

# cprime

**Master-Slave Architecture**



# Jenkins

**Master & Agents**

**NAME : T MANOJ**

**EMPID : LYAKE2KHS**

# What is Jenkins?

Jenkins is an open-source tool written in Java that automates software development lifecycle tasks like build, test, deploy, and more. In this article, we will discuss how to configure Jenkins master-slave setup also called master-slave or master-agent architecture.

## Why do we need Master-Slave architecture?

When we build the Jenkins job in a single Jenkins master node then Jenkins uses the resource of the base machine and If no executor is available then the jobs are queued in the Jenkins server. Sometimes you might need several different environments to test your builds. A single Jenkins server cannot do this. It is recommended not to run different jobs in the same system that required a different environment. In such scenarios where we need a different machine with a different environment that takes the specific job from the master to build.

## Jenkins Distributed Architecture

Jenkins uses A Master-Slave architecture to manage distributed builds. The machine where we install Jenkins software will be Jenkins master and that runs on port 8080 by default. On the slave machine, we install a program called Agent. This agent requires JVM. This agent executes the tasks provided by the Jenkins master. We can launch n numbers of agents and we can configure which task will be run on which agent server from Jenkins master by assigning the agent to the task.

# Detailed Steps for Jenkins Master-Slave Architecture Setup

## Prerequisites

- 2 Linux servers (one for Master, one for Slave/Agent)
- Internet access on both servers

The image displays two screenshots of the AWS Management Console, showing the details of two EC2 instances. The top screenshot shows the instance summary for **i-021c5025b02d7fea3** (aml), which is a **t2.micro** instance running in the **us-east-1** region. The instance is in the **Running** state. The bottom screenshot shows the instance summary for **i-053a1de95d112e5fa** (UBUNTU), which is also a **t2.micro** instance running in the **us-east-1** region. The instance is in the **Running** state. Both instances are part of the **cloudprotocols-hybrid-devops** group.

**Instance summary for i-021c5025b02d7fea3 (aml)**

Property	Value
Instance ID	i-021c5025b02d7fea3
Public IPv4 address	18.226.180.185   <a href="#">open address</a>
Private IPv4 addresses	172.31.10.127
Instance state	Running
Public IPv4 DNS	ec2-18-226-180-185.us-east-2.compute.amazonaws.com   <a href="#">open address</a>
Private IP DNS name (IPv4 only)	ip-172-31-10-127.us-east-2.compute.internal
Instance type	t2.micro
Hostnames	IP name: ip-172-31-10-127.us-east-2.compute.internal
Answer private resource DNS name	IPv4 (A)
Elastic IP addresses	-

**Instance summary for i-053a1de95d112e5fa (UBUNTU)**

Property	Value
Instance ID	i-053a1de95d112e5fa
Public IPv4 address	54.209.249.218   <a href="#">open address</a>
Private IPv4 addresses	172.31.21.70
Instance state	Running
Public IPv4 DNS	ec2-54-209-249-218.compute-1.amazonaws.com   <a href="#">open address</a>
Private IP DNS name (IPv4 only)	ip-172-31-21-70.ec2.internal
Instance type	t2.micro
Hostnames	IP name: ip-172-31-21-70.ec2.internal
Answer private resource DNS name	IPv4 (A)
Elastic IP addresses	-
Auto-assigned IP address	54.209.249.218
VPC ID	vpc-03310000
AWS Compute Optimizer finding	Optimized

