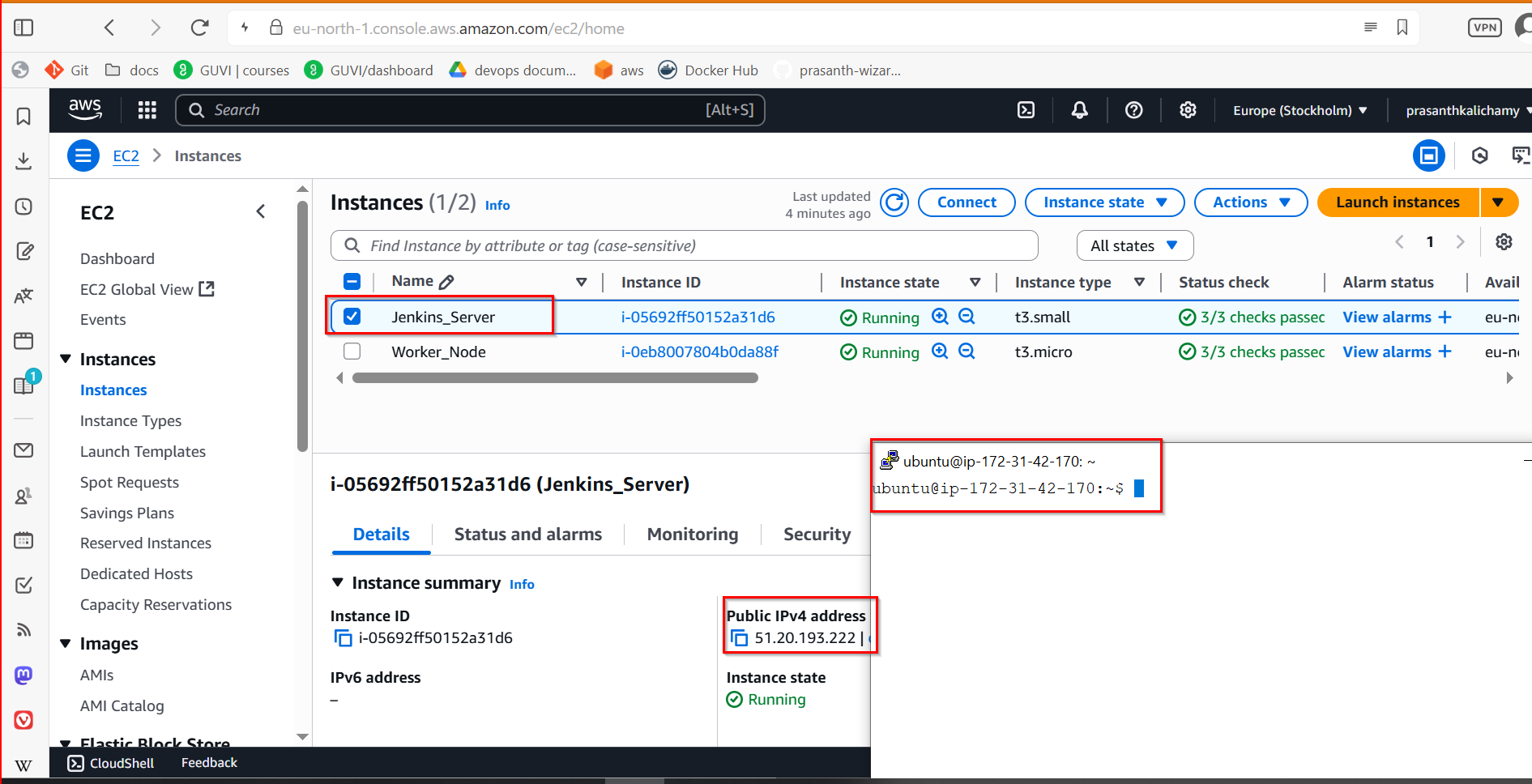
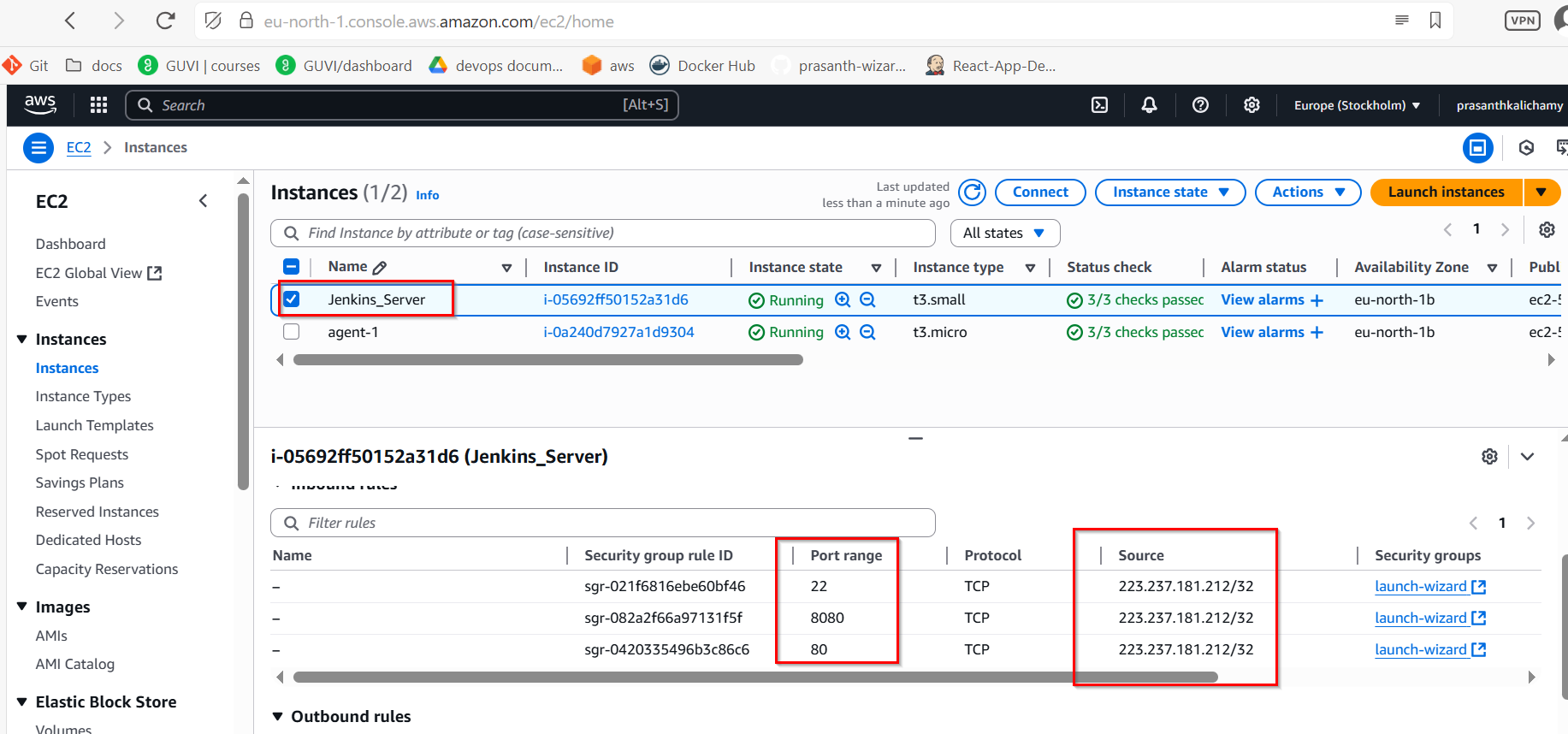
**REACTJS E-COMMERCE APPLICATION**

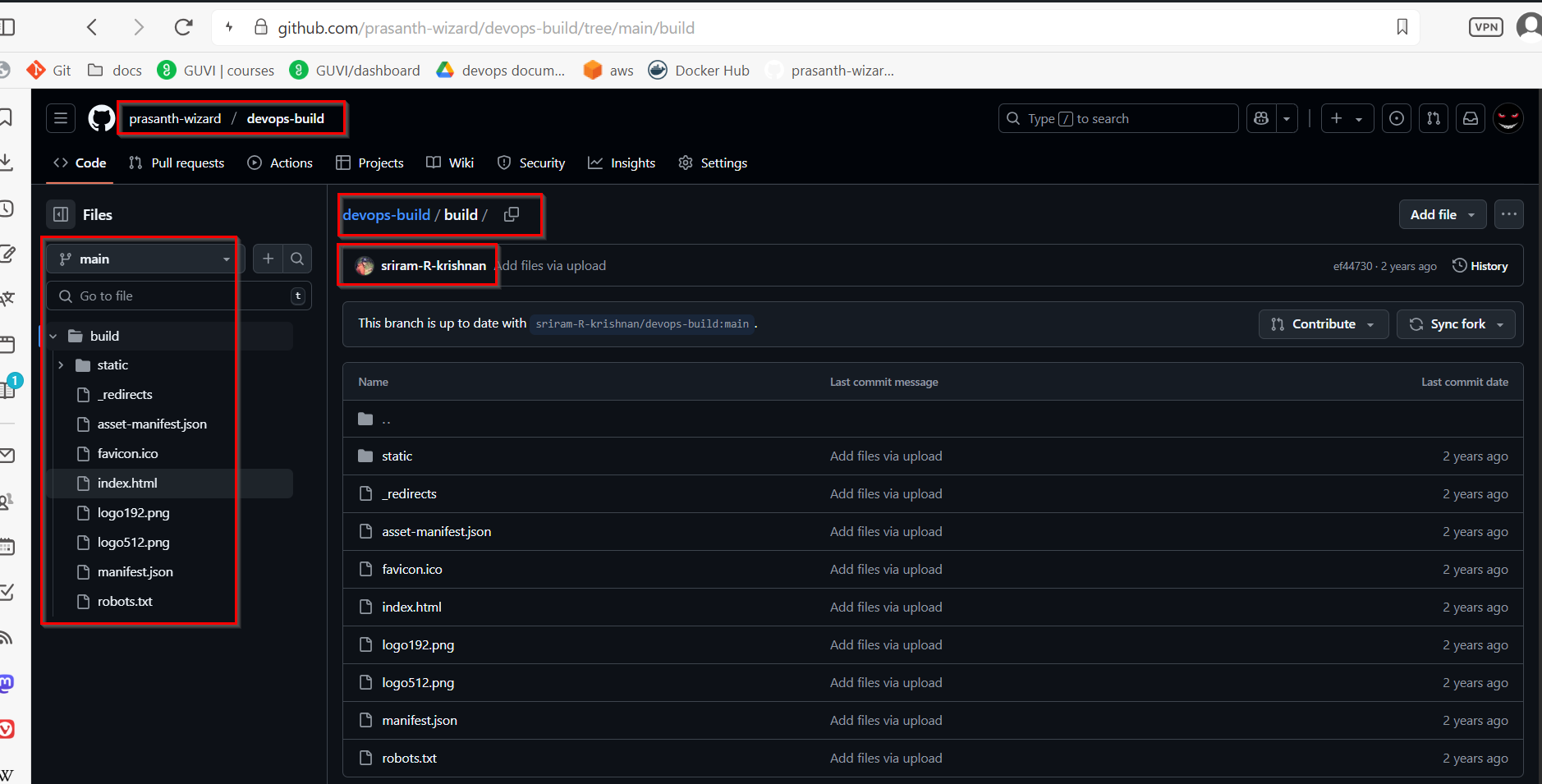
**Jenkins\_Server:**

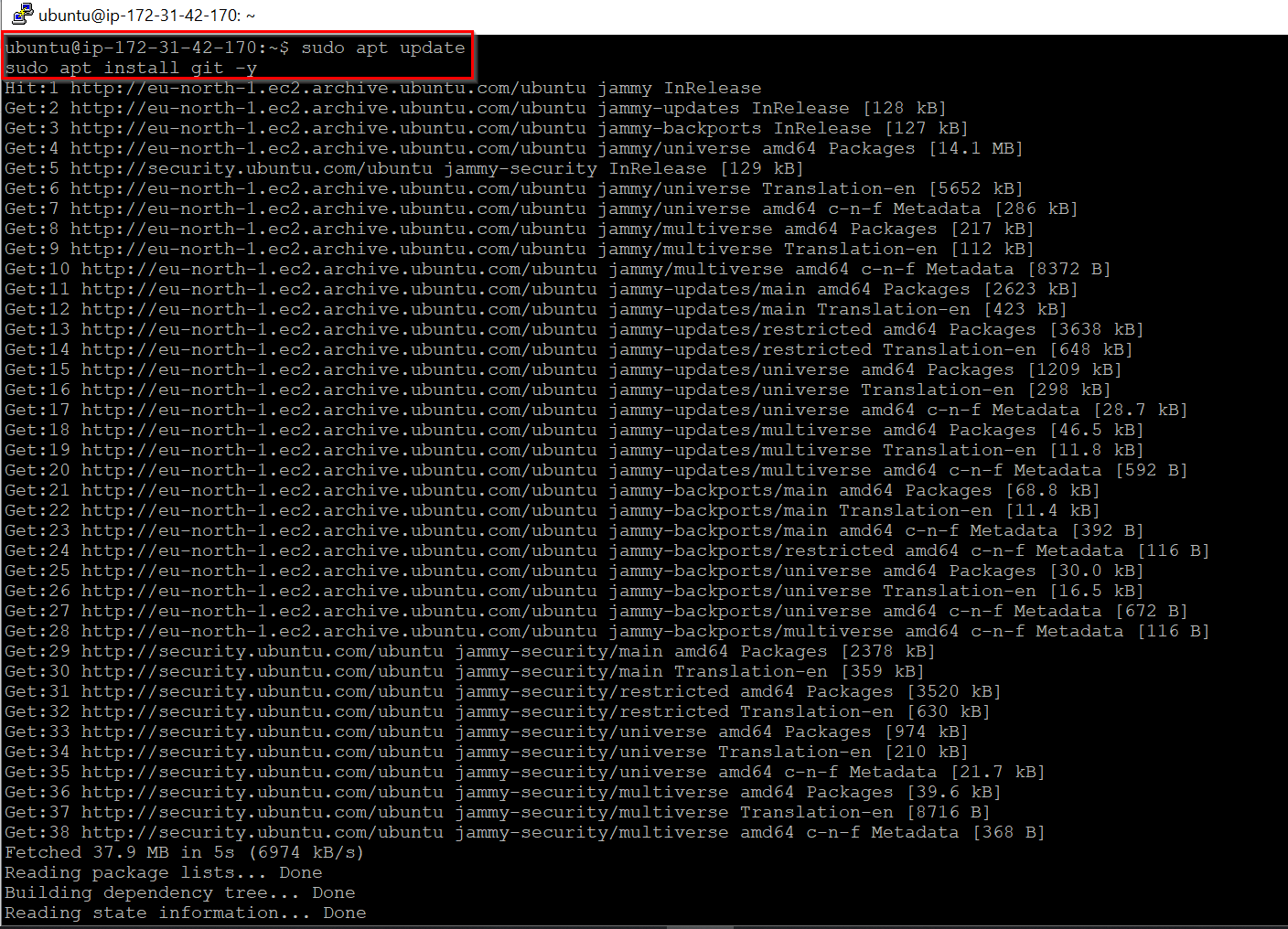


**Jenkins\_server Security\_Group Inbound rules:**

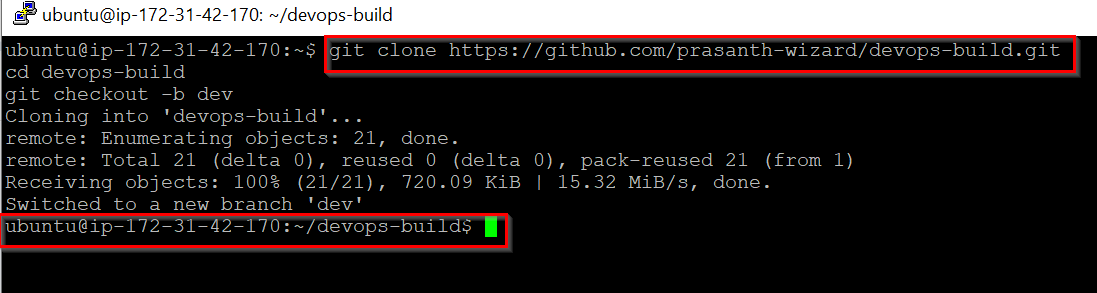
****

**Forking the Repository and Git Setup**



**Install Git:**

**Clone my Forked Repository (on Jenkins Server):**



**Docker Setup and Application Dockerization**

sudo apt update

sudo apt install apt-transport-https ca-certificates curl gnupg-agent software-properties-common -y

