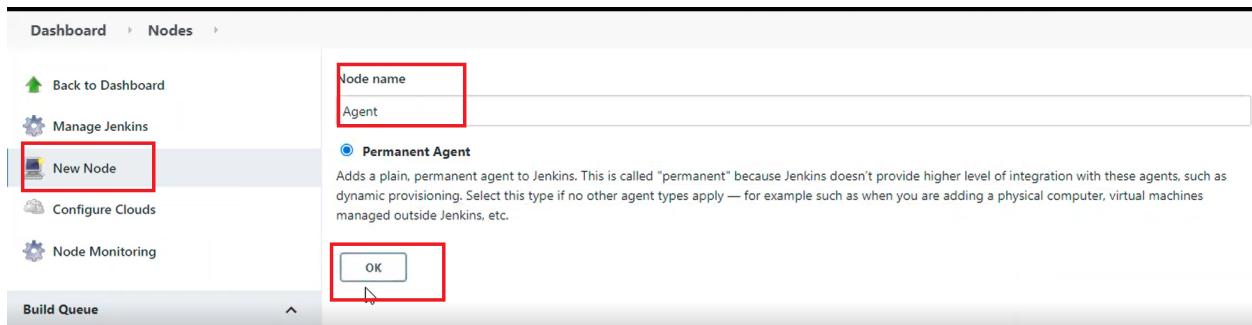


Manage Plugins
Add, remove, disable or enable plugins that can extend the functionality of Jenkins.



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

40. Now click on **New Node**.

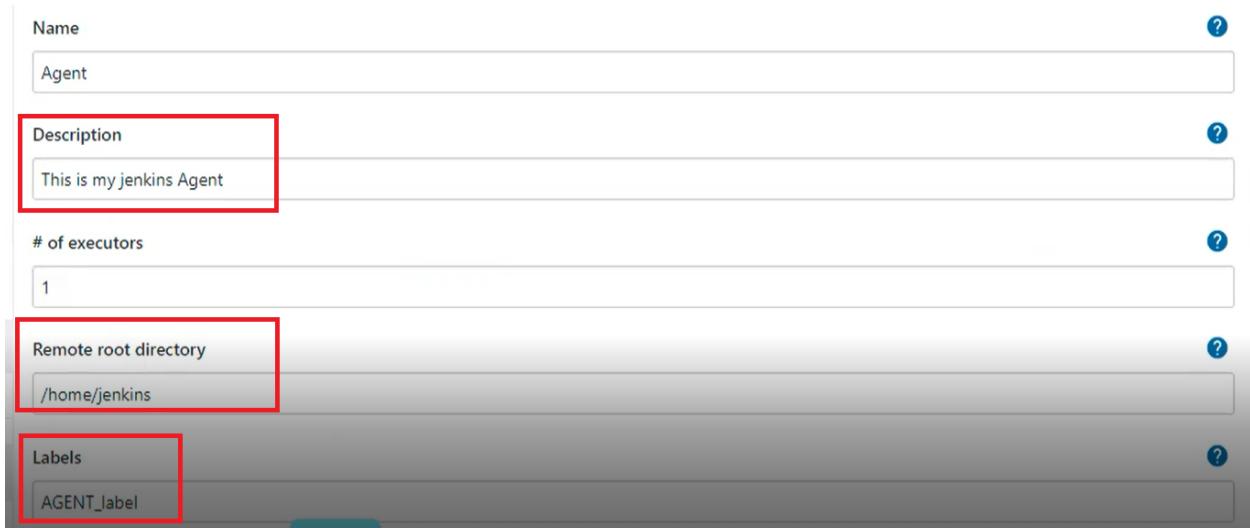


The screenshot shows the Jenkins 'Nodes' configuration screen. On the left, there's a sidebar with links: 'Back to Dashboard', 'Manage Jenkins', 'New Node' (which is highlighted with a red box), 'Configure Clouds', and 'Node Monitoring'. The main area has a 'Node name' input field containing 'Agent', a radio button for 'Permanent Agent' (selected), and an 'OK' button at the bottom right which is also highlighted with a red box.

