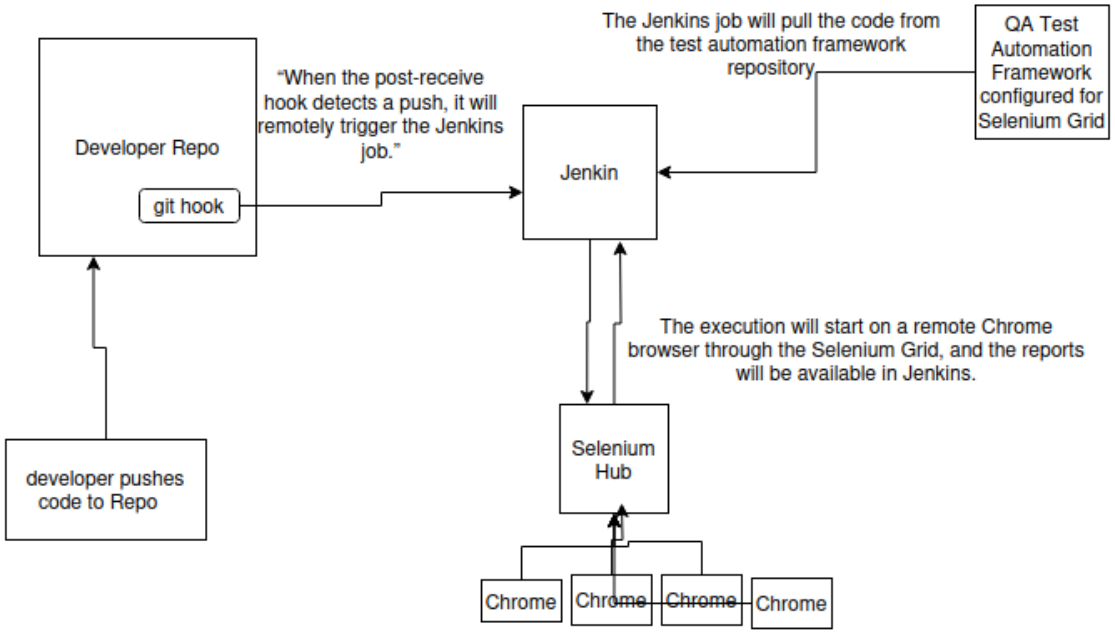
This documentation does not cover all the work and effort involved in creating the CI pipeline. I have only included the details that I believe may be of interest.

Flow diagram



Credentials (only work on my local network):

-> Jenkins and GitLab VM (shiv:1234)

-> Jenkins server IP address: 172.16.18.11:8080
Jenkins Credentials: shiv:1234

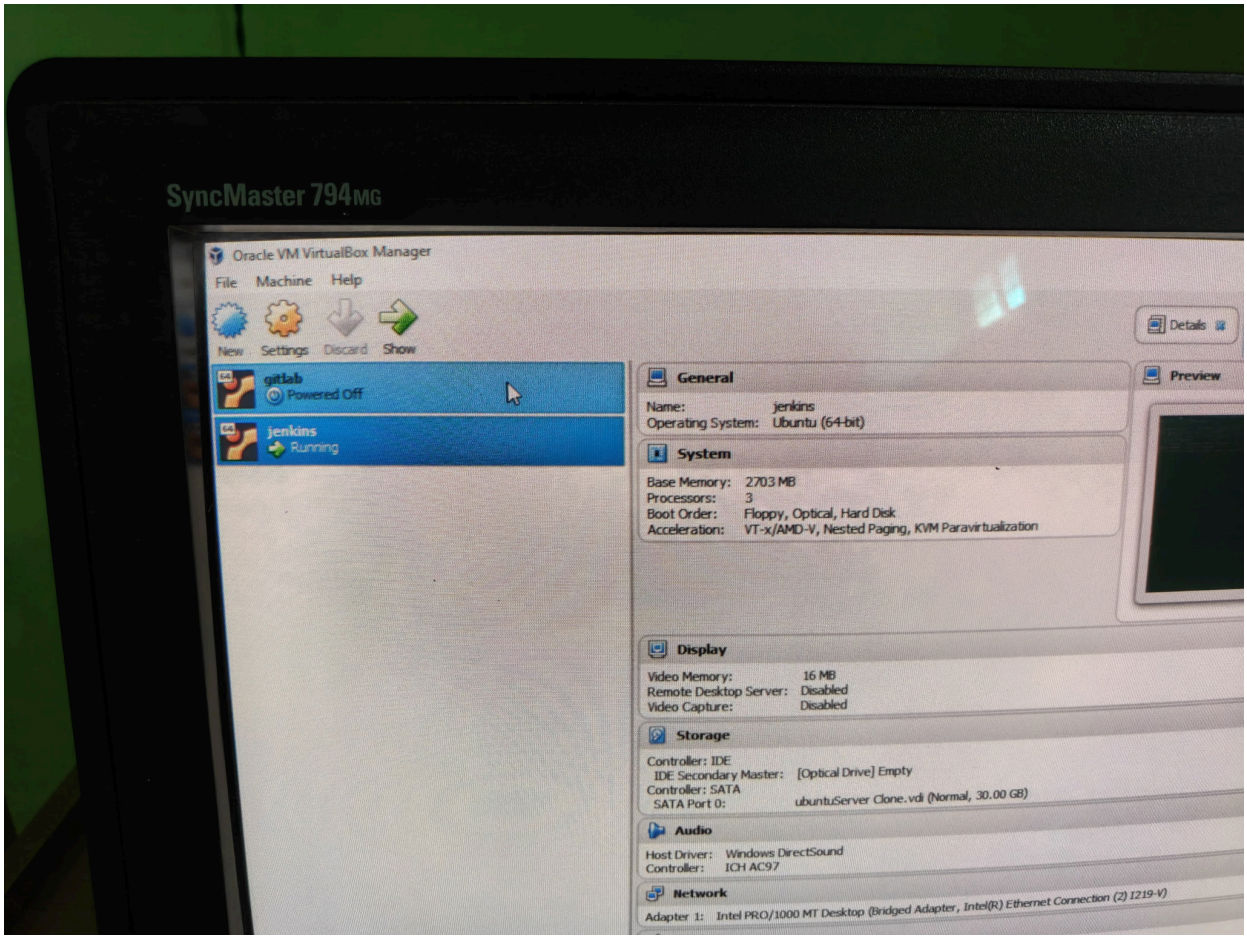
-> GitLab server IP address: 172.16.18.9:80**90**
GitLab Credentials: shiv:whatuhave & root:Chandigarh@$#130