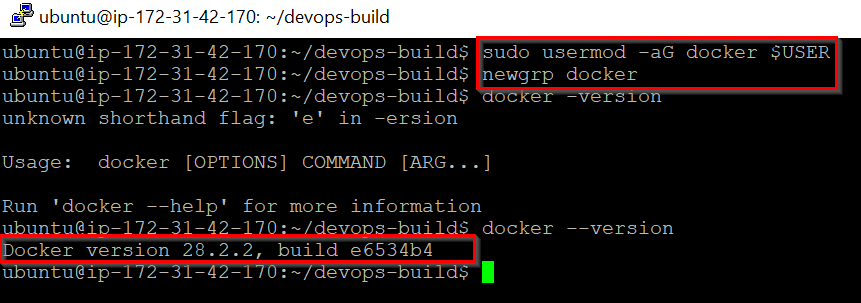
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add –

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable"

sudo apt update

sudo apt install docker-ce docker-ce-cli containerd.io -y

sudo usermod -aG docker $USER

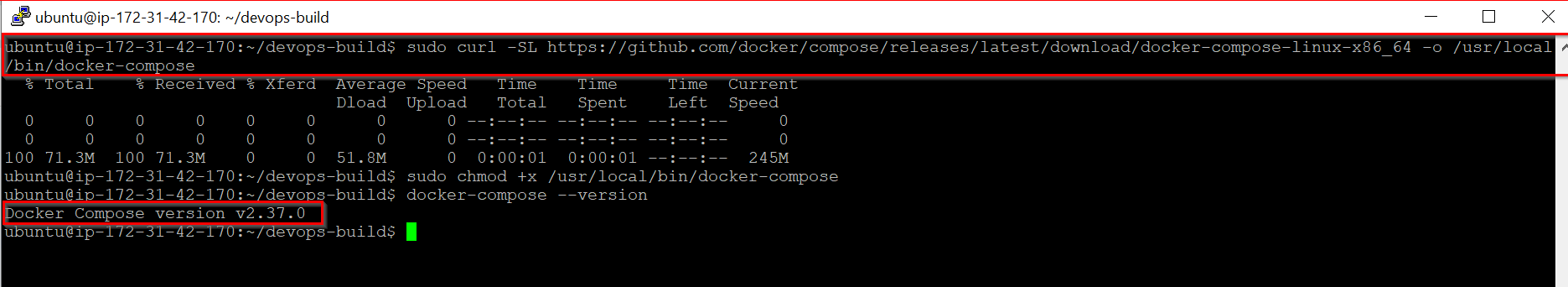


**Install Docker Compose:**

sudo curl -SL https://github.com/docker/compose/releases/latest/download/docker-compose-linux-x86\_64 -o /usr/local/bin/docker-compose

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

docker-compose –version



**Create Dockerfile**

FROM nginx:alpine

WORKDIR /usr/share/nginx/html

RUN rm -rf ./\*

COPY build/ .

EXPOSE 80

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

**Create docker-compose.yml**

version: '3'

services:

react\_app:

build: .

image: prasanth0003/react\_app

container\_name: react\_app

ports:

- "80:80"

restart: unless-stopped

**Create .dockerignore**

node\_modules

.git

.gitignore

**Create .gitignore**

node\_modules/

.DS\_Store

.env

dist/

**Bash Scripting**

**build.sh**

#!/bin/bash

docker build -t prasanth0003/react\_app:dev .

**deploy.sh**

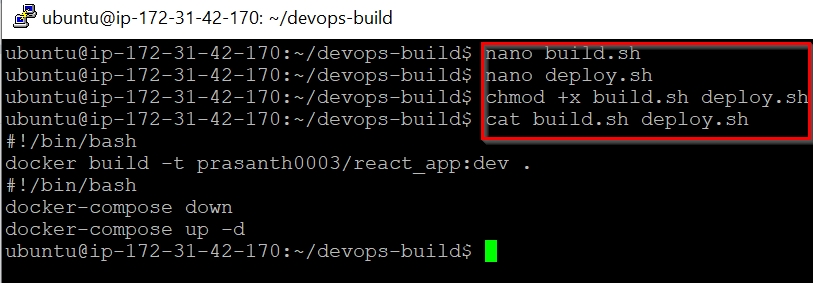
#!/bin/bash

docker-compose down

docker-compose up -d

**Change .sh file permission executable:**

chmod +x build.sh deploy.sh



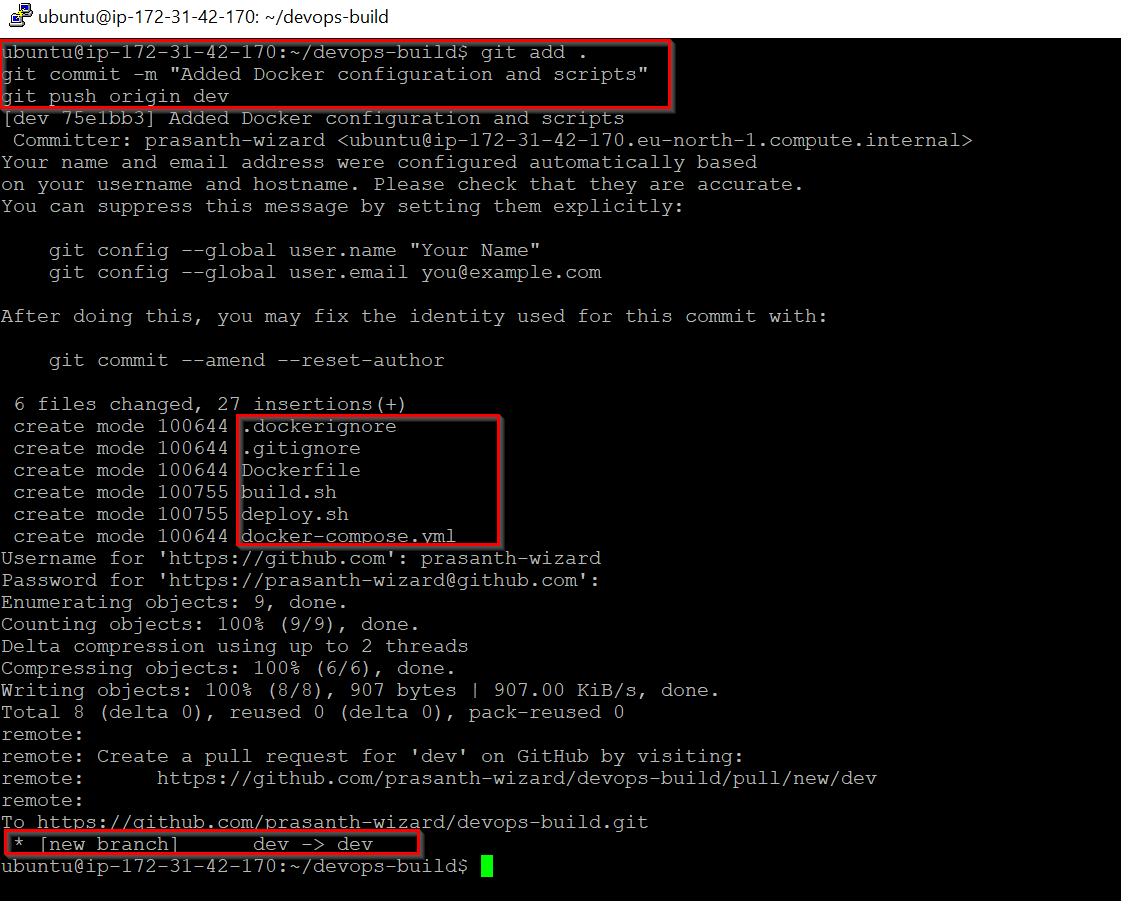
**Version Control**

**Git Commands to Push to Dev Branch**

git add .

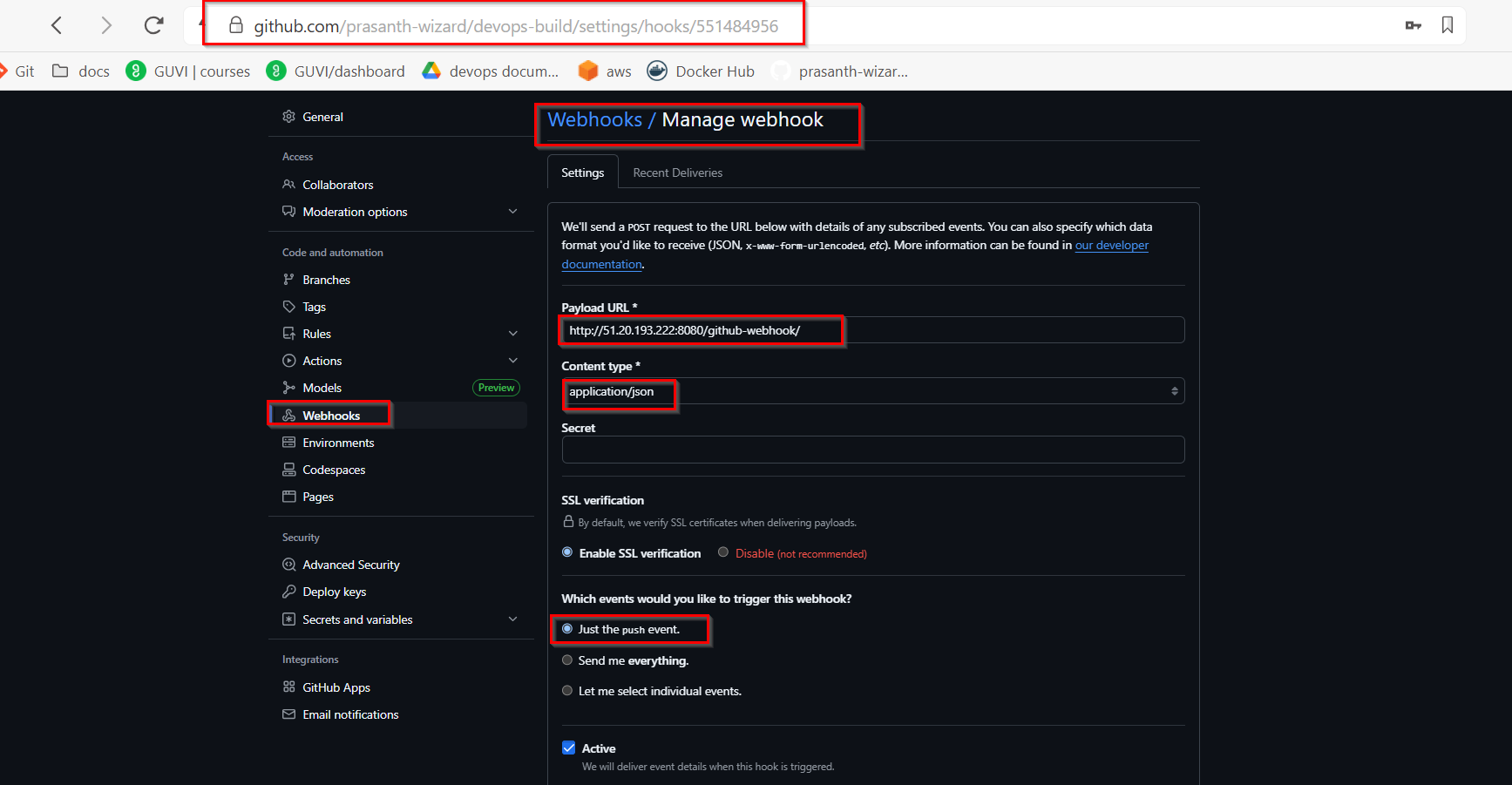
git commit -m "Added Docker configuration and scripts"

git push origin dev



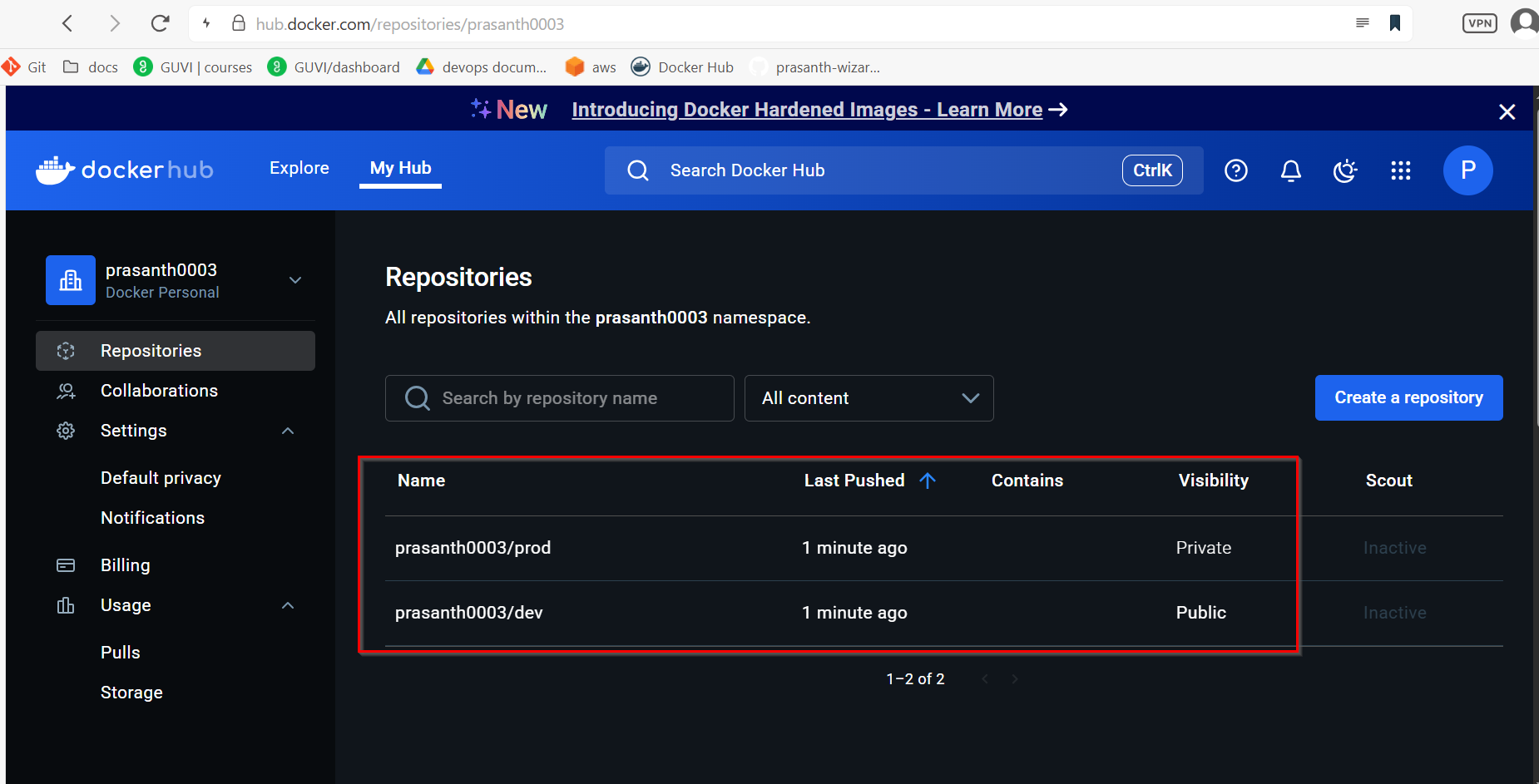
**Set Up GitHub Webhook:**

1. GitHub repository -> Settings -> Webhooks
2. Add webhook:
   * Payload URL: <http://51.20.193.222:8080/github-webhook/>
   * Content type: application/json
   * Which events: Just the push event
3. Save