41. Here, you will be prompted to enter a **Node Name**.

42. Then select **Permanent Agent** and click **OK**.

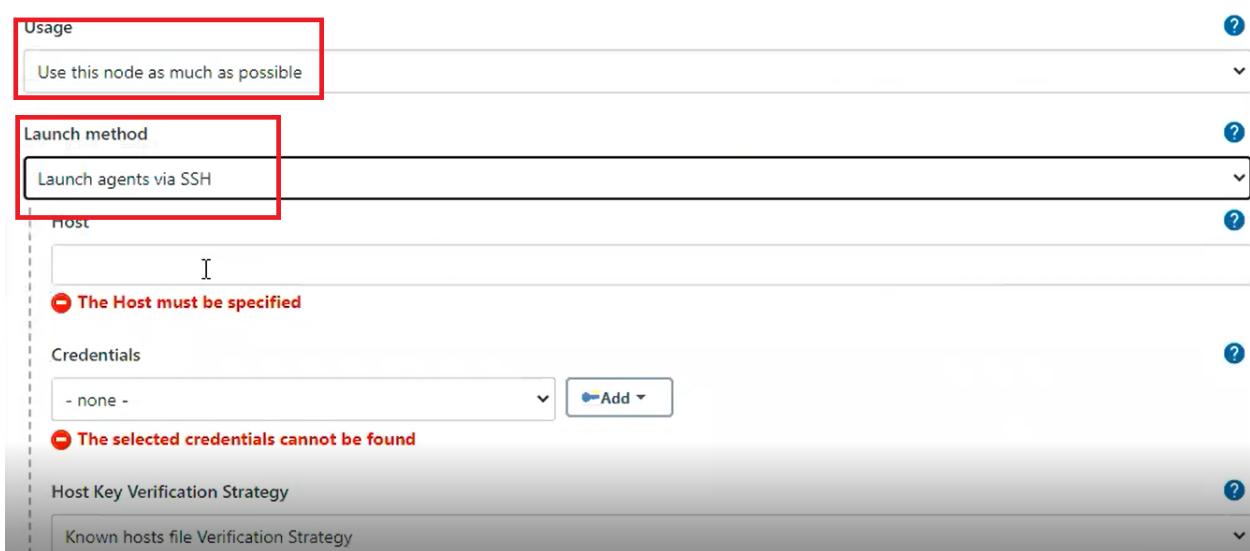
43. Provide the details for the **Description**, **Remote root directory** and **Labels** for the node, as shown below.



The screenshot shows the configuration page for a new Jenkins node. The fields are as follows:

- Name:** Agent
- Description:** This is my jenkins Agent
- # of executors:** 1
- Remote root directory:** /home/jenkins
- Labels:** AGENT_label

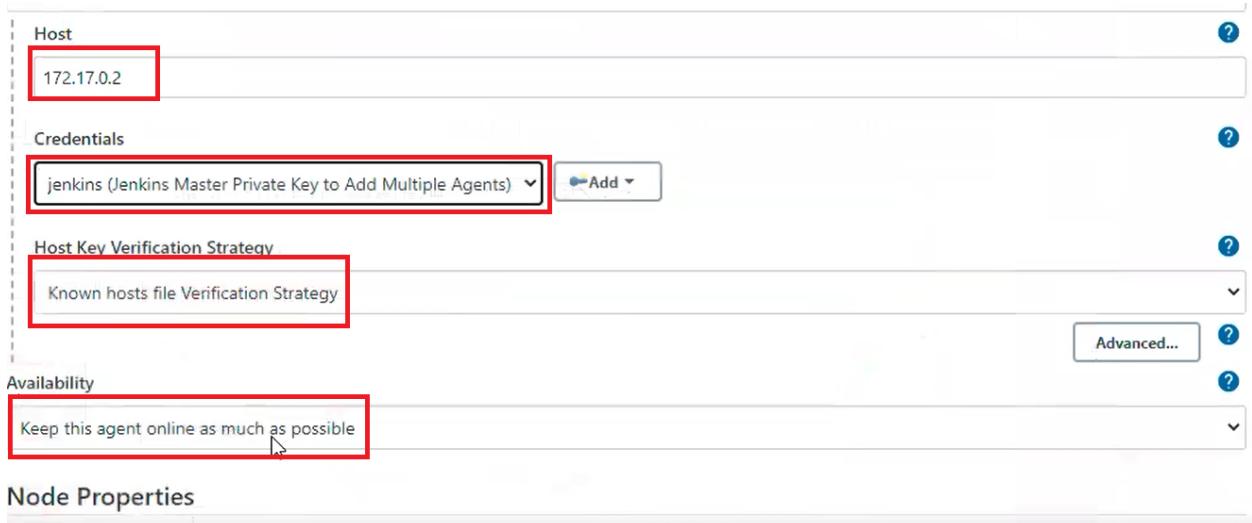
44. In the **Usage** column, select the **Use this node as much as possible** option.
45. In the **Launch method** column, select the **Launch agents via SSH** option.