-> Selenium grid address: 172.16.18.11:4444/grid/console

Link of Automation Repository:
https://github.com/ShivSahil/Automation/tree/master

Setting up Virtual MachineS on a Server (I am using my desktop as the host server.)

I have created two virtual machines, allocating appropriate RAM, CPU cores, and disk space to each. Both VMs are running Ubuntu Server.



Software Installed on Each Ubuntu Server:

- **SSH:** Installed on both servers to enable communication between them and allow remote access via my laptop.
- **Docker Engine and Docker Compose:** Installed for container management and orchestration.
- **Vi Editor:** Installed for editing configuration files directly on the servers.

Note: Jenkins container is based on Debian & gitLab on Ubuntu and both support Bash shell

Creating Jenkins Container

Below is the docker-compose.yml file used to create the Jenkins **container**.

```
shiv@shivubuntu:~/jenkins-data$ ls
docker-compose.yml  jenkins_home
shiv@shivubuntu:~/jenkins-data$ cat docker-compose.yml
version: '3'

services:

  jenkins:

    container_name: jenkins

    image: jenkins/jenkins

    ports:

    - "8080:8080"

    volumes:

     - "$PWD/jenkins_home:/var/jenkins_home"

    networks:

     - net

networks:

 net:
```

Docker file explanation

services:

jenkins:	We can give any name to newly created service.I have given “jenkins”

container_name: jenkins	This is the name of the container we have given to the service jenkins.

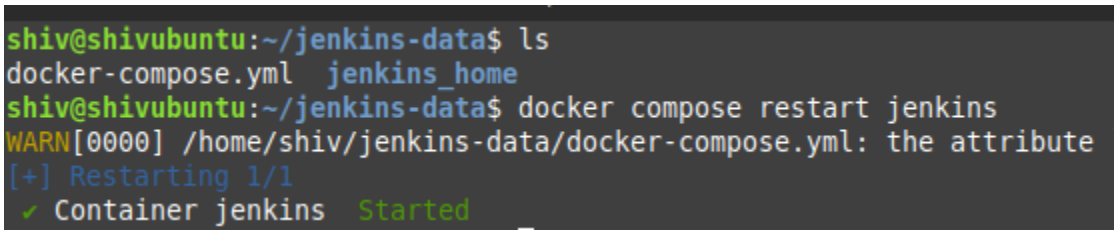
image: jenkins/jenkins	This is the name of the image which gets downloaded

ports: - "8080:8080"	We want to expose jenkin’s port 8080 to my VM’s internal port 8080. Later we will use this port to connect to jenkins

volumes: - "$PWD/jenkins_home:/var/jenkins_home"	All the information of jenkins container (/var/jenkins_home) will be saved in VM’s $PWD/jenkins_home

