Capstone project documentation

1. React application:

It is a web application built using React, which is a JavaScript library for building user interfaces. Developed and maintained by Facebook, React allows developers to create large web applications that can update and render efficiently in response to data changes.

React applications are known for their performance, modularity, and ease of maintenance.

They are widely used for developing modern web applications, including social media platforms, e-commerce sites, dashboards, and more.

2. Node.js :

1. Node.js is a JavaScript-based platform for server-side and networking applications.

2. It is a runtime environment that allows developers to run JavaScript code on the server side.

3. Node.js is commonly used for building web servers, real-time applications (like chat applications), APIs, and microservices, among other types of applications.

4. Node.js is a software platform for scalable server-side and networking applications.

5. Node.js applications are designed to maximize throughput and efficiency.

6. Node.js internally uses the Google V8 JavaScript engine to execute code.

Developer gives

1. .json file →it denotes write docker file for node.js
2. .txt file → it denotes write docker file for python
3. .jar file → it denotes write docker file for java
4. .xml file → it denotes write docker file for java.

→ But developer does not give .jar file. We can create .jar file using build tool [maven /gradel]

1. Maven [pom.xml]
2. Gradel [build.gradle]

File format	Docker file	Package management tool
package. json	Node.js	Npm [Node Package Manager]
requirement.txt	python	pip
app.jar pom.xml	java	Maven [pom.xml] Gradel [build.gradle]

Suppose package.json file [default file name] not given in github repo, it means it's an already build application. For run that application, we need web server [nginx or Apache]

difference between docker file and docker compose file

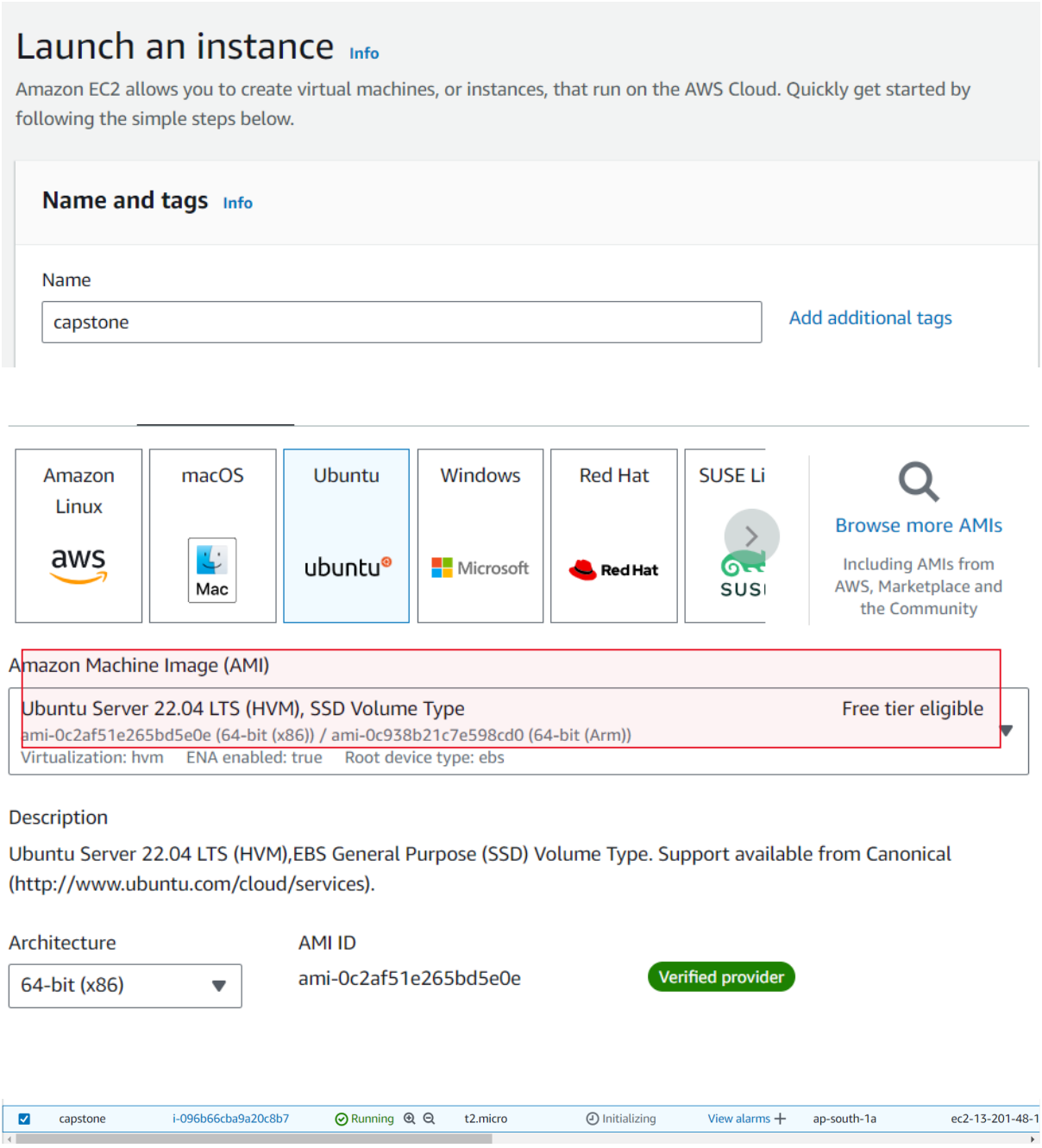
	Docker file	**Docker compose file**
purpose	Defines how to build a single Docker image.	Defines how to run multi-container applications.it is used for running multiple containers.
File type	Text file	.yaml / yml file
syntax	FROM, RUN, COPY, etc	YAML syntax to define services, networks, and volumes.
Content	It Contains instructions to set up an environment inside a container	It contains service definitions and configurations for running containers

By understanding these differences, you can effectively use both Docker file and Docker Compose together to build and deploy complex Docker-based applications.

Installing the necessary software's & services for this task:

1. **Git**
2. **Docker**
3. **Docker compose**
4. **Java**
5. **Jenkins**

Before install git. we have to launch an instance. select ubuntu



Connect that instance

1. Git installation:

sudo apt update

sudo apt install git -y

git --version

sudo apt update

```
Get:39 http://security.ubuntu.com/ubuntu noble-security/main amd64 c-n-f Metadata [3668 B]
Get:40 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [249 kB]
Get:41 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [108 kB]
Get:42 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Components [8632 B]
Get:43 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [9220 B]
Get:44 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 Packages [208 kB]
Get:45 http://security.ubuntu.com/ubuntu noble-security/restricted Translation-en [40.7 kB]
Get:46 http://security.ubuntu.com/ubuntu noble-security/restricted amd64 c-n-f Metadata [420 B]
Get:47 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Packages [10.6 kB]
Get:48 http://security.ubuntu.com/ubuntu noble-security/multiverse Translation-en [2808 B]
Get:49 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 Components [208 B]
Get:50 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [344 B]
Fetched 28.2 MB in 6s (5082 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
47 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-172-31-32-39:~$ sudo apt install git -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.43.0-1ubuntu7.1).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 47 not upgraded.
ubuntu@ip-172-31-32-39:~$ git --version
git version 2.43.0
ubuntu@ip-172-31-32-39:~$
```

Clone the given repo

git clone https://github.com/sriram-R-krishnan/devops-build

```
ubuntu@ip-172-31-32-39:~$ git clone https://github.com/sriram-R-krishnan/devops-build
Cloning into 'devops-build'...
remote: Enumerating objects: 21, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 21 (delta 0), reused 0 (delta 0), pack-reused 18
Receiving objects: 100% (21/21), 720.09 KiB | 15.00 MiB/s, done.
ubuntu@ip-172-31-32-39:~$
```

```
ubuntu@ip-172-31-32-39:~$ mkdir caspstone1
ubuntu@ip-172-31-32-39:~$ cd caspstone1/
ubuntu@ip-172-31-32-39:~/caspstone1$
```

2. Docker installation:

sudo apt update

sudo apt install docker.io

sudo docker version

sudo usermod -aG docker ubuntu

sudo systemctl status docker

sudo systemctl start docker

after that, I got some error regarding permission.so I give below commands. search the below commands in ChatGPT

sudo usermod -aG docker ubuntu

newgrp docker

sudo systemctl start docker

sudo systemctl enable docker

just check, docker can pull the image from docker hub. This command not needed for this task

sudo docker run hello-world

sudo apt update

```
ubuntu@ip-172-31-32-39:~$ sudo apt update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu noble-security InRelease
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
47 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

sudo apt install docker.io

```
ubuntu@ip-172-31-32-39:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-buildx docker-compose-v2 docker-doc rinse zfs-
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io pigz runc ubuntu-fan
0 upgraded, 8 newly installed, 0 to remove and 47 not upgraded.
Need to get 76.8 MB of archives.
After this operation, 289 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 pigz amd64 2.8-1 [65.6 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 bridge-utils amd64 1.7.1-1ubuntu2 [33.9 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 runc amd64 1.1.12-0ubuntu3 [8599 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 containerd amd64 1.7.12-0ubuntu4 [38.6 MB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 dns-root-data all 2023112702~willsync1 [4450
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/main amd64 dnsmasq-base amd64 2.90-2build2 [375 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 docker.io amd64 24.0.7-0ubuntu4 [29.1 MB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/universe amd64 ubuntu-fan all 0.12.16 [35.2 kB]
Fetched 76.8 MB in 1s (56.2 MB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 67739 files and directories currently installed.)
```

sudo docker version

```
ubuntu@ip-172-31-32-39:~$ sudo docker version
Client:
 Version:           24.0.7
 API version:       1.43
 Go version:        go1.22.2
 Git commit:        24.0.7-0ubuntu4
 Built:             Wed Apr 17 20:08:25 2024
 OS/Arch:           linux/amd64
 Context:           default

Server:
 Engine:
  Version:          24.0.7
  API version:      1.43 (minimum version 1.12)
  Go version:       go1.22.2
  Git commit:       24.0.7-0ubuntu4
  Built:            Wed Apr 17 20:08:25 2024
  OS/Arch:          linux/amd64
  Experimental:     false
 containerd:
  Version:          1.7.12
  GitCommit:
 runc:
  Version:          1.1.12-0ubuntu3
  GitCommit:
 docker-init:
```

sudo usermod -aG docker ubuntu

```
  GitCommit:
ubuntu@ip-172-31-32-39:~$ sudo usermod -aG docker ubuntu
ubuntu@ip-172-31-32-39:~$
```

sudo systemctl status docker

```
ubuntu@ip-172-31-32-39:~$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
     Active: active (running) since Thu 2024-08-08 06:50:22 UTC; 28min ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 2556 (dockerd)
      Tasks: 8
     Memory: 32.6M (peak: 33.2M)
        CPU: 442ms
     CGroup: /system.slice/docker.service
             └─2556 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.soc
```

sudo systemctl start docker

```
ubuntu@ip-172-31-46-99:~$ sudo usermod -aG docker ubuntu
ubuntu@ip-172-31-46-99:~$ newgrp docker
ubuntu@ip-172-31-46-99:~$ sudo systemctl start docker
ubuntu@ip-172-31-46-99:~$ sudo systemctl enable docker
ubuntu@ip-172-31-46-99:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:1408fec50309afee38f3535383f5b09419e6dc0925bc69891e79d84cc4cdcec6
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.
```

```
To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/

ubuntu@ip-172-31-46-99:~$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED         SIZE
hello-world   latest    d2c94e258dcb   15 months ago   13.3kB
```

docker file:

- In our task given application was already built.
- So, we did not need to write node.js docker file. Just write docker file for nginx server.
- For running the application, we need web server. So, write docker file for nginx web server.
- Because in that server we have to run an already built application.
- **Docker file written in the folder devops-build [in given github this folder available].**
- **so, get into the folder [cd devops-build]**

Explanation of the Dockerfile:

1. **FROM nginx:latest**:
    - This line specifies the base image to use for the Docker image. In this case, it pulls the latest version of the official Nginx image from Docker Hub.
2. **WORKDIR /usr/share/nginx/html**:
    - This sets the working directory inside the container to /usr/share/nginx/html. This is the default directory where Nginx serves static files (like HTML, CSS, and JavaScript).
3. **COPY build/ .**:
    - This copies the contents of the build/ directory [this dir in the given repo]on your host machine into the current working directory in the container (/usr/share/nginx/html). This typically contains the static files that your web application will serve.
4. **EXPOSE 80**:
    - This line informs Docker that the container will listen on port 80 at runtime. Port 80 is the default HTTP port, so this is where Nginx will serve the web application.
5. **CMD ["nginx", "-g", "daemon off;"]**:
    - This command runs Nginx in the foreground (instead of in the background as a daemon), which is necessary for Docker containers so that the container doesn't exit immediately after starting. The command starts Nginx and keeps it running.

vi dockerfile

```
FROM nginx:latest

WORKDIR /usr/share/nginx/html

COPY build/ .

EXPOSE 80

CMD ["nginx", "-g", "daemon off;"]
```

Convert docker file into docker image:

docker build -t nginximage . **[nginximage -we give that image name for our choice]. This command used for build docker image**

```
ubuntu@ip-172-31-46-99:~$ vi dockerfile
ubuntu@ip-172-31-46-99:~$ cd devops-build/
ubuntu@ip-172-31-46-99:~/devops-build$ vi dockerfile
ubuntu@ip-172-31-46-99:~/devops-build$ docker build -t nginximage .
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  3.415MB
Step 1/5 : FROM nginx:latest
 ---> a72860cb95fd
Step 2/5 : WORKDIR /usr/share/nginx/html
 ---> Using cache
 ---> 834607514a93
Step 3/5 : COPY build/ .
 ---> 9a869321475c
Step 4/5 : EXPOSE 80
 ---> Running in 2db9e8a75aea
Removing intermediate container 2db9e8a75aea
 ---> e144c57ce161
Step 5/5 : CMD ["nginx", "-g", "daemon off;"]
 ---> Running in 60c950b6a23f
Removing intermediate container 60c950b6a23f
 ---> 6ba82f9c1767
Successfully built 6ba82f9c1767
Successfully tagged nginximage:latest
ubuntu@ip-172-31-46-99:~/devops-build$ docker images
```

docker images **[check image created or not]**

```
ubuntu@ip-172-31-46-99:~/devops-build$ docker images
REPOSITORY    TAG             IMAGE ID       CREATED          SIZE
nginximage    latest          6ba82f9c1767   25 seconds ago   190MB
nodeimage     latest          d44d4a7a0b69   27 minutes ago   188MB
<none>        <none>          ae52f8af4119   27 minutes ago   81.1MB
nginx         latest          a72860cb95fd   6 weeks ago      188MB
hello-world   latest          d2c94e258dcb   15 months ago    13.3kB
node          12.2.0-alpine   f391dabf9dce   5 years ago      77.7MB
```

docker run -d --name mynignxcointainer -p 80:80 nginximage **[it is used to create and run a container from a Docker image (nginximage) with specific options]**

```
ubuntu@ip-172-31-46-99:~/devops-build$ docker run -d --name mynginxcointainer -p 80:80 nginximage
1c82a7a8e8f05bcd8fce44d66e1bdae6fcd163163fe1cad573c78c81a4adb25f
```

docker ps **[check container created or not]**