The screenshot shows the configuration page for a new Jenkins node. The fields are as follows:

- Usage:** Use this node as much as possible
- Launch method:** Launch agents via SSH
- Host:** (Field is empty and has an error message: "The Host must be specified")
- Credentials:** - none - (has an error message: "The selected credentials cannot be found")
- Host Key Verification Strategy:** Known hosts file Verification Strategy

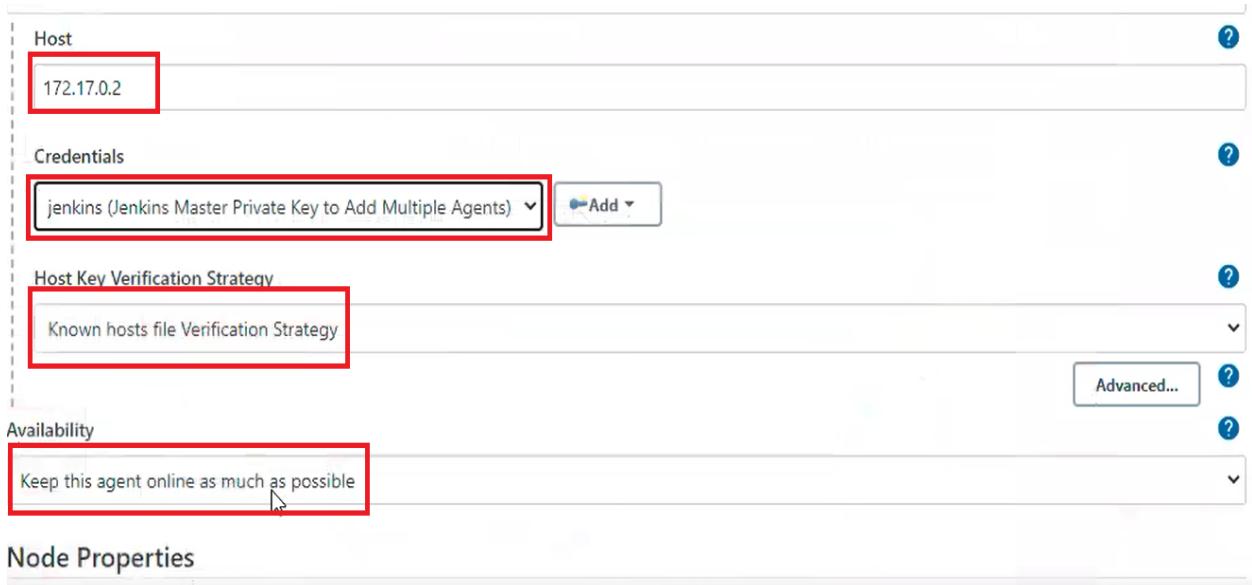
46. Next, we need to provide the **hostname**, i.e., ip_address of the slave Ubuntu container, which is our slave machine.



The screenshot shows the Jenkins Node configuration interface. The fields are as follows:

- Host:** 172.17.0.2
- Credentials:** jenkins (Jenkins Master Private Key to Add Multiple Agents) (highlighted with a red box)
- Host Key Verification Strategy:** Known hosts file Verification Strategy (highlighted with a red box)
- Availability:** Keep this agent online as much as possible (highlighted with a red box)

47. In the **Credentials**, select the **Jenkins Master Private key to Add Multiple Agents** option.
48. In the **Host Key Verification Strategy** column, select the **Known hosts file Verification Strategy** option.
49. In the **Availability** column, select the **Keep this agent online as much as possible** option.



The screenshot shows the Jenkins Node configuration interface. The fields are as follows:

- Host:** 172.17.0.2
- Credentials:** jenkins (Jenkins Master Private Key to Add Multiple Agents) (highlighted with a red box)
- Host Key Verification Strategy:** Known hosts file Verification Strategy (highlighted with a red box)
- Availability:** Keep this agent online as much as possible (highlighted with a red box)

50. Scroll down and click **Save**.

Node Properties

Disable deferred wipeout on this node

Environment variables

Tool Locations

Save



- After a few seconds, you can see that the agent list has been added and is in running condition, as shown in the image below.

