

Jenkins Master Slave Configuration Project

A Jenkins master comes with the basic installation of Jenkins and in this configuration the master handles all the tasks for building the system. Jenkins uses a Master-Slave architecture to manage distributed builds. In this architecture, Master and slave nodes communicate through TCP/IP protocol. The main Jenkins server acts as the Master node that manages the slave.

Jenkins slaves connects to the Jenkins master using the Java Network Launch protocol.

Below are the steps for configuration of Jenkins Master and Slave Configuration :-

Step1 : Create 2 servers in the same account region

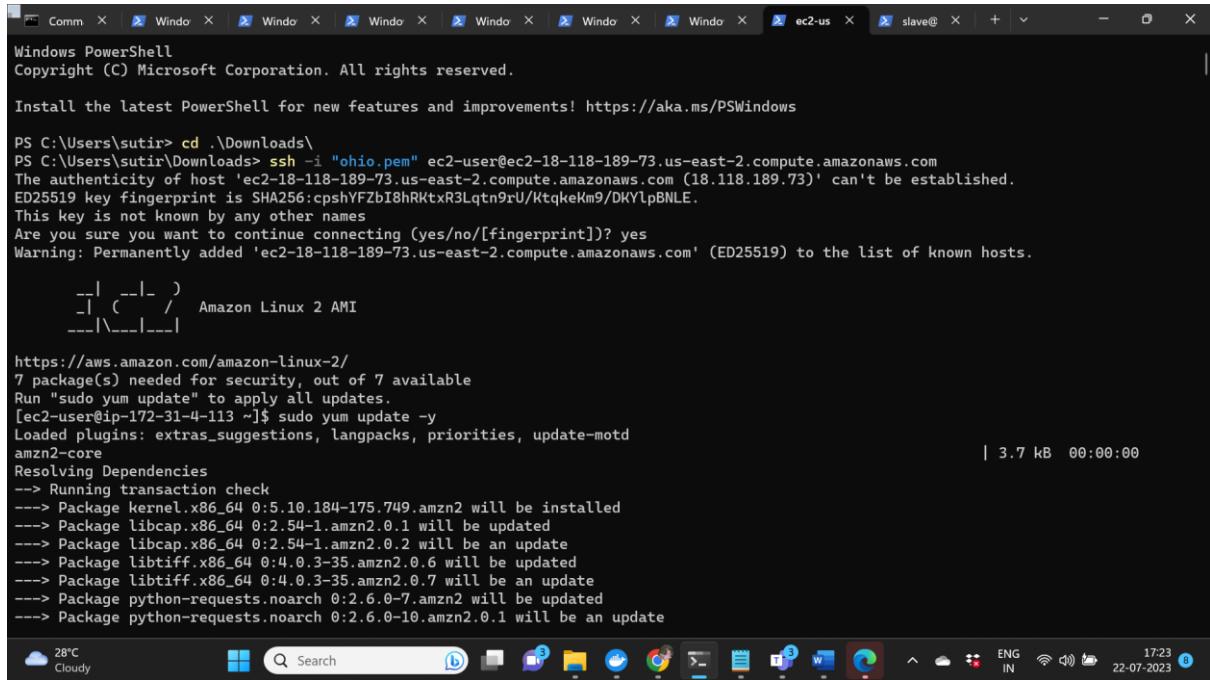
Master EC2 Instance

The screenshot shows the AWS Management Console EC2 Instances page. On the left sidebar, under 'Instances', 'Instances' is selected. The main table lists two instances: 'master' (running, t2.medium, 2/2 checks passed) and 'slave' (running, t2.medium, 2/2 checks passed). The 'master' instance is selected. The right pane displays the details for the 'master' instance, including its Public IPv4 address (18.118.189.73) which is highlighted with a green border and a tooltip: 'Public IPv4 address copied'. Other details shown include Instance ID (i-08c98363790d364b4), Instance state (Running), Instance type (t2.medium), Status check (2/2 checks passed), and Alarm status (No alarms).

Step2: Create a master in one server with default connections and install Jenkins configuration.

The screenshot shows the AWS Management Console Security Groups page. Under 'Security Groups', 'sg-0286716915e1a136e - launch-wizard-9' is selected. The 'Edit inbound rules' section is displayed. It shows two rules: 1. A rule for 'SSH' (Protocol: TCP, Port range: 22, Source: Custom, 0.0.0.0/0) with a 'Delete' button. 2. A rule for 'Custom TCP' (Protocol: TCP, Port range: 8080, Source: Custom) with a 'Delete' button. A tooltip 'Selected' appears over the 8080 rule's source field. Below the table is a 'Add rule' button.

Launch the master server and apply all the updates required in the server.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

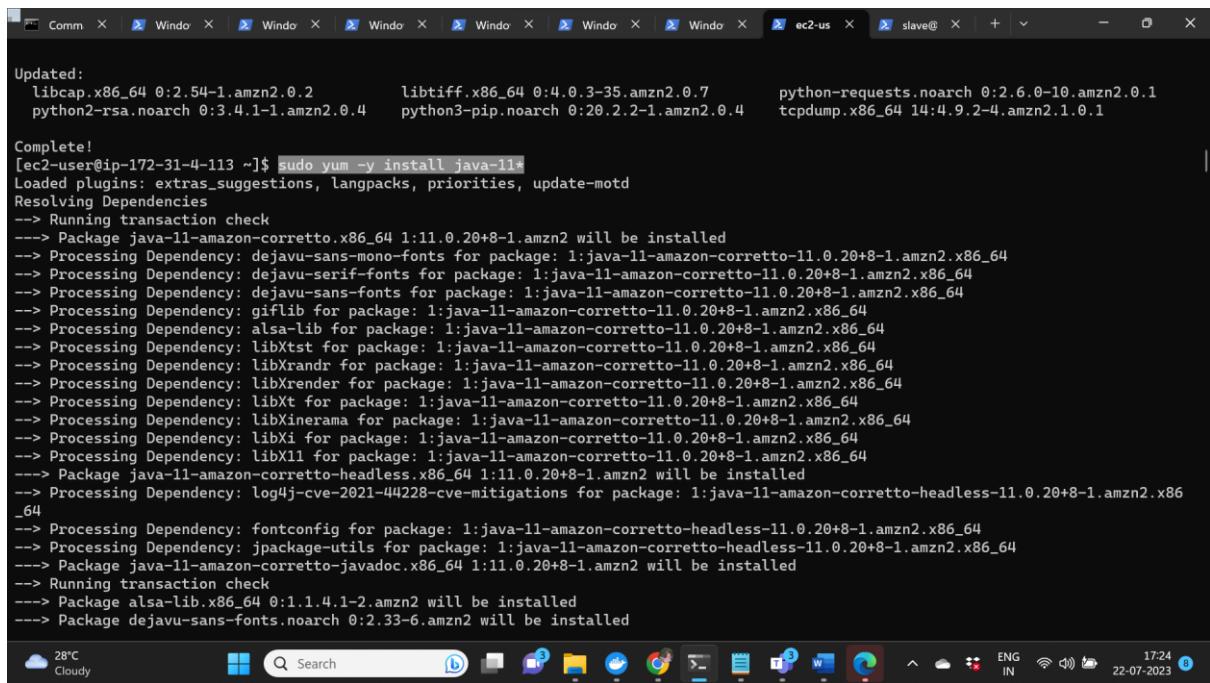
PS C:\Users\sutir> cd .\Downloads\
PS C:\Users\sutir\Downloads> ssh -i "ohio.pem" ec2-user@ec2-18-118-189-73.us-east-2.compute.amazonaws.com
The authenticity of host 'ec2-18-118-189-73.us-east-2.compute.amazonaws.com (18.118.189.73)' can't be established.
ED25519 key fingerprint is SHA256:cphsYF2bIBhRKtxR3LqtN9rU/Ktqkem9/DKyLpBNLE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-18-118-189-73.us-east-2.compute.amazonaws.com' (ED25519) to the list of known hosts.

__| __|_ )
_| ( _ /   Amazon Linux 2 AMI
___|_\__|__|_

https://aws.amazon.com/amazon-linux-2/
7 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-4-113 ~]$ sudo yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
----> Package kernel.x86_64 0:5.10.184-175.749.amzn2 will be installed
----> Package libcap.x86_64 0:2.54-1.amzn2.0.1 will be updated
----> Package libcap.x86_64 0:2.54-1.amzn2.0.2 will be an update
----> Package libtiff.x86_64 0:4.0.3-35.amzn2.0.6 will be updated
----> Package libtiff.x86_64 0:4.0.3-35.amzn2.0.7 will be an update
----> Package python-requests.noarch 0:2.6.0-7.amzn2 will be updated
----> Package python-requests.noarch 0:2.6.0-10.amzn2.0.1 will be an update
| 3.7 kB  00:00:00

Cloudy 28°C Search 17:23 IN 22-07-2023
```

Install Java in the master server

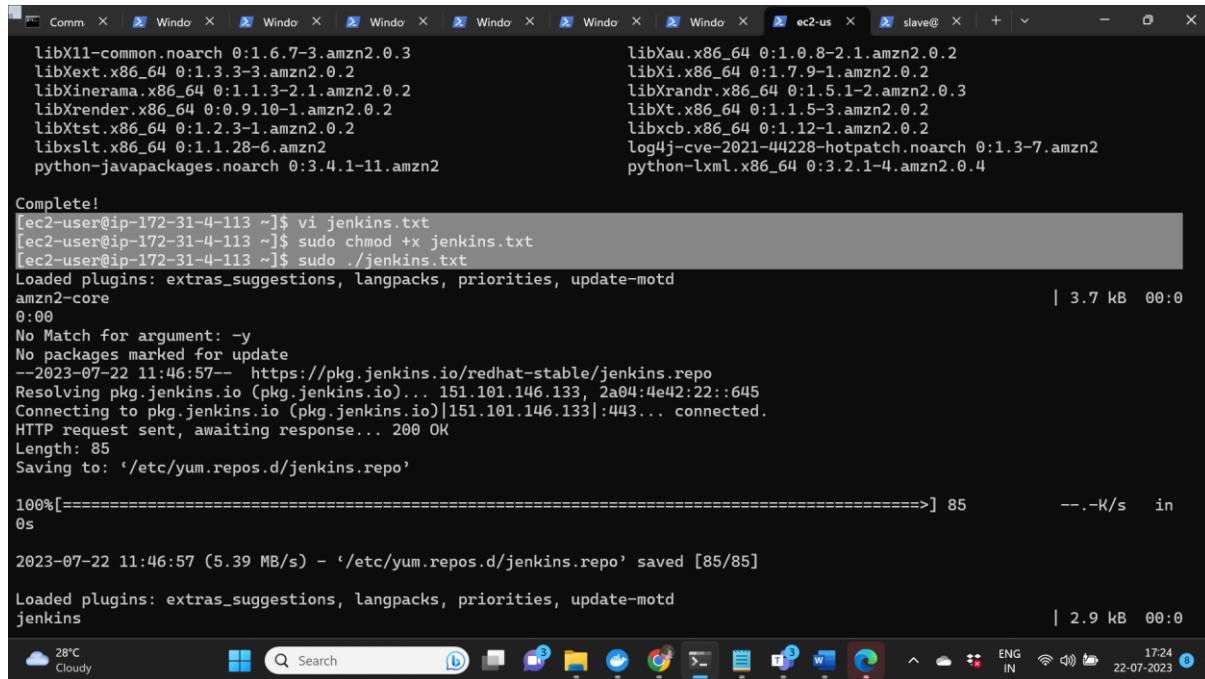


```
Updated:
libcap.x86_64 0:2.54-1.amzn2.0.2      libtiff.x86_64 0:4.0.3-35.amzn2.0.7      python-requests.noarch 0:2.6.0-10.amzn2.0.1
python2-rsa.noarch 0:3.4.1-1.amzn2.0.4    python3-pip.noarch 0:20.2.2-1.amzn2.0.4    tcpdump.x86_64 14:4.9.2-4.amzn2.1.0.1

Complete!
[ec2-user@ip-172-31-4-113 ~]$ sudo yum -y install java-11*
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
----> Package java-11-amazon-corretto.x86_64 1:11.0.20+8-1.amzn2 will be installed
----> Processing Dependency: dejavu-sans-mono-fonts for package: 1:java-11-amazon-corretto-11.0.20+8-1.amzn2.x86_64
----> Processing Dependency: dejavu-serif-fonts for package: 1:java-11-amazon-corretto-11.0.20+8-1.amzn2.x86_64
----> Processing Dependency: dejavu-sans-fonts for package: 1:java-11-amazon-corretto-11.0.20+8-1.amzn2.x86_64
----> Processing Dependency: giflib for package: 1:java-11-amazon-corretto-11.0.20+8-1.amzn2.x86_64
----> Processing Dependency: alsalib for package: 1:java-11-amazon-corretto-11.0.20+8-1.amzn2.x86_64
----> Processing Dependency: libXtst for package: 1:java-11-amazon-corretto-11.0.20+8-1.amzn2.x86_64
----> Processing Dependency: libXrandr for package: 1:java-11-amazon-corretto-11.0.20+8-1.amzn2.x86_64
----> Processing Dependency: libXrender for package: 1:java-11-amazon-corretto-11.0.20+8-1.amzn2.x86_64
----> Processing Dependency: libXt for package: 1:java-11-amazon-corretto-11.0.20+8-1.amzn2.x86_64
----> Processing Dependency: libXinerama for package: 1:java-11-amazon-corretto-11.0.20+8-1.amzn2.x86_64
----> Processing Dependency: libXi for package: 1:java-11-amazon-corretto-11.0.20+8-1.amzn2.x86_64
----> Processing Dependency: libX11 for package: 1:java-11-amazon-corretto-11.0.20+8-1.amzn2.x86_64
----> Package java-11-amazon-corretto-headless.x86_64 1:11.0.20+8-1.amzn2 will be installed
----> Processing Dependency: log4j-cve-2021-44228-cve-mitigations for package: 1:java-11-amazon-corretto-headless-11.0.20+8-1.amzn2.x86_64
----> Processing Dependency: fontconfig for package: 1:java-11-amazon-corretto-headless-11.0.20+8-1.amzn2.x86_64
----> Processing Dependency: jpackage-utils for package: 1:java-11-amazon-corretto-headless-11.0.20+8-1.amzn2.x86_64
----> Package java-11-amazon-corretto-javadoc.x86_64 1:11.0.20+8-1.amzn2 will be installed
--> Running transaction check
----> Package alsalib.x86_64 0:1.1.4.1-2.amzn2 will be installed
----> Package dejavu-sans-fonts.noarch 0:2.33-6.amzn2 will be installed
Cloudy 28°C Search 17:24 IN 22-07-2023
```

Create a txt file and load all the Jenkins installation commands.

Provide all the permissions to the file and execute the txt file by using sudo command.