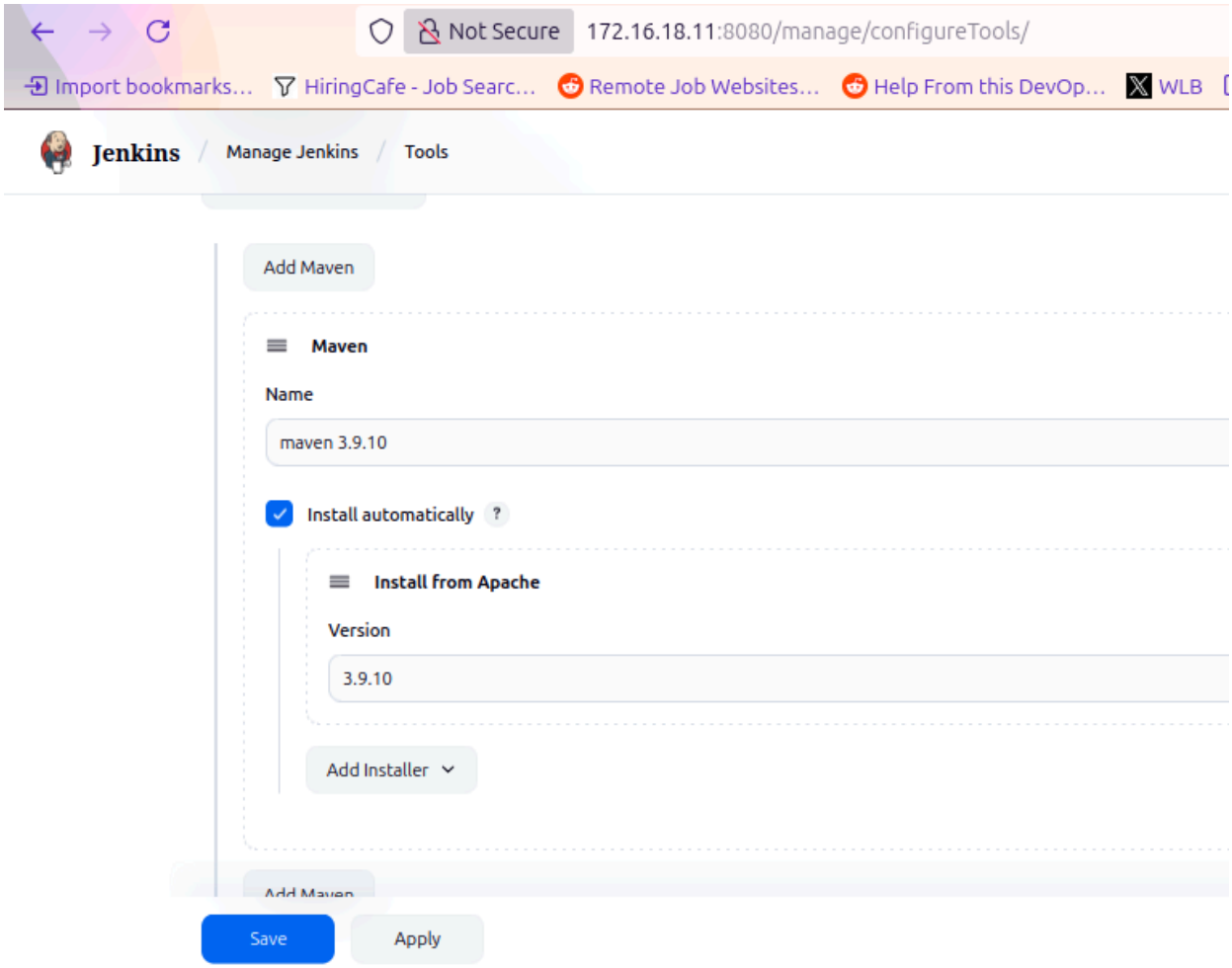
Running `docker compose up -d` will create or recreate the service(s) defined in the `docker-compose.yml` file

The command `docker compose restart jenkins` stops and then starts the Jenkins service.



Installing and Configuring Plugins on Jenkins

Once Jenkins is up and running, install & configure the **Maven Integration** plugin, **git** and the **Strict Crumb Issuer** plugin, which helps you trigger jobs remotely in a secure way.



Below is a sample **Bash script** I used to generate a crumb token, which enables triggering Jenkins jobs remotely. I also used a variation of this script inside **Git hooks**.

```
#!/bin/bash

# Retrieve crumb token

crumb=$(curl -u "shiv:1452A" -s
'http://jenkins:8080/crumbIssuer/api/xml?xpath=concat(//crumbReq
uestField,":",//crumb)')
```

```
# Trigger the Jenkins job

curl -u "username:password" -H "$crumb" -X POST
'http://172.16.18.11:8080/job/JobName/build?delay=0sec'
```