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	 Agent	Linux (amd64)	In sync	4.10 GB	- 0 B	4.10 GB	153ms
	 master	Linux (amd64)	In sync	4.10 GB	- 0 B	4.10 GB	0ms
	Data obtained	23 sec	23 sec	23 sec	23 sec	23 sec	23 sec
Refresh status							

- Return back to the **Jenkins Dashboard** and create a new **Freestyle project** named 'AgentTrigger'.

Enter an item name

» Required field

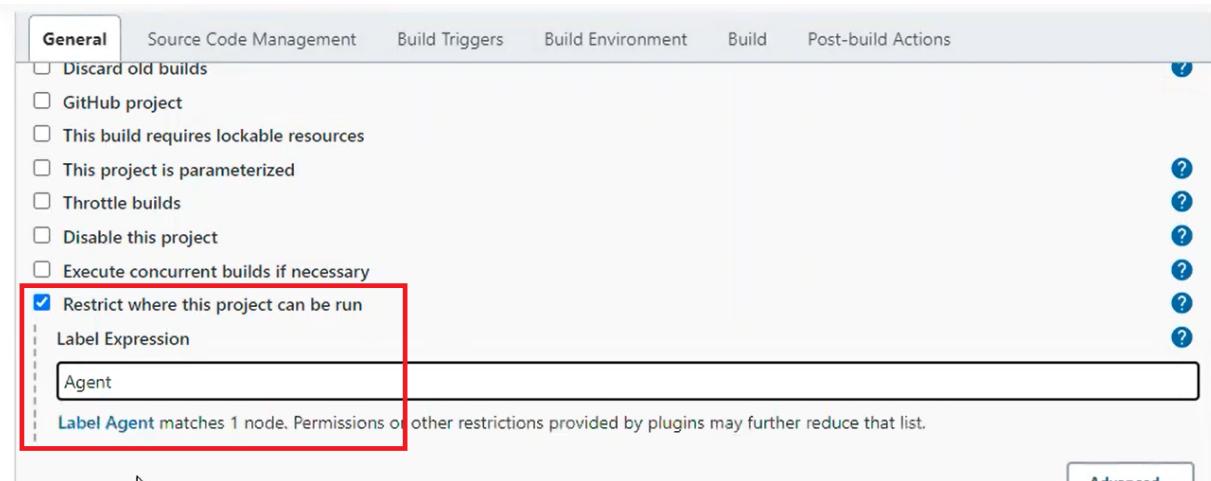
Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
OK

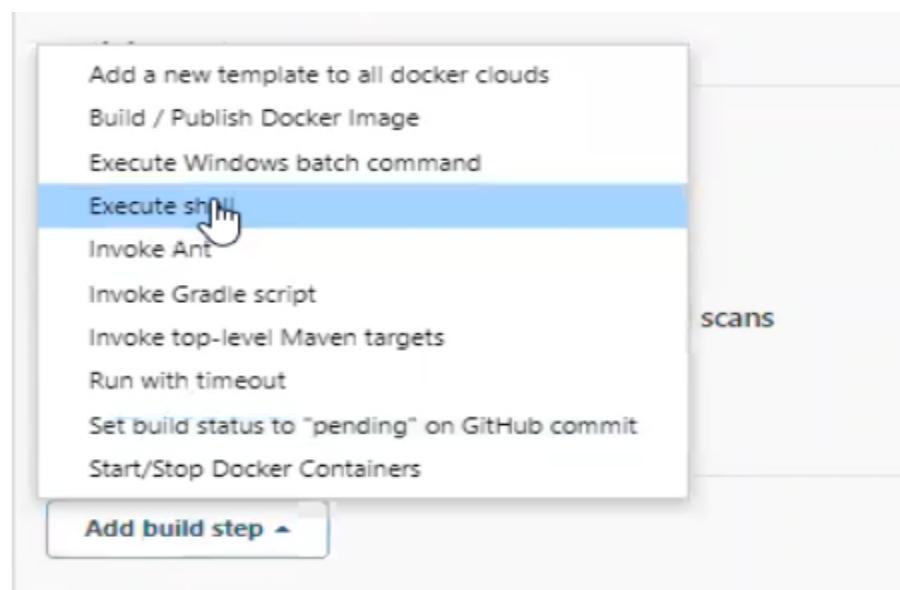
- In the **General** tab, select the option **Restrict where this project can**

be run, and then enter 'Agent' under the **Label Expression** column.