## Step 1: Install Jenkins on Master Node

Install Java on the Master node:

```
sudo apt install openjdk-11-jdk -y
```

```
ubuntu@ip-172-31-21-70: ~  
ubuntu@ip-172-31-21-70:~$ sudo apt update  
sudo apt install openjdk-11-jdk -y  
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease  
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease [126 kB]  
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease [126 kB]  
Get:4 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]  
Ign:5 https://pkg.jenkins.io/debian-stable binary/ InRelease  
Hit:6 https://pkg.jenkins.io/debian-stable binary/ Release  
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [979 kB]  
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Components [151 kB]  
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [1046 kB]  
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Components [365 kB]  
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/restricted amd64 Components [212 B]  
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 Components [940 B]  
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 Components [7060 B]  
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Components [15.7 kB]  
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 Components [212 B]  
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 Components [212 B]  
Get:18 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [731 kB]  
Get:19 http://security.ubuntu.com/ubuntu noble-security/main amd64 Components [8996 B]  
Get:20 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [827 kB]  
Get:21 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [52.0 kB]  
Get:22 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Components [212 B]  
Get:23 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [212 B]  
Fetched 4563 kB in 2s (2708 kB/s)  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
34 packages can be upgraded. Run 'apt list --upgradable' to see them.  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
openjdk-11-jdk is already the newest version (17.0.14+7-1-24.04).  
0 upgraded, 0 newly installed, 0 to remove and 34 not upgraded.  
ubuntu@ip-172-31-21-70:~$
```

Add the Jenkins repository and install Jenkins:

```
sudo curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key  
| sudo tee /usr/share/keyrings/jenkins-keyring.asc > /dev/null
```

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

```
ubuntu@ip-172-31-21-70: ~  
ubuntu@ip-172-31-21-70:~$ echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \  
https://pkg.jenkins.io/debian-stable binary/ | sudo tee \  
/etc/apt/sources.list.d/jenkins.list > /dev/null  
ubuntu@ip-172-31-21-70:~$
```

```
sudo apt-get update
```

```
ubuntu@ip-172-31-21-70:~$ sudo apt-get update  
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble InRelease  
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease  
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease  
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease  
Ign:5 https://pkg.jenkins.io/debian-stable binary/ InRelease  
Hit:6 https://pkg.jenkins.io/debian-stable binary/ Release  
Reading package lists... Done
```

```
sudo apt-get install jenkins -y
```

```
ubuntu@ip-172-31-21-70:~$ sudo apt-get install jenkins
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
jenkins is already the newest version (2.492.2).
0 upgraded, 0 newly installed, 0 to remove and 34 not upgraded.
ubuntu@ip-172-31-21-70:~$
```

Verify Jenkins is running:

```
sudo systemctl status jenkins
```

```
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; preset: enabled)
   Active: active (running) since Tue 2025-04-01 06:49:49 UTC; 7h ago
     Main PID: 4854 (java)
       Tasks: 38 (limit: 1129)
      Memory: 435.3M (peak: 473.2M)
         CPU: 1min 32.405s
        CGroup: /system.slice/jenkins.service
               └─4854 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=/var/cache/jenkins/war --httpPort=8080

Apr 01 06:51:39 ip-172-31-21-70 jenkins[4854]: 2025-04-01 06:51:39.756+0000 [id=263] INFO jenkins.InitReactorRunner$1onAttained: Started all
Apr 01 06:51:39 ip-172-31-21-70 jenkins[4854]: 2025-04-01 06:51:39.761+0000 [id=263] INFO jenkins.InitReactorRunner$1onAttained: Augmented all
Apr 01 06:51:40 ip-172-31-21-70 jenkins[4854]: 2025-04-01 06:51:40.225+0000 [id=263] INFO jenkins.InitReactorRunner$1onAttained: System config
Apr 01 06:51:40 ip-172-31-21-70 jenkins[4854]: 2025-04-01 06:51:40.225+0000 [id=263] INFO jenkins.InitReactorRunner$1onAttained: System config
Apr 01 06:51:40 ip-172-31-21-70 jenkins[4854]: 2025-04-01 06:51:40.226+0000 [id=263] INFO jenkins.InitReactorRunner$1onAttained: Loaded all
Apr 01 06:51:40 ip-172-31-21-70 jenkins[4854]: 2025-04-01 06:51:40.249+0000 [id=263] INFO jenkins.InitReactorRunner$1onAttained: Configuration
Apr 01 06:51:40 ip-172-31-21-70 jenkins[4854]: 2025-04-01 06:51:40.485+0000 [id=263] INFO jenkins.InitReactorRunner$1onAttained: Completed in
Apr 01 06:51:40 ip-172-31-21-70 jenkins[4854]: 2025-04-01 06:51:40.489+0000 [id=32] INFO hudson.UpdateCenter$CompleteBatchJob#run: Completed in
Apr 01 06:53:57 ip-172-31-21-70 jenkins[4854]: 2025-04-01 06:53:57.349+0000 [id=15] WARNING hudson.security.csrf.CrumbFilter#doFilter: Found
Apr 01 06:53:57 ip-172-31-21-70 jenkins[4854]: 2025-04-01 06:53:57.350+0000 [id=15] WARNING hudson.security.csrf.CrumbFilter#doFilter: No val
lines 1-20/20 (END)
```

