The following table lists the IP addresses of the VMs installed with AlmaLinux 9 for the labs based on Git, GitHub, Jenkins, Maven, Tomcat, Apache Webserver, Ansible, Docker, and Docker Hub.

VM Name	**IP Address**
Developers	192.168.10.110
Jenkins	192.168.10.111
WebServer	192.168.10.112
Ansible	192.168.10.113
Docker	192.168.10.114
Jslave1	192.168.10.115

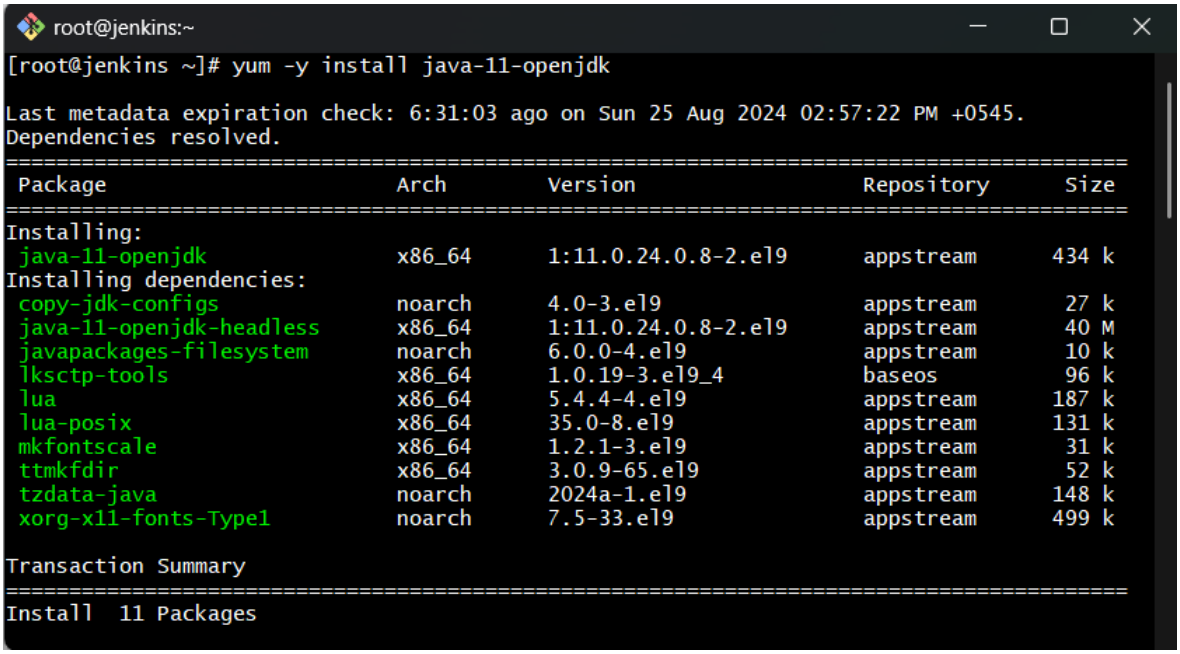
Table of Contents

Project 1

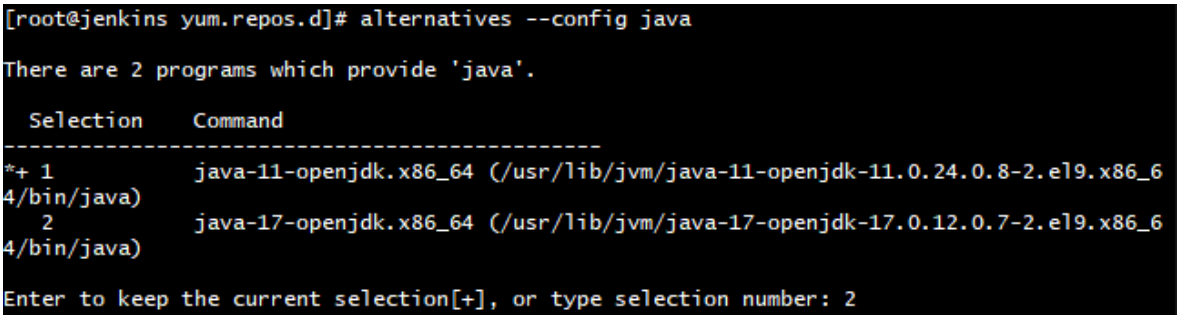
Setup CI/CD with GitHub, Jenkins, Apache Webserver to deploy an artifact (http-based website).

Step 1: Set up Jenkins

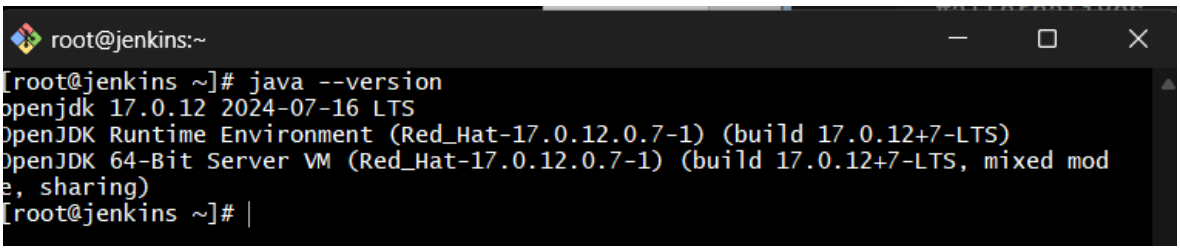
a. Install Java



- If you have an old version of JDK installed, configure the alternative Java version



- Select the appropriate Java version and verify:



b. Set the JAVA_HOME environment variable:

```
root@jenkins:/usr/lib/jvm/java-17-openjdk-17.0.12.0.7-2.el9.x86_64
[root@jenkins ~]# cd /usr/lib/jvm/java-17-openjdk-17.0.12.0.7-2.el9.x86_64/
[root@jenkins java-17-openjdk-17.0.12.0.7-2.el9.x86_64]# pwd
/usr/lib/jvm/java-17-openjdk-17.0.12.0.7-2.el9.x86_64
[root@jenkins java-17-openjdk-17.0.12.0.7-2.el9.x86_64]#
```

- Copy the output path and add it to the .bash_profile:

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi
JAVA_HOME=/usr/lib/jvm/java-17-openjdk-17.0.12.0.7-2.el9.x86_64
PATH=$PATH:$HOME/bin:$JAVA_HOME
# User specific environment and startup programs
~
```

- Refresh the environment:

```
[root@jenkins yum.repos.d]# . /root/.bash_profile
[root@jenkins yum.repos.d]#
```

c. Install Jenkins

```
root@jenkins:/etc/yum.repos.d
[root@jenkins /]# cd /etc/yum.repos.d/
[root@jenkins yum.repos.d]#
```

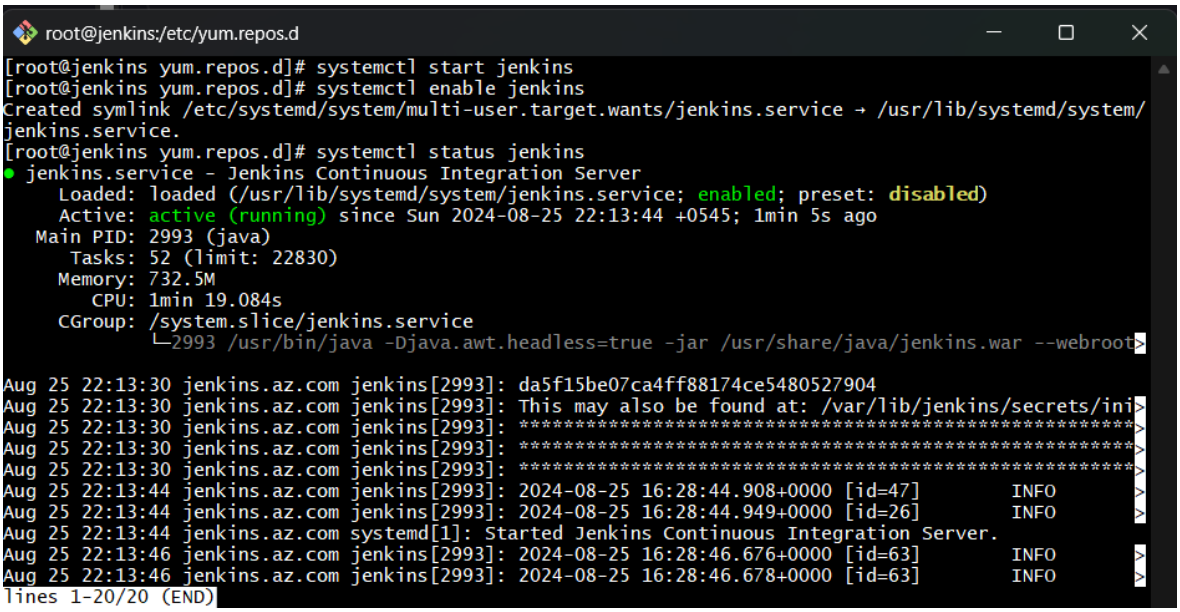
- Browse the official Jenkins website for the genuine Jenkins repository URL and key.

```
[root@jenkins yum.repos.d]# wget https://pkg.jenkins.io/redhat-stable/jenkins.repo
--2024-08-25 22:07:45--  https://pkg.jenkins.io/redhat-stable/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io)... 199.232.22.133, 2a04:4e42:42::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|199.232.22.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 85
Saving to: 'jenkins.repo'

jenkins.repo              100%[===================================>]      85  --.-KB/s    in 0s

2024-08-25 22:07:45 (1.92 MB/s) - 'jenkins.repo' saved [85/85]

[root@jenkins yum.repos.d]# rpm --import https://pkg.jenkins.io/redhat/jenkins.io-2023.key
[root@jenkins yum.repos.d]# yum upgrade
Jenkins-stable                                                  16 kB/s |  29 kB     00:01
Dependencies resolved.
Nothing to do.
Complete!
```

```
[root@jenkins yum.repos.d]# yum install jenkins
Last metadata expiration check: 0:00:19 ago on Sun 25 Aug 2024 10:08:35 PM +0545.
Dependencies resolved.
================================================================================
 Package              Architecture         Version                  Repository              Size
================================================================================
Installing:
 jenkins              noarch               2.462.1-1.1              jenkins                 89 M

Transaction Summary
================================================================================
```

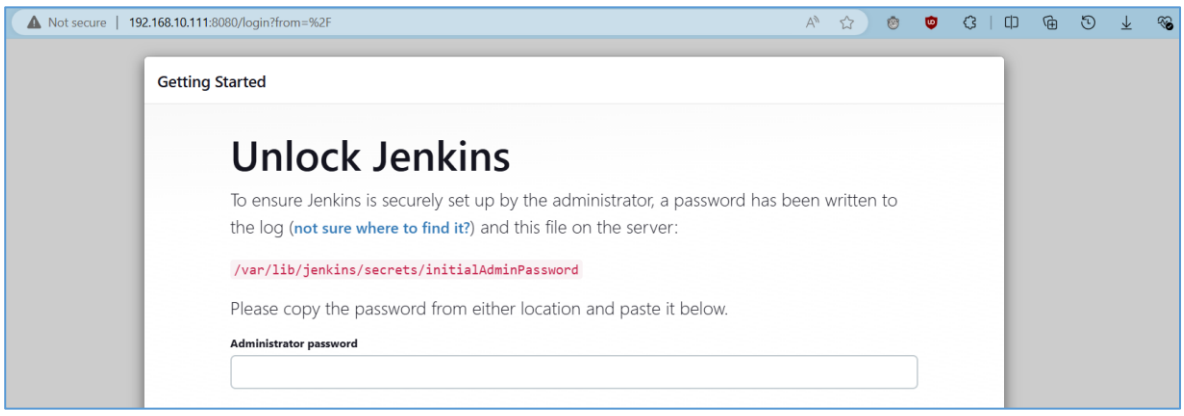
```
[root@jenkins yum.repos.d]# rpm -q jenkins
jenkins-2.462.1-1.1.noarch
```

d. Start the Jenkins service:

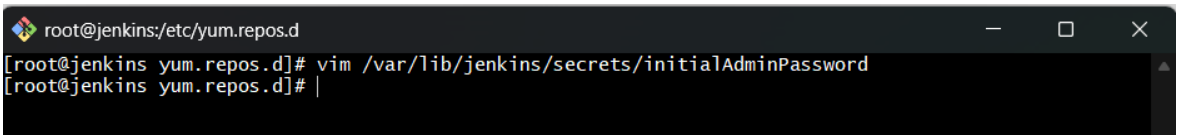


e. Add the Jenkins port to the firewall for global access:

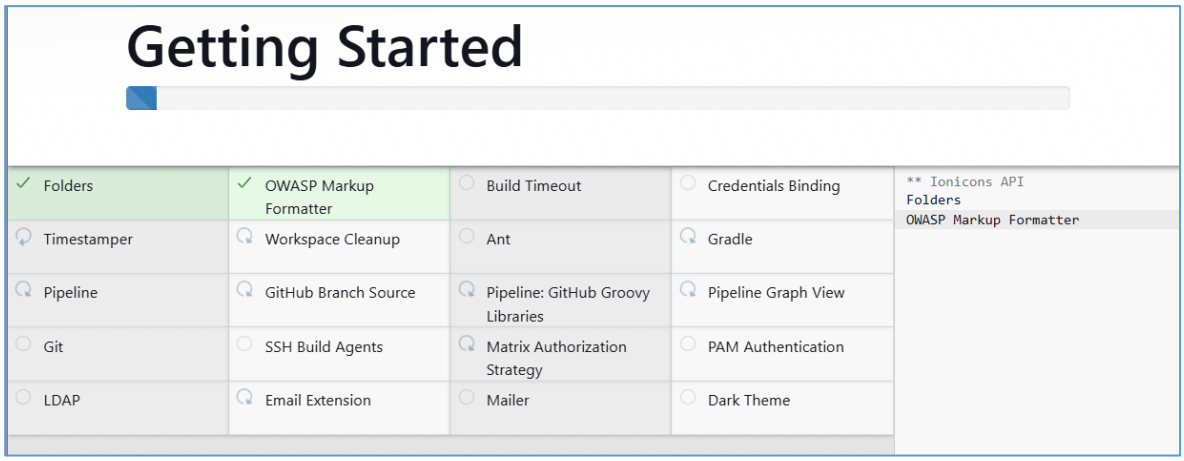
f. Access the Jenkins dashboard:



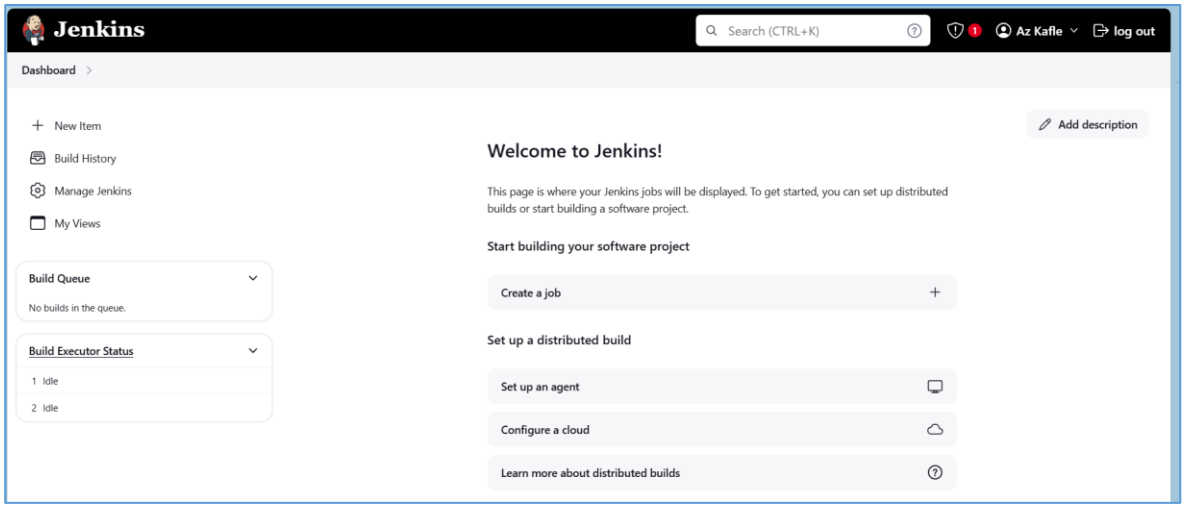
- Follow the instructions to unlock Jenkins using the password provided in the specified location.



- Install the suggested plugins.

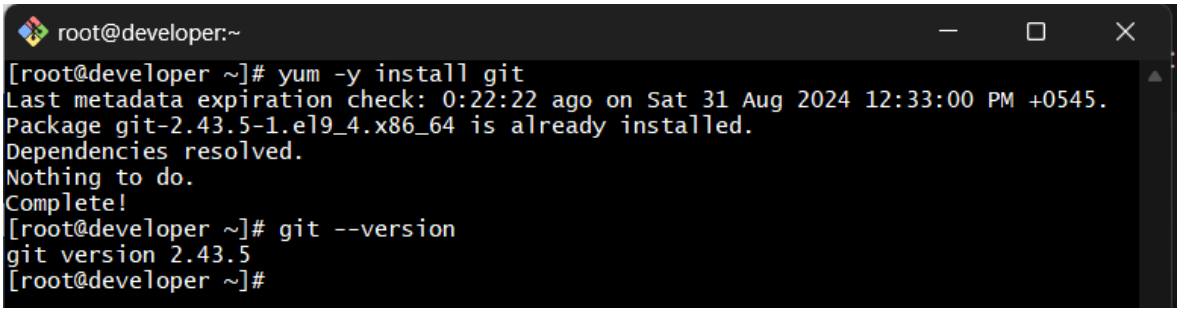


- Create the first admin user.
- Start using Jenkins!



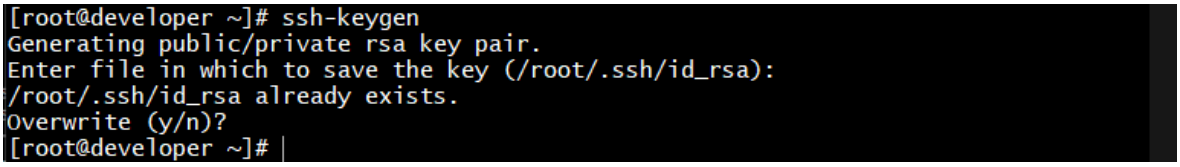
Step 2: Git installation and configuration in Developer and Jenkins Machine

a. Installing Git in Developer Machine

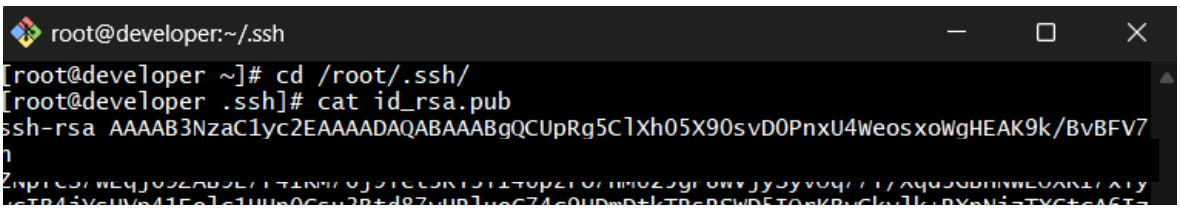


b. Configuring SSH Authentication with GitHub

- Generate an SSH key pair

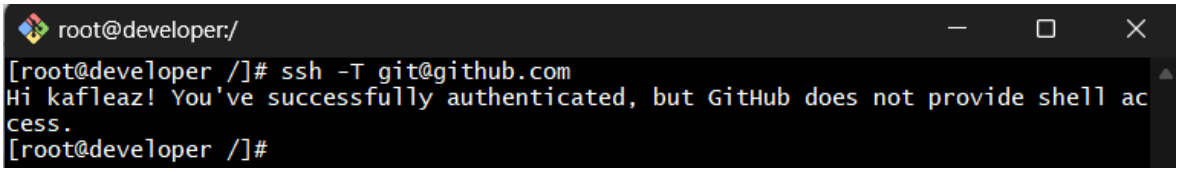


- Navigate to the SSH directory and display the public key.



- Copy the displayed key and add it to GitHub:

- Test the SSH connection:



Note: Follow the same process in Jenkins machine to configure Git.

Step 3: Set up Apache Webserver

a. Installing Apache Server and PHP

```
root@webserver:~
[root@webserver ~]# yum -y install httpd php
Last metadata expiration check: 0:02:17 ago on Wed 28 Aug 2024 12:26:40 PM +0545.
Package httpd-2.4.57-11.el9_4.1.x86_64 is already installed.
Package php-8.0.30-1.el9_2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@webserver ~]#
```

b. Start and Enable the Apache service to Start on Boot.

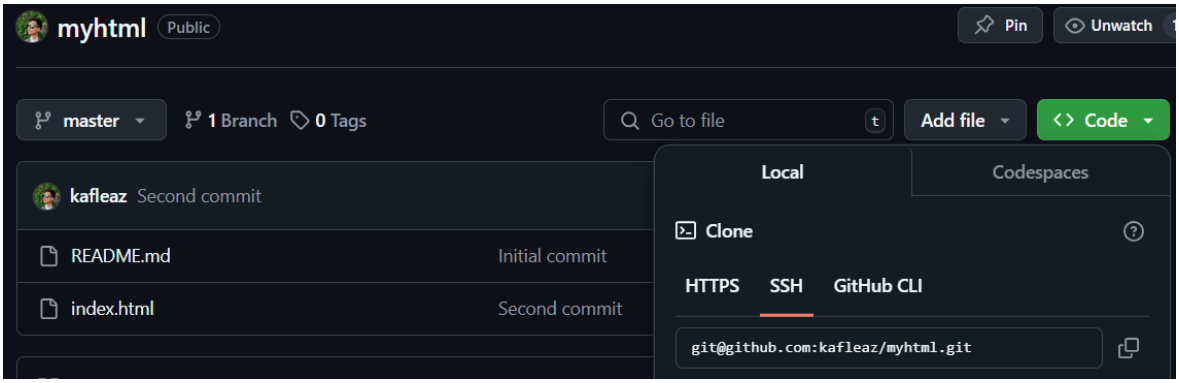
```
[root@webserver ~]# systemctl start httpd
[root@webserver ~]# systemctl enable httpd
```

c. Configure Firewall to Allow HTTP Traffic:

```
[root@webserver ~]# systemctl start httpd
[root@webserver ~]# systemctl enable httpd
[root@webserver ~]# firewall-cmd --permanent --add-service=http
Warning: ALREADY_ENABLED: http
success
[root@webserver ~]# firewall-cmd --reload
success
[root@webserver ~]#
```

Step 4: Creating a Local Git Repository

a. Create a new repository named "myhtml" in github and copy the URL.



b. Create a project workspace directory:

```
root@developer:~/project/githubtest
[root@developer ~]# mkdir -p /root/project/githubtest
[root@developer ~]# cd /root/project/githubtest/
```

c. Initialize, Link and pull a Git repository:

```
MINGW64:/c/Users/user
[root@developer githubtest]# git init
Reinitialized existing Git repository in /root/project/githubtest/.git/
[root@developer githubtest]# git remote add origin git@github.com:kafleaz/myhtml.git
[root@developer githubtest]# git pull origin master
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 9 (delta 0), reused 6 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (9/9), 1.43 KiB | 122.00 KiB/s, done.
From github.com:kafleaz/myhtml
 * branch            master     -> FETCH_HEAD
 * [new branch]      master     -> origin/master
```

d. Create a new file and add content.

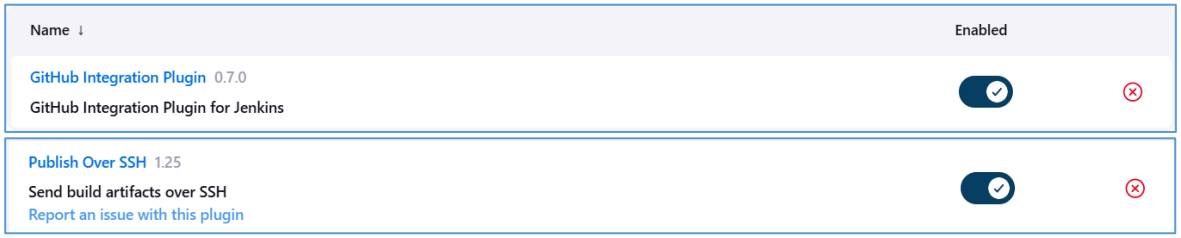
```
[root@developer githubtest]# vi index.html
[root@developer githubtest]# git add .
[root@developer githubtest]# git commit -m "githubtest first commit"
```