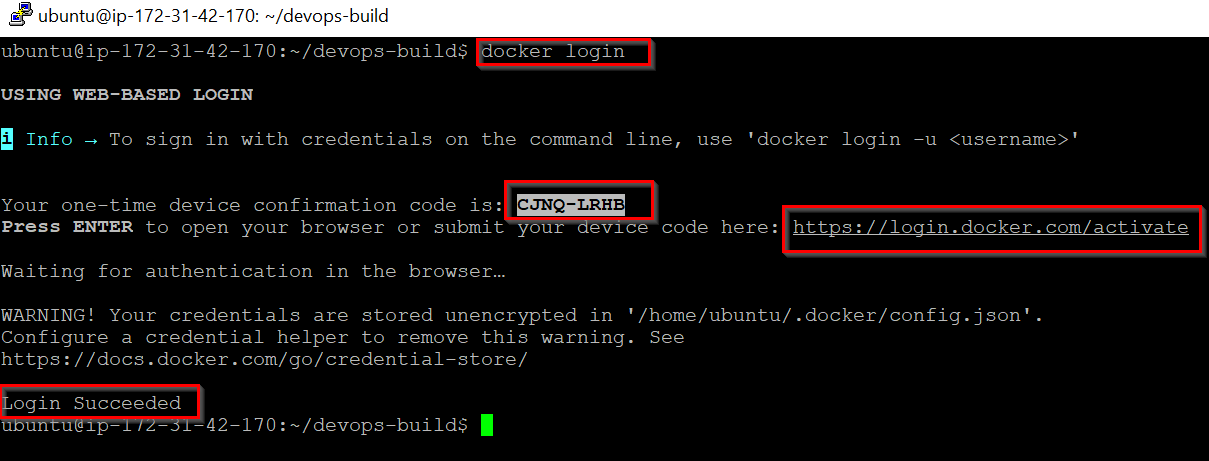
****

**Docker Hub Setup**

1. Creating two repositories:
   * "dev" (Public)
   * "prod" (Private)

****

**Docker login:**

****

**Jenkins Setup**

**Install Java 17**

sudo apt update

sudo apt install openjdk-17-jdk -y

java --version

**Install Jenkins**

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

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

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

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

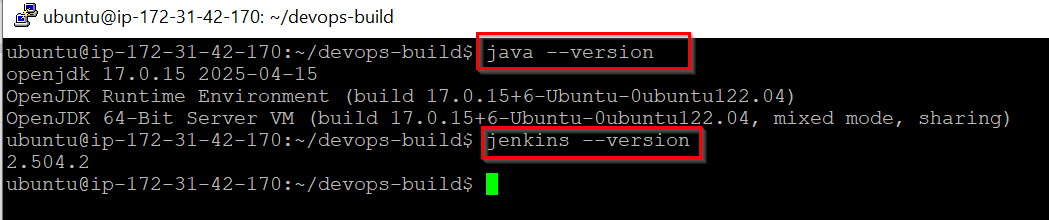
/etc/apt/sources.list.d/jenkins.list > /dev/null