## Step 2: Install Java on the Agent/Slave Node

Install Java on the Agent node:

```
sudo apt install openjdk-11-jdk -y
```

```
ec2-user@ip-172-31-6-25:~  
Verifying : pixman-0.43.4-1.amzn2023.0.4.x86_64  
Verifying : xml-common-0.6.3-56.amzn2023.0.2.noarch  
34/35  
35/35  
-----  
WARNING:  
A newer release of "Amazon Linux" is available.  
  
Available Versions:  
  
Version 2023.7.20250331:  
Run the following command to upgrade to 2023.7.20250331:  
  
dnf upgrade --releasever=2023.7.20250331  
  
Release notes:  
https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes-2023.7.20250331.html  
-----  
Installed:  
alsa-lib-1.2.7.2-1.amzn2023.0.2.x86_64  
dejavu-sans-fonts-2.37-16.amzn2023.0.2.noarch  
dejavu-serif-fonts-2.37-16.amzn2023.0.2.noarch  
fonts-filesystem-1:2.0.5-12.amzn2023.0.2.noarch  
giflib-5.2.1-9.amzn2023.0.1.x86_64  
google-noto-sans-vf-fonts-20201206-2.amzn2023.0.2.noarch  
harfbuzz-7.0.0-2.amzn2023.0.2.x86_64  
java-17-amazon-corretto-headless-1:17.0.14+7-1.amzn2023.1.x86_64  
langpacks-core-font-en-3.0-21.amzn2023.0.4.noarch  
libSM-1.2.4-3.amzn2023.0.1.x86_64  
libX11-common-1.0.10-2.amzn2023.0.1.noarch  
libXext-1.3.6-1.amzn2023.0.1.x86_64  
libXinerama-1.1.5-6.amzn2023.0.1.x86_64  
libXrender-0.9.11-6.amzn2023.0.1.x86_64  
libXtst-1.2.5-1.amzn2023.0.1.x86_64  
libjpeg-turbo-2.1.4-2.amzn2023.0.5.x86_64  
libxcb-1.17.0-1.amzn2023.0.1.x86_64  
xml-common-0.6.3-56.amzn2023.0.2.noarch  
cairo-1.18.0-4.amzn2023.0.1.x86_64  
dejavu-sans-mono-fonts-2.37-16.amzn2023.0.2.noarch  
fontconfig-2.13.94-2.amzn2023.0.2.x86_64  
freetype-2.13.2-5.amzn2023.0.1.x86_64  
google-noto-fonts-common-20201206-2.amzn2023.0.2.noarch  
graphite2-1.3.14-7.amzn2023.0.2.x86_64  
java-17-amazon-corretto-1:17.0.14+7-1.amzn2023.1.x86_64  
javapackages-filesystem-6.0.0-7.amzn2023.0.6.noarch  
libICE-1.1.1-3.amzn2023.0.1.x86_64  
libX11-1.8.10-2.amzn2023.0.1.x86_64  
libXau-1.0.11-6.amzn2023.0.1.x86_64  
libXi-1.8.2-1.amzn2023.0.1.x86_64  
libXrandr-1.5.4-3.amzn2023.0.1.x86_64  
libXt-1.3.0-3.amzn2023.0.1.x86_64  
libbrotli-1.0.9-4.amzn2023.0.2.x86_64  
libpng-2:1.6.37-10.amzn2023.0.6.x86_64  
pixman-0.43.4-1.amzn2023.0.4.x86_64  
-----  
Complete!  
[ec2-user@ip-172-31-6-25 ~]$ java --version  
openjdk 17.0.14 2025-01-21 LTS  
OpenJDK Runtime Environment Corretto-17.0.14.7.1 (build 17.0.14+7-LTS)  
OpenJDK 64-Bit Server VM Corretto-17.0.14.7.1 (build 17.0.14+7-LTS, mixed mode, sharing)  
[ec2-user@ip-172-31-6-25 ~]$
```