```
[root@developer githubtest]# git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 324 bytes | 324.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:kafleaz/myhtml.git
   69e690c..d4237ab  master -> master
[root@developer githubtest]# client_loop: send disconnect: Connection reset by peer
```

Step 5: jenkins configuration for "Github" and "publish over ssh"

a. Install Required Plugins in Jenkins

- Navigate to Manage **Jenkins** > **Plugins** > **Available Plugins** > Search plugins
- Select it and click **Install without restart**.



b. jenkins and Github ssh authentication configuration

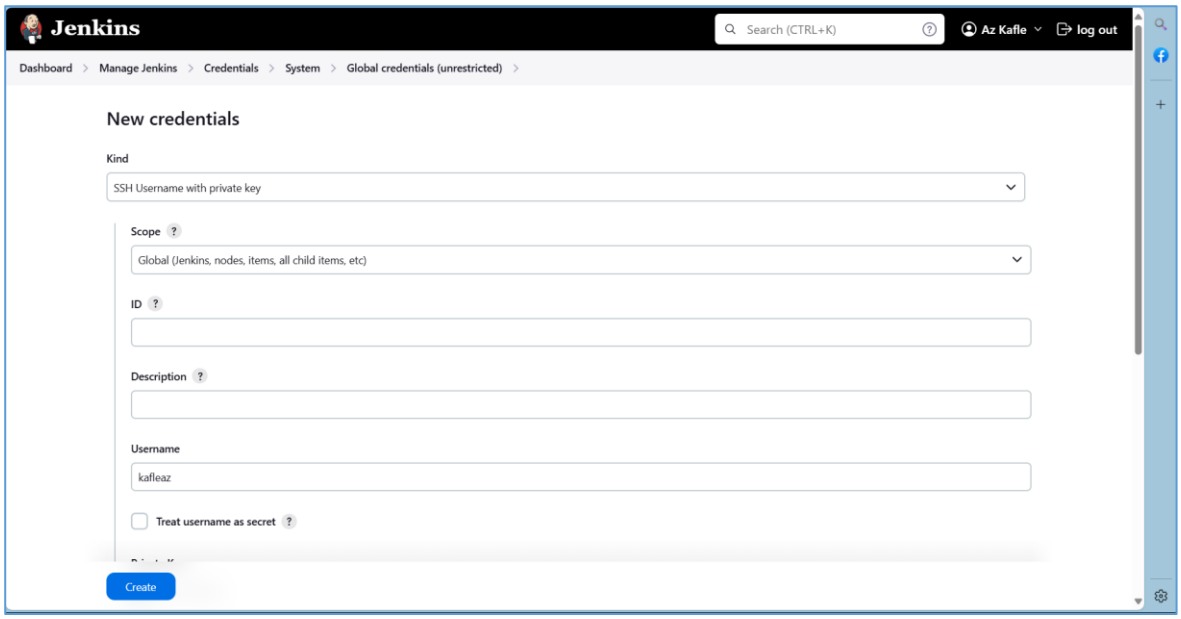
o Display the private key

```
[root@jenkins /]# cd /root/.ssh/
[root@jenkins .ssh]# cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAtibmtOA6pKvQ1B8TEan7f8L7KvVQUauCLBSt3TgTGLrJ1iwhhIU+
rQvLwnRA3TWvn8onL/MUIzSRJwHB+W8Otmkxx7bbSKjFhOZkeXqxOlgRj91miuasvBaD62
```

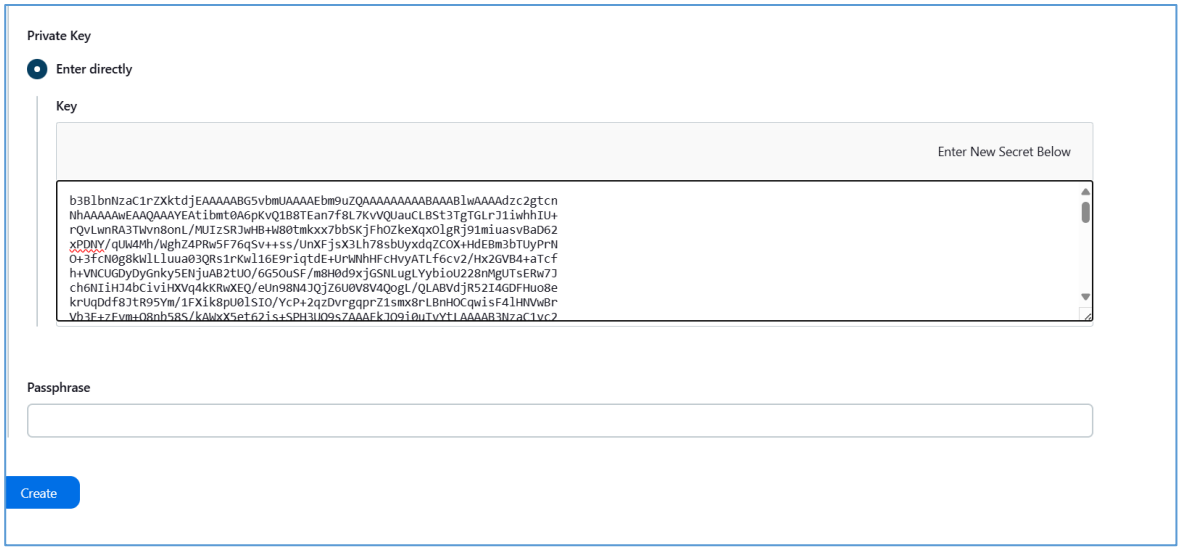
o In Jenkins, navigate to **Manage Jenkins** > **Credentials** > **System** > **Global credentials**.

o Click **Add Credentials**.

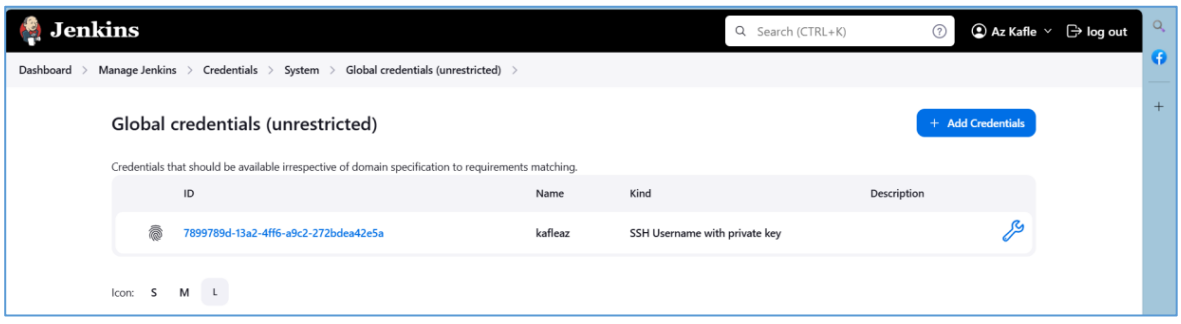
- Enter your **GitHub username**.



- Paste the copied private key.



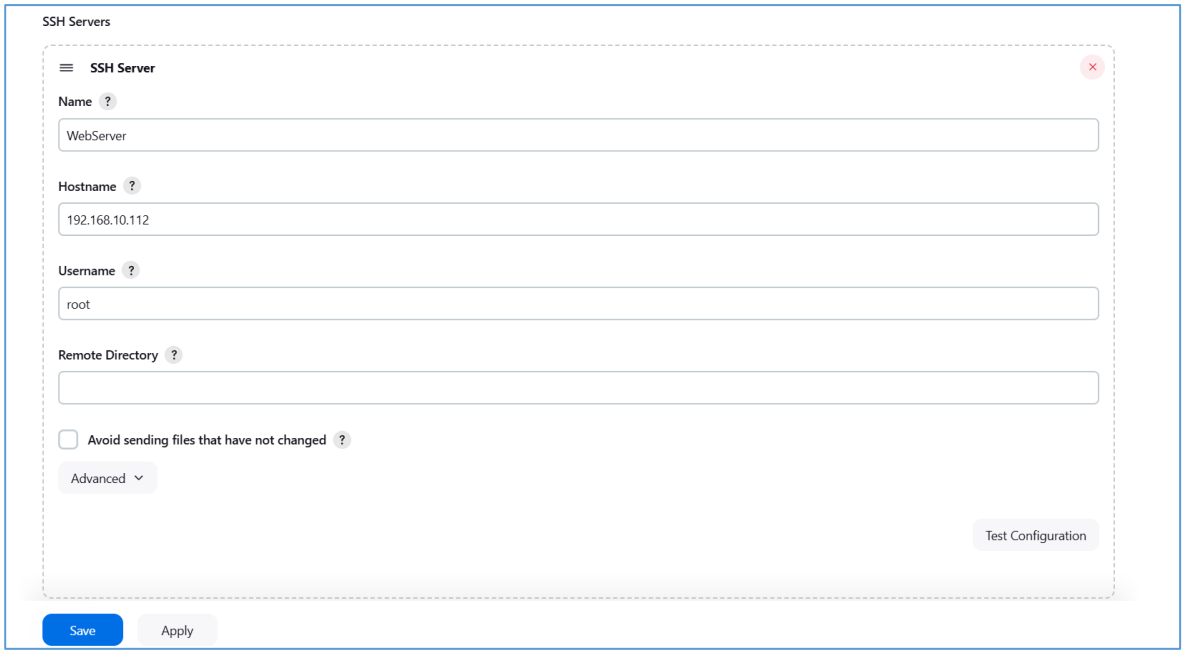
- Click **Create**.



Step 6: Set IP address, username and password of webserver in jenkins for automated file transfer (publish over ssh)

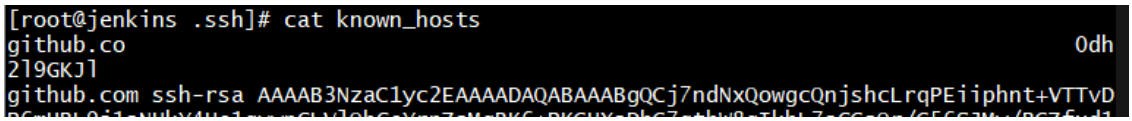
a. In Jenkins, navigate to **Manage Jenkins > Configure system > System > Publish Over SSH > SSH Servers**

- Enter SSH Server Details

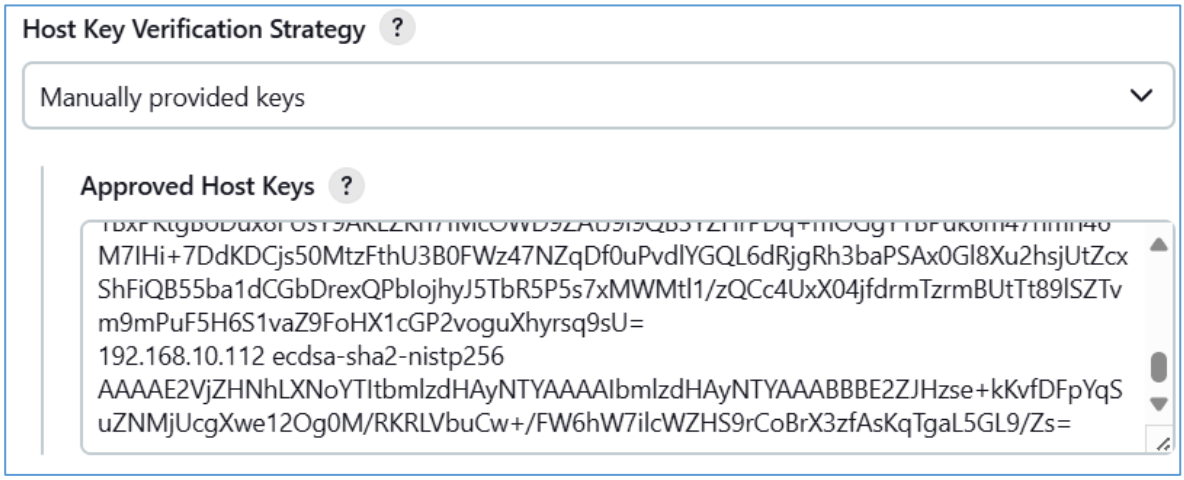


b. In Jenkins, navigate to **Manage Jenkins > Security > Git Host Key Verification Configuration**

- Obtain the known host key of Jenkins machine



- Manually provided key

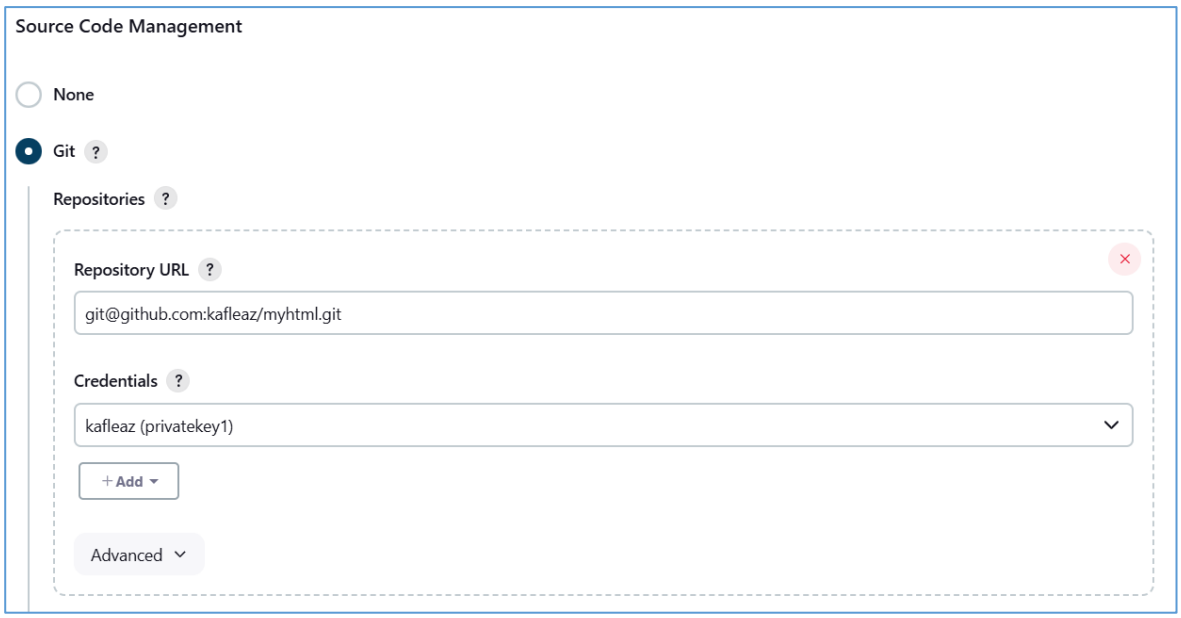


- **Save** Configuration:

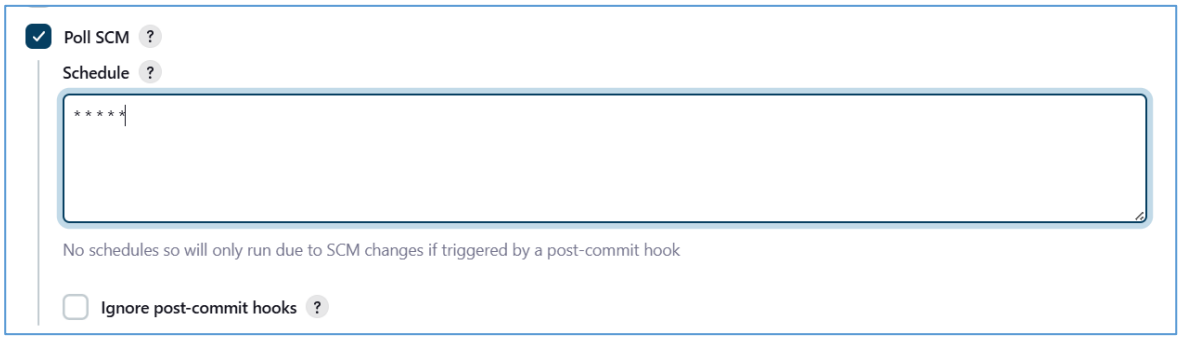
Step 7: Createing a job in jenkins to pull index.html from github repo. and send it to the webserver into /var/www/html directory

a. In Jenkins, navigate to **Jenkins Dashboard > New item >** enter item name **(Myhtml_p1) > Freestyle project > ok**

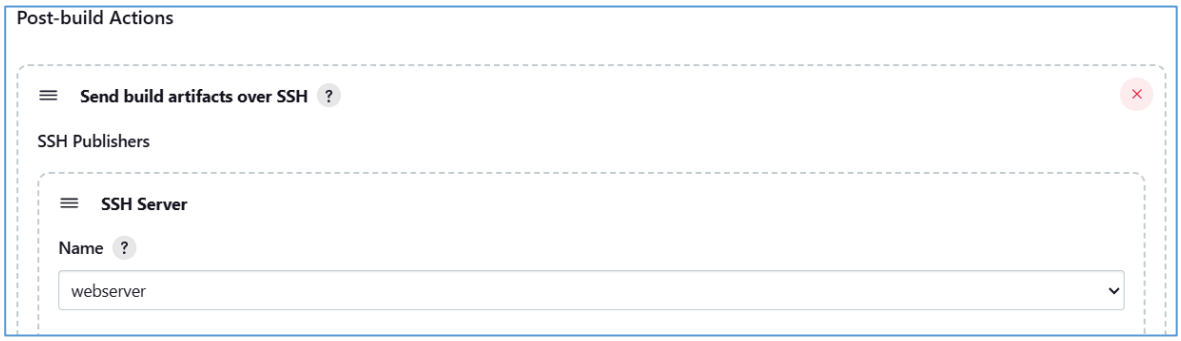
- Select **Git**, add **repo url** and **credentials**.



- **Build Trigger** > **Poll SCM**



- **post build Action > Send build artifact over ssh**

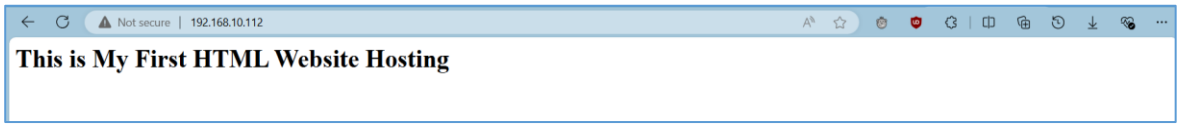


Step 8: Test the Deployment

- Check Console Output:

```
Started by user Az Kafle
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/Myhtml_p1
The recommended git tool is: NONE
using credential d5c8b220-5e76-406c-a00b-d317319da893
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Myhtml_p1/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url git@github.com:kafleaz/myhtml.git # timeout=10
Fetching upstream changes from git@github.com:kafleaz/myhtml.git
 > git --version # timeout=10
 > git --version # 'git version 2.43.5'
using GIT_SSH to set credentials privatekey1
[INFO] Currently running in a labeled security context
[INFO] Currently SELinux is 'enforcing' on the host
 > /usr/bin/chcon --type=ssh_home_t /var/lib/jenkins/workspace/Myhtml_p1@tmp/jenkins-gitclient-ssh15146473077806405728.key
Verifying host key using manually-configured host key entries
 > git fetch --tags --force --progress -- git@github.com:kafleaz/myhtml.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision d4237abd515ea0d687b191e5ffab19291fc654b4 (refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f d4237abd515ea0d687b191e5ffab19291fc654b4 # timeout=10
Commit message: "githubtest first commit"
 > git rev-list --no-walk d4237abd515ea0d687b191e5ffab19291fc654b4 # timeout=10
SSH: Connecting from host [jenkins.az.com]
SSH: Connecting with configuration [webserver] ...
SSH: Disconnecting configuration [webserver] ...
SSH: Transferred 1 file(s)
Finished: SUCCESS
```

- Verify Deployment

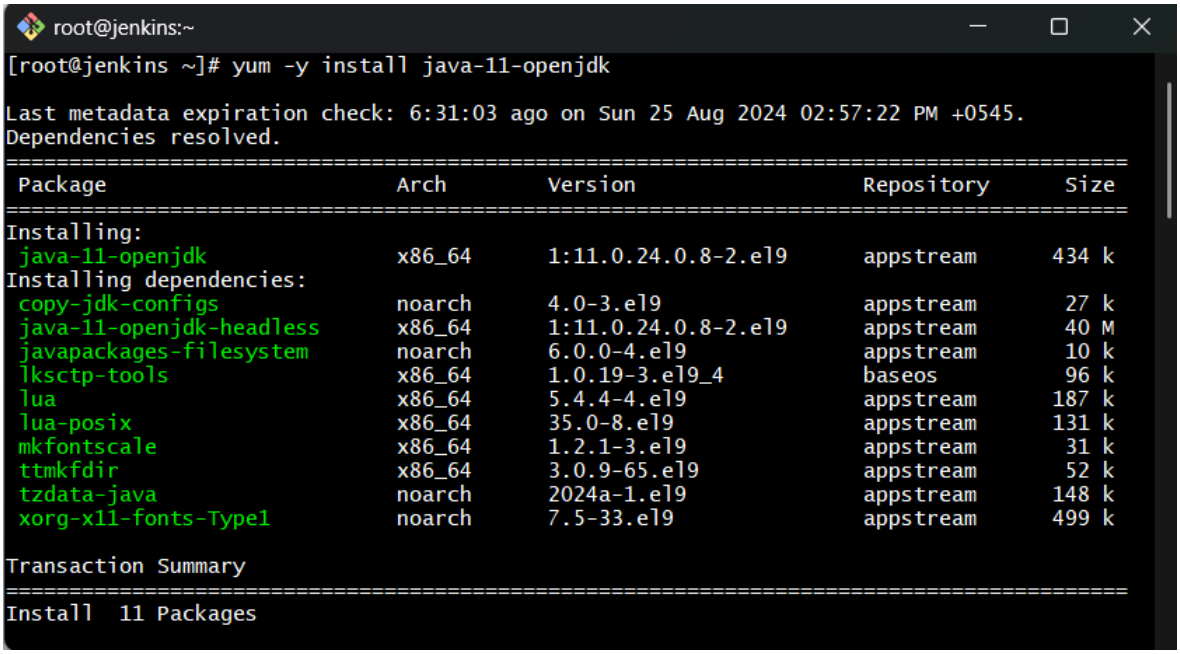


Project 2

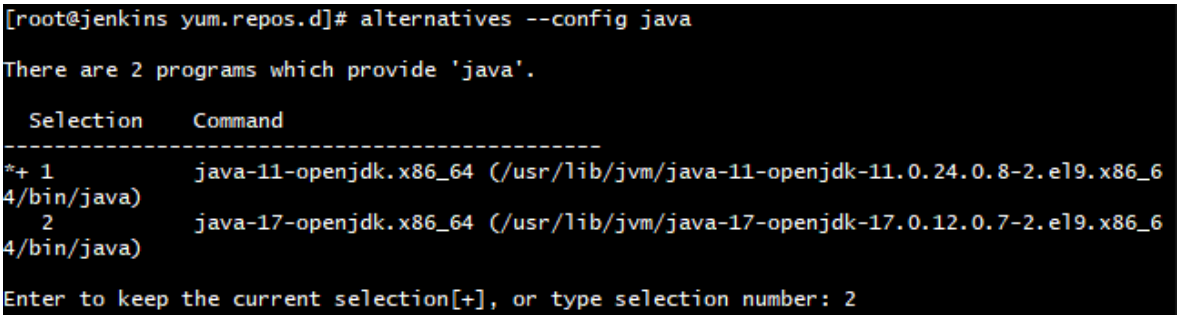
Setup CI/CD with GitHub, Jenkins, Apache Webserver to deploy an artifact (PHP-based website).

Step 1: Set up Jenkins

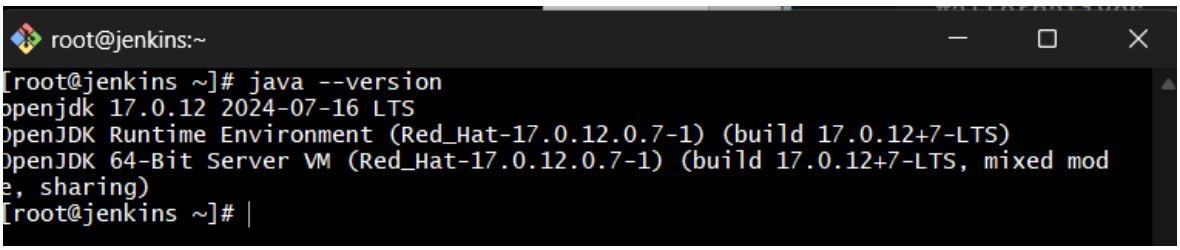
a. Install Java



- If you have an old version of JDK installed, configure the alternative Java version



- Select the appropriate Java version and verify:



- Set the JAVA_HOME environment variable:

```
root@jenkins:/usr/lib/jvm/java-17-openjdk-17.0.12.0.7-2.el9.x86_64
[root@jenkins ~]# cd /usr/lib/jvm/java-17-openjdk-17.0.12.0.7-2.el9.x86_64/
[root@jenkins java-17-openjdk-17.0.12.0.7-2.el9.x86_64]# pwd
/usr/lib/jvm/java-17-openjdk-17.0.12.0.7-2.el9.x86_64
[root@jenkins java-17-openjdk-17.0.12.0.7-2.el9.x86_64]#
```

- Copy the output path and add it to the .bash_profile:

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi
JAVA_HOME=/usr/lib/jvm/java-17-openjdk-17.0.12.0.7-2.el9.x86_64
PATH=$PATH:$HOME/bin:$JAVA_HOME
# User specific environment and startup programs
~
```