sudo apt update

sudo apt install jenkins -y

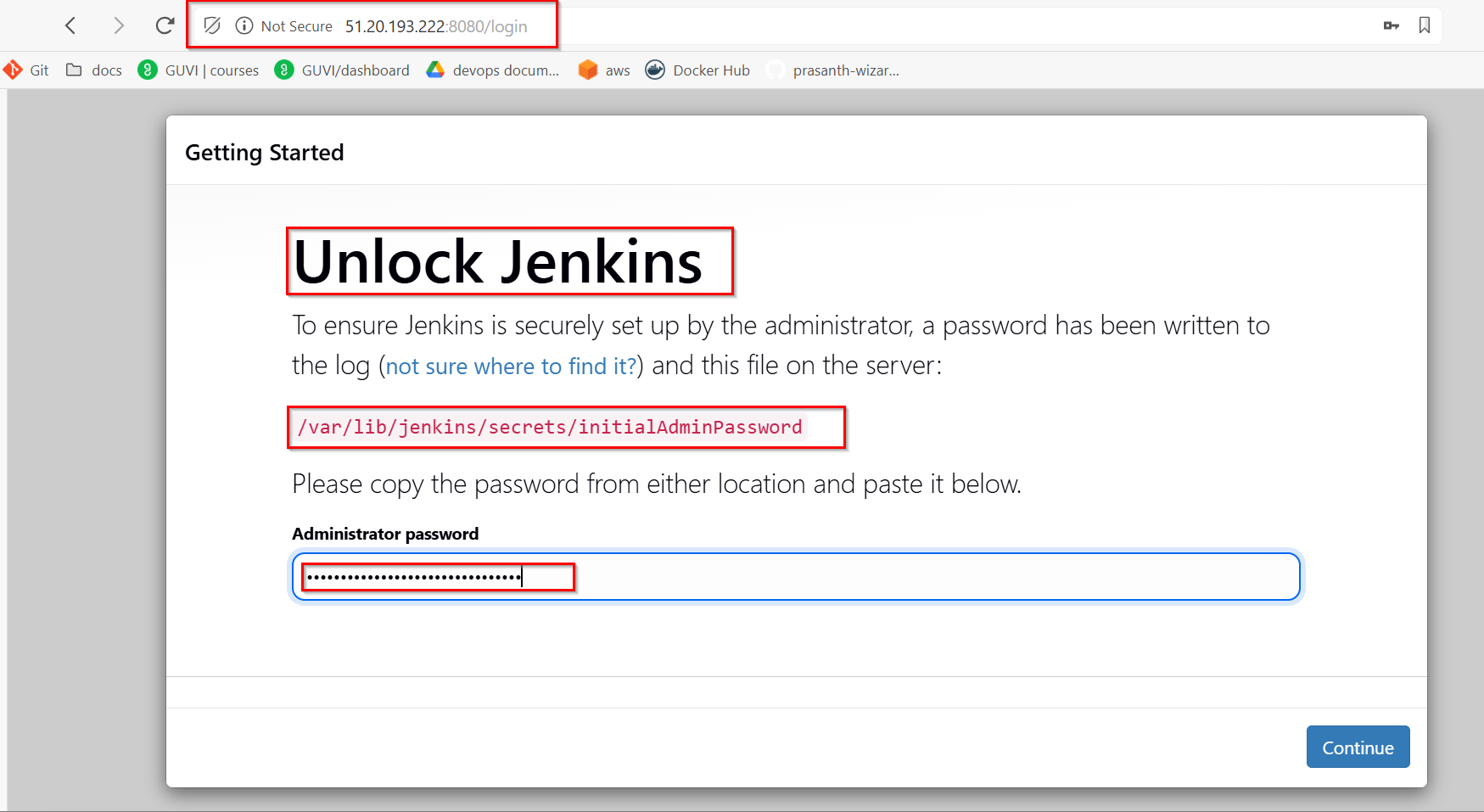
sudo systemctl start jenkins

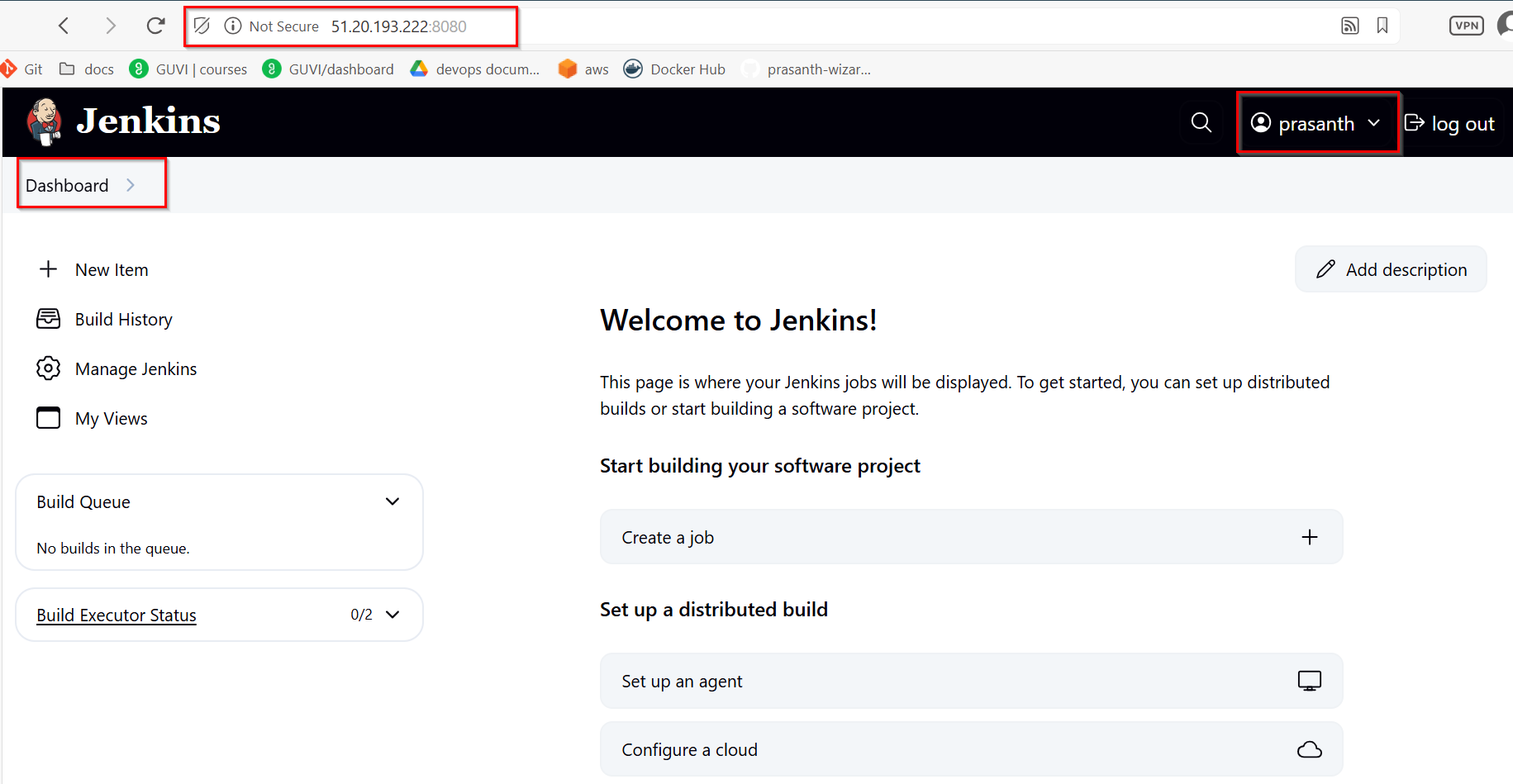
sudo systemctl enable Jenkins



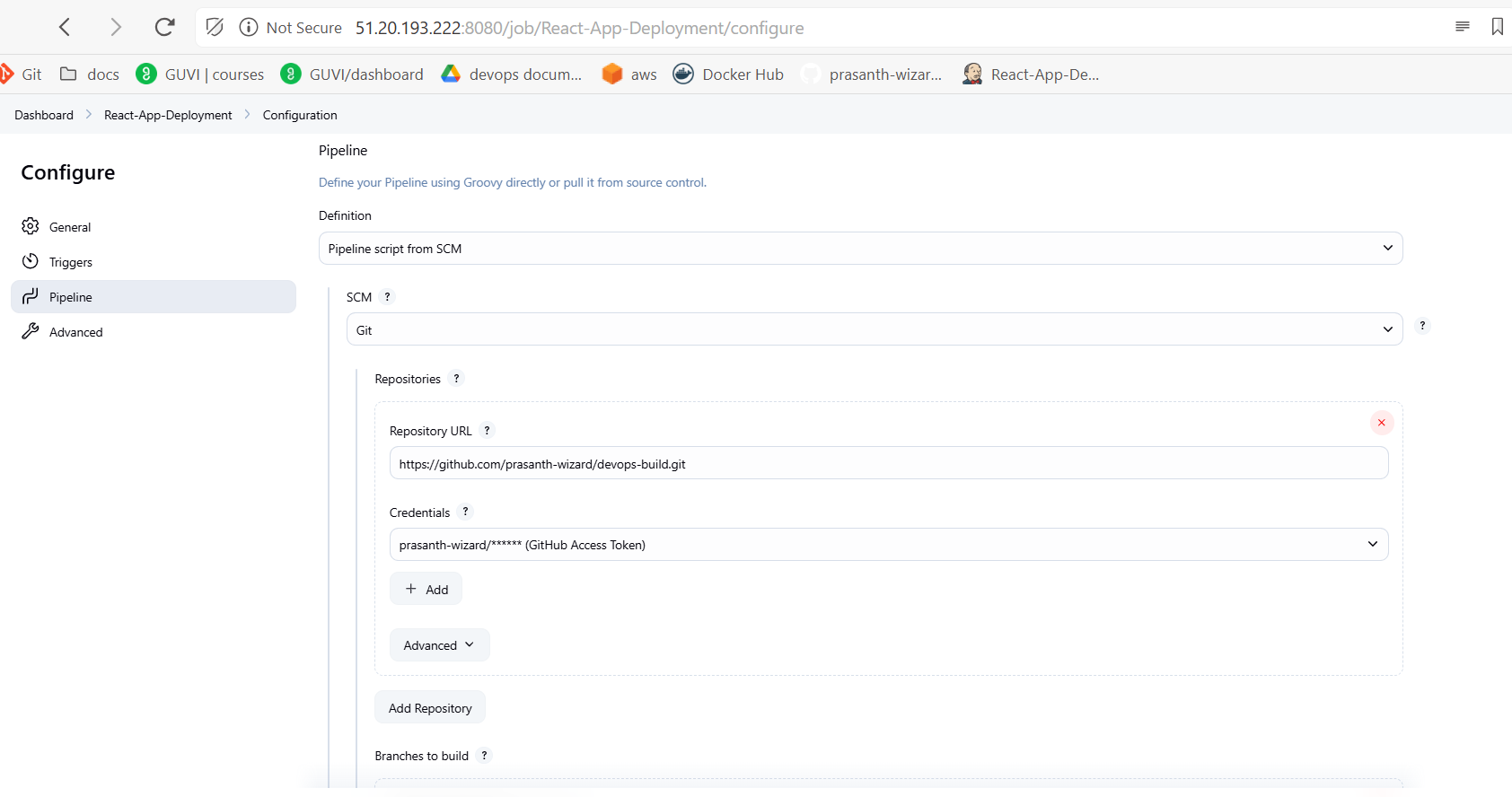
**Access Jenkins**

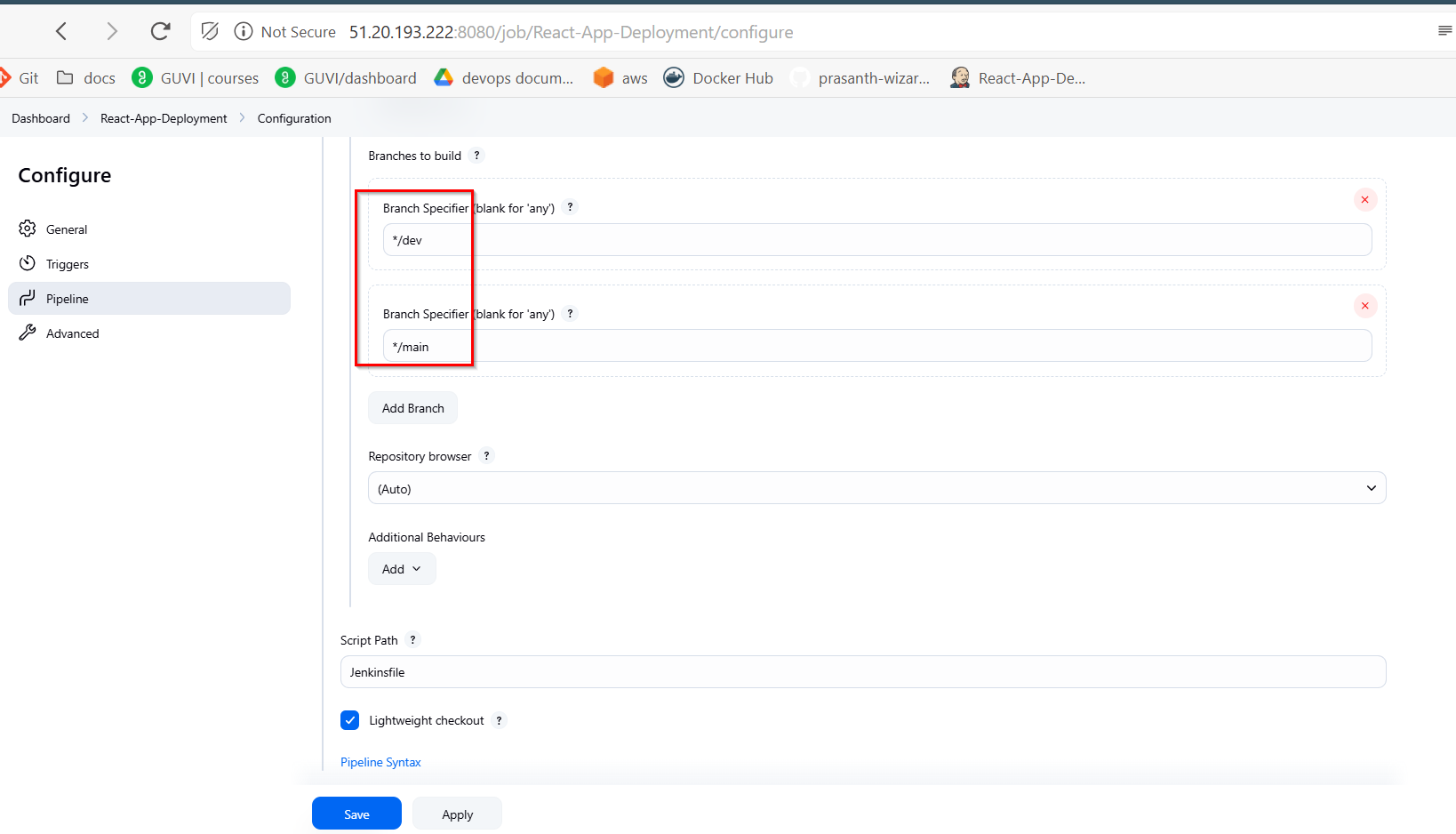
1. Open a browser to [http://51.20.193.222:8080](http://51.20.193.222:8080/)
2. Get admin password: sudo cat /var/lib/jenkins/secrets/initialAdminPassword
3. Complete setup wizard, install suggested plugins





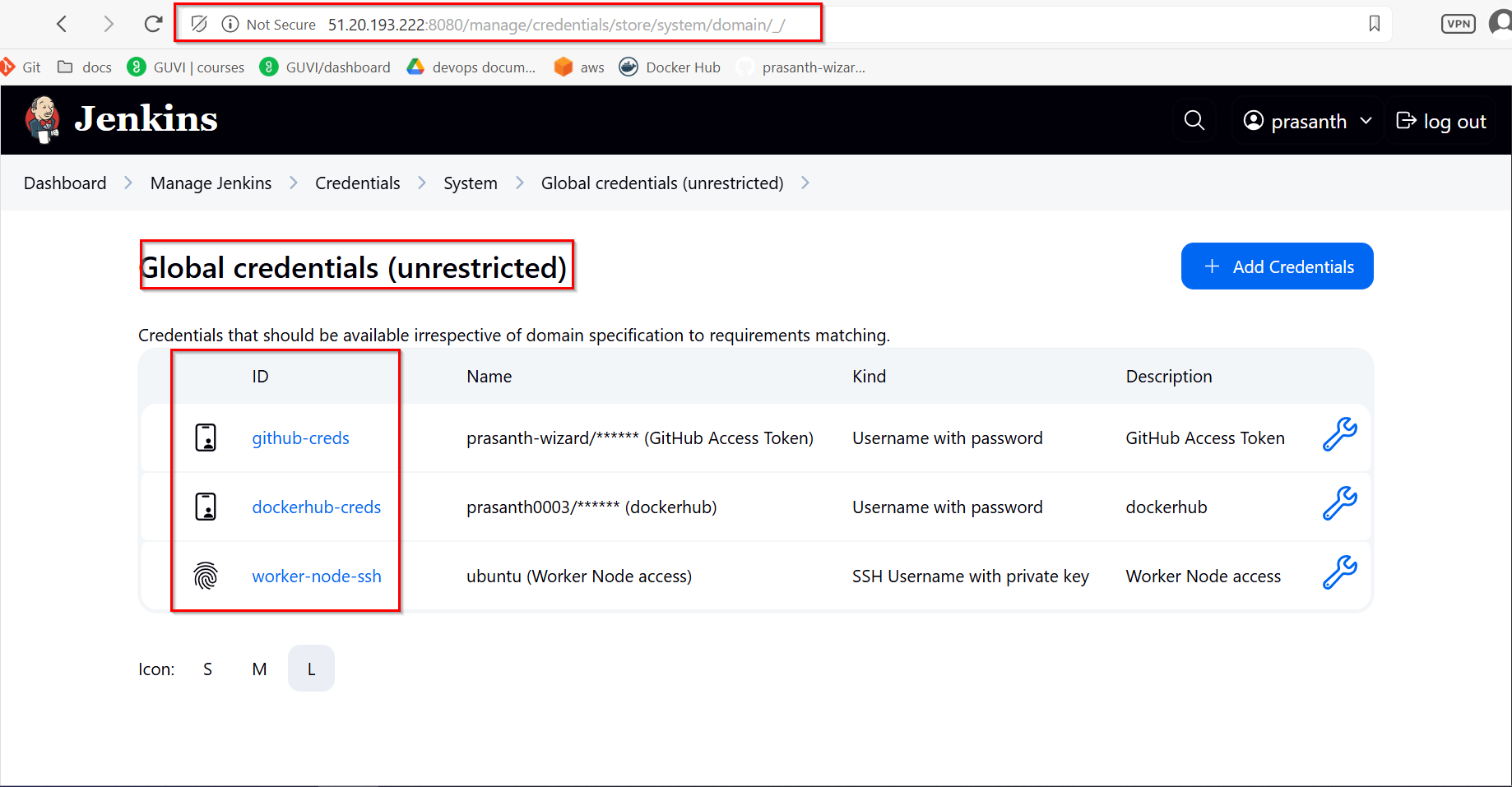
**Configure Jenkins Pipeline:**





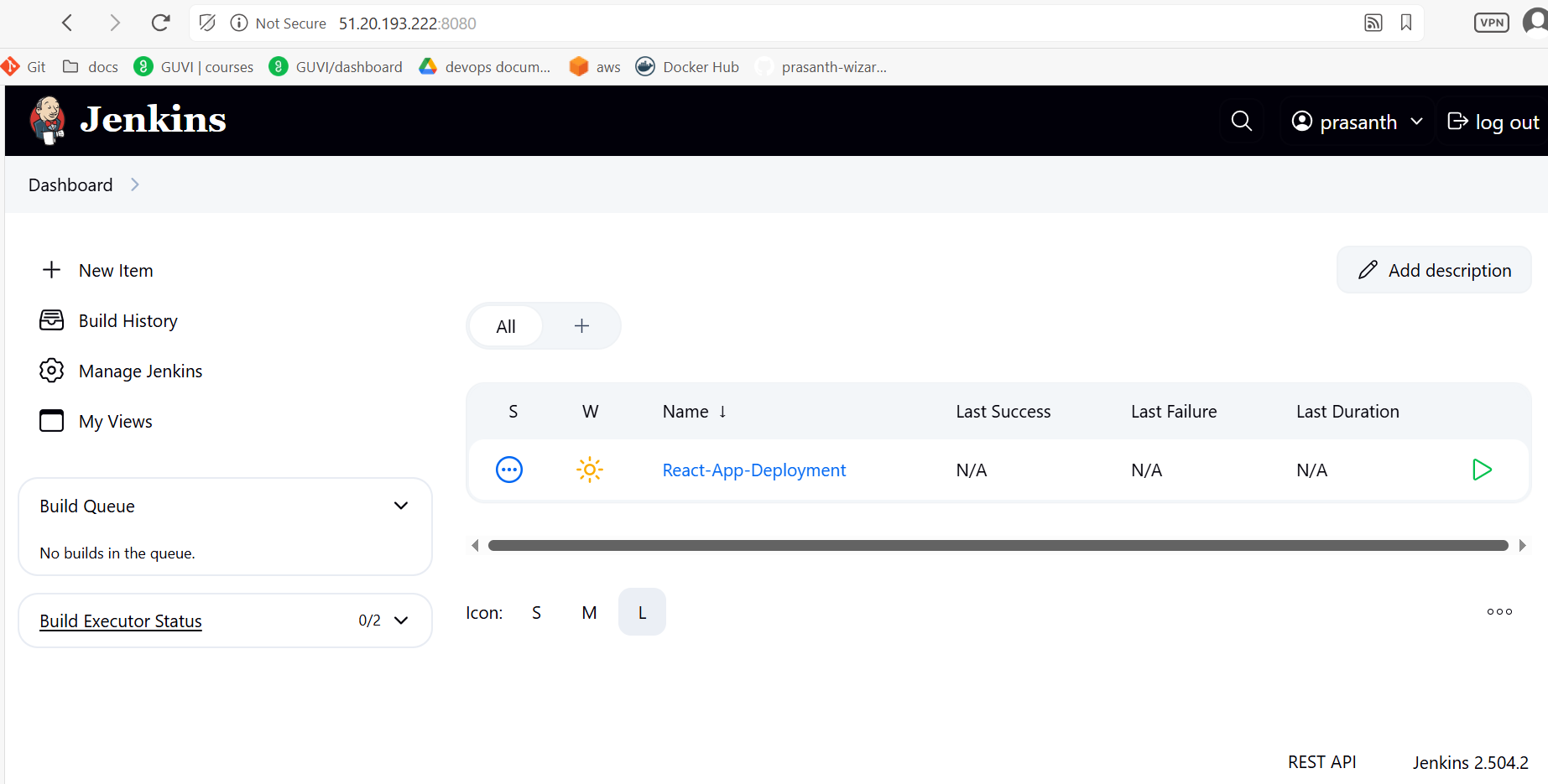
**Add Credentials in Jenkins**

1. Add three credentials:
   * Git Hub credentials (Username with password)
     + ID: github-creds
     + Username: prasanth-wizard
     + Password: Access Token
   * Docker Hub credentials (Username with password)
     + ID: dockerhub-creds
     + Username: prasanth0003
     + Password: Access Token
   * SSH credentials for Worker Node
     + ID: worker-node-ssh
     + Username: ubuntu
     + Private key: SSH private key

****

**Creating Pipeline:**

1. Create new item -> Pipeline
2. Name: "React-App-Deployment"
3. Under Pipeline section:
   * Definition: Pipeline script from SCM
   * SCM: Git
   * Repository URL: <https://github.com/prasanth-wizard/devops-build.git>
   * Credentials: Added GitHub credentials
   * Branches to build: \*/dev and \*/main
   * Script Path: Jenkinsfile

****

**Create Jenkinsfile**

**pipeline {**

**agent any**

**environment {**

**DOCKER\_DEV\_REPO = "prasanth0003/dev"**

**DOCKER\_PROD\_REPO = "prasanth0003/prod"**

**DOCKER\_REGISTRY = "docker.io"**

**AGENT\_IP = "51.20.2.247"**

**AGENT\_SSH\_CREDS = "agent-1-ssh-creds"**

**}**

**stages {**

**stage('Checkout Code') {**

**steps {**

**checkout scm**

**script {**

**env.COMMIT\_HASH = sh(**

**script: 'git rev-parse --short=7 HEAD',**

**returnStdout: true**

**).trim()**

**env.BRANCH\_NAME = env.GIT\_BRANCH?.replace("origin/", "") ?: sh(**

**script: "git rev-parse --abbrev-ref HEAD",**

**returnStdout: true**

**).trim()**

**if (env.BRANCH\_NAME == 'dev') {**

**env.BASE\_IMAGE = "${DOCKER\_REGISTRY}/${DOCKER\_DEV\_REPO}"**

**} else if (env.BRANCH\_NAME == 'main') {**

**env.BASE\_IMAGE = "${DOCKER\_REGISTRY}/${DOCKER\_PROD\_REPO}"**

**} else {**

**error("Unsupported branch '${env.BRANCH\_NAME}'. Only 'dev' and 'main' are allowed.")**

**}**

**// Define tags as a string joined by spaces for docker build command**

**env.DOCKER\_TAGS = "${env.BASE\_IMAGE}:${env.COMMIT\_HASH} ${env.BASE\_IMAGE}:latest ${env.BASE\_IMAGE}:${env.BRANCH\_NAME} ${env.BASE\_IMAGE}:${env.BUILD\_NUMBER}"**

