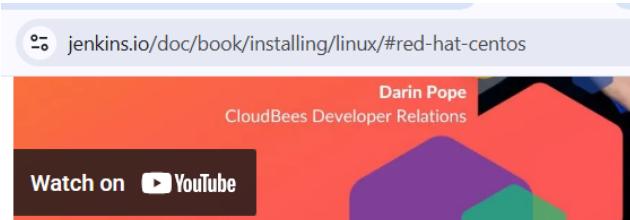


JENKINS 01

1 . Install Jenkins and run Jenkins on port number 8081.

- Open aws console
- Launch an ec2 instance
- Install Jenkins
- > browser search for Jenkins install for linux environment



You need to choose either the Jenkins Long Term Support release or Jenkins weekly release.

Long Term Support release

A [LTS \(Long-Term Support\) release](#) is chosen every 12 weeks from a stream of regular releases as the stable release for that time period. It can be installed from the [redhat-stable](#) yum repository.

```
sudo wget -O /etc/yum.repos.d/jenkins.repo \
           https://pkg.jenkins.io/redhat-stable/jenkins.repo
sudo rpm --import https://pkg.jenkins.io/redhat-stable/
sudo yum upgrade
# Add required dependencies for the jenkins package
sudo yum install fontconfig java-21-openjdk
sudo yum install jenkins
sudo systemctl daemon-reload
```

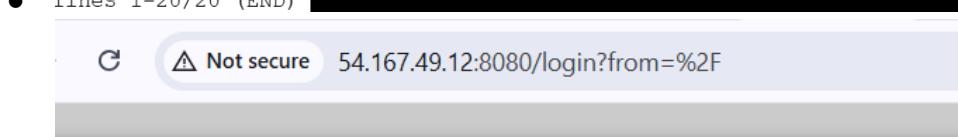
- Copy the command and paste in ec2 instance Jenkins will be installed

```
Version 2023.9.20251105:  
Run the following command to upgrade to 2023.9.20251105:  
  
dnf upgrade --releasever=2023.9.20251105  
  
Release notes:  
https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes-2023.  
.20251105.html  
  
Version 2023.9.20251110:  
Run the following command to upgrade to 2023.9.20251110:  
  
dnf upgrade --releasever=2023.9.20251110  
  
Release notes:  
https://docs.aws.amazon.com/linux/al2023/release-notes/relnotes-2023.  
.20251110.html  
  
=====  
  
Installed:  
jenkins-2.528.1-1.1.noarch  
  
Complete!
```

- sudo wget -O /etc/yum.repos.d/jenkins.repo \
- <https://pkg.jenkins.io/redhat-stable/jenkins.repo>
-
- sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
- sudo yum upgrade
- # Add required dependencies for the jenkins package
- Install java
- sudo dnf install java-17-amazon-corretto -y

```
Complete!  
[root@ip-10-0-1-110 ec2-user]# java -version  
openjdk version "17.0.17" 2025-10-21 LTS  
OpenJDK Runtime Environment Corretto-17.0.17.10.1 (build 17.0.17+10-LTS)  
OpenJDK 64-Bit Server VM Corretto-17.0.17.10.1 (build 17.0.17+10-LTS, mixed mode, sharing)  
● [root@ip-10-0-1-110 ec2-user]# █  
● sudo yum install jenkins  
● systemctl start Jenkins
```

```
[root@ip-10-0-1-110 ec2-user]# systemctl status jenkins.service
● jenkins.service - Jenkins Continuous Integration Server
  Loaded: loaded (/usr/lib/systemd/system/jenkins.service; disabled; pri-
    Active: active (running) since Tue 2025-11-11 09:19:49 UTC; 4s ago
      Main PID: 3555 (java)
        Tasks: 40 (limit: 1053)
       Memory: 359.1M
          CPU: 20.439s
        CGroup: /system.slice/jenkins.service
                └─3555 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/jen
Nov 11 09:19:45 ip-10-0-1-110.ec2.internal jenkins[3555]: [LF]> This may a
Nov 11 09:19:45 ip-10-0-1-110.ec2.internal jenkins[3555]: [LF]>
Nov 11 09:19:45 ip-10-0-1-110.ec2.internal jenkins[3555]: [LF]> ****
Nov 11 09:19:45 ip-10-0-1-110.ec2.internal jenkins[3555]: [LF]> ****
Nov 11 09:19:45 ip-10-0-1-110.ec2.internal jenkins[3555]: [LF]> ****
Nov 11 09:19:49 ip-10-0-1-110.ec2.internal jenkins[3555]: 2025-11-11 09:19:49
Nov 11 09:19:49 ip-10-0-1-110.ec2.internal jenkins[3555]: 2025-11-11 09:19:49
Nov 11 09:19:49 ip-10-0-1-110.ec2.internal systemd[1]: Started jenkins.ser
Nov 11 09:19:50 ip-10-0-1-110.ec2.internal jenkins[3555]: 2025-11-11 09:19:50
Nov 11 09:19:50 ip-10-0-1-110.ec2.internal jenkins[3555]: 2025-11-11 09:19:50
lines 1-20/20 (END)
```



Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, the password has been written to the log ([not sure where to find it](#)) and this file on the server:

`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

- Now, navigate to default location of Jenkins = `/var/lib/Jenkins`
- Find secrets
- Go to secrets location and list them
- And cat the initial password

```
[root@ip-10-0-1-110 ec2-user]# ^C
[root@ip-10-0-1-110 ec2-user]# cd /var/lib/jenkins
[root@ip-10-0-1-110 jenkins]# ls
config.xml           nodeMonitors.xml      secrets
hudson.model.UpdateCenter.xml   plugins       updates
jenkins.telemetry.Correlator.xml secret.key    userContent
jobs                 secret.key.not-so-secret users
[root@ip-10-0-1-110 jenkins]# cat secret
cat: secret: No such file or directory
[root@ip-10-0-1-110 jenkins]# cd secrets/
[root@ip-10-0-1-110 secrets]# ls
hudson.model.User.DIRNAMES  jenkins.model.Jenkins.crumbSalt
initialAdminPassword        master.key
[root@ip-10-0-1-110 secrets]# cat initialAdminPassword
a3c4d613227b4db08466902a3aebacca
[root@ip-10-0-1-110 secrets]#
```

- Now the copy the password and paste it in Jenkins page ,
- Now to change port number
- Navigate to this location.
- sudo vi /usr/lib/systemd/system/jenkins.service
- search for environment and change port number.
-

```
# IP address to listen on for HTTP requests.
# The default is to listen on all interfaces (0.0.0.0).
#Environment="JENKINS_LISTEN_ADDRESS="

# Port to listen on for HTTP requests. Set to -1 to disable.
# To be able to listen on privileged ports (port numbers less than 1024),
# add the CAP_NET_BIND_SERVICE capability to the AmbientCapabilities
# directive below.
Environment="JENKINS_PORT=8081"

# IP address to listen on for HTTPS requests. Default is disabled.
#Environment="JENKINS_HTTPS_LISTEN_ADDRESS="

# Port to listen on for HTTPS requests. Default is disabled.
# To be able to listen on privileged ports (port numbers less than 1024),
# add the CAP_NET_BIND_SERVICE capability to the AmbientCapabilities
# directive below.
Environment="JENKINS_HTTPS_PORT=443"

# Path to the keystore in JKS format (as created by the JDK's keytool).
# Default is disabled.
#Environment="JENKINS_HTTPS_KEYSTORE=/path/to/keystore.jks"
```