- Refresh the environment:

```
[root@jenkins yum.repos.d]# . /root/.bash_profile
[root@jenkins yum.repos.d]#
```

b. Install Jenkins

```
root@jenkins:/etc/yum.repos.d
[root@jenkins /]# cd /etc/yum.repos.d/
[root@jenkins yum.repos.d]#
```

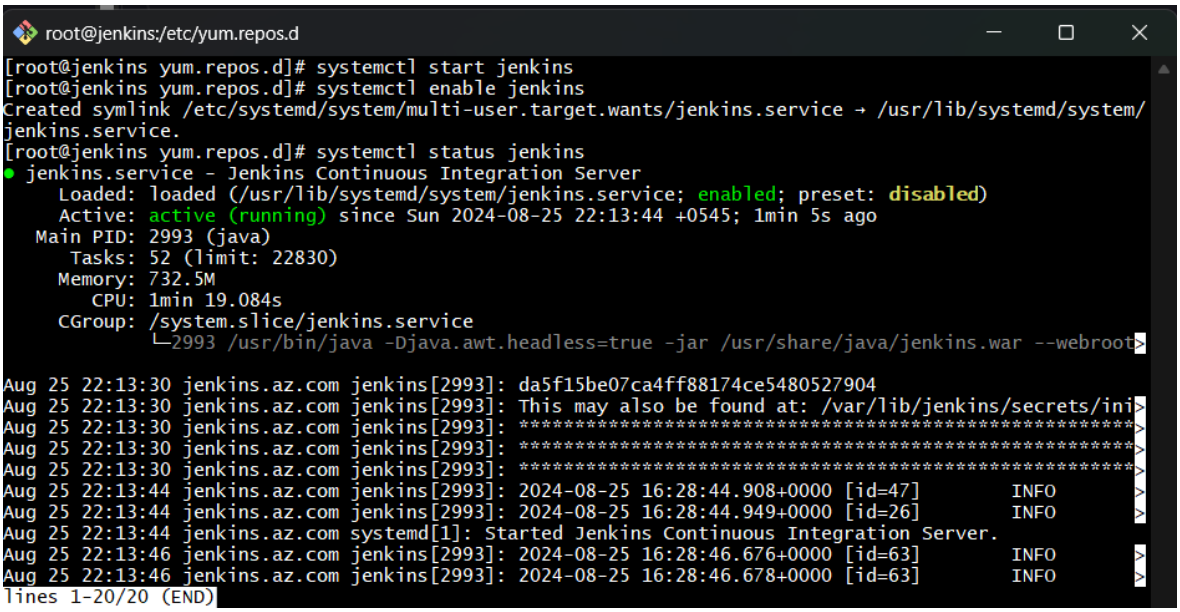
- Browse the official Jenkins website for the genuine Jenkins repository URL and key.

```
[root@jenkins yum.repos.d]# wget https://pkg.jenkins.io/redhat-stable/jenkins.repo
--2024-08-25 22:07:45--  https://pkg.jenkins.io/redhat-stable/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io)... 199.232.22.133, 2a04:4e42:42::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|199.232.22.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 85
Saving to: 'jenkins.repo'

jenkins.repo              100%[===================================>]      85  --.-KB/s    in 0s

2024-08-25 22:07:45 (1.92 MB/s) - 'jenkins.repo' saved [85/85]

[root@jenkins yum.repos.d]# rpm --import https://pkg.jenkins.io/redhat/jenkins.io-2023.key
[root@jenkins yum.repos.d]# yum upgrade
Jenkins-stable                                                  16 kB/s |  29 kB     00:01
Dependencies resolved.
Nothing to do.
Complete!
```

```
[root@jenkins yum.repos.d]# yum install jenkins
Last metadata expiration check: 0:00:19 ago on Sun 25 Aug 2024 10:08:35 PM +0545.
Dependencies resolved.
================================================================================
 Package              Architecture         Version                  Repository              Size
================================================================================
Installing:
 jenkins              noarch               2.462.1-1.1              jenkins                 89 M

Transaction Summary
================================================================================
```

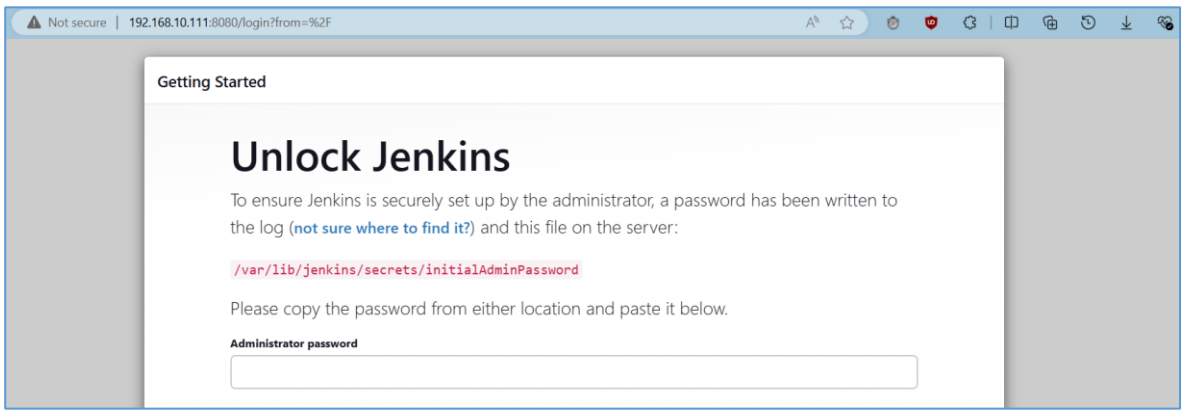
```
[root@jenkins yum.repos.d]# rpm -q jenkins
jenkins-2.462.1-1.1.noarch
```

- Start the Jenkins service:

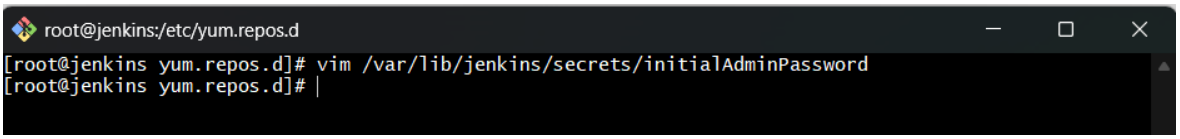


- Add the Jenkins port to the firewall for global access:

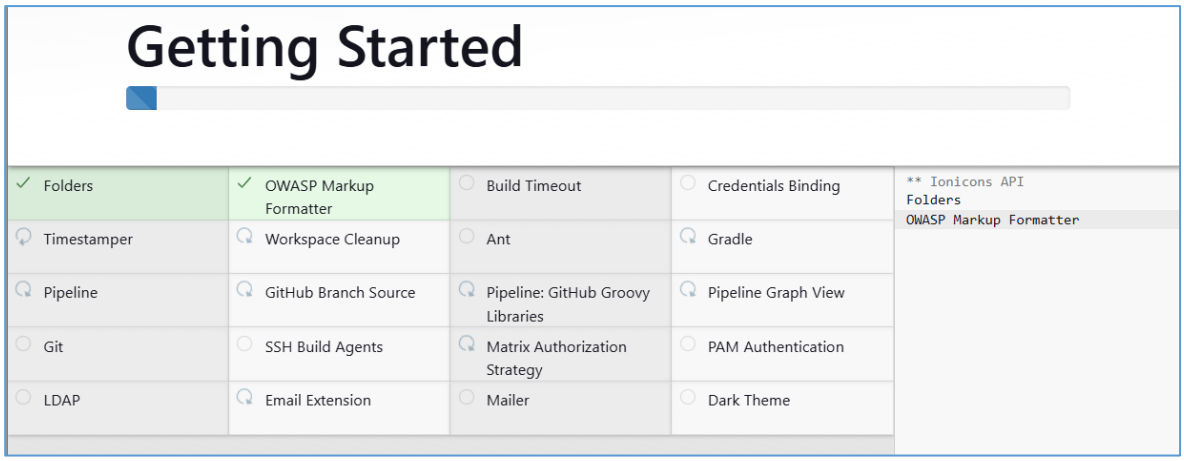
- Access the Jenkins dashboard:



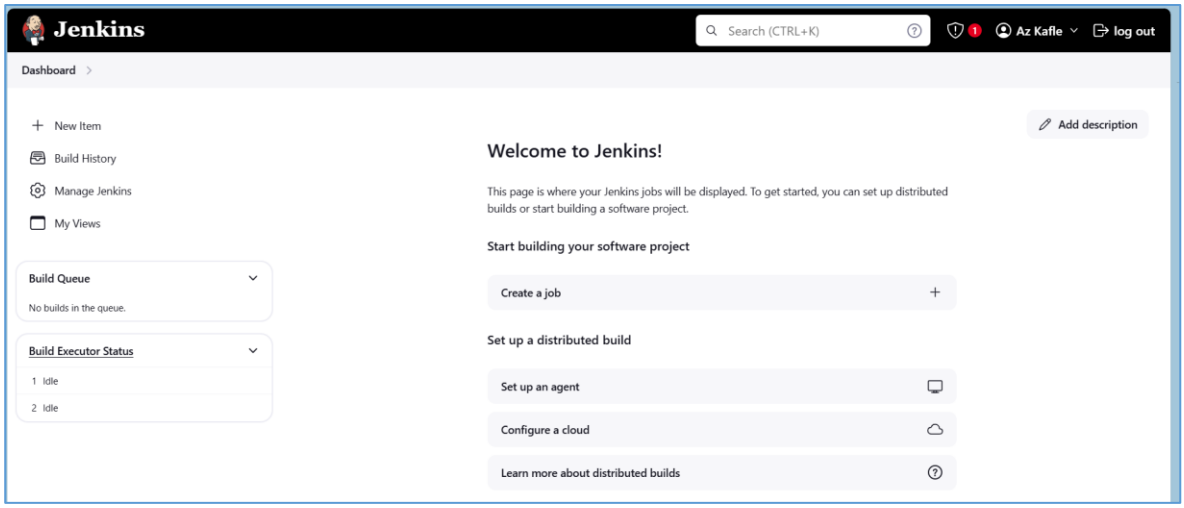
- Follow the instructions to unlock Jenkins using the password provided in the specified location.



- Install the suggested plugins.

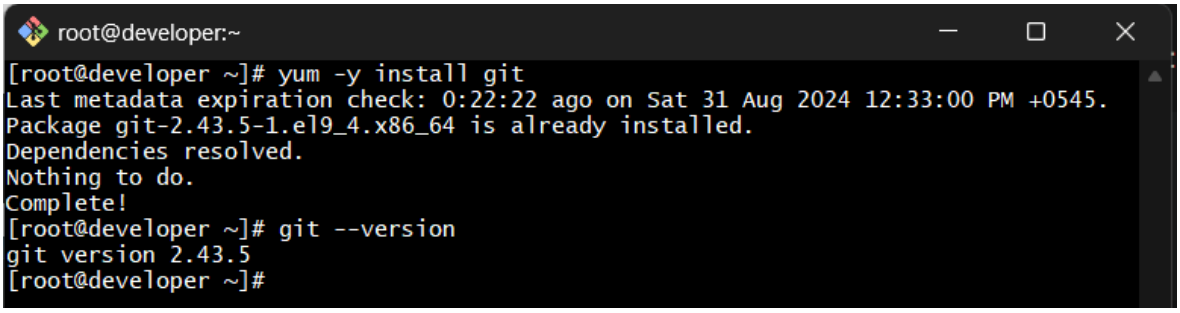


- Create the first admin user.
- Start using Jenkins!



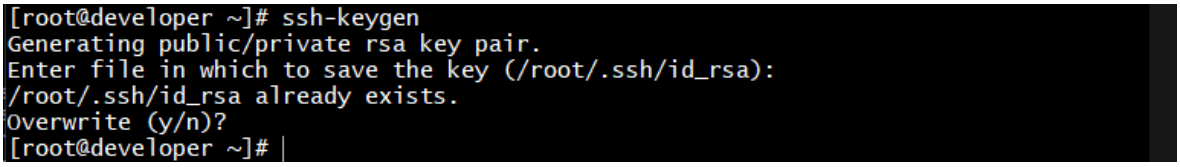
Step 2: Git installation and configuration in Developer and Jenkins Machine

a. Installing Git in Developer Machine

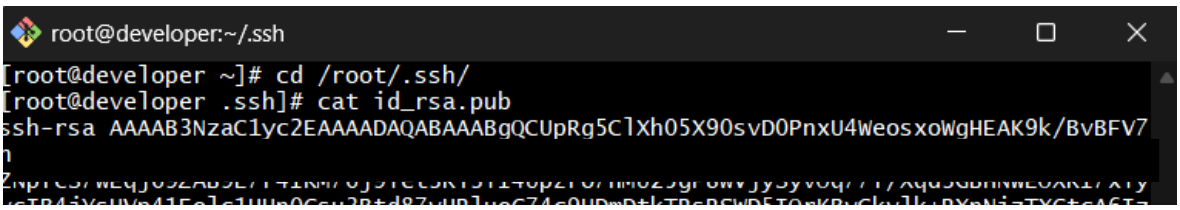


b. Configuring SSH Authentication with GitHub

- Generate an SSH key pair

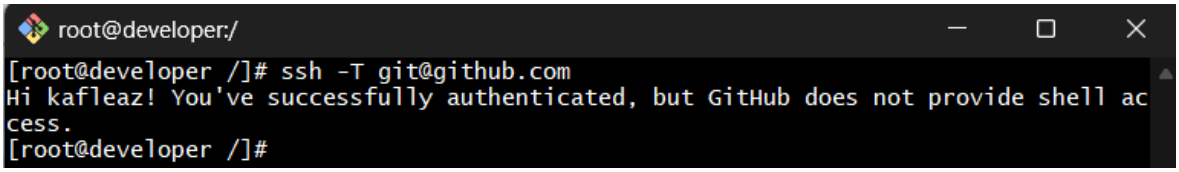


- Navigate to the SSH directory and display the public key.



- Copy the displayed key and add it to GitHub:

- Test the SSH connection:



Note: Follow the same process in Jenkins machine to configure Git.

Step 3: Set up Apache Webserver

a. Installing Apache Server and PHP

```
root@webserver:~
[root@webserver ~]# yum -y install httpd php
Last metadata expiration check: 0:02:17 ago on Wed 28 Aug 2024 12:26:40 PM +0545.
Package httpd-2.4.57-11.el9_4.1.x86_64 is already installed.
Package php-8.0.30-1.el9_2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@webserver ~]#
```

b. Start and Enable the Apache service to Start on Boot.

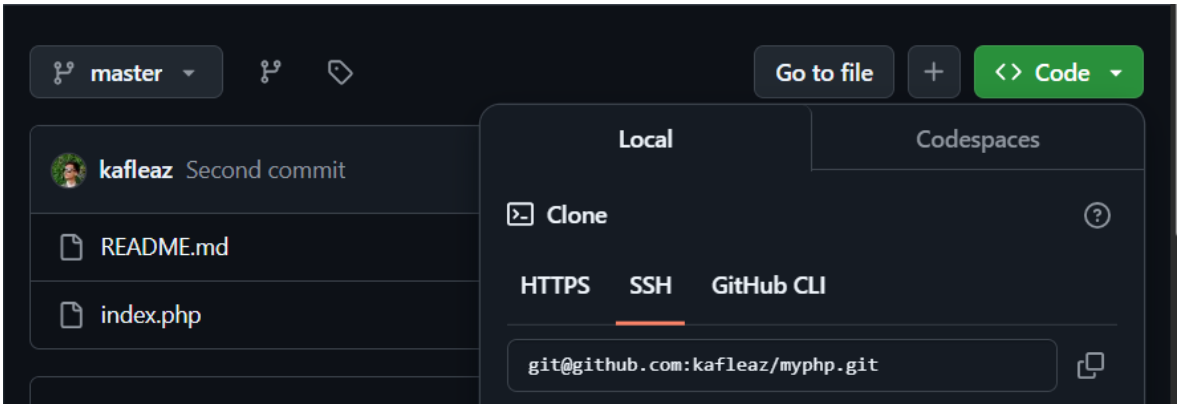
```
[root@webserver ~]# systemctl start httpd
[root@webserver ~]# systemctl enable httpd
```

c. Configure Firewall to Allow HTTP Traffic:

```
[root@webserver ~]# systemctl start httpd
[root@webserver ~]# systemctl enable httpd
[root@webserver ~]# firewall-cmd --permanent --add-service=http
Warning: ALREADY_ENABLED: http
success
[root@webserver ~]# firewall-cmd --reload
success
[root@webserver ~]#
```

Step 4: Creating a Local Git Repository

a. Create a new repository named "myphp" in github and copy the URL.



b. Create a project workspace directory:

```
[root@developer myphp]# mkdir -p /project/myphp
[root@developer myphp]# cd /project/myphp/
```

c. Initialize, Link and pull a Git repository:

```
root@developer:/project/myphp
[root@developer myphp]# git init
Reinitialized existing Git repository in /project/myphp/.git/
[root@developer myphp]# git remote add origin git@github.com:kafleaz/myphp.git
[root@developer myphp]# git pull origin master
remote: Enumerating objects: 12, done.
remote: Counting objects: 100% (12/12), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 12 (delta 0), reused 9 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (12/12), 1.77 KiB | 78.00 KiB/s, done.
From github.com:kafleaz/myphp
 * branch            master     -> FETCH_HEAD
 * [new branch]      master     -> origin/master
```

d. Create a new file and add content.

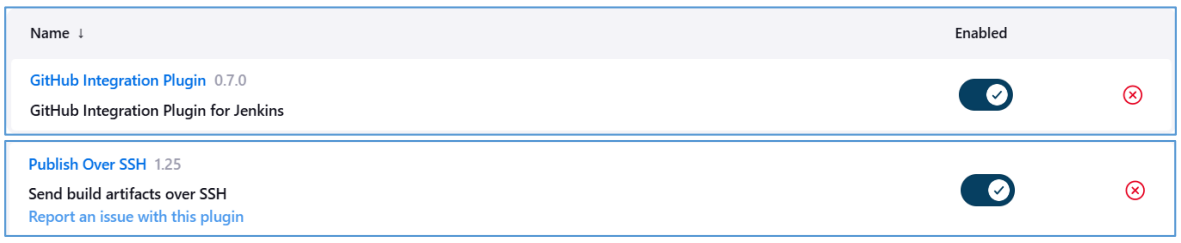
```
[root@developer myphp]# vi index.php
[root@developer myphp]# git add .
[root@developer myphp]# git commit -m "myphp firstt commit"
[master 60b2008] myphp firstt commit
 Committer: root <root@developer.az.com>
Your name and email address were configured automatically based
```

```
[root@developer myphp]# git push origin master
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 331 bytes | 165.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To github.com:kafleaz/myphp.git
   997a8ef..60b2008  master -> master
[root@developer myphp]#
```

Step 5: jenkins configuration for "Github" and "publish over ssh"

a. Install Required Plugins in Jenkins

- Navigate to Manage **Jenkins** > **Plugins** > **Available Plugins** > Search plugins
- Select it and click **Install without restart**.



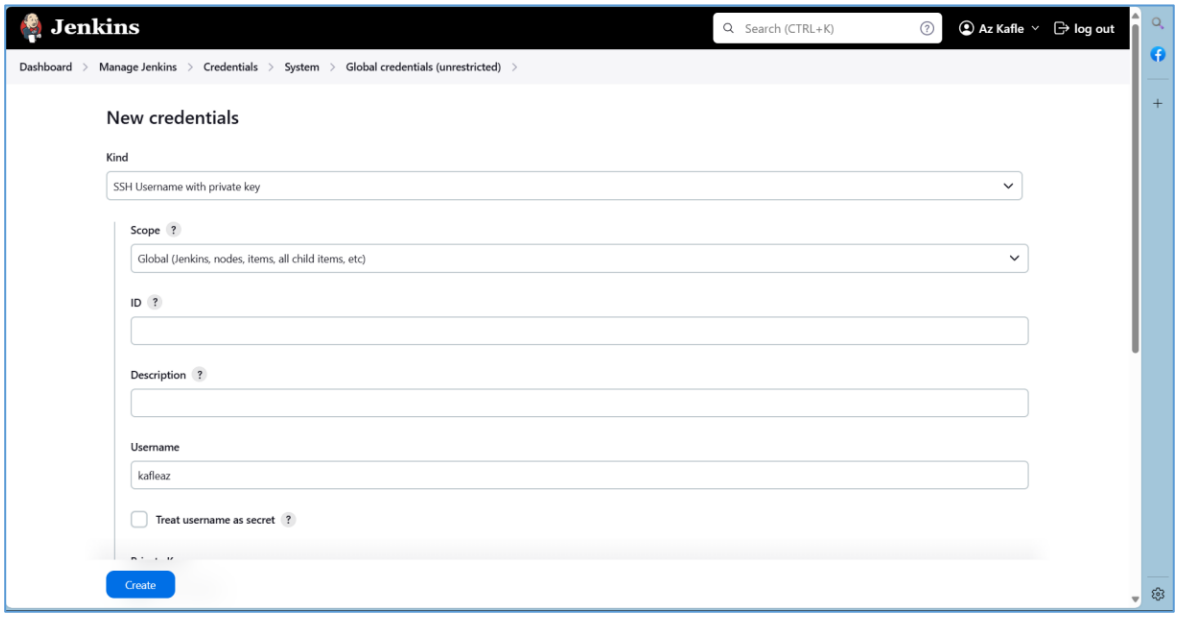
b. jenkins and Github ssh authentication configuration

- Display the private key

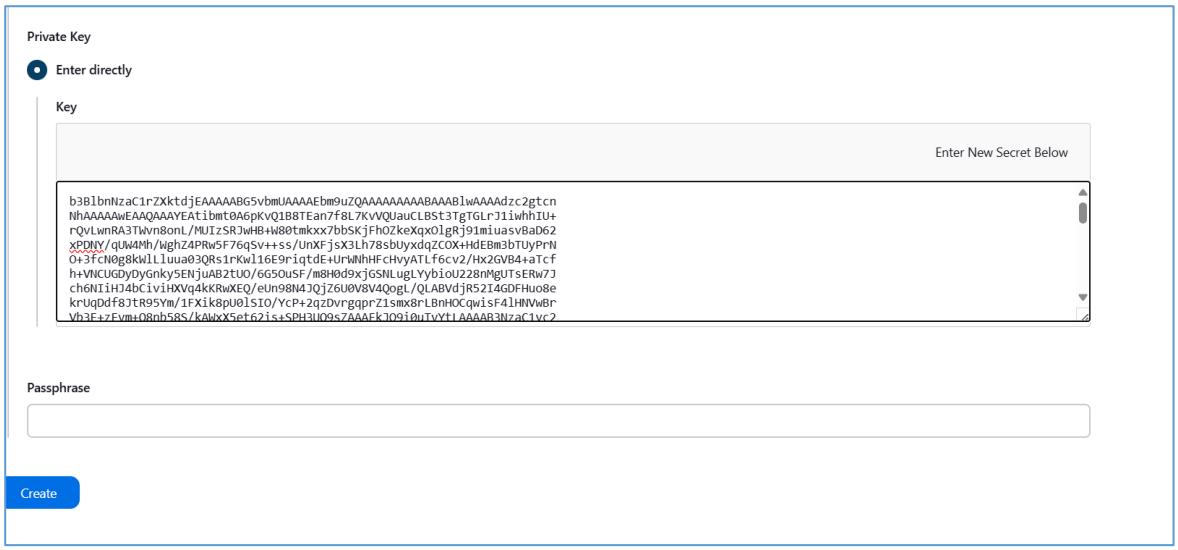
```
[root@jenkins /]# cd /root/.ssh/
[root@jenkins .ssh]# cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAtibmtOA6pKvQ1B8TEan7f8L7KvVQUauCLBSt3TgTGLrJ1iwhhIU+
rQvLwnRA3TWvn8onL/MUIzSRJwHB+W80tmkxx7bbSKjFhOZkeXqxOlgRj91miuasvBaD62
```

- In Jenkins, navigate to **Manage Jenkins** > **Credentials** > **System** > **Global credentials**.

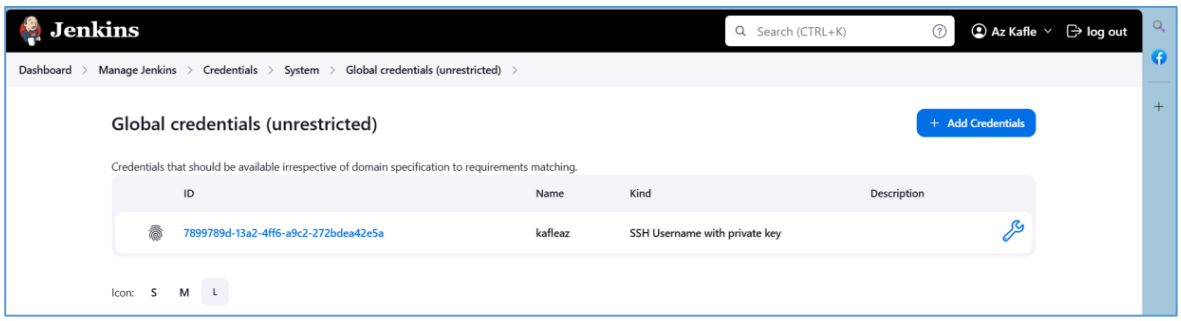
- Click **Add Credentials**.
- Enter your **GitHub username**.