The screenshot shows the Jenkins General configuration page. Under the 'General' tab, there is a section titled 'Restrict where this project can be run'. A checkbox labeled 'Restrict where this project can be run' is checked. Below it, the 'Label Expression' field contains the value 'Agent'. A red box highlights this entire section. At the bottom of the 'Label Expression' field, a tooltip says 'Label Agent matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.'

54. Scroll down to the **Build** section and click on the **Add build Step** option.



The screenshot shows a dropdown menu for selecting a build step. The options listed are: 'Add a new template to all docker clouds', 'Build / Publish Docker Image', 'Execute Windows batch command', 'Execute shell' (which is highlighted with a blue background and a cursor icon), 'Invoke Ant', 'Invoke Gradle script', 'Invoke top-level Maven targets', 'Run with timeout', 'Set build status to "pending" on GitHub commit', and 'Start/Stop Docker Containers'. At the bottom of the dropdown, there is a button labeled 'Add build step ▾'.

55. Select the **Execute shell** option, and then write the script that you want to execute.

Build

Execute shell

Command

```
df -ha
```

See the list of available environment variables

Add build step ▾

Save Apply Advanced...

56. Click on **Apply** and then **Save**.
57. You can see your job getting built, by clicking on **Build Now** on the dashboard.

Apps Consumer Portal ... welcome to iitb.co... :: Om Arunachala ::... Bangalore Log

Dashboard > AgentTrigger >

- Changes
- Workspace
- Build Now**
- Configure
- Delete Project
- Rename

Build History trend ^

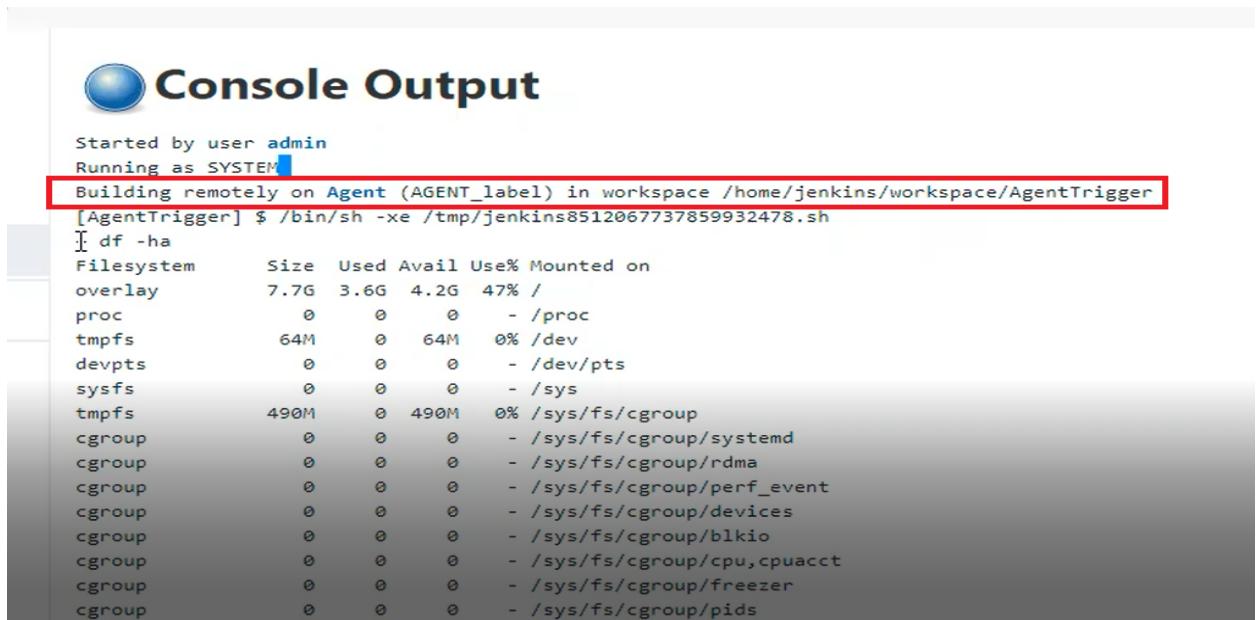
find

#1	Mar 30, 2021, 12:22 PM
----	------------------------

Workspace Recent Changes

Permalinks

58. Once the project is built, click on the build number to view the log history. The slave node was used to build the project.