```
libX11-common.noarch 0:1.6.7-3.amzn2.0.3
libXext.x86_64 0:1.3.3-3.amzn2.0.2
libXinerama.x86_64 0:1.1.3-2.1.amzn2.0.2
libXrender.x86_64 0:0.9.10-1.amzn2.0.2
libXtst.x86_64 0:1.2.3-1.amzn2.0.2
libXslt.x86_64 0:1.1.28-6.amzn2
python-javapackages.noarch 0:3.4.1-11.amzn2
python-lxml.x86_64 0:3.2.1-4.amzn2.0.4

Complete!
[ec2-user@ip-172-31-4-113 ~]$ vi jenkins.txt
[ec2-user@ip-172-31-4-113 ~]$ sudo chmod +x jenkins.txt
[ec2-user@ip-172-31-4-113 ~]$ sudo ./jenkins.txt
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
0:00
No Match for argument: -y
No packages marked for update
--2023-07-22 11:46:57-- https://pkg.jenkins.io/redhat-stable/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io)... 151.101.146.133, 2a04:4e42:22::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|151.101.146.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 85
Saving to: '/etc/yum.repos.d/jenkins.repo'

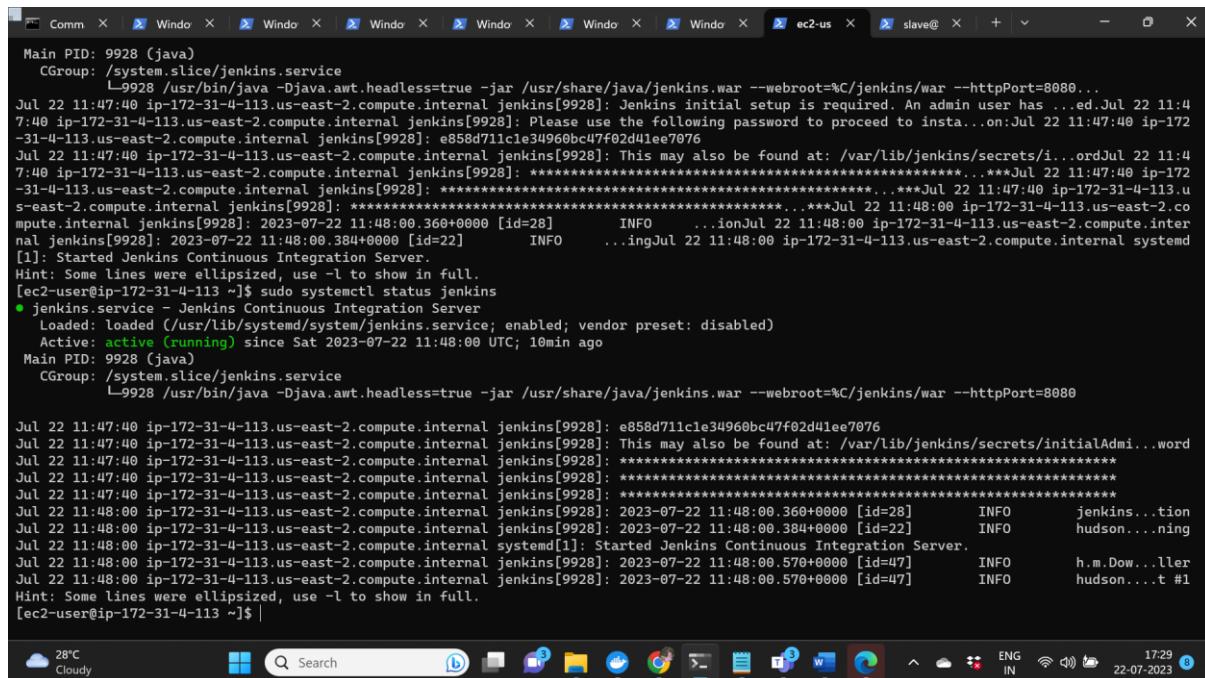
100%[=====] 85 --.-K/s   in
0s

2023-07-22 11:46:57 (5.39 MB/s) - '/etc/yum.repos.d/jenkins.repo' saved [85/85]

Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
jenkins
| 2.9 kB 00:0
Cloudy 28°C Search 17:24 22-07-2023
```

Validate the status of the Jenkins server by providing the command

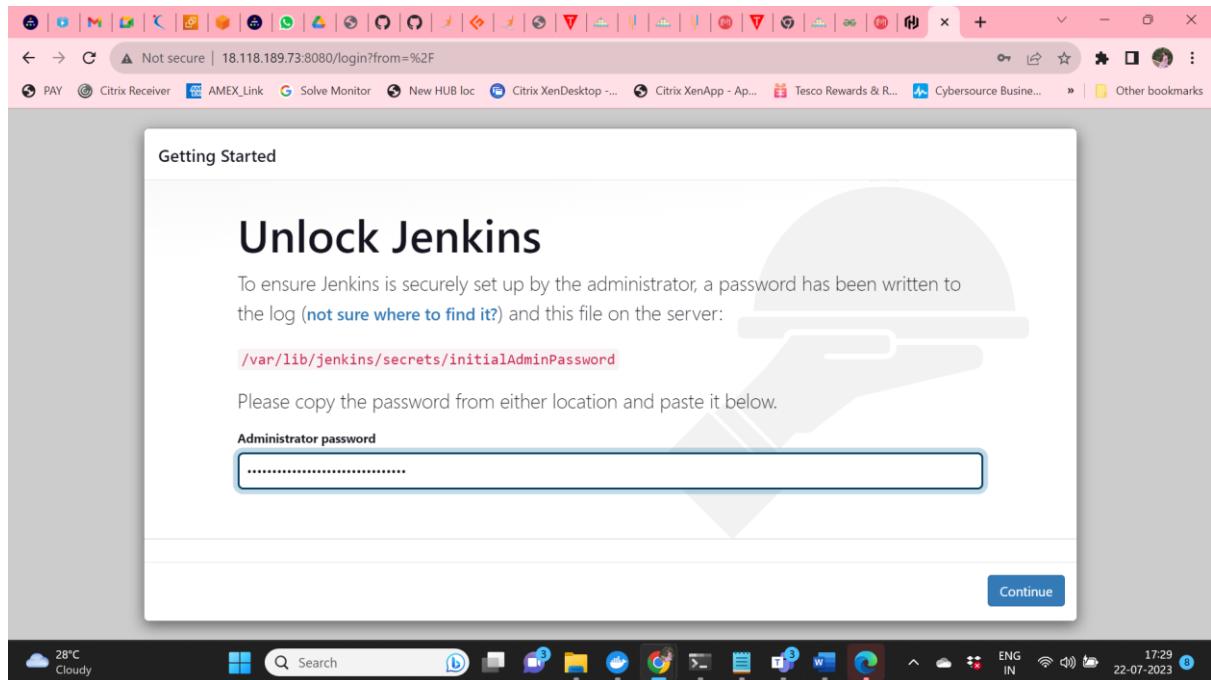
`sudo systemctl status jenkins`



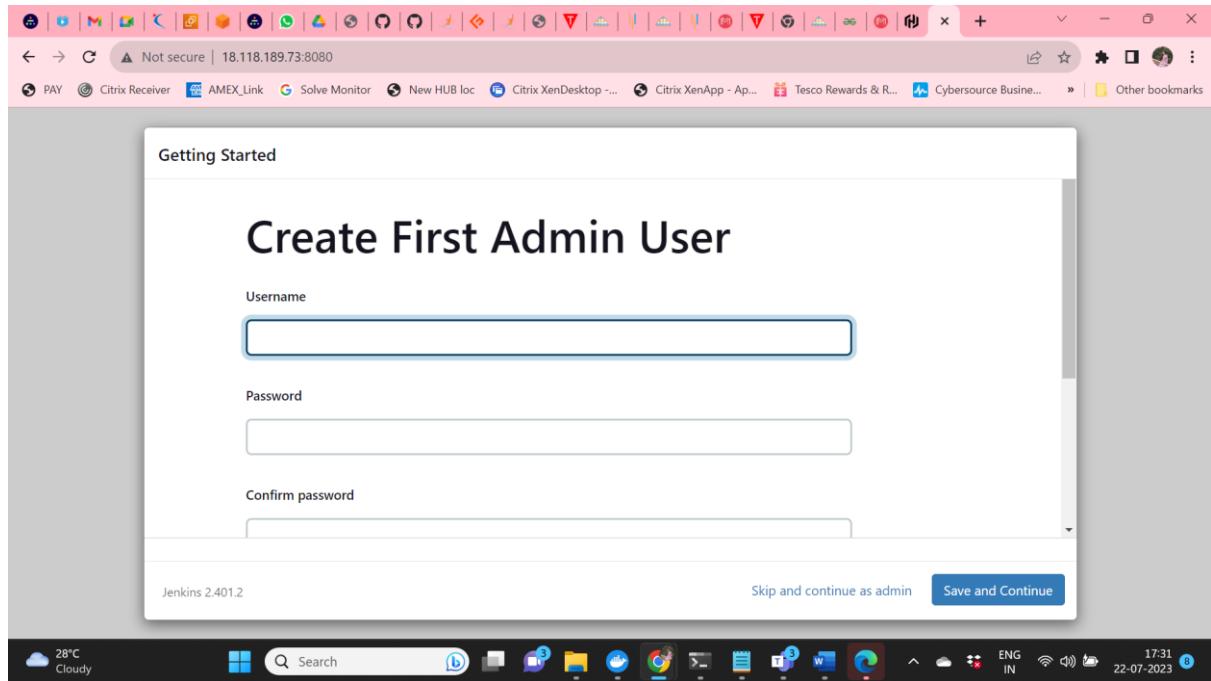
```
Main PID: 9928 (java)
CGroup: /system.slice/jenkins.service
└─9928 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war --httpPort=8080...
Jul 22 11:47:40 ip-172-31-4-113.us-east-2.compute.internal jenkins[9928]: Jenkins initial setup is required. An admin user has ...ed.Jul 22 11:47:40 ip-172-31-4-113.us-east-2.compute.internal jenkins[9928]: Please use the following password to proceed to insta...on:Jul 22 11:47:40 ip-172-31-4-113.us-east-2.compute.internal jenkins[9928]: This may also be found at: /var/lib/jenkins/secrets/...ordJul 22 11:47:40 ip-172-31-4-113.us-east-2.compute.internal jenkins[9928]: ****...***Jul 22 11:47:40 ip-172-31-4-113.us-east-2.compute.internal jenkins[9928]: ****...***Jul 22 11:47:40 ip-172-31-4-113.us-east-2.compute.internal jenkins[9928]: 2023-07-22 11:48:00.360+0000 [id=28]      INFO  ...ionJul 22 11:48:00 ip-172-31-4-113.us-east-2.compute.internal jenkins[9928]: 2023-07-22 11:48:00.384+0000 [id=22]      INFO  ...ingJul 22 11:48:00 ip-172-31-4-113.us-east-2.compute.internal systemd [1]: Started Jenkins Continuous Integration Server.
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-4-113 ~]$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
  Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; vendor preset: disabled)
  Active: active (running) since Sat 2023-07-22 11:48:00 UTC; 10min ago
    Main PID: 9928 (java)
   CGroup: /system.slice/jenkins.service
           └─9928 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war --httpPort=8080...

