1) create a user Jack with unilao password and find jack user from /etc/passwd file along with their UID and home directory. and assign sudo delegation for jack user. So, he can create a user with name Barbosa.

```
[root@ip-172-31-10-62 ~]# useradd Jack
[root@ip-172-31-10-62 ~]# passwd Jack
Changing password for user Jack.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
```

```
[root@ip-172-31-10-62 ~]# cat /etc/passwd | grep -i Jack
Jack:x:1001:1001::/home/Jack:/bin/bash
```

```
[root@ip-172-31-10-62 ~]# whereis useradd
useradd: /usr/sbin/useradd /usr/share/man/man8/useradd.8.gz
[root@ip-172-31-10-62 ~]# whereis passwd
passwd: /usr/bin/passwd /etc/passwd /usr/share/man/man1/passwd.1ossl.gz /usr/share/man/man1/passwd.1.gz /usr/share/man/man5/passwd.5.
gz
[root@ip-172-31-10-62 ~]# vim /etc/sudoers
[root@ip-172-31-10-62 ~]# su - Jack
[Jack@ip-172-31-10-62 ~]$ sudo useradd Barbosa

We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:

    #1) Respect the privacy of others.
    #2) Think before you type.
    #3) With great power comes great responsibility.

For security reasons, the password you type will not be visible.

[sudo] password for Jack:
```

```
[Jack@ip-172-31-10-62 ~]$ cat /etc/passwd | grep -i Barbosa
Barbosa:x:1002:1002::/home/Barbosa:/bin/bash
[Jack@ip-172-31-10-62 ~]$ ~
```

2) For an existing project on GitHub, create a Jenkins project. This project should be triggered automatically whenever there is any change in any of the file in GitHub repository. Also make sure that the artifacts should be built and saved in the pipeline.

I have set our hostname as jenkins-server.example.com

And this is the installation process of jenkins

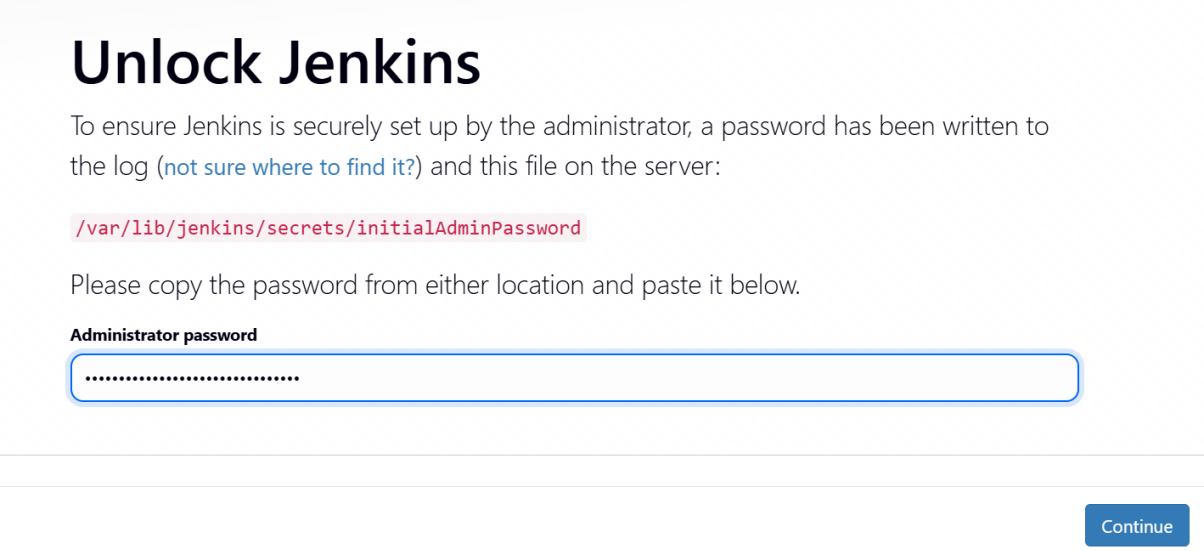
```
[root@ip-172-31-10-62 ~]# hostnamectl set-hostname jenkins-server.example.com
[root@ip-172-31-10-62 ~]# bash
[root@jenkins-server ~]# hostname
jenkins-server.example.com
[root@jenkins-server ~]# yum update -y
Amazon Linux 2023 Kernel Livepatch repository                                                    217 kB/s |  23 kB     00:00
Dependencies resolved.
Nothing to do.
Complete!
[root@jenkins-server ~]# yum install wget -y
Last metadata expiration check: 0:01:26 ago on Sat Sep 27 10:57:08 2025.
Package wget-1.21.3-1.amzn2023.0.4.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
```

```
[root@jenkins-server ~]# wget -O /etc/yum.repos.d/jenkins.repo \
    https://pkg.jenkins.io/redhat-stable/jenkins.repo
--2025-09-27 10:58:43--  https://pkg.jenkins.io/redhat-stable/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.38.133, 2a04:4e42:78::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|146.75.38.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 85
Saving to: '/etc/yum.repos.d/jenkins.repo'

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

2025-09-27 10:58:43 (1.20 MB/s) - '/etc/yum.repos.d/jenkins.repo' saved [85/85]

[root@jenkins-server ~]# rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
[root@jenkins-server ~]# yum upgrade
Jenkins-stable                                                                                   559 kB/s |  32 kB     00:00
Dependencies resolved.
Nothing to do.
Complete!
```