- Paste the copied private key.



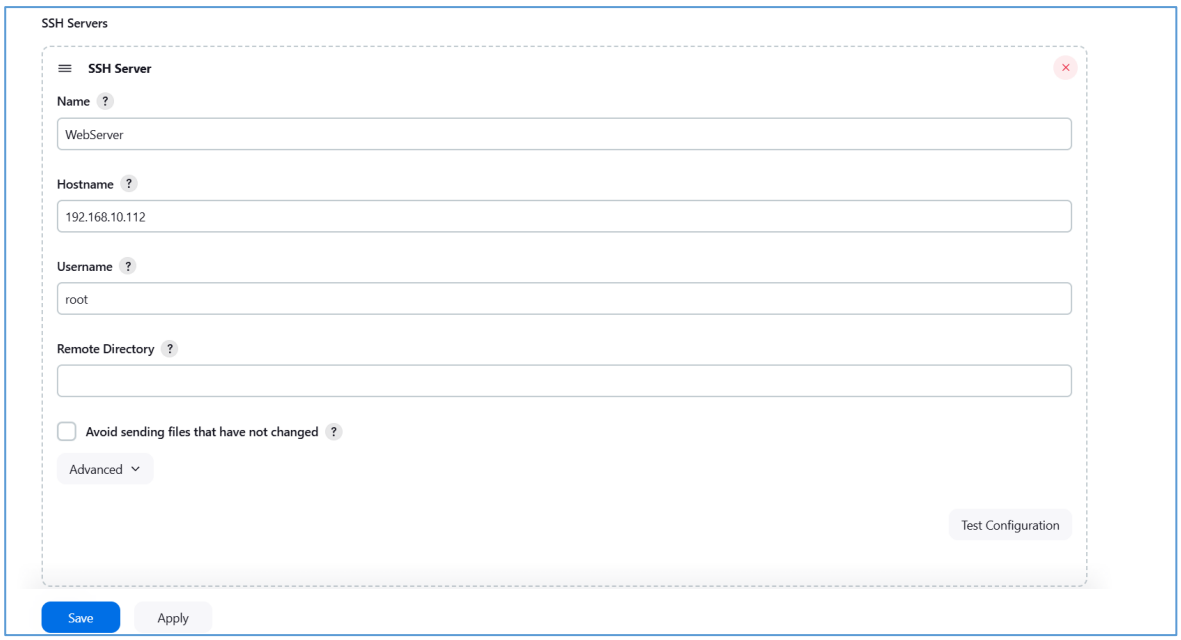
- Click **Create**.



Step 6: Set IP address, username and password of webserver in jenkins for automated file transfer (publish over ssh)

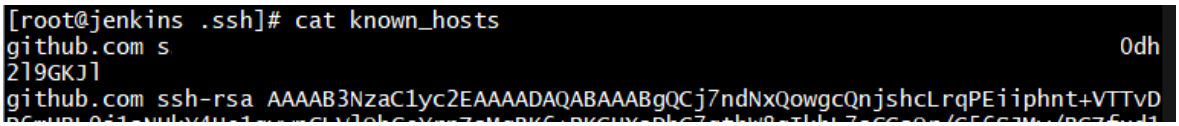
a. In Jenkins, navigate to **Manage Jenkins > Configure system > System > Publish Over SSH > SSH Servers**

- Enter SSH Server Details

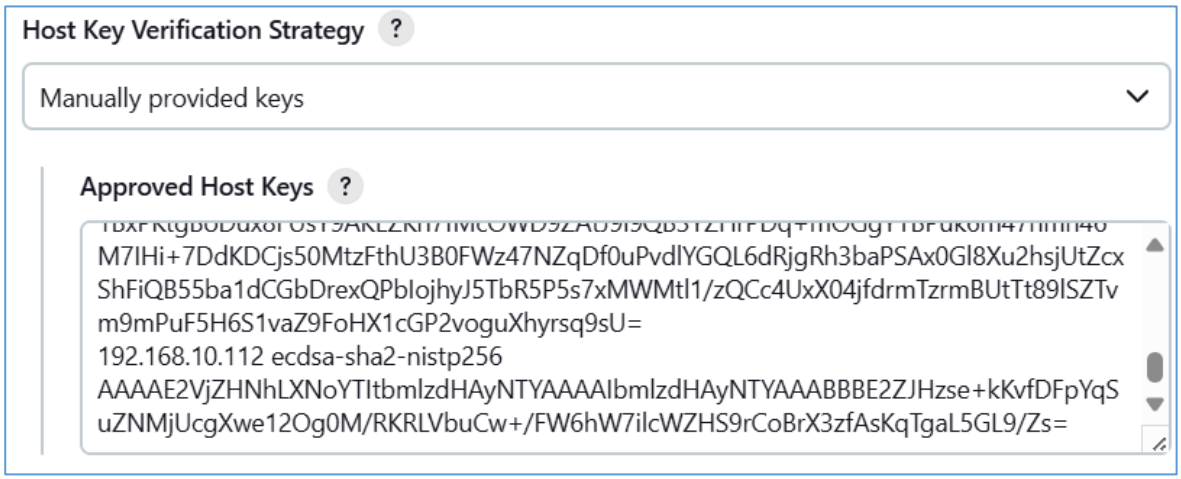


b. In Jenkins, navigate to **Manage Jenkins > Security > Git Host Key Verification Configuration**

- Obtain the known host key of Jenkins machine



- Manually provided key

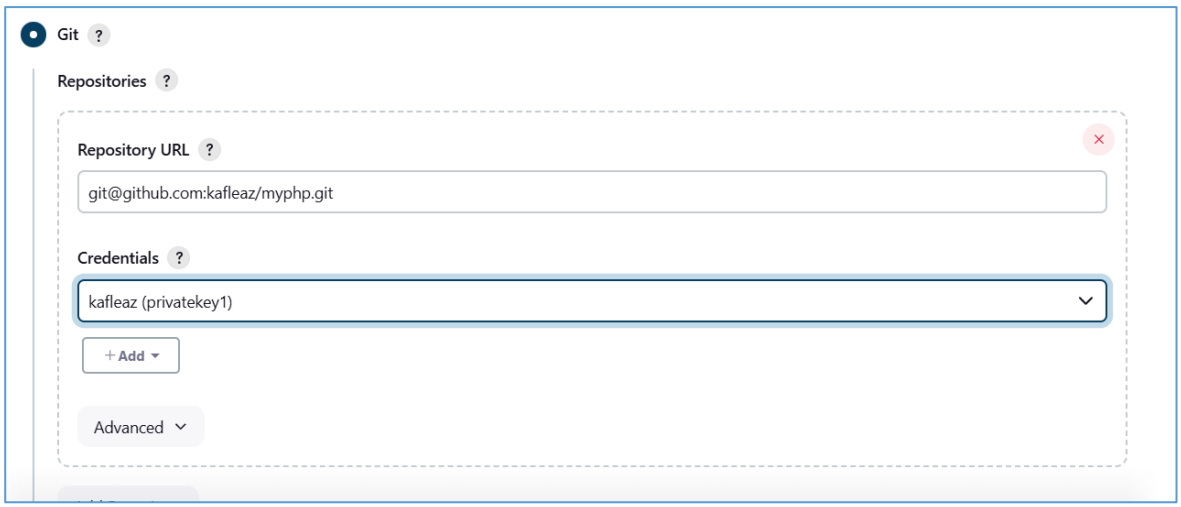


- **Save** Configuration:

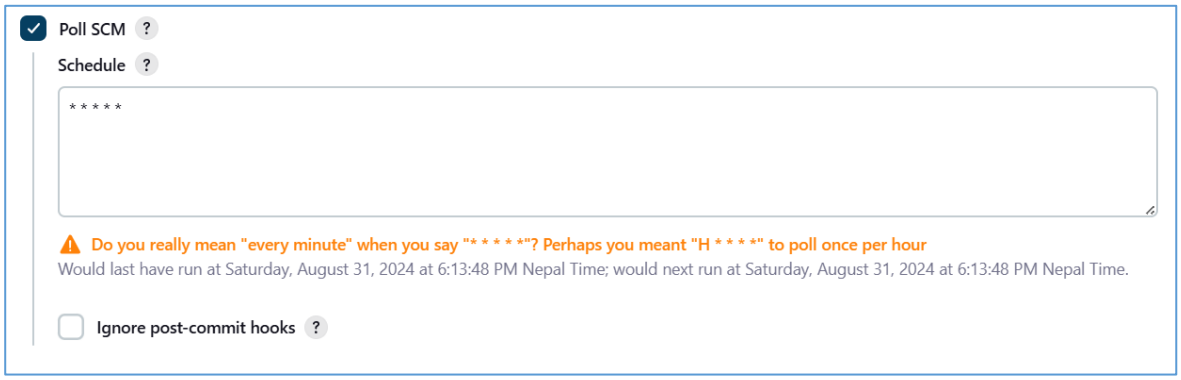
Step 7: Createing a job in jenkins to pull index.html from github repo. and send it to the webserver into /var/www/html directory

a. In Jenkins, navigate to **Jenkins Dashboard > New item >** enter item name **(MyPhp_P2) > Freestyle project > ok**

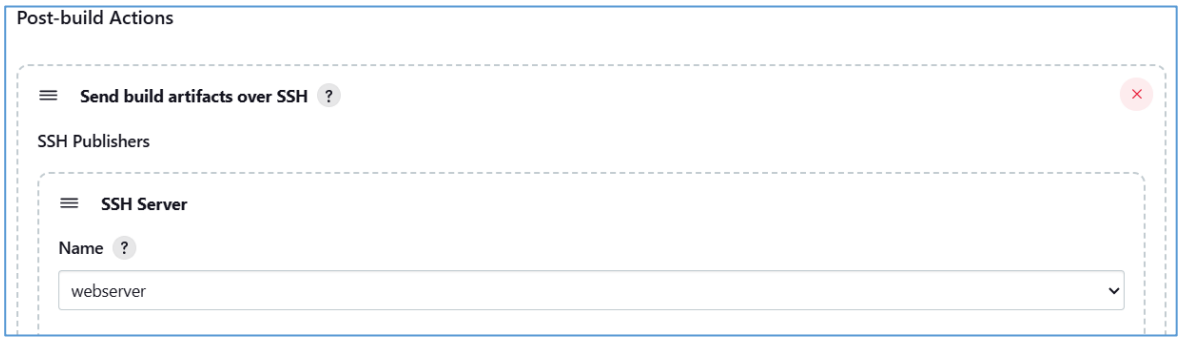
- Select **Git**, add **repo url** and **credentials**.



- **Build Trigger** > **Poll SCM**



- **post build Action > Send build artifact over ssh**



Step 8: Test the Deployment

- Remove index.html from /var/www/html/ from WebServer.

```
index.html
[root@webserver html]# rm -f index.html
```

- Check Console Output:

```
Started by user Az Kafle
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/MyPhp_P2
The recommended git tool is: NONE
using credential d5c8b220-5e76-406c-a00b-d317319da893
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/MyPhp_P2/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url git@github.com:kafleaz/myphp.git # timeout=10
Fetching upstream changes from git@github.com:kafleaz/myphp.git
 > git --version # timeout=10
 > git --version # 'git version 2.43.5'
using GIT_SSH to set credentials privatekey1
[INFO] Currently running in a labeled security context
[INFO] Currently SELinux is 'enforcing' on the host
 > /usr/bin/chcon --type=ssh_home_t /var/lib/jenkins/workspace/MyPhp_P2@tmp/jenkins-gitclient-ssh14916015825130873086.key
Verifying host key using manually-configured host key entries
 > git fetch --tags --force --progress -- git@github.com:kafleaz/myphp.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision 60b2008915ae3552c9233b54f6f0116cbcb11ea2 (refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f 60b2008915ae3552c9233b54f6f0116cbcb11ea2 # timeout=10
Commit message: "myphp firstt commit"
 > git rev-list --no-walk 60b2008915ae3552c9233b54f6f0116cbcb11ea2 # timeout=10
SSH: Connecting from host [jenkins.az.com]
SSH: Connecting with configuration [webserver] ...
SSH: Disconnecting configuration [webserver] ...
SSH: Transferred 1 file(s)
Finished: SUCCESS
```

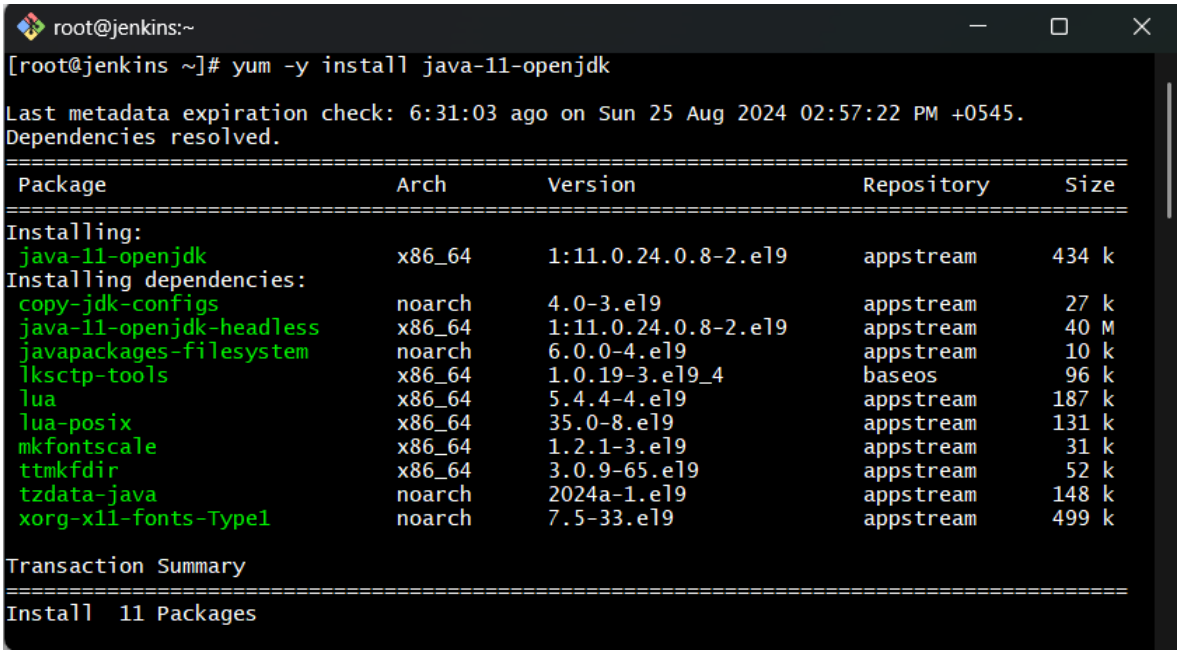
- Verify Deployment

Project 3

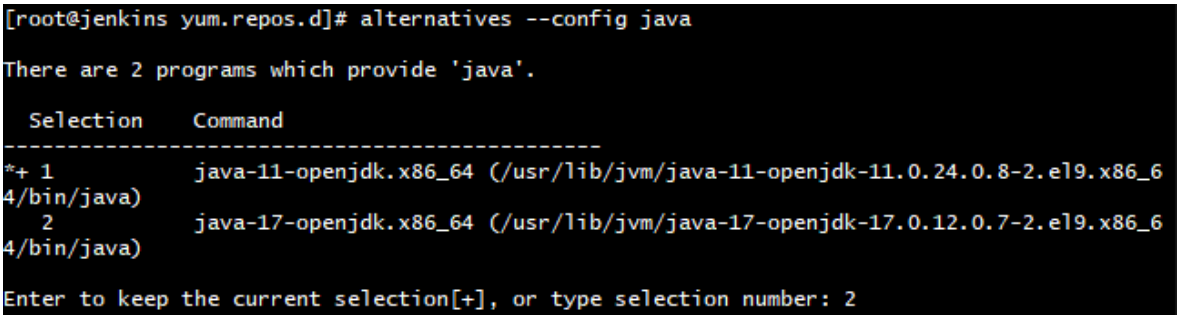
Setup CI/CD with GitHub, Jenkins, Apache Webserver to deploy an artifact (Python based website).

Step 1: Set up Jenkins

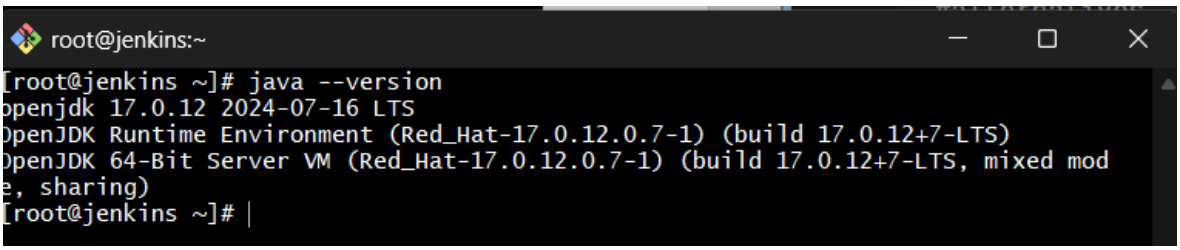
a. Install Java



- If you have an old version of JDK installed, configure the alternative Java version



- Select the appropriate Java version and verify:



- Set the JAVA_HOME environment variable:

```
root@jenkins:/usr/lib/jvm/java-17-openjdk-17.0.12.0.7-2.el9.x86_64
[root@jenkins ~]# cd /usr/lib/jvm/java-17-openjdk-17.0.12.0.7-2.el9.x86_64/
[root@jenkins java-17-openjdk-17.0.12.0.7-2.el9.x86_64]# pwd
/usr/lib/jvm/java-17-openjdk-17.0.12.0.7-2.el9.x86_64
[root@jenkins java-17-openjdk-17.0.12.0.7-2.el9.x86_64]#
```

- Copy the output path and add it to the .bash_profile:

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi
JAVA_HOME=/usr/lib/jvm/java-17-openjdk-17.0.12.0.7-2.el9.x86_64
PATH=$PATH:$HOME/bin:$JAVA_HOME
# User specific environment and startup programs
~
```

- Refresh the environment:

```
[root@jenkins yum.repos.d]# . /root/.bash_profile
[root@jenkins yum.repos.d]#
```

a. Install Jenkins

```
root@jenkins:/etc/yum.repos.d
[root@jenkins /]# cd /etc/yum.repos.d/
[root@jenkins yum.repos.d]#
```

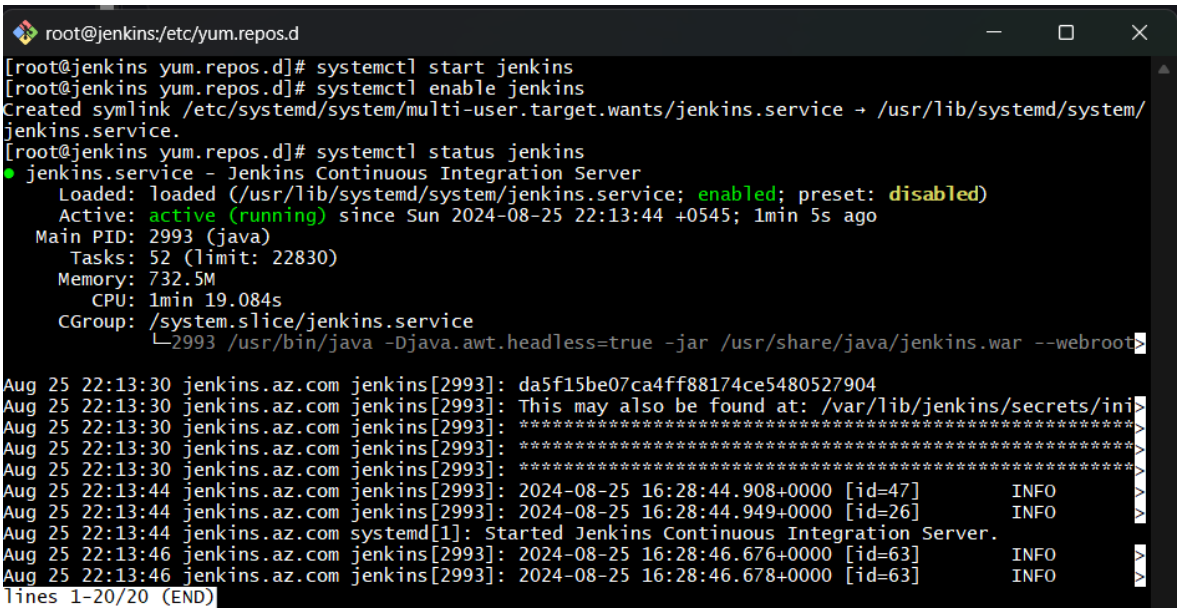
- Browse the official Jenkins website for the genuine Jenkins repository URL and key.

```
[root@jenkins yum.repos.d]# wget https://pkg.jenkins.io/redhat-stable/jenkins.repo
--2024-08-25 22:07:45--  https://pkg.jenkins.io/redhat-stable/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io)... 199.232.22.133, 2a04:4e42:42::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|199.232.22.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 85
Saving to: 'jenkins.repo'

jenkins.repo              100%[===================================>]      85  --.-KB/s    in 0s

2024-08-25 22:07:45 (1.92 MB/s) - 'jenkins.repo' saved [85/85]

[root@jenkins yum.repos.d]# rpm --import https://pkg.jenkins.io/redhat/jenkins.io-2023.key
[root@jenkins yum.repos.d]# yum upgrade
Jenkins-stable                                                  16 kB/s |  29 kB     00:01
Dependencies resolved.
Nothing to do.
Complete!
```

```
[root@jenkins yum.repos.d]# yum install jenkins
Last metadata expiration check: 0:00:19 ago on Sun 25 Aug 2024 10:08:35 PM +0545.
Dependencies resolved.
================================================================================
 Package              Architecture         Version                  Repository              Size
================================================================================
Installing:
 jenkins              noarch               2.462.1-1.1              jenkins                 89 M