```
ubuntu@ip-172-31-46-99:~/devops-build$ docker ps
CONTAINER ID   IMAGE         COMMAND                  CREATED          STATUS          PORTS                                      NAMES
1c82a7a8e8f0   nginximage    "/docker-entrypoint.…"   5 seconds ago    Up 3 seconds    0.0.0.0:80->80/tcp, :::80->80/tcp          mynginxcointainer
ae71713569e8   nodeimage     "/docker-entrypoint.…"   25 minutes ago   Up 25 minutes   0.0.0.0:8000->80/tcp, :::8000->80/tcp      nginxcointainer
```

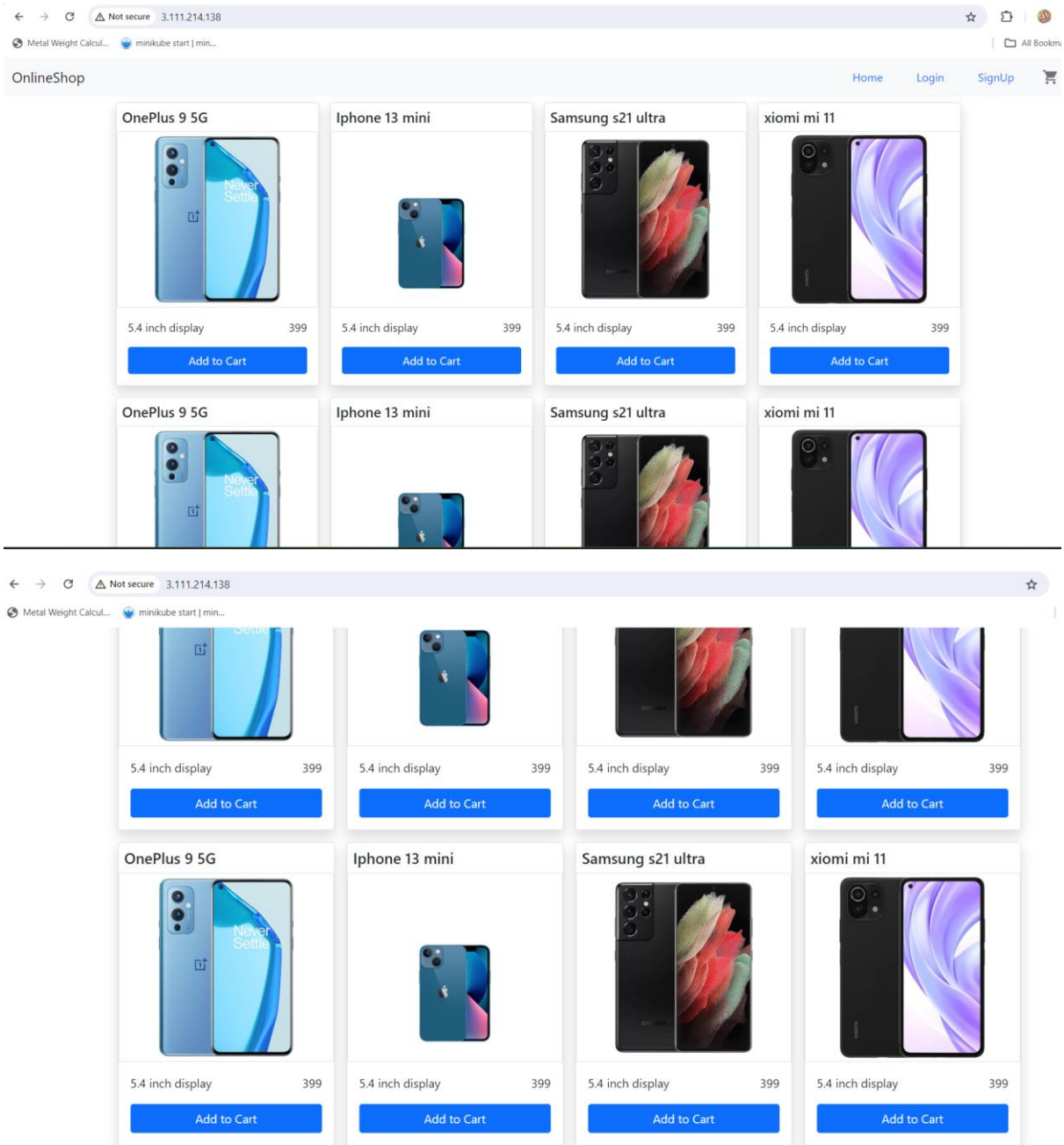
Open the port no :80 in ec2. Copy and paste the ip address

i-01676b853cc1e86fb (capstone)

Filter rules

Name	Security group rule ID	Port range	Protocol	Source	Security groups
–	sgr-0efb4cd406c9f01f8	8000	TCP	0.0.0.0/0	launch-wizard-3
–	sgr-04d7a00c6fbb578c4	22	TCP	0.0.0.0/0	launch-wizard-3
–	sgr-0d46262f30ca2f723	80	TCP	0.0.0.0/0	launch-wizard-3

Output page of given application:



3. Docker compose installation:

sudo apt update

sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose

docker-compose --version

sudo apt update [update ubuntu machine]

```
ubuntu@ip-172-31-46-99:~$ sudo apt update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1930 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [341 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [17.8 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [2301 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [396 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1110 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [25.9 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [43.3 kB]
Get:13 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1712 kB]
Get:14 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [283 kB]
Get:15 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [13.1 kB]
Get:16 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [2226 kB]
Get:17 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [383 kB]
Fetched 11.0 MB in 3s (3177 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
29 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose [install d.c]

```
ubuntu@ip-172-31-46-99:~$ sudo curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docke
r-compose
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0
100 12.1M  100 12.1M    0     0  8033k      0  0:00:01  0:00:01 --:--:-- 12.3M
```

sudo chmod +x /usr/local/bin/docker-compose [gave permission for the d.c]

```
ubuntu@ip-172-31-46-99:~$ sudo chmod +x /usr/local/bin/docker-compose
ubuntu@ip-172-31-46-99:~$ docker-compose --version
docker-compose version 1.29.2, build 5becea4c
```

docker-compose --version [to check docker installed or not / docker version]

Docker compose file for node js:

get into folder devops-build. Then write this

```
ubuntu@ip-172-31-46-99:~$ cd devops-build/
ubuntu@ip-172-31-46-99:~/devops-build$ vi docker-compose.yaml
```

vi docker-compose.yaml

```
version: '3'

services:
 webcointainer:
  image: nginximage
  ports:
   - "80:80"
```

version: '3'

services:

webcointainer:

image: nginximage [already we created docker image. give that image name]

ports:

- “80:80”

Give permission for this file:

Chmod 777 docker-compose.yaml [give permission for the docker-compose.yaml file]

```
ubuntu@ip-172-31-46-99:~/devops-build$ vi docker-compose.yaml
ubuntu@ip-172-31-46-99:~/devops-build$ chmod 777 docker-compose.yaml
```

Execute the docker-compose.yaml :

docker-compose up -d [execute the docker compose file]

```
ubuntu@ip-172-31-46-99:~/devops-build$ docker-compose up -d
Starting devops-build_webcointainer_1 ... done
```

docker ps [to check container created or not]

```
ubuntu@ip-172-31-46-99:~/devops-build$ docker ps
CONTAINER ID   IMAGE        COMMAND                  CREATED          STATUS          PORTS                                   NAMES
4e1ec10be231   nginximage   "/docker-entrypoint.…"   10 minutes ago   Up 10 seconds   0.0.0.0:80->80/tcp, :::80->80/tcp   devops-build_webcointainer_1
```

Bash scripting:

- **build.sh**
- **deploy.sh**

build.sh:

- The build.sh file is a shell script commonly used in software projects to automate the build process.
- It typically contains a series of commands that compile the source code.

- This script is often used in environments where consistent and repeatable builds are essential, such as in continuous integration (CI) pipelines.

Build.sh file:

```
ubuntu@ip-172-31-46-99:~/devops-build$ vi build.sh
```

#!/bin/bash

docker stop mynignxcointainer

docker rm mynignxcointainer

docker build -t nginximage .

docker run -d --name mynignxcointainer -p 80:80 nginximage

```
#!/bin/bash
docker stop mynignxcointainer
docker rm mynignxcointainer
docker build -t nginximage .
docker run -d --name mynignxcointainer -p 80:80 nginximage
```

Your docker login details [create the new docker hub account]

```
ubuntu@ip-172-31-46-99:~/devops-build$ docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com/ to create
 one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is required for organizations using SSO.
Learn more at https://docs.docker.com/go/access-tokens/

Username: sharmi2504
Password:
WARNING! Your password will be stored unencrypted in /home/ubuntu/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
ubuntu@ip-172-31-46-99:~/devops-build$
```

chmod 777 build.sh [Give permission for build.sh file] **./ build.sh**[execute the file]

```
ubuntu@ip-172-31-46-99:~/devops-build$ chmod 777 build.sh
ubuntu@ip-172-31-46-99:~/devops-build$ ./build.sh
Error response from daemon: No such container: mynignxcointainer
Error response from daemon: No such container: mynignxcointainer
DEPRECATED: The legacy builder is deprecated and will be removed in a future release.
            Install the buildx component to build images with BuildKit:
            https://docs.docker.com/go/buildx/

Sending build context to Docker daemon  3.417MB
Step 1/5 : FROM nginx:latest
 ---> a72860cb95fd
Step 2/5 : WORKDIR /usr/share/nginx/html
 ---> Using cache
 ---> 834607514a93
Step 3/5 : COPY build/ .
 ---> Using cache
 ---> 9a869321475c
Step 4/5 : EXPOSE 80
 ---> Using cache
 ---> e144c57ce161
Step 5/5 : CMD ["nginx", "-g", "daemon off;"]
 ---> Using cache
 ---> 6ba82f9c1767
Successfully built 6ba82f9c1767
Successfully tagged nginximage:latest
fedf999a071f7699cf6fab3e87a58fa474188936e4f45ffae171d1c6c6e128a5
```

docker images

```
ubuntu@ip-172-31-46-99:~/devops-build$ docker images
REPOSITORY    TAG             IMAGE ID       CREATED          SIZE
nginximage    latest          6ba82f9c1767   5 days ago       190MB
nodeimage     latest          d44d4a7a0b69   5 days ago       188MB
<none>        <none>          ae52f8af4119   5 days ago       81.1MB
nginx         latest          a72860cb95fd   7 weeks ago      188MB
hello-world   latest          d2c94e258dcb   15 months ago    13.3kB
node          12.2.0-alpine   f391dabf9dce   5 years ago      77.7MB
```

docker ps

```
ubuntu@ip-172-31-46-99:~/devops-build$ docker ps
CONTAINER ID   IMAGE        COMMAND                  CREATED         STATUS         PORTS                                   NAMES
36ad9bca04c2   nginximage   "/docker-entrypoint.…"   7 seconds ago   Up 5 seconds   0.0.0.0:80->80/tcp, :::80->80/tcp   mynignxcointainer
```

deploy.sh:

- The deploy.sh file is not a standard or widely recognized filename in most deployment processes or tools.
- It could be a custom script or configuration file specific to a particular system, project, or deployment process

deploy.sh file:

```
ubuntu@ip-172-31-46-99:~/devops-build$ vi deploy.sh
```

#!/bin/bash

docker tag nginximage sharmi2504/dev

docker push sharmi2504/dev

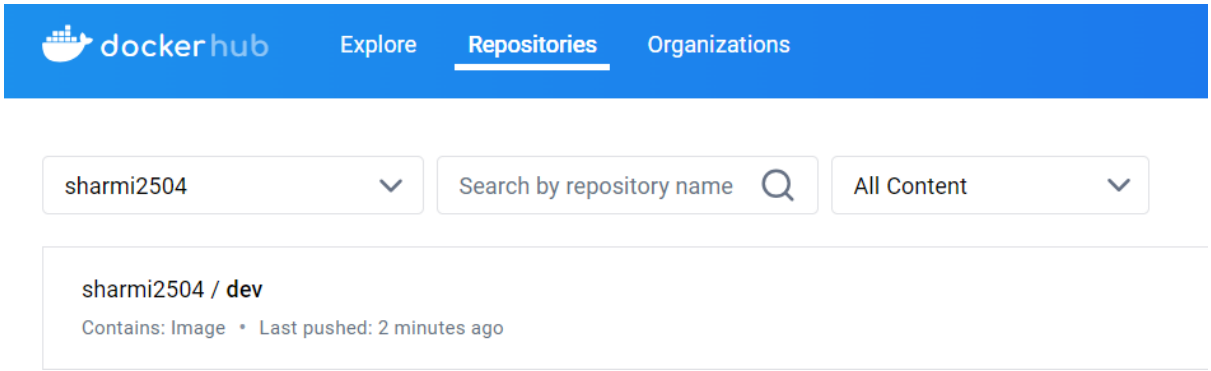
```
docker tag nginximage sharmi2504/dev
docker push sharmi2504/dev
```

chmod 777 deploy.sh

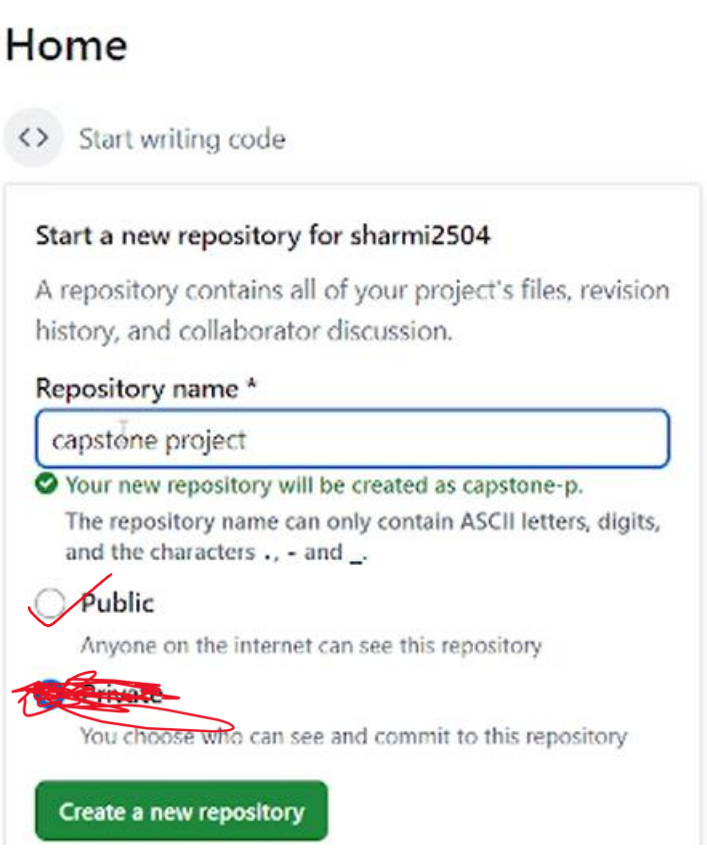
./deploy.sh

```
ubuntu@ip-172-31-46-99:~/devops-build$ vi deploy.sh
ubuntu@ip-172-31-46-99:~/devops-build$ chmod 777 deploy.sh
ubuntu@ip-172-31-46-99:~/devops-build$ ./deploy.sh
Using default tag: latest
The push refers to repository [docker.io/sharmi2504/dev]
98d3ba44771c: Pushed
60e72fbb314e: Pushed
599e8de62018: Pushed
09581b9299a2: Pushed
a39383416a22: Pushed
a6355e7844d5: Pushed
fcfa12460e7d: Mounted from library/nginx
e0781bc8667f: Mounted from library/nginx
latest: digest: sha256:e010ed627fcccc8c7c4934494bb9150a6390bd48a6be0c3a4877e330a0ceba27 size: 1988
ubuntu@ip-172-31-46-99:~/devops-build$
```

```
ubuntu@ip-172-31-46-99:~/devops-build$ docker images
REPOSITORY          TAG             IMAGE ID       CREATED          SIZE
nginximage          latest          6ba82f9c1767   5 days ago       190MB
sharmi2504/dev      latest          6ba82f9c1767   5 days ago       190MB
```



Create new repo in git:



Give public repo

```
ubuntu@ip-172-31-46-99:~/devops-build$ cd
```

After create repo. you get this https link. copy and paste it in clone command

```
ubuntu@ip-172-31-46-99:~$ git clone https://github.com/sharmi2504/capstone-project.git
Cloning into 'capstone-project'...
warning: You appear to have cloned an empty repository.
```

```
ubuntu@ip-172-31-46-99:~$ git clone https://github.com/sharmi2504/capstone-project.git
Cloning into 'capstone-project'...
warning: You appear to have cloned an empty repository.
ubuntu@ip-172-31-46-99:~$ ls
capstone-project  devops-build  dockerfile
ubuntu@ip-172-31-46-99:~$ cd devops-build/
ubuntu@ip-172-31-46-99:~/devops-build$
```

Here capstone-project directory created repo in GitHub. get into that dir . them move that all files from devops-build to capstone-project dir . in our own repo we push all our files.

```
ubuntu@ip-172-31-46-99:~$ cd devops-build/
ubuntu@ip-172-31-46-99:~/devops-build$ ls
build  build.sh  deploy.sh  docker-compose.yaml  dockerfile
ubuntu@ip-172-31-46-99:~/devops-build$ mv * /home/ubuntu/capstone-project
ubuntu@ip-172-31-46-99:~/devops-build$ cd ..
ubuntu@ip-172-31-46-99:~$ cd capstone-project/
ubuntu@ip-172-31-46-99:~/capstone-project$ ls
build  build.sh  deploy.sh  docker-compose.yaml  dockerfile
ubuntu@ip-172-31-46-99:~/capstone-project$
```

```
create mode 100644 build/favicon.ico
create mode 100644 build/index.html
create mode 100644 build/logo192.png
create mode 100644 build/logo512.png
create mode 100644 build/manifest.json
create mode 100644 build/robots.txt
create mode 100644 build/static/css/main.cf5c13c5.css
create mode 100644 build/static/css/main.cf5c13c5.css.map
create mode 100644 build/static/js/787.2f5360e2.chunk.js
create mode 100644 build/static/js/787.2f5360e2.chunk.js.map
create mode 100644 build/static/js/main.f1c48542.js
create mode 100644 build/static/js/main.f1c48542.js.LICENSE.txt
create mode 100644 build/static/js/main.f1c48542.js.map
create mode 100755 deploy.sh
create mode 100755 docker-compose.yaml
create mode 100644 dockerfile
```

git add . [add all files in our repo]

git commit -m “add all files”

```
ubuntu@ip-172-31-46-99:~/capstone-project$ git add .
ubuntu@ip-172-31-46-99:~/capstone-project$ git commit -m "add all files"
[main (root-commit) 13ef6f8] add all files
 Committer: Ubuntu <ubuntu@ip-172-31-46-99.ap-south-1.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 19 files changed, 198 insertions(+)
 create mode 100755 build.sh
 create mode 100644 build/_redirects
 create mode 100644 build/asset-manifest.json
 create mode 100644 build/favicon.ico
 create mode 100644 build/index.html
 create mode 100644 build/logo192.png
 create mode 100644 build/logo512.png
 create mode 100644 build/manifest.json
 create mode 100644 build/robots.txt
 create mode 100644 build/static/css/main.cf5c13c5.css
```

All files move in that repo [capstone-project]. then created dev branch. Inside dev branch we add all files.