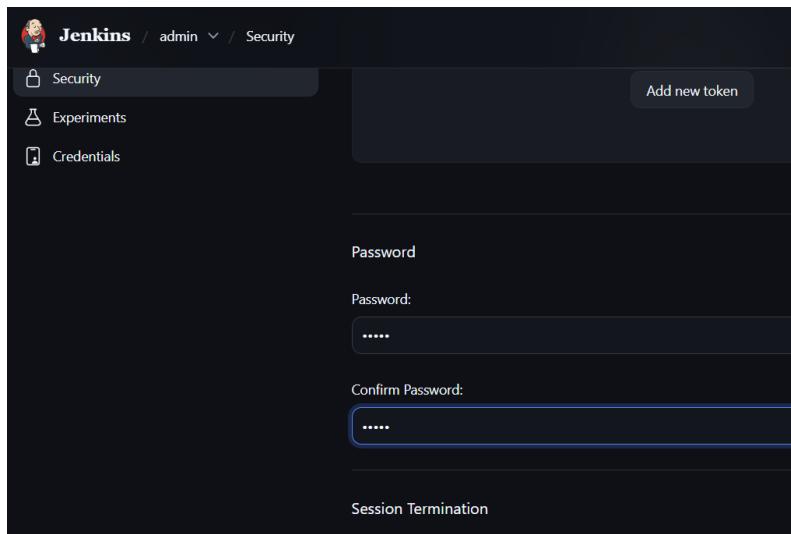
- Now restart Jenkins
- Systemctl daemon-reload
- Now,
- Systemctl restart Jenkins
- Now., Jenkins running on port number 8081
- Allow inbound rules form 8081 port in ec2

2 . Secure Jenkins server

-

Copy password from

```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```



- Add new password and user name and save it

3 . Create users called DevOps, Testing in Jenkins with Limited access.

- Login to Jenkins
- Navigate to manage Jenkins

A screenshot of the Jenkins 'Manage Jenkins' page at 'http://3.81.227.88:8081/manage/'. The top navigation bar shows 'Not secure' and the URL. Below it, there's a 'System Configuration' section with links for 'System', 'Nodes', and 'Clouds'. Under 'System', there's a link to 'Security'. The main area shows the 'Security' section with links for 'Security', 'Credentials', and 'Users'. The 'Security' link is highlighted with a blue border.

- Select user for creating user
- Add user name , password and member name

Users 3**+ Create User**These users can log into Jenkins. This is a sub set of [this list](#), which also contains auto-created users who really just made some commits on some projects and have no direct Jenkins access.

User ID ↓	Name	
 admin	admin	 
 Devops	teammate	 
 testing	teammate	 

- User created.
- Now giving limited acces to devops and tester
- Navigate to manage Jenkins > and the n security and find for authorization and select > project-based matrix authorization strategy

 Disable "Keep me signed in" ?

Security Realm

Jenkins' own user database

 Allow users to sign up ?

Authorization

Logged-in users can do anything

Anyone can do anything

Legacy mode

Logged-in users can do anything

Matrix-based security

Project-based Matrix Authorization Strategy

User/group	Overall	Credentials	Agent	Job	Run	Workspace
Anonymous	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authenticated Users	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
teammate	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
teammate	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Add user... Add group... ?

- Gave read acces to devops and testing users

4 . Configure labels and restrict the jobs to execute based on label only.

- Navigate to manage Jenkins
- Open nodes
- Add label name as devops
- Now downloads pulgins for node js
- And go free style project and restrict where this project can be run

/ first-job / Configuration

Plain text Preview

Enable project-based security

Discard old builds ?

GitHub project

This project is parameterized ?

Throttle builds ?

Execute concurrent builds if necessary ?

Restrict where this project can be run ?

Label Expression ?

devops

Label devops matches 1 node. Permissions or other restrictions provided

Advanced ▾

- \

The screenshot shows the Jenkins 'Nodes' page. At the top, there's a navigation bar with the Jenkins logo, 'Manage Jenkins', and 'Nodes'. Below the header, the word 'Nodes' is centered above a table. A blue button on the right says '+ New Node'. The table has columns: S, Name ↓, Architecture, Clock Difference, Free Disk Space, Free Swap Space, and Free Temp Space. One row is visible for 'Built-In Node', which is a 'Linux (amd64)' node. It shows 'In sync', '5.39 GiB', '0 B' (with a red exclamation mark), and '447.91 MiB'. The 'Data obtained' column shows '27 min' for both the node and the last check.

S	Name ↓	Architecture	Clock Difference	Free Disk Space	Free Swap Space	Free Temp Space
	Built-In Node	Linux (amd64)	In sync	5.39 GiB	0 B	447.91 MiB
		Data obtained	27 min	27 min	27 min	27 min

5 . Create Three sample jobs using the below

URL. https://github.com/betawins/Techie_horizon_Login_app.git

- Add git in to Jenkins
- Install git in terminal
- Now, give git path in tools
- /user/bin/git
- Copy the git repo
- Add to source code management

Source Code Management

Connect and manage your code repository to automatically pull the latest code for you.

None

Git [?](#)

Repositories [?](#)

Repository URL [?](#)

`https://github.com/betawins/Techie_horizon_Login_app.git`

- Schedule a build
- Start building now

Builds ...

Filter /

Today

- #3 12:23 PM ▾
- #2 12:20 PM ▾

● #3 (Nov 12, 2025, 12:23:51 PM)

Started by user [admin](#)

This run spent:

- 3 ms waiting;
- 0.15 sec build duration;
- 0.15 sec total from scheduled to completion.

Revision: ef9c453d4993843ac49d37321316c2588f154495
Repository: https://github.com/betawins/Techie_horizon_Login_app.git

- refs/remotes/origin/master

No changes.

-
- Create 3 jobs
- Navigate to Jenkins
- Create a new item
- Add name
- Select the - restrict where this project can be run

The screenshot shows the Jenkins job configuration page for a job named 'devops'. Under the 'Build Triggers' section, the 'Restrict where this project can be run' checkbox is checked, and the 'Label Expression' field contains 'devops'. Below this, a note states 'Label devops matches 1 node. Permissions or other restrictions provided by the node configuration will apply.' An 'Advanced' dropdown menu is open, listing the following items:

-
- Add git repo
- Add branch
- Add time stamps, terminate if it stuck

At the bottom of the configuration page, there are tabs for 'All' and '+', and a button labeled 'Add description'.

S	W	Name ↓	Last Success	Last Failure	Last Duration
		build	3 min 46 sec #1	N/A	0.23 sec
		deploy	5.6 sec #1	N/A	16 ms
		first-job	10 min #3	21 min #1	0.15 sec
		test	1 min 11 sec #1	N/A	0.19 sec

6 . Create one Jenkins job using git hub Private repository.

- Connect Jenkins and github
- By generating ssh keygen
- Switch to Jenkins user
- sudo su – Jenkins
- now. Generate key
- ssh-keygen -t rsa -b 4096 -C "jenkins"
- save the key
- to view the key

- cat ~/.ssh/id_rsa.pub