Transaction Summary
================================================================================
```

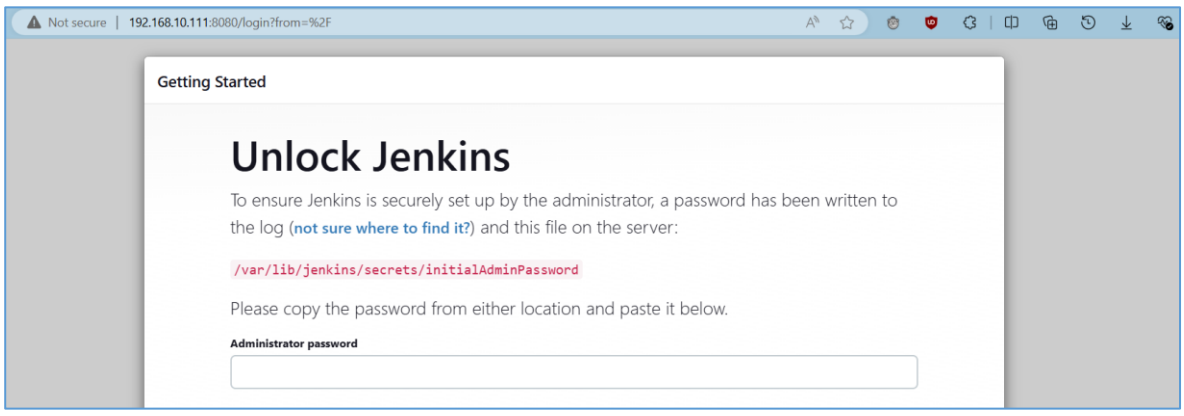
```
[root@jenkins yum.repos.d]# rpm -q jenkins
jenkins-2.462.1-1.1.noarch
```

- Start the Jenkins service:

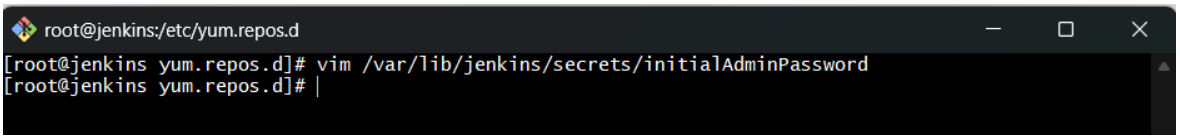


b. Add the Jenkins port to the firewall for global access:

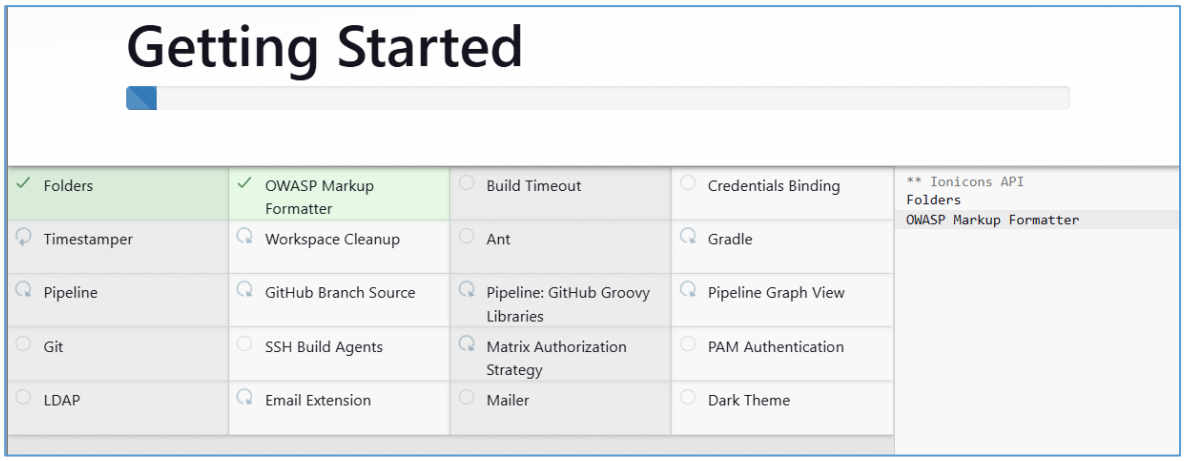
c. Access the Jenkins dashboard:



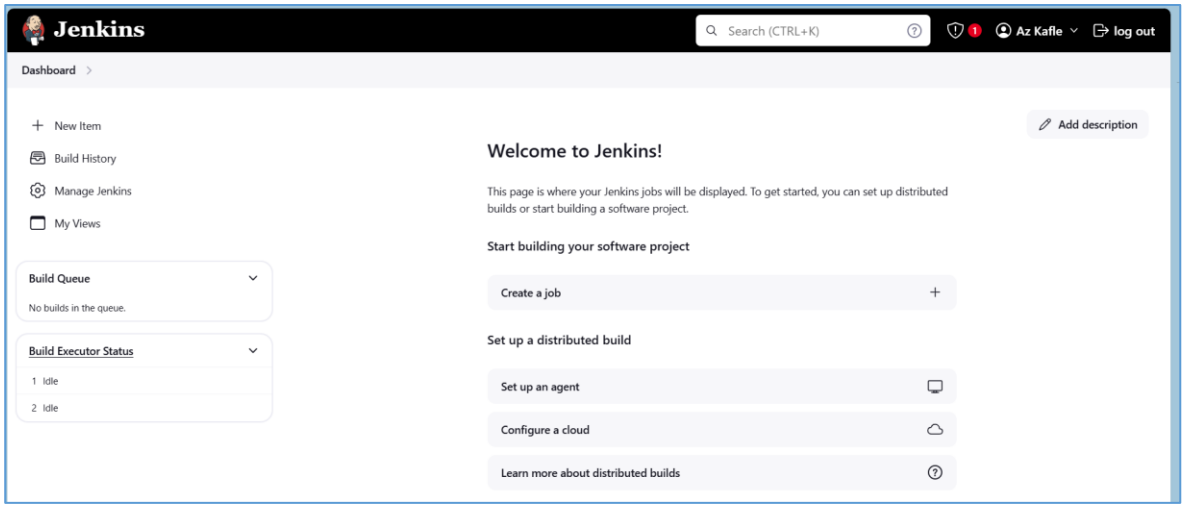
- Follow the instructions to unlock Jenkins using the password provided in the specified location.



- Install the suggested plugins.

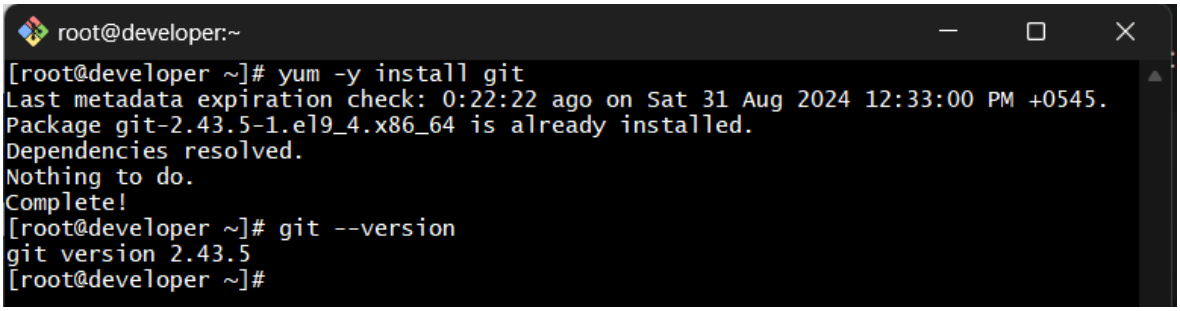


- Create the first admin user.
- Start using Jenkins!



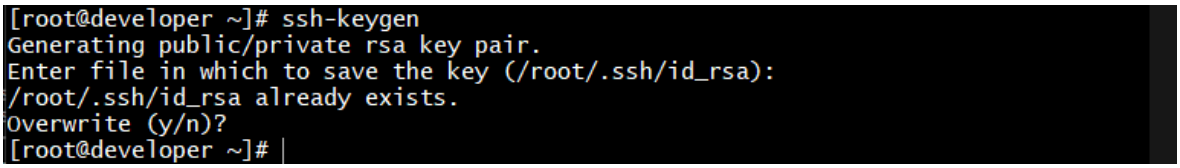
Step 2: Git installation and configuration in Developer and Jenkins Machine

a. Installing Git in Developer Machine

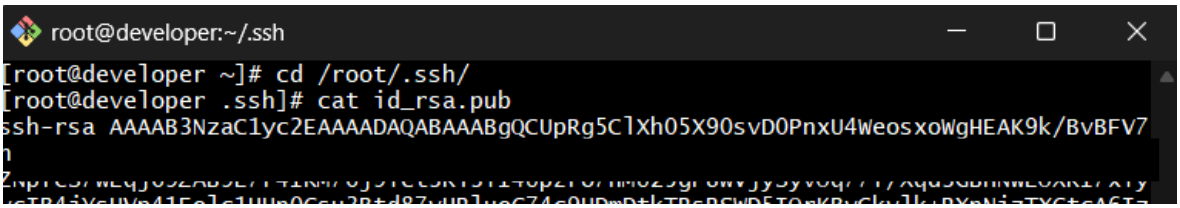


b. Configuring SSH Authentication with GitHub

- Generate an SSH key pair

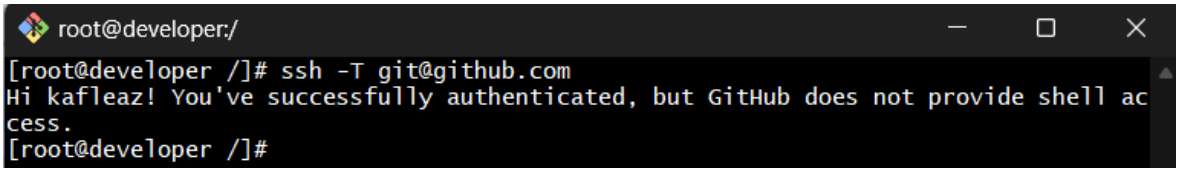


- Navigate to the SSH directory and display the public key.



- Copy the displayed key and add it to GitHub:

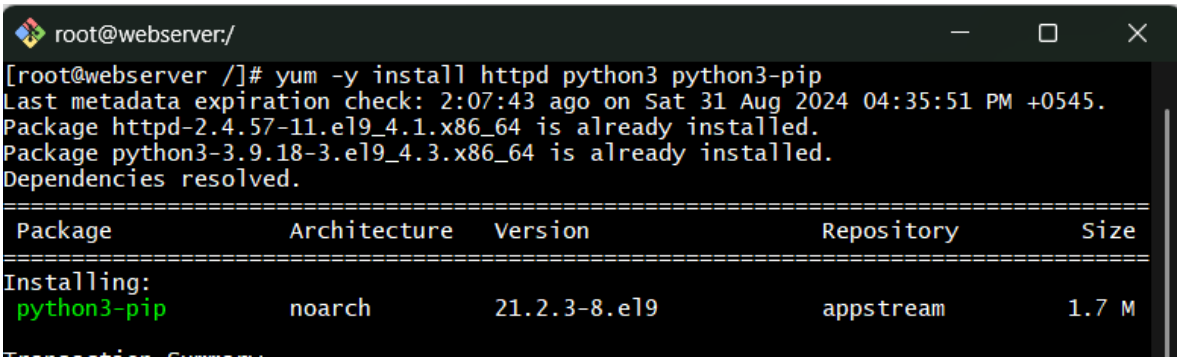
- Test the SSH connection:



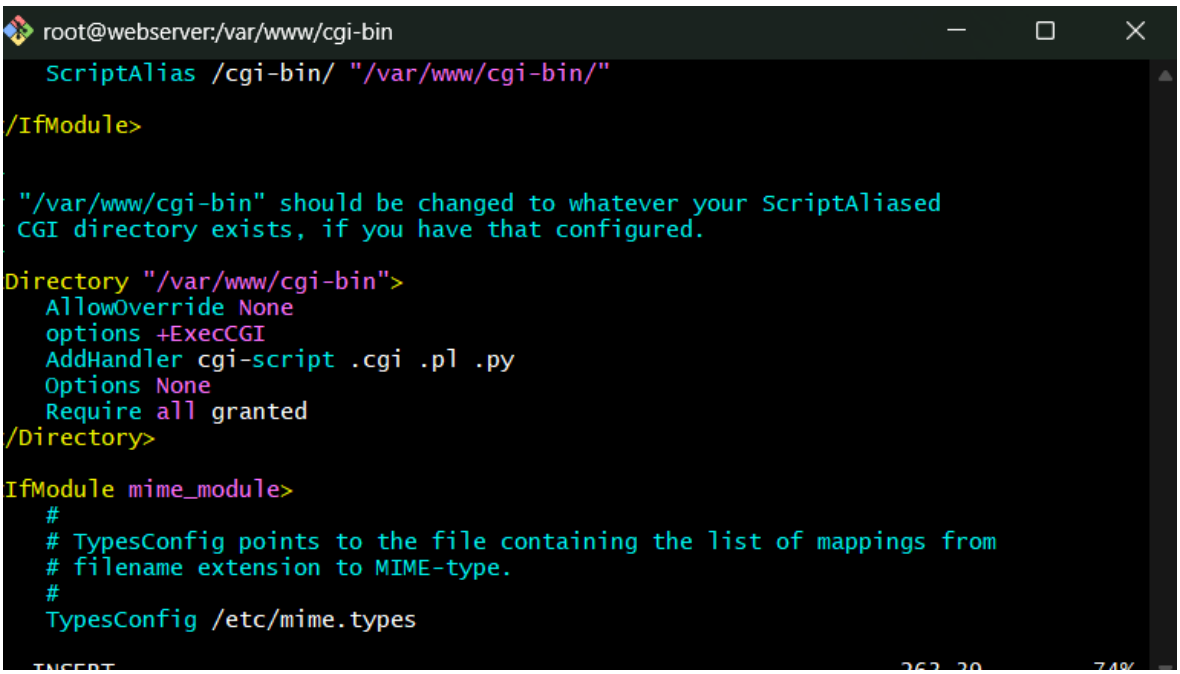
Note: Follow the same process in Jenkins machine to configure Git.

Step 3: Set up Apache Webserver

a. Installing Apache Server and python packages

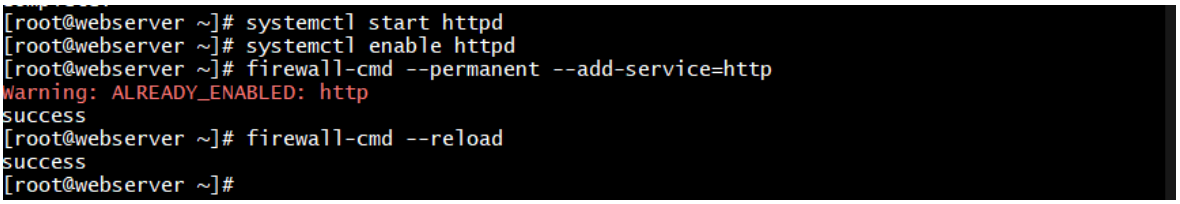


b. Edit CGI scripts



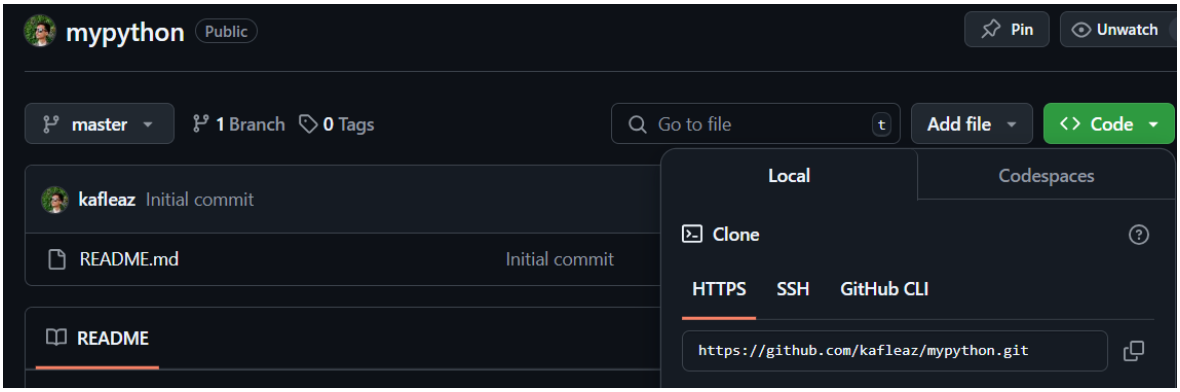
c. Start and Enable the Apache service to Start on Boot.

d. Configure Firewall to Allow HTTP Traffic:



Step 4: Creating a Local Git Repository

a. Create a new repository named "mypython" in github and copy the URL.



b. Create a project workspace directory:

```
[root@developer /]# mkdir -p /project/mypython
[root@developer /]# cd /project/mypython
```

c. Initialize, Link and pull a Git repository:

```
[root@developer project]# git init
Initialized empty Git repository in /project/.git/
[root@developer project]# git remote add origin git@github.com:kafleaz/mypython.
git
[root@developer project]# git pull origin master
```

d. Create a new file and add content.

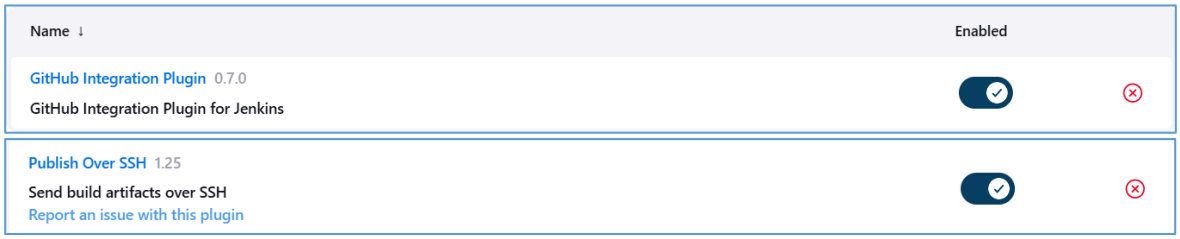
```
[root@developer mypython]# git init
Initialized empty Git repository in /project/mypython/.git/
[root@developer mypython]# git remote add origin git@github.com:kafleaz/mypython
.git
[root@developer mypython]# git pull origin master
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (3/3), 859 bytes | 122.00 KiB/s, done.
From github.com:kafleaz/mypython
 * branch            master     -> FETCH_HEAD
 * [new branch]      master     -> origin/master
```