## Step 3: Configure the Master-Agent Setup

1. Access the Jenkins web interface (typically [http://\[master-ip\]:8080/](http://[master-ip]:8080/))
2. Complete initial Jenkins setup if required (follow the prompts to unlock Jenkins)
3. Set up the agent:
  - Navigate to Dashboard
  - Click on "Manage Jenkins"
  - Click on "Nodes and Clouds"
  - Click on "New Node" or "Set up an agent"
  - Add a node name of your choice
  - Select "Permanent Agent" and click "OK"

← → ↻ Not secure 54.209.249.218:8080/computer/new

**Jenkins** 🔍 🔔 🔒 🔴 T MANOJ ⌵ 🚪 log out

Dashboard > Nodes > New node

### New node

Node name

Master-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

REST API Jenkins 2.492.2

4. Configure the agent node with these settings:

- Add a description (optional)
- Number of executors: 2 (this determines how many parallel jobs the node can execute)

← → ↻ Not secure 54.209.249.218:8080/computer/createItem

**Jenkins** 🔍 🔔 🔒 🔴 T MANOJ ⌵ 🚪 log out

Dashboard > Nodes >

Name ?

Master-slave

Description ?

Master Slave for Jenkins Connection

Plain text Preview

Number of executors ?

2

Save

- Remote root directory: `/home/ec2-user` (workspace directory on the agent)
- Labels: Add a descriptive label (e.g., "linux-agent")
- Usage: Select "Use this node as much as possible"
- Launch method: Choose "Launch agent by connecting it to the controller"
- Check "Disable WorkDir"
- Availability: Set to "Keep the agent online as much as possible"
- Click "Save"

The screenshot shows the Jenkins 'Create Item' page for a new node. The browser address bar shows '54.209.249.218:8080/computer/createItem'. The Jenkins logo is in the top left. The page title is 'Nodes'. The form fields are as follows:

- Name: Master-slave
- Description: Master Slave for Jenkins Connection
- Plain text: Preview
- Number of executors: 2
- Remote root directory: /opt/build
- Labels: slave-node
- Usage: Use this node as much as possible
- Launch method: Launch agent by connecting it to the controller
- Availability: Keep this agent online as much as possible
- Node Properties: ☒ Disable deferred wipeout on this node

A 'Save' button is at the bottom left.

## 5. Connect the agent to the master:

- After saving, you'll see a command to run on the agent
- Copy this command
- SSH into your agent node

Run from agent command line: (Unix)

```
curl -sO http://54.209.249.218:8080/jnlpJars/agent.jar
java -jar agent.jar -url http://54.209.249.218:8080/ -secret a65e0e0fbb0693ec65c39afc05d45d5c6abcc434811e4c82146bfd651764bc8 -name "master-slave" -webSocket -
workDir "/home/ec2-user"
```