Create dev branch in git:

git checkout -b dev **[** Create dev branch**]**

```
ubuntu@ip-172-31-46-99:~/capstone-project$ git checkout -b dev
Switched to a new branch 'dev'
```

```
ubuntu@ip-172-31-46-99:~/capstone-project$ ls
build  build.sh  deploy.sh  docker-compose.yaml  dockerfile
```

git add . [add all files in our repo]

git commit -m “add all files”

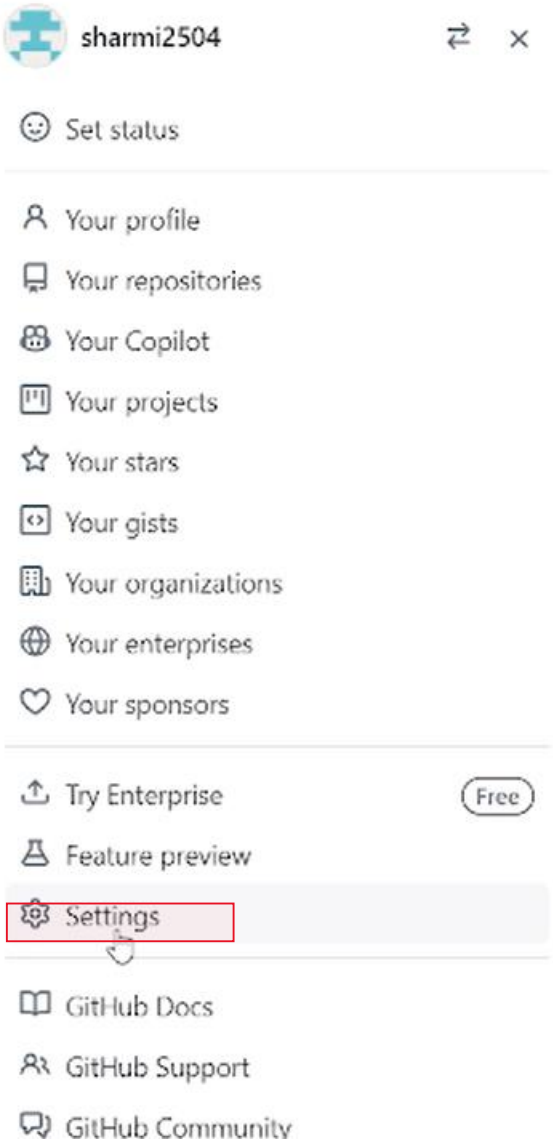
```
ubuntu@ip-172-31-46-99:~/capstone-project$ git add .
ubuntu@ip-172-31-46-99:~/capstone-project$ git commit -m "add all files"
On branch dev
nothing to commit, working tree clean
```

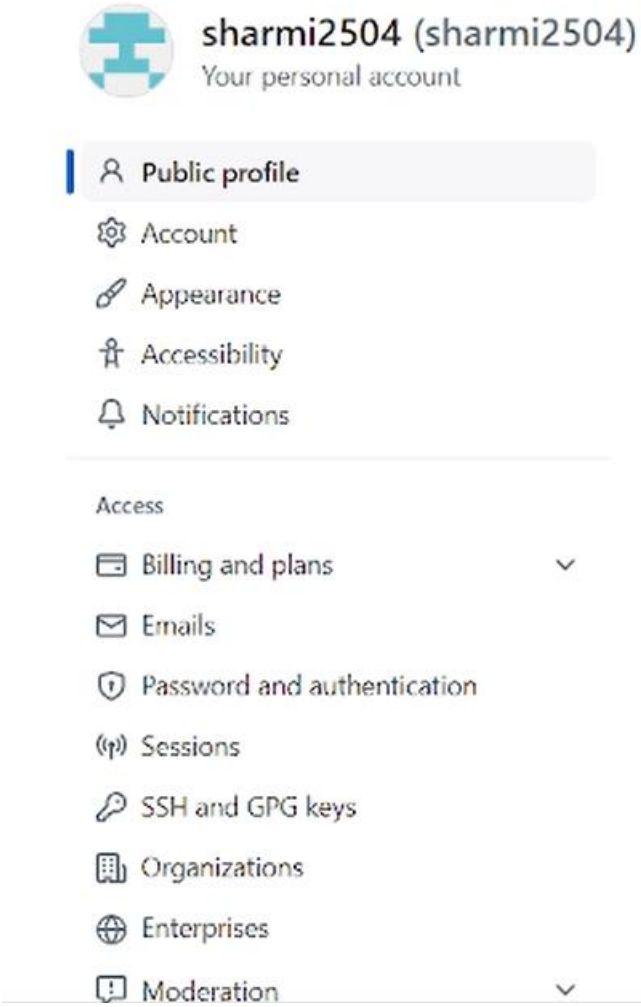
git push -u origin dev [push all files in our repo]

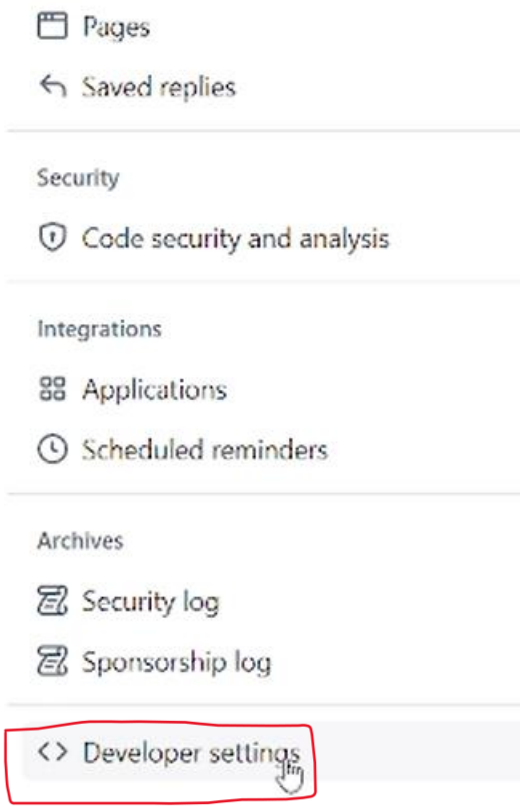
For password we generate token [that steps given below]

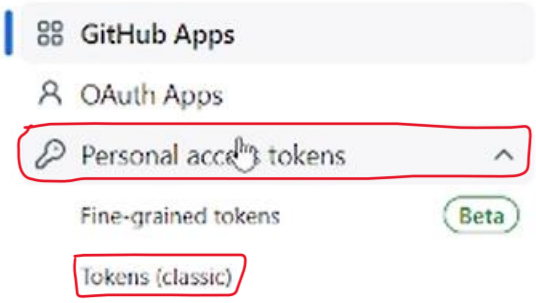
```
ubuntu@ip-172-31-46-99:~/capstone-project$ git push -u origin dev
Username for 'https://github.com': sharmi2504
Password for 'https://sharmi2504@github.com':
Enumerating objects: 25, done.
Counting objects: 100% (25/25), done.
Compressing objects: 100% (24/24), done.
Writing objects: 100% (25/25), 720.19 KiB | 4.56 MiB/s, done.
Total 25 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/sharmi2504/capstone-project.git
 * [new branch]      dev -> dev
Branch 'dev' set up to track remote branch 'dev' from 'origin'.
ubuntu@ip-172-31-46-99:~/capstone-project$
```

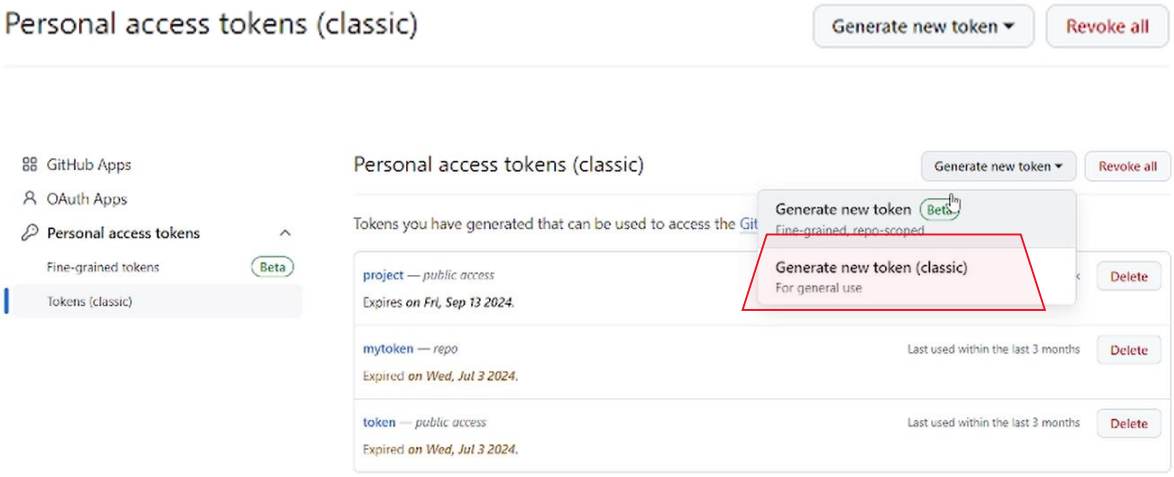
For git password, we generate token


sharmi2504
Set status
Your profile
Your repositories
Your Copilot
Your projects
Your stars
Your gists
Your organizations
Your enterprises
Your sponsors
Try Enterprise
Free
Feature preview
Settings
GitHub Docs
GitHub Support
GitHub Community


sharmi2504 (sharmi2504)
Your personal account
Public profile
Account
Appearance
Accessibility
Notifications
Access
Billing and plans
Emails
Password and authentication
Sessions
SSH and GPG keys
Organizations
Enterprises
Moderation


Pages
Saved replies
Security
Code security and analysis
Integrations
Applications
Scheduled reminders
Archives
Security log
Sponsorship log
Developer settings


GitHub Apps
OAuth Apps
Personal access tokens
Fine-grained tokens
Beta
Tokens (classic)


Personal access tokens (classic)
Generate new token
Revoke all
GitHub Apps
OAuth Apps
Personal access tokens
Fine-grained tokens
Beta
Tokens (classic)
Personal access tokens (classic)
Generate new token
Revoke all
Tokens you have generated that can be used to access the Git
Generate new token
Beta
Fine-grained, repo-scoped
Generate new token (classic)
For general use
project — public access
Expires on Fri, Sep 13 2024.
Delete
mytoken — repo
Last used within the last 3 months
Expired on Wed, Jul 3 2024.
Delete
token — public access
Last used within the last 3 months
Expired on Wed, Jul 3 2024.
Delete

New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Note

capstone project

What's this token for?

Expiration *

30 days — The token will expire on Fri, Sep 13 2024

Select scopes

Scopes define the access for personal tokens. Read more about OAuth scopes.

repo	Full control of private repositories
repo:status	Access commit status
repo_deployment	Access deployment status
public_repo	Access public repositories
repo:invite	Access repository invitations
security_events	Read and write security events

Note

capstone project

What's this token for?

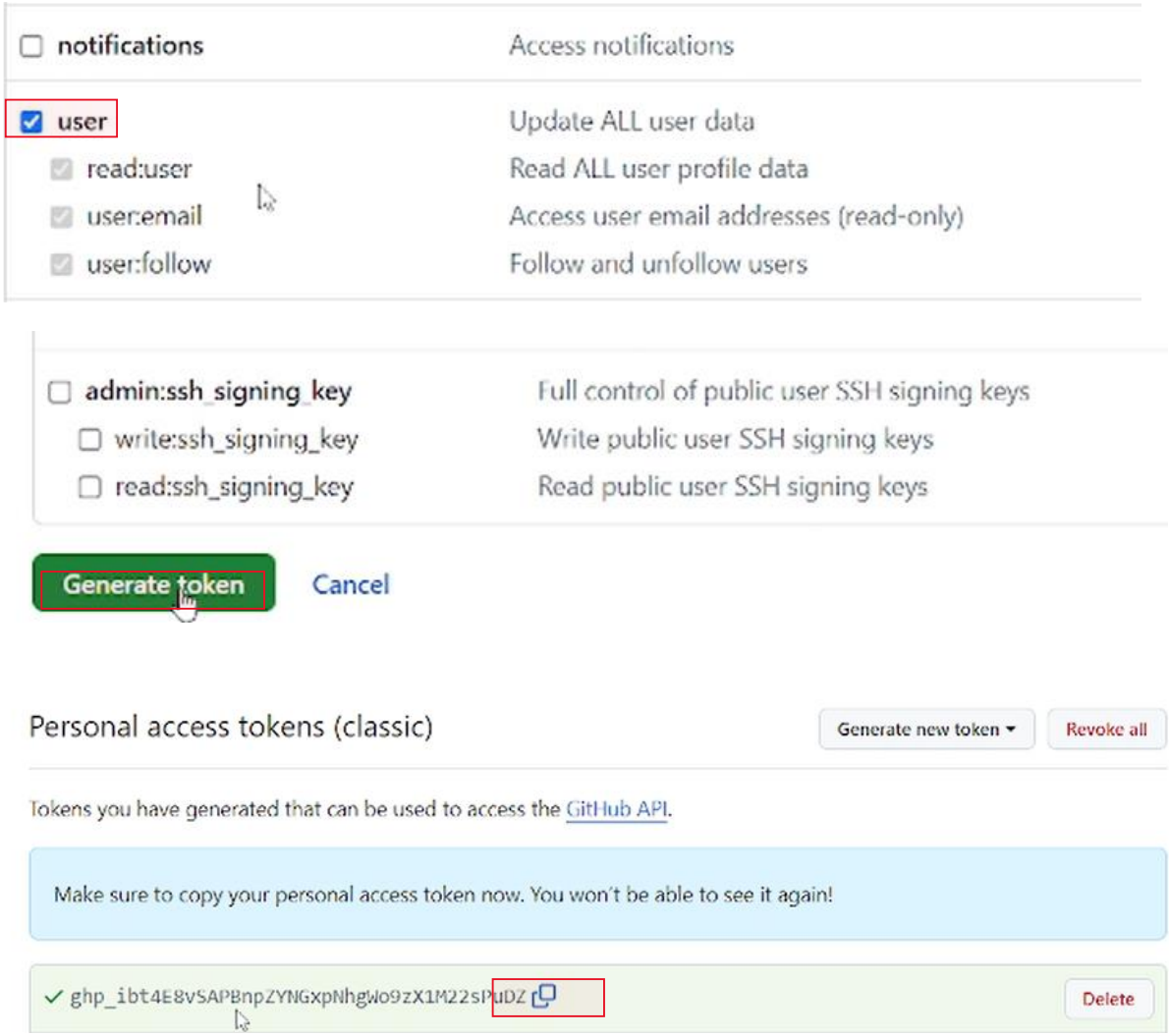
Expiration *

30 days — The token will expire on Fri, Sep 13 2024

Select scopes

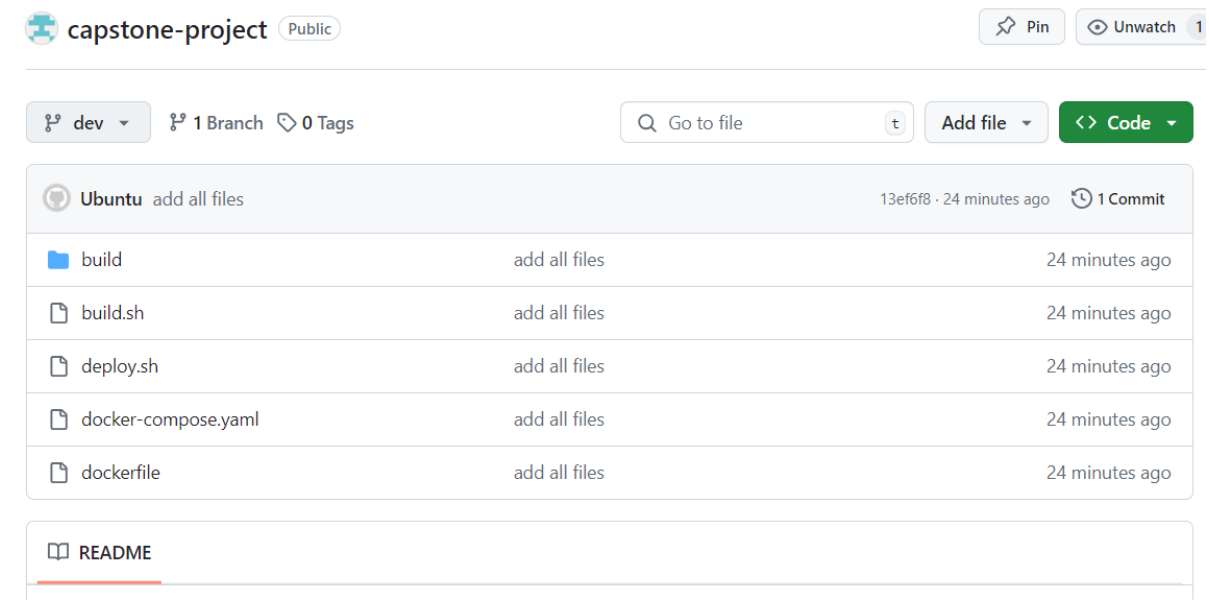
Scopes define the access for personal tokens. Read more about OAuth scopes.

repo	Full control of private repositories
repo:status	Access commit status
repo_deployment	Access deployment status
public_repo	Access public repositories
repo:invite	Access repository invitations
security_events	Read and write security events



Copy and paste the token in password

```
ubuntu@ip-172-31-46-99:~/capstone-project$ git push -u origin dev
Username for 'https://github.com': sharmi2504
Password for 'https://sharmi2504@github.com':
Enumerating objects: 25, done.
Counting objects: 100% (25/25), done.
Compressing objects: 100% (24/24), done.
Writing objects: 100% (25/25), 720.19 KiB | 4.56 MiB/s, done.
Total 25 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/sharmi2504/capstone-project.git
 * [new branch]      dev -> dev
Branch 'dev' set up to track remote branch 'dev' from 'origin'.
ubuntu@ip-172-31-46-99:~/capstone-project$
```



All files pushed to our repo.