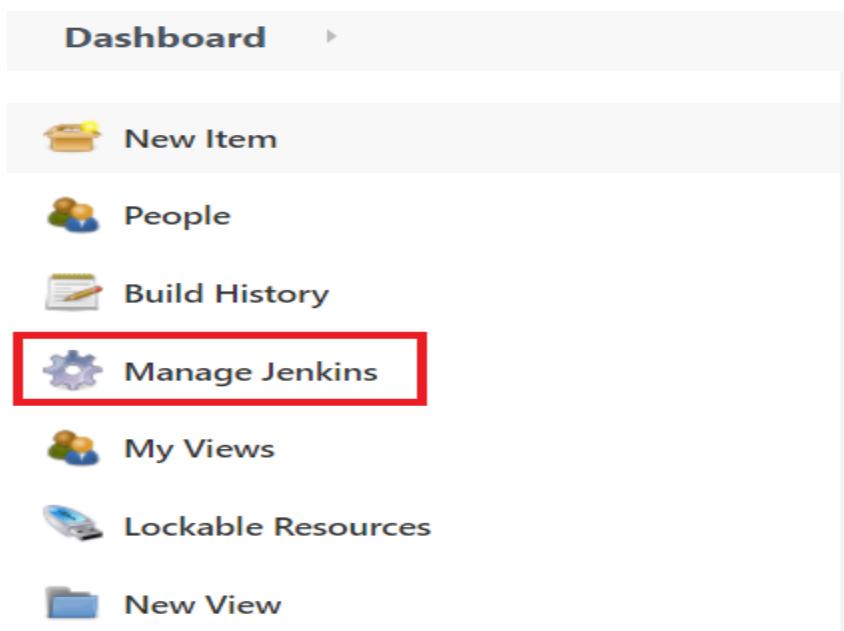


```

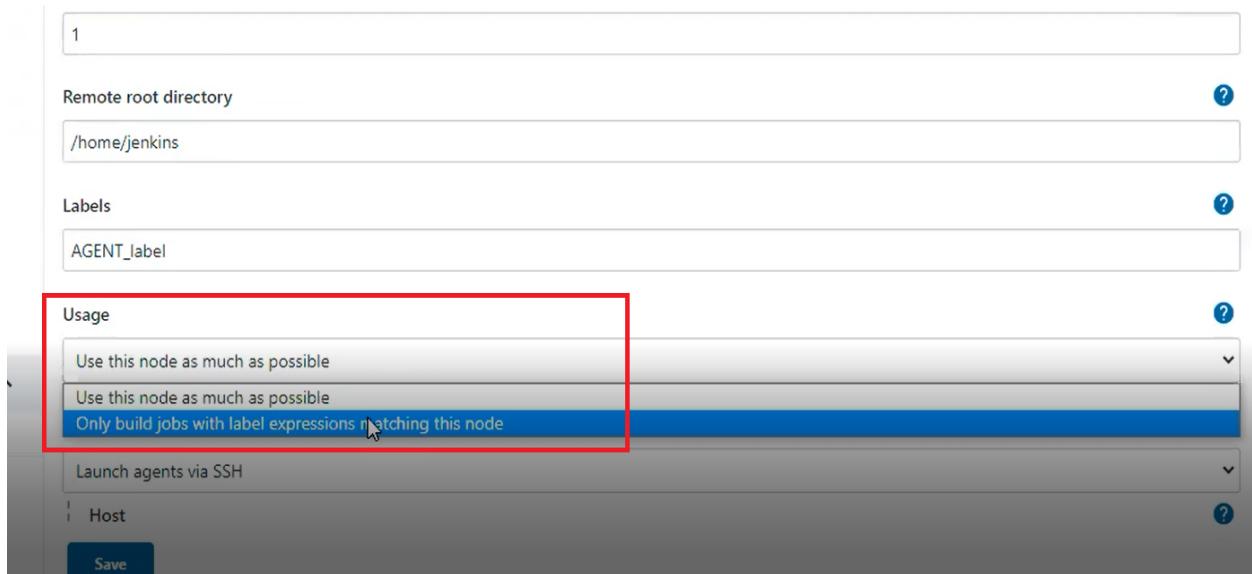
Started by user admin
Running as SYSTEM
Building remotely on Agent (AGENT_label) in workspace /home/jenkins/workspace/AgentTrigger
[AgentTrigger] $ /bin/sh -xe /tmp/jenkins8512067737859932478.sh
[1] df -ha
Filesystem      Size   Used  Avail Use% Mounted on
overlay        7.7G   3.6G   4.2G  47% /
proc            0     0     0    - /proc
tmpfs          64M    0     64M   0% /dev
devpts          0     0     0    - /dev/pts
sysfs          0     0     0    - /sys
tmpfs          490M   0    490M  0% /sys/fs/cgroup
cgroup          0     0     0    - /sys/fs/cgroup/systemd
cgroup          0     0     0    - /sys/fs/cgroup/rdma
cgroup          0     0     0    - /sys/fs/cgroup/perf_event
cgroup          0     0     0    - /sys/fs/cgroup/devices
cgroup          0     0     0    - /sys/fs/cgroup/blkio
cgroup          0     0     0    - /sys/fs/cgroup/cpu,cpuacct
cgroup          0     0     0    - /sys/fs/cgroup/freezer
cgroup          0     0     0    - /sys/fs/cgroup/pids
  
