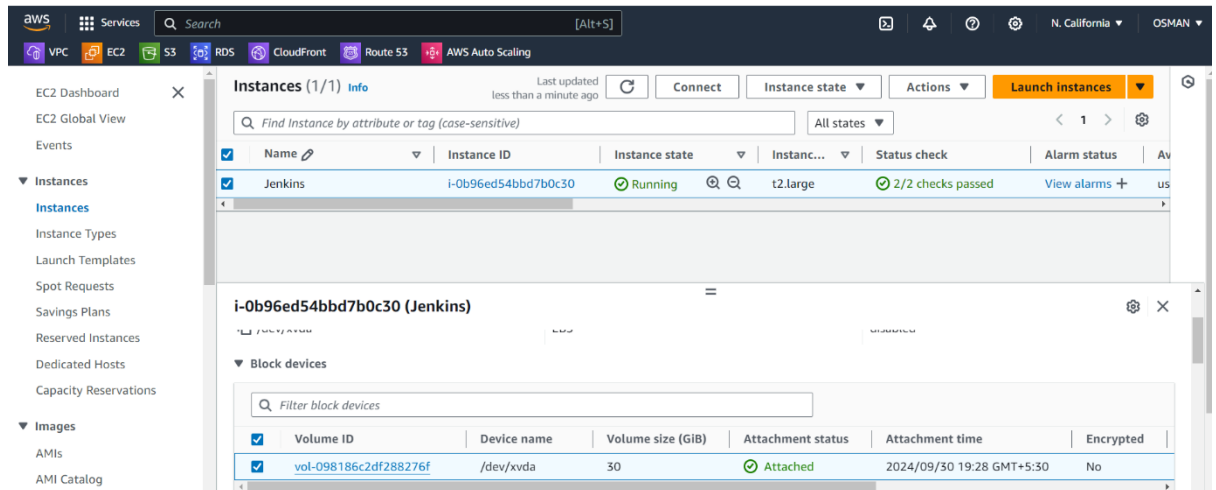

1) Install jenkins and run jenkins on port number 8081.

- To install Jenkins first we want Ec2 instance.
- So first you can launch EC2 instance with t2 medium (or) t2 large with 30 GB.
- Here I am launched one server with t2 large with 30 GB.



- Now I want to connect the server and install the Jenkins server.
- Here first I am installing the java because this is prerequisite.
- `yum -y install java-17*`

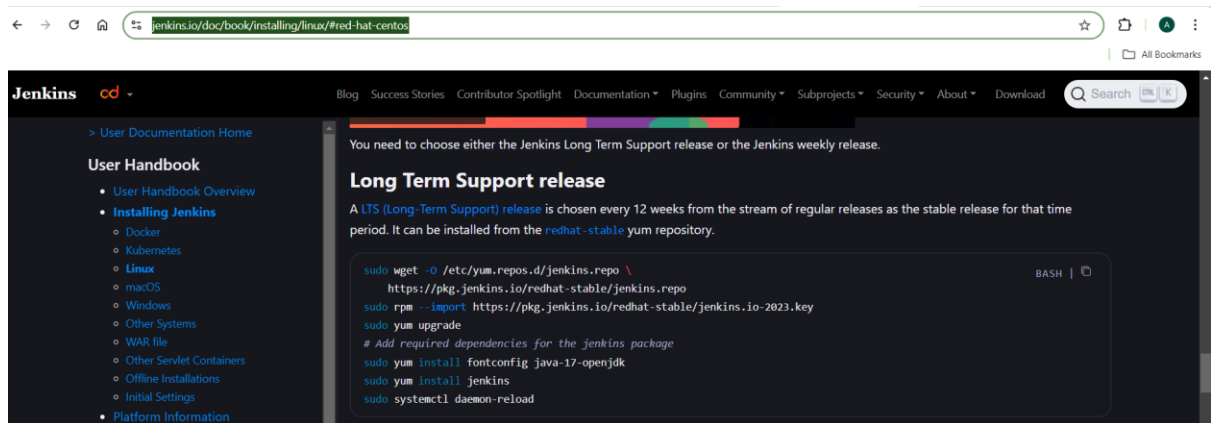
```
PS C:\Users\ramee\downloads> ssh -i "C:\Users\ramee.pem" ec2-user@ec2-13-57-182-232.us-west-1.compute.amazonaws.com
The authenticity of host 'ec2-13-57-182-232.us-west-1.compute.amazonaws.com (13.57.182.232)' can't be established.
ED25519 key fingerprint is SHA256:91m1ZB3y09ymkLLZ5wX6doZqYwI8DM6CLDbzqMq.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-13-57-182-232.us-west-1.compute.amazonaws.com' (ED25519) to the list of known hosts.

_
#
##### Amazon Linux 2
#####
AL2 End of Life is 2025-06-30.
#####
#
V-! ->
A newer version of Amazon Linux is available!
#####
Amazon Linux 2023, GA and supported until 2028-03-15.
https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-13-31 ~]$ sudo su -
[root@ip-172-31-13-31 ~]# sudo yum -y install java-17*
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core | 3.6 kB 00:00:00

Resolving Dependencies
-> Running transaction check
-> Package java-17-amazon-corretto-x86_64 1:17.0.12+7-1.amzn2.1 will be installed
-> Processing Dependency: libx11 for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
-> Processing Dependency: libxkb for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
-> Processing Dependency: libxinerama for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
-> Processing Dependency: libxt for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
-> Processing Dependency: libXrender for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
-> Processing Dependency: libXrandr for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
-> Processing Dependency: libXtst for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
-> Processing Dependency: glib1 for package: 1:java-17-amazon-corretto-17.0.12+7-1.amzn2.1.x86_64
-> Package java-17-amazon-corretto-devel-x86_64 1:17.0.12+7-1.amzn2.1 will be installed
-> Package java-17-amazon-corretto-headless-x86_64 1:17.0.12+7-1.amzn2.1 will be installed
-> Processing Dependency: jpackage-utils for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
-> Processing Dependency: fontconfig for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
-> Processing Dependency: dejavu-sans-fonts for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
-> Processing Dependency: dejavu-serif-fonts for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
-> Processing Dependency: dejavu-sans-mono-fonts for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
-> Processing Dependency: alsa-lib for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
-> Processing Dependency: log4j-cve-2021-44228-cve-mitigations for package: 1:java-17-amazon-corretto-headless-17.0.12+7-1.amzn2.1.x86_64
```

- Now I want install Jenkins.
- So click on the link – <https://www.jenkins.io/doc/book/installing/linux/#red-hat-centos>
- You see the below script and Run.



These all command don't Run at a time.

```
[root@ip-172-31-13-31 ~]# sudo wget -O /etc/yum.repos.d/jenkins.repo \
> https://pkg.jenkins.io/redhat-stable/jenkins.repo
--2024-09-30 14:15:19-- https://pkg.jenkins.io/redhat-stable/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io)... 151.101.42.133, 2a04:4e42:a::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)[151.101.42.133]: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

2024-09-30 14:15:19 (10.5 MB/s) - '/etc/yum.repos.d/jenkins.repo' saved [85/85]

[root@ip-172-31-13-31 ~]# sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
[root@ip-172-31-13-31 ~]# sudo yum upgrade
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core | 3.6 kB 00:00:00
jenkins | 2.9 kB 00:00:00
jenkins/primary_db | 51 kB 00:00:00
No packages marked for update
[root@ip-172-31-13-31 ~]# sudo yum install jenkins
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package jenkins.noarch 0:2.462.2-1.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Installing:
jenkins noarch 2.462.2-1.1 jenkins 89 M
=====

Transaction Summary
=====
Install 1 Package
Total download size: 89 M
Installed size: 89 M
```

CMD ---- sudo systemctl daemon-reload

---- The command sudo systemctl daemon-reload reloads **systemd**'s configuration files to apply changes made to service unit files without restarting the system.

CMD--- systemctl enable jenkins

The command systemctl enable jenkins configures Jenkins to start automatically at system boot by creating the necessary symlink in the system's boot process.

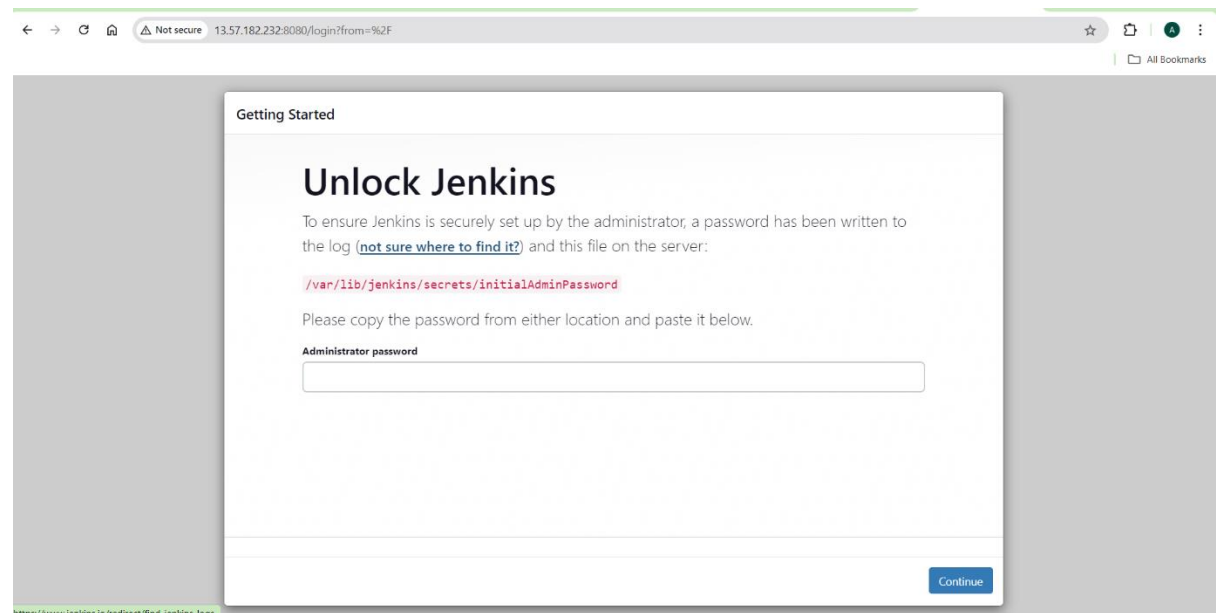
To change the port number I am not find the file where I want to change the port number at the point off time using these below command I found the path Here

CMD --- The command `sudo grep -R "8080" /etc` recursively searches for occurrences of the string "8080" in all files and subdirectories within the /etc directory, using superuser privileges.