4. Jenkins installation:

sudo apt-get update

```
ubuntu@ip-172-31-46-99:~$ sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1941 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [343 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [17.7 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [2314 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [397 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1110 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [25.9 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [43.3 kB]
Fetched 6448 kB in 2s (3193 kB/s)
Reading package lists... Done
```

sudo apt-get install -y openjdk-11-jdk [install docker ,must install java. This is java installation]

```
ubuntu@ip-172-31-46-99:~$ sudo apt-get install -y openjdk-11-jdk
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  alsa-topology-conf alsa-ucm-conf at-spi2-core ca-certificates-java dconf-gsettings-backend dconf-service fontconfig-config fonts-dejavu-core
  fonts-dejavu-extra gsettings-desktop-schemas java-common libasound2 libasound2-data libatk-bridge2.0-0 libatk-wrapper-java libatk-wrapper-java-jni libatk1.0-0
  libatk1.0-data libatspi2.0-0 libavahi-client3 libavahi-common-data libavahi-common3 libcups2 libdconf1 libdrm-amdgpu1 libdrm-intel1 libdrm-nouveau2
  libdrm-radeon1 libfontconfig1 libfontenc1 libgif7 libgl1 libgl1-amber-dri libgl1-mesa-dri libglapi-mesa libglvnd0 libglx-mesa0 libglx0 libgraphite2-3
  libharfbuzz0b libice-dev libice6 libjpeg-turbo8 libjpeg8 liblcms2-2 liblvm15 libpciaccess0 libpcsclite1 libpthread-stubs0-dev libsensors-config libsensors5
  libsm-dev libsm6 libx11-dev libx11-xcb1 libxau-dev libxaw7 libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-present0 libxcb-randr0 libxcb-shape0 libxcb-shm0
  libxcb-sync1 libxcb-xfixes0 libxcb1-dev libxcomposite1 libxdmcp-dev libxfixes3 libxft2 libxi6 libxinerama1 libxkbfile1 libxmu6 libxpm4 libxrandr2 libxrender1
  libxshmfence1 libxt-dev libxt6 libxtst6 libxv1 libxxf86dga1 libxxf86vm1 openjdk-11-jdk-headless openjdk-11-jre openjdk-11-jre-headless session-migration
  x11-common x11-utils x11proto-dev xorg-sgml-doctools xtrans-dev
```

sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \

https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key

```
ubuntu@ip-172-31-46-99:~$ sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
--2024-08-10 06:37:02--  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
Resolving pkg.jenkins.io (pkg.jenkins.io)... 151.101.154.133, 2a04:4e42:24::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|151.101.154.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3175 (3.1K) [application/pgp-keys]
Saving to: '/usr/share/keyrings/jenkins-keyring.asc'

/usr/share/keyrings/jenkins-keyring.asc  100%[=================================================>]

2024-08-10 06:37:02 (22.5 MB/s) - '/usr/share/keyrings/jenkins-keyring.asc' saved [3175/3175]
```

echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \

https://pkg.jenkins.io/debian-stable binary/ | sudo tee \

/etc/apt/sources.list.d/jenkins.list > /dev/null

```
ubuntu@ip-172-31-46-99:~$ echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
```

sudo apt-get update

```
ubuntu@ip-172-31-46-99:~$ sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:4 https://pkg.jenkins.io/debian-stable binary/ InRelease
Get:5 https://pkg.jenkins.io/debian-stable binary/ Release [2044 B]
Get:6 https://pkg.jenkins.io/debian-stable binary/ Release.gpg [833 B]
Hit:7 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:8 https://pkg.jenkins.io/debian-stable binary/ Packages [27.6 kB]
Fetched 30.4 kB in 1s (36.9 kB/s)
Reading package lists... Done
```

sudo apt-get install Jenkins

```
ubuntu@ip-172-31-46-99:~$ sudo apt-get install jenkins
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  net-tools
The following NEW packages will be installed:
  jenkins net-tools
0 upgraded, 2 newly installed, 0 to remove and 13 not upgraded.
Need to get 91.4 MB of archives.
After this operation, 94.2 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 net-tools amd64
Get:2 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.462.1 [91.2 MB]
Fetched 91.4 MB in 6s (16.4 MB/s)
Selecting previously unselected package net-tools.
(Reading database ... 98349 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20181103.0eebece-1ubuntu5_amd64.deb ...
Unpacking net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Selecting previously unselected package jenkins.
Preparing to unpack .../jenkins_2.462.1_all.deb ...
Unpacking jenkins (2.462.1) ...
```

```
Get:1 http://ap-south-1.ec                          jammy/main amd64 net-tools amd64 1.60+git20181103.0eebece-1ubuntu5 [204 kB]
Get:2 https://pkg.jenkins.io/debian-stable binary/ jenkins 2.462.1 [91.2 MB]
Fetched 91.4 MB in 6s (16.4 MB/s)
Selecting previously unselected package net-tools.
(Reading database ... 98349 files and directories currently installed.)
Preparing to unpack .../net-tools_1.60+git20181103.0eebece-1ubuntu5_amd64.deb ...
Unpacking net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Selecting previously unselected package jenkins.
Preparing to unpack .../jenkins_2.462.1_all.deb ...
Unpacking jenkins (2.462.1) ...
Setting up net-tools (1.60+git20181103.0eebece-1ubuntu5) ...
Setting up jenkins (2.462.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/jenkins.service → /lib/systemd/system/jenkins.service.
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

sudo systemctl start Jenkins

```
ubuntu@ip-172-31-46-99:~$ sudo systemctl start jenkins
```

sudo systemctl status Jenkins

```
ubuntu@ip-172-31-46-99:~$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2024-08-10 06:41:04 UTC; 4min 53s ago
   Main PID: 4951 (java)
      Tasks: 37 (limit: 1120)
     Memory: 284.6M
        CPU: 17.882s
     CGroup: /system.slice/jenkins.service
             └─4951 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war
```

Above command not work. so run the below commands. below commands searched in ChatGPT. If suppose run the below commands ,before that install java.

(OR)

sudo apt-get update [update machine]

```
ubuntu@ip-172-31-46-99:~$ sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1941 kB]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [343 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [17.7 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [2314 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [397 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1110 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [25.9 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [43.3 kB]
Fetched 6448 kB in 2s (3193 kB/s)
Reading package lists... Done
```

sudo apt-get install -y openjdk-11-jdk [install docker, must install java. This is java installation]

```
ubuntu@ip-172-31-46-99:~$ sudo apt-get install -y openjdk-11-jdk
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  alsa-topology-conf alsa-ucm-conf at-spi2-core ca-certificates-java dconf-gsettings-backend dconf-service fontconfig-config fonts-dejavu-core
  fonts-dejavu-extra gsettings-desktop-schemas java-common libasound2 libasound2-data libatk-bridge2.0-0 libatk-wrapper-java libatk-wrapper-java-jni libatk1.0-0
  libatk1.0-data libatspi2.0-0 libavahi-client3 libavahi-common-data libavahi-common3 libcups2 libdconf1 libdrm-amdgpu1 libdrm-intel1 libdrm-nouveau2
  libdrm-radeon1 libfontconfig1 libfontenc1 libgif7 libgl1 libgl1-amber-dri libgl1-mesa-dri libglapi-mesa libglvnd0 libglx-mesa0 libglx0 libgraphite2-3
  libharfbuzz0b libice-dev libice6 libjpeg-turbo8 libjpeg8 liblcms2-2 libllvm15 libpciaccess0 libpcsclite1 libpthread-stubs0-dev libsensors-config libsensors5
  libsm-dev libsm6 libx11-dev libx11-xcb1 libxau-dev libxaw7 libxcb-dri2-0 libxcb-dri3-0 libxcb-glx0 libxcb-present0 libxcb-randr0 libxcb-shape0 libxcb-shm0
  libxcb-sync1 libxcb-xfixes0 libxcb1-dev libxcomposite1 libxdmcp-dev libxfixes3 libxft2 libxi6 libxinerama1 libxkbfile1 libxmu6 libxpm4 libxrandr2 libxrender1
  libxshmfence1 libxt-dev libxt6 libxtst6 libxv1 libxxf86dga1 libxxf86vm1 openjdk-11-jdk-headless openjdk-11-jre openjdk-11-jre-headless session-migration
  x11-common x11-utils x11proto-dev xorg-sgml-doctools xtrans-dev
```

curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee \

/usr/share/keyrings/jenkins-keyring.asc > /dev/null

sudo sh -c 'echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \

https://pkg.jenkins.io/debian-stable binary/ > \

/etc/apt/sources.list.d/jenkins.list'

```
ubuntu@ip-172-31-46-99:~$ curl -fsSL https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key | sudo tee \
  /usr/share/keyrings/jenkins-keyring.asc > /dev/null
sudo sh -c 'echo deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] \
  https://pkg.jenkins.io/debian-stable binary/ > \
  /etc/apt/sources.list.d/jenkins.list'
```

sudo apt-get update

```
ubuntu@ip-172-31-46-99:~$ sudo apt-get update
Hit:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Ign:4 https://pkg.jenkins.io/debian-stable binary/ InRelease
Hit:5 https://pkg.jenkins.io/debian-stable binary/ Release
Hit:6 http://security.ubuntu.com/ubuntu jammy-security InRelease
Reading package lists... Done
```

sudo apt-get install Jenkins

```
ubuntu@ip-172-31-46-99:~$ sudo apt-get install jenkins
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
jenkins is already the newest version (2.462.1).
0 upgraded, 0 newly installed, 0 to remove and 13 not upgraded.
```

sudo systemctl start Jenkins

```
ubuntu@ip-172-31-46-99:~$ sudo systemctl start jenkins
```

sudo systemctl status Jenkins

```
ubuntu@ip-172-31-46-99:~$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
     Loaded: loaded (/lib/systemd/system/jenkins.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2024-08-10 06:41:04 UTC; 4min 53s ago
   Main PID: 4951 (java)
      Tasks: 37 (limit: 1120)
     Memory: 284.6M
        CPU: 17.882s
     CGroup: /system.slice/jenkins.service
             └─4951 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war
```

```
ubuntu@ip-172-31-46-99:~/capstone-project$ vi jenkinsfile
```

Jenkins file:

```
pipeline {
    agent any

    environment {
        docker_username = credentials('docker-username-id') // Set up your Jenkins credentials
        docker_password = credentials('docker-password-id')
    }

    stages {
        stage('Build') {
            steps {
                // Build Docker image using build script file
                sh './build.sh'
            }
        }

        stage('Login') {
            steps {
                sh 'docker login -u $docker_username -p $docker_password'
            }
        }

        stage('Deploy') {
```

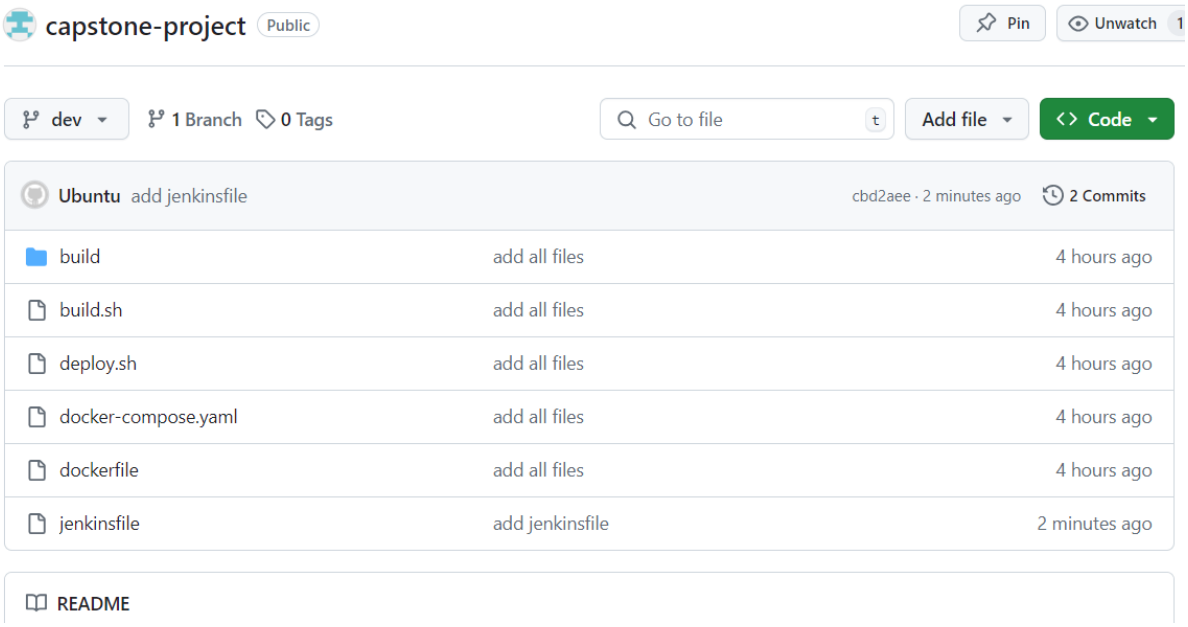
```
        steps {
            script {
                if (env.GIT_BRANCH == "origin/dev") {
                    sh './deploy.sh'
                }
            }
        }
    }
  }
}
```

Add the Jenkins file to github . use the below commands.

git add jenkinsfile[add Jenkins files in our repo]

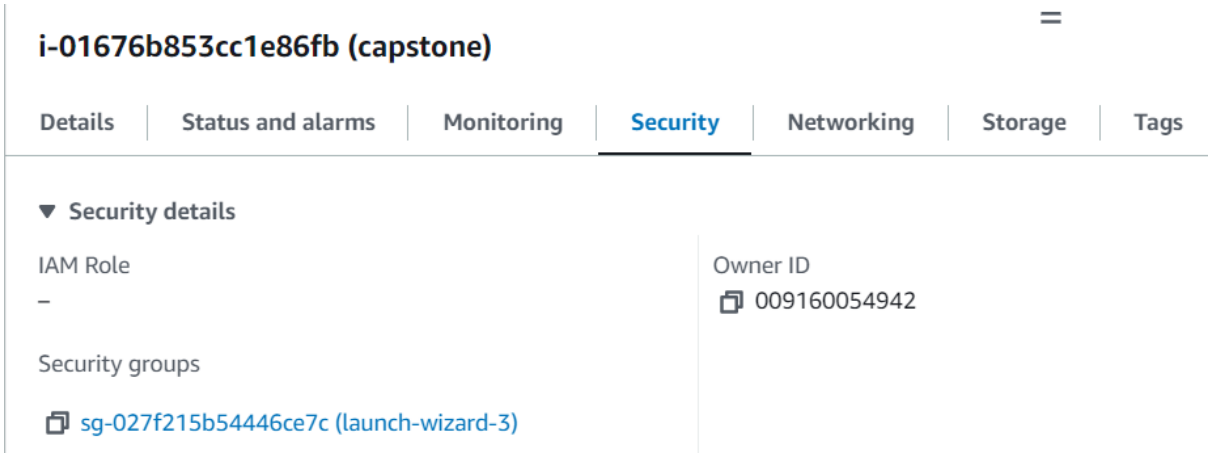
git commit -m “add jenkinsfiles”

