

# Setting up Master Slave



VS Charan

3 min read · Just now



first we need to create 3 VM using amazon linux

```
1 EC2 Instance -- master
2 EC2 Instacne -- slave-1
3 EC2 Instacne -- slave-2
```

The screenshot shows the AWS Management Console 'Instances' page. It lists three EC2 instances: 'Master', 'slave-1', and 'slave-2'. All three are in a 'Running' state, using 't2.micro' instance types, and have '2/2 checks passed'. They are located in the 'us-east-1c' availability zone. The 'Master' instance has a public IPv4 DNS of 'ec2-107-23-221-112.co', 'slave-1' has 'ec2-52-205-147-45.co', and 'slave-2' has 'ec2-54-159-117-117.co'.

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
<input checked="" type="checkbox"/>	Master	i-060cd9d8a139d79fc	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c	ec2-107-23-221-112.co
<input type="checkbox"/>	slave-1	i-0e907a6d29a4a90fa	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c	ec2-52-205-147-45.co
<input type="checkbox"/>	slave-2	i-09f1c8ac1d3d483d5	Running	t2.micro	2/2 checks passed	No alarms	us-east-1c	ec2-54-159-117-117.co

## Install Java — Master

```
sudo yum install java-17-amazon-corretto-devel -y
```

## Install Jenkins

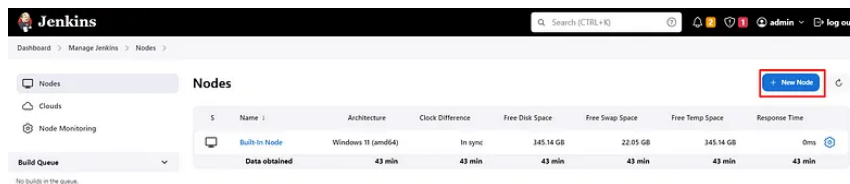
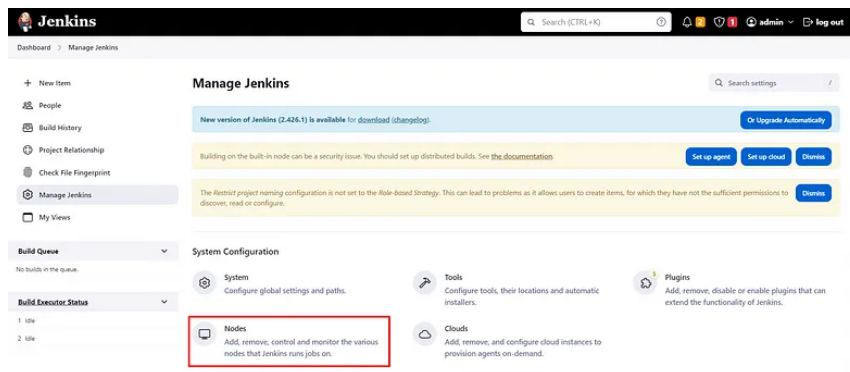
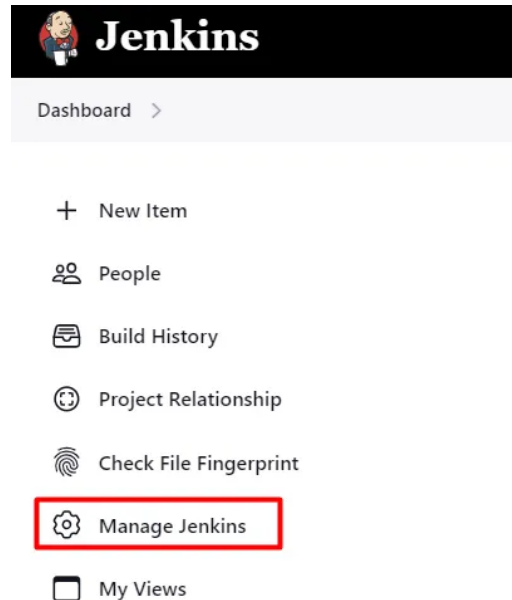
```
sudo yum update -y
sudo wget -O /etc/yum.repos.d/jenkins.repo http://pkg.jenkins.io/redhat/jenkins.
sudo rpm --import https://pkg.jenkins.io/redhat/jenkins.io-2023.key
sudo yum install jenkins -y
systemctl daemon-reload
sudo systemctl start jenkins
systemctl status jenkins
```