**echo "Branch: ${env.BRANCH\_NAME}"**

**echo "Image Tags: ${env.DOCKER\_TAGS}"**

**}**

**}**

**}**

**stage('Build Docker Image') {**

**steps {**

**script {**

**// Convert tags string to build arguments**

**def buildArgs = env.DOCKER\_TAGS.split().collect { "-t ${it}" }.join(' ')**

**sh """**

**docker build ${buildArgs} .**

**docker images | grep prasanth0003**

**"""**

**// Verify the image built successfully**

**def imageCheck = sh(**

**script: "docker inspect --type=image ${env.BASE\_IMAGE}:${env.COMMIT\_HASH}",**

**returnStatus: true**

**)**

**if (imageCheck != 0) {**

**error("Docker image failed to build")**

**}**

**}**

**}**

**}**

**stage('Push Docker Image') {**

**steps {**

**withCredentials([usernamePassword(**

**credentialsId: 'dockerhub-creds',**

**usernameVariable: 'DOCKER\_USER',**

**passwordVariable: 'DOCKER\_PASS'**

**)]) {**

**script {**

**sh """**

**echo "$DOCKER\_PASS" | docker login -u "$DOCKER\_USER" --password-stdin**

**"""**

**// Push each tag**

**env.DOCKER\_TAGS.split().each { tag ->**

**retry(3) {**

**sh "docker push ${tag}"**

**}**

**}**

**}**

**}**

**}**

**}**

**stage('Deploy to Agent-1') {**

**steps {**

**script {**

**withCredentials([sshUserPrivateKey(**

**credentialsId: env.AGENT\_SSH\_CREDS,**

**usernameVariable: 'SSH\_USER',**

**keyFileVariable: 'SSH\_KEY'**

**)]) {**

**// Use the commit hash tag for deployment**

**def dockerImage = "${env.BASE\_IMAGE}:${env.COMMIT\_HASH}"**

**sh """**

**ssh -o StrictHostKeyChecking=no -i $SSH\_KEY ${SSH\_USER}@${env.AGENT\_IP} "**

**docker pull ${dockerImage}**

**docker stop react-app || true**

**docker rm react-app || true**

**docker run -d --name react-app -p 80:80 ${dockerImage}**

**"**

**"""**

**echo "Successfully deployed ${dockerImage} to ${env.AGENT\_IP}"**

**}**

**}**

**}**

**}**

**}**

**post {**

**always {**

**script {**

**sh 'docker logout || true'**

**// Clean up images safely**

**try {**

**env.DOCKER\_TAGS.split().each { tag ->**

**sh "docker rmi ${tag} || true"**

**}**

**} catch (e) {**

**echo "Warning: Error during image cleanup - ${e.message}"**

**}**

**cleanWs()**

**}**

**}**

**success {**

**echo "Successfully built and pushed images with tags: ${env.DOCKER\_TAGS}"**

**}**

**failure {**

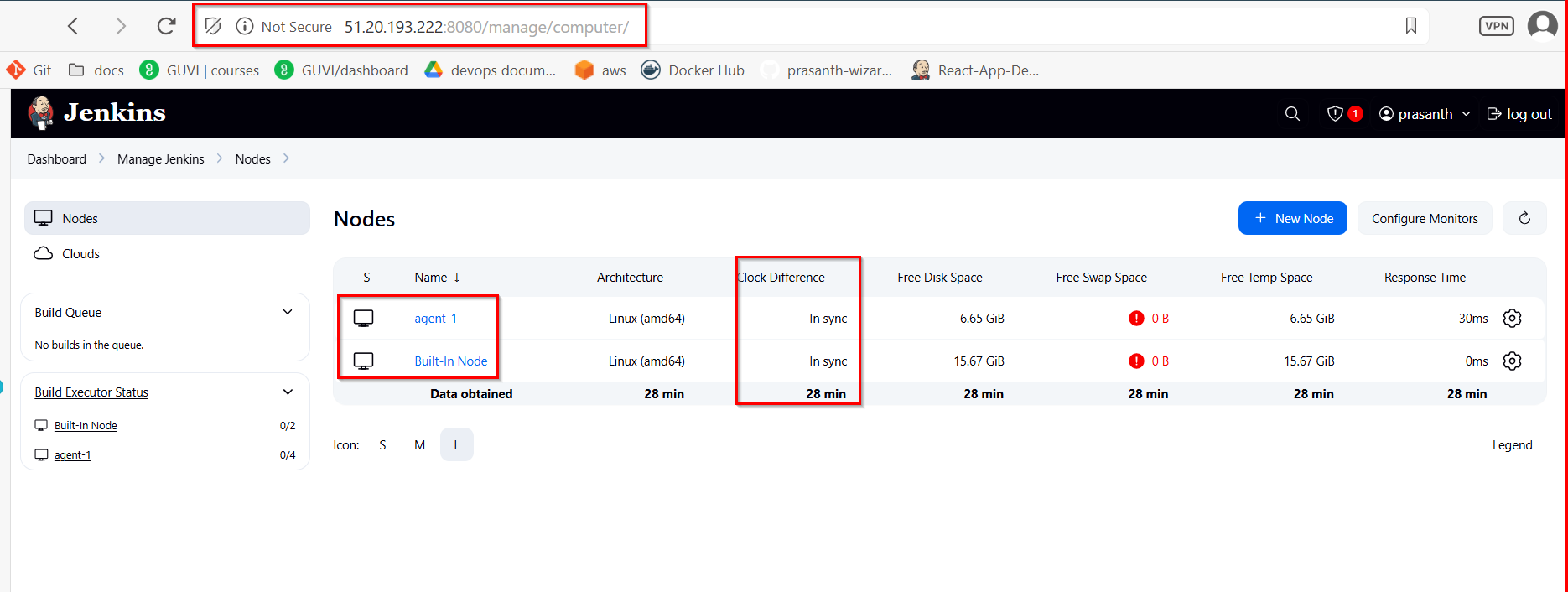
**echo "Pipeline failed for branch ${env.BRANCH\_NAME}"**

**}**

**}**

**}**

**Nodes:**

****

**Enable GitHub Webhook Trigger in Jenkins Pipeline:**

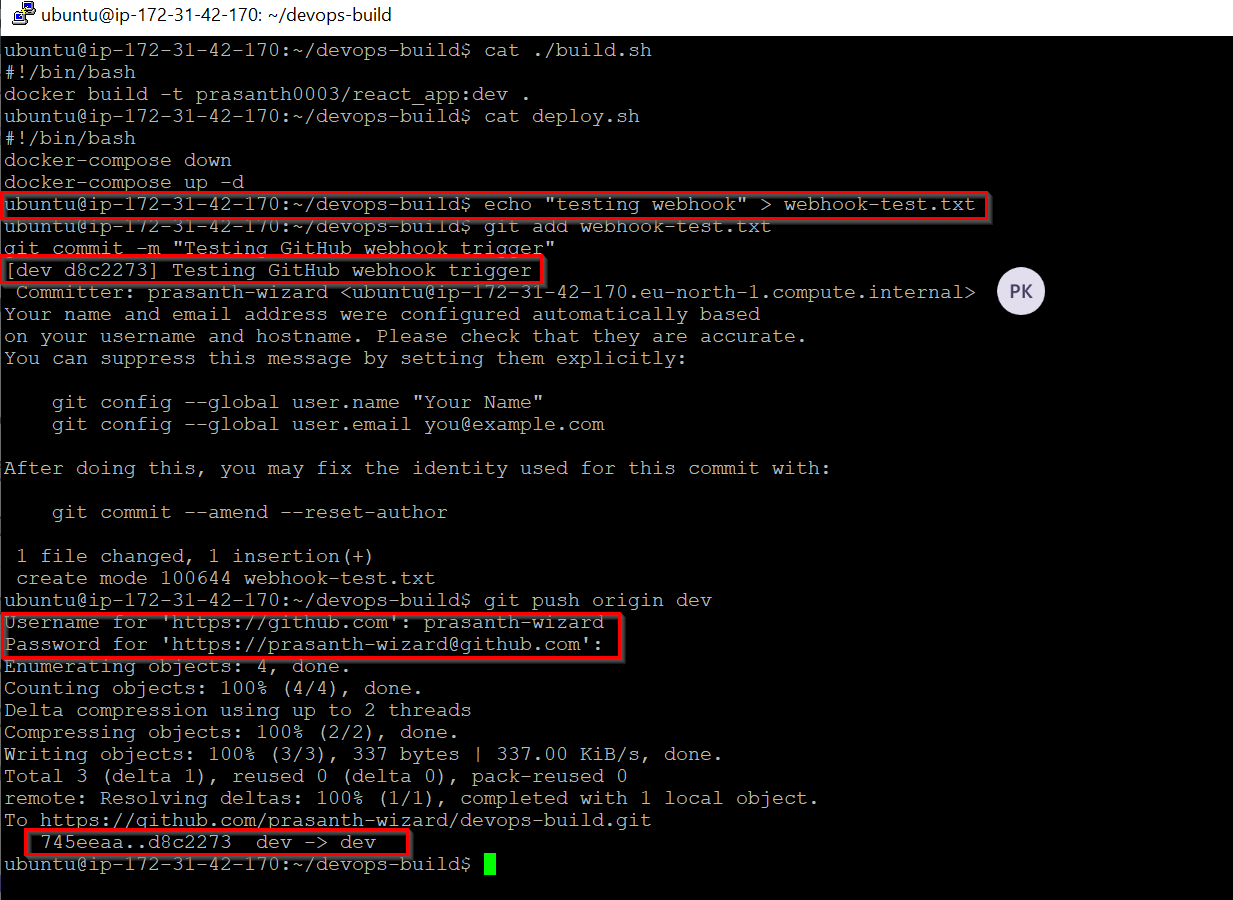
echo "testing webhook" > webhook-test.txt

git add webhook-test.txt

git commit -m "Testing GitHub webhook trigger"

[dev d8c2273] Testing GitHub webhook trigger

git push origin dev

****

**Console Output for dev branch:**

Started by GitHub push by prasanth-wizard

Lightweight checkout support not available, falling back to full checkout.

Checking out git <https://github.com/prasanth-wizard/devops-build.git> into /var/lib/jenkins/workspace/React-App-Deployment@script/6b50294159ccdb405f909776885845b2e9d6bfa09c9732cbd4b075055044ce91 to read Jenkinsfile

Selected Git installation does not exist. Using Default

The recommended git tool is: NONE

using credential github-creds

> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/React-App-Deployment@script/6b50294159ccdb405f909776885845b2e9d6bfa09c9732cbd4b075055044ce91/.git # timeout=10

Fetching changes from the remote Git repository

> git config remote.origin.url <https://github.com/prasanth-wizard/devops-build.git> # timeout=10

Fetching upstream changes from <https://github.com/prasanth-wizard/devops-build.git>

> git --version # timeout=10

> git --version # 'git version 2.34.1'

using GIT\_ASKPASS to set credentials GitHub Access Token

> git fetch --tags --force --progress -- <https://github.com/prasanth-wizard/devops-build.git> +refs/heads/\*:refs/remotes/origin/\* # timeout=10

Seen branch in repository origin/dev

Seen branch in repository origin/main

Seen 2 remote branches

> git show-ref --tags -d # timeout=10

Checking out Revision 6d2e84dec5c3ccee0e1d596cd518e222a0b97437 (origin/dev)

> git config core.sparsecheckout # timeout=10

> git checkout -f 6d2e84dec5c3ccee0e1d596cd518e222a0b97437 # timeout=10

Commit message: "jenkinsfile update"

> git rev-list --no-walk 4b8fb15bd357bd977f300d7d58ae290acbcaaa1d # timeout=10

[Pipeline] Start of Pipeline

[Pipeline] node

Running on [agent-1](http://51.20.193.222:8080/computer/agent%2D1/) in /home/ubuntu/jenkins/workspace/React-App-Deployment

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Declarative: Checkout SCM)

[Pipeline] checkout

Selected Git installation does not exist. Using Default

The recommended git tool is: NONE

using credential github-creds

Fetching changes from the remote Git repository

Checking out Revision 6d2e84dec5c3ccee0e1d596cd518e222a0b97437 (origin/dev)

Commit message: "jenkinsfile update"

[Pipeline] }

[Pipeline] // stage

[Pipeline] withEnv

[Pipeline] {

[Pipeline] withEnv

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Checkout Code)

[Pipeline] checkout

Selected Git installation does not exist. Using Default

The recommended git tool is: NONE

using credential github-creds

Fetching changes from the remote Git repository

Checking out Revision 6d2e84dec5c3ccee0e1d596cd518e222a0b97437 (origin/dev)

Commit message: "jenkinsfile update"

> git rev-parse --resolve-git-dir /home/ubuntu/jenkins/workspace/React-App-Deployment/.git # timeout=10

> git config remote.origin.url <https://github.com/prasanth-wizard/devops-build.git> # timeout=10

Fetching upstream changes from <https://github.com/prasanth-wizard/devops-build.git>

> git --version # timeout=10

> git --version # 'git version 2.34.1'

using GIT\_ASKPASS to set credentials GitHub Access Token

> git fetch --tags --force --progress -- <https://github.com/prasanth-wizard/devops-build.git> +refs/heads/\*:refs/remotes/origin/\* # timeout=10

Seen branch in repository origin/dev

Seen branch in repository origin/main

Seen 2 remote branches

> git show-ref --tags -d # timeout=10

> git config core.sparsecheckout # timeout=10

> git checkout -f 6d2e84dec5c3ccee0e1d596cd518e222a0b97437 # timeout=10

[Pipeline] script

[Pipeline] {

[Pipeline] sh

+ git rev-parse --short=7 HEAD

[Pipeline] echo

Branch: dev

[Pipeline] echo

Image Tags: docker.io/prasanth0003/dev:6d2e84d docker.io/prasanth0003/dev:latest docker.io/prasanth0003/dev:dev docker.io/prasanth0003/dev:62

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Build Docker Image)

[Pipeline] script

[Pipeline] {

> git rev-parse --resolve-git-dir /home/ubuntu/jenkins/workspace/React-App-Deployment/.git # timeout=10

> git config remote.origin.url <https://github.com/prasanth-wizard/devops-build.git> # timeout=10

Fetching upstream changes from <https://github.com/prasanth-wizard/devops-build.git>

> git --version # timeout=10

> git --version # 'git version 2.34.1'

using GIT\_ASKPASS to set credentials GitHub Access Token

> git fetch --tags --force --progress -- <https://github.com/prasanth-wizard/devops-build.git> +refs/heads/\*:refs/remotes/origin/\* # timeout=10

Seen branch in repository origin/dev

Seen branch in repository origin/main

Seen 2 remote branches

> git show-ref --tags -d # timeout=10

> git config core.sparsecheckout # timeout=10

> git checkout -f 6d2e84dec5c3ccee0e1d596cd518e222a0b97437 # timeout=10

[Pipeline] sh

+ docker build -t docker.io/prasanth0003/dev:6d2e84d -t docker.io/prasanth0003/dev:latest -t docker.io/prasanth0003/dev:dev -t docker.io/prasanth0003/dev:62 .

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 2.633MB

Step 1/6 : FROM nginx:alpine

---> 6769dc3a703c

Step 2/6 : WORKDIR /usr/share/nginx/html

---> Using cache

---> 7a981b94e632

Step 3/6 : RUN rm -rf ./\*

---> Using cache

---> 24d854bee91b

Step 4/6 : COPY build/ .