```
ubuntu@ip-172-31-46-99:~/capstone-project$ git add jenkinsfile
ubuntu@ip-172-31-46-99:~/capstone-project$ git commit -m "add jenkinsfile"
[dev cbd2aee] add jenkinsfile
 Committer: Ubuntu <ubuntu@ip-172-31-46-99.ap-south-1.compute.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 1 file changed, 29 insertions(+)
 create mode 100644 jenkinsfile
ubuntu@ip-172-31-46-99:~/capstone-project$ git push -u origin dev
Username for 'https://github.com': sharmi2504
Password for 'https://sharmi2504@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Compressing objects: 100% (3/3), done.
Writing objects: 100% (3/3), 452 bytes | 452.00 KiB/s, done.
```

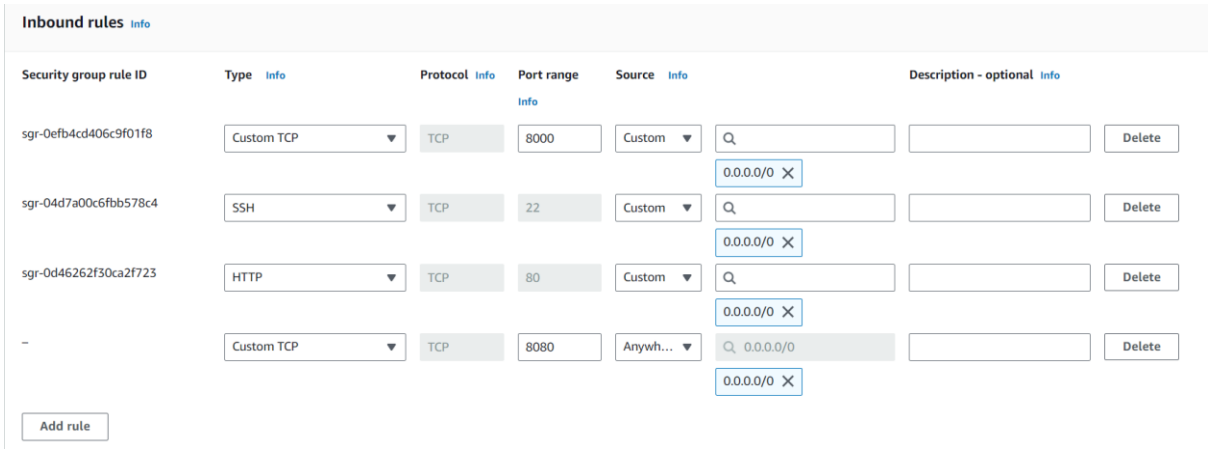


Setup Jenkins dashboard:

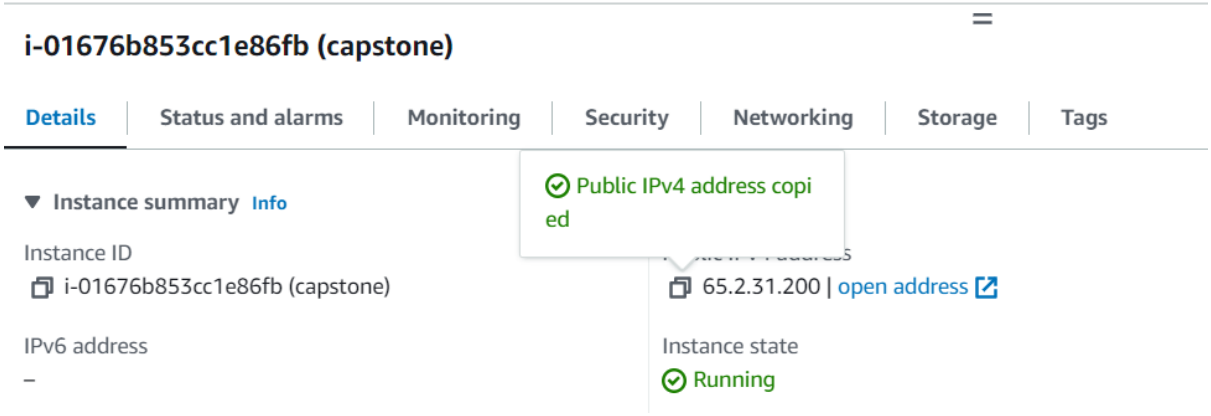
Add port no:8080 for access the Jenkins browser. Because it is a GUI tool



Click edit inbound rule

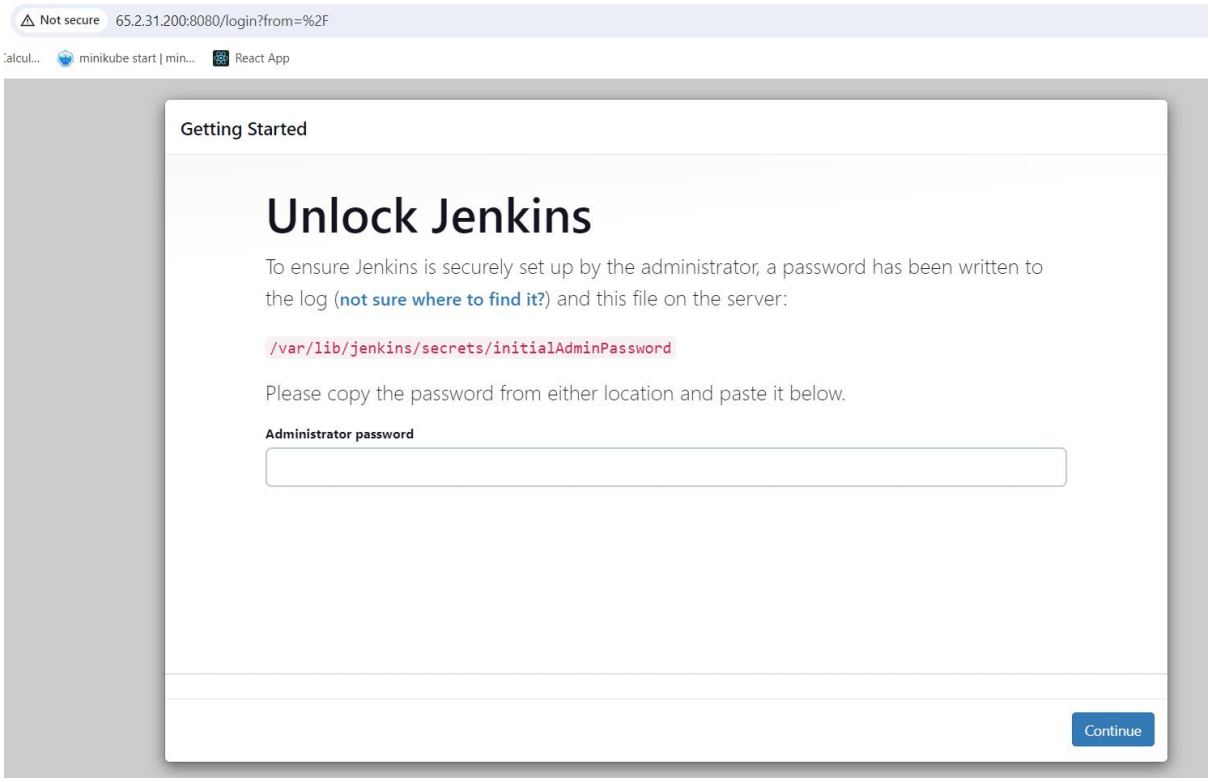


Click add rule → save rules. Copy ip address. Paste the address [65.2.31.200:8080]. give the Jenkins port number.



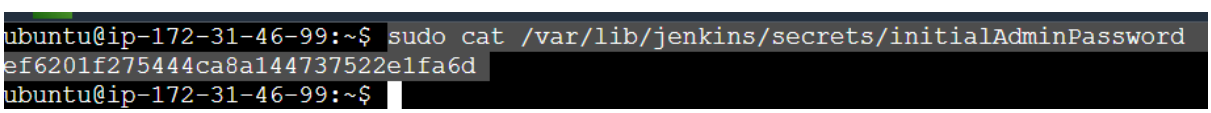
Copy ip address. Paste the address [65.2.31.200:8080]. give the Jenkins port number. Jenkins page opened.

Open Jenkins in browser:



Get above password

sudo cat /var/lib/jenkins/secrets/initialAdminPassword [get password for Jenkins]



ef6201f275444ca8a144737522e1fa6d [copy the password in above]

Getting Started

Unlock Jenkins

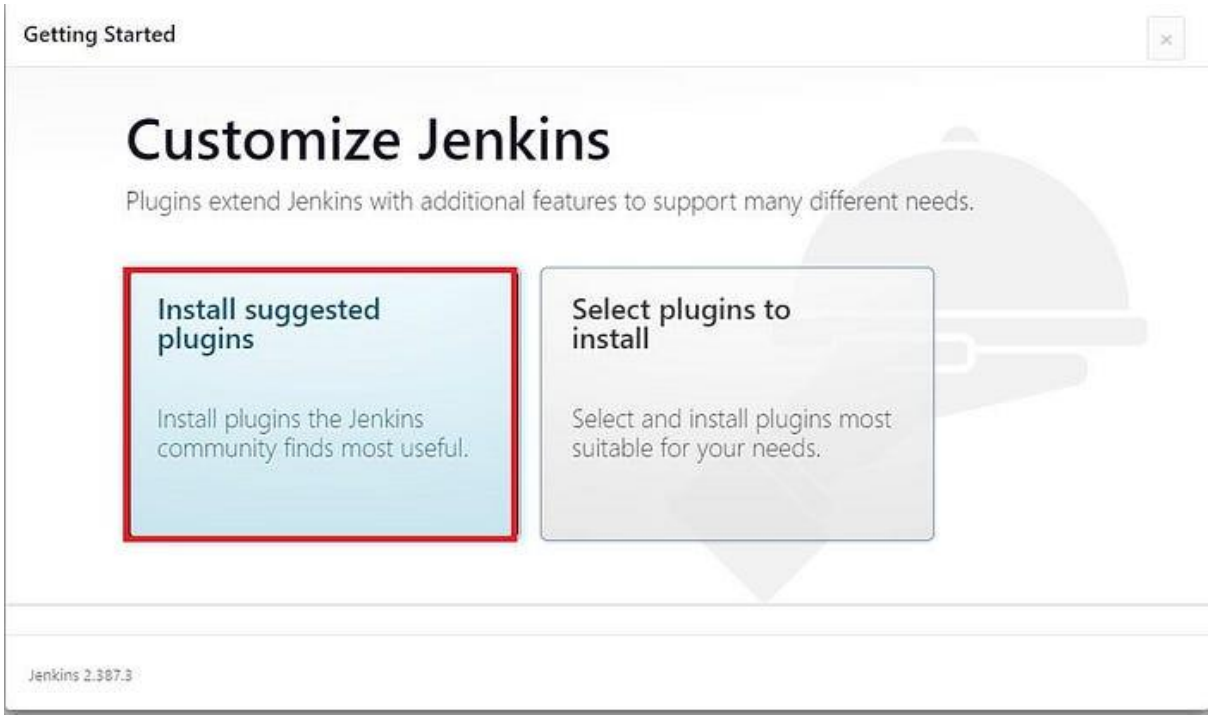
To ensure Jenkins is securely set up by the administrator, a password has been written to the log (**not sure where to find it?**) and this file on the server:

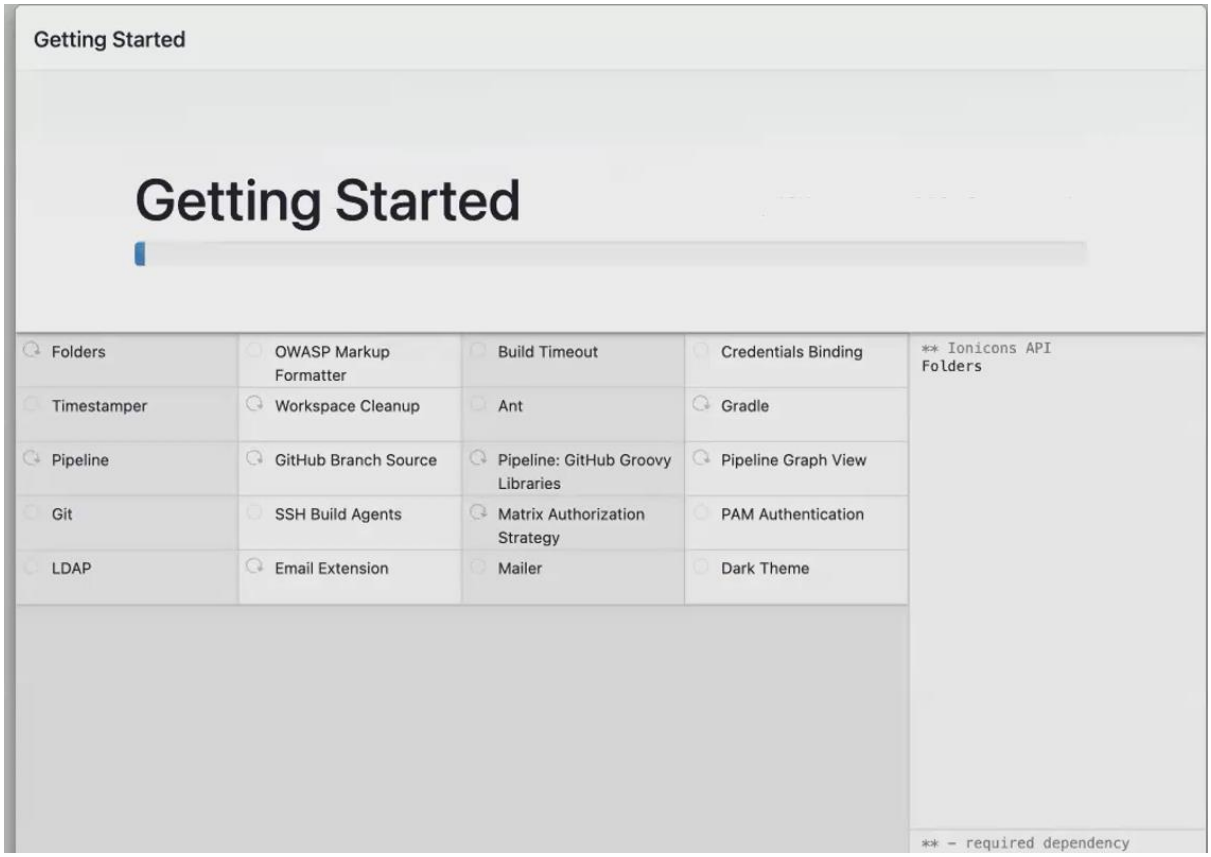
`/var/lib/jenkins/secrets/initialAdminPassword`

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

Administrator password

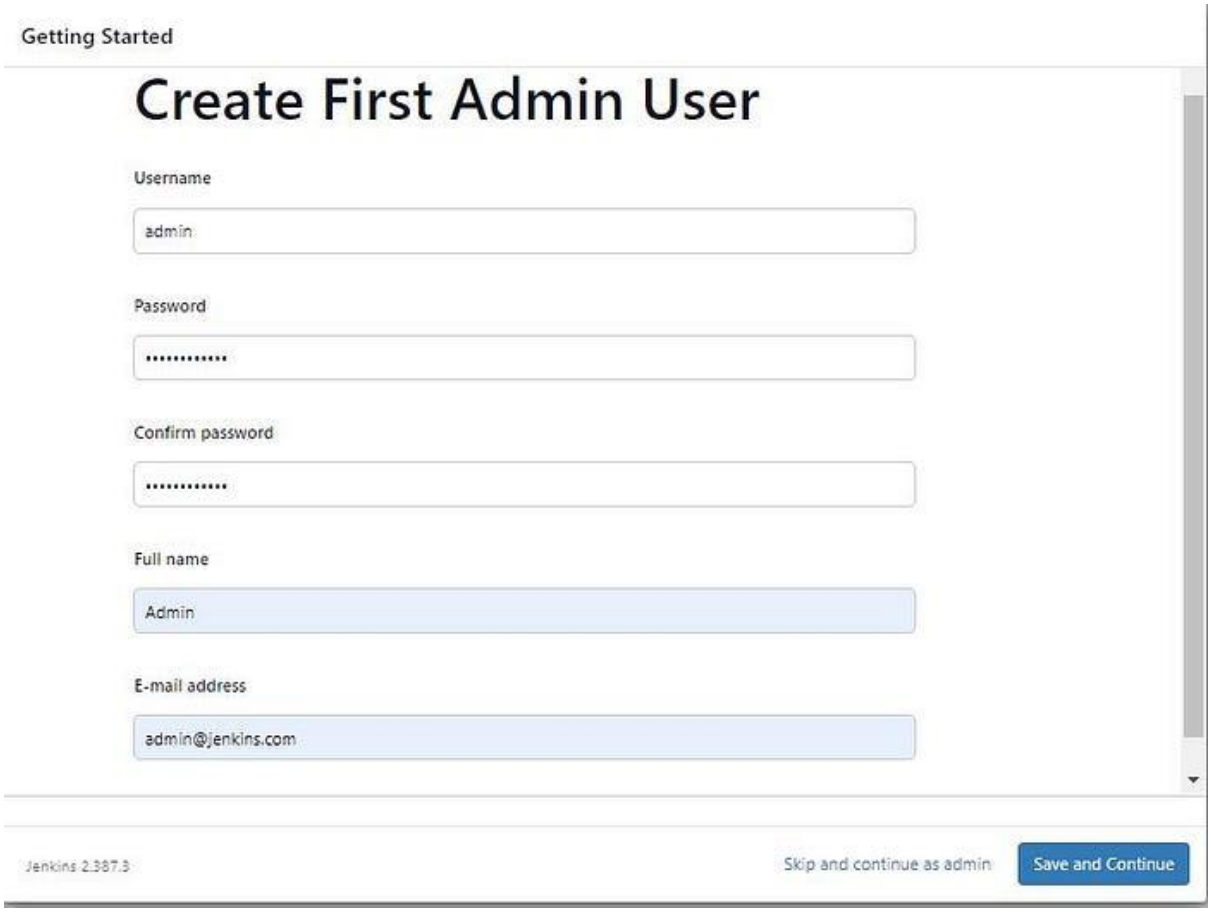
Continue

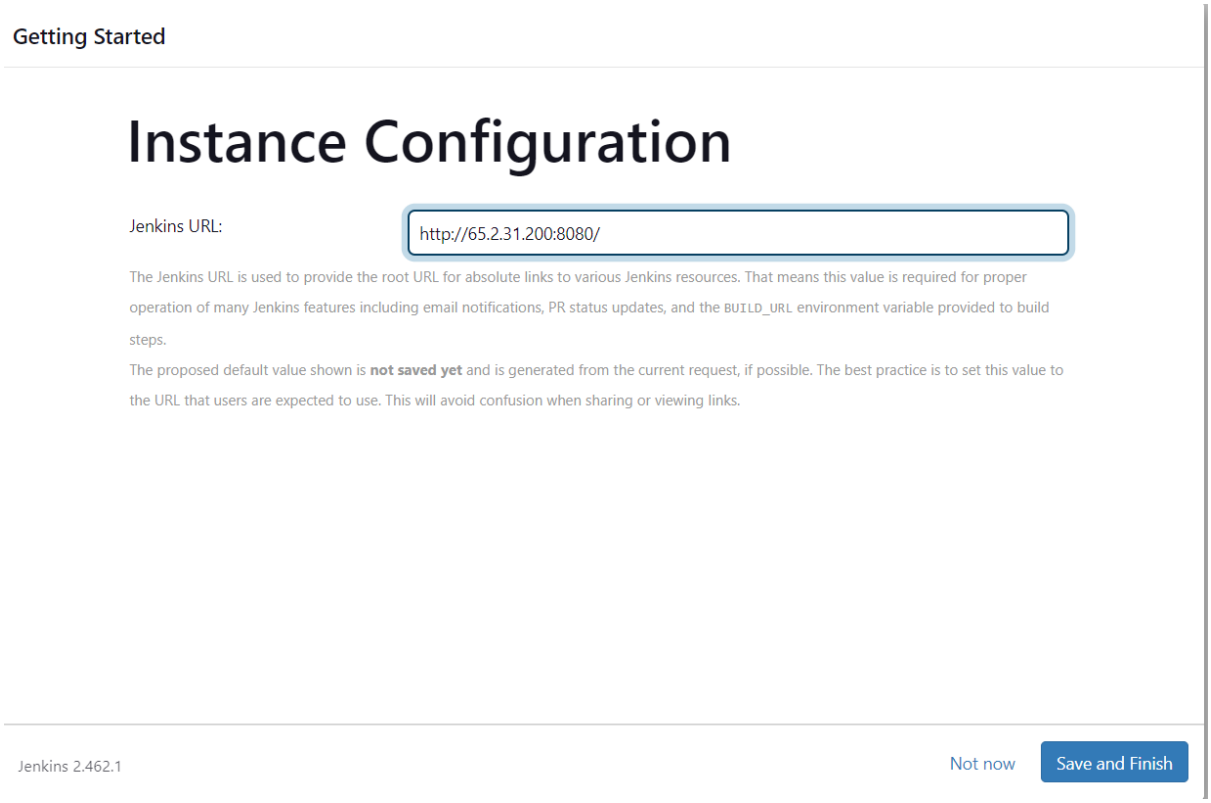


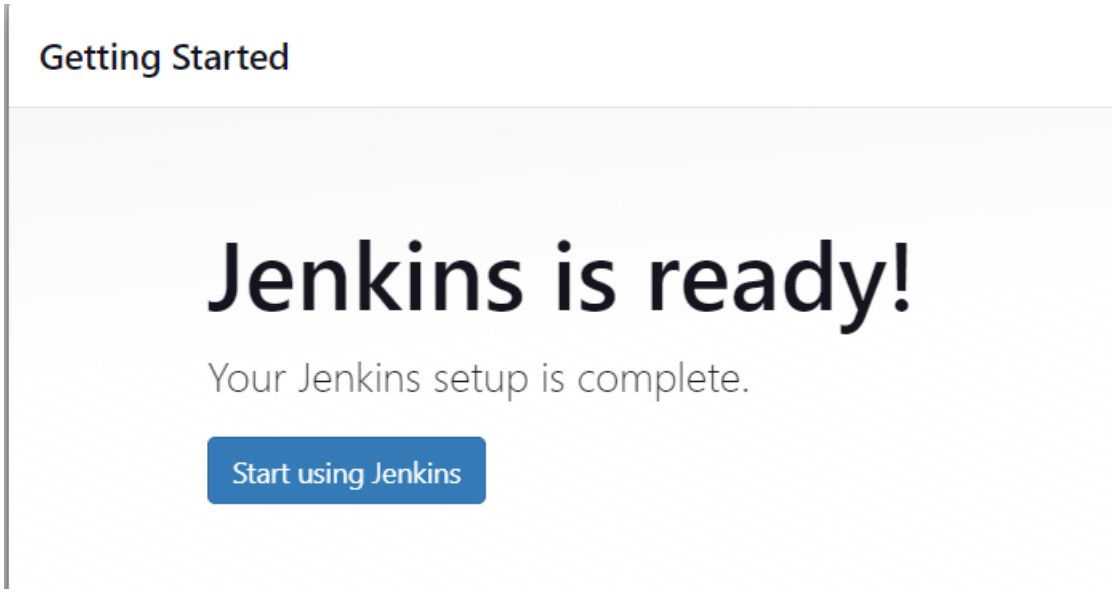


Now let’s create our first Admin user and provide the required info:

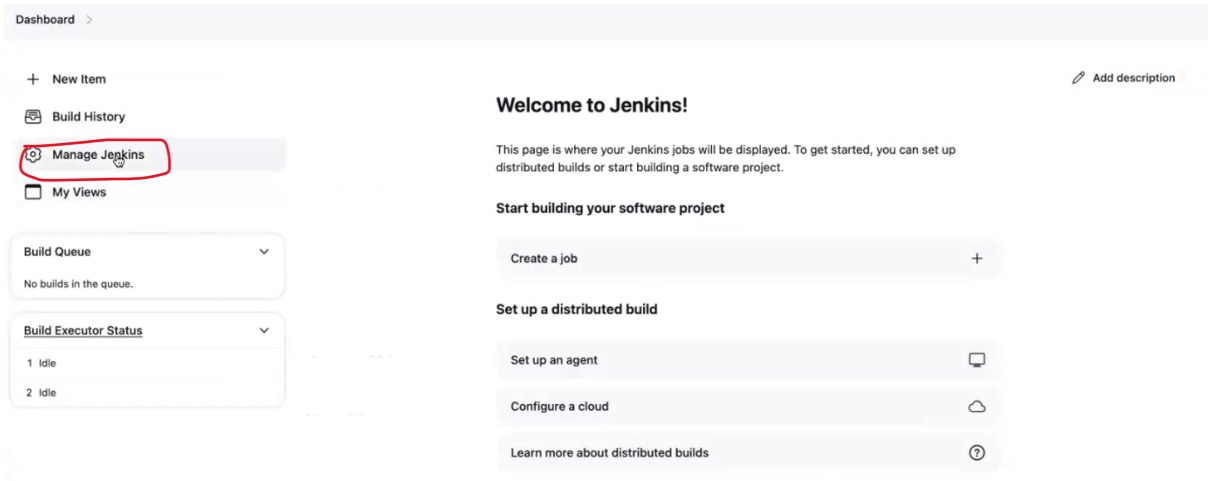
Give username as admin, then only we can access all inside Jenkins. In username don’t give your name. In full name you give your name or any other name can be given.

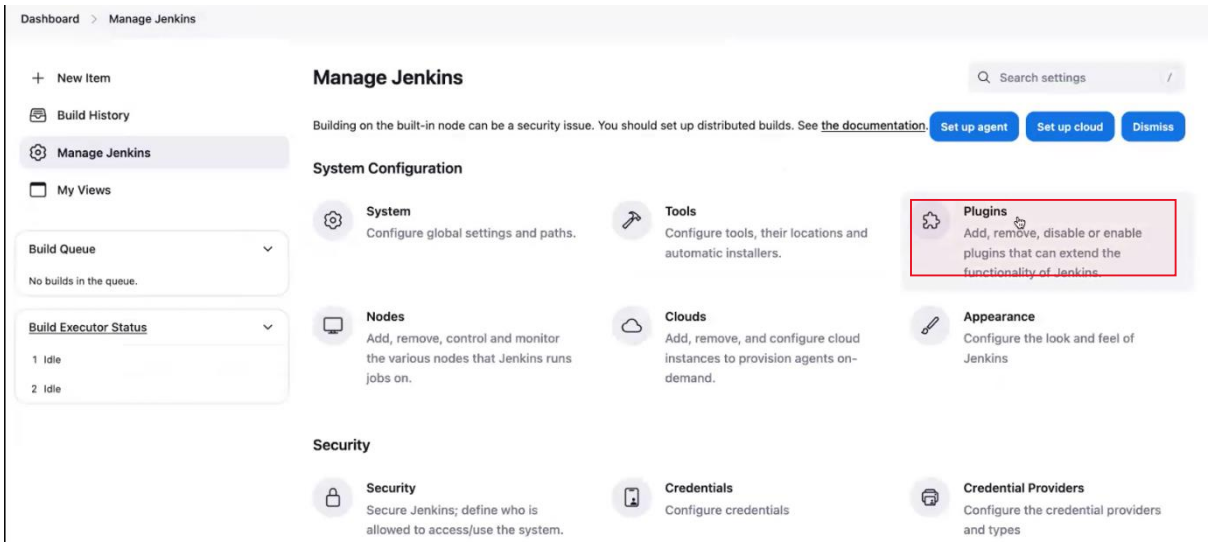

Getting Started
Create First Admin User
Username
admin
Password
Confirm password
Full name
Admin
E-mail address
admin@jenkins.com
Jenkins 2.387.3
Skip and continue as admin
Save and Continue

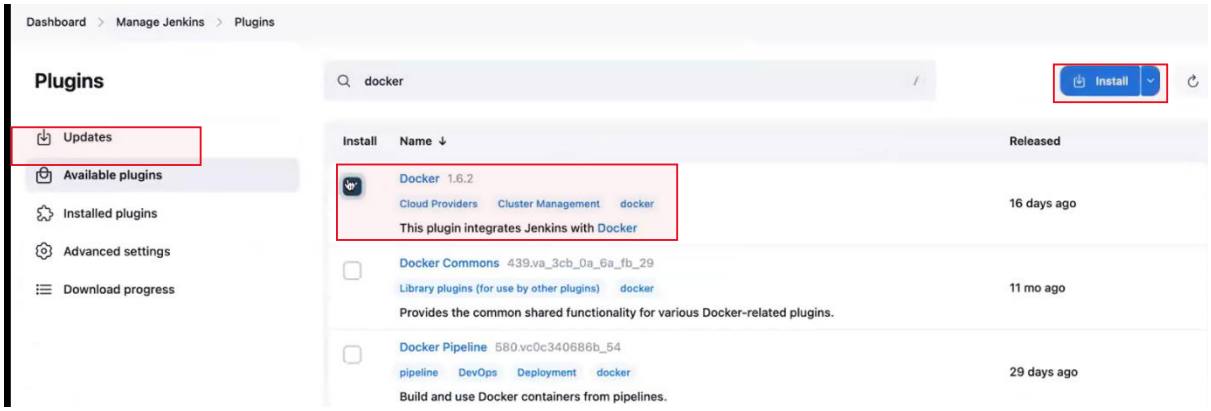

Getting Started
Instance Configuration
Jenkins URL:
http://65.2.31.200:8080/
The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the BUILD_URL environment variable provided to build steps.
The proposed default value shown is not saved yet and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.
Jenkins 2.462.1
Not now
Save and Finish

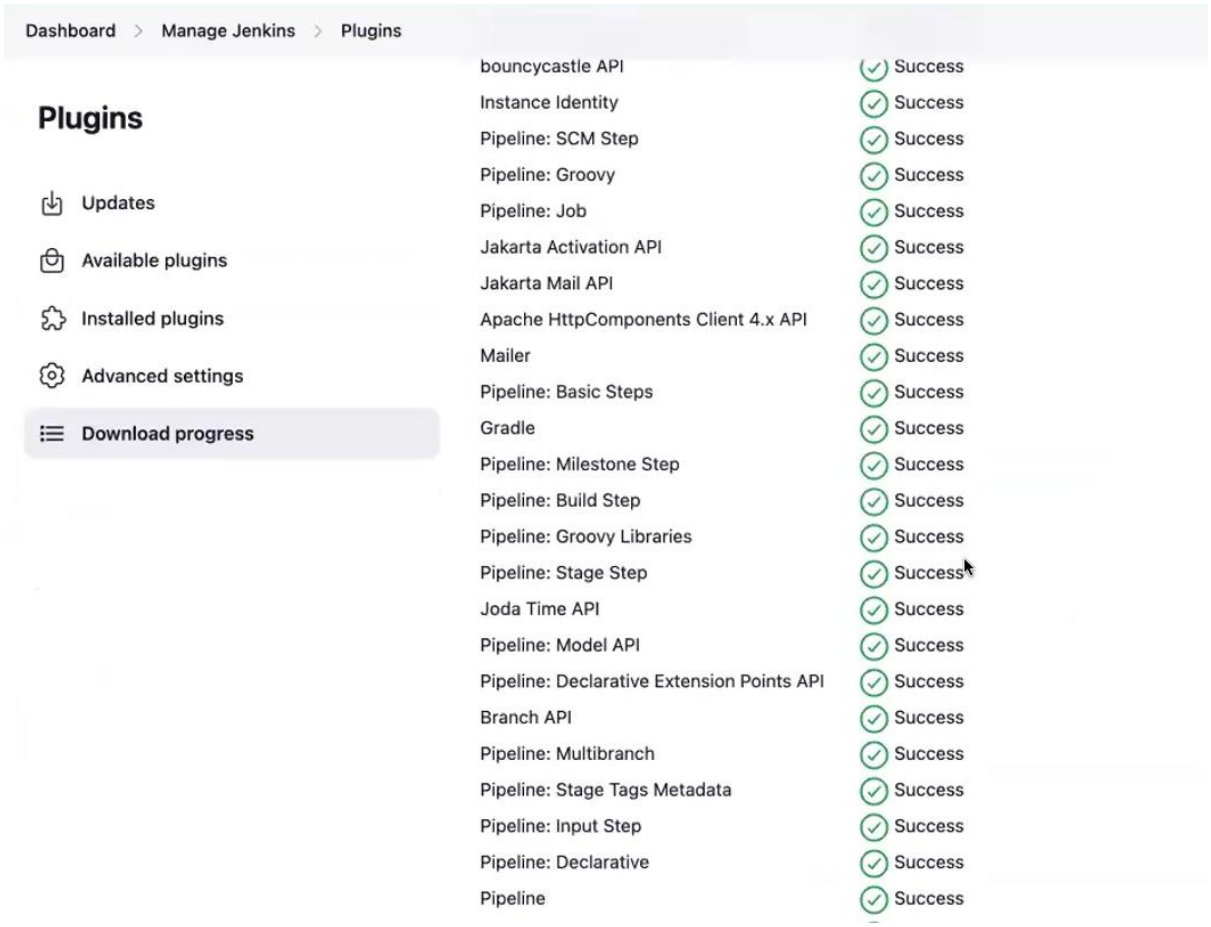


Before creating the job installed docker plugin in Jenkins and install docker in ec2 also.









Suppose docker already in ec2. then check it is installed or not by using command [docker –version]

Give docker permission for Jenkins. then only we have to build the docker image.