Creating GitLab Container

Below is the docker-compose.yml file used to create the GitLab **container**.

```
shiv@shivubuntu:~/jenkins-data$ cat docker-compose.yml
version: '3'

services:

  git:

    container_name: git-server

    image: 'gitlab/gitlab-ee:latest'

    hostname: 'gitlab.xtremebots.com'

    ports:

      - '8090:80'

    volumes:

      - '$GITLAB_HOME/config:/etc/gitlab'

      - '$GITLAB_HOME/logs:/var/log/gitlab'

      - '$GITLAB_HOME/data:/var/opt/gitlab'

    shm_size: '256m'

    networks:

      - net

networks:

 net:
```

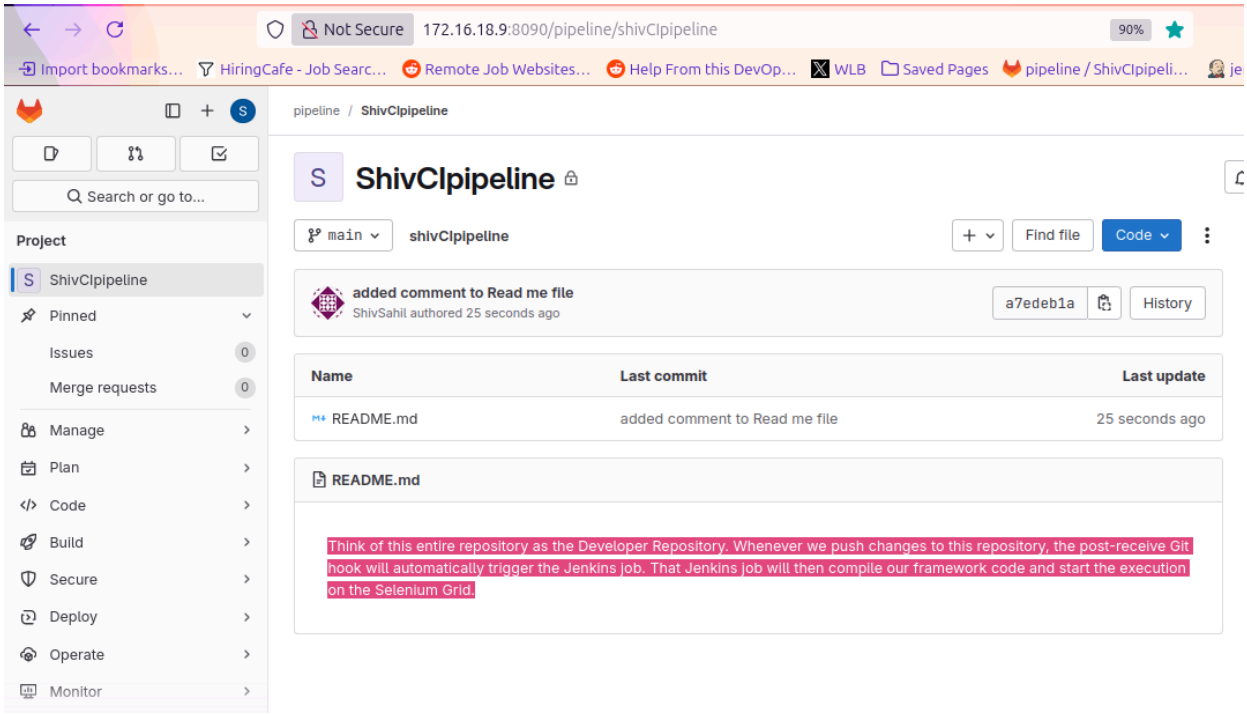
Running `docker compose up -d` will create or recreate the service(s) defined in the `docker-compose.yml` file

The command `docker compose restart git` stops and then starts the Git service.

```
shiv@shivubuntu:~/jenkins-data$ docker compose restart git
WARN[0000] The "GITLAB_HOME" variable is not set. Defaulting to a blank string.
WARN[0000] The "GITLAB_HOME" variable is not set. Defaulting to a blank string.
WARN[0000] The "GITLAB_HOME" variable is not set. Defaulting to a blank string.
WARN[0000] /home/shiv/jenkins-data/docker-compose.yml: the attribute `version` i
[+] Restarting 1/1
 ✔ Container git-server  Started
```

Adding a repository to gitlab

Think of this entire repository as the Developer Repository. Whenever we push changes to this repository, the `post-receive` Git hook will automatically trigger the Jenkins job. That Jenkins job will then compile our framework code and start the execution on the Selenium Grid.



⭐Adding GitHook to GitLab Container

Heart of CI Pipeline: I placed a Git hook file, known as post-receive, in the repository inside the GitLab container.

The post-receive hook is a server-side Git hook that runs after the repository has received a push.