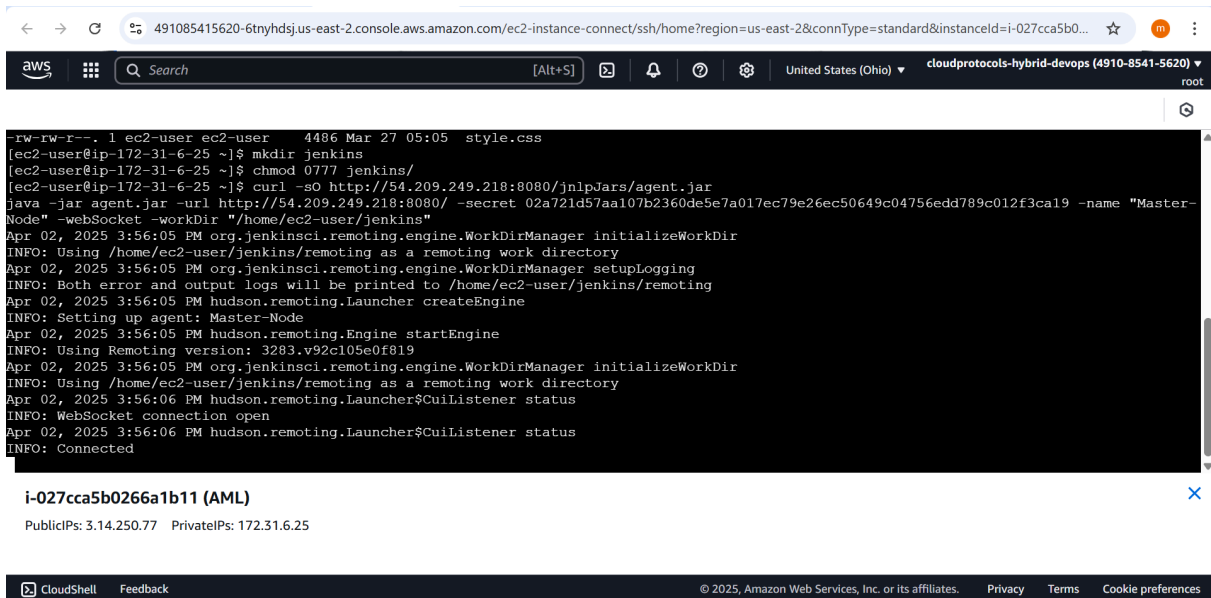
Make sure the directory has appropriate permissions:

```
sudo chown -R ec2-user:ec2-user /home/ec2-user
```

- Paste and run the Java command provided by Jenkins
- ## 6. Verify the connection:

- Go back to Jenkins UI
- Check "Nodes and Clouds" page
- Your agent should now show as "Connected"





```
-rw-rw-r--. 1 ec2-user ec2-user 4486 Mar 27 05:05 style.css
[ec2-user@ip-172-31-6-25 ~]$ mkdir jenkins
[ec2-user@ip-172-31-6-25 ~]$ chmod 0777 jenkins/
[ec2-user@ip-172-31-6-25 ~]$ curl -sO http://54.209.249.218:8080/jnlpJars/agent.jar
java -jar agent.jar -url http://54.209.249.218:8080/ -secret 02a721d57aa107b2360de5e7a017ec79e26ec50649c04756edd789c012f3ca19 --name "Master-Node" -webSocket -workDir "/home/ec2-user/jenkins"
Apr 02, 2025 3:56:05 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/ec2-user/jenkins/remoting as a remoting work directory
Apr 02, 2025 3:56:05 PM org.jenkinsci.remoting.engine.WorkDirManager setupLogging
INFO: Both error and output logs will be printed to /home/ec2-user/jenkins/remoting
Apr 02, 2025 3:56:05 PM hudson.remoting.Launcher createEngine
INFO: Setting up agent: Master-Node
Apr 02, 2025 3:56:05 PM hudson.remoting.Engine startEngine
INFO: Using Remoting version: 3283.v92c105e0f819
Apr 02, 2025 3:56:05 PM org.jenkinsci.remoting.engine.WorkDirManager initializeWorkDir
INFO: Using /home/ec2-user/jenkins/remoting as a remoting work directory
Apr 02, 2025 3:56:06 PM hudson.remoting.Launcher$CuiListener status
INFO: WebSocket connection open
Apr 02, 2025 3:56:06 PM hudson.remoting.Launcher$CuiListener status
INFO: Connected
```