```
[root@developer mypython]# vi hello.py
[root@developer mypython]# git add .
[root@developer mypython]# git commit -m "changed to hello.py"
On branch master
nothing to commit, working tree clean
[root@developer mypython]# git push origin master
Everything up-to-date
[root@developer mypython]#
```

Step 5: jenkins configuration for "Github" and "publish over ssh"

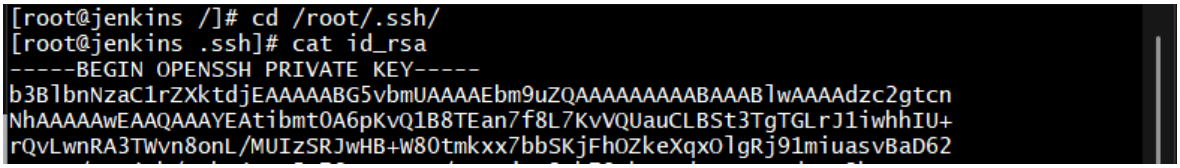
a. Install Required Plugins in Jenkins

- Navigate to Manage **Jenkins** > **Plugins** > **Available Plugins** > Search plugins
- Select it and click **Install without restart**.



b. jenkins and Github ssh authentication configuration

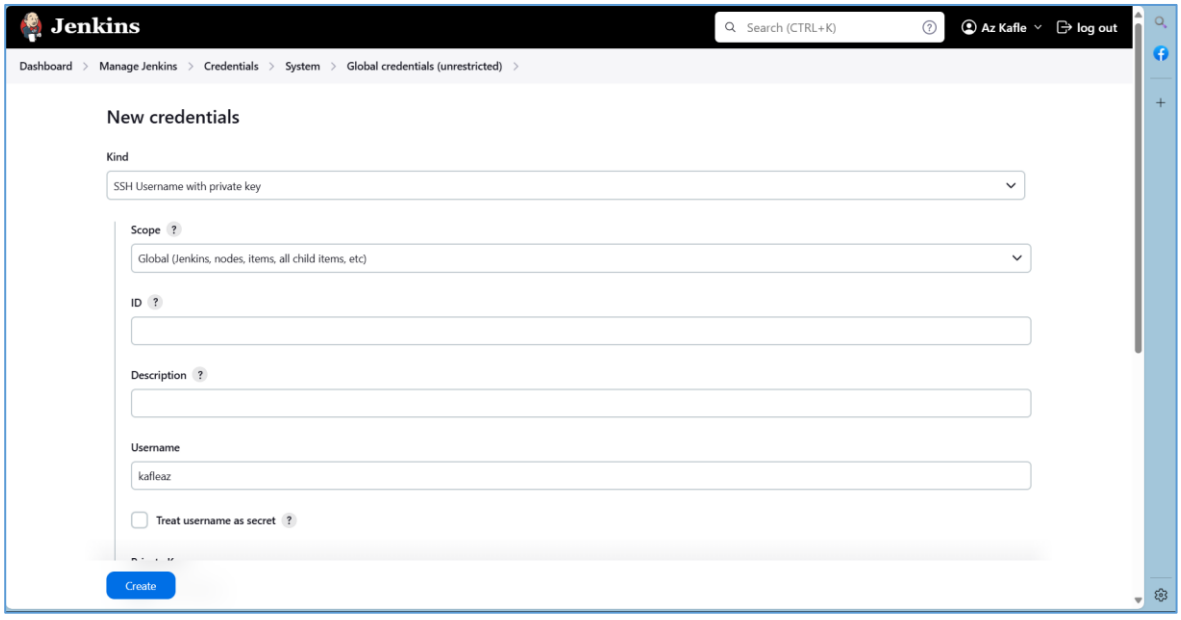
o Display the private key



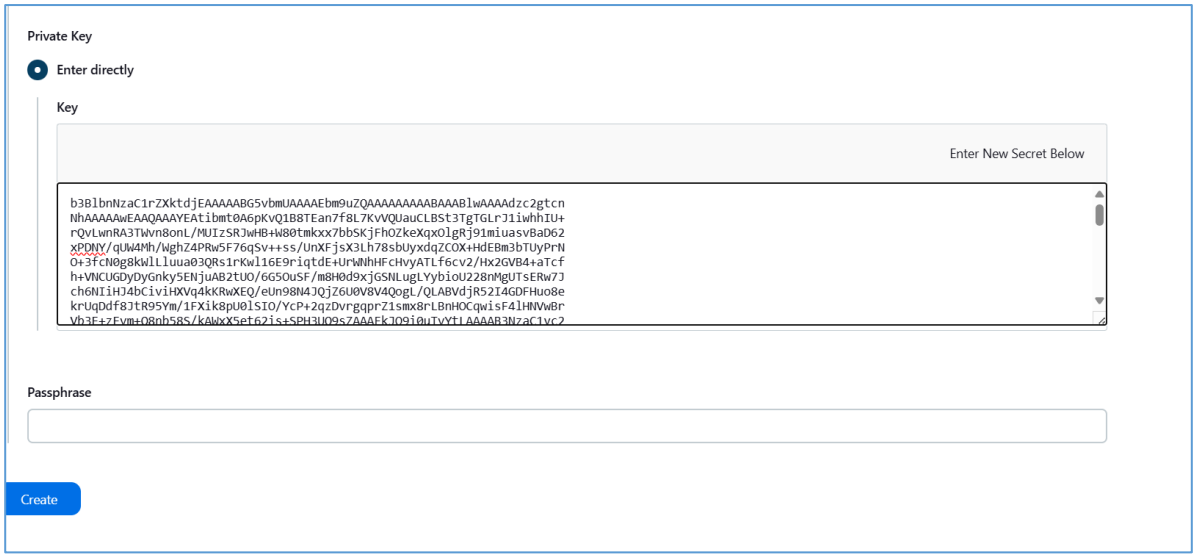
o In Jenkins, navigate to **Manage Jenkins** > **Credentials** > **System** > **Global credentials**.

o Click **Add Credentials**.

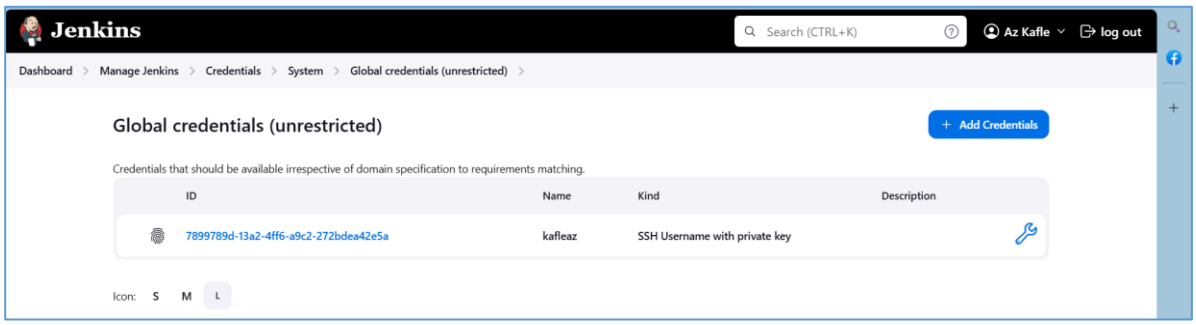
o Enter your **GitHub username**.



o Paste the copied private key.



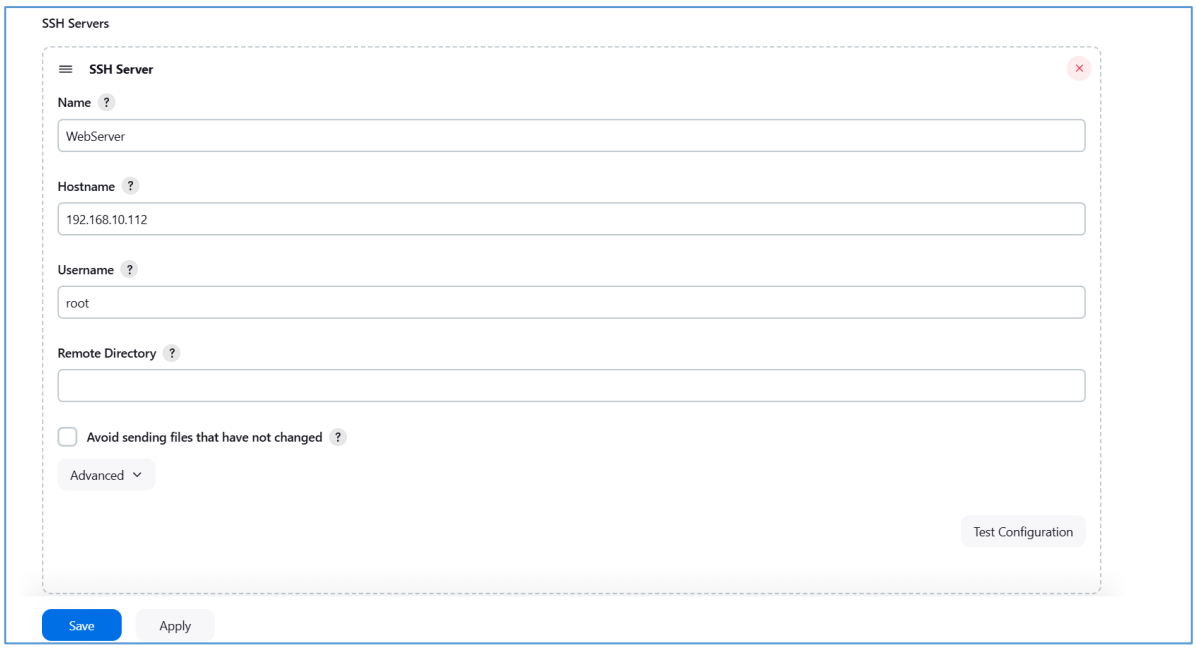
- Click **Create**.



Step 6: Set IP address, username and password of webserver in jenkins for automated file transfer (publish over ssh)

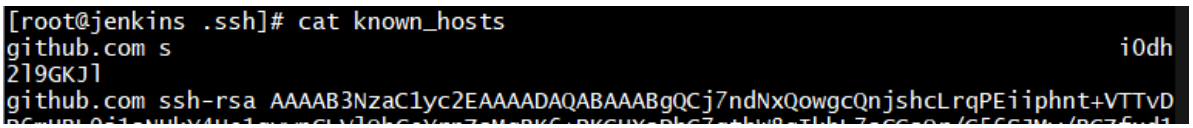
c. In Jenkins, navigate to **Manage Jenkins > Configure system > System > Publish Over SSH > SSH Servers**

- Enter SSH Server Details

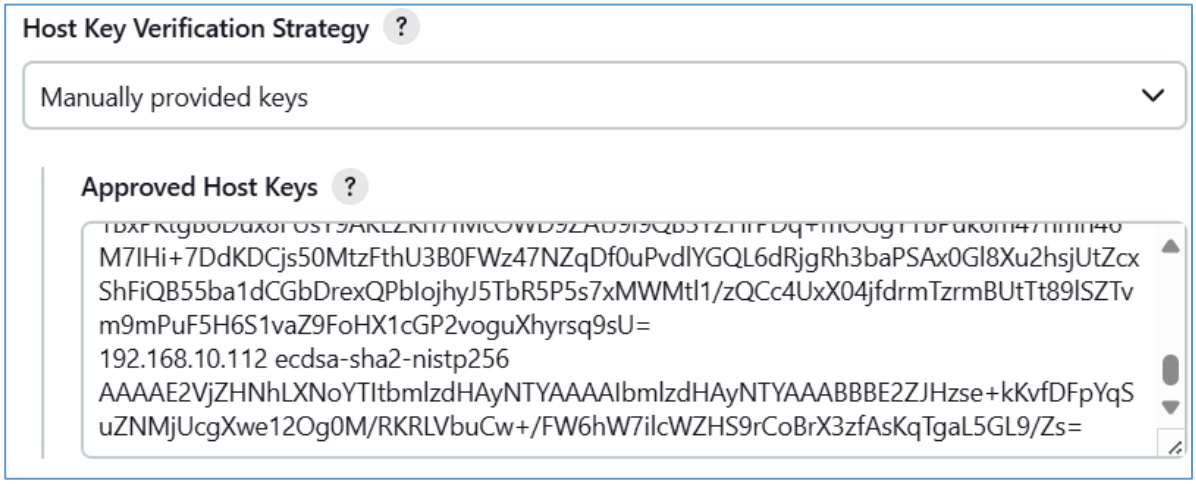


d. In Jenkins, navigate to **Manage Jenkins > Security > Git Host Key Verification Configuration**

- Obtain the known host key of Jenkins machine



- Manually provided key

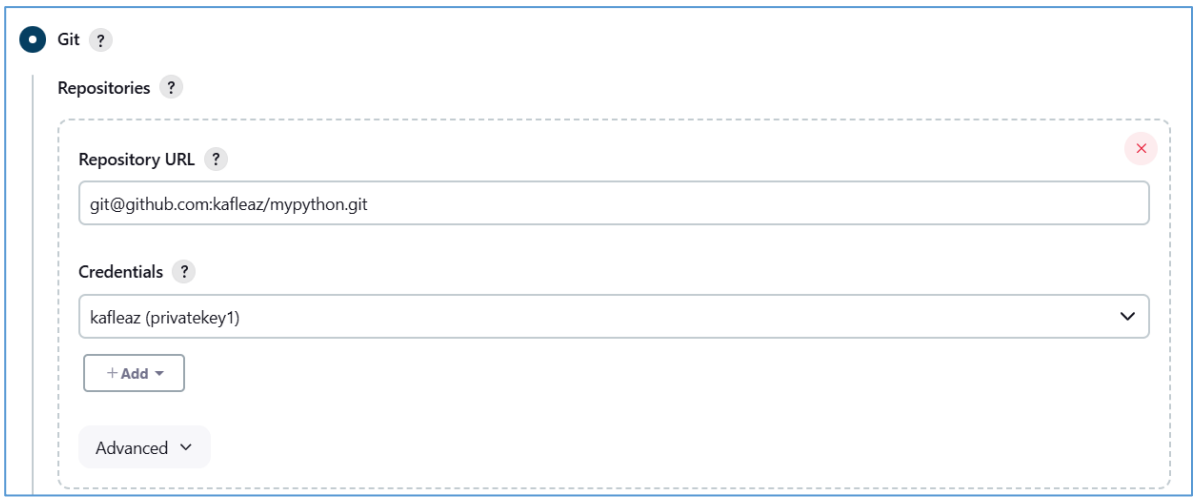


- **Save** Configuration:

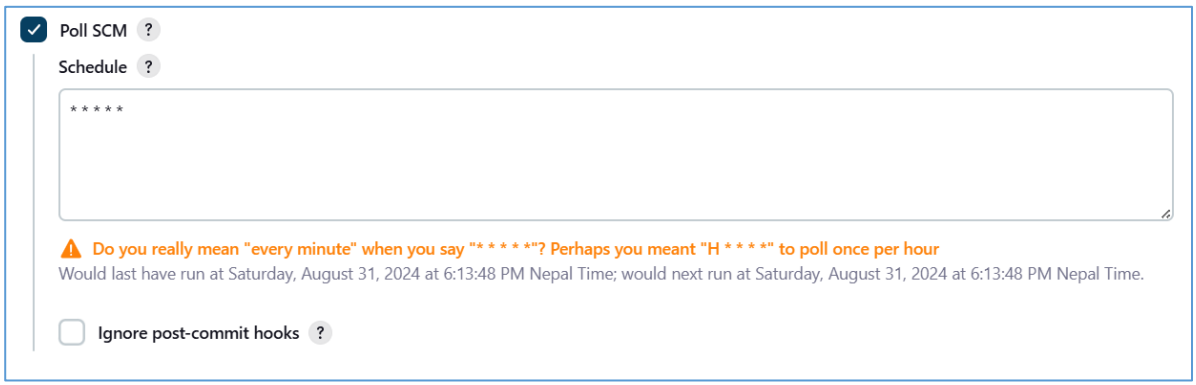
Step 7: Createing a job in jenkins to pull index.html from github repo. and send it to the webserver into /var/www/html directory

a. In Jenkins, navigate to **Jenkins Dashboard > New item >** enter item name **(mypython_P3) > Freestyle project > ok**

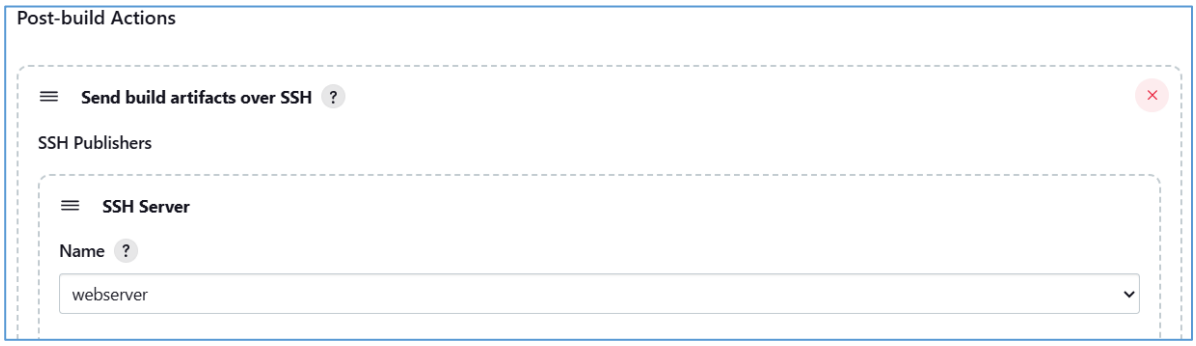
- Select **Git**, add **repo url** and **credentials**.

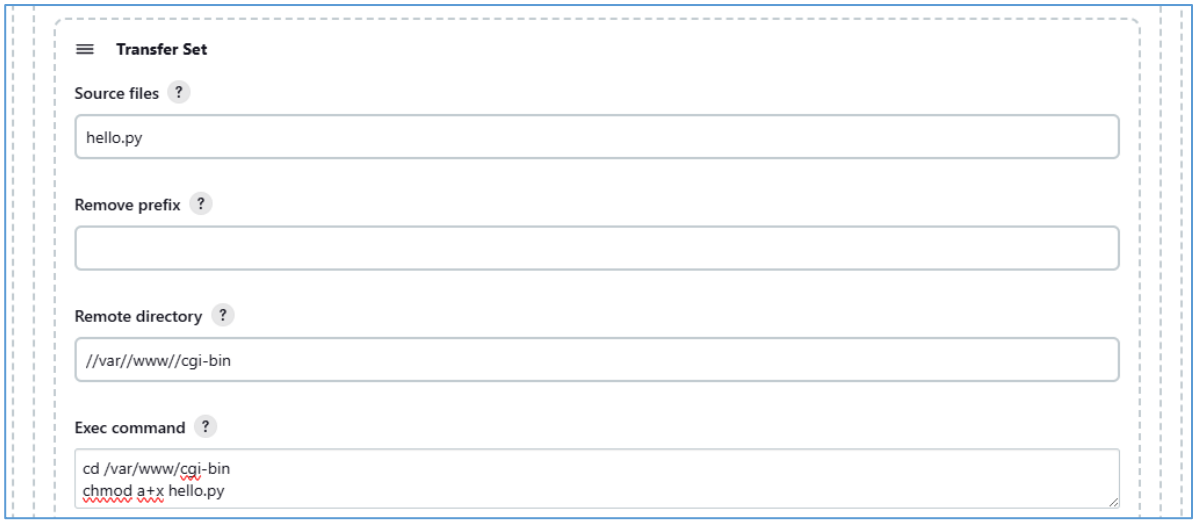


- **Build Trigger > Poll SCM**



- **post build Action > Send build artifact over ssh**





Step 8: Test the Deployment

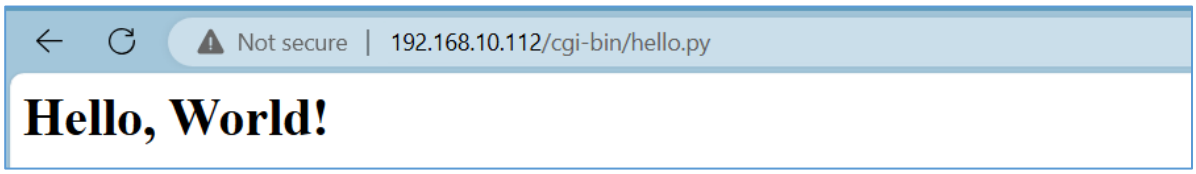
- verify if the script executable in WebServer.

```
[root@webserver cgi-bin]# ls -l
total 4
-rwxr-xr-x. 1 root root 245 Sep  8 22:42 hello.py
```

- Check Console Output in jenkins:

```
Started by user Az Kafle
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/mypython_P3
The recommended git tool is: NONE
using credential d5c8b220-5e76-406c-a00b-d317319da893
 > git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/mypython_P3/.git # timeout=10
Fetching changes from the remote Git repository
 > git config remote.origin.url git@github.com:kafleaz/mypython.git # timeout=10
Fetching upstream changes from git@github.com:kafleaz/mypython.git
 > git --version # timeout=10
 > git --version # 'git version 2.43.5'
using GIT_SSH to set credentials privatekey1
[INFO] Currently running in a labeled security context
[INFO] Currently SELinux is 'enforcing' on the host
 > /usr/bin/chcon --type=ssh_home_t /var/lib/jenkins/workspace/mypython_P3@tmp/jenkins-gitclient-ssh13614357653397722388.key
Verifying host key using manually-configured host key entries
 > git fetch --tags --force --progress -- git@github.com:kafleaz/mypython.git +refs/heads/*:refs/remotes/origin/* # timeout=10
 > git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision e9141fe3b3aeac1d48f4ee77d02607cda1b03f8d (refs/remotes/origin/master)
 > git config core.sparsecheckout # timeout=10
 > git checkout -f e9141fe3b3aeac1d48f4ee77d02607cda1b03f8d # timeout=10
Commit message: "changed to hello.py"
 > git rev-list --no-walk e9141fe3b3aeac1d48f4ee77d02607cda1b03f8d # timeout=10
SSH: Connecting from host [jenkins.az.com]
SSH: Connecting with configuration [webserver] ...
SSH: Disconnecting configuration [webserver] ...
SSH: Transferred 1 file(s)
Finished: SUCCESS
```

- Verify Deployment

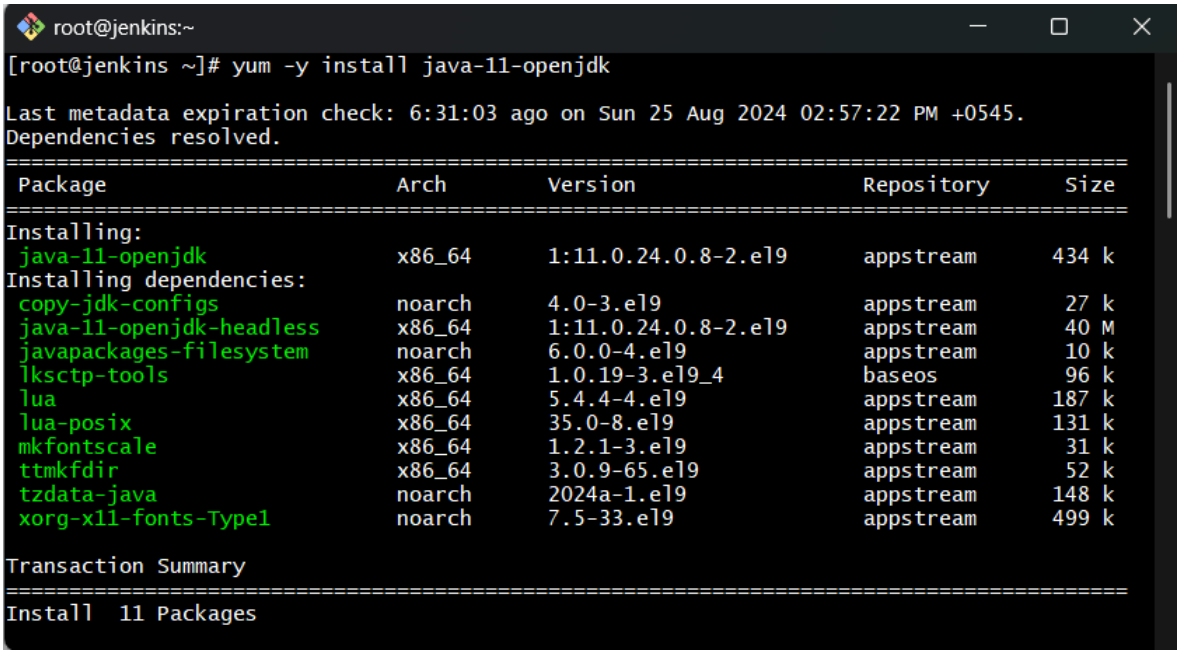


Project 4

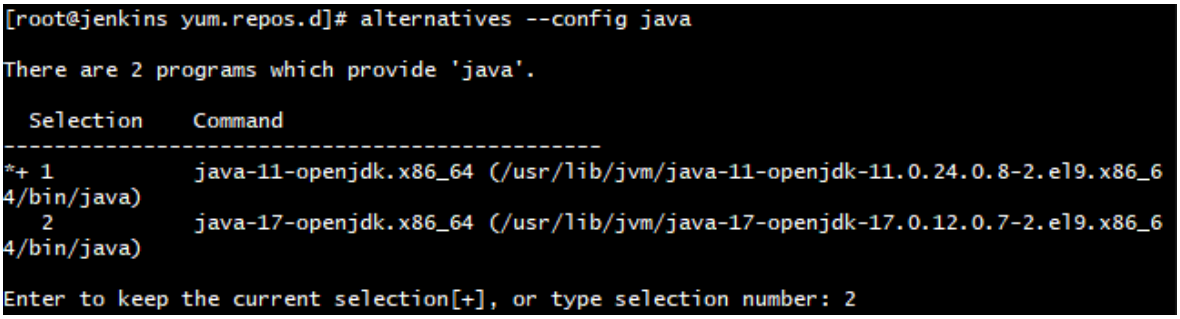
Setup CI/CD with GitHub, Jenkins, Maven & Tomcat to deploy an artifact (Java web app).

Step 1: Set up Jenkins

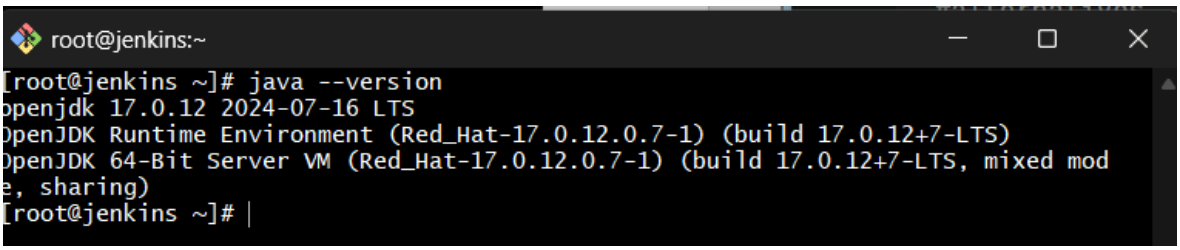
a. Install Java



- If you have an old version of JDK installed, configure the alternative Java version



- Select the appropriate Java version and verify:



- Set the JAVA_HOME environment variable:

```
root@jenkins:/usr/lib/jvm/java-17-openjdk-17.0.12.0.7-2.el9.x86_64
[root@jenkins ~]# cd /usr/lib/jvm/java-17-openjdk-17.0.12.0.7-2.el9.x86_64/
[root@jenkins java-17-openjdk-17.0.12.0.7-2.el9.x86_64]# pwd
/usr/lib/jvm/java-17-openjdk-17.0.12.0.7-2.el9.x86_64
[root@jenkins java-17-openjdk-17.0.12.0.7-2.el9.x86_64]#
```

- Copy the output path and add it to the .bash_profile:

```
# .bash_profile

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi
JAVA_HOME=/usr/lib/jvm/java-17-openjdk-17.0.12.0.7-2.el9.x86_64
PATH=$PATH:$HOME/bin:$JAVA_HOME
# User specific environment and startup programs
~
```

- Refresh the environment:

```
[root@jenkins yum.repos.d]# . /root/.bash_profile
[root@jenkins yum.repos.d]#
```

b. Install Jenkins

```
root@jenkins:/etc/yum.repos.d
[root@jenkins /]# cd /etc/yum.repos.d/
[root@jenkins yum.repos.d]#
```

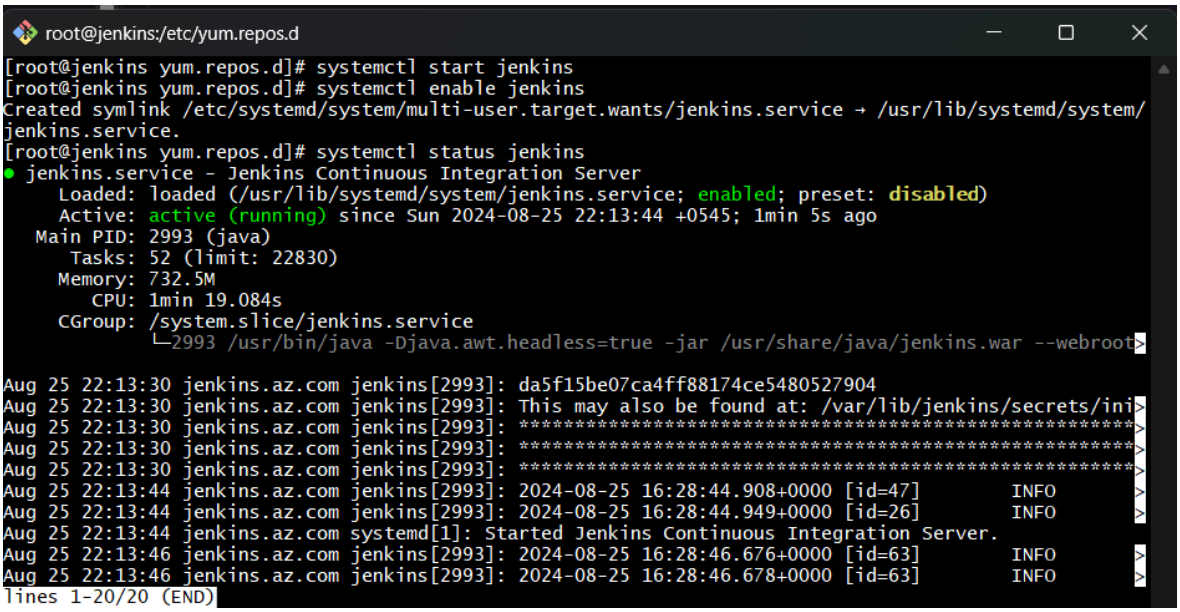
- Browse the official Jenkins website for the genuine Jenkins repository URL and key.

```
[root@jenkins yum.repos.d]# wget https://pkg.jenkins.io/redhat-stable/jenkins.repo
--2024-08-25 22:07:45--  https://pkg.jenkins.io/redhat-stable/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io)... 199.232.22.133, 2a04:4e42:42::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|199.232.22.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 85
Saving to: 'jenkins.repo'

jenkins.repo              100%[=====================================>]      85  --.-KB/s    in 0s

2024-08-25 22:07:45 (1.92 MB/s) - 'jenkins.repo' saved [85/85]

[root@jenkins yum.repos.d]# rpm --import https://pkg.jenkins.io/redhat/jenkins.io-2023.key
[root@jenkins yum.repos.d]# yum upgrade
Jenkins-stable                                                  16 kB/s |  29 kB     00:01
Dependencies resolved.
Nothing to do.
Complete!
```

```
[root@jenkins yum.repos.d]# yum install jenkins
Last metadata expiration check: 0:00:19 ago on Sun 25 Aug 2024 10:08:35 PM +0545.
Dependencies resolved.
================================================================================
 Package              Architecture         Version                  Repository              Size
================================================================================
Installing:
 jenkins              noarch               2.462.1-1.1              jenkins                 89 M