Here's how I did it:

- I entered the GitLab container using:

  docker exec -it git-server bash

```
exit
shiv@shivubuntu:~/jenkins-data$ docker exec -it git-server bash
root@gitlab:/# cd var/opt/gitlab/git-data/repositories/\@hashed/6b/86
```

- Then I navigated to the .git directory of the repository named **ShivCIpipeline**:

  cd /var/opt/gitlab/git-data/repositories/@hashed/6b/86

```
root@gitlab:/var/opt/gitlab/git-data/repositories/@hashed/6b/86/6b86b273ff34fce19d6b804eff5a3f5747ada4eaa22f1d49c01e52ddb7875b4b.git/custom_hooks# cat post-receive
#!/bin/bash


if ! [ -t 0 ]; then
  read -a ref
fi



IFS='/' read -ra REF <<< "${ref[2]}"
branch="${REF[2]}"


if [ "$branch" == "main" ]; then
  crumb=$(curl -u "shiv:1234" -s 'http://172.16.18.11:8080/crumbIssuer/api/xml?xpath=concat(//crumbRequestField,":",//crumb)')
  curl -u "shiv:1234" -H "$crumb" -X POST http://172.16.18.11:8080/job/gitAutomation/build?delay=0sec


  if [ $? -eq 0 ]; then
    echo "*** Ok"
  else
```

Below is the content of the post-receive hook script:

```
#!/bin/bash

# Read the ref information from stdin
if ! [ -t 0 ]; then
  read -a ref
fi

# Extract the branch name from the ref
IFS='/' read -ra REF <<< "${ref[2]}"
branch="${REF[2]}"

# If the pushed branch is 'main', trigger the Jenkins job
if [ "$branch" == "main" ]; then
  crumb=$(curl -u "shiv:1234" -s
'http://172.16.18.11:8080/crumbIssuer/api/xml?xpath=concat(//cru
mbRequestField,":",//crumb)')
  curl -u "shiv:1234" -H "$crumb" -X POST
'http://172.16.18.11:8080/job/gitAutomation/build?delay=0sec'

  if [ $? -eq 0 ]; then
     echo "*** Ok"
  else
     echo "*** Error"
```

```
  fi

fi
```

Code says that if branch is main then we are going to generate the crumb.	`if [ "$branch" == "main" ]; then`
	
	`  crumb=$(curl -u "jenkinUser:jenkinsPassword" -s 'http://jenkins:8080/crumbIssuer/api/xml?xpath=concat(//crumbRequestFiel d,":",//crumb)')`

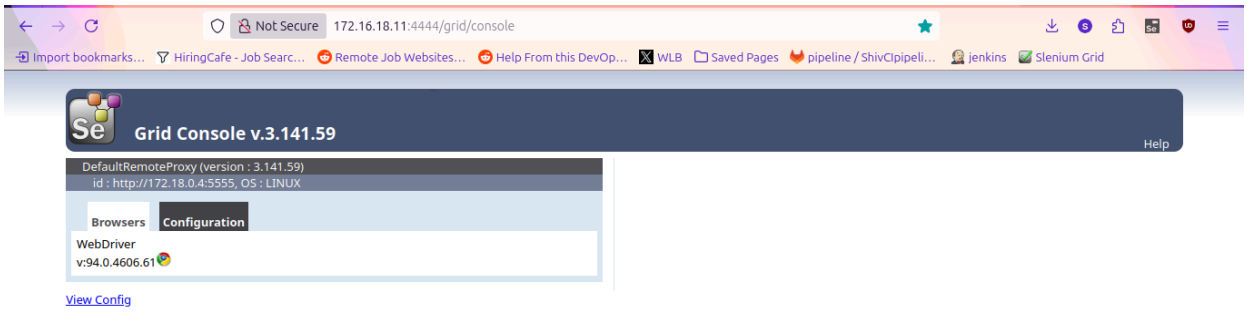
After crumb is generated, maven-job will be triggered	`curl -u "jenkinsUser:jenkinsPassword" -H "$crumb" -X POST http://jenkins:8080/job/maven-job/build?delay=0sec`