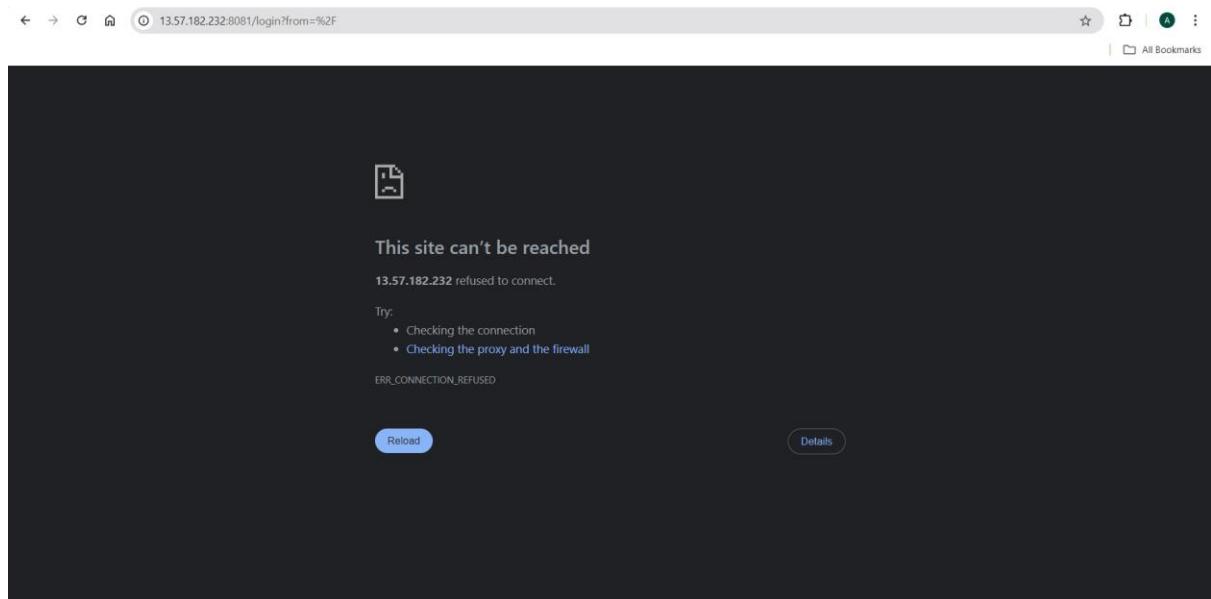
```
root@ip-172-31-13-31 ~]# sudo grep -R "8080" /etc/
Binary file /etc/pki/ca-trust/extracted/java/cacerts matches
Binary file /etc/pki/java/cacerts matches
/etc/systemd/system/multi-user.target.wants/jenkins.service:Environment="JENKINS_PORT=8080"
Binary file /etc/udev/hwdb.bin matches
/etc/services:webcache      8080/tcp      http-alt      # WWW caching service
/etc/services:webcache      8080/udp      http-alt      # WWW caching service
Binary file /etc/alternatives/java-17-amazon-corretto/lib/modules matches
Binary file /etc/alternatives/java-17-amazon-corretto/lib/security/cacerts matches
Binary file /etc/alternatives/java-17-amazon-corretto/lib/server/classes.jsa matches
Binary file /etc/alternatives/java-17-amazon-corretto/jmods/java.desktop.jmod matches
Binary file /etc/alternatives/jre/lib/modules matches
Binary file /etc/alternatives/jre/lib/security/cacerts matches
Binary file /etc/alternatives/jre/lib/server/classes.jsa matches
Binary file /etc/alternatives/jre/jmods/java.desktop.jmod matches
Binary file /etc/alternatives/jre_openjdk/lib/modules matches
Binary file /etc/alternatives/jre_openjdk/lib/security/cacerts matches
Binary file /etc/alternatives/jre_openjdk/lib/server/classes.jsa matches
Binary file /etc/alternatives/jre_openjdk/jmods/java.desktop.jmod matches
Binary file /etc/alternatives/jre_17/lib/modules matches
Binary file /etc/alternatives/jre_17/lib/security/cacerts matches
Binary file /etc/alternatives/jre_17/lib/server/classes.jsa matches
Binary file /etc/alternatives/jre_17/jmods/java.desktop.jmod matches
Binary file /etc/alternatives/jre_17_openjdk/lib/modules matches
Binary file /etc/alternatives/jre_17_openjdk/lib/security/cacerts matches
Binary file /etc/alternatives/jre_17_openjdk/lib/server/classes.jsa matches
Binary file /etc/alternatives/jre_17_openjdk/jmods/java.desktop.jmod matches
Binary file /etc/alternatives/java_sdk/lib/modules matches
Binary file /etc/alternatives/java_sdk/lib/security/cacerts matches
```

Here first I have to check the Jenkins.

Here I am open with port number 8080--- it will worked.



After I am using the port 8081—not working this port.



Now I am changing the port number 8081 in these below file it will be work.

After changing the port number you need to enter the below two commands.

```
[root@ip-172-31-13-31 ~]# sudo vi /etc/systemd/system/multi-user.target.wants/jenkins.service
149L, 5596B written
[root@ip-172-31-13-31 ~]# sudo systemctl daemon-reload
[root@ip-172-31-13-31 ~]# systemctl restart jenkins
[root@ip-172-31-13-31 ~]# The command systemctl enable jenkins configures Jenkins to start automatically at system boot by creating the necessary symlink in
the system's boot process.

# Directory where Jenkins stores its configuration and workspaces
Environment="JENKINS_HOME=/var/lib/jenkins"
WorkingDirectory=/var/lib/jenkins

# Location of the Jenkins WAR
#Environment="JENKINS_WAR=/usr/share/java/jenkins.war"

# Location of the exploded WAR
Environment="JENKINS_WEBROOT=%C/jenkins/war"

# Location of the Jenkins log. By default, systemd-journald(8) is used.
#Environment="JENKINS_LOG=%L/jenkins/jenkins.log"

# The Java home directory. When left empty, JENKINS_JAVA_CMD and PATH are consulted.
#Environment="JAVA_HOME=/usr/lib/jvm/java-17-openjdk-amd64"

# The Java executable. When left empty, JAVA_HOME and PATH are consulted.
#Environment="JENKINS_JAVA_CMD=/etc/alternatives/java"

# Arguments for the Jenkins JVM
Environment="JAVA_OPTS=-Djava.awt.headless=true"

# Unix Domain Socket to listen on for local HTTP requests. Default is disabled.
#Environment="JENKINS_UNIX_DOMAIN_PATH=/run/jenkins/jenkins.socket"

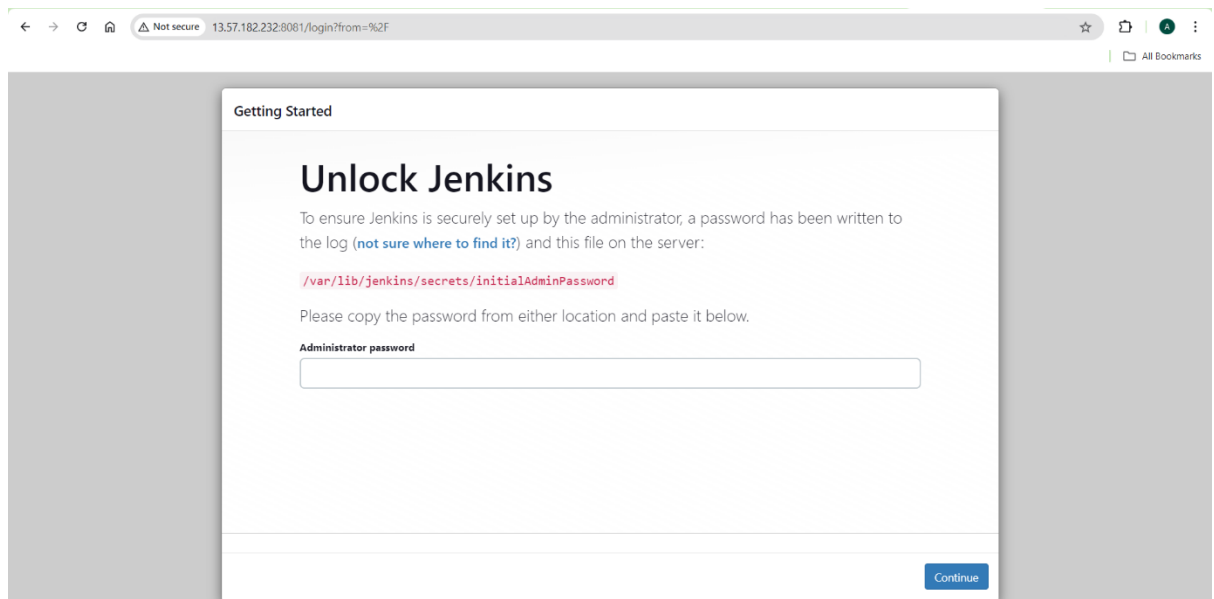
# IP address to listen on for HTTP requests.
# The default is to listen on all interfaces (0.0.0.0).
#Environment="JENKINS_LISTEN_ADDRESS="

# 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.
-- INSERT --
```

Now it's work's



#####Task is Done#####

2) Secure Jenkins server

There are methods to secure Jenkins servers.

1. Delete the initial password
2. To secure the Jenkins server we have to give User name and password.

Delete the initial password:

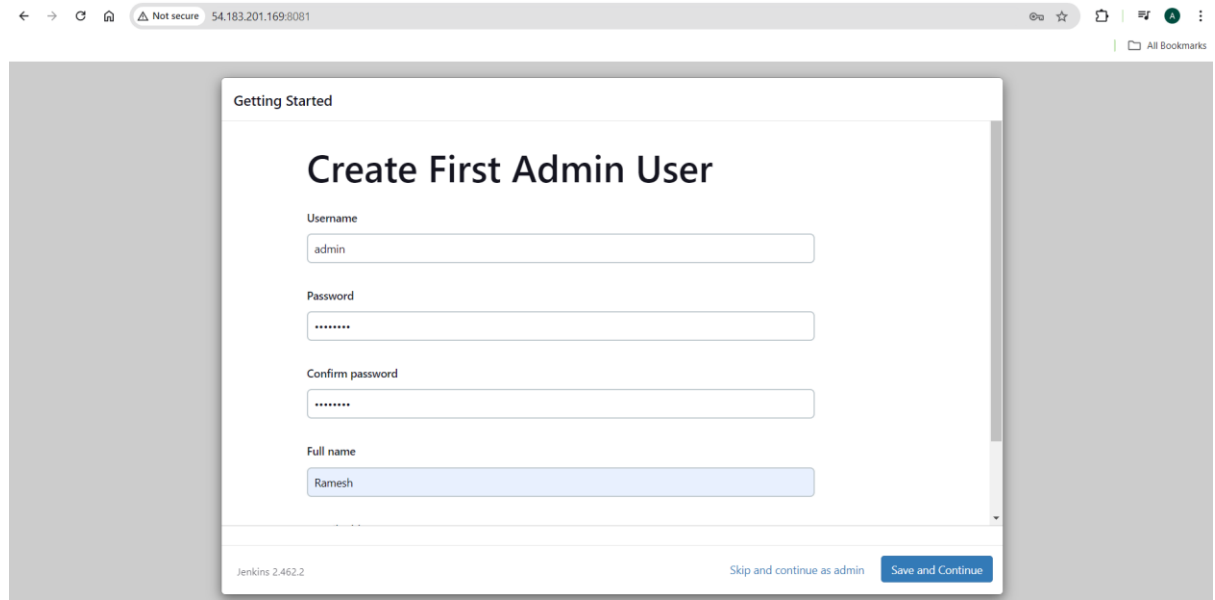
- While Login Jenkins you can give initial password.
- That password you need to delete to secure Jenkins server.
- Go to this location ---- `cd /var/lib/Jenkins/secrets/`
- Delete the password--- `rm -rf initialAdminpassword`

```
[root@ip-172-31-13-31 ~]# cat /var/lib/jenkins/secrets/initialAdminPassword
2a575c2b3d154bf68e2c9e86bf9607de
```

User name and password

While Login time you can give.

User Name and Password.



The screenshot shows a web browser window with the address bar displaying '54.183.201.169:8081'. The page title is 'Getting Started'. The main heading is 'Create First Admin User'. The form contains four input fields: 'Username' with the value 'admin', 'Password' with masked characters '*****', 'Confirm password' with masked characters '*****', and 'Full name' with the value 'Ramesh'. At the bottom of the form, there are two buttons: 'Skip and continue as admin' and 'Save and Continue'. The footer of the page indicates 'Jenkins 2.462.2'.