```

59. Now go back to the **Configure** section and disable **Restrict where this project can be run** option.

60. Thereafter, go to **Manage Jenkins**.

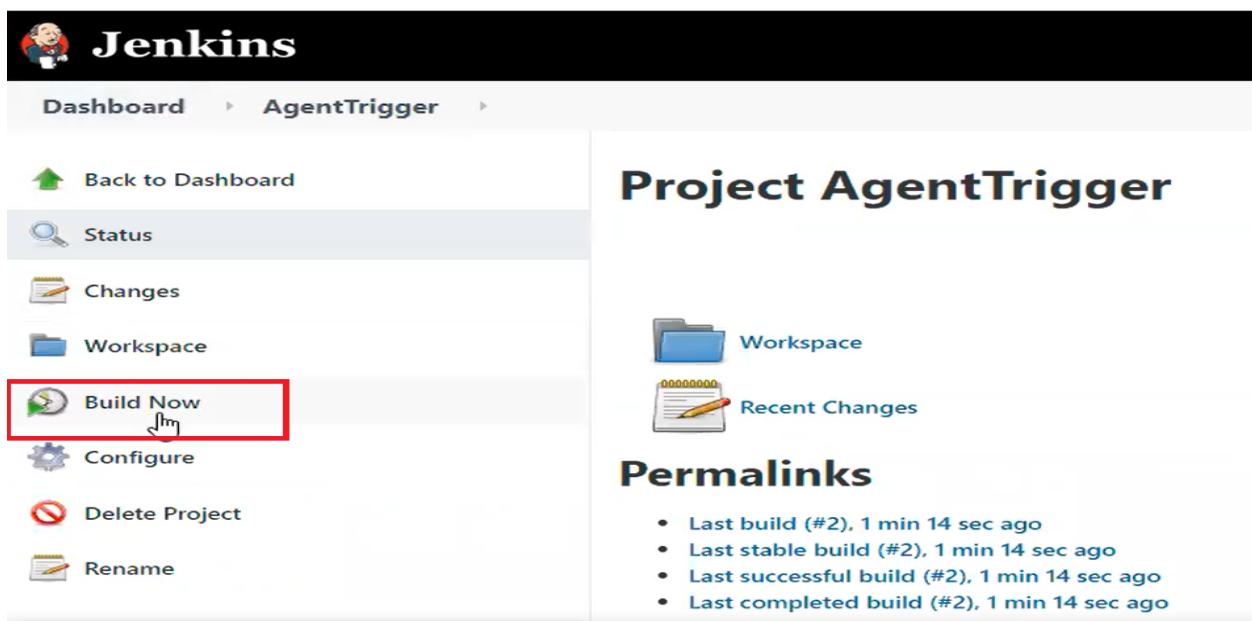


61. Click on **Manage Nodes and Clouds**.
62. Click on **Agent** and then select **Configure**.
63. In the **Usage** column, select the **Only build jobs with label expressions matching this node** option.