## connect to slave — 1

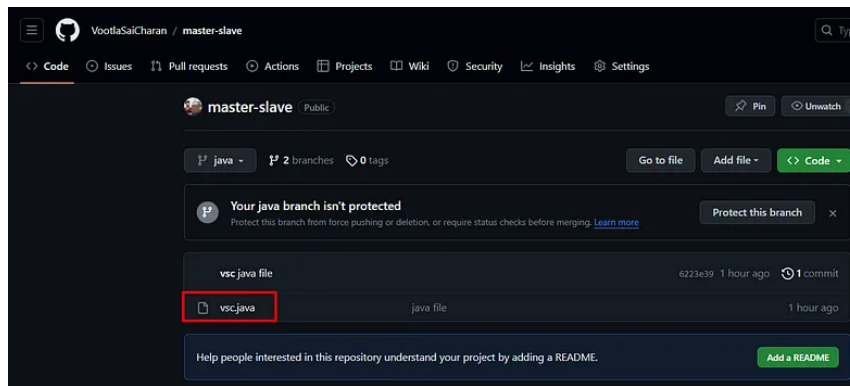
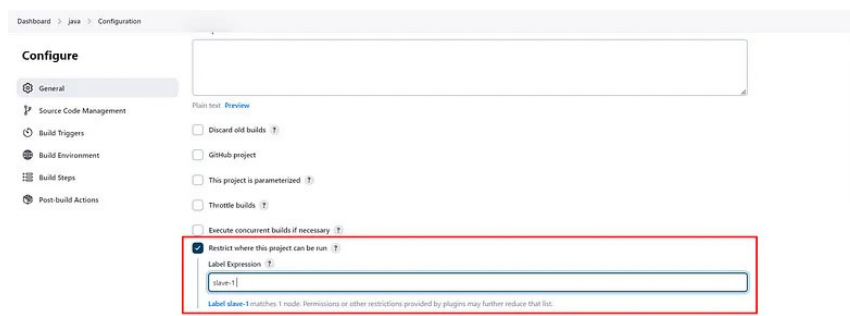
after connecting to the slave -1 install java and git (depends on your project install those applications). Then create a Directory and enter the command and copy the path.

pwd

## Connect to Jenkins and Create 2 — Node (slave-1 & slave-2)








In Build environment → add build step → Execute shell → save → **Build Now**

```
javac filename.java  
java filename
```



# Jenkins

Dashboard > java >

Status

</> Changes

Workspace

Build Now

Configure

Delete Project

Rename

Build History

trend

Filter builds... /

#1

Nov 21, 2023, 4:05 PM

Atom feed for all Atom feed for failures

build Console Output

Jenkins

Search (CTRL+K)

admin

log out

Dashboard > java > #1 > Console Output

Status

</> Changes

Console Output

View as plain text

Edit Build Information

Delete build #1

Git Build Data

Console Output

Started by user admin

Running as SYSTEM

Building remotely on slave-1 in workspace /home/ec2-user/java/workspace/java

The recommended git tool is: NONE

No credentials specified

> git rev-parse --resolve-git-dir /home/ec2-user/java/workspace/java/.git # timeout=10

Fetching changes from the remote Git repository

> git config remote.origin.url https://github.com/Voot1d4Charan/master-slave.git # timeout=10