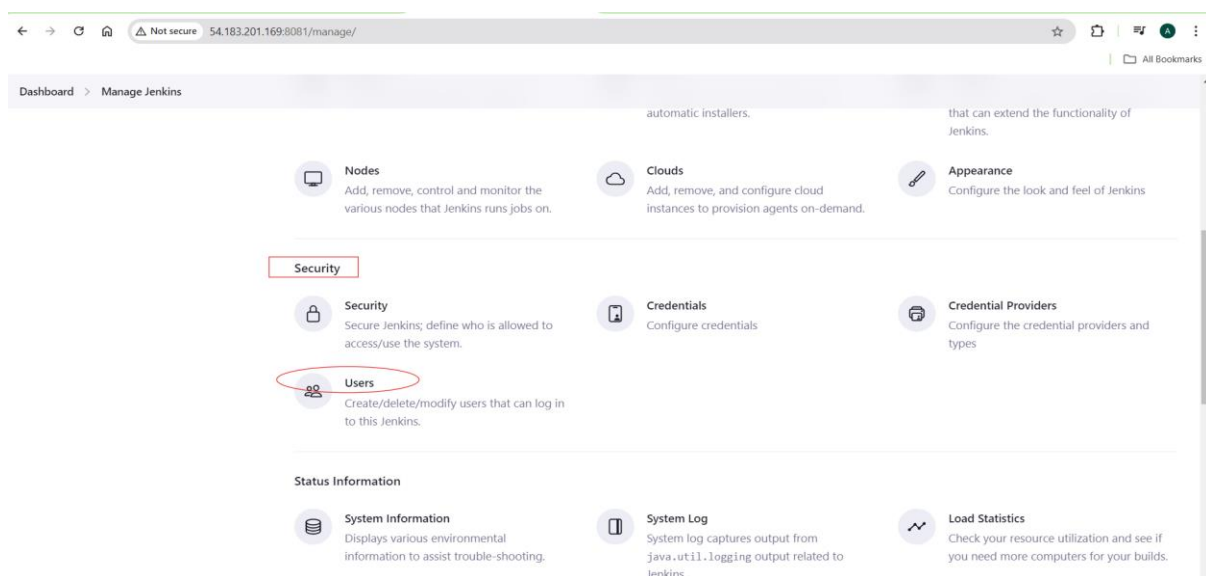
Using these two Methods your Jenkins Server is Secure.

#####Task Is Done #####

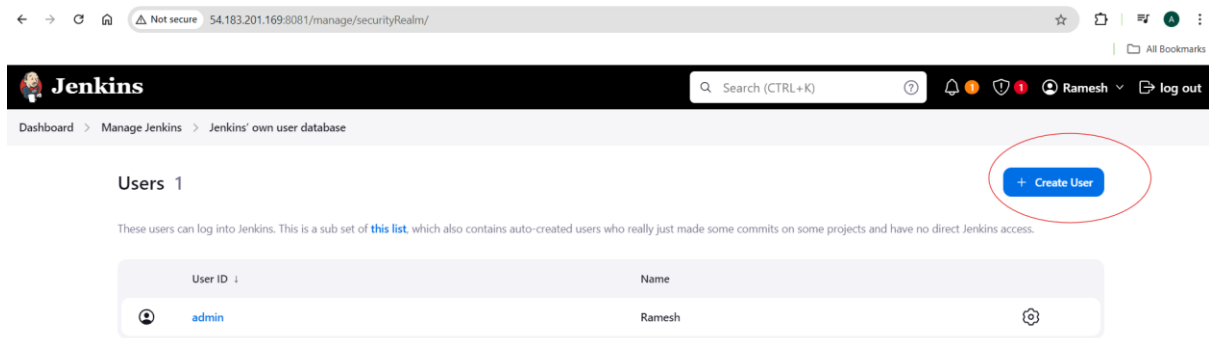
3) Create users called Devops, Testing in Jenkins with Limited access.

To create user follow the bellow steps.

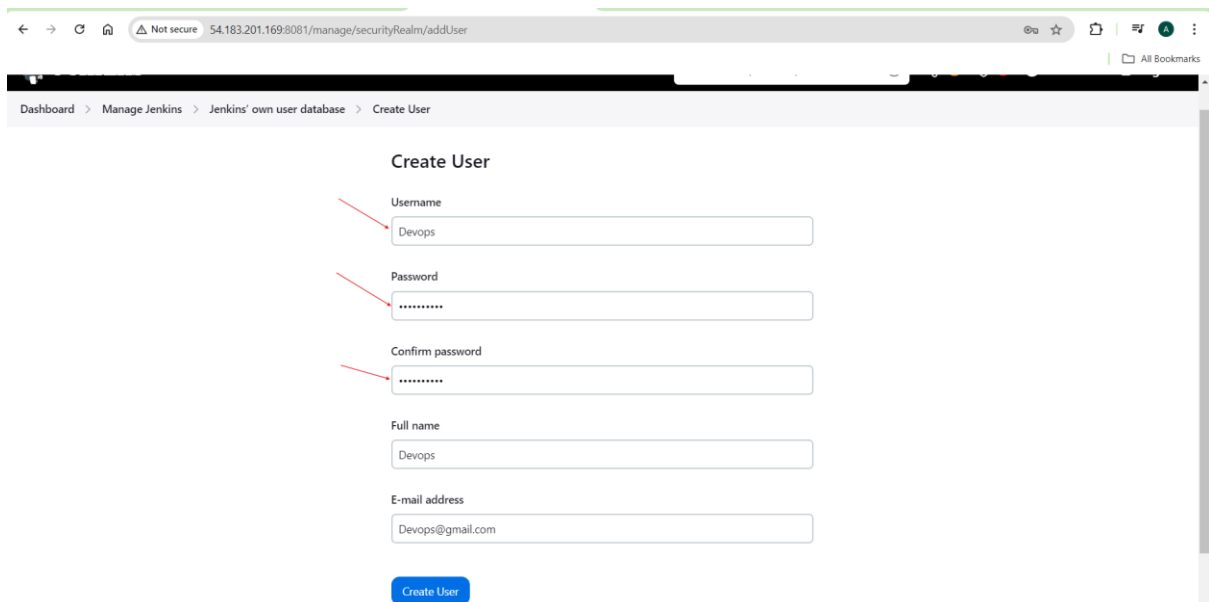
- Go to the manage Jenkins.
- There under the security you see User and click on users



Click on create User.



- Here give user Name, Password, Name, and Gmail.
- Now Just click on User.



Now I am assigning the limited accesess.

- Go Manage Jenkins, Security.
- There you see the option is Authorization.
- Authorization under you need to select Project-based matrix Authorization Strategy.
- And click on save and Login as Devops user.

The screenshot shows the Jenkins Security Configuration page. The breadcrumb trail is Dashboard > Manage Jenkins > Security. A toggle for 'Allow users to sign up' is present. The 'Authorization' section is set to 'Project-based Matrix Authorization Strategy'. Below this is a table with columns for 'User/group' and various permissions grouped under 'Overall', 'Credentials', 'Agent', 'Job', 'Run', 'View', 'SCM', and 'Metrics'. The 'Devops' user is listed with 'Read' permissions checked for 'Job' and 'View' categories. Buttons for 'Add user...' and 'Add group...' are at the bottom of the table.

User/group	Overall		Credentials				Agent				Job				Run		View		SCM	Metrics													
	Administer	Read	Create	Delete	ManageDomains	Update	View	Build	Configure	Connect	Create	Delete	Disconnect	Provision	Cancel	Configure	Create	Delete	Discover	Move	Read	Workspace	Delete	Replay	Update	Configure	Create	Delete	Read	Tag	HealthCheck	ThreadDump	View
Anonymous	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authenticated Users	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Devops	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Buttons: Add user..., Add group..., ?

Markup Formatter section: Plain text, Save, Apply

- Here I am just given only job read option and view read option only.
- I am not given Overall Read permission that's I am getting Error.

The screenshot shows the Jenkins dashboard. The breadcrumb trail is Dashboard >. The Jenkins logo and search bar are at the top. The user is logged in as 'Devops'. A red error message is displayed: 'Access Denied' followed by 'Devops is missing the Overall/Read permission'.

Access Denied

Devops is missing the Overall/Read permission

- Now I am change the permission over all read permission then we will see what happened.

The screenshot shows the Jenkins Security Configuration page. The 'Authorization' section is set to 'Project-based Matrix Authorization Strategy'. Below this is a table with columns for 'User/group' and various permissions. The 'User/group' column lists 'Anonymous', 'Authenticated Users', 'Ramesh', and 'Devops'. The permissions are grouped into categories: Overall, Credentials, Agent, Job, Run, View, SCM, and Metrics. The 'Devops' user has 'Read' permissions checked under the 'Overall' and 'View' categories. A red box highlights the 'Read' checkbox under the 'Overall' category for the 'Devops' user.

User/group	Overall	Credentials	Agent	Job	Run	View	SCM	Metrics
Anonymous	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Authenticated Users	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Ramesh	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Devops	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Buttons: Add user..., Add group..., Save, Apply

- Now we see the some option here.

The screenshot shows the Jenkins Dashboard. The 'Build Queue' section shows 'No builds in the queue.' The 'Build Executor Status' section shows two executors, both in 'idle' state. A red box highlights the 'Build Queue' and 'Build Executor Status' sections.

Build Queue

No builds in the queue.

Build Executor Status

1 idle

2 idle

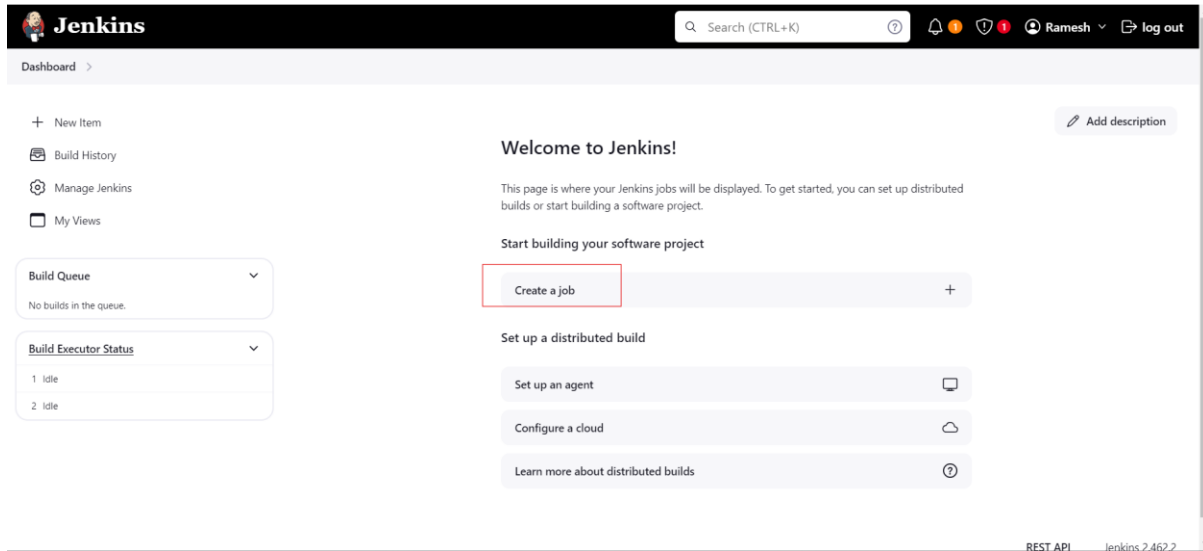
REST API Jenkins 2.462.2

- These way you can assign limit accesses any user.