**i-027cca5b0266a1b11 (AML)**

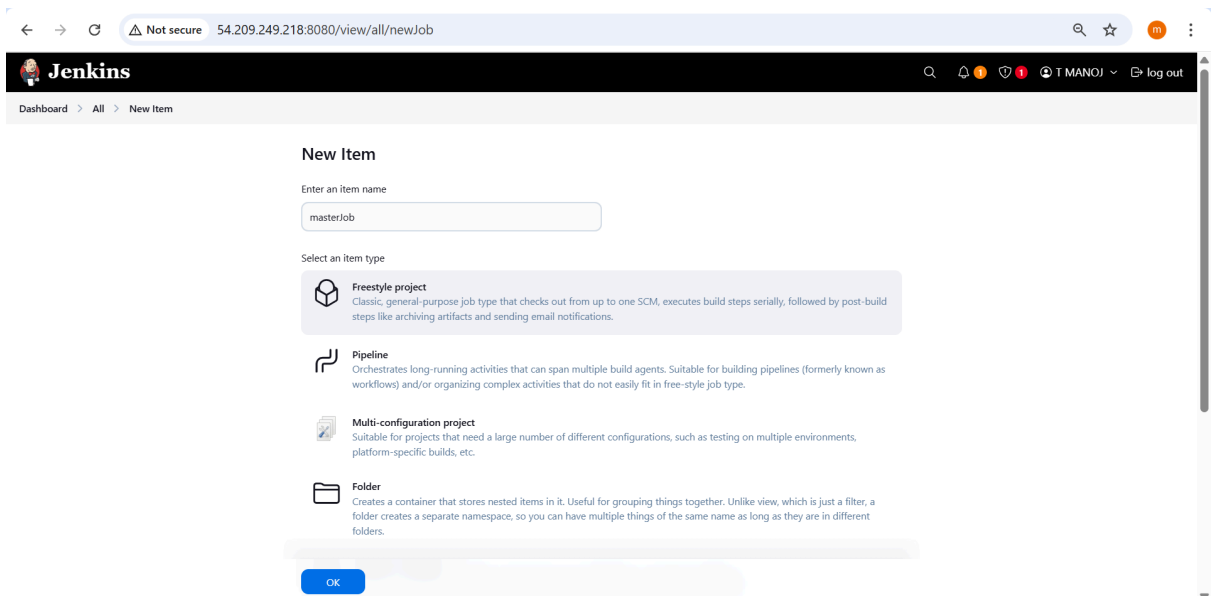
PublicIPs: 3.14.250.77 PrivateIPs: 172.31.6.25

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

## Step 4: Test the Master-Slave Setup

### 1. Create a test job:

- Click "New Item" on the Jenkins dashboard
- Enter a name for the job (e.g., "Test-Agent-Job")
- Select "Freestyle project" and click "OK"