---> Using cache

---> ef3addf8967e

Step 5/6 : EXPOSE 80

---> Using cache

---> 4729f3fc2f6a

Step 6/6 : CMD ["nginx", "-g", "daemon off;"]

---> Using cache

---> 9cac7341a0dd

Successfully built 9cac7341a0dd

Successfully tagged prasanth0003/dev:6d2e84d

Successfully tagged prasanth0003/dev:latest

Successfully tagged prasanth0003/dev:dev

Successfully tagged prasanth0003/dev:62

+ docker images

+ grep prasanth0003

prasanth0003/dev 62 9cac7341a0dd 57 minutes ago 50.8MB

prasanth0003/dev 6d2e84d 9cac7341a0dd 57 minutes ago 50.8MB

prasanth0003/dev dev 9cac7341a0dd 57 minutes ago 50.8MB

prasanth0003/dev latest 9cac7341a0dd 57 minutes ago 50.8MB

prasanth0003/prod 1.0.58 9cac7341a0dd 57 minutes ago 50.8MB

prasanth0003/prod 1.0.60 9cac7341a0dd 57 minutes ago 50.8MB

prasanth0003/prod latest 9cac7341a0dd 57 minutes ago 50.8MB

prasanth0003/dev <none> 0d761b2d38d1 27 hours ago 50.8MB

[Pipeline] sh

+ docker inspect --type=image docker.io/prasanth0003/dev:6d2e84d

[

{

"Id": "sha256:9cac7341a0ddf876542c4c8898fb02f887fdf4597758a075cd37982ec40dee2d",

"RepoTags": [

"prasanth0003/dev:62",

"prasanth0003/dev:6d2e84d",

"prasanth0003/dev:dev",

"prasanth0003/dev:latest",

"prasanth0003/prod:1.0.58",

"prasanth0003/prod:1.0.60",

"prasanth0003/prod:latest"

],

"RepoDigests": [

"prasanth0003/dev@sha256:6fe52943a7b972cef9c2615d6d5a22220eb912ecfbe1ba77375f385a5da720a6",

"prasanth0003/prod@sha256:6fe52943a7b972cef9c2615d6d5a22220eb912ecfbe1ba77375f385a5da720a6"

],

"Parent": "sha256:4729f3fc2f6a950cf7f7dd9430e0159dbe05346d30f414174142d7c0019cd45e",

"Comment": "",

"Created": "2025-06-11T17:08:50.798025408Z",

"DockerVersion": "27.5.1",

"Author": "",

"Config": {

"Hostname": "",

"Domainname": "",

"User": "",

"AttachStdin": false,

"AttachStdout": false,

"AttachStderr": false,

"ExposedPorts": {

"80/tcp": {}

},

"Tty": false,

"OpenStdin": false,

"StdinOnce": false,

"Env": [

"PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",

"NGINX\_VERSION=1.27.5",

"PKG\_RELEASE=1",

"DYNPKG\_RELEASE=1",

"NJS\_VERSION=0.8.10",

"NJS\_RELEASE=1"

],

"Cmd": [

"nginx",

"-g",

"daemon off;"

],

"Image": "sha256:4729f3fc2f6a950cf7f7dd9430e0159dbe05346d30f414174142d7c0019cd45e",

"Volumes": null,

"WorkingDir": "/usr/share/nginx/html",

"Entrypoint": [

"/docker-entrypoint.sh"

],

"OnBuild": null,

"Labels": {

"maintainer": "NGINX Docker Maintainers <docker-maint@nginx.com>"

},

"StopSignal": "SIGQUIT"

},

"Architecture": "amd64",

"Os": "linux",

"Size": 50842926,

"GraphDriver": {

"Data": {

"LowerDir": "/var/lib/docker/overlay2/b7384d2a13d0d8dd222103044ef3e0ecb18e20e7f49360ac4e1bad206c0696b9/diff:/var/lib/docker/overlay2/2e53d516ba6efe288ee1c3e1910c0ec144704e03bc624c5334a978ab64fa787c/diff:/var/lib/docker/overlay2/063be06a73b0e59d3e23d3fc604d70c6efa113c40d645d32e074d78544dbfc70/diff:/var/lib/docker/overlay2/bfe6a78617ea0c01fd83530f7a63ca2da6588277bfe05f3ea9b27afa8708c3bc/diff:/var/lib/docker/overlay2/0526f6e2120601cee541b71f55573e0bd52e01d8a0ce3828da59cdd45dc38b87/diff:/var/lib/docker/overlay2/e99e398efb3034da28bcb2fb58147808859cafafa01919e78746f60d99e5ecd7/diff:/var/lib/docker/overlay2/1e2d5648db4b7db345be0a075d7ec3dcf9b16400a06343220bd256408f138b2f/diff:/var/lib/docker/overlay2/66a68b332a6872a0907ce4d714372c176673b96e6357aca642f5cd2225e9d080/diff:/var/lib/docker/overlay2/5ace7f0c811f4ccea9b85c7ecf0118cd6d2d209e76bf98ad607589e34ad2c2eb/diff",

"MergedDir": "/var/lib/docker/overlay2/070b7b4559afdfd2d317acf9ca7531427b88a9c9c1fc4314276999846a474146/merged",

"UpperDir": "/var/lib/docker/overlay2/070b7b4559afdfd2d317acf9ca7531427b88a9c9c1fc4314276999846a474146/diff",

"WorkDir": "/var/lib/docker/overlay2/070b7b4559afdfd2d317acf9ca7531427b88a9c9c1fc4314276999846a474146/work"

},

"Name": "overlay2"

},

"RootFS": {

"Type": "layers",

"Layers": [

"sha256:08000c18d16dadf9553d747a58cf44023423a9ab010aab96cf263d2216b8b350",

"sha256:d71eae0084c1aa823dd8fb2ecf8604d5c0f4911226c042bb1f8297e819f4b192",

"sha256:c56f134d380585340a68d0db2f2c170641a1c0ff72ccf2438cf2f693df756a85",

"sha256:e244aa659f612a80c40dd8645812301e3def6b15ec67b9e486ed2201172b51d1",

"sha256:b8d7d1d2263425d6044e059b2810017d062d659b9b755241f3747eda77726250",

"sha256:811a4dbbf4a5309e4390cf655c12db92e1a4304fb9d9731f83e7b02e95a617c6",

"sha256:947e805a4ac71f68e6703550c0b36c2aa2e554c4fa670ca2da6a25c6d7dccb66",

"sha256:0d853d50b128aa460b47e7121849463a14b18d4fd976caf5014744aae24d28aa",

"sha256:0741406cbd2ef066f67447fc167b14bb4451821de55541197fe2fa6a11d50c2c",

"sha256:320d0d9d1060d5ea2a301d007c3f7b3646a109583286a53031dee51ffecbf5a2"

]

},

"Metadata": {

"LastTagTime": "2025-06-11T18:05:55.684599876Z"

}

}

]

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Push Docker Image)

[Pipeline] withCredentials

Masking supported pattern matches of $DOCKER\_PASS

[Pipeline] {

[Pipeline] script

[Pipeline] {

[Pipeline] sh

Warning: A secret was passed to "sh" using Groovy String interpolation, which is insecure.

Affected argument(s) used the following variable(s): [DOCKER\_PASS]

See <https://jenkins.io/redirect/groovy-string-interpolation> for details.

+ echo \*\*\*\*

+ docker login -u prasanth0003 --password-stdin

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/#credential-stores>

Login Succeeded

[Pipeline] retry

[Pipeline] {

[Pipeline] sh

+ docker push docker.io/prasanth0003/dev:6d2e84d

The push refers to repository [docker.io/prasanth0003/dev]

320d0d9d1060: Preparing

0741406cbd2e: Preparing

0d853d50b128: Preparing

947e805a4ac7: Preparing

811a4dbbf4a5: Preparing

b8d7d1d22634: Preparing

e244aa659f61: Preparing

c56f134d3805: Preparing

d71eae0084c1: Preparing

08000c18d16d: Preparing

b8d7d1d22634: Waiting

e244aa659f61: Waiting

c56f134d3805: Waiting

d71eae0084c1: Waiting

08000c18d16d: Waiting

947e805a4ac7: Layer already exists

0741406cbd2e: Layer already exists

0d853d50b128: Layer already exists

811a4dbbf4a5: Layer already exists

320d0d9d1060: Layer already exists

c56f134d3805: Layer already exists

b8d7d1d22634: Layer already exists

e244aa659f61: Layer already exists

08000c18d16d: Layer already exists

d71eae0084c1: Layer already exists

6d2e84d: digest: sha256:6fe52943a7b972cef9c2615d6d5a22220eb912ecfbe1ba77375f385a5da720a6 size: 2406

[Pipeline] }

[Pipeline] // retry

[Pipeline] retry

[Pipeline] {

[Pipeline] sh

+ docker push docker.io/prasanth0003/dev:latest

The push refers to repository [docker.io/prasanth0003/dev]

320d0d9d1060: Preparing

0741406cbd2e: Preparing

0d853d50b128: Preparing

947e805a4ac7: Preparing

811a4dbbf4a5: Preparing

b8d7d1d22634: Preparing

e244aa659f61: Preparing

c56f134d3805: Preparing

d71eae0084c1: Preparing

08000c18d16d: Preparing

b8d7d1d22634: Waiting

e244aa659f61: Waiting

c56f134d3805: Waiting

d71eae0084c1: Waiting

08000c18d16d: Waiting

0d853d50b128: Layer already exists

0741406cbd2e: Layer already exists

947e805a4ac7: Layer already exists

320d0d9d1060: Layer already exists

811a4dbbf4a5: Layer already exists

b8d7d1d22634: Layer already exists

d71eae0084c1: Layer already exists

c56f134d3805: Layer already exists

e244aa659f61: Layer already exists

08000c18d16d: Layer already exists

latest: digest: sha256:6fe52943a7b972cef9c2615d6d5a22220eb912ecfbe1ba77375f385a5da720a6 size: 2406

[Pipeline] }

[Pipeline] // retry

[Pipeline] retry

[Pipeline] {

[Pipeline] sh

+ docker push docker.io/prasanth0003/dev:dev

The push refers to repository [docker.io/prasanth0003/dev]

320d0d9d1060: Preparing

0741406cbd2e: Preparing

0d853d50b128: Preparing

947e805a4ac7: Preparing

811a4dbbf4a5: Preparing

b8d7d1d22634: Preparing

e244aa659f61: Preparing

c56f134d3805: Preparing

d71eae0084c1: Preparing

08000c18d16d: Preparing

b8d7d1d22634: Waiting

e244aa659f61: Waiting

c56f134d3805: Waiting

d71eae0084c1: Waiting

08000c18d16d: Waiting

947e805a4ac7: Layer already exists

320d0d9d1060: Layer already exists

811a4dbbf4a5: Layer already exists

0d853d50b128: Layer already exists

0741406cbd2e: Layer already exists

e244aa659f61: Layer already exists

b8d7d1d22634: Layer already exists

c56f134d3805: Layer already exists

08000c18d16d: Layer already exists

d71eae0084c1: Layer already exists

dev: digest: sha256:6fe52943a7b972cef9c2615d6d5a22220eb912ecfbe1ba77375f385a5da720a6 size: 2406

[Pipeline] }

[Pipeline] // retry

[Pipeline] retry

[Pipeline] {

[Pipeline] sh

+ docker push docker.io/prasanth0003/dev:62

The push refers to repository [docker.io/prasanth0003/dev]

320d0d9d1060: Preparing

0741406cbd2e: Preparing

0d853d50b128: Preparing

947e805a4ac7: Preparing

811a4dbbf4a5: Preparing

b8d7d1d22634: Preparing

e244aa659f61: Preparing

c56f134d3805: Preparing

d71eae0084c1: Preparing

08000c18d16d: Preparing

b8d7d1d22634: Waiting

e244aa659f61: Waiting

c56f134d3805: Waiting

d71eae0084c1: Waiting

08000c18d16d: Waiting

947e805a4ac7: Layer already exists

0741406cbd2e: Layer already exists

0d853d50b128: Layer already exists

320d0d9d1060: Layer already exists

811a4dbbf4a5: Layer already exists

b8d7d1d22634: Layer already exists

d71eae0084c1: Layer already exists

c56f134d3805: Layer already exists

e244aa659f61: Layer already exists

08000c18d16d: Layer already exists

62: digest: sha256:6fe52943a7b972cef9c2615d6d5a22220eb912ecfbe1ba77375f385a5da720a6 size: 2406

[Pipeline] }

[Pipeline] // retry

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // withCredentials

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Deploy to Agent-1)

[Pipeline] script

[Pipeline] {

[Pipeline] withCredentials

Masking supported pattern matches of $SSH\_KEY

[Pipeline] {

[Pipeline] sh

Warning: A secret was passed to "sh" using Groovy String interpolation, which is insecure.

Affected argument(s) used the following variable(s): [SSH\_KEY]

See <https://jenkins.io/redirect/groovy-string-interpolation> for details.

+ ssh -o StrictHostKeyChecking=no -i \*\*\*\* ubuntu@51.20.2.247

docker pull docker.io/prasanth0003/dev:6d2e84d

docker stop react-app || true

docker rm react-app || true

docker run -d --name react-app -p 80:80 docker.io/prasanth0003/dev:6d2e84d

6d2e84d: Pulling from prasanth0003/dev

Digest: sha256:6fe52943a7b972cef9c2615d6d5a22220eb912ecfbe1ba77375f385a5da720a6

Status: Image is up to date for prasanth0003/dev:6d2e84d

docker.io/prasanth0003/dev:6d2e84d

react-app

react-app

af17b8a73491244f73310c5acc58fe7765d70c9a4a44eee6646b60e77fd734bb

[Pipeline] echo

🚀 Successfully deployed docker.io/prasanth0003/dev:6d2e84d to 51.20.2.247

[Pipeline] }

[Pipeline] // withCredentials

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Declarative: Post Actions)

[Pipeline] script

[Pipeline] {

[Pipeline] sh

+ docker logout

Removing login credentials for <https://index.docker.io/v1/>

[Pipeline] sh

+ docker rmi docker.io/prasanth0003/dev:6d2e84d

Untagged: prasanth0003/dev:6d2e84d

[Pipeline] sh

+ docker rmi docker.io/prasanth0003/dev:latest

Untagged: prasanth0003/dev:latest

[Pipeline] sh

+ docker rmi docker.io/prasanth0003/dev:dev

Untagged: prasanth0003/dev:dev

[Pipeline] sh

+ docker rmi docker.io/prasanth0003/dev:62

Untagged: prasanth0003/dev:62

Untagged: prasanth0003/dev@sha256:6fe52943a7b972cef9c2615d6d5a22220eb912ecfbe1ba77375f385a5da720a6

[Pipeline] cleanWs

[WS-CLEANUP] Deleting project workspace...

[WS-CLEANUP] Deferred wipeout is used...