#####Task is Done#####

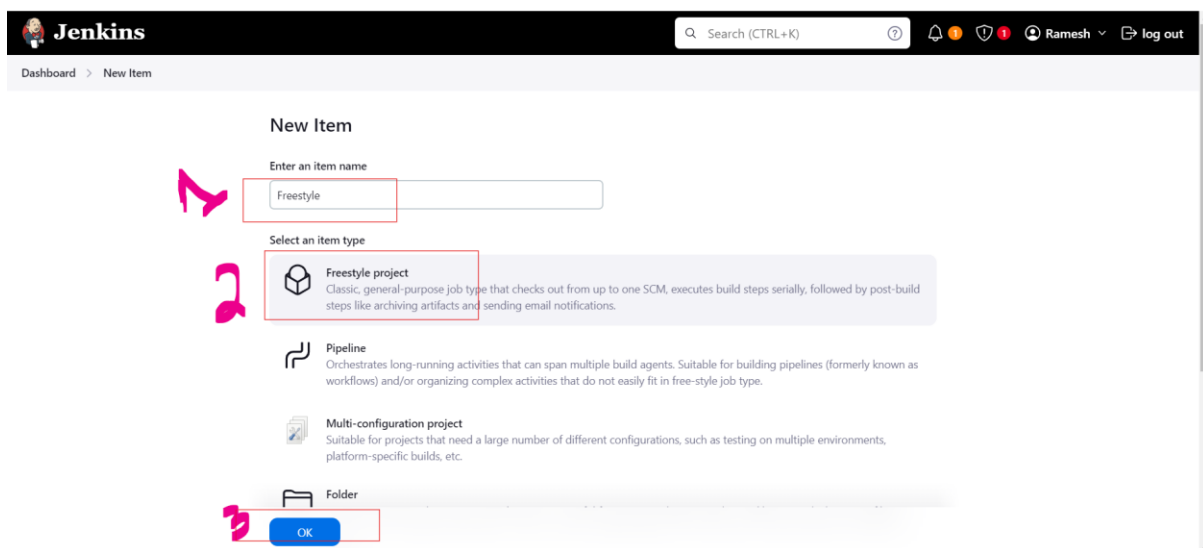
4) Configure labels and restrict the jobs to execute based on label only.

- To do that first I am creating the two Jobs through free style.
- To create job click on create Job.

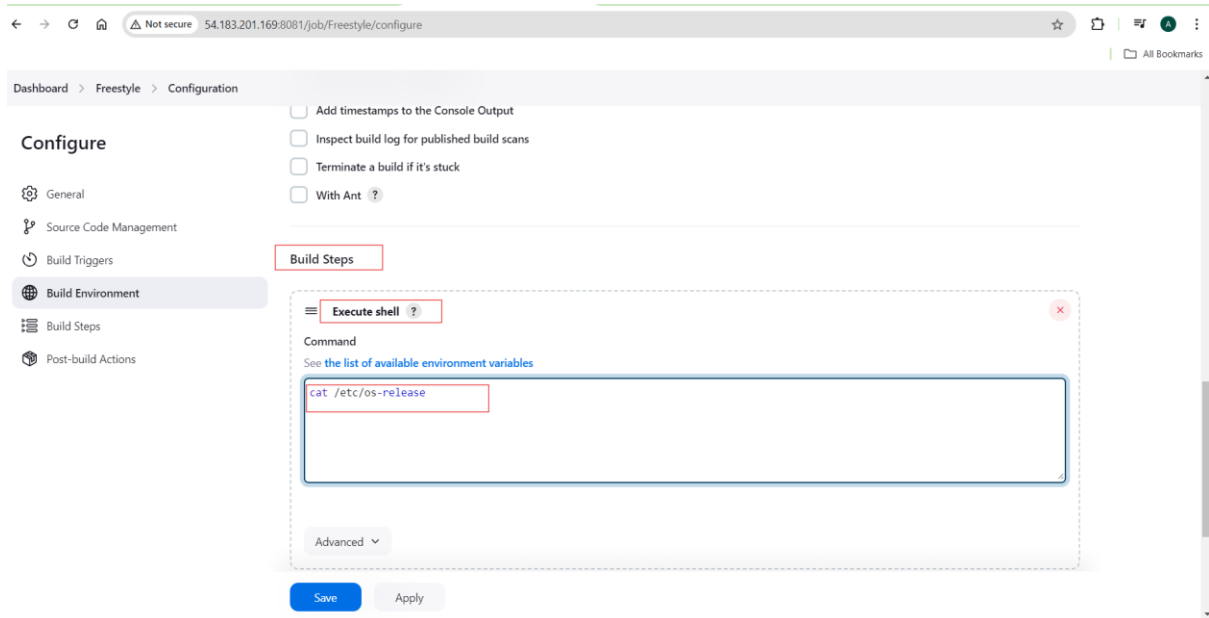


Give name --- what you want.

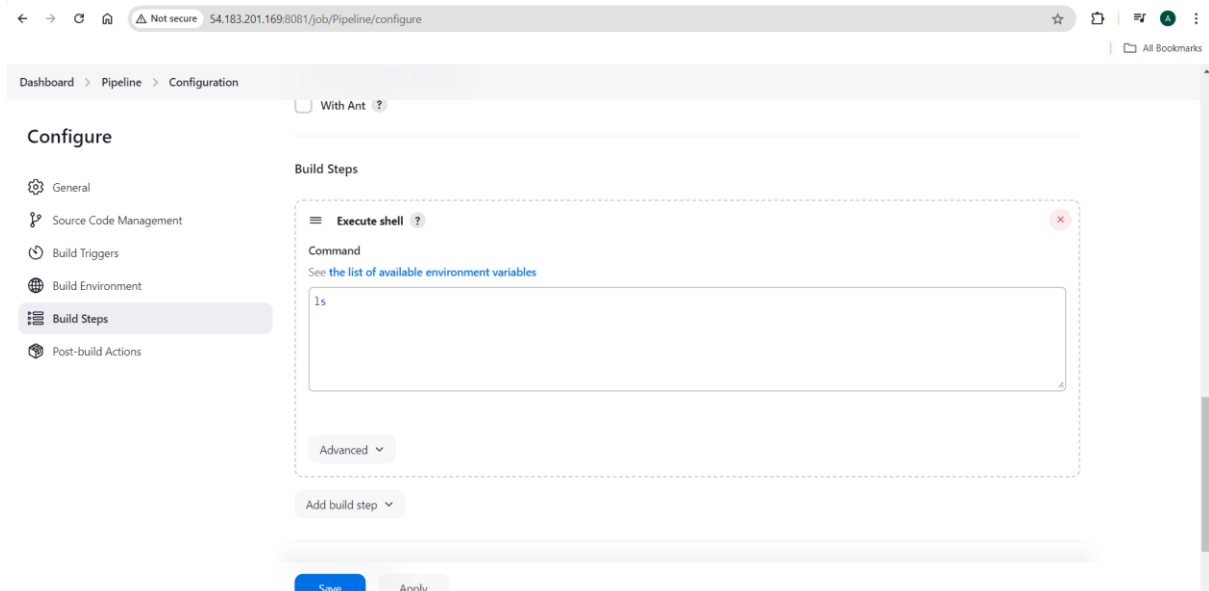
Select Freestyle Project and click on ok.



- Go to the Build Steps and select Execute shell.
- There write any Commands what you want.
- Here I am just given these command ----- `cat /etc/os-release` and click on save.



- Here I am create job name with Pipeline.
- in execute shell just `ls`.



- Click on save and build the job.

- Here I am try to build these jobs.

The screenshot shows the Jenkins Dashboard. On the left, there's a sidebar with links: New Item, Build History, Manage Jenkins, and My Views. The main area displays the 'Build Queue' and 'Build Executor Status'.

Build Queue: No builds in the queue.

Build Executor Status:

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	Freestyle	37 sec #17	N/A	10 sec
✓	☀	Pipeline	35 sec #23	11 min #2	10 sec

Below the table, there's a 'Build Executor Status' section showing two executors: 1 Pipeline (26%) and 2 Freestyle (20%).

- These time these jobs are executed.
- To create Label go to the manage Jenkins and System.
- There you find the labels—there give label Name and Select only build job with label.

The screenshot shows the 'Manage Jenkins > System' configuration page. The 'Usage' dropdown is set to 'Only build jobs with label expressions matching this node'. Other settings include 2 executors, Pipeline label, 5 quiet period, 0 SCM checkout retry count, and 'Restrict project naming' is unchecked.

At the bottom, there are 'Save' and 'Apply' buttons.

After that click on save.

- Now try to build these two jobs.
- These jobs are in build queue but not executed.

The screenshot shows the Jenkins Dashboard. On the left, there's a sidebar with links: New Item, Build History, Manage Jenkins, and My Views. Below this, there are two panels: 'Build Queue (2)' showing 'Freestyle' and 'Pipeline' jobs, and 'Build Executor Status' showing two idle executors. The main area displays a table of jobs in the queue.

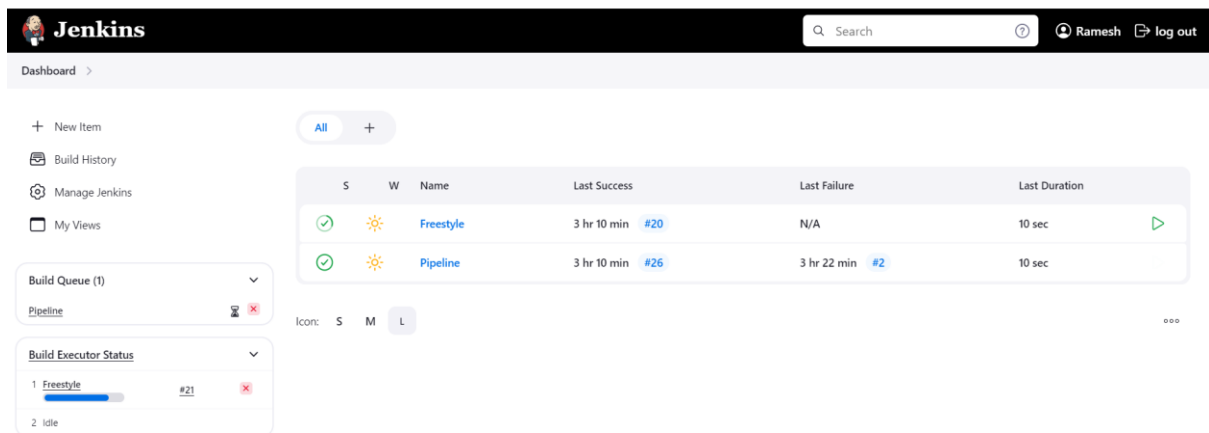
S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	Freestyle	2 hr 46 min #20	N/A	10 sec
✓	☀	Pipeline	2 hr 46 min #26	2 hr 58 min #2	10 sec

Below the table, there are filters for 'Icon: S M L'.

- Now I went to freestyle job.
- Select option is ---- Restrict where this Project can be run?
- Label expression --- Pipeline.

The screenshot shows the 'Configure' page for a Freestyle job. The left sidebar has links: General, Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The 'General' tab is selected. In the 'Restrict where this project can be run' section, the checkbox is checked, and the 'Label Expression' is set to 'Pipeline'. Below this, there's a section for 'Source Code Management' with 'None' selected. At the bottom, there are 'Save' and 'Apply' buttons.