Fetching upstream changes from https://github.com/Voot1d4Charan/master-slave.git

> git --version # timeout=10

> git --version # git version 2.40.1

> git fetch --tags --force --progress -- https://github.com/Voot1d4Charan/master-slave.git --refs/heads/\*:refs/remotes/origin/\* # timeout=10

> git rev-parse refs/remotes/origin/java^{commit} # timeout=10

Checking out Revision 6223a395710ccf732756a02bed7da912ee53 (refs/remotes/origin/java)

> git config core.sparsecheckout # timeout=10

> git checkout -f 6223a395710ccf732756a02bed7da912ee53 # timeout=10

Commit message: "java file"

First time build. Skipping changelog

[java] \$ /bin/sh -xe /tmp/jenkins17861407087684099328.sh

• java vsc.java

• java vsc

a 13

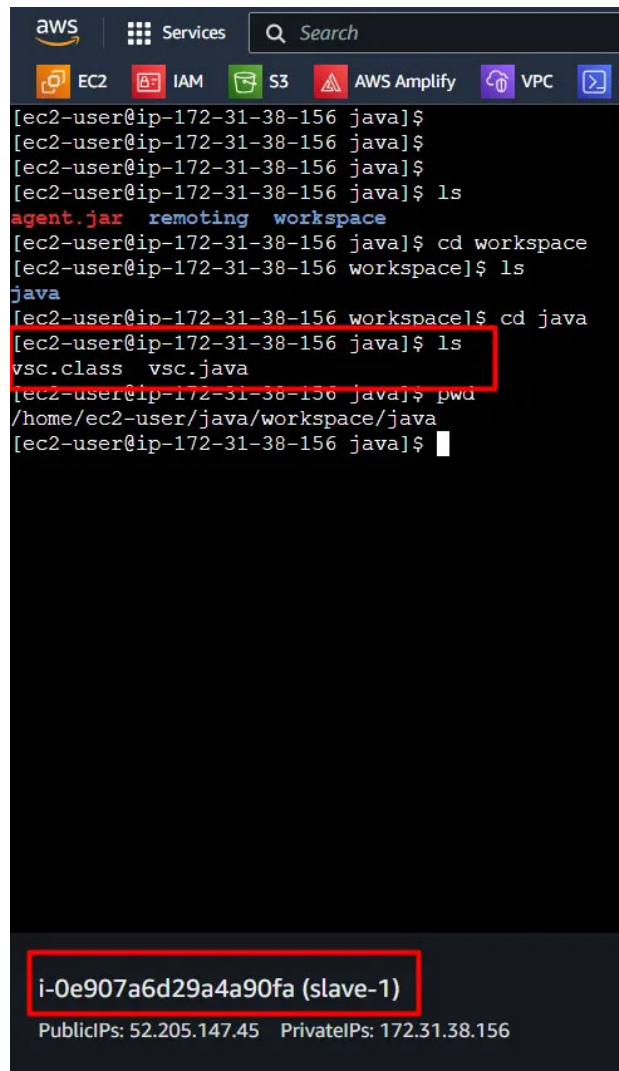
b 15

c 15

d 18

Finished: SUCCESS

we can also see .class files in slave — 1

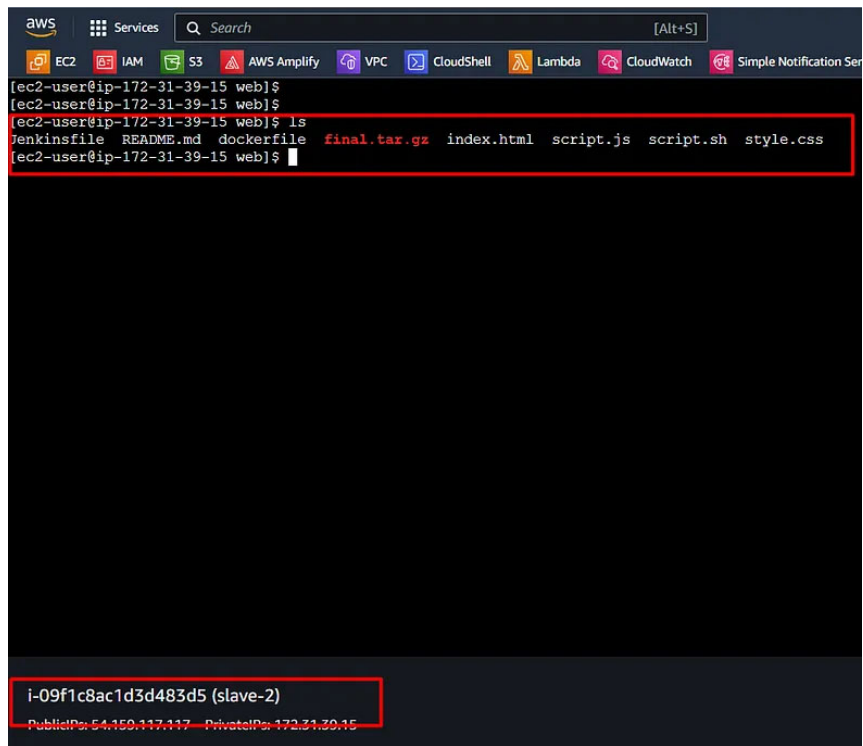


```
aws Services Search
EC2 IAM S3 AWS Amplify VPC
[ec2-user@ip-172-31-38-156 java]$
[ec2-user@ip-172-31-38-156 java]$
[ec2-user@ip-172-31-38-156 java]$
[ec2-user@ip-172-31-38-156 java]$ ls
agent.jar remoting workspace
[ec2-user@ip-172-31-38-156 java]$ cd workspace
[ec2-user@ip-172-31-38-156 workspace]$ ls
java
[ec2-user@ip-172-31-38-156 workspace]$ cd java
[ec2-user@ip-172-31-38-156 java]$ ls
vsc.class vsc.java
[ec2-user@ip-172-31-38-156 java]$ pwd
/home/ec2-user/java/workspace/java
[ec2-user@ip-172-31-38-156 java]$
```