Creating Selenium Hub and Node

Due to lack of space I have installed only chrome

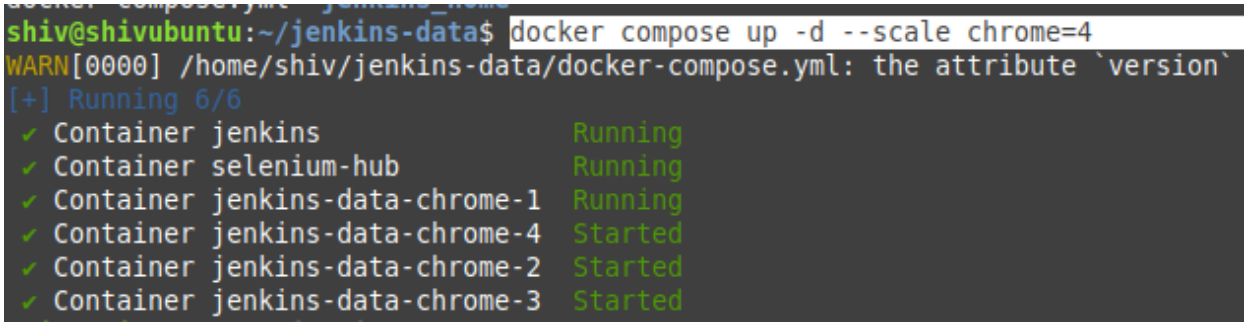
```
shiv@shivubuntu:~/jenkins-data$ cat docker-compose.yml
version: '3'

services:

  jenkins:
    container_name: jenkins
    image: jenkins/jenkins
    ports:
      - "8080:8080"
    volumes:
      - "$PWD/jenkins_home:/var/jenkins_home"
    networks:
      - net

  hub:
    image: selenium/hub:3.141.59
    container_name: selenium-hub
    ports:
      - "4444:4444"
    networks:
      - net

  chrome:
    image: selenium/node-chrome:3.141.59
    depends_on:
      - hub
    environment:
      - HUB_HOST=hub
    networks:
      - net

networks:
  net:
```

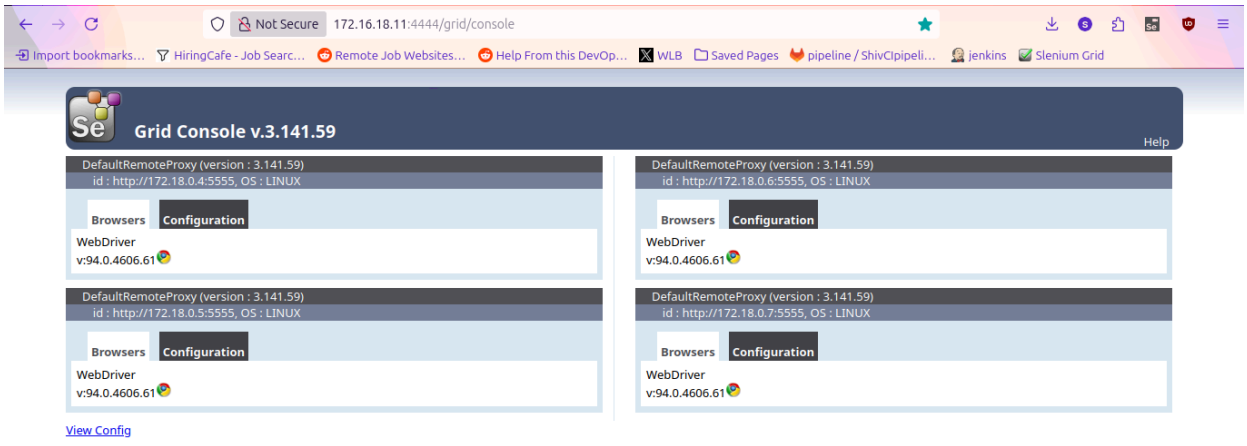
```
shiv@shivubuntu:~/jenkins-data$ docker ps
CONTAINER ID   IMAGE                          COMMAND                  CREATED              STATUS              PORTS
 NAMES
0696a0c0cdfb   selenium/node-chrome:3.141.59   "/opt/bin/entry_poin…"   About a minute ago   Up About a minute
 jenkins-data-chrome-1
7eee71b47385   selenium/hub:3.141.59           "/opt/bin/entry_poin…"   About a minute ago   Up About a minute   0.0.0.0:4444->4444/tcp, [::]:4444->4444/tcp
 selenium-hub
```



In case more instances of the google chrome or any other browser are needed then use

```
docker compose up -d --scale chrome=4
```