```
Enter same passphrase again.
Your identification has been saved in /home/ec2-user/.ssh/id_rsa
Your public key has been saved in /home/ec2-user/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:70eVLBcV1QT0uEKQI3dYZDaliR228x+YW8LBW3VnBMc jenkins
The key's randomart image is:
+---[RSA 4096]---+
|       .=Oo*B@|
|       .=Bo* *E|
|       o.oBoo..|
|       . ..+*. |
|       S   o*oo |
|       .   . ++..|
|       . o =. .|
|       o +     |
|       .         |
+---[SHA256]---+
[ec2-user@ip-10-0-1-110 ~]$ cat ~/.ssh/id_rsa.pub
```

- now copy key and paste in github deploy
- now Jenkins ,
- go to manage Jenkins
- credentials add git repo
- and add secret key to it

Credentials

T	P	Store ↓	Domain	ID	Name
		System	(global)	58670c12-0222-4f46-87d9-fda93cf8640f	git

- now create a job add git repo
- add ssh git in the section
- apply and save it
- start building

Console Output

```
Started by user admin
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/jenkins
The recommended git tool is: NONE
Warning: CredentialId "github" could not be found.
> /usr/bin/git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/jenkins/.git # timeout=10
Fetching changes from the remote Git repository
> /usr/bin/git config remote.origin.url git@github.com:kamal-kkk/local-to-github.git # timeout=10
Fetching upstream changes from git@github.com:kamal-kkk/local-to-github.git
> /usr/bin/git --version # timeout=10
> git --version # 'git version 2.50.1'
> /usr/bin/git fetch --tags --force --progress -- git@github.com:kamal-kkk/local-to-github.git
> /usr/bin/git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision b1aeaf26c71ef741cc806cd3f3441ba43cdd3d16 (refs/remotes/origin/master)
> /usr/bin/git config core.sparsecheckout # timeout=10
> /usr/bin/git checkout -f b1aeaf26c71ef741cc806cd3f3441ba43cdd3d16 # timeout=10
Commit message: "your update"
First time build. Skipping changelog.
Finished: SUCCESS
```