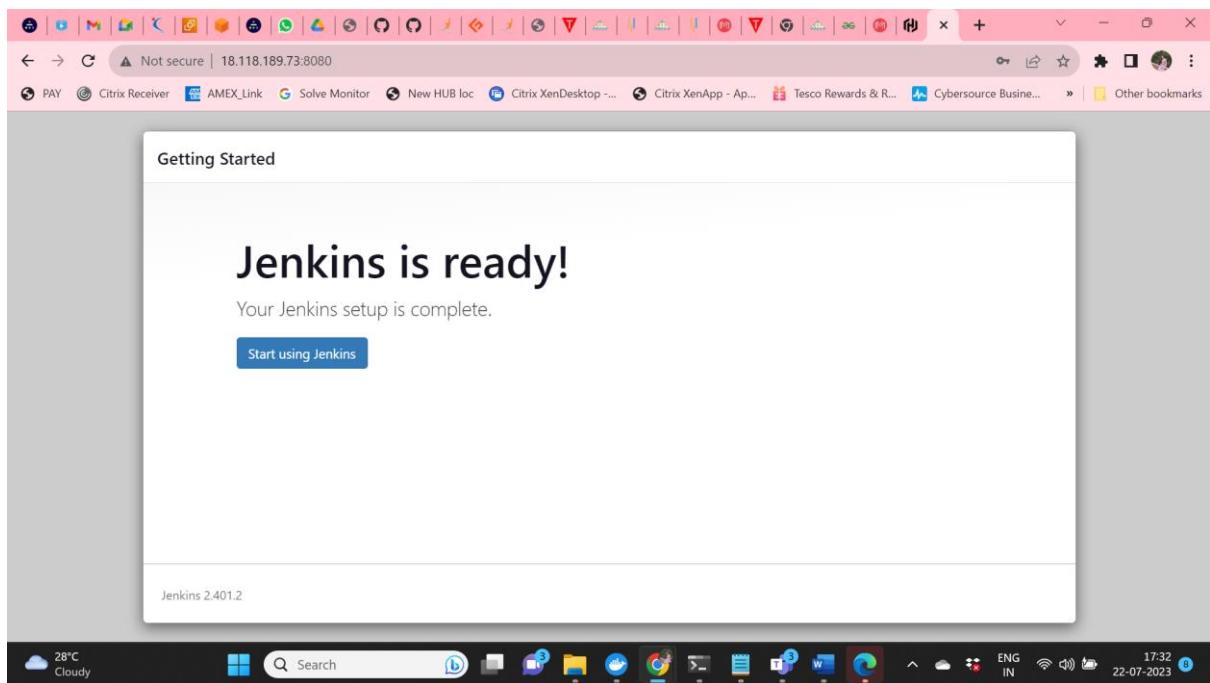
Jul 22 11:47:40 ip-172-31-4-113.us-east-2.compute.internal jenkins[9928]: e858d711c1e34960bc47f02d41ee7076
Jul 22 11:47:40 ip-172-31-4-113.us-east-2.compute.internal jenkins[9928]: This may also be found at: /var/lib/jenkins/secrets/initialAdm...word
Jul 22 11:47:40 ip-172-31-4-113.us-east-2.compute.internal jenkins[9928]: ****...***Jul 22 11:47:40 ip-172-31-4-113.us-east-2.compute.internal jenkins[9928]: ****...***Jul 22 11:47:40 ip-172-31-4-113.us-east-2.compute.internal jenkins[9928]: ****...***Jul 22 11:47:40 ip-172-31-4-113.us-east-2.compute.internal jenkins[9928]: 2023-07-22 11:48:00.360+0000 [id=28]      INFO  jenkins...tion
Jul 22 11:48:00 ip-172-31-4-113.us-east-2.compute.internal jenkins[9928]: 2023-07-22 11:48:00.384+0000 [id=22]      INFO  hudson...ning
Jul 22 11:48:00 ip-172-31-4-113.us-east-2.compute.internal systemd[1]: Started Jenkins Continuous Integration Server.
Jul 22 11:48:00 ip-172-31-4-113.us-east-2.compute.internal jenkins[9928]: 2023-07-22 11:48:00.570+0000 [id=47]      INFO  h.m.Dow...ller
Jul 22 11:48:00 ip-172-31-4-113.us-east-2.compute.internal jenkins[9928]: 2023-07-22 11:48:00.570+0000 [id=47]      INFO  hudson...t #1
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-4-113 ~]$ |
```

Use the public IP to open Jenkins in port 8080



Create first time username and password





Jenkins server is ready for use.

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Create a job

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Set up a distributed build

Set up an agent

Step3: Create a slave server in the same account in the region with default connections.

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with links like EC2 Dashboard, EC2 Global View, Events, Instances (selected), Instances Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, and Capacity Reservations. The main area displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Ava
master	i-08c98363790d364b4	Running	t2.medium	2/2 checks passed	No alarms	us-e
slave	i-0152b07418b1b68ed	Running	t2.medium	2/2 checks passed	No alarms	us-e

Below the table, a modal window is open for the 'slave' instance, titled 'Instance: i-0152b07418b1b68ed (slave)'. It has tabs for Details, Security, Networking, Storage, Status checks, Monitoring, and Tags. Under 'Details', it shows the Instance ID (i-0152b07418b1b68ed (slave)), Public IPv4 address (3.15.2.63 | open address), and Private IPv4 addresses (172.31.6.11).

The screenshot shows the 'ConnectToInstance' dialog for the 'slave' instance. It starts with the Instance ID (i-0152b07418b1b68ed (slave)). Below it is a numbered list of steps:

1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is ohio.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
chmod 400 ohio.pem
4. Connect to your instance using its Public DNS:
ec2-3-15-2-63.us-east-2.compute.amazonaws.com

A green message bubble says 'Command copied' over the command line. The command line itself shows 'ssh -i "ohio.pem" ec2-user@ec2-3-15-2-63.us-east-2.compute.amazonaws.com'. A note at the bottom says: 'Note: In most cases, the guessed user name is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.' At the bottom right is a 'Cancel' button.

Open the slave instance in the terminal and provide the required update commands.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\sutir> cd .\Downloads\
PS C:\Users\sutir\Downloads> ssh -i "ohio.pem" ec2-user@ec2-3-15-2-63.us-east-2.compute.amazonaws.com
The authenticity of host 'ec2-3-15-2-63.us-east-2.compute.amazonaws.com (3.15.2.63)' can't be established.
ED25519 key fingerprint is SHA256:/MCTVP+ixV6+XPfjwJFVail8ni0IJuYq9V/kx85OxLE.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-15-2-63.us-east-2.compute.amazonaws.com' (ED25519) to the list of known hosts.

  _|_ _|_
  _|(_ _| / Amazon Linux 2 AMI
  ___\_|_ _|_|

https://aws.amazon.com/amazon-linux-2/
7 package(s) needed for security, out of 7 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-6-11 ~]$ sudo yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
Resolving Dependencies
--> Running transaction check
--> Package kernel.x86_64 0:5.10.184-175.749.amzn2 will be installed
--> Package libcap.x86_64 0:2.54-1.amzn2.0.1 will be updated
--> Package libcap.x86_64 0:2.54-1.amzn2.0.2 will be an update
--> Package libtiff.x86_64 0:4.0.3-35.amzn2.0.6 will be updated
--> Package libtiff.x86_64 0:4.0.3-35.amzn2.0.7 will be an update
--> Package python-requests.noarch 0:2.6.0-7.amzn2 will be updated
--> Package python-requests.noarch 0:2.6.0-10.amzn2.0.1 will be an update
--> Package python-requests.noarch 0:2.6.0-10.amzn2.0.1 will be an update

| 3.7 kB  00:00:00
```

Step4: Create username, password, ssh_key.

```
Complete!
[ec2-user@ip-172-31-6-11 ~]$ sudo useradd slave
[ec2-user@ip-172-31-6-11 ~]$ sudo passwd slave
Changing password for user slave.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[ec2-user@ip-172-31-6-11 ~]$ sudo vi /etc/ssh/sshd_config
[ec2-user@ip-172-31-6-11 ~]$ sudo systemctl restart sshd
[ec2-user@ip-172-31-6-11 ~]$ exit
logout
Connection to ec2-3-15-2-63.us-east-2.compute.amazonaws.com closed.
PS C:\Users\sutir\Downloads> ssh slave@3.15.2.63
The authenticity of host '3.15.2.63 (3.15.2.63)' can't be established.
ED25519 key fingerprint is SHA256:/MCTVP+ixV6+XPfjwJFVail8ni0IJuYq9V/kx85OxLE.
This host key is known by the following other names/addresses:
  C:\Users\sutir\.ssh\known_hosts:651: ec2-3-15-2-63.us-east-2.compute.amazonaws.com
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '3.15.2.63' (ED25519) to the list of known hosts.
slave@3.15.2.63's password:

  _|_ _|_
  _|(_ _| / Amazon Linux 2 AMI
  ___\_|_ _|_|

https://aws.amazon.com/amazon-linux-2/
[slave@ip-172-31-6-11 ~]$ sudo su - slave
We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:
[slave@ip-172-31-6-11 ~]#
```

```
[sudo] password for slave:
slave is not in the sudoers file. This incident will be reported.
[slave@ip-172-31-6-11 ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/slave/.ssh/id_rsa):
Created directory '/home/slave/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/slave/.ssh/id_rsa.
Your public key has been saved in /home/slave/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:7vhno1iQ42M9vz8Tvb6Acj/EY2QNaLGa/806DzB6lQ slave@ip-172-31-6-11.us-east-2.compute.internal
The key's randomart image is:
+---[RSA 2048]---+
|          o
|         . *
|        . +
|       o o . o
|      o S E o .
|     o + o .oo .
|    . + .+.*.=
|     *=X.= +
|      B%:B+=
+---[SHA256]---+
[slave@ip-172-31-6-11 ~]$ cd .ssh
[slave@ip-172-31-6-11 .ssh]$ cat id_rsa.pub > authorized_keys
[slave@ip-172-31-6-11 .ssh]$ chmod 700 authorized_keys
[slave@ip-172-31-6-11 .ssh]$ ls
authorized_keys  id_rsa  id_rsa.pub
[slave@ip-172-31-6-11 .ssh]$ pwd
/home/slave/.ssh
[slave@ip-172-31-6-11 .ssh]$ cat id_rsa
```

after the privileges to user ,in ec2 server

go --> cd/etc/shell/sshd_config

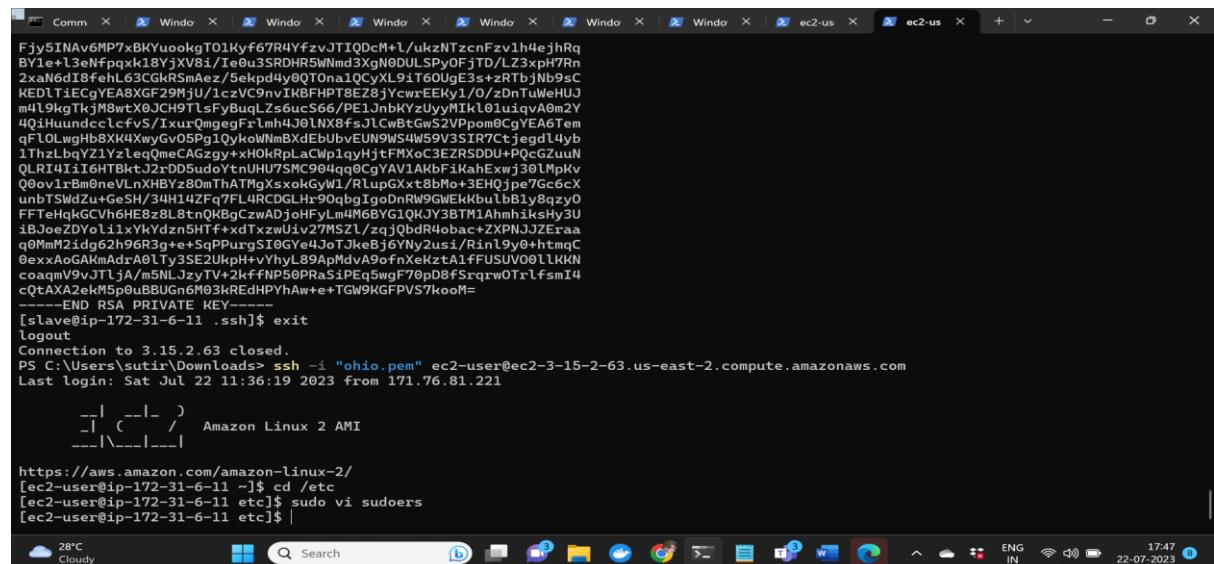
search where you need to update modify authentication password permission"Yes"