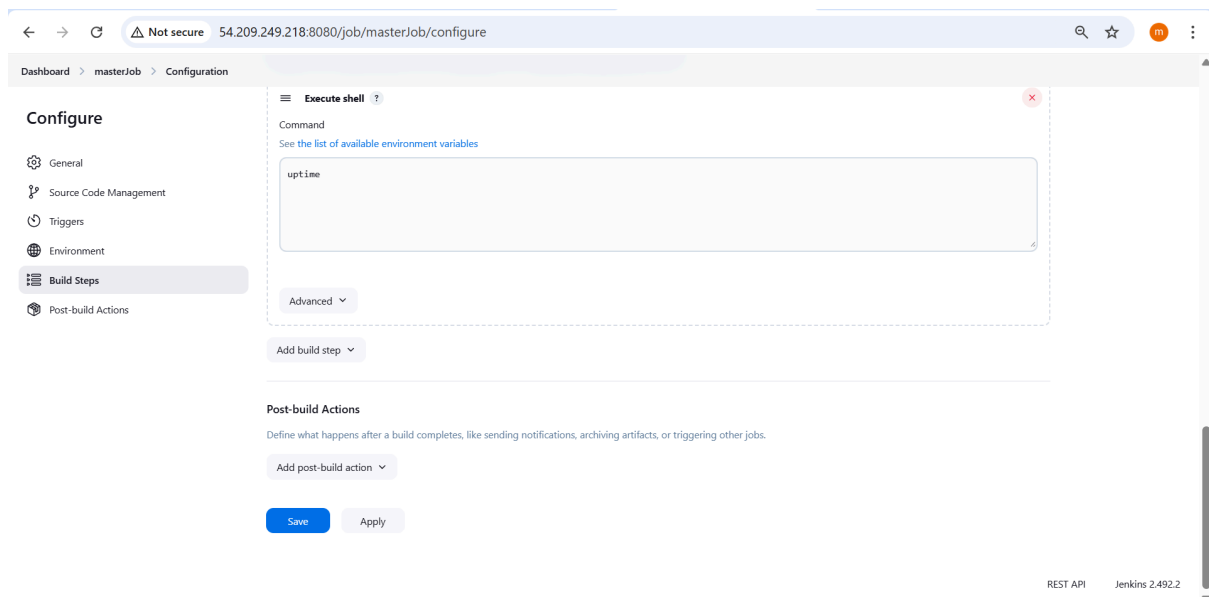
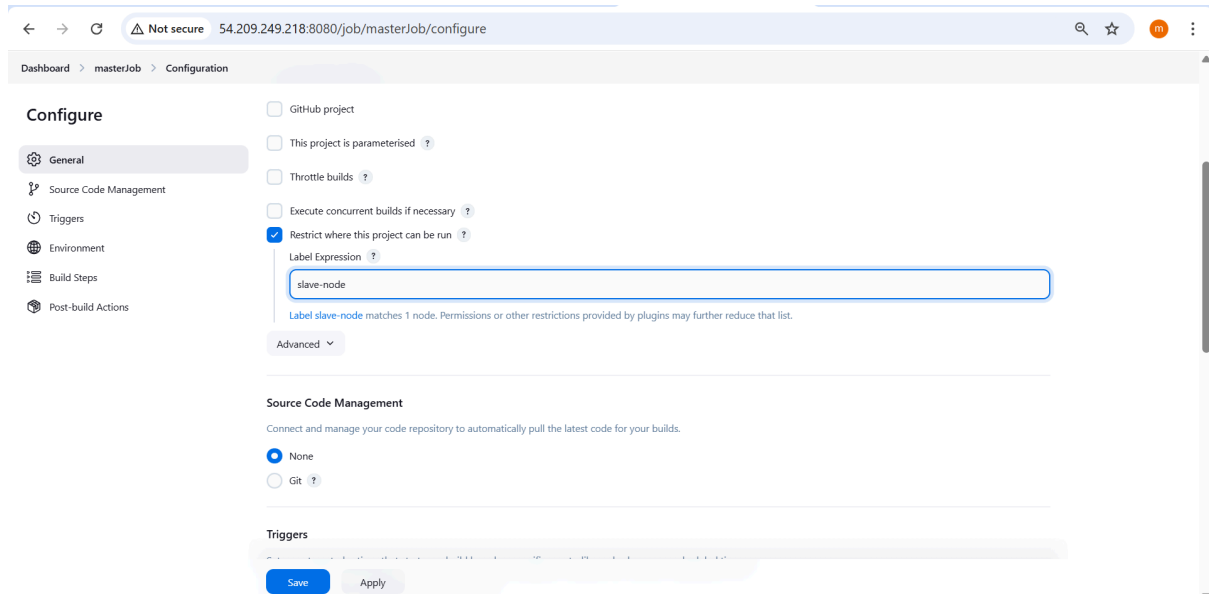


### 2. Configure the job:

- Under "General", check "Restrict where this project can be run"
- Enter the agent label you created earlier in the "Label Expression" field
- Scroll down to "Build Steps" and click "Add build step"
- Select "Execute shell"

Add these commands to verify the agent is working:  
uptimeecho \$WORKSPACE

- 
- Click "Save" and "Apply"



### 3. Run the job:

- Click "Build Now"
- Once complete, check the console output
- Verify that the uptime shown matches your agent's uptime and the workspace path contains `/home/ec2-user`

```
[ec2-user@ip-172-31-6-25 ~]$ cd jenkins/
[ec2-user@ip-172-31-6-25 jenkins]$ cd jenkins/
-bash: cd: jenkins/: No such file or directory
[ec2-user@ip-172-31-6-25 jenkins]$ ll
total 0
drwxr-xr-x. 4 ec2-user ec2-user 34 Apr  2 15:56 remoting
[ec2-user@ip-172-31-6-25 jenkins]$ cd remoting/
[ec2-user@ip-172-31-6-25 remoting]$ ll
total 0
drwxr-xr-x. 11 ec2-user ec2-user 96 Apr  2 15:56 jarCache
drwxr-xr-x.  2 ec2-user ec2-user 54 Apr  2 15:56 logs
[ec2-user@ip-172-31-6-25 remoting]$
```

 CloudShell

[Feedback](#)

© 2025, Amazon Web Services, Inc. or its affiliates.

[Privacy](#)

[Terms](#)

[Cookie preferences](#)