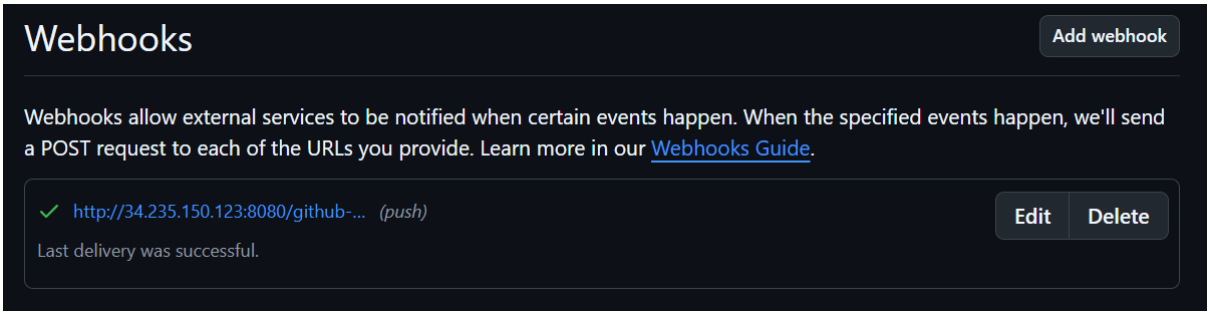
```
Installed:
  jenkins-2.516.3-1.1.noarch

Complete!
```

Finally Jenkins is installed.

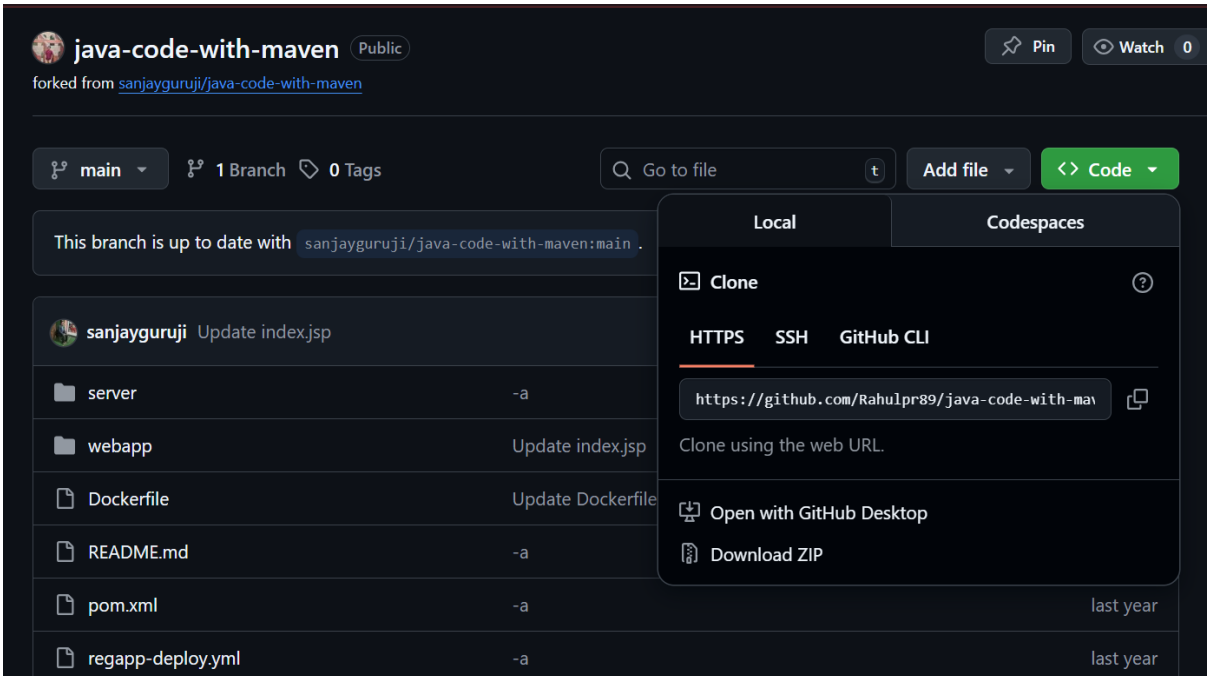
And after setting the Jenkins its set to open



