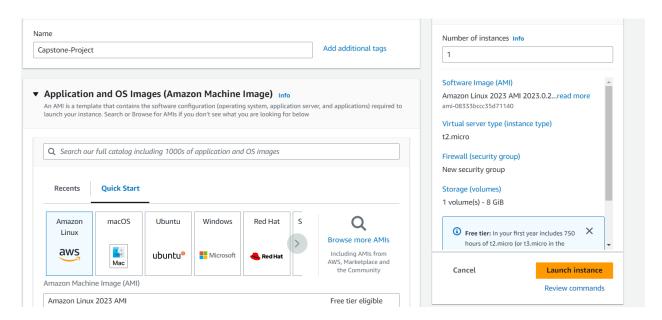
# Creating Ec2 machine from AWS.



#### Connected the EC2 Via SSH

#### Switch the User and installing Git

```
[root@ip-172-31-35-176 ~]# sudo yum install git -y
Last metadata expiration check: 0:03:53 ago on Tue May 9 05:53:00 2023.
Dependencies resolved.
 Package
                                                                  Architecture
Installing:
                                                                  x86_64
 git
Installing dependencies:
 git-core
                                                                  x86_64
 git-core-doc
                                                                  noarch
 perl-Error
perl-File-Find
                                                                  noarch
                                                                  noarch
 perl-Git
                                                                  noarch
 perl-TermReadKey
perl-lib
                                                                  x86_64
                                                                  x86_64
Transaction Summary
Install 8 Packages
```

#### Clone the code from github

```
[root@ip-172-31-35-176 ~]# sudo git clone https://github.com/rvsp/reactjs-demo.git Cloning into 'reactjs-demo'...
remote: Enumerating objects: 40, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 40 (delta 2), reused 2 (delta 2), pack-reused 36
Receiving objects: 100% (40/40), 175.93 KiB | 3.52 MiB/s, done.
Resolving deltas: 100% (11/11), done.
[root@ip-172-31-35-176 ~]# cd reactjs-demo
```

## Dockerfile created in reactjs-demo folder

```
[root@ip-172-31-35-176 reactis-demo]# cat dockerfile
# Use an official Node.js runtime as a parent image
FROM node:14-alpine
# Set the working directory to /app
WORKDIR /app
# Copy package.json and package-lock.json to the working directory
COPY package*.json ./
# Install dependencies
RUN npm install
# Copy the rest of the application files to the working directory
COPY . .
# Build the application
RUN npm run build
# Expose port 80 for the application
EXPOSE 80
# Start the application using the production server CMD [ "npm", "start" ]
```

#### Docker-compose file created as below

```
[root@ip-172-31-35-176 reactjs-demo]# cat docker-compose.ym]
version: '3'
services:
   app:
   image: my-react-app
   ports:
        - "80:80"
```

#### Image created and tagged

```
[Docker] $ /bin/sh -xe /tmp/jenkins7371394953029846646.sh
+ docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
newapp V1.2
                   3352450df187 7 hours ago 143MB
         V1.1
                   e0a1e8a351f2 7 hours ago 143MB
newapp
[Docker] $ /bin/sh -xe /tmp/jenkins3297053389929006449.sh
+ docker ps -a
CONTAINER ID IMAGE
                   COMMAND CREATED STATUS PORTS NAMES
+ docker tag newapp:V1.2 raghavanp08/dev:latest
+ docker images
REPOSITORY
              TAG
                       IMAGE ID
                                    CREATED
                                                SIZE
              V1.2
newapp
                      3352450df187 7 hours ago
                                                143MB
raghavanp08/dev latest
                       3352450df187 7 hours ago
                                                143MB
          V1.1
                       e0a1e8a351f2 7 hours ago
newapp
                                                143MB
Finished: SUCCESS
```