Now the freestyle job is executed.



#####Task is Done #####

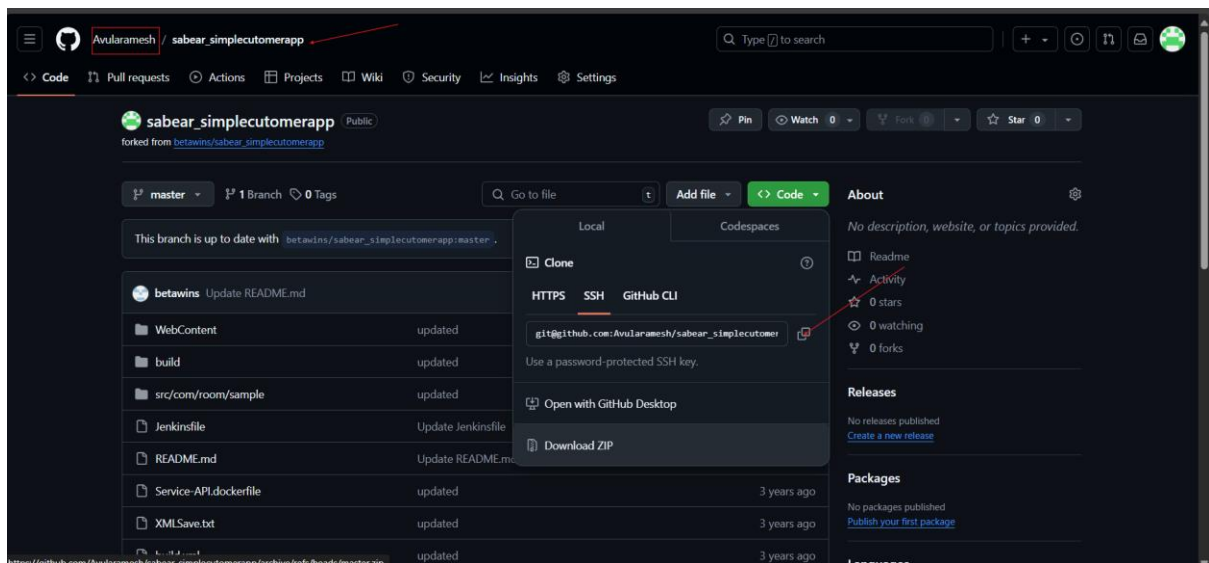
5) Create Three sample jobs using the below URL.

https://github.com/betawins/sabear_simplecutomerapp.git

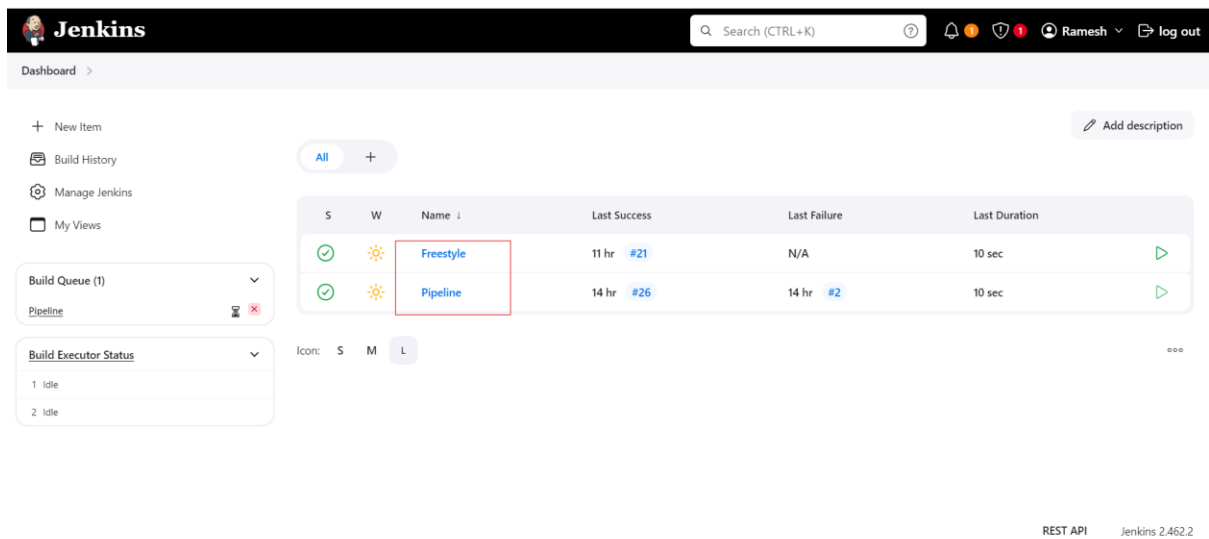
To Create Three sample jobs.

Now first I am fork to my GitHub account.

Here I am copy the my code.



Here I have already two Jobs.



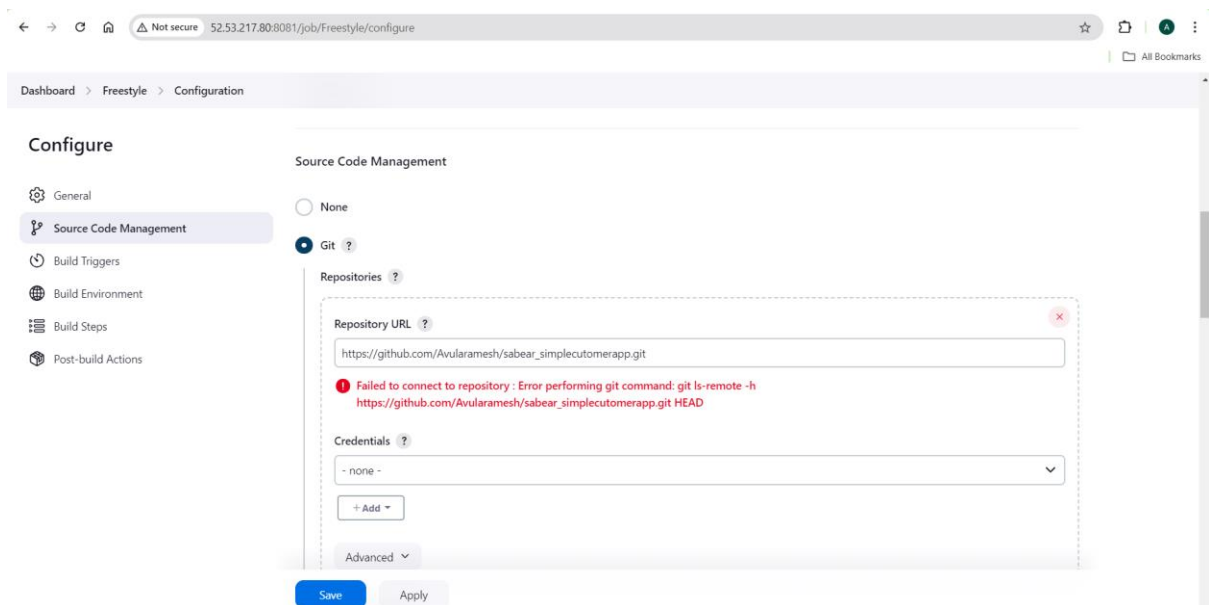
The screenshot shows the Jenkins Dashboard. At the top, there's a search bar and user information for 'Ramesh'. The main area displays a table of jobs. Two jobs are listed: 'Freestyle' and 'Pipeline'. Both have a status of 'Success' (green checkmark) and a duration of '10 sec'. The 'Freestyle' job has a last success time of '11 hr #21' and no last failure. The 'Pipeline' job has a last success time of '14 hr #26' and a last failure time of '14 hr #2'. On the left sidebar, there are links for 'New Item', 'Build History', 'Manage Jenkins', and 'My Views'. Below these are sections for 'Build Queue (1)' and 'Build Executor Status'.

S	W	Name	Last Success	Last Failure	Last Duration
✓	☀	Freestyle	11 hr #21	N/A	10 sec
✓	☀	Pipeline	14 hr #26	14 hr #2	10 sec

Just went to these two jobs.

In repositories URL --- Give github repo link---

https://github.com/Avularamesh/sabear_simplecutomerapp.git



The screenshot shows the Jenkins Configuration page for the 'Freestyle' job. The 'Source Code Management' section is selected. Under 'Source Code Management', the 'Git' option is chosen. The 'Repository URL' field contains the GitHub link: 'https://github.com/Avularamesh/sabear_simplecutomerapp.git'. Below this, there is an error message: 'Failed to connect to repository : Error performing git command: git ls-remote -h https://github.com/Avularamesh/sabear_simplecutomerapp.git HEAD'. The 'Credentials' dropdown is set to 'none'. At the bottom, there are 'Save' and 'Apply' buttons.

Here I am facing the issue.

To solve that we need install git in the server.