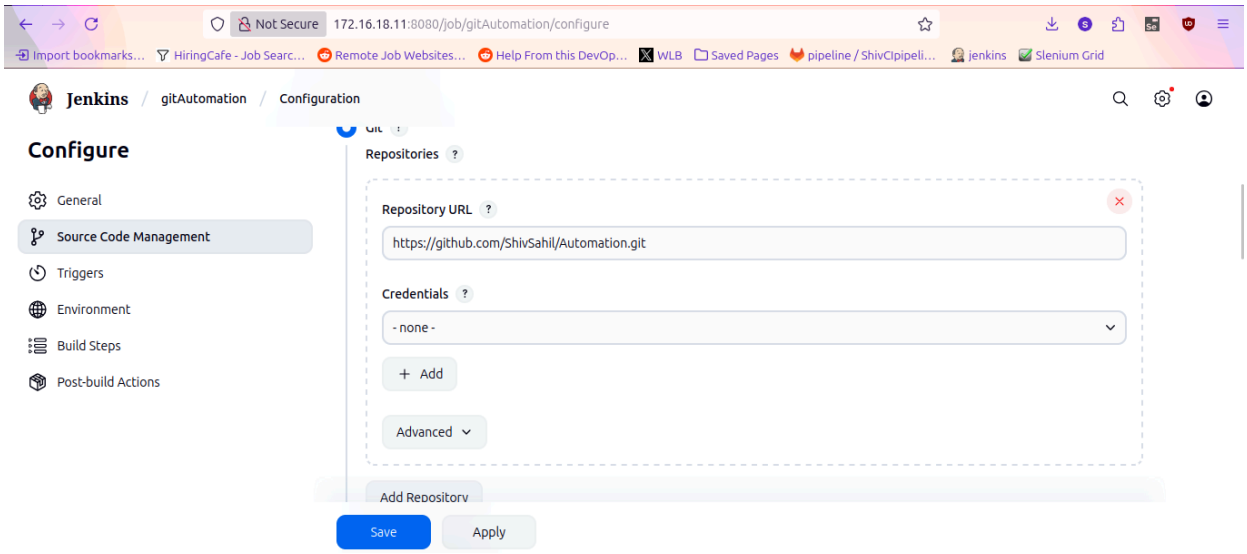


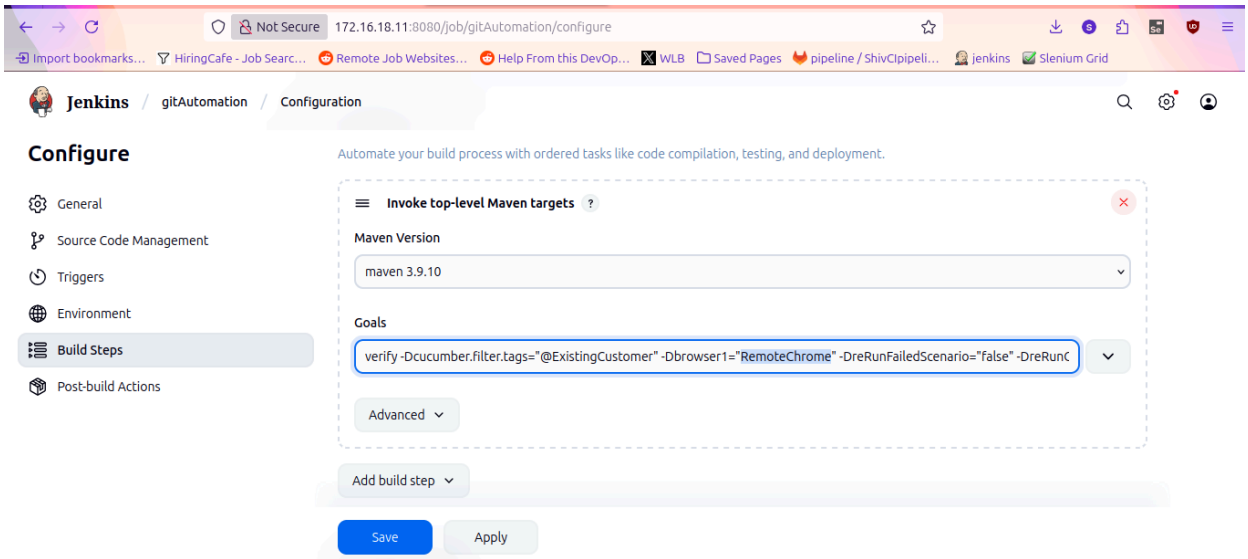


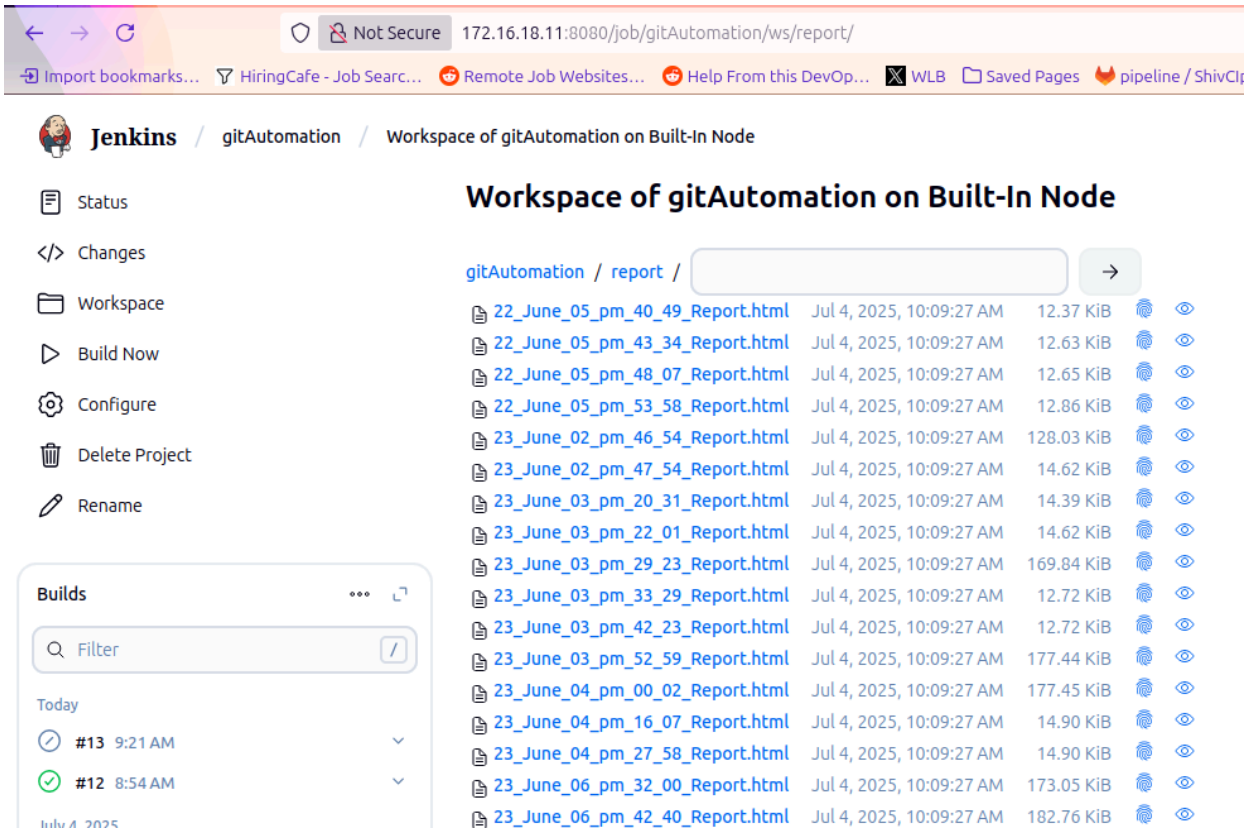
Creation of Jenkins job

This job is triggered by the Git hooks configured in GitLab.
When it starts, it pulls the below Test Automation code from GitHub (not from GitLab):

https://github.com/ShivSahil/Automation/tree/master






Not Secure
172.16.18.11:8080/job/gitAutomation/ws/report/
Import bookmarks...
HiringCafe - Job Searc...
Remote Job Websites...
Help From this DevOp...
WLB
Saved Pages
pipeline / ShivCI
Jenkins
gitAutomation
Workspace of gitAutomation on Built-In Node
Status
Changes
Workspace
Build Now
Configure
Delete Project
Rename
Builds
Filter
Today
#13 9:21 AM
#12 8:54 AM
Workspace of gitAutomation on Built-In Node
gitAutomation / report /
22_June_05_pm_40_49_Report.html Jul 4, 2025, 10:09:27 AM 12.37 KiB
22_June_05_pm_43_34_Report.html Jul 4, 2025, 10:09:27 AM 12.63 KiB
22_June_05_pm_48_07_Report.html Jul 4, 2025, 10:09:27 AM 12.65 KiB