**i-0e907a6d29a4a90fa (slave-1)**  
PublicIPs: 52.205.147.45 PrivateIPs: 172.31.38.156

## Now Connect to slave — 2

connect the node (slave-2) in slave-2 (EC2) then create a Jenkins free style project and change slave — 2 in Restrict where this project can be run and add git repo url in SCM change branch name to web save and build. you can also check the files in slave — 2 (EC2)



The screenshot shows an AWS CloudShell terminal window. The top bar includes the AWS logo, a 'Services' menu, a search bar, and a '[Alt+S]' shortcut. Below this is a row of service icons: EC2, IAM, S3, AWS Amplify, VPC, CloudShell, Lambda, CloudWatch, and Simple Notification Service. The terminal content shows a user logged in as 'ec2-user' on an instance with ID 'i-09f1c8ac1d3d483d5' (labeled as 'slave-2'). The user runs the 'ls' command, and the output lists several files: 'jenkinsfile', 'README.md', 'dockerfile', 'final.tar.gz', 'index.html', 'script.js', 'script.sh', and 'style.css'. The instance's public IP is 54.150.117.117 and the private IP is 172.31.20.15.

```
aws
Services
Search [Alt+S]
EC2 IAM S3 AWS Amplify VPC CloudShell Lambda CloudWatch Simple Notification Ser

[ec2-user@ip-172-31-39-15 web]$
[ec2-user@ip-172-31-39-15 web]$
[ec2-user@ip-172-31-39-15 web]$ ls
jenkinsfile README.md dockerfile final.tar.gz index.html script.js script.sh style.css
[ec2-user@ip-172-31-39-15 web]$

i-09f1c8ac1d3d483d5 (slave-2)
PublicPoi: 54.150.117.117 PrivatePoi: 172.31.20.15
```