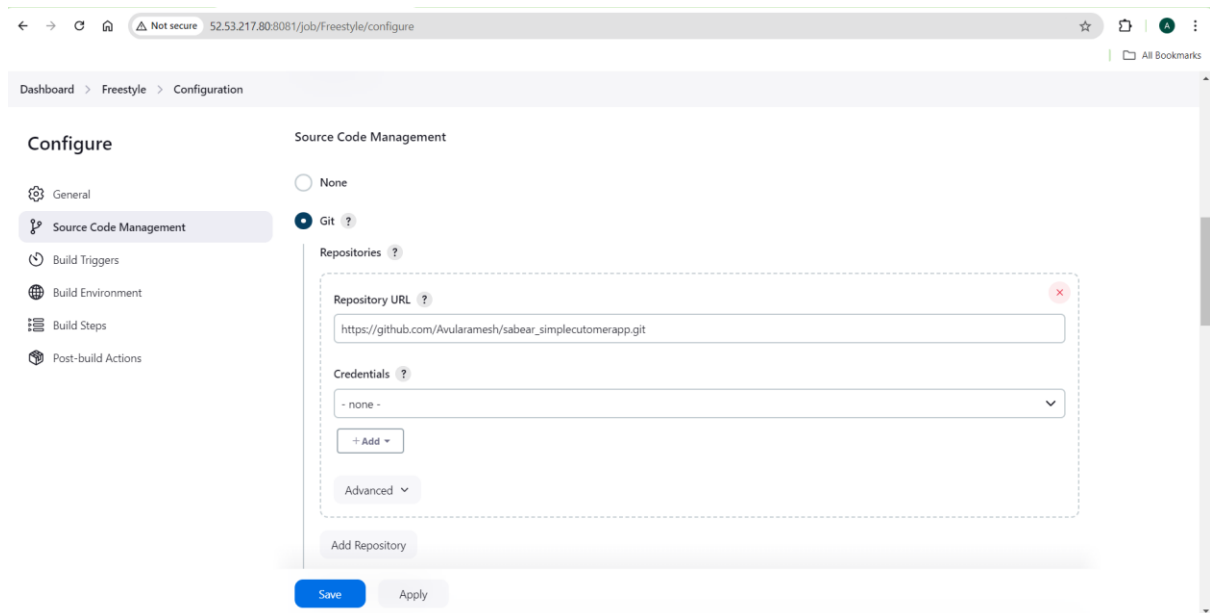
```
root@ip-172-31-13-31 ~]# sudo yum -y install git
loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core | 3.6 kB 00:00:00
Resolving Dependencies
--> Running transaction check
--> Package git.x86_64 0:2.40.1-1.amzn2.0.3 will be installed
--> Processing Dependency: git-core = 2.40.1-1.amzn2.0.3 for package: git-2.40.1-1.amzn2.0.3.x86_64
--> Processing Dependency: git-core-doc = 2.40.1-1.amzn2.0.3 for package: git-2.40.1-1.amzn2.0.3.x86_64
--> Processing Dependency: perl-Git = 2.40.1-1.amzn2.0.3 for package: git-2.40.1-1.amzn2.0.3.x86_64
--> Processing Dependency: perl(Git) for package: git-2.40.1-1.amzn2.0.3.x86_64
--> Processing Dependency: perl(Term::ReadKey) for package: git-2.40.1-1.amzn2.0.3.x86_64
--> Running transaction check
--> Package git-core.x86_64 0:2.40.1-1.amzn2.0.3 will be installed
--> Package git-core-doc.noarch 0:2.40.1-1.amzn2.0.3 will be installed
--> Package perl-Git.noarch 0:2.40.1-1.amzn2.0.3 will be installed
--> Processing Dependency: perl(Error) for package: perl-Git-2.40.1-1.amzn2.0.3.noarch
--> Package perl-TermReadKey.x86_64 0:2.30-20.amzn2.0.2 will be installed
--> Running transaction check
--> Package perl-Error.noarch 1:0.17020-2.amzn2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Installing:
git x86_64 2.40.1-1.amzn2.0.3 amzn2-core 54 k
Installing for dependencies:
git-core x86_64 2.40.1-1.amzn2.0.3 amzn2-core 10 M
git-core-doc noarch 2.40.1-1.amzn2.0.3 amzn2-core 3.0 M
perl-Error noarch 1:0.17020-2.amzn2 amzn2-core 32 k
perl-Git noarch 2.40.1-1.amzn2.0.3 amzn2-core 42 k
perl-TermReadKey x86_64 2.30-20.amzn2.0.2 amzn2-core 31 k
=====
Transaction Summary
=====
Install 1 Package (+5 Dependent packages)

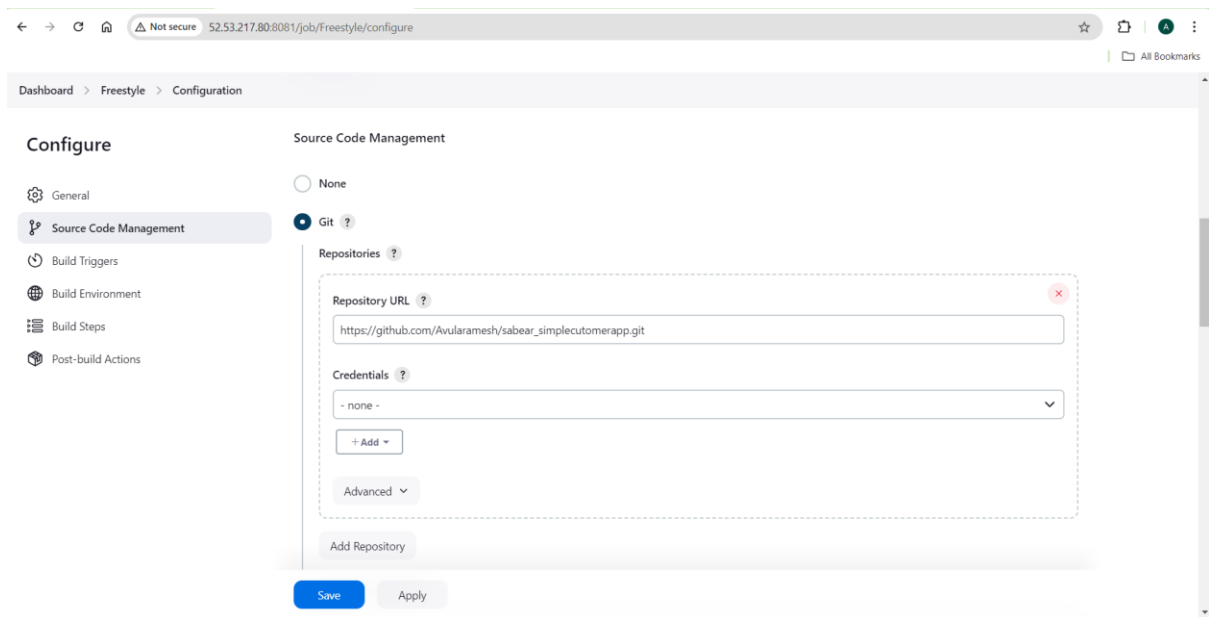
Total download size: 13 M
Installed size: 44 M
```

Now go to the Jenkins see there error is there or not. Just refresher.



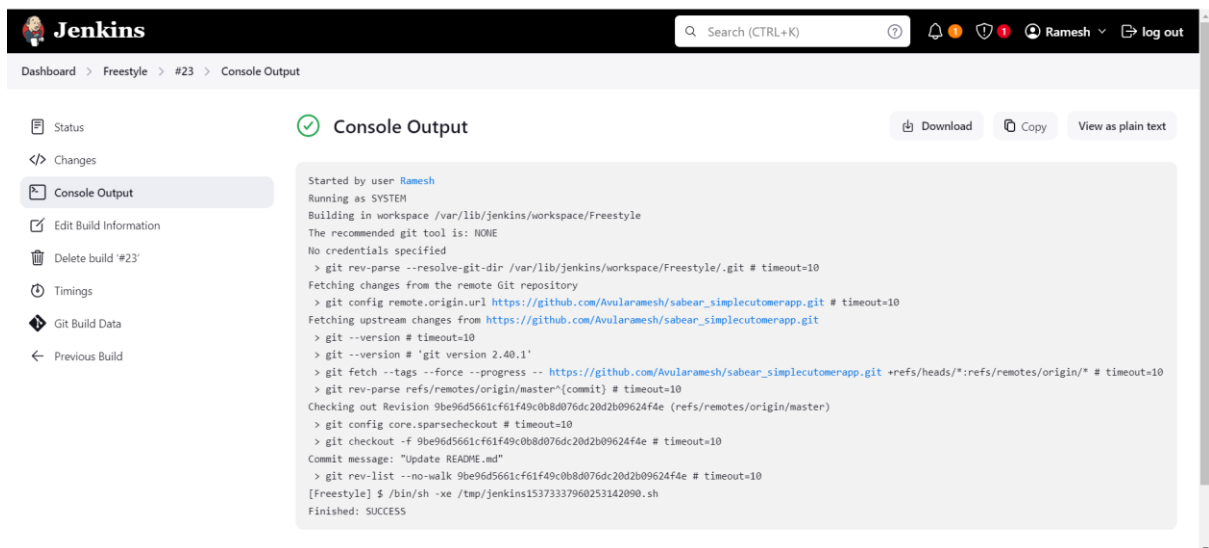
Click on save.

Now go to these another job and do same what we done in the previous Job.

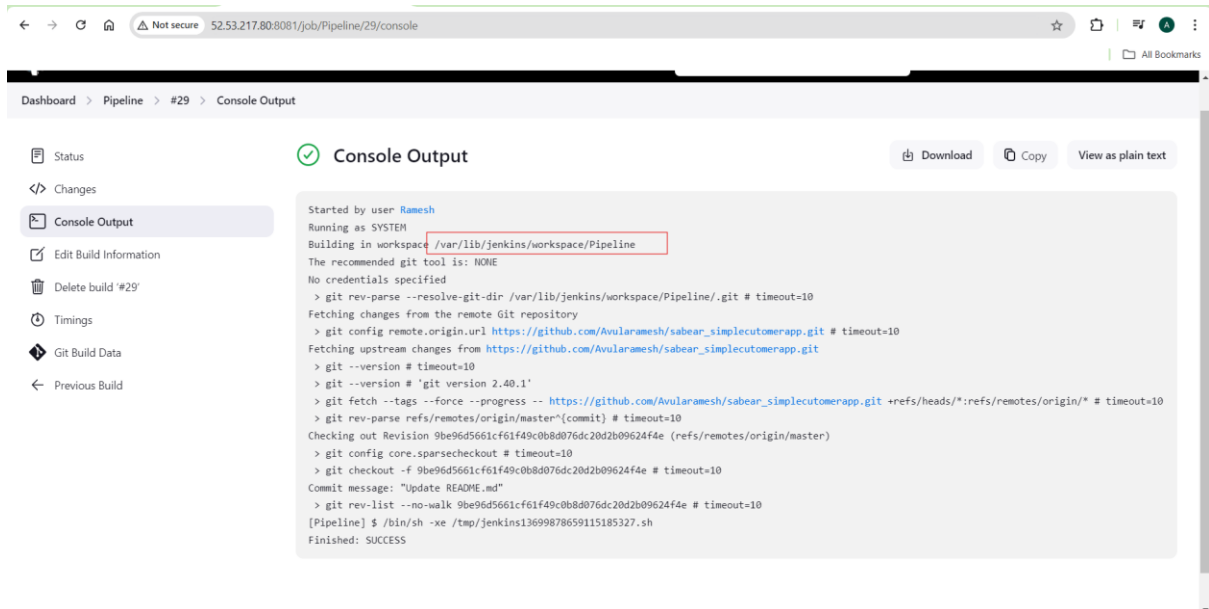


Now try to build these two jobs.

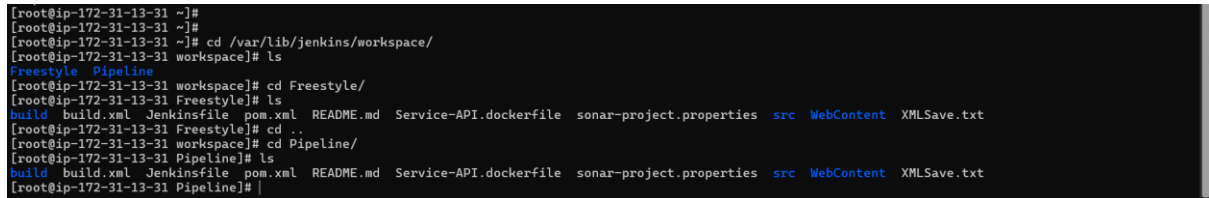
Build is success freestyle.



- Now Build the Another Job.
- These job also build is success.
- If you want see these code --- Go to this dir---/var/lib/jenkins/workspace/



Here our code is Available.