Transaction Summary
================================================================================
```

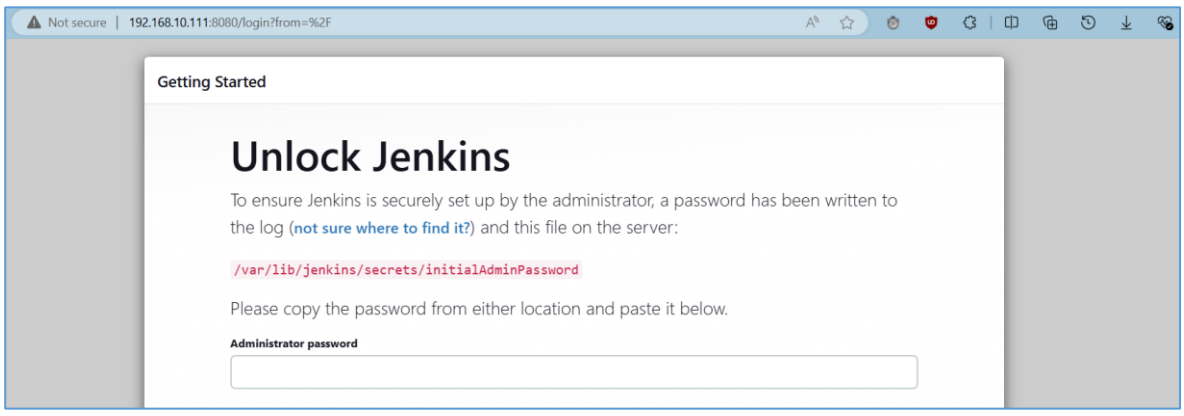
```
[root@jenkins yum.repos.d]# rpm -q jenkins
jenkins-2.462.1-1.1.noarch
```

- Start the Jenkins service:

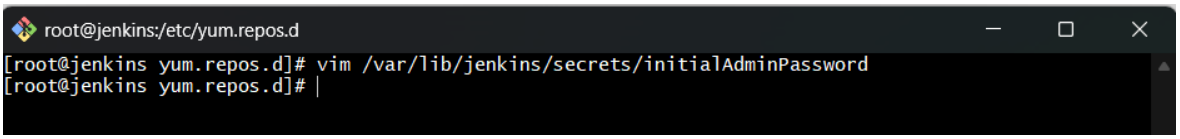


c. Add the Jenkins port to the firewall for global access:

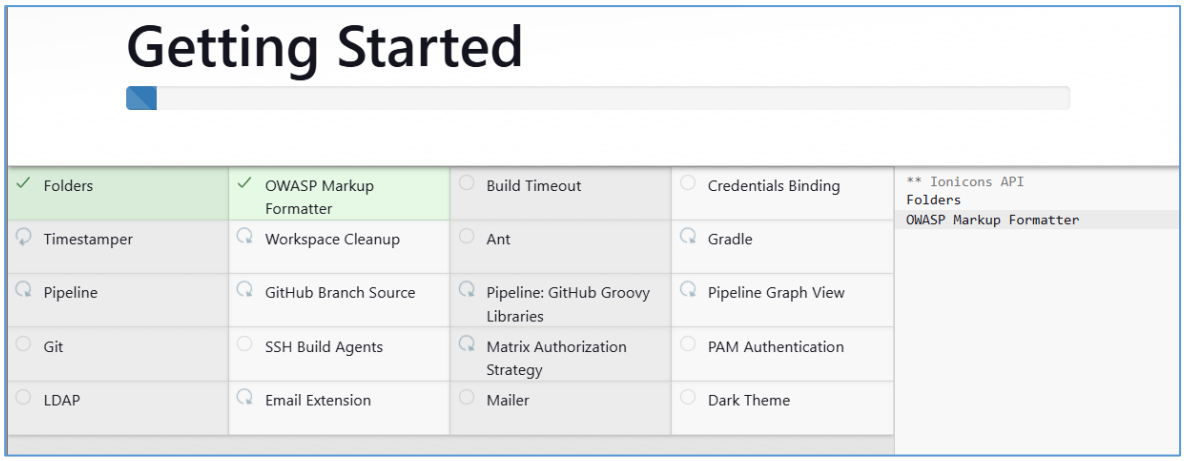
d. Access the Jenkins dashboard:



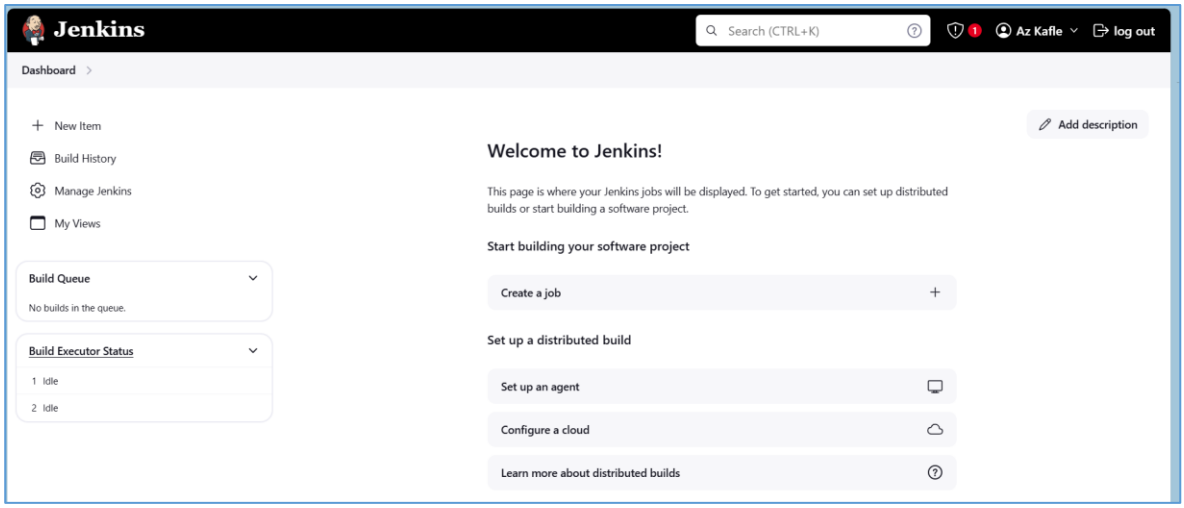
- Follow the instructions to unlock Jenkins using the password provided in the specified location.



- Install the suggested plugins.



- Create the first admin user.
- Start using Jenkins!



Step2: Setup Tomcat Server

a. Install java

```
root@webserver:~
[root@webserver ~]# yum -y install java-17-openjdk
AlmaLinux 9 - AppStream                                  1.8 kB/s | 4.2 kB     00:02
AlmaLinux 9 - BaseOS                                     2.4 kB/s | 3.8 kB     00:01
AlmaLinux 9 - CRB                                        2.7 kB/s | 4.2 kB     00:01
AlmaLinux 9 - Extras                                     1.7 kB/s | 3.3 kB     00:01
Dependencies resolved.
================================================================================
 Package                  Architecture   Version                  Repository      Size
================================================================================
Installing:
 java-17-openjdk          x86_64         1:17.0.12.0.7-2.el9      appstream      429 k
Installing dependencies:
 copy-jdk-configs         noarch         4.0-3.el9                appstream       27 k
 java-17-openjdk-headless x86_64         1:17.0.12.0.7-2.el9      appstream       45 M
 javapackages-filesystem  noarch         6.0.0-4.el9              appstream       10 k
 lksctp-tools             x86_64         1.0.19-3.el9_4           baseos          96 k
 lua                      x86_64         5.4.4-4.el9              appstream      187 k
 lua-posix                x86_64         35.0-8.el9               appstream      131 k
```

- Verify the installation:

```
[root@webserver ~]# java -version
openjdk version "17.0.12" 2024-07-16 LTS
OpenJDK Runtime Environment (Red_Hat-17.0.12.0.7-1) (build 17.0.12+7-LTS)
OpenJDK 64-Bit Server VM (Red_Hat-17.0.12.0.7-1) (build 17.0.12+7-LTS, mixed mode, sharing)
[root@webserver ~]#
```

b. Set JAVA_HOME Variable Path

- Navigate to the Java installation directory:

```
root@webserver:/usr/lib/jvm/java-17-openjdk-17.0.12.0.7-2.el9.x86_64
[root@webserver /]# cd /usr/lib/jvm/java-17-openjdk-17.0.12.0.7-2.el9.x86_64/
[root@webserver java-17-openjdk-17.0.12.0.7-2.el9.x86_64]# pwd
/usr/lib/jvm/java-17-openjdk-17.0.12.0.7-2.el9.x86_64
[root@webserver java-17-openjdk-17.0.12.0.7-2.el9.x86_64]#
```

- Edit the .bash_profile

```
root@webserver:/
[root@webserver java-17-openjdk-17.0.12.0.7-2.el9.x86_64]# cd /
[root@webserver /]# vi /root/.bash_profile
```

```
root@webserver:/
# .bash_profile
JAVA_HOME= /usr/lib/jvm/java-17-openjdk-17.0.12.0.7-2.el9.x86_64
PATH=$PATH:$HOME/bin:$JAVA_HOME

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi

# User specific environment and startup programs
~
```

- Refresh the profile:

```
[root@webserver /]# . /root/.bash_profile
-bash: /usr/lib/jvm/java-17-openjdk-17.0.12.0.7-2.el9.x86_64: Is a directory
```

c. Download and Extract Apache Tomcat

- Create a directory for Tomcat

```
root@webserver:/opt
[root@webserver ~]# cd /opt/
[root@webserver opt]# wget https://dlcdn.apache.org/tomcat/tomcat-11/v11.0.0-M24/bin/apache-to
mcat-11.0.0-M24.tar.gz
--2024-09-02 08:21:52--  https://dlcdn.apache.org/tomcat/tomcat-11/v11.0.0-M24/bin/apache-tomc
at-11.0.0-M24.tar.gz
Resolving dlcdn.apache.org (dlcdn.apache.org)... 151.101.2.132, 2a04:4e42::644
Connecting to dlcdn.apache.org (dlcdn.apache.org)|151.101.2.132|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 13055617 (12M) [application/x-gzip]
Saving to: 'apache-tomcat-11.0.0-M24.tar.gz'

apache-tomcat-11.0.0-M2 100%[============================>]  12.45M  6.61MB/s    in 1.9s

2024-09-02 08:21:54 (6.61 MB/s) - 'apache-tomcat-11.0.0-M24.tar.gz' saved [13055617/13055617]
```

- Extract the downloaded file

```
[root@webserver opt]# tar -zxvf apache-tomcat-11.0.0-M24.tar.gz
apache-tomcat-11.0.0-M24/conf/
apache-tomcat-11.0.0-M24/conf/catalina.properties
apache-tomcat-11.0.0-M24/conf/context.xml
```

- Create soft links for tomcat startup and shutdown scripts

```
root@webserver:/opt/apache-tomcat-11.0.0-M24/conf
[root@webserver opt]# cd apache-tomcat-11.0.0-M24/bin/
[root@webserver bin]# ln -s /opt/apache-tomcat-11.0.0-M24/bin/startup.sh /usr/local/bin/tomcat
up
[root@webserver bin]# ln -s /opt/apache-tomcat-11.0.0-M24/bin/shutdown.sh /usr/local/bin/tomca
tdown
[root@webserver bin]#
```

d. Change Tomcat Server Port to 8090

- Navigate to the configuration directory

```
[root@webserver bin]# cd /opt/apache-tomcat-11.0.0-M24/conf/
[root@webserver conf]# ls
catalina.properties  jaspic-providers.xml  logging.properties  tomcat-users.xml  web.xml
context.xml          jaspic-providers.xsd  server.xml          tomcat-users.xsd
[root@webserver conf]# vi context.xml
[root@webserver conf]# cd /opt/apache-tomcat-11.0.0-M24/conf/
[root@webserver conf]# vi server.xml
```

- Edit server.xml (Change the connector port to 8090.)

```
root@webserver:/opt/apache-tomcat-11.0.0-M24/conf
    -->
    <Connector port="8090" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />
    <!-- A "Connector" using the shared thread pool-->
    <!--
    <Connector executor="tomcatThreadPool"
               port="8080" protocol="HTTP/1.1"
               connectionTimeout="20000"
               redirectPort="8443" />
    -->
```

e. Allow Port 8090 in Firewall

```
root@webserver:/opt/apache-tomcat-11.0.0-M24/conf
[root@webserver conf]# firewall-cmd --permanent --add-port=8090/tcp
success
[root@webserver conf]# firewall-cmd --reload
success
[root@webserver conf]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens160
  sources:
  services: cockpit dhcpv6-client http ssh
  ports: 8090/tcp
  protocols:
  forward: yes
  masquerade: no
```

f. Allow Remote Access to Tomcat Server

- Edit context.xml for host-manager (Comment out the Valve section.)

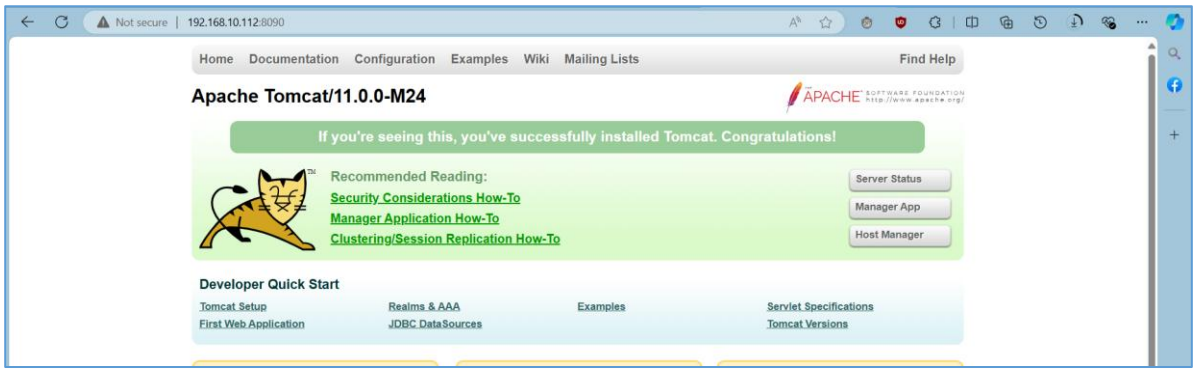
```
root@webserver:/opt/apache-tomcat-11.0.0-M24/webapps/host-manager/META-INF
[root@webserver /]# cd /opt/apache-tomcat-11.0.0-M24/webapps/host-manager/META-INF/
[root@webserver META-INF]# vi context.xml
-->
<Context antiResourceLocking="false" privileged="true" ignoreAnnotations="true">
  <CookieProcessor className="org.apache.tomcat.util.http.Rfc6265CookieProcessor"
                   sameSiteCookies="strict" />
  <!-- Valve className="org.apache.catalina.valves.RemoteAddrValve"
         allow="127\.\d+\.\d+\.\d+|::1|0:0:0:0:0:0:0:1" / -->
  <Manager sessionAttributeValueClassNameFilter="java\.lang\.(?:Boolean|Integer|Long|Number|St
ring)|org\.apache\.catalina\.filters\.CsrfPreventionFilter\$LruCache(?:\$1)?|java\.util\.(?:Li
nked)?HashMap"/>
</Context>
```

g. Test the Tomcat server in a browser

- Start tomcat server

```
[root@webserver META-INF]# tomcatup
Using CATALINA_BASE:   /opt/apache-tomcat-11.0.0-M24
Using CATALINA_HOME:   /opt/apache-tomcat-11.0.0-M24
Using CATALINA_TMPDIR: /opt/apache-tomcat-11.0.0-M24/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/apache-tomcat-11.0.0-M24/bin/bootstrap.jar:/opt/apache-tomcat-11.0
.0-M24/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
```

- URL: http://192.168.100.153:8090



h. Setup Credentials for Tomcat Manager

```
root@webserver:/opt/apache-tomcat-11.0.0-M24/conf
[root@webserver /]# cd /opt/apache-tomcat-11.0.0-M24/conf/
[root@webserver conf]# ls
Catalina            jaspic-providers.xml  server.xml       web.xml
catalina.properties jaspic-providers.xsd  tomcat-users.xml
context.xml         logging.properties    tomcat-users.xsd
[root@webserver conf]# vi tomcat-users.xml
```

```
        <role rolename="manager-gui"/>
        <role rolename="manager-script"/>
        <role rolename="manager-jmx"/>
        <role rolename="manager-status"/>
        <role rolename="admin-gui"/>
        <role rolename="admin"/>
        <role rolename="manager"/>
        <role rolename="admin-script"/>

        <user username="kafleaz" password="kafle126" roles="admin,admin-gui,admin-script,m
nager,manager-gui,manager-script,manager-jmx,manager-status"/>
        <!-- user username="kafleaz" password="kafle126" roles="manager-gui"/ -->
        <!-- user username="kafleaz" password="7878" roles="admin-gui"/ -->
</tomcat-users>
"tomcat-users.xml" 68L, 3321B                                    68,15         Bot
```

- Restart Tomcat

```
[root@webserver /]# tomcatdown
Using CATALINA_BASE:   /opt/apache-tomcat-11.0.0-M24
Using CATALINA_HOME:   /opt/apache-tomcat-11.0.0-M24
Using CATALINA_TMPDIR: /opt/apache-tomcat-11.0.0-M24/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/apache-tomcat-11.0.0-M24/bin/bootstrap.jar:/opt/apache-
tomcat-11.0.0-M24/bin/tomcat-juli.jar
```

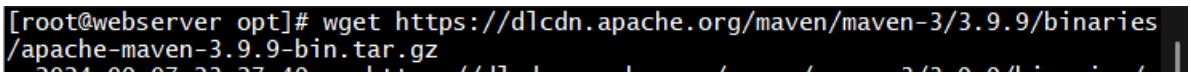
```
[root@webserver META-INF]# tomcatup
Using CATALINA_BASE:   /opt/apache-tomcat-11.0.0-M24
Using CATALINA_HOME:   /opt/apache-tomcat-11.0.0-M24
Using CATALINA_TMPDIR: /opt/apache-tomcat-11.0.0-M24/temp
Using JRE_HOME:        /usr
Using CLASSPATH:       /opt/apache-tomcat-11.0.0-M24/bin/bootstrap.jar:/opt/apache-tomcat-11.0
.0-M24/bin/tomcat-juli.jar
Using CATALINA_OPTS:
Tomcat started.
```

Step3: Setup & Configure Maven

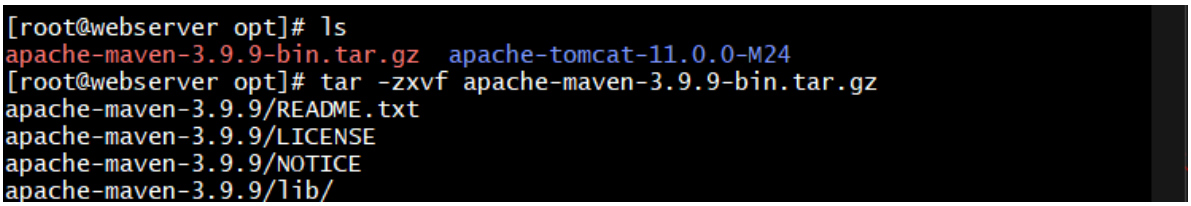
a. Install Maven

- Create a directory for Maven:

- Download Maven binary

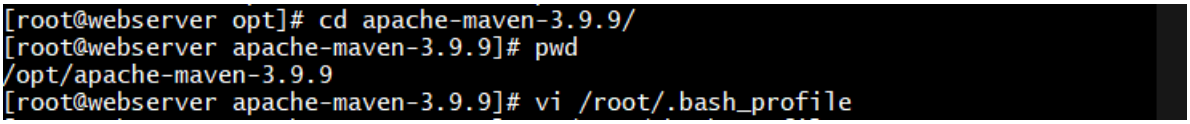


- Extract the downloaded file

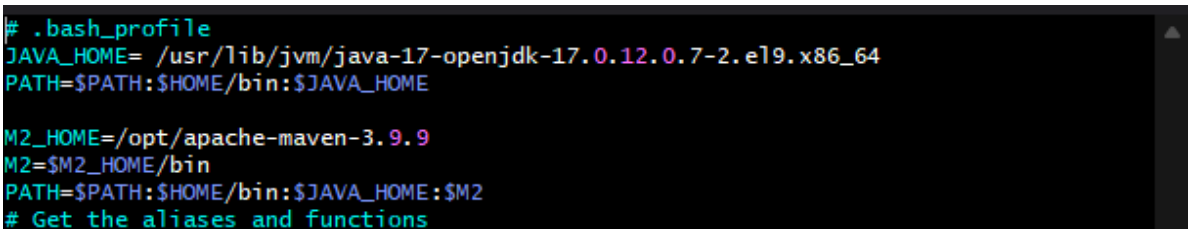


b. Set M2 and M2_HOME Variable Path

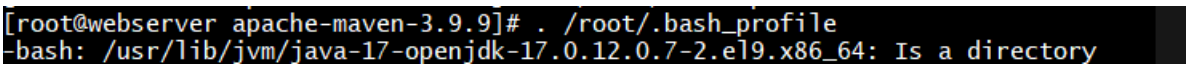
- Navigate to the Maven directory



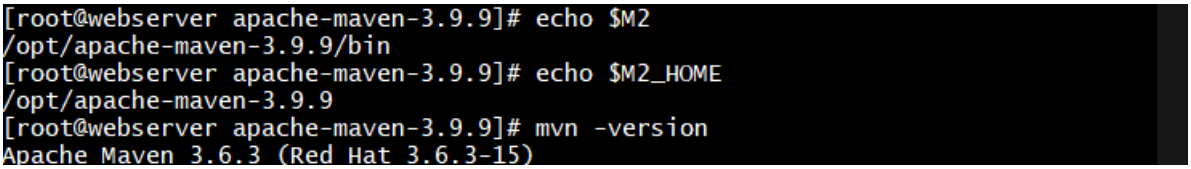
- Edit the .bash_profile



- Refresh the profile

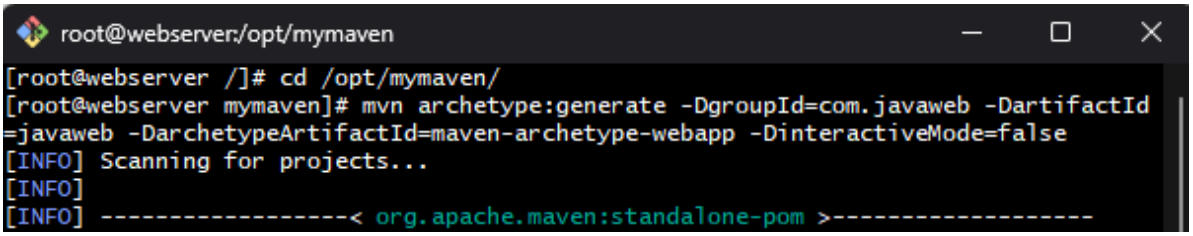


- Verify Maven installation



c. Create a Java-based Web Application Using Maven

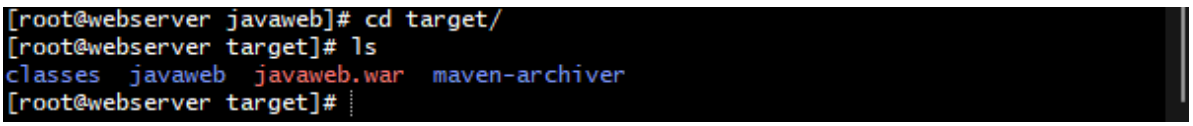
- Create a directory for the Maven project and generate a web application project.



- Navigate to the project directory

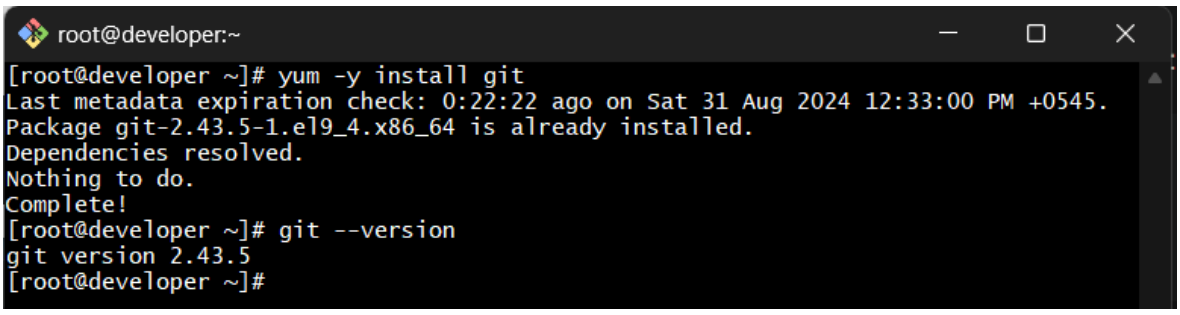
- Compile and package the application

- Verify the generated .war file



Step 4: Setup Git

a. Installing Git in Developer Machine



b. Configuring SSH Authentication with GitHub

- Generate an SSH key pair

```
[root@developer ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
/root/.ssh/id_rsa already exists.
Overwrite (y/n)?
[root@developer ~]#
```

- Navigate to the SSH directory and display the public key.