22_June_05_pm_53_58_Report.html Jul 4, 2025, 10:09:27 AM 12.86 KiB
23_June_02_pm_46_54_Report.html Jul 4, 2025, 10:09:27 AM 128.03 KiB
23_June_02_pm_47_54_Report.html Jul 4, 2025, 10:09:27 AM 14.62 KiB
23_June_03_pm_20_31_Report.html Jul 4, 2025, 10:09:27 AM 14.39 KiB
23_June_03_pm_22_01_Report.html Jul 4, 2025, 10:09:27 AM 14.62 KiB
23_June_03_pm_29_23_Report.html Jul 4, 2025, 10:09:27 AM 169.84 KiB
23_June_03_pm_33_29_Report.html Jul 4, 2025, 10:09:27 AM 12.72 KiB
23_June_03_pm_42_23_Report.html Jul 4, 2025, 10:09:27 AM 12.72 KiB
23_June_03_pm_52_59_Report.html Jul 4, 2025, 10:09:27 AM 177.44 KiB
23_June_04_pm_00_02_Report.html Jul 4, 2025, 10:09:27 AM 177.45 KiB
23_June_04_pm_16_07_Report.html Jul 4, 2025, 10:09:27 AM 14.90 KiB
23_June_04_pm_27_58_Report.html Jul 4, 2025, 10:09:27 AM 14.90 KiB
23_June_06_pm_32_00_Report.html Jul 4, 2025, 10:09:27 AM 173.05 KiB
23_June_06_pm_42_40_Report.html Jul 4, 2025, 10:09:27 AM 182.76 KiB