ssh-keygen -t rsa -N "" -f /home/jenkins-slave/.ssh/id_rsa or ssh-keygen

cd .ssh

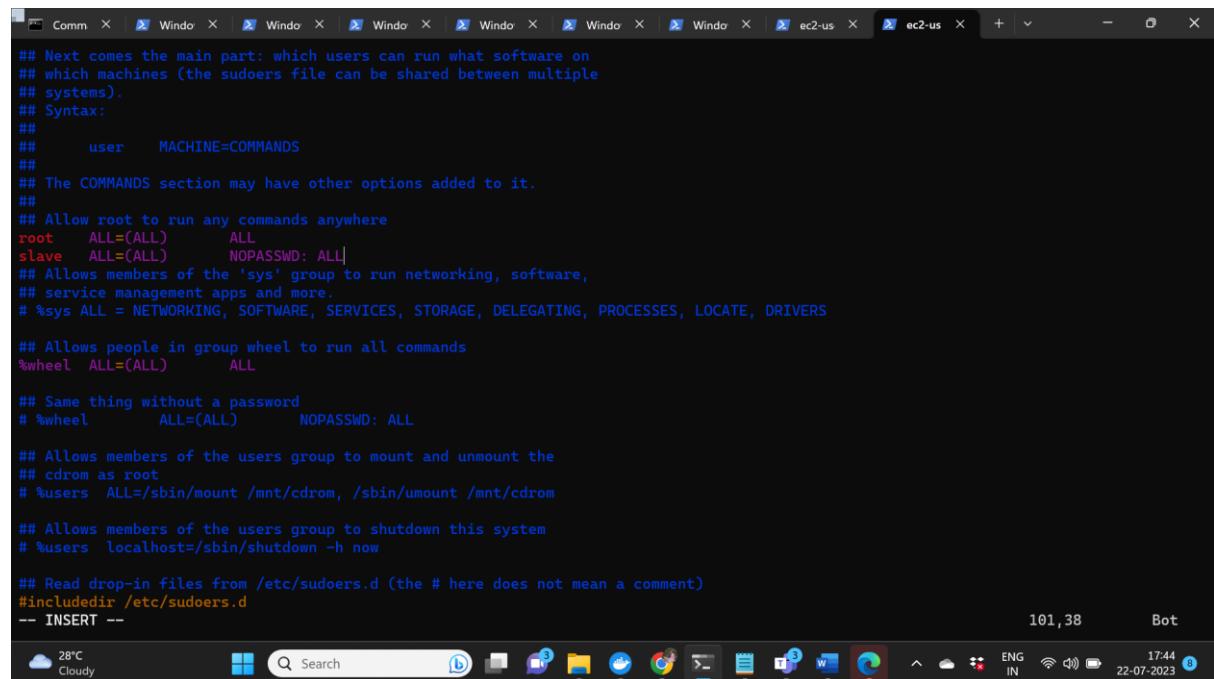
cat id_rsa.pub > authorized_keys

chmod 700 authorized_keys



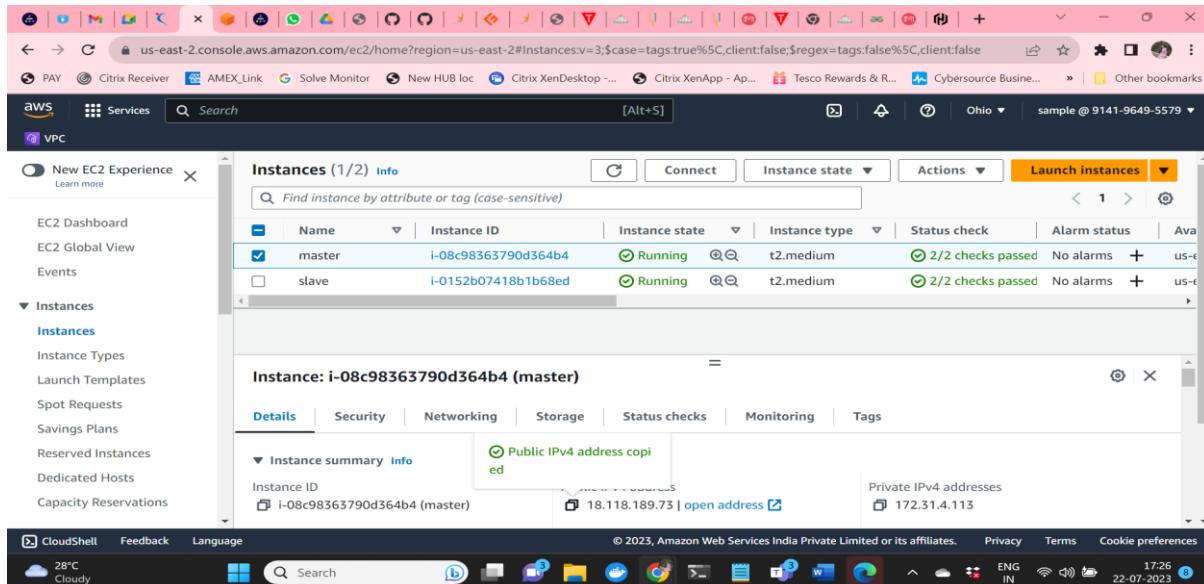
```
Fjy5INA6MP7xKYUoogkT01KyF67R4YfvzvJTIQDcM+l/ukzNTzcnFzv1h4ejhRq
BY1e+13eNfpqgx18YjXV8i/Ie0u3SRDHR5Wlmd3XgN0DULSPyOfjTD/LZ3xpH7Rn
2xaN6dI8fehL63CGrSmAez/5ekpd4y@TOna1QCyXL91t60UgE3s+zRTbjNb9c
KEDLTiECgYEAX8GF29Mju/1czVCvNvIkBFHT8EZbjYcwxEEk1/0/zDnTuWeHJ
m4L9kgTkjM8wtx0JCH9TLsFyBulgZs6ucS66/PE1JnbkYzUyyMIkl01uiqvA0m2Y
4Q1HuundccLcfvS/IxurQmgegFxmh4J0lNX8fJLcwbGwS2VPponoCgYEAE6Tem
qFL0LwghBxK4XwGv05Pq1QykoWNmBxDEbUbvEU9w54w59v3StR7ctjegdl4yb
1ThzLbzQY1zleqQmeCAGzgy+xh0krpLaCwplqyHjtFMxoc3EZRSDDU+P0cGZuuN
QLR14i16HTBktJ2rDD5udoYtnUH7USMC904qq0CgYAV1AkbfikahExwj30lMpkv
Q0ev01Bn0neVLnXHBYz80mThATMgXsokGyW1/RlupGx:t8bfMo+3EHQjpe7Gc6Cx
unbTSwdZu+GeSH/34H14Zfq7FL4RCDGLHr90qbgIgoOnRW9GWEMkbub1y8qzy0
FTTeHqkGCVh6HE8z8L8tn0K8gCzwAdjoFyl.m4M6BYG1QkJY3BT1AhmhiksHy3U
iBjoezDy6l1xYkYdzn5HTf+xdtxwU127MS2L/zqjQbdR4obac+ZXPNJZERaa
q0mm2Idg62h96R3g++SgPPuvgSi0Gye4JoTkebJ6Ny2usi/Rin9y0+hthmcC
0xxAoGAkMAdr@0Ly3SE2Ukph+vYhlyL89ApMdva9fnXeKztAlfFUSUV001lkHN
coaqm9vJTLja/m5NLJzyTV+2kfNP50PrAsiPeqswF70pDf8Szqrw0trlfsmI4
cQTAx2ekM5p0B8UGn6M03kREdHPYhAw+e+TGW9GFpV57kooM=
-----END RSA PRIVATE-----
[slave@ip-172-31-6-11 .ssh]$ exit
logout
Connection to 3.15.2.63 closed.
PS C:\Users\sutir\Downloads> ssh -i "ohio.pem" ec2-user@ec2-3-15-2-63.us-east-2.compute.amazonaws.com
Last login: Sat Jul 22 11:36:19 2023 from 171.76.81.221
    _|_ _|_
    |(_/_ / Amazon Linux 2 AMI
    ___\_\_\_|
https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-6-11 ~]$ cd /etc
[ec2-user@ip-172-31-6-11 etc]$ sudo vi sudoers
[ec2-user@ip-172-31-6-11 etc]$ |
```

Provide the permissions for Jenkins in sudoers using sudo command



```
## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##       user      MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)      ALL
slave   ALL=(ALL)      NOPASSWD: ALL
## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys  ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS
## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)      ALL
## Same thing without a password
# %wheel    ALL=(ALL)      NOPASSWD: ALL
## Allows members of the users group to mount and umount the
## cdrom as root
# %users   ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom
## Allows members of the users group to shutdown this system
# %users   localhost=/sbin/shutdown -h now
## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
#includedir /etc/sudoers.d
-- INSERT --
101,38          Bot
28°C Cloudy      Search      B   S   F   G   E   D   W   C   P   A   V   M   ^   ⌂   ENG IN   17:44   22-07-2023
```

Step5: Configure Jenkins Master Node:

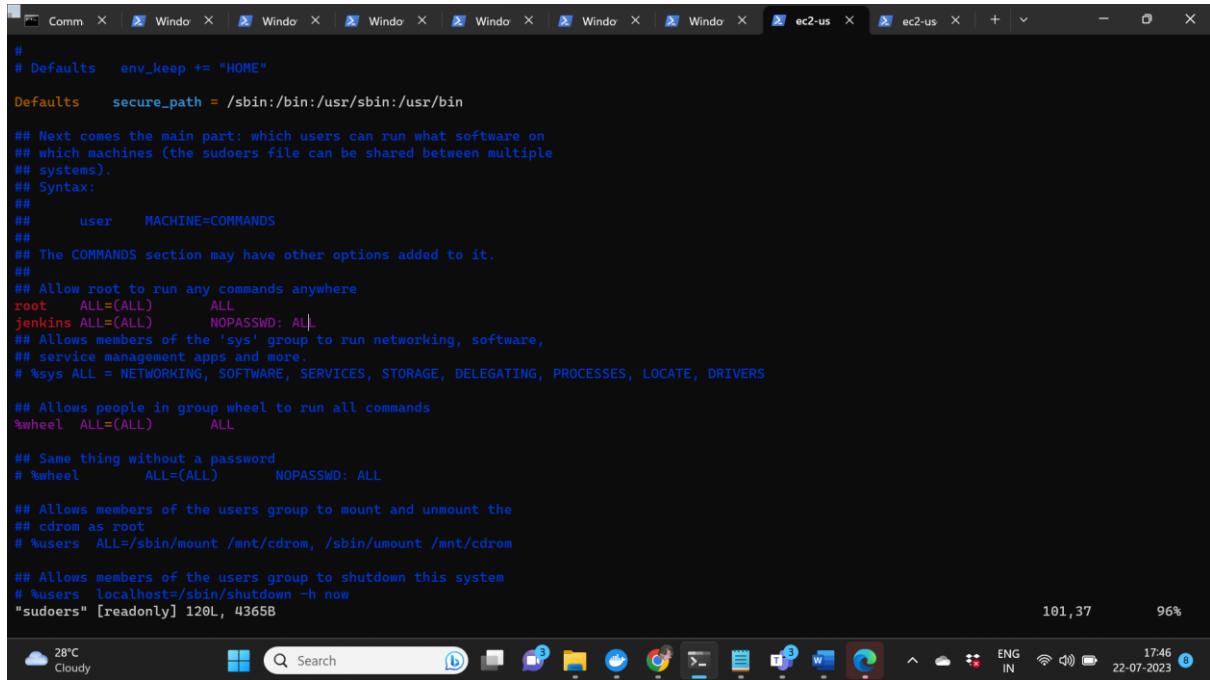


Copy the slave node's public key to master node's known_hosts file.

```
sudo mkdir -p /var/lib/jenkins/.ssh
cd /var/lib/jenkins/.ssh
cd ..
sudo chmod 777 .ssh
cd .ssh
sudo ssh-keyscan -H SLAVE_NODE_PRIVATE_IP >>/var/lib/jenkins/.ssh/known_hosts
sudo chown jenkins:jenkins known_hosts
sudo chmod 700 known_hosts
```

```
L-9928 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war --httpPort=8080
Jul 22 11:47:48 ip-172-31-4-113.us-east-2.compute.internal jenkins[9928]: e858d711c1c34960b47f82d41ee7076
Jul 22 11:47:48 ip-172-31-4-113.us-east-2.compute.internal jenkins[9928]: This may also be found at: /var/lib/jenkins/secrets/initialAdm...word
Jul 22 11:47:48 ip-172-31-4-113.us-east-2.compute.internal jenkins[9928]: ****
Jul 22 11:47:48 ip-172-31-4-113.us-east-2.compute.internal jenkins[9928]: ****
Jul 22 11:47:48 ip-172-31-4-113.us-east-2.compute.internal jenkins[9928]: ****
Jul 22 11:47:48 ip-172-31-4-113.us-east-2.compute.internal jenkins[9928]: 2023-07-22 11:48:00.360+0000 [id=28] INFO jenkins...tion
Jul 22 11:48:00 ip-172-31-4-113.us-east-2.compute.internal jenkins[9928]: 2023-07-22 11:48:00.384+0000 [id=22] INFO hudson...ning
Jul 22 11:48:00 ip-172-31-4-113.us-east-2.compute.internal systemd[1]: Started Jenkins Continuous Integration Server.
Jul 22 11:48:00 ip-172-31-4-113.us-east-2.compute.internal jenkins[9928]: 2023-07-22 11:48:00.570+0000 [id=47] INFO h.m.Dow...ller
Jul 22 11:48:00 ip-172-31-4-113.us-east-2.compute.internal jenkins[9928]: 2023-07-22 11:48:00.570+0000 [id=47] INFO hudson...t #1
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-4-113 ~]$ sudo mkdir -p /var/lib/jenkins/.ssh
[ec2-user@ip-172-31-4-113 ~]$ cd /var/lib/jenkins/.ssh
[ec2-user@ip-172-31-4-113 .ssh]$ cd ..
[ec2-user@ip-172-31-4-113 jenkins]$ sudo chmod 777 .ssh
[ec2-user@ip-172-31-4-113 jenkins]$ cd .ssh
[ec2-user@ip-172-31-4-113 .ssh]$ ssh-keyscan -H 172.31.6.11 >/var/lib/jenkins/.ssh/known_hosts
# 172.31.6.11:22 SSH-2.0-OpenSSH-7.4
# 172.31.6.11:22 SSH-2.0-OpenSSH-7.4
# 172.31.6.11:22 SSH-2.0-OpenSSH-7.4
[ec2-user@ip-172-31-4-113 .ssh]$ sudo chown jenkins:jenkins known_hosts
[ec2-user@ip-172-31-4-113 .ssh]$ sudo chmod 700 known_hosts
[ec2-user@ip-172-31-4-113 .ssh]$ ls
known_hosts
[ec2-user@ip-172-31-4-113 .ssh]$ |
28°C Cloudy  Search  B  ENG IN  17:43  22-07-2023
```