#### Navigate to Cd Reactjs-demo and installing the dependencies

```
[root@ip-172-31-35-176 ~]# ls
reactjs-demo
[root@ip-172-31-35-176 ~]# cd reactjs-demo/
[root@ip-172-31-35-176 reactjs-demo]# npm install
npm WARN deprecated chokidar@2.1.8: Chokidar 2 does not receive sonpm WARN deprecated popper.js@1.16.1: You can find the new Popper
added 6 packages, and audited 1652 packages in 13s
65 packages are looking for funding
    run 'npm fund' for details
76 vulnerabilities (1 low, 26 moderate, 36 high, 13 critical)
To address issues that do not require attention, run:
    npm audit fix
To address all issues (including breaking changes), run:
    npm audit fix --force
Run 'npm audit' for details.
```

#### And NPM Build

```
[root@ip-172-31-35-176 reactjs-demo]# npm run build

> react-front-end@0.1.0 build

> react-scripts build

Creating an optimized production build...

Error: error:30308010C:digital envelope routines::unsupported

at new Hash (node:internal/crypto/hash:71:19)

at module.exports (/root/reactjs-demo/node_modules/webpack/lib/butil/createHash.js:135:53)

at NormalModule._initBuildHash (/root/reactjs-demo/node_modules/webpack/lib/NormalModule.js:471:10)

at root/reactjs-demo/node_modules/webpack/lib/NormalModule.js:471:10)

at /root/reactjs-demo/node_modules/webpack/lib/NormalModule.js:471:10)

at /root/reactjs-demo/node_modules/webpack/lib/NormalModule.js:373:3

at iterateNormalLoaders (/root/reactjs-demo/node_modules/loader-runner/lib/LoaderRunner.js:214:10)

at iterateNormalLoaders (/root/reactjs-demo/node_modules/loader-runner/lib/LoaderRunner.js:221:10)

//root/reactjs-demo/node_modules/react-scripts/scripts/build.js:19

throw err;

//

Error: error:0308010c:digital envelope routines::unsupported

at new Hash (node:internal/crypto/hash:71:19)

at object.createHash (node:crypto:l33:10)

at nodule.exports (/root/reactjs-demo/node_modules/webpack/lib/NormalModule.js:47:16)

at /root/reactjs-demo/node_modules/webpack/lib/NormalModule.js:323:13

at /root/reactjs-demo/node_modules/webpack/lib/NormalModule.js:323:13

at /root/reactjs-demo/node_modules/webpack/lib/NormalModule.js:323:13

at /root/reactjs-demo/node_modules/webpack/lib/NormalModule.js:323:13

at /root/reactjs-demo/node_modules/mode_modules/webpack/lib/NormalModule.js:323:13

at /root/reactjs-demo/node_modules/bader-runner/lib/LoaderRunner.js:336:11

at /root/reactjs-demo/node_modules/bader-runner/lib/LoaderRunner.js:333:18

at context.callback (/root/reactjs-demo/node_modules/webpack/lib/NormalModule.js:333:18

at /root/reactjs-demo/node_modules/bader-runner/lib/LoaderRunner.js:333:18

at /root/reactjs-demo/node_modules/bader-runner/lib/LoaderRunner.js:333:18

at /root/reactjs-demo/node_modules/bader-runner/lib/LoaderRunner.js:111:13)

at /root/reactjs-de
```

#### Installed Apache to host this in web

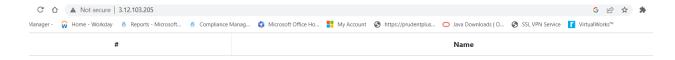
```
root@ip-172-31-20-197:~/reactjs-demo# sudo apt-get install apache2 -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
    apache2-bin apache2-data apache2-utils libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.3-0 mailcap mime-support ssl-cert
Suggested packages:
    apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser
The following NEW packages will be installed:
    apache2 apache2-bin apache2-data apache2-utils libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.3-0 mailcap mime-support ssl-cert
0 upgraded, 12 newly installed, 0 to remove and 29 not upgraded.
Need to get 2102 kB of archives.
After this operation, 8389 kB of additional disk space will be used.
Get:1 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libapr1 amd64 1.7.0-8ubuntu0.22.04.1 [108 kB]
Get:2 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1 amd64 1.6.1-5ubuntu4.22.04.1 [92.6 kB]
Get:3 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-ddap amd64 1.6.1-5ubuntu4.22.04.1 [91.8 kB]
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-ddap amd64 1.6.1-5ubuntu4.22.04.1 [91.8 kB]
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-ddap amd64 1.6.1-5ubuntu4.22.04.1 [91.8 kB]
Get:4 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-ddap amd64 1.6.1-5ubuntu4.22.04.1 [91.8 kB]
Get:5 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-ddap amd64 1.6.1-5ubuntu4.22.04.1 [91.8 kB]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-ddap amd64 1.6.1-5ubuntu4.22.04.1 [91.8 kB]
Get:6 http://us-east-2.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-ddap amd64 1.6.1-5ubuntu4.22.04.1 [91.8 kB]
```