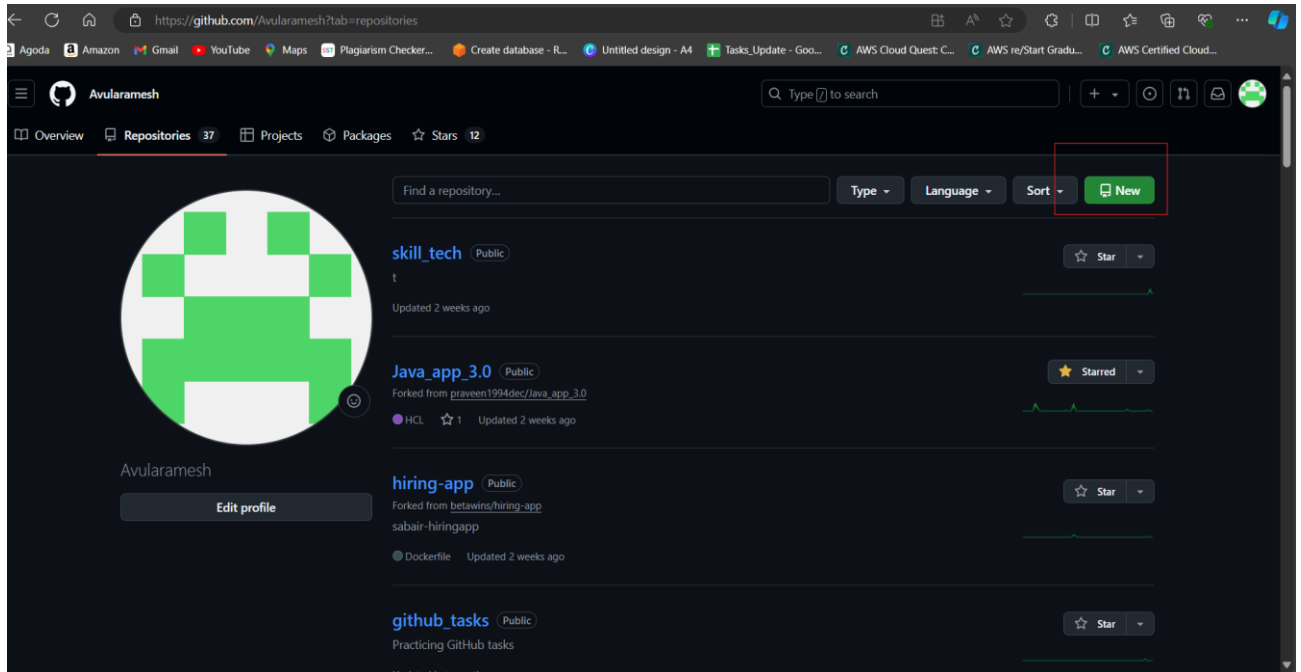


#####Task is Done#####

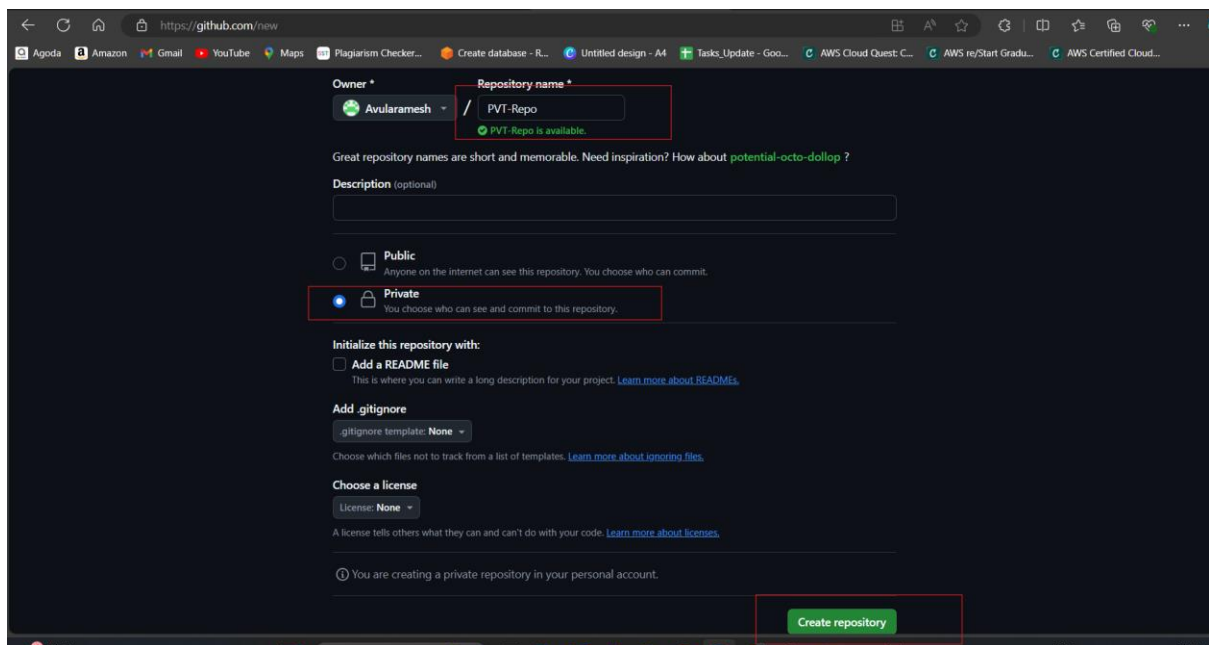
6) Create one jenkins job using git hub Private repository.

To do this Task we have to create Pvt git hub repo.

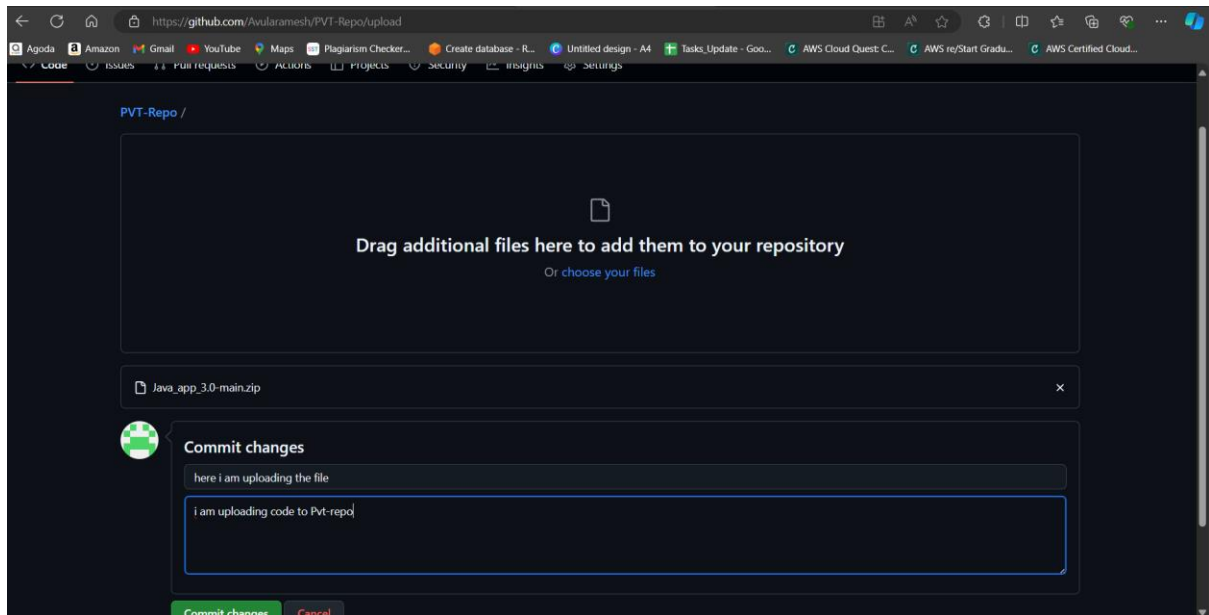
Click on New.



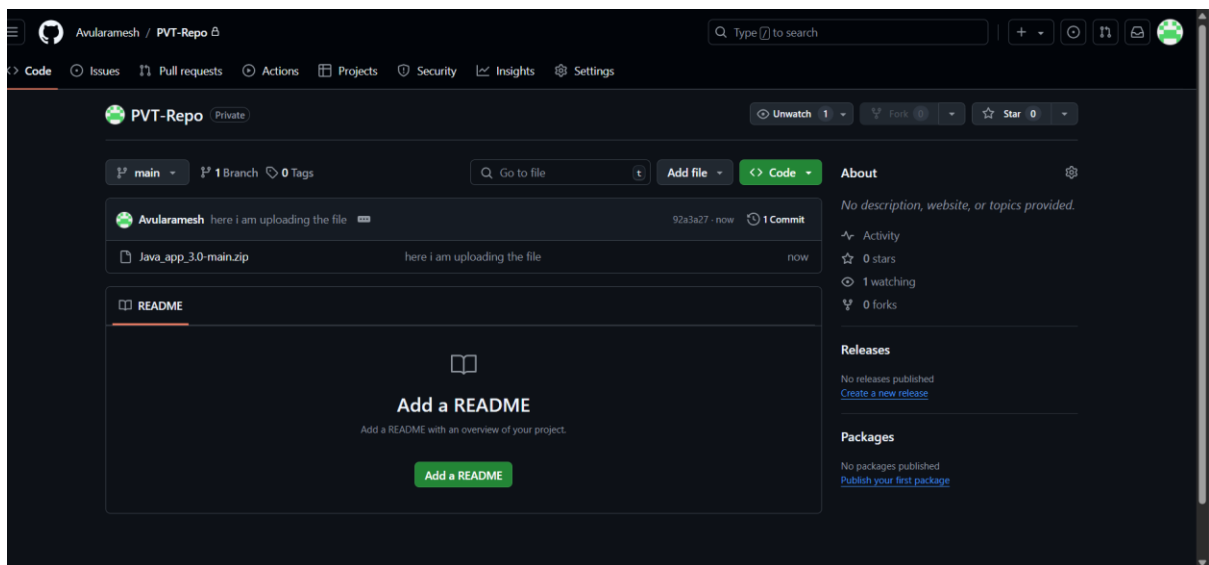
Here give repo name, select as PVT and click on create Repo



Her I am committing and click on commit changes.

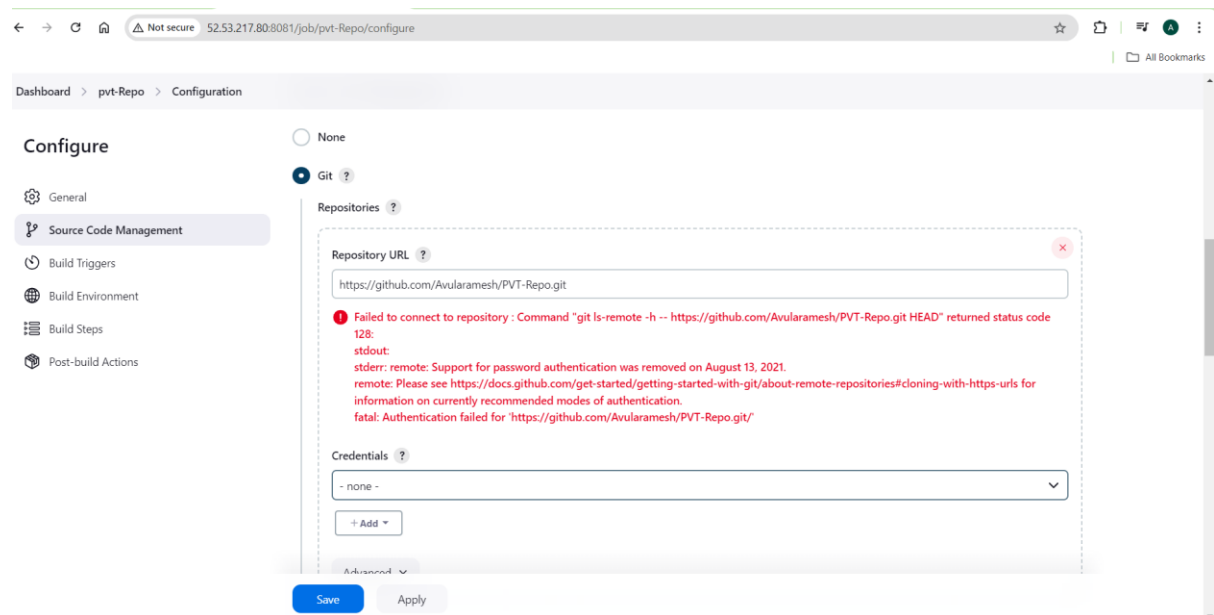


The file is uploaded in github Pvt-Repo

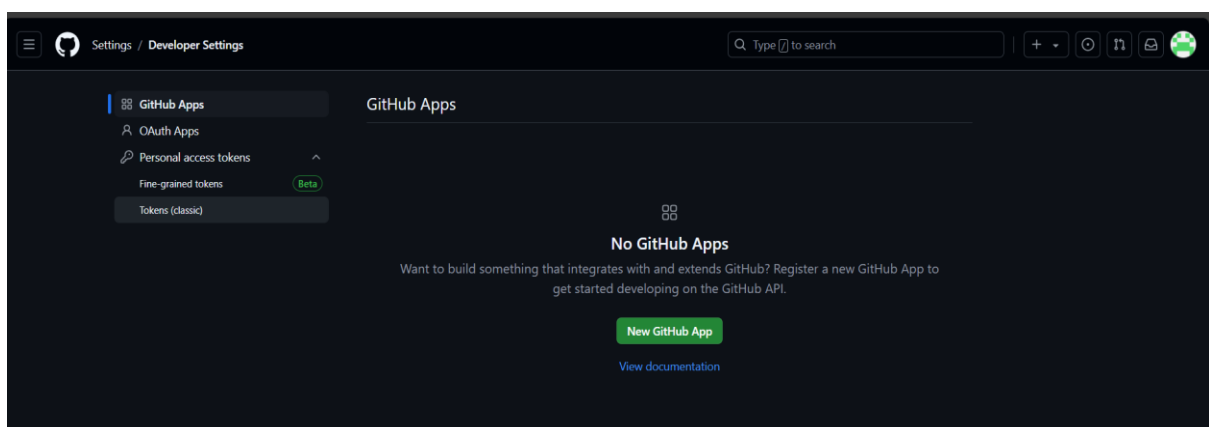


Now the code copy in the Job.