Provide the permissions for Jenkins in sudoers using sudo command.



```
# Defaults env_keep += "HOME"
Defaults secure_path = /sbin:/bin:/usr/sbin:/usr/bin

## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##      user      MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)      ALL
jenkins ALL=(ALL)      NOPASSWD: ALL
## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

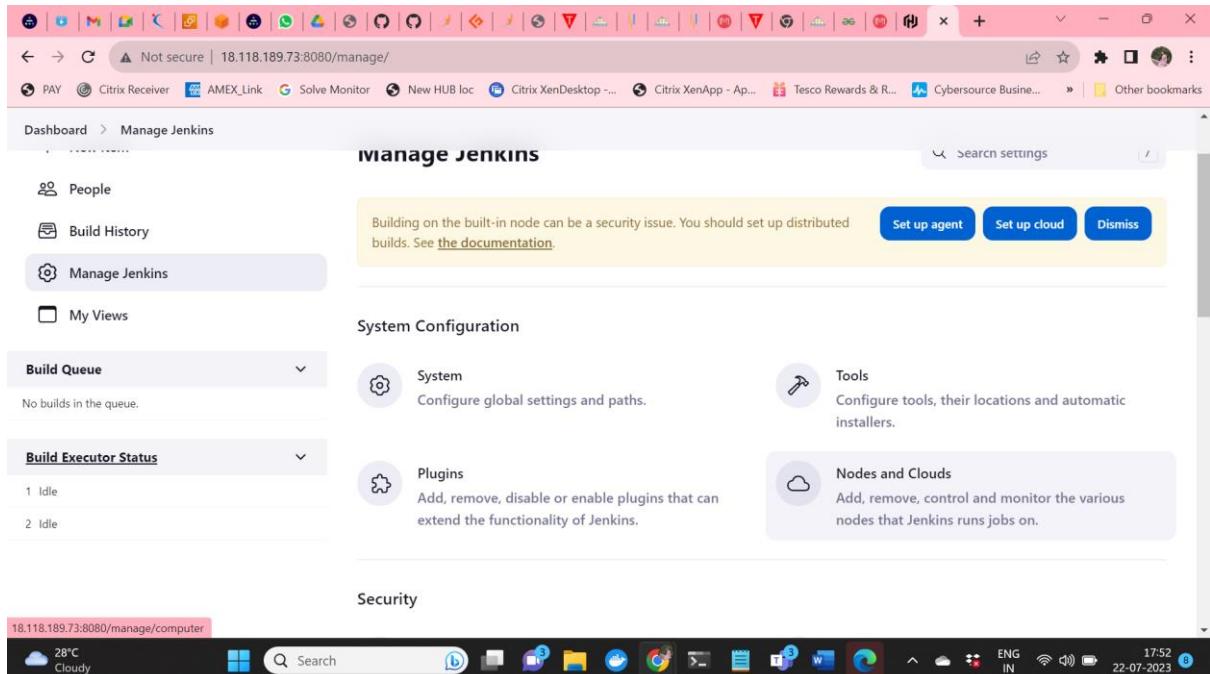
## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)      ALL

## Same thing without a password
# %wheel      ALL=(ALL)      NOPASSWD: ALL

## Allows members of the users group to mount and umount the
## cdrom as root
# %users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
# %users  localhost=/sbin/shutdown -h now
"sudoers" [readonly] 120L, 4365B
```

After the permissions provided in master and slave to Jenkins create a new node



The screenshot shows the Jenkins Manage Jenkins interface. The left sidebar has links for People, Build History, Manage Jenkins (which is selected), and My Views. The main content area is titled "Manage Jenkins". It features a "System Configuration" section with links for Build Queue, Build Executor Status, Plugins, System, Tools, and Nodes and Clouds. A note about building on the built-in node is displayed. At the bottom, there's a "Security" section. The browser's address bar shows the URL 18.118.189.73:8080/manage/computer. The bottom status bar includes icons for weather, search, taskbar, and system information like battery level, signal strength, and date/time.

Jenkins

Dashboard > Manage Jenkins > Nodes >

Nodes

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	5.18 GB	0 B	5.18 GB	0ms
	last checked	34 min	34 min	34 min	34 min	34 min	34 min

+ New Node

18.118.189.73:8080/manage/computer/new

REST API Jenkins 2.401.2

Jenkins

Dashboard > Manage Jenkins > Nodes > New node

New node

Node name

slave

Type

Permanent Agent

Adds a plain, permanent agent to Jenkins. This is called "permanent" because Jenkins doesn't provide higher level of integration with these agents, such as dynamic provisioning. Select this type if no other agent types apply — for example such as when you are adding a physical computer, virtual machines managed outside Jenkins, etc.

Create

Not secure | 18.118.189.73:8080/manage/computer/createtem

Dashboard > Manage Jenkins > Nodes >

Number of executors ?
5

Remote root directory ?
/home/slave

Labels ?
labels

Usage ?
Use this node as much as possible

Launch method ?
Save

Not secure | 18.118.189.73:8080/manage/computer/createtem

Dashboard > Manage Jenkins > Nodes >

Usage ?
Domain
Global credentials (unrestricted)

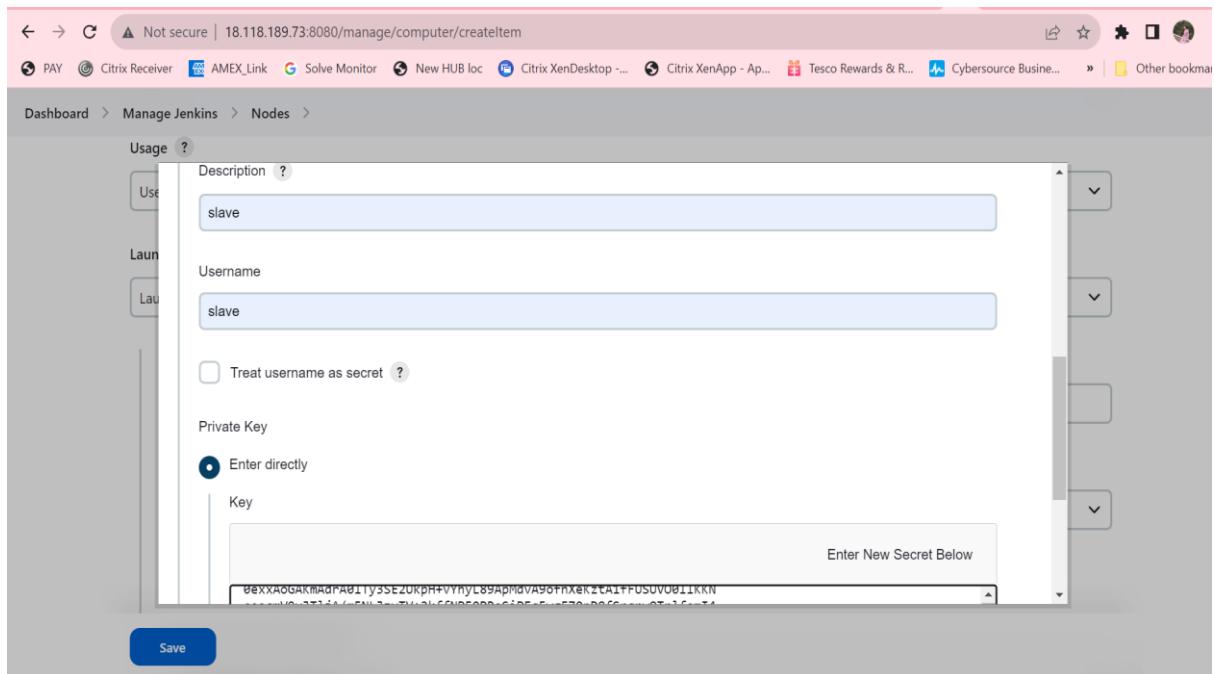
Kind
SSH Username with private key

Scope ?
Global (Jenkins, nodes, items, all child items, etc)

ID ?
slave

Description ?

Save



```
[slave@ip-172-31-6-11 .ssh]$ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEogIBAAKCAQEa2/TPZzjsG6+8D+QRetFjpTl79geIqa4k69q00epCGoR5rPNx
PqY6LjS3ycIy1AV/eQTBFWcJzH8Tp4p/jnNp09FSAkz/Hen87lk0r9JF/rulwQ0
y07zgb+crXujTftVzDrmE9zzjgxJcvXZp/w6bEyaPEo1zlpY2vP3wcYKF00+jsl0
S3Y16u/No0gu4ezbFjBRCtkrSA3eEpalLtifyGD2d6rOXizBf7L040eTSjepDI
hkUdWkt169eg/bC6nLigoADBen1T+QsgnwYxH2dEcpJ1Fr9xbWpegE5iSytD
9ogKpim6xtC/0zPXILpysU2QtgVaCbGZe5cLqIDAQABAcIBAEsGw+AniFGBrbd
5qBv338x9lhb8VB+B+jy6ZgsQVPuL9qQRuZ3qN4nKTFGbjPnLC1hM8LrHHnmgMGI
LWarrltKVFnzLUVKnqnLBzgkqNTH5Gzwlx8Nkm1HnQHcmxmvhIdsX7iY+cHUL7
Fjy5INAv6MP7x8KYUookgT01kyf67R4YFyzvJTIQDcM+l/ukzNTzcnFzv1h4ejhrq
BY1e+13Efnpqxk18yjXV8i/Ie0u3SRDRH5Wmd3XgN0DULSPyOfjTD/LZ3xpH7Rn
2xaN6dI8fehL63CGkRSmAez/5ekpd4y0QT0na1QCyXL9iT60UgE3s+zRTbjNb9sc
KED1TiEcgYEAE8XF29mjU/1czVC9nvIKBFHTBEZBjYcwxEEkY1/0/zDnTuWeHuj
m4l9kgTkjM8wtX0JCH9TlsFyBugLzs6ucS66/PE1JnbkYzLyMikl01uiqvA0m2Y
4QiHuundcc1cfvS/IxurQmgegFrLmh4J0LNX8fsJLCwBtGwS2Vppom0CgYEAtEm
qFLOLwghB8XK4XwyGv05Pg1QykoWNmBxdEBubvEU9WS4w59V3S1R7ctjegd14yb
1ThzLbq1z1VzleqQmeCAGzgy+xh0krpLaCwplqyHjtFMxoc3EZRSDDU+PqcGzuuN
QLR14IiI6HTBktJ2rDD5udoYtnUHU7SMC904qq0CgYAV1AKbfkIahExwj30LMpkv
Q0ov1rBm0neVlnXHBy280mThATMgXsookGyW1/RlupGxxt8bMo+3EHQjpe7Gc6cX
unbTSwdZu+GeSH/34H14Zfq7FL4RCDGLHr90qbgIgoOnRw9GWekkbub1b1yBqzyO
FTTeHqkGCVh6HE8zBL8tnQKBgCzwADjoHfylm4M6BYG1QkJY3BTM1AhmhiksHy3U
ibJoeZDyli1xYkdzn5HTf+xdtxzUiwv27MSzL/zqjbdR4obac+ZXPNNJZEraa
q0Mm2idg62h96R3g++SqPPuRgSIOGYe4JoTjkeBj6Ny2usi/Rinl9y0+htmqC
0exxxAoGAKmAtrA01ty3SE2UkpH+vYhlyL89ApMdva9ofnXekztAlffFUSUV09llKKN
coaqlmV9vJTLja/m5NLJzyTV+2kffNP50PrasipEq5wgF70pD8fsRqrw0TrlfsmI4
cQTAx2ekM5p0UBU6n6M03kRedHPYhAw+e+TGW9KGFPV57kooM=
-----END RSA PRIVATE KEY-----
[slave@ip-172-31-6-11 .ssh]$ exit
logout
Connection to 3.15.2.63 closed.
PS C:\Users\sutir\Downloads> ssh -i "ohio.pem" ec2-user@ec2-3-15-2-63.us-east-2.compute.amazonaws.com
```

Provide the private key of the slave in the new node

The screenshot shows the Jenkins 'Manage Jenkins > Nodes' configuration page. In the 'Private Key' section, the 'Enter directly' radio button is selected. A text area contains the RSA private key:

```
-----BEGIN RSA PRIVATE KEY-----  
MIIEowIBAAKCAQEAo1Iy3SEZUkDH+VYRvL89ApIdVA96fNxERzLq1FFUSUV0811KKN  
coaqmV9vJT1ja/m5NLjzyTv+2kffNP58Pras1Peq5wgF70p08fSrqrwOTr1fsmI4  
cQtAXA2ekM5p0uBU6n6@3kREDhPYhAw+e+TGw9KGFPVs7ko0M=  
-----END RSA PRIVATE KEY-----
```

Below the key, there is a 'Passphrase' field and a 'Save' button.

The screenshot shows the Jenkins 'Manage Jenkins > Nodes' configuration page. In the 'Host' section, the IP address '172.31.6.11' is entered. In the 'Credentials' section, the dropdown menu is set to 'slave (slave)'. At the bottom, there is a 'Save' button.

Not secure | 18.118.189.73:8080/manage/computer/creatitem

PAY Citrix Receiver AMEX_Link Solve Monitor New HUB loc Citrix XenDesktop ... Citrix XenApp - Ap... Tesco Rewards & R... Cybersource Busine... Other bookmarks

Dashboard > Manage Jenkins > Nodes >

Add

Host Key Verification Strategy ?

Known hosts file Verification Strategy

Advanced

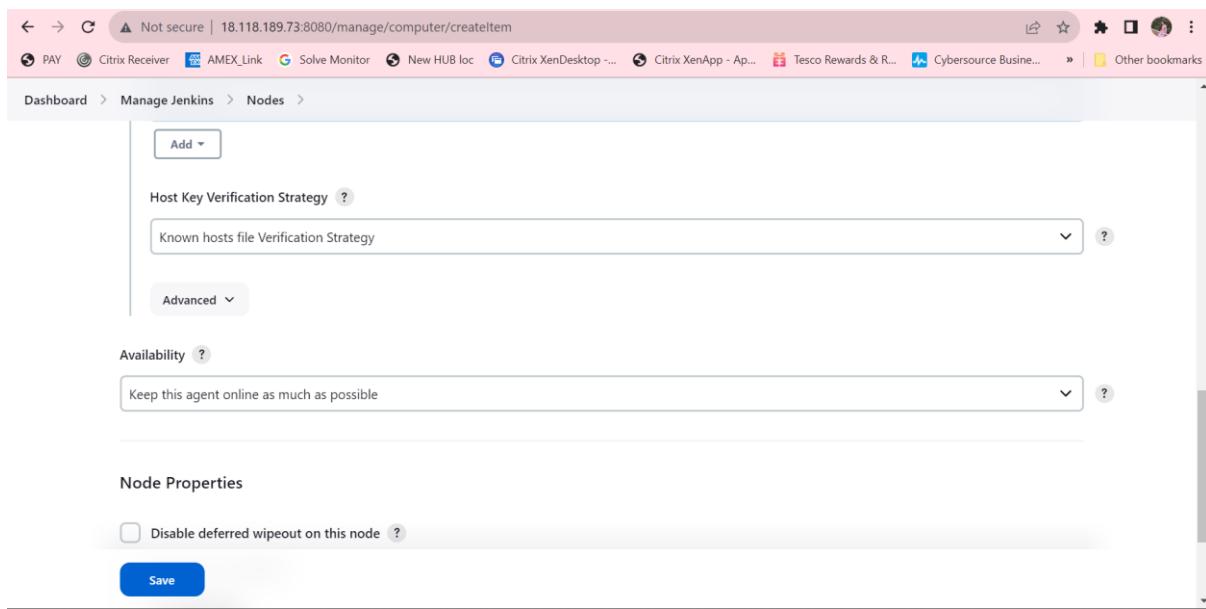
Availability ?