```
ubuntu@ip-172-31-46-99:~$ docker --version
Docker version 24.0.7, build 24.0.7-0ubuntu2~22.04.1
ubuntu@ip-172-31-46-99:~$ sudo usermod -aG docker jenkins
ubuntu@ip-172-31-46-99:~$ sudo systemctl restart jenkins
ubuntu@ip-172-31-46-99:~$
```

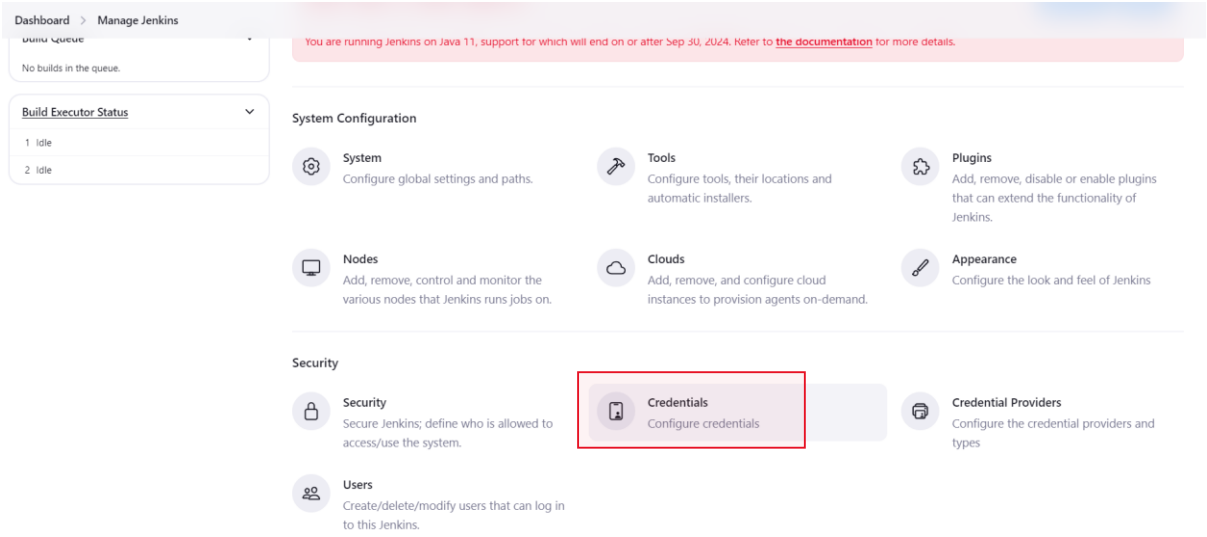
After give restart Jenkins, refresh the Jenkins page. again, ask Jenkins login and password.

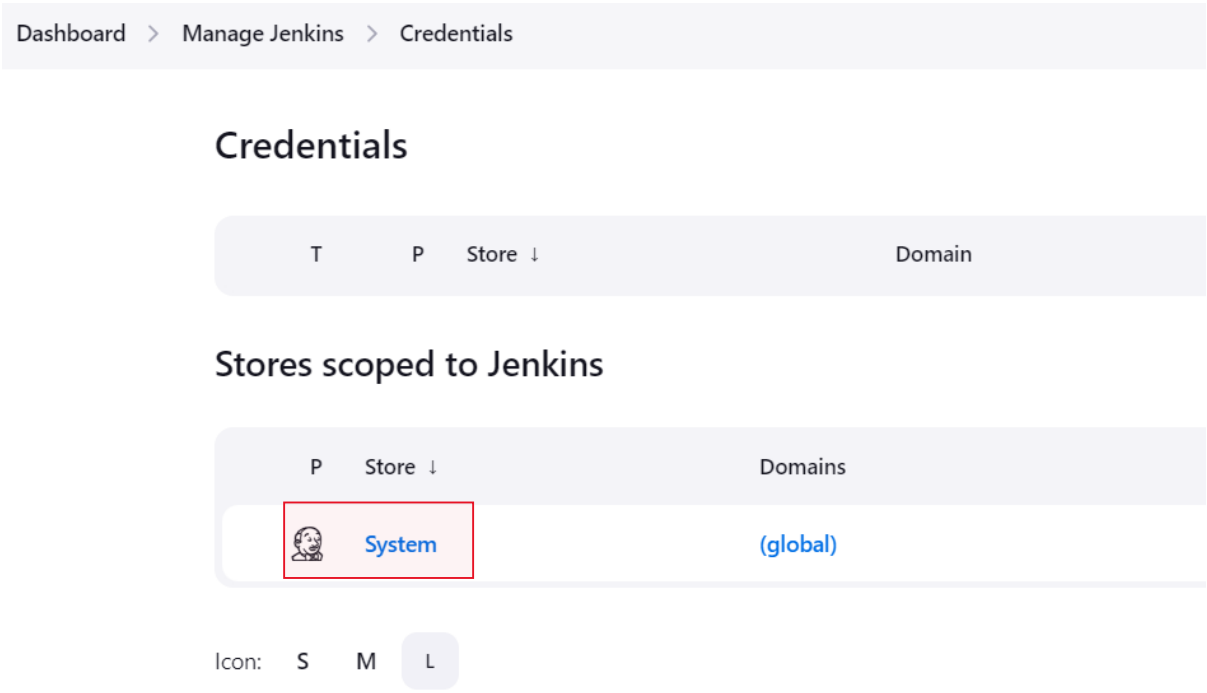
Add credentials:

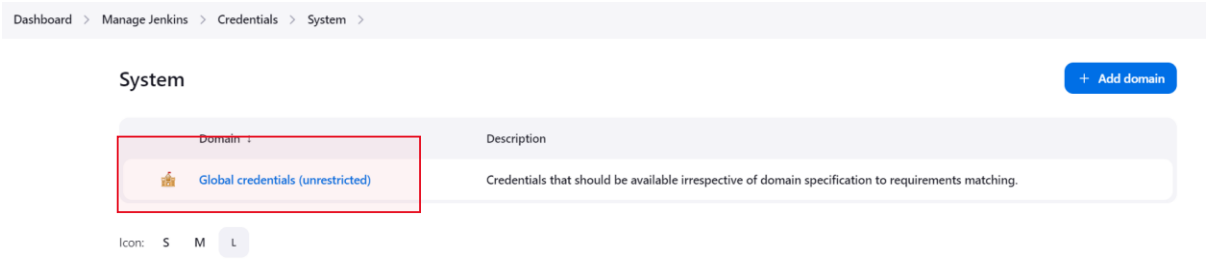
we have to pass the credentials for the Doctor Hub. **to push your image to the docker hub registry. You need to do a docker login. So, for docker login. You have to give the username and password. So, your Jenkins have to do the docker login.**

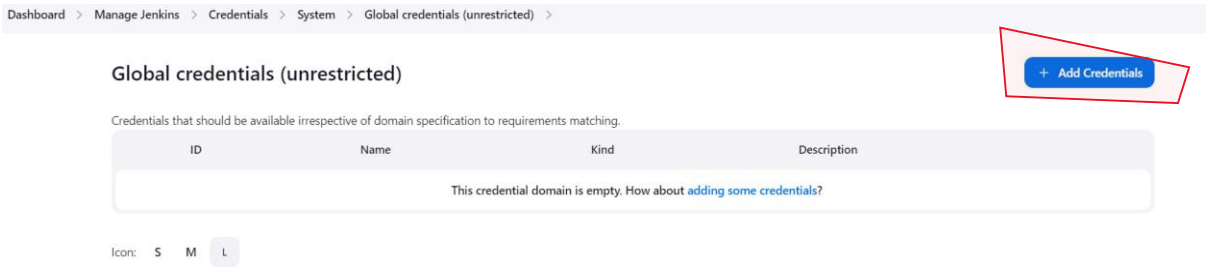
Add credentials

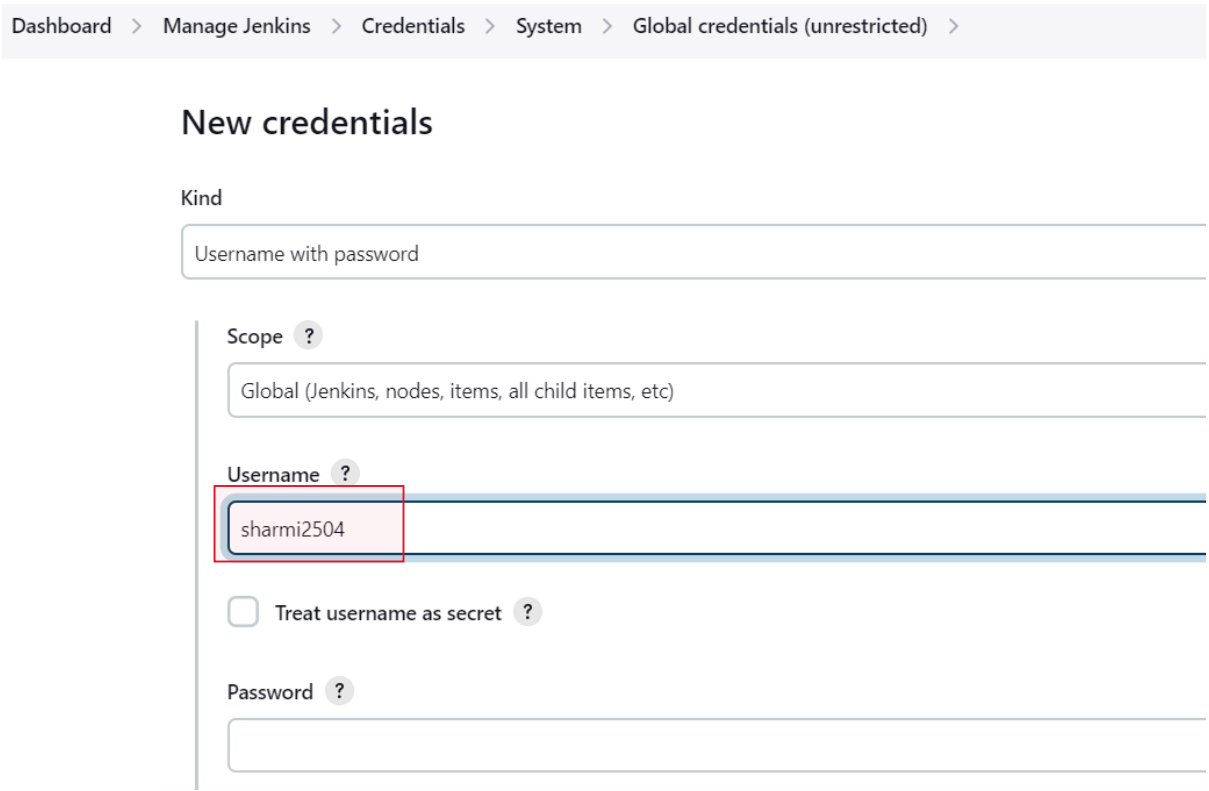
Click dashboard → manage Jenkins→ credentials



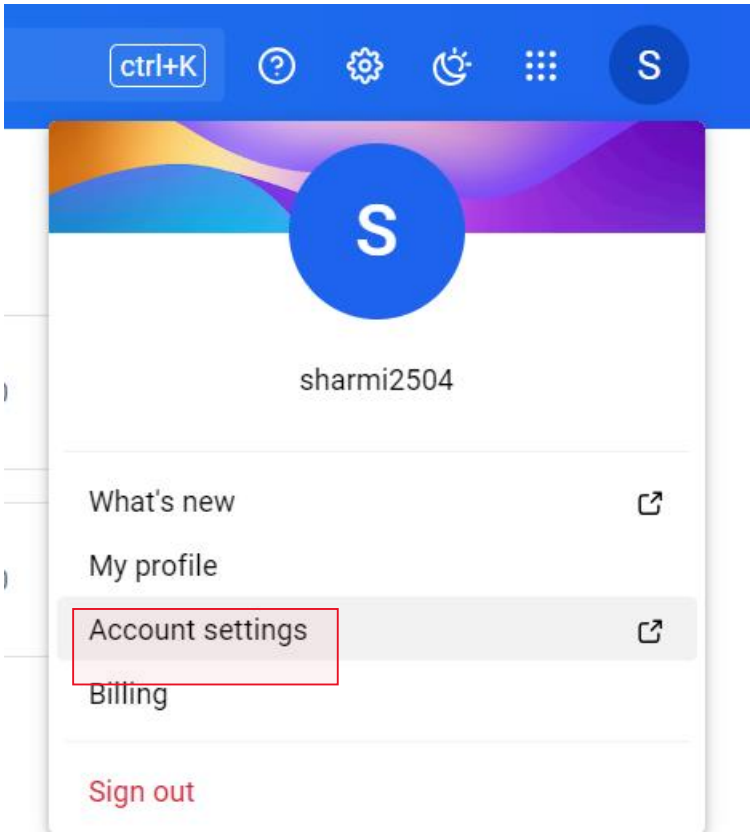








Above username is dockerhub user name.For password we generate tokens like github.



Password

****

You can change your password by initiating a reset via email. Reset password

Security

Two-factor authentication

Two factor authentication is **disabled**.

Personal access tokens

There is 1 personal access token associated with your account.

Settings / Personal access tokens / New access token

Create access token

A personal access token is similar to a password except you can have many tokens and revoke access to each one at any time. Learn more

Access token description

capstone jenkins docker

Access permissions

Public Repo Read-only

Public Repo Read-only tokens allow to view, search, and pull images from any public repositories.

Cancel Generate

Copy access token

Use this token as a password when you sign in from the Docker CLI client. Learn more

Make sure you copy your personal access token now. Your personal access token is only displayed once. It isn't stored and can't be retrieved later.

Access token description
capstone jenkins docker

Access permissions
Public Repo Read-only

To use the access token from your Docker CLI client:

1. Run