#### File copied to /var/www/html

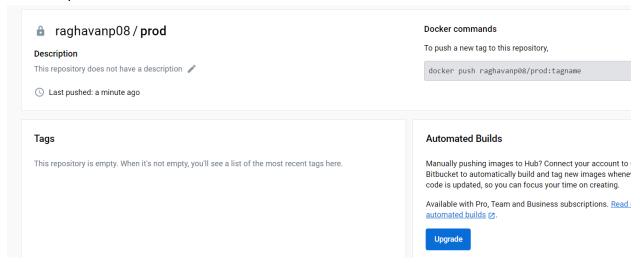


# Final Output from Web,

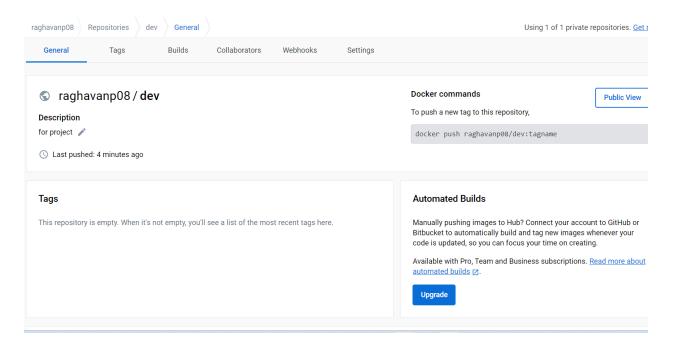
Deployment url: http://3.12.103.205



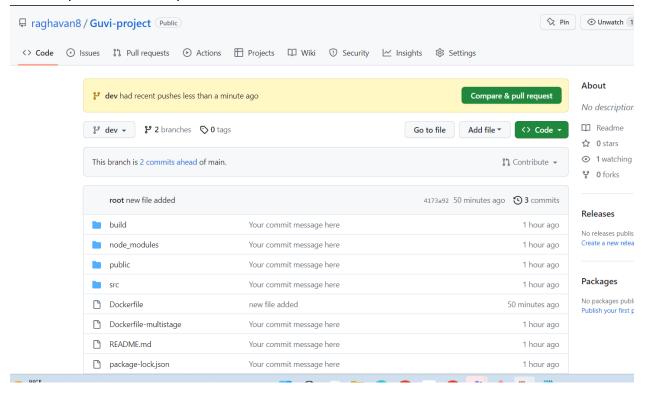
## 1. Private repo



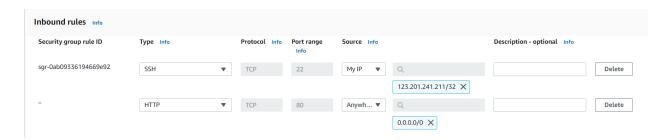
## 2. Public repo



#### Github repo after the files pushed from Ec2 to Github



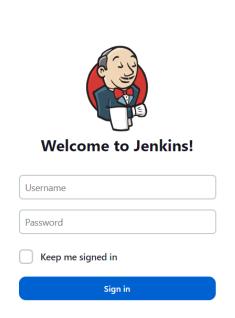
After create the AWS EC2 Machine we have open inbound rules to ssh login from only MY Ip And Application access from any IP