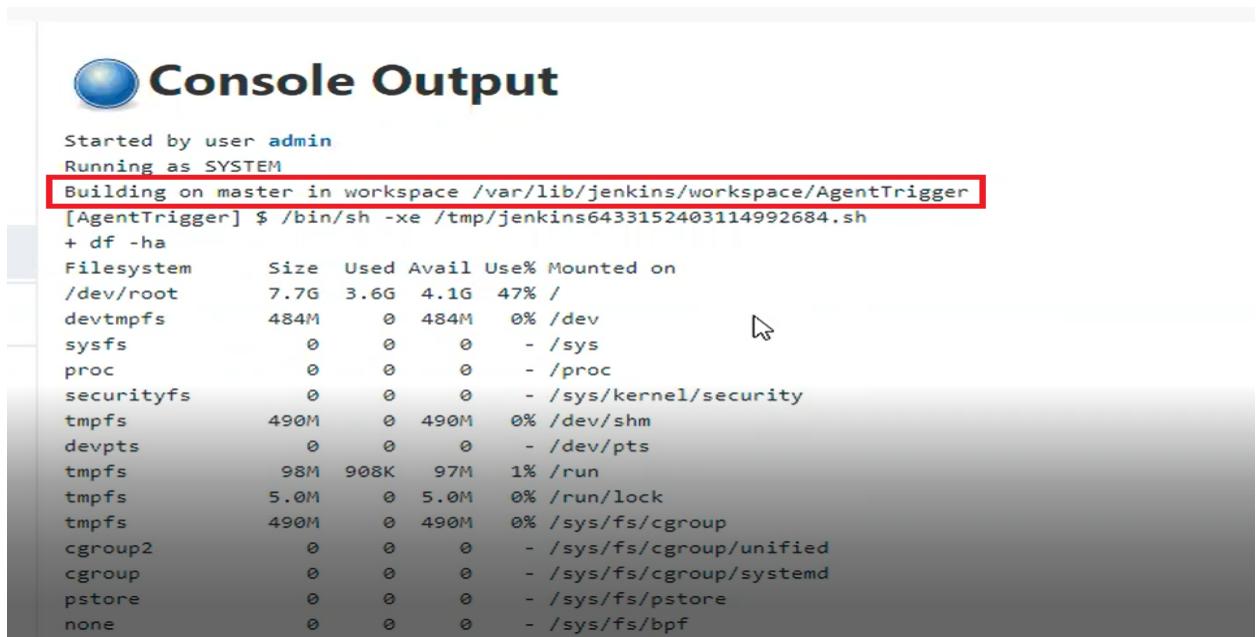
The screenshot shows the Jenkins 'Manage Nodes and Clouds' configuration interface. In the 'Usage' section, three options are listed: 'Use this node as much as possible', 'Use this node as much as possible', and 'Only build jobs with label expressions matching this node'. The third option is highlighted with a blue background and a cursor is hovering over it. A red box highlights the 'Usage' dropdown menu.

64. Click **Save**.
65. Return back to the dashboard and build the job again.



The screenshot shows the Jenkins Project AgentTrigger dashboard. On the left sidebar, there is a 'Build Now' button, which is highlighted with a red box. The main content area displays the 'Project AgentTrigger' title and links for 'Workspace' and 'Recent Changes'.

66. This time the project will be built on the master node, as shown in the image below.



The screenshot shows a Jenkins console output window titled "Console Output". The output starts with "Started by user admin" and "Running as SYSTEM". A red box highlights the line "Building on master in workspace /var/lib/jenkins/workspace/AgentTrigger". Below this, the command "[AgentTrigger] \$ /bin/sh -xe /tmp/jenkins6433152403114992684.sh" is shown, followed by the output of the "df -ha" command. The output shows disk usage for various filesystems like /dev/root, devtmpfs, sysfs, proc, securityfs, tmpfs, devpts, tmpfs, tmpfs, cgroup2, cgroup, pstore, and none. The /dev/root filesystem is mounted at / with a size of 7.7G, used space of 3.6G, available space of 4.1G, and a usage percentage of 47%.

```
Started by user admin
Running as SYSTEM
Building on master in workspace /var/lib/jenkins/workspace/AgentTrigger
[AgentTrigger] $ /bin/sh -xe /tmp/jenkins6433152403114992684.sh
+ df -ha
Filesystem      Size  Used Avail Use% Mounted on
/dev/root        7.7G  3.6G  4.1G  47% /
devtmpfs       484M    0  484M   0% /dev
sysfs          0     0     0   -  /sys
proc            0     0     0   -  /proc
securityfs      0     0     0   -  /sys/kernel/security
tmpfs           490M    0  490M   0% /dev/shm
devpts          0     0     0   -  /dev/pts
tmpfs           98M  908K  97M   1% /run
tmpfs           5.0M    0  5.0M   0% /run/lock
tmpfs           490M    0  490M   0% /sys/fs/cgroup
cgroup2          0     0     0   -  /sys/fs/cgroup/unified
cgroup          0     0     0   -  /sys/fs/cgroup/systemd
pstore          0     0     0   -  /sys/fs/pstore
none            0     0     0   -  /sys/fs/bpf
```