[WS-CLEANUP] done

[Pipeline] }

[Pipeline] // script

[Pipeline] echo

Successfully built and pushed images with tags: docker.io/prasanth0003/dev:6d2e84d docker.io/prasanth0003/dev:latest docker.io/prasanth0003/dev:dev docker.io/prasanth0003/dev:62

[Pipeline] }

[Pipeline] // stage

[Pipeline] }

[Pipeline] // withEnv

[Pipeline] }

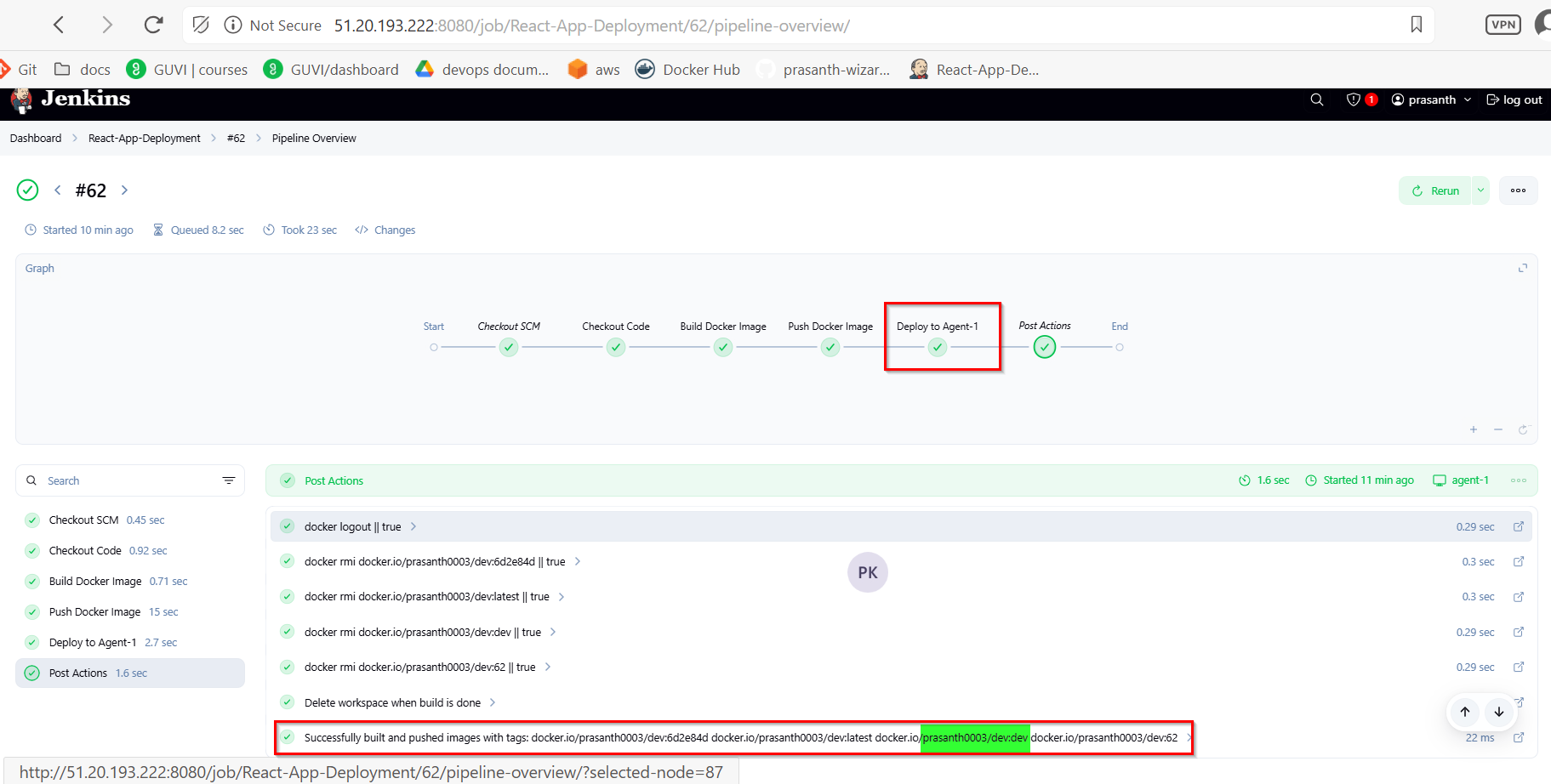
[Pipeline] // withEnv

[Pipeline] }

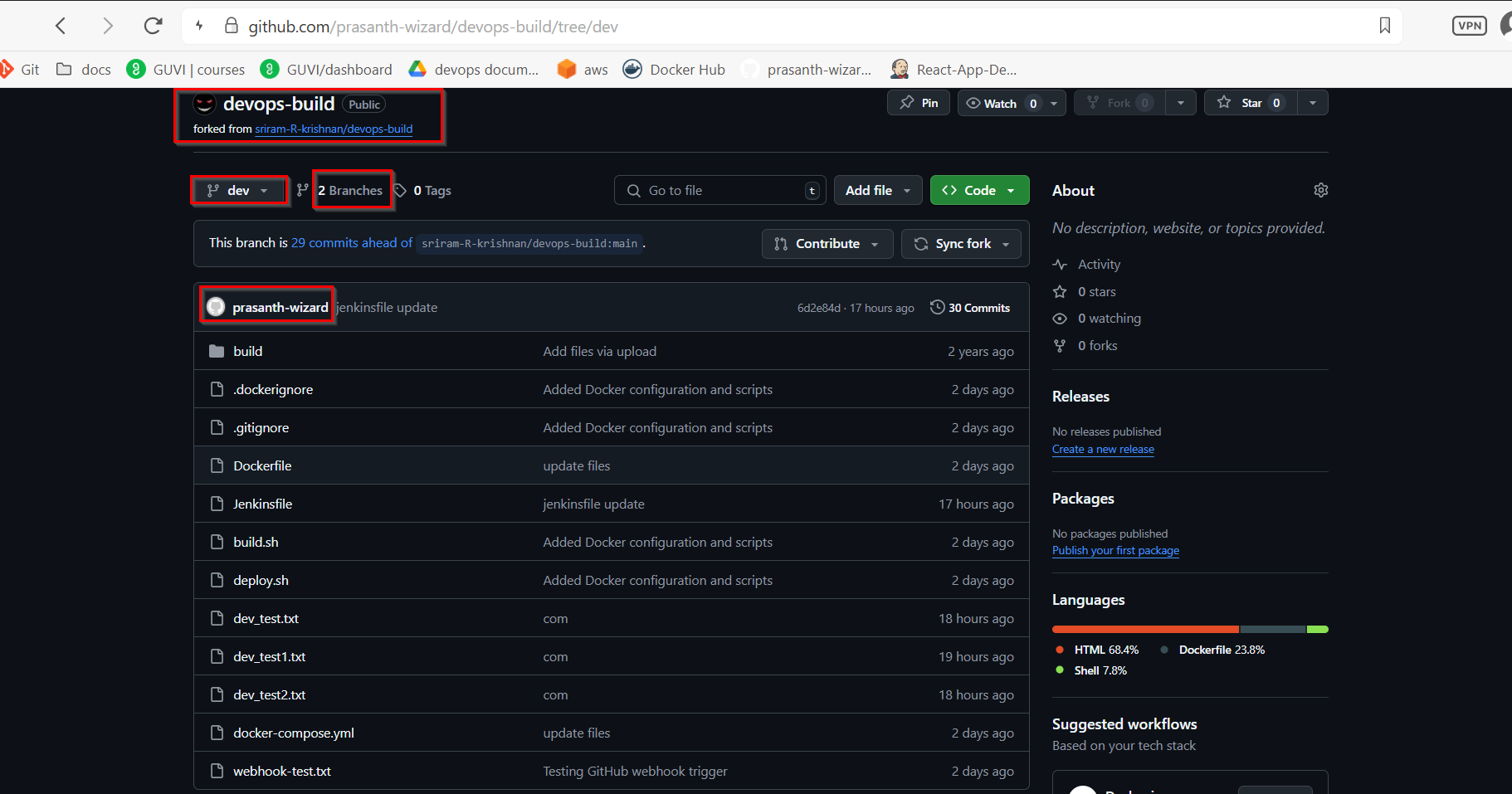
[Pipeline] // node

[Pipeline] End of Pipeline

Finished: SUCCESS

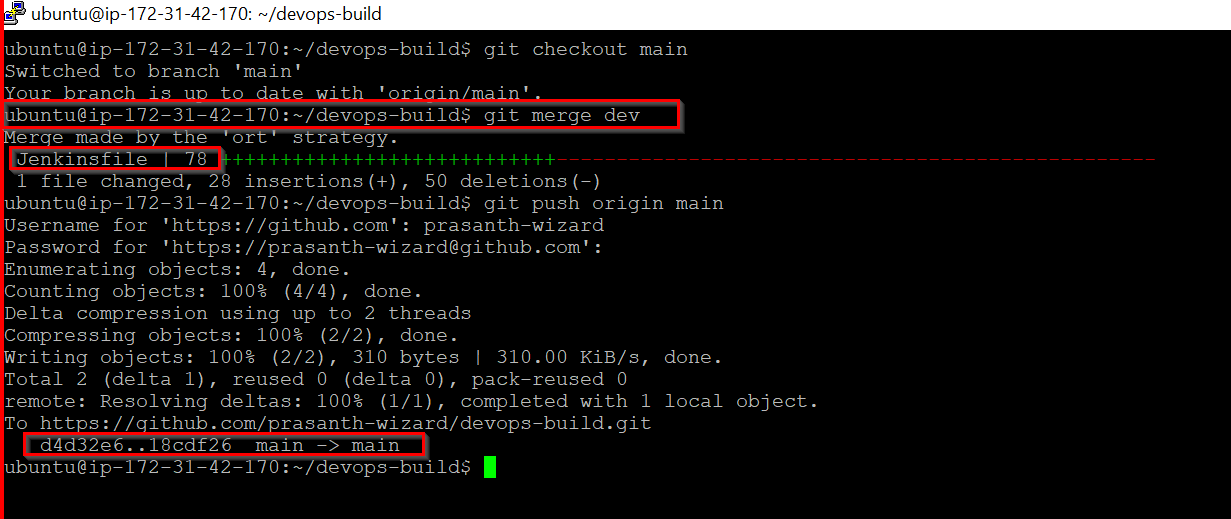
****

**Github dev branch:**

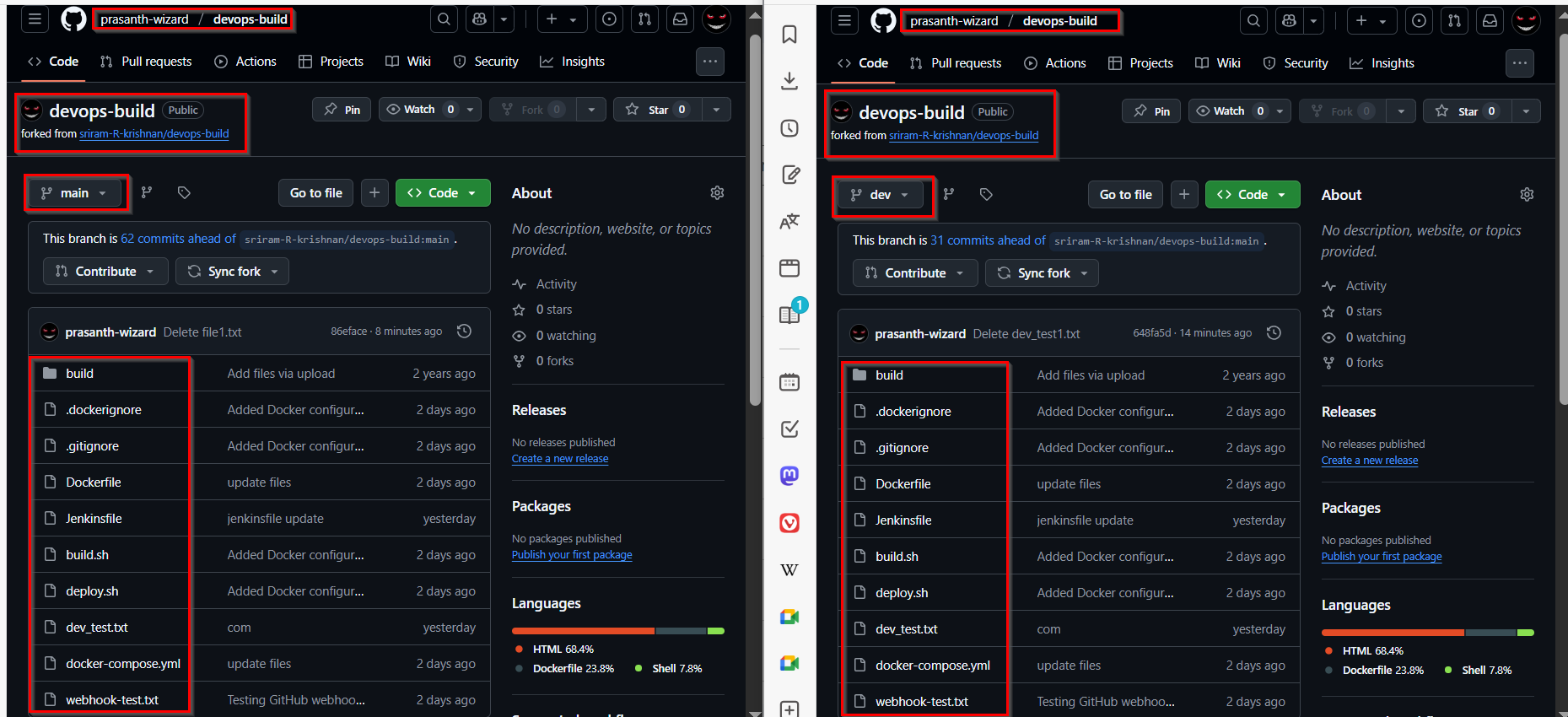
****

**Dev branch merged to main branch:**

****

****

**Github main and dev branches:**

****

**Console Output for main branch:**

Started by GitHub push by prasanth-wizard

Lightweight checkout support not available, falling back to full checkout.