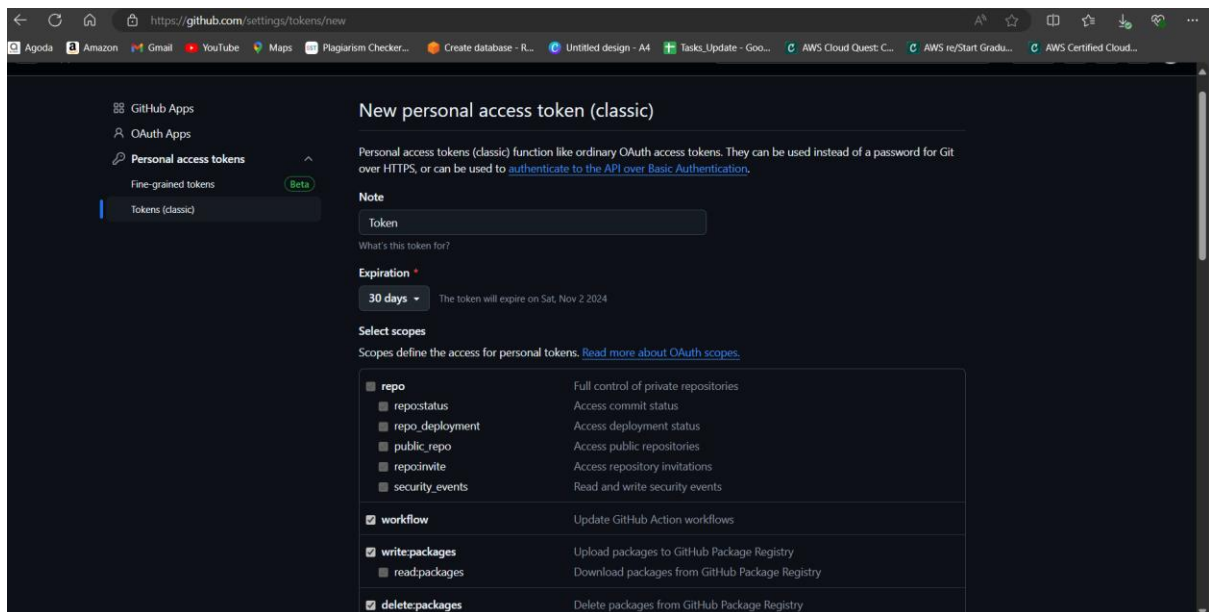
After giving the repo url we are facing these issue because the repo will be Pvt.



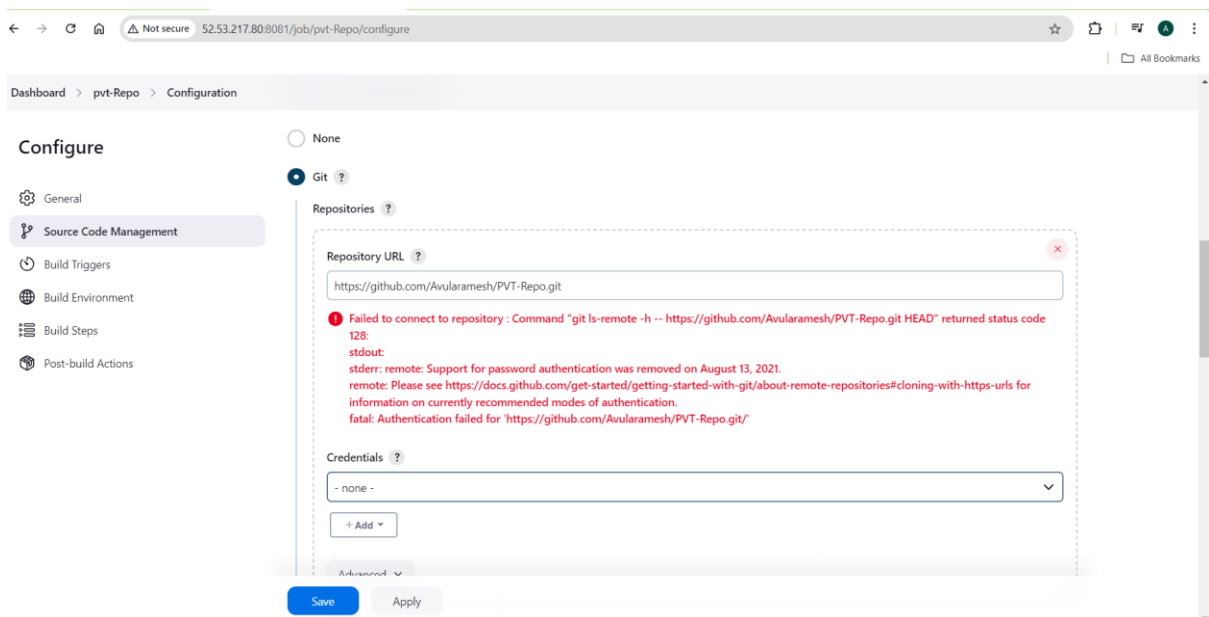
- So first we need to create Token for Github and add the credentials in github.
- Go to the github.
- Go to github repo and create github Token .
- To create Token go to the Setting.
- Go Developer Setting, Click on Token classic and click on create Token



Now select the what are the option do you want and click on generate Token.



Now back to Jenkins and under the credentials click on Add.



The below you will see the below interface and fill all.

User name, Password, ID, give description also.

The screenshot shows the 'Jenkins Credentials Provider: Jenkins' configuration window. It contains the following fields and options:

- Username with password**: A dropdown menu.
- Scope**: A dropdown menu with the value 'Global (Jenkins, nodes, items, all child items, etc)'.
- Username**: A text input field containing 'Avularamesh'.
- Treat username as secret**: An unchecked checkbox.
- Password**: A text input field with masked characters (dots).
- ID**: A text input field containing 'Token'.
- Description**: A text input field.

- Just click on add.
- Just select the credential and add.
- After adding clear the issue.
- Give main branch.

The screenshot shows the 'Configure' page for a Jenkins job. The 'Source Code Management' tab is selected. The 'Repositories' section contains the following fields:

- Repository URL**: A text input field containing 'https://github.com/Avularamesh/PVT-Repo.git'.
- Credentials**: A dropdown menu showing 'Avularamesh/***** (Token)'.
- + Add**: A button to add a new repository.
- Advanced**: A dropdown menu.
- Add Repository**: A button.

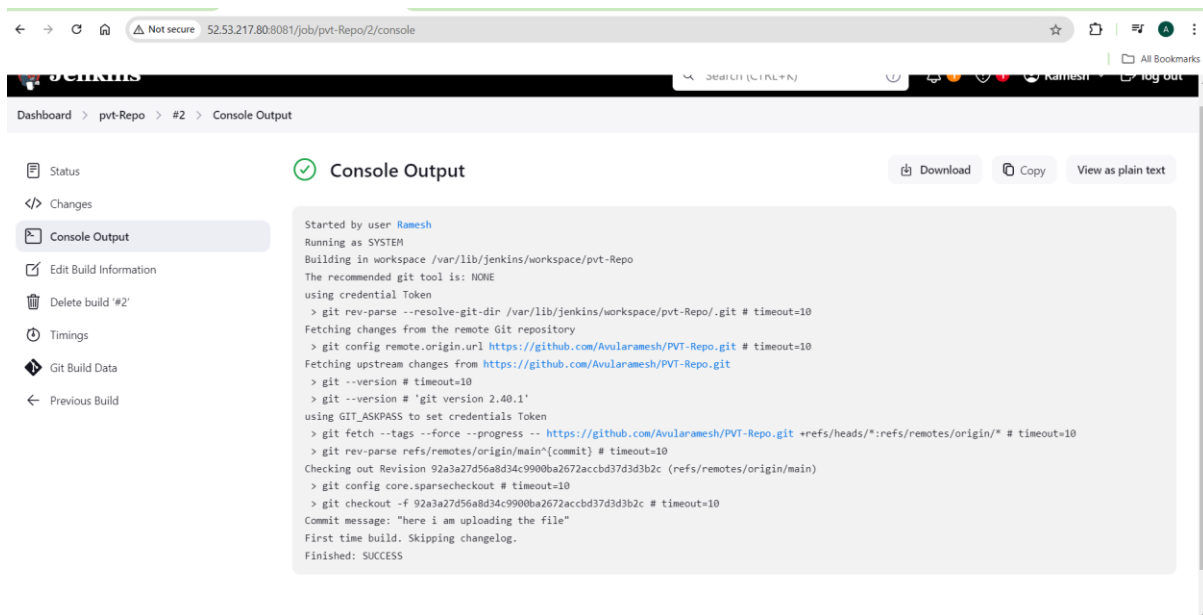
The 'Branches to build' section contains the following field:

- Branch Specifier (blank for 'any')**: A text input field containing '*/main'.

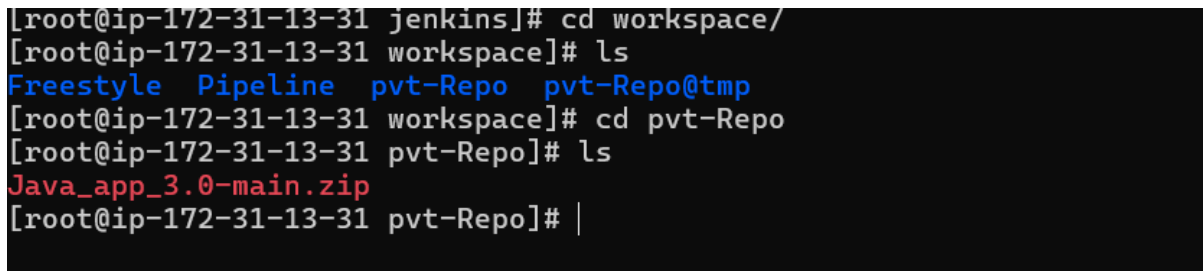
At the bottom, there are 'Save' and 'Apply' buttons.

Now try to build the job.

Build is success.



In Workspace you will see the our file.



#####Task is Done#####