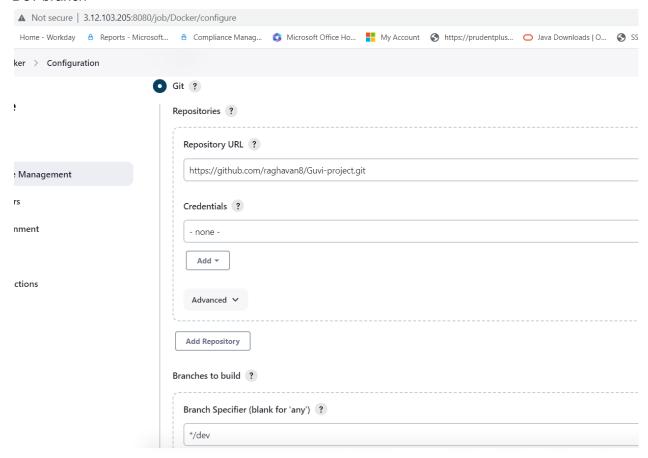
GIT Repo = raghavan8/Guvi-project Docker Public = raghavanp08/dev Docker Private = raghavanp08/prod

# Jenkins Server login.





# Jenkins server pipeline project by clone git code Dev branch



## Build the docker image via pipeline

#### Command

See the list of available environment variables

```
docker build -t newapp:V1.2 .
sudo docker images
```

# Docker images

```
root@ip-172-31-20-197:~# cd Guvi-project/
root@ip-172-31-20-197:~/Guvi-project# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
newapp V1.2 3352450df187 6 minutes ago 143MB
newapp V1.1 e0a1e8a351f2 9 minutes ago 143MB
root@ip-172-31-20-197:~/Guvi-project#
```

## Jenkins output

```
+ 1s
Dockerfile
Dockerfile-multistage
README.md
build
node_modules
package-lock.json
package.json
public
src
[Docker] $ /bin/sh -xe /tmp/jenkins13458413573952636340.sh
+ docker images
REPOSITORY TAG
                    IMAGE ID CREATED
                                                  SIZE
                   3352450df187 37 minutes ago 143MB
          V1.2
newapp
                   e0a1e8a351f2 40 minutes ago 143MB
newapp
           V1.1
[Docker] $ /bin/sh -xe /tmp/jenkins14646867547929613126.sh
+ docker ps -a
CONTAINER ID IMAGE
                      COMMAND CREATED
                                         STATUS
                                                  PORTS
                                                           NAMES
Finished: SUCCESS
```

#### Docker images build and push to dev repo via jenkins

```
raghavanp08/dev latest 3352450df187 8 hours ago 143MB
                 V1.1
                          e0a1e8a351f2 8 hours ago 143MB
newapp
+ pwd
/var/lib/jenkins/workspace/Docker
+ cat Dockerfile
FROM nginx:1.20
COPY build /usr/share/nginx/html
[Docker] $ /bin/sh -xe /tmp/jenkins14887826102851197023.sh
+ docker push raghavanp08/dev:latest
The push refers to repository [docker.io/raghavanp08/dev]
ae81ae2800da: Preparing
07ef16952879: Preparing
881700cb7ab2: Preparing
4f49c6d6dd07: Preparing
a64d597d6b14: Preparing
c2a3d4a53f9a: Preparing
fd95118eade9: Preparing
c2a3d4a53f9a: Waiting
fd95118eade9: Waiting
881700cb7ab2: Layer already exists
07ef16952879: Layer already exists
a64d597d6b14: Layer already exists
4f49c6d6dd07: Layer already exists
fd95118eade9: Layer already exists
c2a3d4a53f9a: Layer already exists
ae81ae2800da: Layer already exists
latest: digest: sha256:628356e30ec42c5dc059703bece1a6f3414c12c4bfc7a7b91abd7674db3f38aa size: 1780
Finished: SUCCESS
```

```
+ pwd
/var/lib/jenkins/workspace/Docker
[Docker] $ /bin/sh -xe /tmp/jenkins2594257349517654230.sh
+ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS
8183e5573711 raghavanp08/dev "/docker-entrypoint..." 3 minutes ago Up 3 minutes
```

# Docker public repo



# Description

for project 🧪

( Last pushed: 2 minutes ago