```
root@developer:~/.ssh
[root@developer ~]# cd /root/.ssh/
[root@developer .ssh]# cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCUpRg5ClXh05X90svD0PnxU4WeosxoWgHEAK9k/BvBFV7
```

- Copy the displayed key and add it to GitHub:

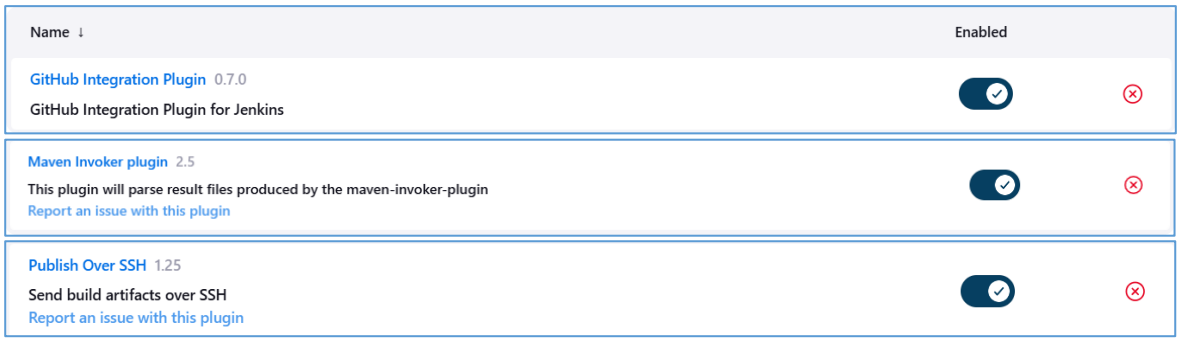
- Test the SSH connection:

```
root@developer:/
[root@developer /]# ssh -T git@github.com
Hi kafleaz! You've successfully authenticated, but GitHub does not provide shell ac
cess.
[root@developer /]#
```

Step 4: Integrate GitHub, Maven, Tomcat Server with Jenkins

a. Install Required Plugins in Jenkins for Github Maven, Tomcat Server Integration

- Navigate to **Manage Jenkins** > **Plugins** > **Available Plugins** > Search plugins
- Select it and click **Install without restart**.

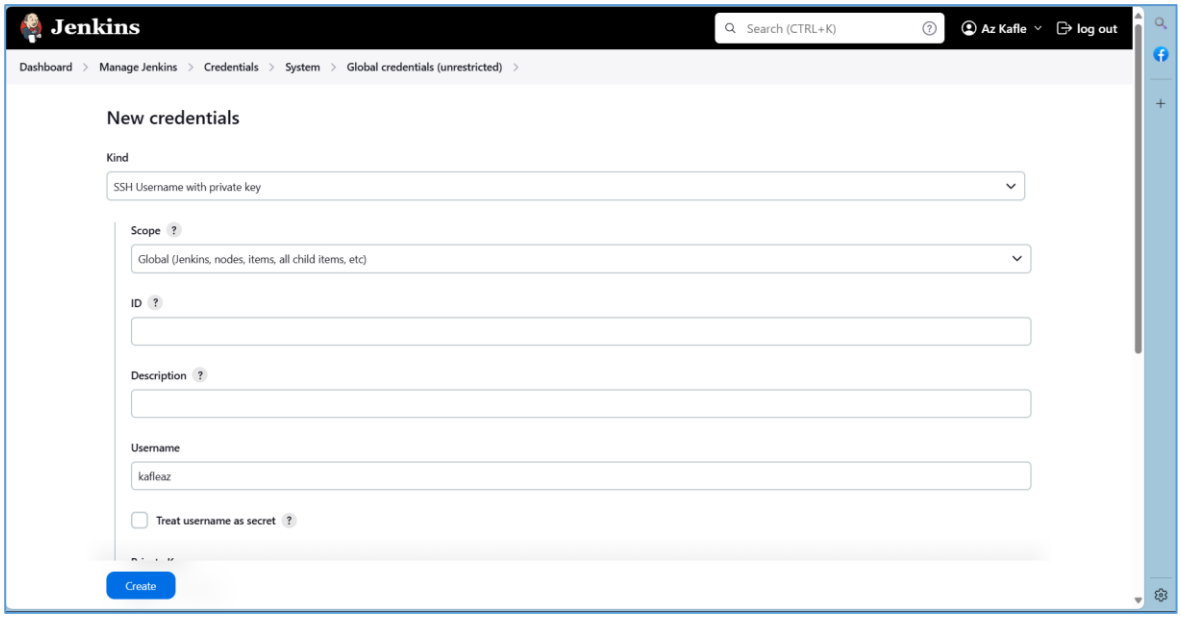


b. jenkins and Github ssh authentication configuration

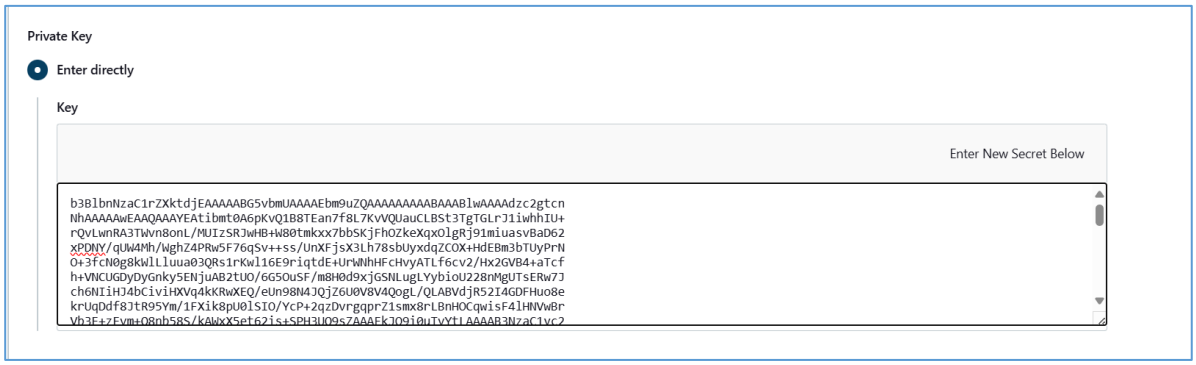
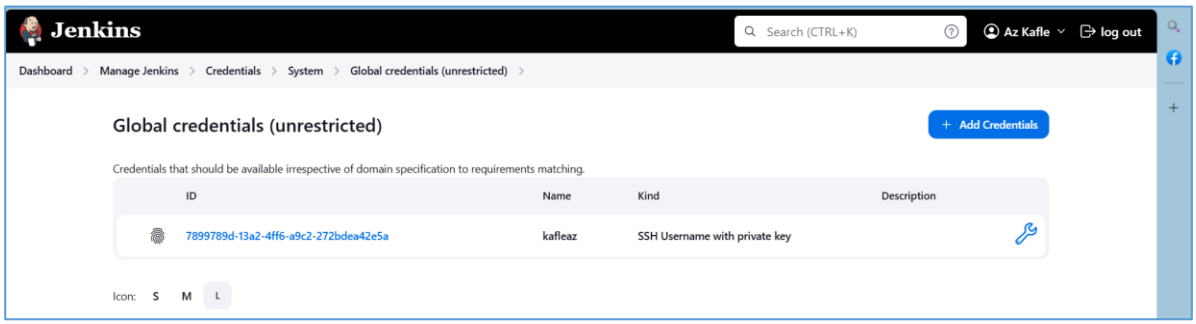
- Display the private key

```
[root@jenkins /]# cd /root/.ssh/
[root@jenkins .ssh]# cat id_rsa
-----BEGIN OPENSSH PRIVATE KEY-----
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAtibmt0A6pKvQ1B8TEan7f8L7KvVQUauCLBSt3TgTGLrJ1iwhhIU+
rQvLwnRA3TWvn8onL/MUIzSRJwHB+W80tmkxx7bbSKjFhOZkeXqxOlgRj91miuasvBaD62
```

- In Jenkins, navigate to **Manage Jenkins** > **Credentials** > **System** > **Global credentials**.
- Click **Add Credentials**.
- Enter your **GitHub username**.

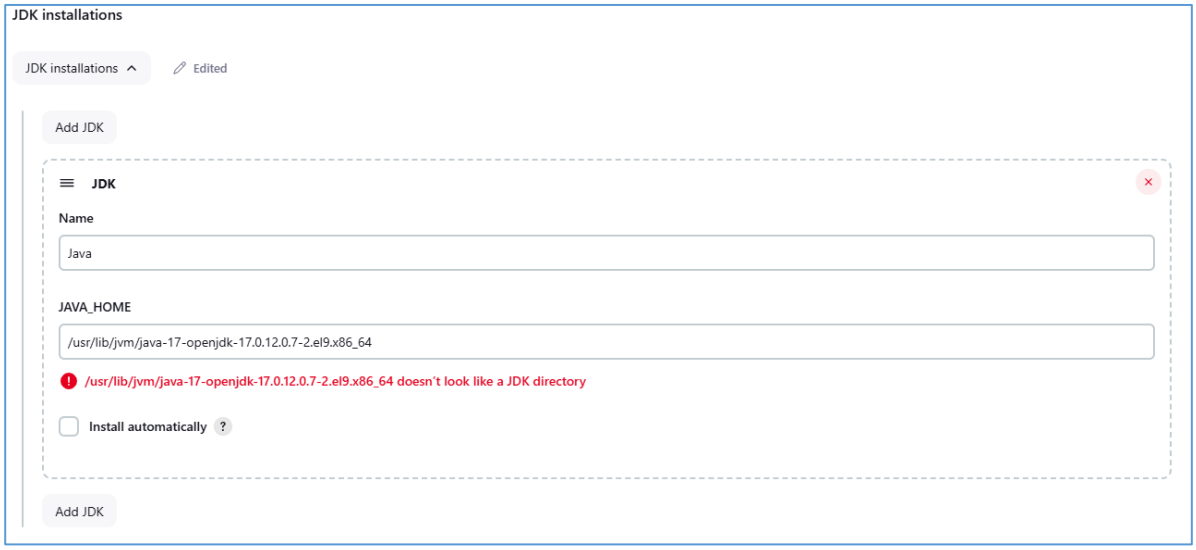


- Paste the copied private key.

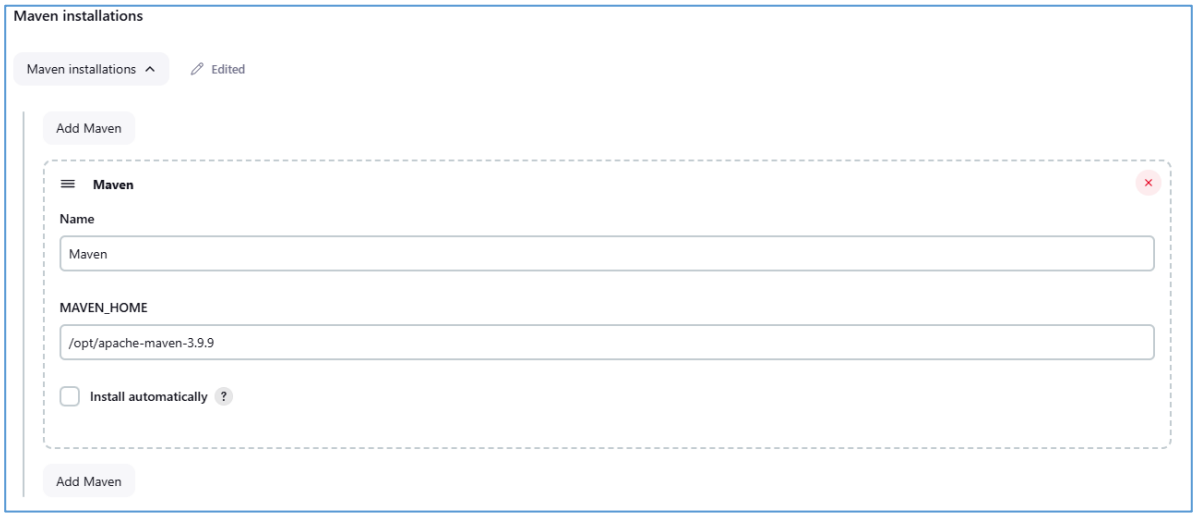

Private Key
Enter directly
Key
Enter New Secret Below
b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn
NhAAAAAwEAAQAAAYEAtibmt0A6pKvQ1B8TEan7f8L7KvVQUauCLBSt3TgTGLrJ1iwhhIU+
rQvLwnRA3TWvn8onL/MUIzSRJwHB+W80tmkxx7bbSKjFhOZkeXqxOlgRj91miuasvBaD62
xPDNY/qUW4Mh/WghZ4PRw5F76qSv++ss/UnXFjsX3Lh78sbUyxdqZCOX+HdEBm3bTUyPrN
O+3fcN0g8kWllluua03QRs1rKwl16E9riqtdE+UrWNhHFcHvyATlf6cv2/Hx2GVB4+aTcf
h+VNCUGDyDyGnky5ENjuAB2tUO/6G5OuSF/m8H0d9xjGSNLugLYybioU228nMgUTsERw7J
ch6NIiHJ4bCiviHXVq4kKRwXEQ/eUn98N4JQjZ6U0V8V4QogL/QLABVdjR52I4GDFHuo8e
krUqDdf8JtR95Ym/1FXik8pU0lSIO/YcP+2qzDvrgqprZ1smx8rLBnHOCqwisF4lHNVwBr
Vb3E+zEvm+O8nb5RS/kAWxX5et62is+SPH3UO9sZAAAFkJO9i0uTvYtIAAAAB3NzaC1yc2
```

- Click **Create**.


```

c. Specify java home and maven home (JAVA_HOME and M2_HOME) path in Jenkins.

- Navigate to **Manage Jenkins** > **Tools** > **Add JDK**



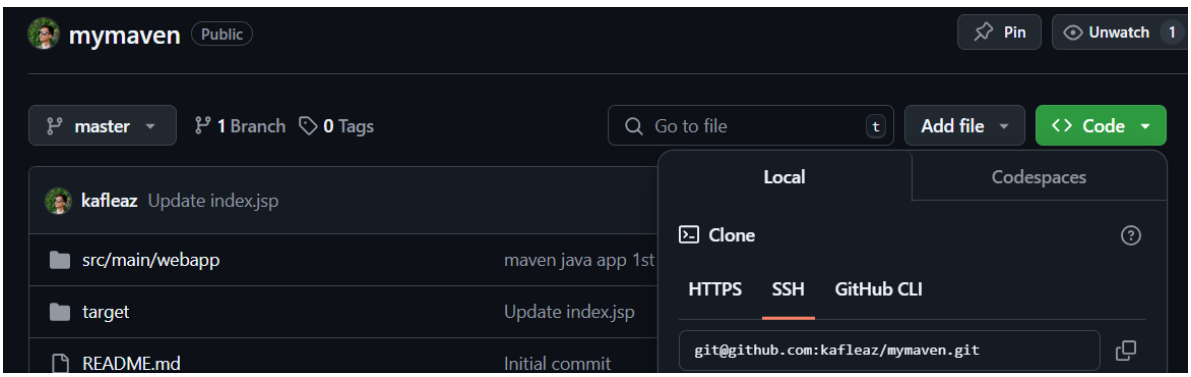
- Navigate to **Manage Jenkins** > **Tools** > **Add Maven**



- **Apply > Save**

Step 5: Create CI and CD job to deploy an artifact (java web app)

a. Create new repo in github



b. Create folder for maven project

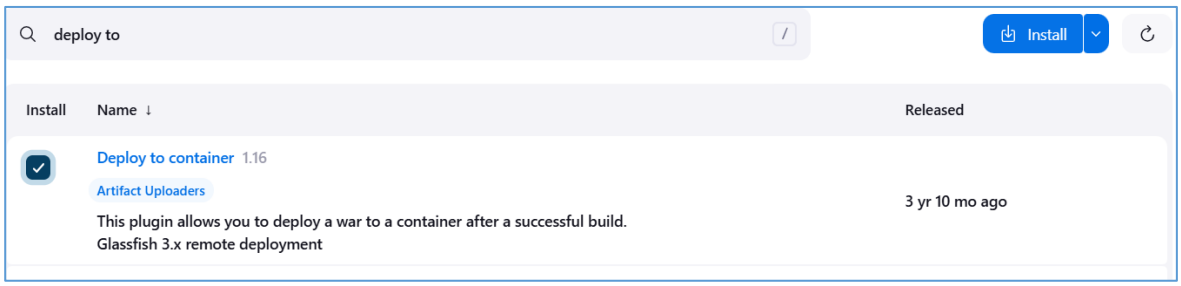
```
root@developer:/project/maven1
[root@developer ~]# cd /project/
[root@developer project]# mkdir maven1
[root@developer project]# cd maven1/
```

c. Create local rep.

```
[root@developer maven1]# git init
Initialized empty Git repository in /project/maven1/.git/
[root@developer maven1]# git remote add origin git@github.com:kafleaz/mymaven.gi
t
[root@developer maven1]# git pull origin master
remote: Enumerating objects: 31, done.
remote: Counting objects: 100% (31/31), done.
remote: Compressing objects: 100% (25/25), done.
remote: Total 31 (delta 3), reused 12 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (31/31), 6.40 KiB | 86.00 KiB/s, done.
From github.com:kafleaz/mymaven
 * branch            master     -> FETCH_HEAD
 * [new branch]      master     -> origin/master
[root@developer maven1]# git add .
[root@developer maven1]# git commit -m "maven java web app"
On branch master
nothing to commit, working tree clean
[root@developer maven1]# git push origin master
```

d. Install "Deploy to container" plugin in the jenkins master

- Navigate to **Manage Jenkins** > **Plugins** > **Available Plugins** > Search plugins
- Select it and click **Install without restart**.



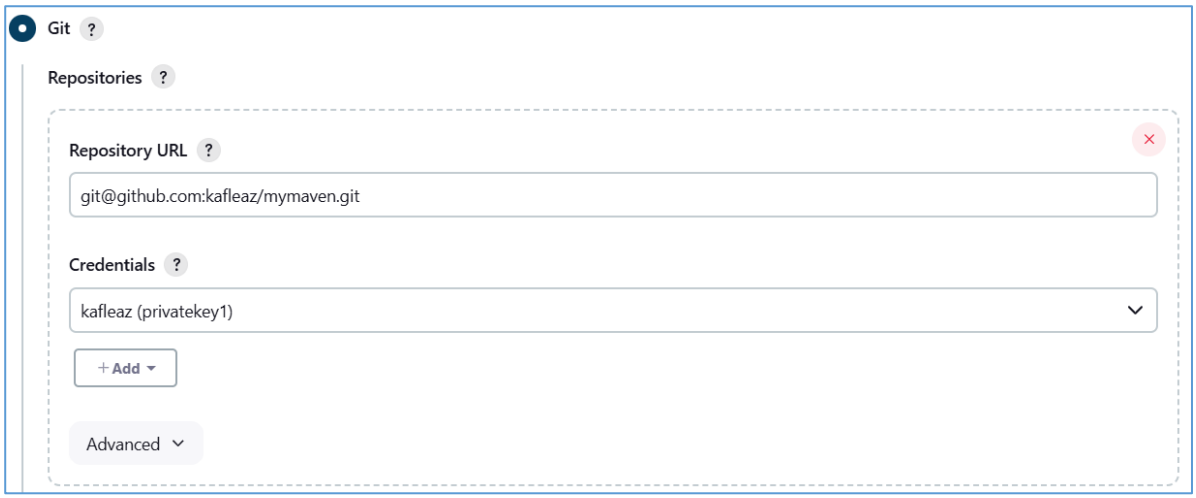
e. Start (sync) NTP server

```
root@jenkins:~
[root@jenkins ~]# systemctl start chronyd
[root@jenkins ~]# systemctl enable chronyd
[root@jenkins ~]# systemctl status chronyd
● chronyd.service - NTP client/server
     Loaded: loaded (/usr/lib/systemd/system/chronyd.service; enabled; preset: >
     Active: active (running) since Sat 2024-09-07 18:27:56 +0545; 4h 20min ago
       Docs: man:chronyd(8)
```

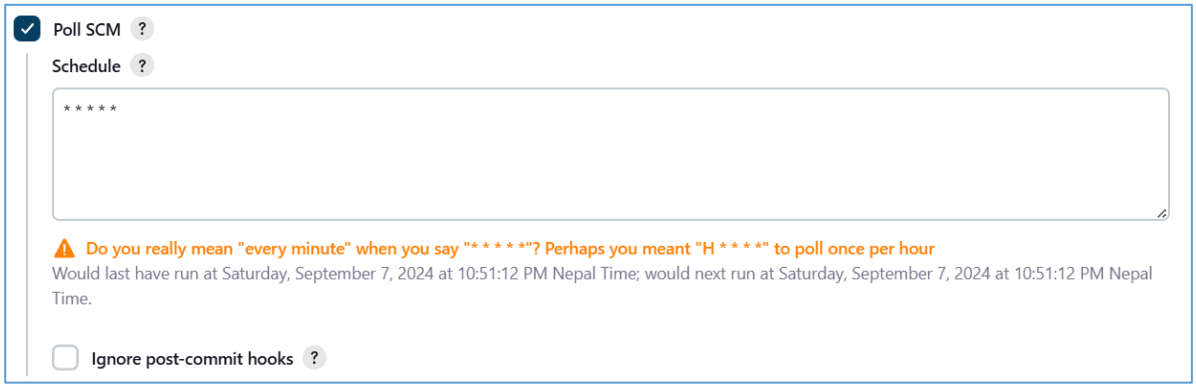
f. Create a job such that the jenkins master pulls code from github repository which then asks to buid (packages) and push the built artifact to the tomcat server.

- In Jenkins, navigate to **Jenkins Dashboard** > **New item** > enter item name **(mymaven_P5) > Freestyle project > ok**

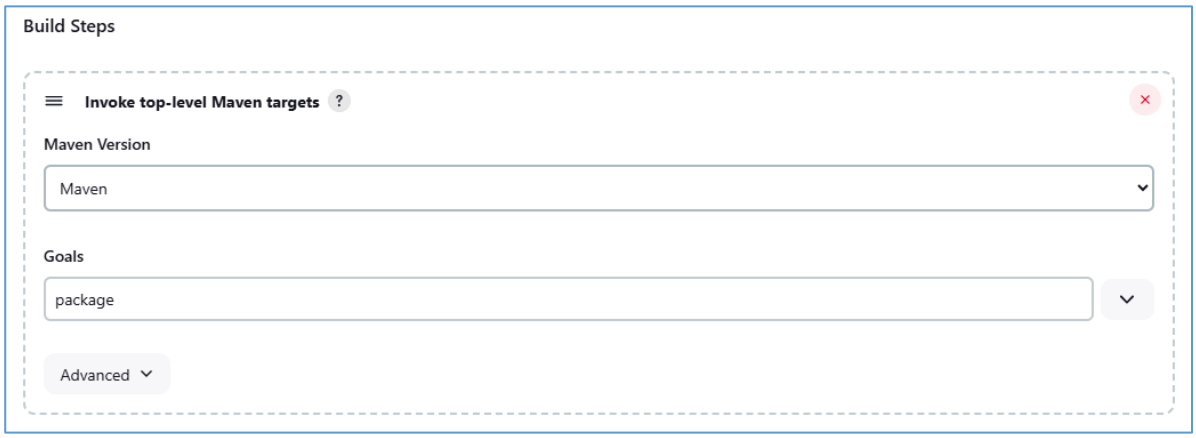
- Select **Git**, add **repo url** and **credentials**.



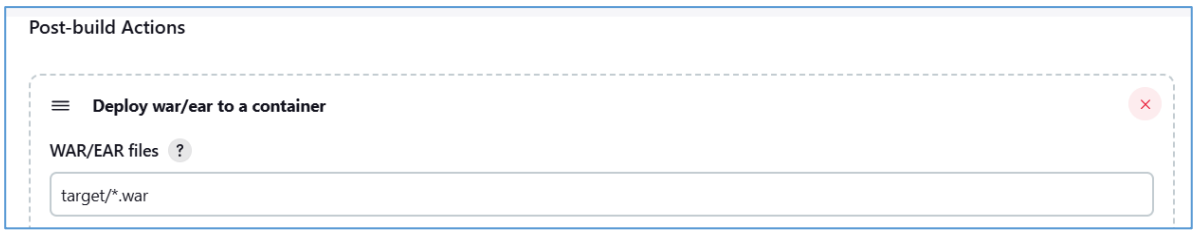
- **Build Trigger > Poll SCM**



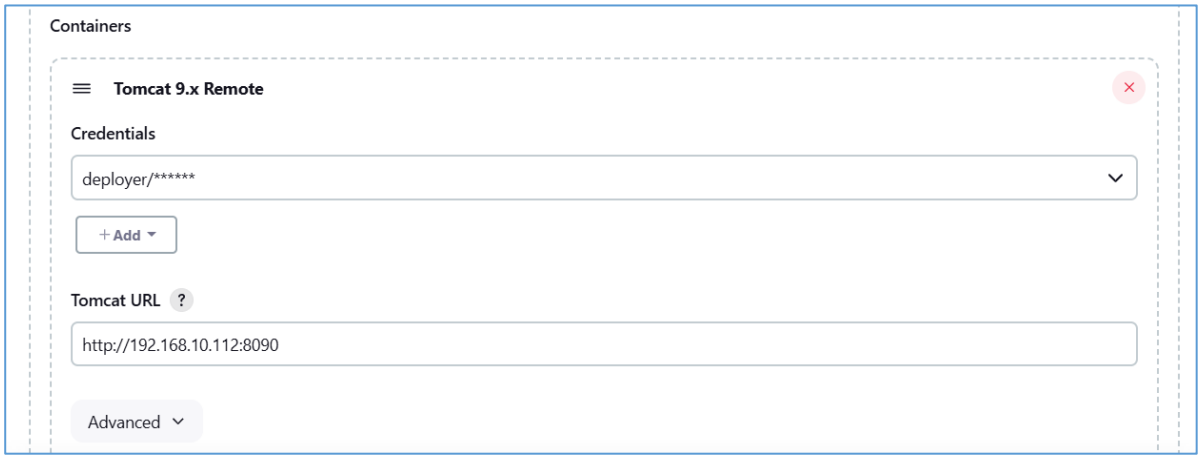
- **Build Steps > Add build step > invoke top-level Maven Targets**



- **Post-build Actions > Add post-build action > Deploy war/ear to a container**



- **Post-build actions > Add post-build action > Deploy war/ear to a container >Add container > Tomcat 9.x Remote > credentaials > username >password >Apply > save**



- **Apply > Save**

Step 5: Test The Deployment

- Check Console Output:

```
[INFO] Packaging webapp
[INFO] Assembling webapp [javaweb] in [/var/lib/jenkins/workspace/mymaven_p5/target/javaweb]
[INFO] Processing war project
[INFO] Copying webapp resources [/var/lib/jenkins/workspace/mymaven_p5/src/main/webapp]
[INFO] Building war: /var/lib/jenkins/workspace/mymaven_p5/target/javaweb.war
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  4.320 s
[INFO] Finished at: 2024-09-07T23:54:11+05:45
[INFO] ------------------------------------------------------------------------
[DeployPublisher][INFO] Attempting to deploy 1 war file(s)
[DeployPublisher][INFO] Deploying /var/lib/jenkins/workspace/mymaven_p5/target/javaweb.war to container Tomcat 9.x Remote with context null
  [/var/lib/jenkins/workspace/mymaven_p5/target/javaweb.war] is not deployed. Doing a fresh deployment.
  Deploying [/var/lib/jenkins/workspace/mymaven_p5/target/javaweb.war]
Finished: SUCCESS
```

- Access the deployed web (Url: 192.168.10.112:8090/javaweb)