Keep this agent online as much as possible

Node Properties

Disable deferred wipeout on this node ?

Save



Not secure | 18.118.189.73:8080/computer/slave/log

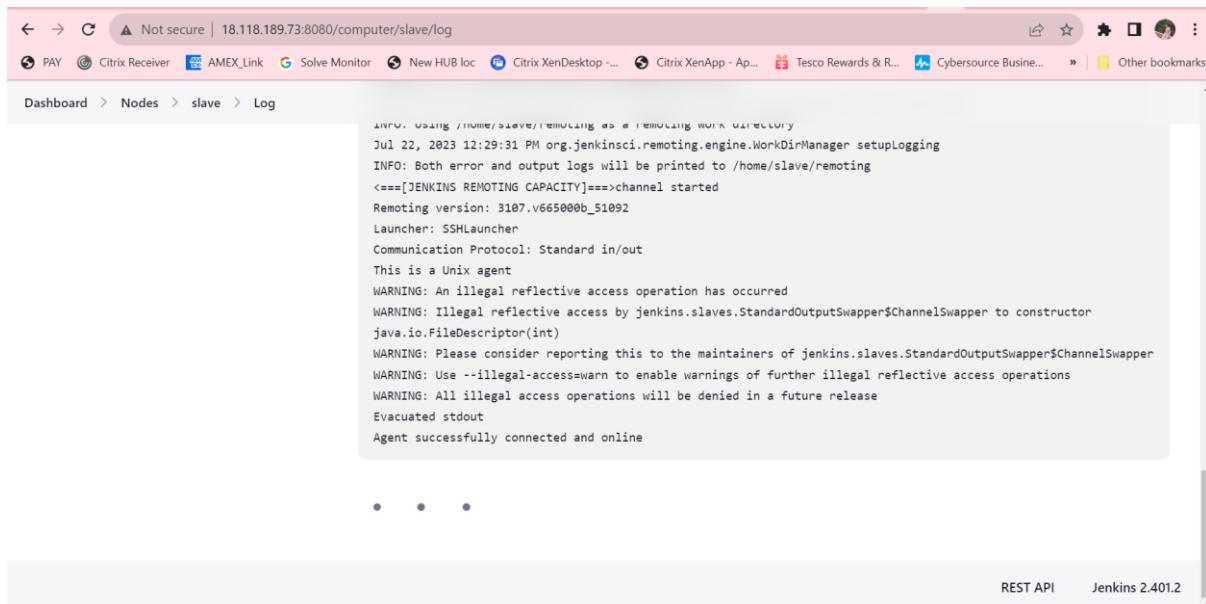
Citrix Receiver AMEX_Link Solve Monitor New HUB loc Citrix XenDesktop ... Citrix XenApp - Ap... Tesco Rewards & R... Cybersource Busine... Other bookmarks

Dashboard > Nodes > slave > Log

```
INFO: Using /home/slave/remoting as a remoting work directory
JUL 22, 2023 12:29:31 PM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /home/slave/remoting
<===[JENKINS REMOTING CAPACITY]==>channel started
Remoting version: 3107.v665000b_51092
Launcher: SSHLauncher
Communication Protocol: Standard in/out
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
```

• • •

REST API Jenkins 2.401.2



The new node has been created and is successful

The screenshot shows the Jenkins 'Nodes' page. On the left, there are sections for 'Build Queue' (empty) and 'Build Executor Status'. The main area displays a table of nodes:

S	Name	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space	Response Time
	Built-In Node	Linux (amd64)	In sync	4.98 GB	! 0 B	4.98 GB	0ms
	slave	Linux (amd64)	In sync	5.56 GB	! 0 B	5.56 GB	78ms

Install GIT and create a new job

The screenshot shows the Jenkins 'Dashboard' page. On the left, there are links for 'People', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', and 'My Views'. The main area shows a table for a job named 'git':

S	W	Name	Last Success	Last Failure	Last Duration
...	...	git	N/A	N/A	N/A

Below the table, there are links for 'Atom feed for all', 'Atom feed for failures', and 'Atom feed for just latest builds'. At the bottom, there is a 'Build Queue' section which is empty.

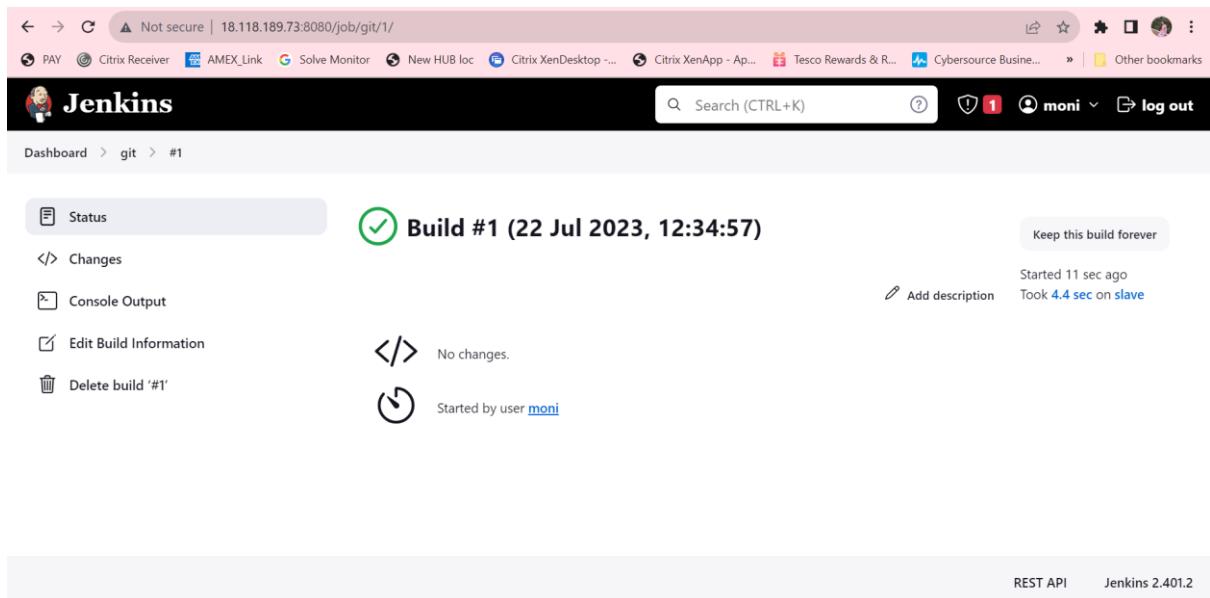
Add the newly created node(slave) in the Label Expression

The screenshot shows the Jenkins configuration interface for a job named 'git'. Under the 'General' tab, there are several checkboxes: 'Discard old builds', 'GitHub project', 'This project is parameterised', 'Throttle builds', 'Execute concurrent builds if necessary', and 'Restrict where this project can be run'. The 'Restrict where this project can be run' checkbox is checked, and its sub-section 'Label Expression' contains the value 'slave'. A note below states: 'Label slave matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.' At the bottom are 'Save' and 'Apply' buttons.

In the build steps, provide the command to install git and save.

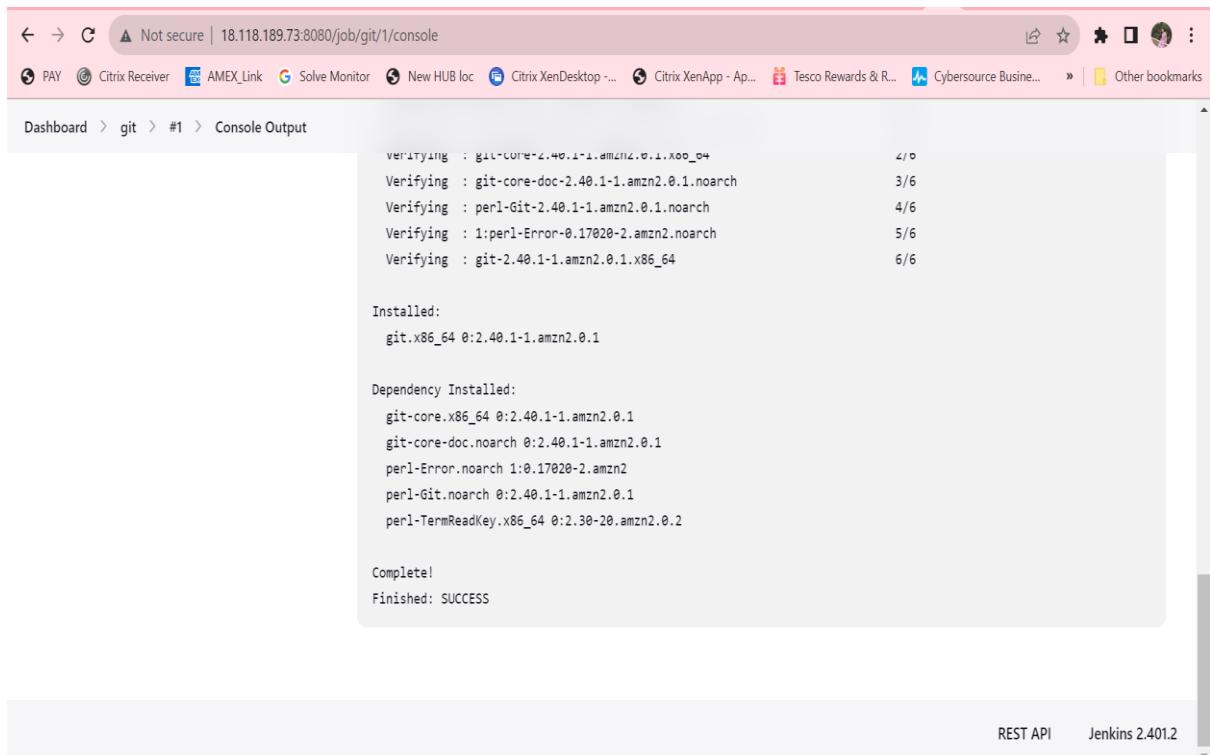
The screenshot shows the Jenkins configuration interface for the same 'git' job. Under the 'Build Steps' tab, there is a single step named 'Execute shell'. The 'Command' field contains the following shell script:
#!/bin/bash
sudo yum -y install git

Build the job.



The screenshot shows the Jenkins interface for a job named 'git'. The build number is #1, which was successful (indicated by a green checkmark icon). The build was started 11 seconds ago and took 4.4 seconds on a slave. The status bar indicates 'Keep this build forever'. Navigation links include 'Status', 'Changes', 'Console Output', 'Edit Build Information', and 'Delete build #1'. A note says 'No changes.' Below the status bar, it shows 'Started by user moni'. At the bottom right, there are links for 'REST API' and 'Jenkins 2.401.2'.

Check the status is Successful.



The screenshot shows the Jenkins console output for build #1. The output displays the results of a package verification process. It lists several packages and their verification status:

Verifying	Status
git-core-2.40.1-1.amzn2.0.1.x86_64	4/4
git-core-doc-2.40.1-1.amzn2.0.1.noarch	3/6
perl-Git-2.40.1-1.amzn2.0.1.noarch	4/6
1:perl-Error-0.17020-2.amzn2.noarch	5/6
git-2.40.1-1.amzn2.0.1.x86_64	6/6

Below this, the output shows the installed packages:

```
git.x86_64 0:2.40.1-1.amzn2.0.1
```

Dependency Installed:

```
git-core.x86_64 0:2.40.1-1.amzn2.0.1
git-core-doc.noarch 0:2.40.1-1.amzn2.0.1
perl-Error.noarch 1:0.17020-2.amzn2
perl-Git.noarch 0:2.40.1-1.amzn2.0.1
perl-TermReadKey.x86_64 0:2.30-20.amzn2.0.2
```

The build concludes with:

```
Complete!
Finished: SUCCESS
```

At the bottom right, there are links for 'REST API' and 'Jenkins 2.401.2'.

Creating a new job to automate terraform infrastructure using Jenkins

The screenshot shows the Jenkins 'Create New Item' dialog. The title bar says 'Enter an item name'. A text input field contains the word 'slave', which is highlighted in red and marked as a 'Required field'. Below the input field, there are two project type options: 'Freestyle project' and 'Pipeline'. The 'Freestyle project' option is selected, and its description is visible: '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.' An 'OK' button is visible at the bottom of this section.

The screenshot shows the Jenkins 'Configure' screen for the 'slave' job. The left sidebar has tabs for General, Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The 'General' tab is selected. Under 'General', there are several checkboxes: 'Discard old builds', 'GitHub project', 'This project is parameterised', 'Throttle builds', 'Execute concurrent builds if necessary', and 'Restrict where this project can be run'. The 'Restrict where this project can be run' checkbox is checked. Below it, a 'Label Expression' input field contains the value 'slave'. A note below the input field states: 'Label slave matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.' At the bottom of the configuration page, there are 'Save' and 'Apply' buttons.