```
$ docker login -u sharmi2504
```

Copy

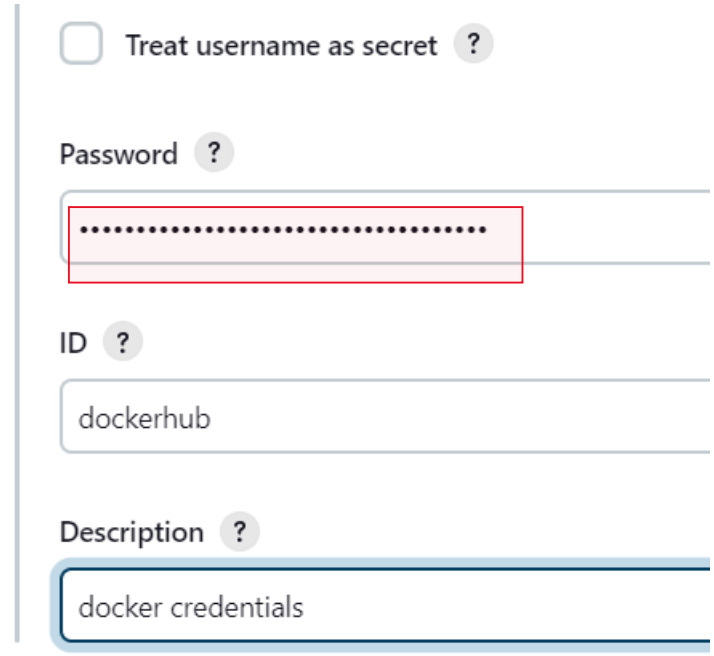
2. At the password prompt, enter the personal access token.

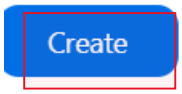
```
dckr_pat_tg-HQApdGliv8vy95CdXhtszCbM
```

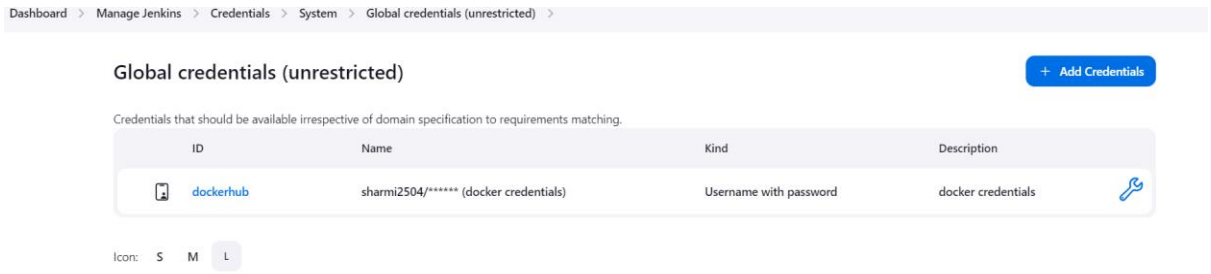
Copy

Back to access tokens

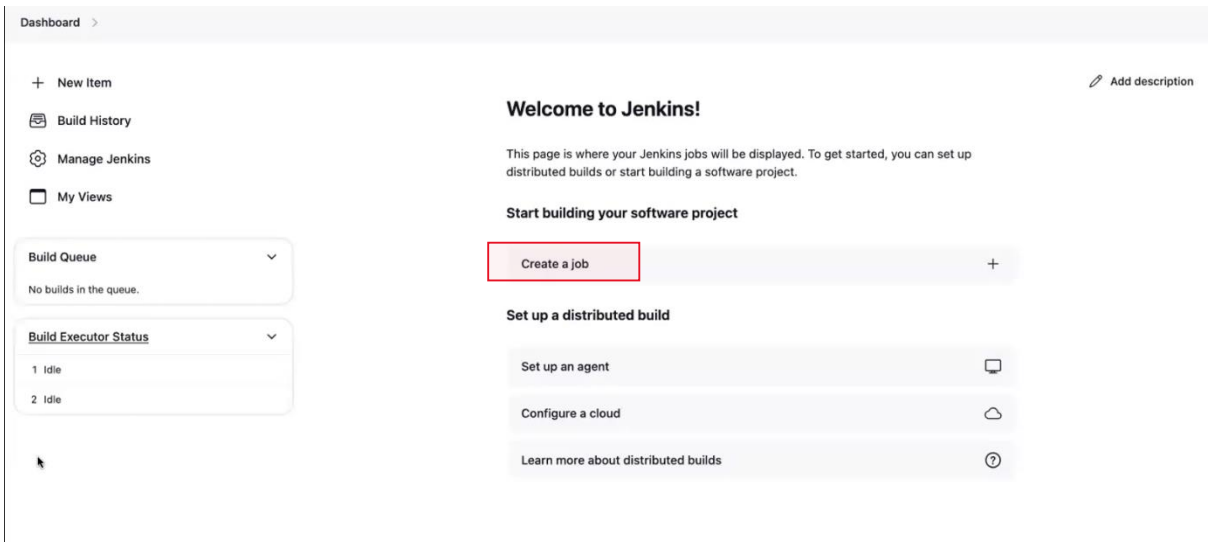
Copy the token, paste it in Jenkins

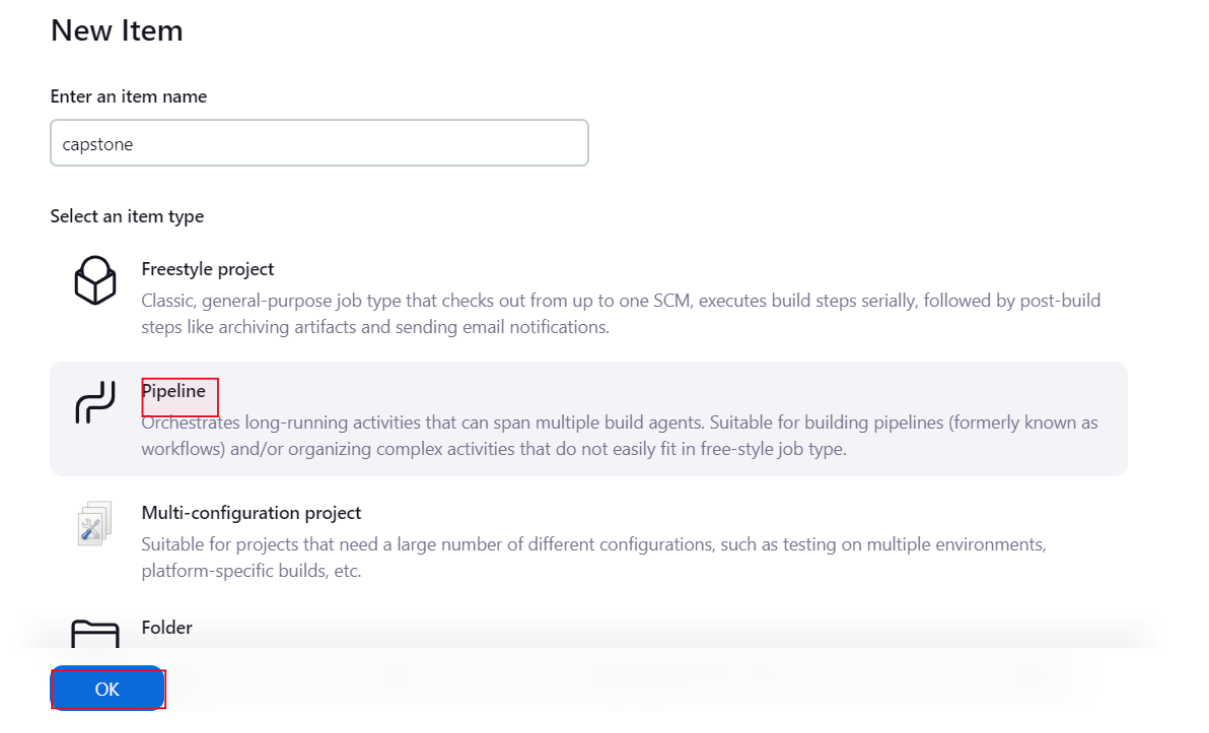


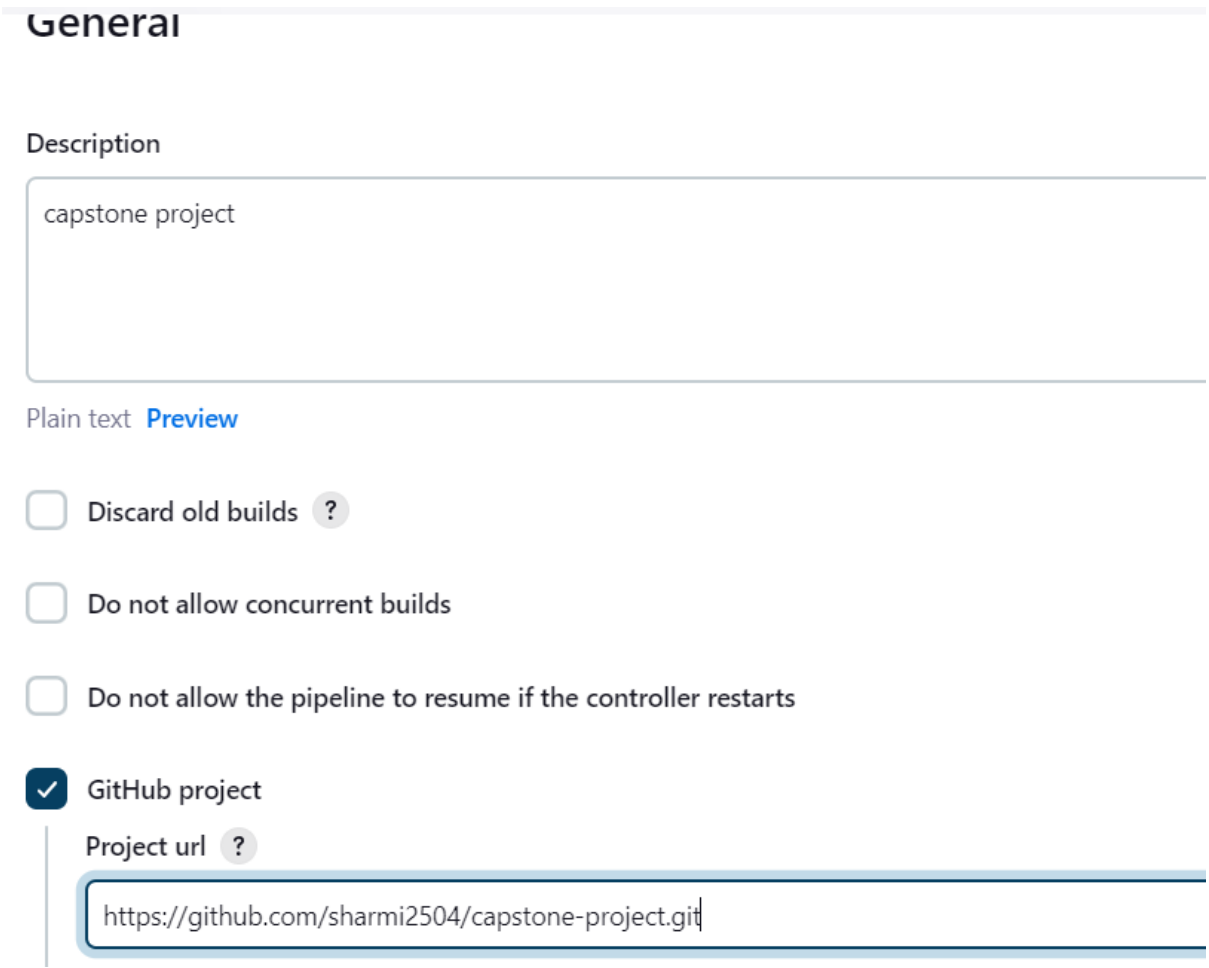




Now create the job in Jenkins:




New Item
Enter an item name
capstone
Select an item type
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.
Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.
Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.
Folder
OK


General
Description
capstone project
Plain text
Preview
Discard old builds
Do not allow concurrent builds
Do not allow the pipeline to resume if the controller restarts
GitHub project
Project url
https://github.com/sharmi2504/capstone-project.git

Pipeline

Definition

Pipeline script from SCM

SCM

Git

None

Git

Repository URL

https://github.com/sharmi2504/capstone-project.git

Credentials

- none -

+ Add

SCM

Git

Repositories

Repository URL

https://github.com/sharmi2504/capstone-project.git

Credentials

- none -

- none -

sharmi2504/****** (docker credentials)

Advanced

SCM

Git

Repositories

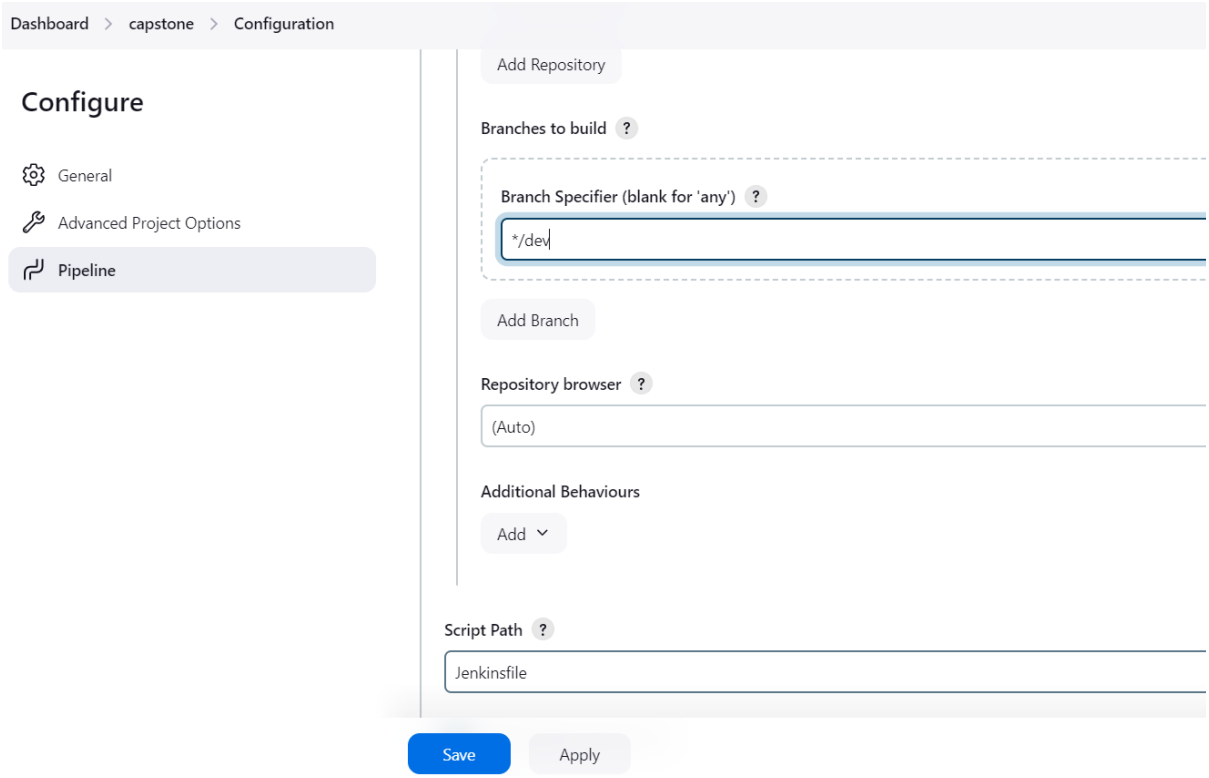
Repository URL

https://github.com/sharmi2504/capstone-project.git

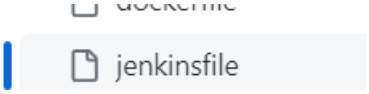
Credentials

sharmi2504/****** (docker credentials)

+ Add

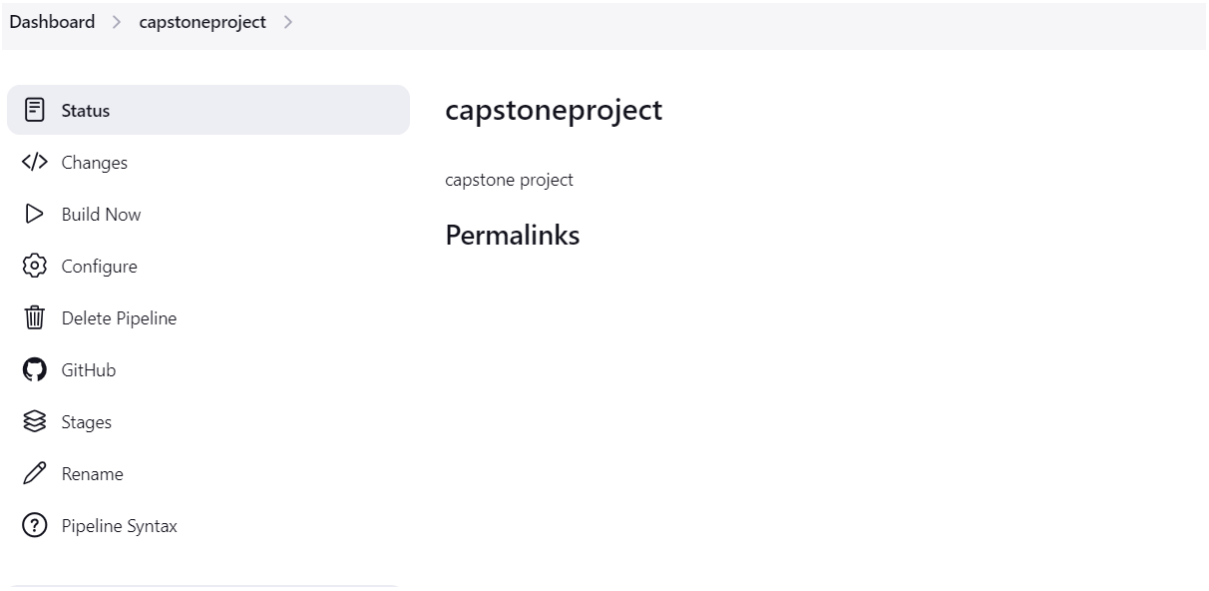


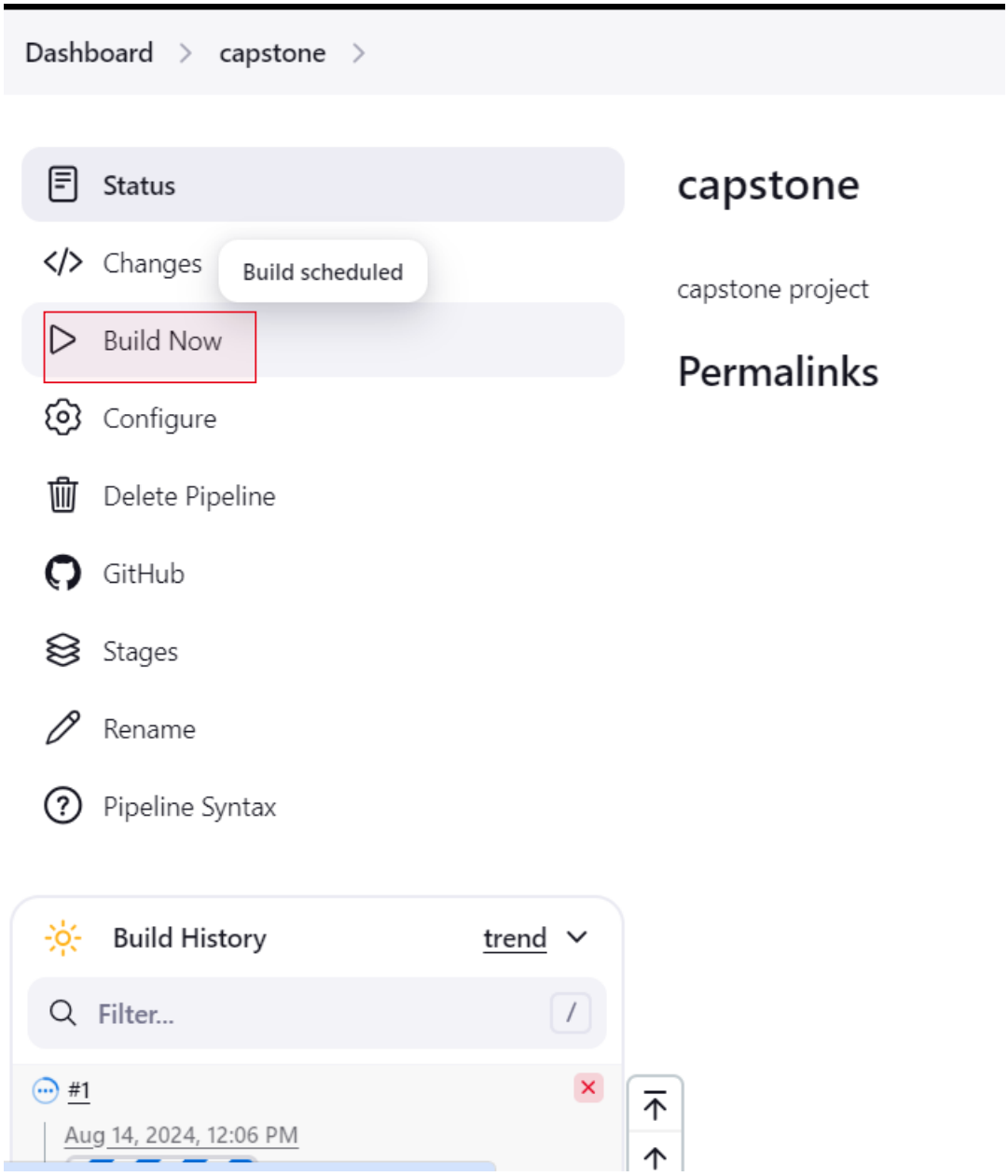
Here I gave j as capital. you give small j . because in git, Jenkins file name as jenkinsfile.



give correct name. then only fetch the code from github.

Click apply and save





After build I have errors. So I stopped.