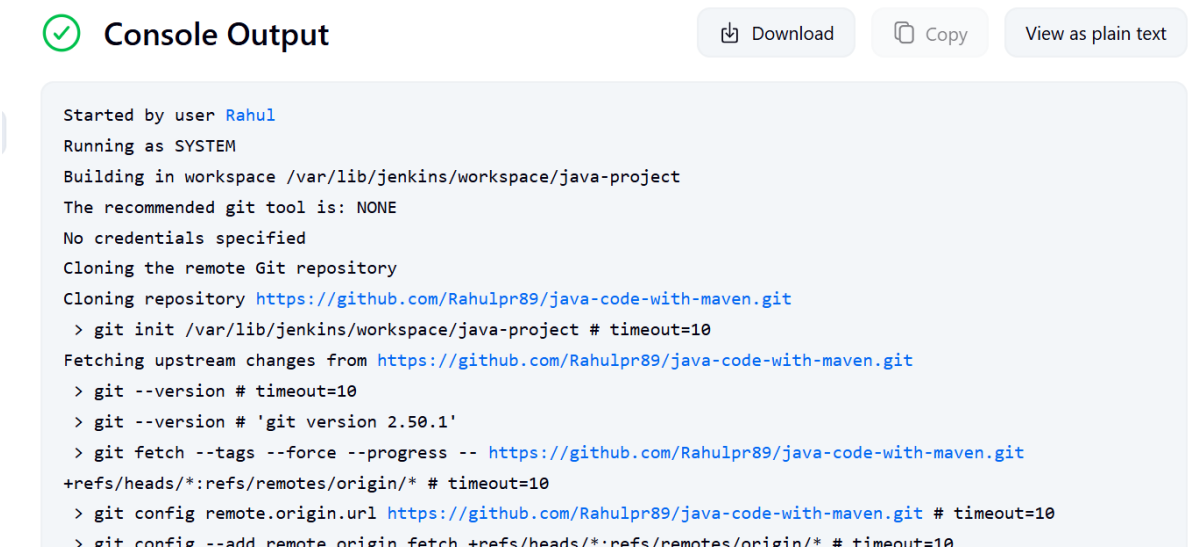
Webhook is created in github after generating token from jenkins



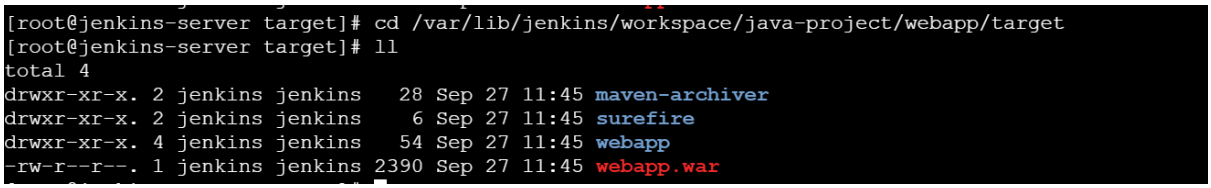
Copy the link of git repository



Console output is success



This is the final output



This ensures that build is automated

Triggers

Set up automated actions that start your build based on specific events, like code changes or scheduled times.

- [ ] Build whenever a SNAPSHOT dependency is built ?
- [ ] Trigger builds remotely (e.g., from scripts) ?
- [ ] Build after other projects are built ?
- [ ] Build periodically ?
- [ ] GitHub Branches
- [ ] GitHub Pull Requests ?
- [x] GitHub hook trigger for GITScm polling ?
- [ ] Poll SCM ?

Add description

All +

S	W	Name ↓	Last Success	Last Failure	Last Duration
✓	☁	java-project	8 min 43 sec #2	10 min #1	23 sec

java-project

Add description

Latest Test Result (no failures)

Permalinks

- Last build (#2), 9 min 3 sec ago
- Last stable build (#2), 9 min 3 sec ago
- Last successful build (#2), 9 min 3 sec ago
- Last failed build (#1), 11 min ago
- Last unsuccessful build (#1), 11 min ago
- Last completed build (#2), 9 min 3 sec ago