Git repository containing the Terraform scripts

The screenshot shows a GitHub repository page for 'Terraform-python-TF'. The repository has 1 branch and 0 tags. The 'About' section describes it as 'Terraform scripts for python application'. It includes links to 'Readme', 'Activity', '0 stars', '1 watching', and '0 forks'. The 'Releases' section indicates 'No releases published' and 'Create a new release'. The 'Packages' section shows 'No packages published' and 'Publish your first package'. The central area lists files: README.md, Route_table.tf, data.sh, ec2.tf, igw.tf, ireland-2, main.tf, subnet.tf, vpc.tf, and web_sg.tf. The 'data.sh' file is highlighted with a tooltip: 'Update data.sh'. The 'Clone' section shows a copied HTTPS URL: <https://github.com/sangitagit/Terraform-python-TF>.

Setting up the job by providing the username and password

The screenshot shows a Jenkins configuration screen for a slave job. The left sidebar shows navigation options: Dashboard, slave, Configuration, General, Source Code, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The 'Configuration' screen is active. Under the 'Source Code' tab, the 'Kind' is set to 'Username with password'. The 'Scope' is set to 'Global (Jenkins, nodes, items, all child items, etc)'. The 'Username' field contains 'sangitagit'. The 'Treat username as secret' checkbox is unchecked. At the bottom, there are 'Save' and 'Apply' buttons.

Not secure | 18.118.189.73:8080/job/slave/configure

Dashboard > slave > Configuration

Configure

Treat username as secret ?

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Password ?
.....

ID ?
slave-deploy

Description ?
slave-deploy|

Add Cancel

Save Apply

This screenshot shows the 'Configuration' dialog for a Jenkins slave job. The 'Source Code Management' tab is selected. The 'General' section contains fields for 'Password' (redacted), 'ID' (set to 'slave-deploy'), and 'Description' (set to 'slave-deploy|'). Other tabs visible include 'Build Triggers', 'Build Environment', 'Build Steps', and 'Post-build Actions'. Buttons at the bottom include 'Add', 'Cancel', 'Save', and 'Apply'.

Not secure | 3.142.40.186:8080/job/slave/configure

Dashboard > slave > Configuration

Configure

Source Code Management

None

Git ?

Repositories ?

Repository URL ?
https://github.com/sangitagit/Terraform-python-TF.git

Credentials ?
sangitagit/******** (slave-deploy)

Add

Save Apply

This screenshot shows the 'Configuration' dialog for a Jenkins slave job, with the 'Source Code Management' tab selected. It is configured to use 'Git' with a repository URL of 'https://github.com/sangitagit/Terraform-python-TF.git' and credentials named 'sangitagit/******** (slave-deploy)'. Other tabs like 'General' and 'Build Triggers' are also visible.

Providing the commands to create the infrastructure automatically by installing terraform and executing the cloned repository of Terraform scripts

The screenshot shows the Jenkins job configuration interface. On the left, a sidebar lists various configuration sections: General, Source Code Management, Build Triggers, Build Environment, **Build Steps** (which is selected and highlighted in grey), and Post-build Actions. The main content area is titled "Configure" and contains a "Execute shell" section. It displays a command block with the following content:

```
#!/bin/bash
sudo yum install -y yum-utils
sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo
sudo yum -y install terraform
cd workspace/slave
sudo terraform init
sudo terraform apply --auto-approve
```

Below the command block are "Advanced" settings, a "Save" button, and an "Apply" button.

Build the newly created job

The screenshot shows the Jenkins job details page for build #4. The top navigation bar includes links for PAY, Citrix Receiver, AMEX_Link, Solve Monitor, New HUB loc, Citrix XenDesktop, Citrix XenApp, Tesco Rewards & R..., Cybersource Busine..., and Other bookmarks. The Jenkins logo is visible on the left, and the search bar contains "Search (CTRL+K)". The user "moni" is logged in, indicated by a user icon and the name "moni". The "log out" link is also present.

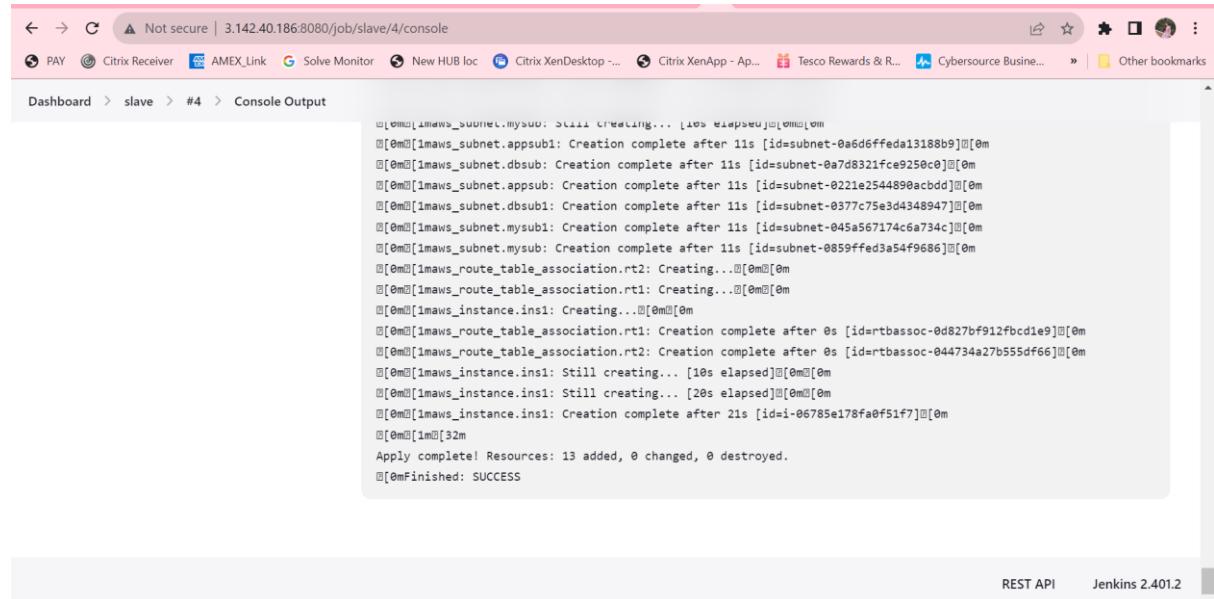
The main content area shows the build status as "Build #4 (22 Jul 2023, 15:45:17)" with a green checkmark icon. To the right, there are buttons for "Keep this build forever", "Add description", and "Delete build '#4'". The "Delete build" button is currently selected and highlighted in grey. Below the status, there are sections for "Changes", "Console Output", "Edit Build Information", "Delete build '#4'", "Git Build Data", and "Previous Build".

On the right side, detailed information about the build is provided, including the start time ("Started 2 hr 16 min ago"), duration ("Took 49 sec on slave"), and git commit information:

No changes.
Started by user [moni](#)
Revision: 3d6e01c3efaab82b7ff7b5e4e414cdadcf94eb7b
Repository: <https://github.com/sangitagit/Terraform-python-TF.git>
refs/remotes/origin/main

At the bottom of the page, a red button labeled "3.142.40.186:8080/job/slave/4/confirmDelete" is visible, indicating a pending deletion action.

Validate the job ran successfully

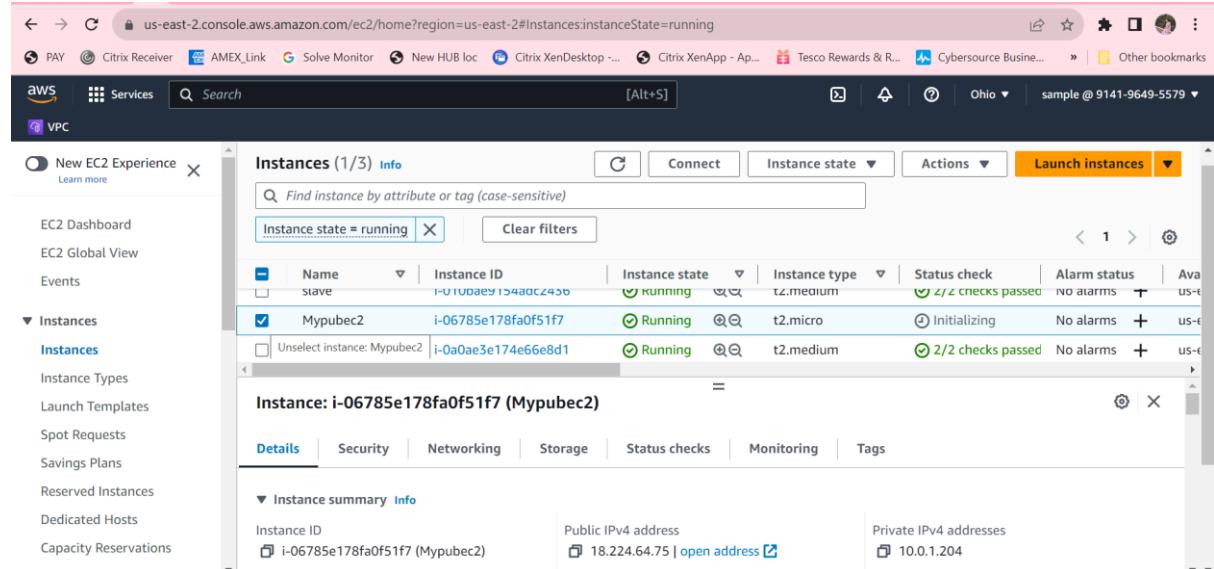


The screenshot shows a Jenkins console output window. The URL is 3.142.40.186:8080/job/slave/4/console. The log output is as follows:

```
[0m[1maws_subnet.mysub: Still creating... [0m
[0m[1maws_subnet.apsub1: Creation complete after 11s [id=subnet-0a6d6ffeda13188b9][0m
[0m[1maws_subnet.dbsub: Creation complete after 11s [id=subnet-0a7d8321fce9250c0][0m
[0m[1maws_subnet.apsub: Creation complete after 11s [id=subnet-0221e2544890acbdd][0m
[0m[1maws_subnet.dbsub1: Creation complete after 11s [id=subnet-0377c75e3d4348947][0m
[0m[1maws_subnet.mysub1: Creation complete after 11s [id=subnet-045a567174c6a734c][0m
[0m[1maws_subnet.mysub: Creation complete after 11s [id=subnet-0859ffed3a54f9686][0m
[0m[1maws_route_table_association.rt2: Creating...[0m[0m
[0m[1maws_route_table_association.rt1: Creating...[0m[0m
[0m[1maws_instance.ins1: Creating...[0m[0m
[0m[1maws_route_table_association.rt1: Creation complete after 0s [id=rtbassoc-0d827bf912fbcd1e9][0m
[0m[1maws_route_table_association.rt2: Creation complete after 0s [id=rtbassoc-044734a27b555df66][0m
[0m[1maws_instance.ins1: Still creating... [10s elapsed][0m[0m
[0m[1maws_instance.ins1: Still creating... [20s elapsed][0m[0m
[0m[1maws_instance.ins1: Creation complete after 21s [id=i-06785e178fa0f51f7][0m
[0m[1maws_instance.ins1: Apply complete! Resources: 13 added, 0 changed, 0 destroyed.
[0mFinished: SUCCESS
```

Newly created infrastructure

a) EC2 Instance : Mypubec2



The screenshot shows the AWS EC2 Instances page. The URL is us-east-2.console.aws.amazon.com/ec2/home?region=us-east-2#instances:instanceState=running. The instance list shows two instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status
slave	i-010daee9154adc245b	Running	t2.medium	2/2 checks passed	No alarms
Mypubec2	i-06785e178fa0f51f7	Running	t2.micro	Initializing	No alarms
	i-0a0ae3e174e66e8d1	Running	t2.medium	2/2 checks passed	No alarms

The instance Mypubec2 is currently initializing.

Using the public IP of the newly created instance browse on port 8000.

What is an Abalone:

Y What is abalone?

Watch later Share

What is abalone?

Sample Image Of Abalone

abalone Shell

The believed abalone healing properties of today are very similar to ancient beliefs of the shell. Abalone healing carries energies of protection and emotional balance. It brings with it a natural shielding that blesses the person holding it with

b) VPC : my_vpc

Your VPCs (2)

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR
-	vpc-0f2057c1e3b74e358	Available	172.31.0.0/16	-
my-vpc	vpc-057d61f62e195d8ea	Available	10.0.0.0/16	-

Virtual private cloud

Your VPCs New

Subnets

Route tables

Internet gateways

Egress-only internet gateways

DHCP option sets

Elastic IPs

Managed prefix lists

Endpoints

c) Security Groups : My-sub, My-sub1, app-sub, app-sub1, db-sub, db-sub1

Name	Subnet ID	State	VPC
db-sub1	subnet-0377c75e3d4348947	Available	vpc-057d61f62e195d8ea my-
-	subnet-0e2aa38289821dfc6	Available	vpc-0f2057c1e3b74e358
db-sub	subnet-0a7d8321fce9250c0	Available	vpc-057d61f62e195d8ea my-
-	subnet-012ee0524778b1dec	Available	vpc-0f2057c1e3b74e358
My-sub1	subnet-045a567174c6a734c	Available	vpc-057d61f62e195d8ea my-
app-sub	subnet-0a6d6ffeda13188b9	Available	vpc-057d61f62e195d8ea my-
app-sub	subnet-0221e2544890acbdd	Available	vpc-057d61f62e195d8ea my-
-	subnet-0ae3ce13f1c57fb8	Available	vpc-0f2057c1e3b74e358
My-sub	subnet-0859ffed3a54f9686	Available	vpc-057d61f62e195d8ea my-

d) Route Table : my-rt

Name	Route table ID	Explicit subnet associations	Edge associations	...
my-rt	rtb-0f2e830bb3f2849de	2 subnets	-	None
default-public	rtb-09b912a36c6df046b	subnet-0e2aa38289821d...	-	Yes
default-pvt	rtb-0eecef041cedbb11	subnet-012ee0524778b1...	-	No

Select a route table

Wordpress Installation with docker compose in Jenkins

The screenshot shows the AWS EC2 Instances page. The left sidebar is collapsed. The main area displays a table of instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Ava
slave	i-010bae9154adc2436	Running	t2.medium	2/2 checks passed	No alarms	us-e
master	i-0a0ae3e174e66e8d1	Running	t2.medium	2/2 checks passed	No alarms	us-e

Below the table, the details for the selected instance ('slave') are shown:

Instance: i-010bae9154adc2436 (slave)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance summary

Instance ID i-010bae9154adc2436 (slave)	Public IPv4 address 18.217.220.47 open address	Private IPv4 addresses 172.31.3.69
--	---	---------------------------------------

The screenshot shows the Jenkins job configuration page for a job named 'slave'. The left sidebar lists configuration tabs: General, Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The 'General' tab is selected.

Configure

General

GitHub project
 This project is parameterised ?
 Throttle builds ?
 Execute concurrent builds if necessary ?
 Restrict where this project can be run ?

Label Expression ?
slave

Label slave matches 1 node. Permissions or other restrictions provided by plugins may further reduce that list.

Source Code Management

Advanced

Save | Apply

Cloning the GIT repository containing the docker-compose.git

The screenshot shows the Jenkins configuration page for a slave job. Under the 'Source Code Management' section, 'Git' is selected. The 'Repository URL' is set to `https://github.com/sangitagit/docker-compose.git`, and the 'Credentials' dropdown contains `sangitagit/******** (slave-deploy)`. There are 'Save' and 'Apply' buttons at the bottom.

Updating the execute shell commands to :

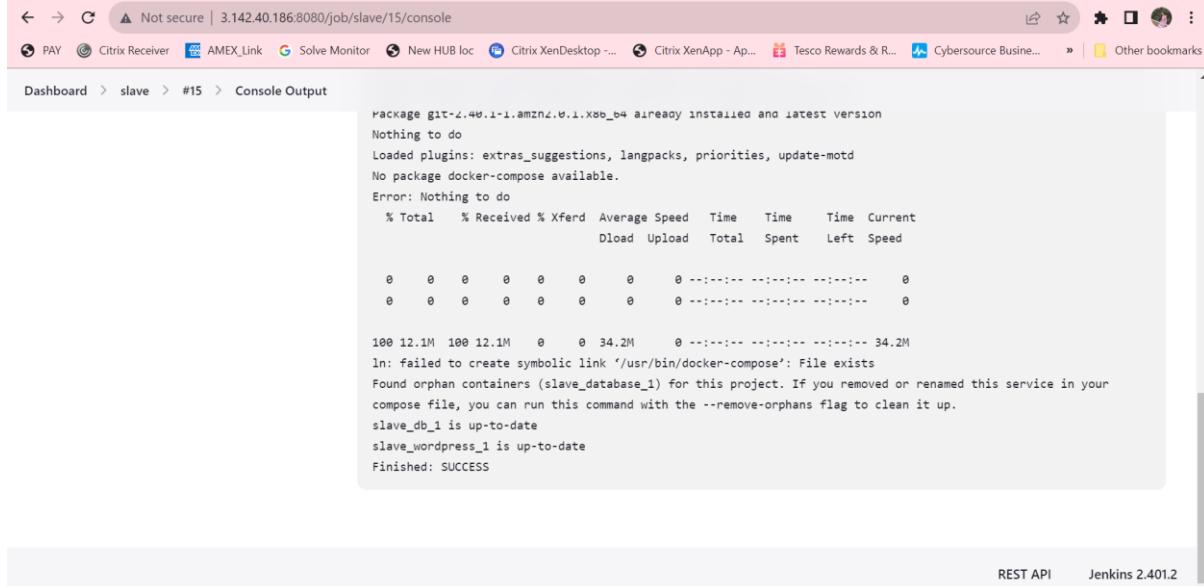
- Install docker
- Install git
- Install docker-compose
- Execute the docker-compose in detach mode

The screenshot shows the Jenkins configuration page for a slave job. Under the 'Build Steps' section, there is a large text area containing the following command:

```
#!/bin/bash
sudo yum -y install docker
sudo service docker start
sudo usermod -a -G docker ec2-user
sudo chmod 666 /var/run/docker.sock
sudo systemctl enable docker
sudo chkconfig docker on
sudo yum install -y git
sudo yum -y install docker-compose -y
sudo curl -L https://github.com/docker/compose/releases/download/1.29.2/docker-compose-Linux-x86_64 -o
sudo chmod +x /usr/local/bin/docker-compose
sudo ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
cd /home/slave/workspace/slave
docker-compose up -d
```

There are 'Save' and 'Apply' buttons at the bottom.

Build the job and validate it is successful.

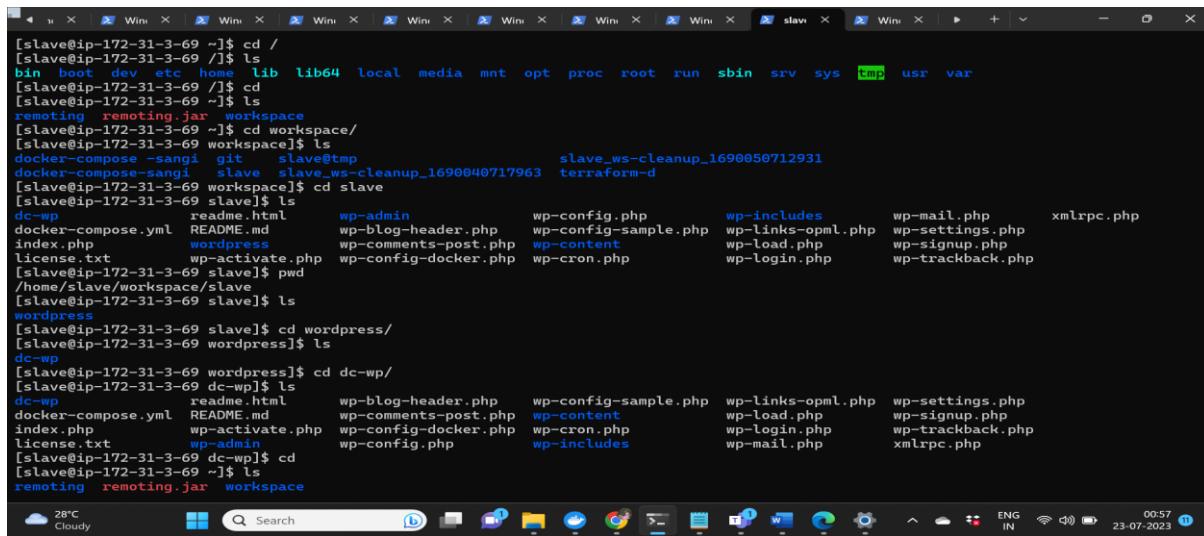


The screenshot shows a browser window with the URL 3.142.40.186:8080/job/slave/15/console. The page title is "Console Output". The content of the console is as follows:

```
Package git-2.40.1-1.amzn2.0.1.x86_64 already installed and latest version
Nothing to do
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No package docker-compose available.
Error: Nothing to do
      % Total    % Received   % Xferd  Average Speed   Time     Time   Current
          Dload   Upload   Total Spent  Left  Speed
0       0       0       0       0       0       0 --::-- --::-- --::-- 0
0       0       0       0       0       0       0 --::-- --::-- --::-- 0

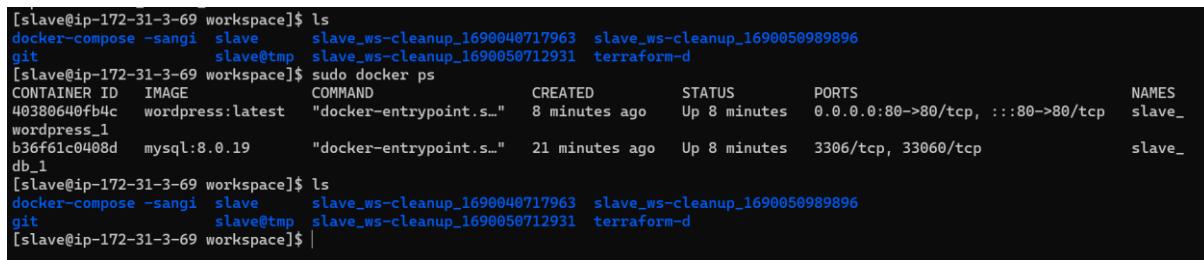
100 12.1M 100 12.1M 0 34.2M 0 --::-- --::-- --::-- 34.2M
ln: failed to create symbolic link '/usr/bin/docker-compose': File exists
Found orphan containers (slave_database_1) for this project. If you removed or renamed this service in your
compose file, you can run this command with the --remove-orphan flag to clean it up.
slave_db_1 is up-to-date
slave_wordpress_1 is up-to-date
Finished: SUCCESS
```

Also validate the files created in the terminal



The screenshot shows a Windows terminal window with multiple tabs. The active tab shows the following directory structure and files:

```
[slave@ip-172-31-3-69 ~]$ cd /
[slave@ip-172-31-3-69 /]$ ls
bin boot dev etc home lib lib64 local media mnt opt proc root run sbin srv sys tmp usr var
[slave@ip-172-31-3-69 /]$ cd workspace/
[slave@ip-172-31-3-69 workspace]$ ls
remoteing remoting.jar workspace
[slave@ip-172-31-3-69 ~]$ cd workspace/
[slave@ip-172-31-3-69 workspace]$ ls
docker-compose -sangi git slave@tmp slave_ws-cleanup_1690040717963 terraform-d
[slave@ip-172-31-3-69 workspace]$ cd slave
[slave@ip-172-31-3-69 slave]$ ls
dc-wp readme.html wp-admin wp-config.php wp-includes wp-mail.php xmlrpc.php
docker-compose.yml README.md wp-blog-header.php wp-config-sample.php wp-links-opml.php wp-settings.php
index.php wordpress wp-comments-post.php wp-content wp-load.php wp-signup.php
license.txt wp-activate.php wp-config-docker.php wp-cron.php wp-login.php wp-trackback.php
[slave@ip-172-31-3-69 slave]$ pwd
/home/slave/workspace/slave
[slave@ip-172-31-3-69 slave]$ ls
wordpress
[slave@ip-172-31-3-69 slave]$ cd wordpress/
[slave@ip-172-31-3-69 wordpress]$ ls
dc-wp
[slave@ip-172-31-3-69 wordpress]$ cd dc-wp/
[slave@ip-172-31-3-69 dc-wp]$ ls
dc-wp readme.html wp-blog-header.php wp-config-sample.php wp-links-opml.php wp-settings.php
docker-compose.yml README.md wp-comments-post.php wp-content wp-load.php wp-signup.php
index.php wp-activate.php wp-config-docker.php wp-cron.php wp-login.php wp-trackback.php
license.txt wp-admin wp-config.php wp-includes wp-mail.php xmlrpc.php
[slave@ip-172-31-3-69 dc-wp]$ cd ..
[slave@ip-172-31-3-69 ~]$ ls
remoteing remoting.jar workspace
```



The screenshot shows a terminal window with the following commands and output:

```
[slave@ip-172-31-3-69 workspace]$ ls
docker-compose -sangi slave slave_ws-cleanup_1690040717963 slave_ws-cleanup_1690050989896
git slave@tmp slave_ws-cleanup_1690050712931 terraform-d
[slave@ip-172-31-3-69 workspace]$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
40380640fb4c wordpress:latest "docker-entrypoint.s..." 8 minutes ago Up 8 minutes 0.0.0.0:80->80/tcp, :::80->80/tcp slave_
wordpress_1
b36f61c0408d mysql:8.0.19 "docker-entrypoint.s..." 21 minutes ago Up 8 minutes 3306/tcp, 33060/tcp slave_
db_1
[slave@ip-172-31-3-69 workspace]$ ls
docker-compose -sangi slave slave_ws-cleanup_1690040717963 slave_ws-cleanup_1690050989896
git slave@tmp slave_ws-cleanup_1690050712931 terraform-d
[slave@ip-172-31-3-69 workspace]$ |
```

Use the public IP of the slave and browse the IP using the port 80 in the browser to see the Wordpress application up and running

The screenshot shows the AWS EC2 Instances page. There are two instances listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Ava
slave	i-010bae9154adc2436	Running	t2.medium	2/2 checks passed	No alarms	us-e
master	i-0a0ae3e174e66e8d1	Running	t2.medium	2/2 checks passed	No alarms	us-e

The 'slave' instance is selected. A tooltip for its Public IPv4 address (18.217.220.47) is displayed, along with its Instance ID (i-010bae9154adc2436). Other details shown include Private IPv4 addresses (172.31.3.69) and Status checks (2/2 checks passed).

The screenshot shows the AWS EC2 Instances page. The 'slave' instance is selected. The network interface details for the slave instance are displayed:

Interface	MAC Address	Port	Protocol	Status
-	sgr-03d603884111e6b92	80	TCP	0.0.0.0/0
-	sgr-0c6398ff697958b85	8080	TCP	0.0.0.0/0
-	sgr-095c62e0e8c0f7b4c	3306	TCP	0.0.0.0/0
-	sgr-08a70144dd32c4dfc	22	TCP	0.0.0.0/0

The screenshot shows a web browser displaying the WordPress installation page at 18.217.220.47/wp-admin/install.php. The page features the classic blue and white WordPress logo. Below it, a dropdown menu lists various languages:

- English (United States)
- Afrikaans
- አማርኛ
- Aragonés
- العربية
- العربية المغربية
- অসমীয়া
- گۈئىي ادەرەجىان
- Azerbaiyan dili
- Беларуская мова
- Български
- বাংলা
- Bosanski
- Català
- Cebuano
- Čeština
- Cymraeg
- Dansk