Checking out git <https://github.com/prasanth-wizard/devops-build.git> into /var/lib/jenkins/workspace/React-App-Deployment@script/6b50294159ccdb405f909776885845b2e9d6bfa09c9732cbd4b075055044ce91 to read Jenkinsfile

Selected Git installation does not exist. Using Default

The recommended git tool is: NONE

using credential github-creds

> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/React-App-Deployment@script/6b50294159ccdb405f909776885845b2e9d6bfa09c9732cbd4b075055044ce91/.git # timeout=10

Fetching changes from the remote Git repository

> git config remote.origin.url <https://github.com/prasanth-wizard/devops-build.git> # timeout=10

Fetching upstream changes from <https://github.com/prasanth-wizard/devops-build.git>

> git --version # timeout=10

> git --version # 'git version 2.34.1'

using GIT\_ASKPASS to set credentials GitHub Access Token

> git fetch --tags --force --progress -- <https://github.com/prasanth-wizard/devops-build.git> +refs/heads/\*:refs/remotes/origin/\* # timeout=10

Seen branch in repository origin/dev

Seen branch in repository origin/main

Seen 2 remote branches

> git show-ref --tags -d # timeout=10

Checking out Revision 18cdf26a339a78d5f2fb474d4966bfa03af6dbd4 (origin/main)

> git config core.sparsecheckout # timeout=10

> git checkout -f 18cdf26a339a78d5f2fb474d4966bfa03af6dbd4 # timeout=10

Commit message: "Merge branch 'dev'"

> git rev-list --no-walk d4d32e693f7322d5eb419e5d99346048310d66ec # timeout=10

[Pipeline] Start of Pipeline

[Pipeline] node

Running on [agent-1](http://51.20.193.222:8080/computer/agent%2D1/) in /home/ubuntu/jenkins/workspace/React-App-Deployment

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Declarative: Checkout SCM)

[Pipeline] checkout

Selected Git installation does not exist. Using Default

The recommended git tool is: NONE

using credential github-creds

Cloning the remote Git repository

Avoid second fetch

Checking out Revision 18cdf26a339a78d5f2fb474d4966bfa03af6dbd4 (origin/main)

Commit message: "Merge branch 'dev'"

[Pipeline] }

[Pipeline] // stage

[Pipeline] withEnv

[Pipeline] {

[Pipeline] withEnv

[Pipeline] {

[Pipeline] stage

[Pipeline] { (Checkout Code)

[Pipeline] checkout

Selected Git installation does not exist. Using Default

The recommended git tool is: NONE

using credential github-creds

Fetching changes from the remote Git repository

> git rev-parse --resolve-git-dir /home/ubuntu/jenkins/workspace/React-App-Deployment/.git # timeout=10

> git config remote.origin.url <https://github.com/prasanth-wizard/devops-build.git> # timeout=10

Cloning repository <https://github.com/prasanth-wizard/devops-build.git>

> git init /home/ubuntu/jenkins/workspace/React-App-Deployment # timeout=10

Fetching upstream changes from <https://github.com/prasanth-wizard/devops-build.git>

> git --version # timeout=10

> git --version # 'git version 2.34.1'

using GIT\_ASKPASS to set credentials GitHub Access Token

> git fetch --tags --force --progress -- <https://github.com/prasanth-wizard/devops-build.git> +refs/heads/\*:refs/remotes/origin/\* # timeout=10

> git config remote.origin.url <https://github.com/prasanth-wizard/devops-build.git> # timeout=10

> git config --add remote.origin.fetch +refs/heads/\*:refs/remotes/origin/\* # timeout=10

Seen branch in repository origin/dev

Seen branch in repository origin/main

Seen 2 remote branches

> git show-ref --tags -d # timeout=10

> git config core.sparsecheckout # timeout=10

> git checkout -f 18cdf26a339a78d5f2fb474d4966bfa03af6dbd4 # timeout=10

Checking out Revision 18cdf26a339a78d5f2fb474d4966bfa03af6dbd4 (origin/main)

Commit message: "Merge branch 'dev'"

[Pipeline] script

[Pipeline] {

[Pipeline] sh

+ git rev-parse --short=7 HEAD

[Pipeline] echo

Branch: main

[Pipeline] echo

Image Tags: docker.io/prasanth0003/prod:18cdf26 docker.io/prasanth0003/prod:latest docker.io/prasanth0003/prod:main docker.io/prasanth0003/prod:63

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Build Docker Image)

[Pipeline] script

[Pipeline] {

Fetching upstream changes from <https://github.com/prasanth-wizard/devops-build.git>

> git --version # timeout=10

> git --version # 'git version 2.34.1'

using GIT\_ASKPASS to set credentials GitHub Access Token

> git fetch --tags --force --progress -- <https://github.com/prasanth-wizard/devops-build.git> +refs/heads/\*:refs/remotes/origin/\* # timeout=10

Seen branch in repository origin/dev

Seen branch in repository origin/main

Seen 2 remote branches

> git show-ref --tags -d # timeout=10

> git config core.sparsecheckout # timeout=10

> git checkout -f 18cdf26a339a78d5f2fb474d4966bfa03af6dbd4 # timeout=10

[Pipeline] sh

+ docker build -t docker.io/prasanth0003/prod:18cdf26 -t docker.io/prasanth0003/prod:latest -t docker.io/prasanth0003/prod:main -t docker.io/prasanth0003/prod:63 .

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 2.64MB

Step 1/6 : FROM nginx:alpine

---> 6769dc3a703c

Step 2/6 : WORKDIR /usr/share/nginx/html

---> Using cache

---> 7a981b94e632

Step 3/6 : RUN rm -rf ./\*

---> Using cache

---> 24d854bee91b

Step 4/6 : COPY build/ .

---> Using cache

---> ef3addf8967e

Step 5/6 : EXPOSE 80

---> Using cache

---> 4729f3fc2f6a

Step 6/6 : CMD ["nginx", "-g", "daemon off;"]

---> Using cache

---> 9cac7341a0dd

Successfully built 9cac7341a0dd

Successfully tagged prasanth0003/prod:18cdf26

Successfully tagged prasanth0003/prod:latest

Successfully tagged prasanth0003/prod:main

Successfully tagged prasanth0003/prod:63

+ docker images

+ grep prasanth0003

prasanth0003/prod 1.0.58 9cac7341a0dd About an hour ago 50.8MB

prasanth0003/prod 1.0.60 9cac7341a0dd About an hour ago 50.8MB

prasanth0003/prod 18cdf26 9cac7341a0dd About an hour ago 50.8MB

prasanth0003/prod 63 9cac7341a0dd About an hour ago 50.8MB

prasanth0003/prod latest 9cac7341a0dd About an hour ago 50.8MB

prasanth0003/prod main 9cac7341a0dd About an hour ago 50.8MB

prasanth0003/dev <none> 0d761b2d38d1 27 hours ago 50.8MB

[Pipeline] sh

+ docker inspect --type=image docker.io/prasanth0003/prod:18cdf26

[

{

"Id": "sha256:9cac7341a0ddf876542c4c8898fb02f887fdf4597758a075cd37982ec40dee2d",

"RepoTags": [

"prasanth0003/prod:1.0.58",

"prasanth0003/prod:1.0.60",

"prasanth0003/prod:18cdf26",

"prasanth0003/prod:63",

"prasanth0003/prod:latest",

"prasanth0003/prod:main"

],

"RepoDigests": [

"prasanth0003/prod@sha256:6fe52943a7b972cef9c2615d6d5a22220eb912ecfbe1ba77375f385a5da720a6"

],

"Parent": "sha256:4729f3fc2f6a950cf7f7dd9430e0159dbe05346d30f414174142d7c0019cd45e",

"Comment": "",

"Created": "2025-06-11T17:08:50.798025408Z",

"DockerVersion": "27.5.1",

"Author": "",

"Config": {

"Hostname": "",

"Domainname": "",

"User": "",

"AttachStdin": false,

"AttachStdout": false,

"AttachStderr": false,

"ExposedPorts": {

"80/tcp": {}

},

"Tty": false,

"OpenStdin": false,

"StdinOnce": false,

"Env": [

"PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin",

"NGINX\_VERSION=1.27.5",

"PKG\_RELEASE=1",

"DYNPKG\_RELEASE=1",

"NJS\_VERSION=0.8.10",

"NJS\_RELEASE=1"

],

"Cmd": [

"nginx",

"-g",

"daemon off;"

],

"Image": "sha256:4729f3fc2f6a950cf7f7dd9430e0159dbe05346d30f414174142d7c0019cd45e",

"Volumes": null,

"WorkingDir": "/usr/share/nginx/html",

"Entrypoint": [

"/docker-entrypoint.sh"

],

"OnBuild": null,

"Labels": {

"maintainer": "NGINX Docker Maintainers <docker-maint@nginx.com>"

},

"StopSignal": "SIGQUIT"

},

"Architecture": "amd64",

"Os": "linux",

"Size": 50842926,

"GraphDriver": {

"Data": {

"LowerDir": "/var/lib/docker/overlay2/b7384d2a13d0d8dd222103044ef3e0ecb18e20e7f49360ac4e1bad206c0696b9/diff:/var/lib/docker/overlay2/2e53d516ba6efe288ee1c3e1910c0ec144704e03bc624c5334a978ab64fa787c/diff:/var/lib/docker/overlay2/063be06a73b0e59d3e23d3fc604d70c6efa113c40d645d32e074d78544dbfc70/diff:/var/lib/docker/overlay2/bfe6a78617ea0c01fd83530f7a63ca2da6588277bfe05f3ea9b27afa8708c3bc/diff:/var/lib/docker/overlay2/0526f6e2120601cee541b71f55573e0bd52e01d8a0ce3828da59cdd45dc38b87/diff:/var/lib/docker/overlay2/e99e398efb3034da28bcb2fb58147808859cafafa01919e78746f60d99e5ecd7/diff:/var/lib/docker/overlay2/1e2d5648db4b7db345be0a075d7ec3dcf9b16400a06343220bd256408f138b2f/diff:/var/lib/docker/overlay2/66a68b332a6872a0907ce4d714372c176673b96e6357aca642f5cd2225e9d080/diff:/var/lib/docker/overlay2/5ace7f0c811f4ccea9b85c7ecf0118cd6d2d209e76bf98ad607589e34ad2c2eb/diff",

"MergedDir": "/var/lib/docker/overlay2/070b7b4559afdfd2d317acf9ca7531427b88a9c9c1fc4314276999846a474146/merged",

"UpperDir": "/var/lib/docker/overlay2/070b7b4559afdfd2d317acf9ca7531427b88a9c9c1fc4314276999846a474146/diff",

"WorkDir": "/var/lib/docker/overlay2/070b7b4559afdfd2d317acf9ca7531427b88a9c9c1fc4314276999846a474146/work"

},

"Name": "overlay2"

},

"RootFS": {

"Type": "layers",

"Layers": [

"sha256:08000c18d16dadf9553d747a58cf44023423a9ab010aab96cf263d2216b8b350",

"sha256:d71eae0084c1aa823dd8fb2ecf8604d5c0f4911226c042bb1f8297e819f4b192",

"sha256:c56f134d380585340a68d0db2f2c170641a1c0ff72ccf2438cf2f693df756a85",

"sha256:e244aa659f612a80c40dd8645812301e3def6b15ec67b9e486ed2201172b51d1",

"sha256:b8d7d1d2263425d6044e059b2810017d062d659b9b755241f3747eda77726250",

"sha256:811a4dbbf4a5309e4390cf655c12db92e1a4304fb9d9731f83e7b02e95a617c6",

"sha256:947e805a4ac71f68e6703550c0b36c2aa2e554c4fa670ca2da6a25c6d7dccb66",

"sha256:0d853d50b128aa460b47e7121849463a14b18d4fd976caf5014744aae24d28aa",

"sha256:0741406cbd2ef066f67447fc167b14bb4451821de55541197fe2fa6a11d50c2c",

"sha256:320d0d9d1060d5ea2a301d007c3f7b3646a109583286a53031dee51ffecbf5a2"

]

},

"Metadata": {

"LastTagTime": "2025-06-11T18:09:50.934996153Z"

}

}

]

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Push Docker Image)

[Pipeline] withCredentials

Masking supported pattern matches of $DOCKER\_PASS

[Pipeline] {

[Pipeline] script

[Pipeline] {

[Pipeline] sh

Warning: A secret was passed to "sh" using Groovy String interpolation, which is insecure.

Affected argument(s) used the following variable(s): [DOCKER\_PASS]

See <https://jenkins.io/redirect/groovy-string-interpolation> for details.

+ echo \*\*\*\*

+ docker login -u prasanth0003 --password-stdin

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/#credential-stores>

Login Succeeded

[Pipeline] retry

[Pipeline] {

[Pipeline] sh

+ docker push docker.io/prasanth0003/prod:18cdf26

The push refers to repository [docker.io/prasanth0003/prod]

320d0d9d1060: Preparing

0741406cbd2e: Preparing

0d853d50b128: Preparing

947e805a4ac7: Preparing

811a4dbbf4a5: Preparing

b8d7d1d22634: Preparing

e244aa659f61: Preparing

c56f134d3805: Preparing

d71eae0084c1: Preparing

08000c18d16d: Preparing

e244aa659f61: Waiting

c56f134d3805: Waiting

d71eae0084c1: Waiting

08000c18d16d: Waiting

b8d7d1d22634: Waiting

0d853d50b128: Mounted from prasanth0003/dev

947e805a4ac7: Mounted from prasanth0003/dev

320d0d9d1060: Mounted from prasanth0003/dev

0741406cbd2e: Mounted from prasanth0003/dev

811a4dbbf4a5: Mounted from prasanth0003/dev

b8d7d1d22634: Mounted from prasanth0003/dev

c56f134d3805: Mounted from prasanth0003/dev

e244aa659f61: Mounted from prasanth0003/dev

d71eae0084c1: Mounted from prasanth0003/dev

08000c18d16d: Mounted from prasanth0003/dev

18cdf26: digest: sha256:6fe52943a7b972cef9c2615d6d5a22220eb912ecfbe1ba77375f385a5da720a6 size: 2406

[Pipeline] }

[Pipeline] // retry

[Pipeline] retry

[Pipeline] {

[Pipeline] sh

+ docker push docker.io/prasanth0003/prod:latest

The push refers to repository [docker.io/prasanth0003/prod]

320d0d9d1060: Preparing

0741406cbd2e: Preparing

0d853d50b128: Preparing

947e805a4ac7: Preparing

811a4dbbf4a5: Preparing

b8d7d1d22634: Preparing

e244aa659f61: Preparing

c56f134d3805: Preparing

d71eae0084c1: Preparing

08000c18d16d: Preparing

b8d7d1d22634: Waiting

e244aa659f61: Waiting

c56f134d3805: Waiting

d71eae0084c1: Waiting

08000c18d16d: Waiting

947e805a4ac7: Layer already exists

811a4dbbf4a5: Layer already exists

0741406cbd2e: Layer already exists

320d0d9d1060: Layer already exists

0d853d50b128: Layer already exists

b8d7d1d22634: Layer already exists

e244aa659f61: Layer already exists

c56f134d3805: Layer already exists

d71eae0084c1: Layer already exists

08000c18d16d: Layer already exists

latest: digest: sha256:6fe52943a7b972cef9c2615d6d5a22220eb912ecfbe1ba77375f385a5da720a6 size: 2406

[Pipeline] }

[Pipeline] // retry

[Pipeline] retry

[Pipeline] {

[Pipeline] sh

+ docker push docker.io/prasanth0003/prod:main

The push refers to repository [docker.io/prasanth0003/prod]

320d0d9d1060: Preparing

0741406cbd2e: Preparing

0d853d50b128: Preparing

947e805a4ac7: Preparing

811a4dbbf4a5: Preparing

b8d7d1d22634: Preparing

e244aa659f61: Preparing

c56f134d3805: Preparing

d71eae0084c1: Preparing

08000c18d16d: Preparing

e244aa659f61: Waiting

c56f134d3805: Waiting

d71eae0084c1: Waiting

08000c18d16d: Waiting

b8d7d1d22634: Waiting

0d853d50b128: Layer already exists

811a4dbbf4a5: Layer already exists

320d0d9d1060: Layer already exists

947e805a4ac7: Layer already exists

0741406cbd2e: Layer already exists

e244aa659f61: Layer already exists

d71eae0084c1: Layer already exists

08000c18d16d: Layer already exists

b8d7d1d22634: Layer already exists

c56f134d3805: Layer already exists

main: digest: sha256:6fe52943a7b972cef9c2615d6d5a22220eb912ecfbe1ba77375f385a5da720a6 size: 2406

[Pipeline] }

[Pipeline] // retry

[Pipeline] retry

[Pipeline] {

[Pipeline] sh

+ docker push docker.io/prasanth0003/prod:63

The push refers to repository [docker.io/prasanth0003/prod]

320d0d9d1060: Preparing

0741406cbd2e: Preparing

0d853d50b128: Preparing

947e805a4ac7: Preparing

811a4dbbf4a5: Preparing

b8d7d1d22634: Preparing

e244aa659f61: Preparing

c56f134d3805: Preparing

d71eae0084c1: Preparing

08000c18d16d: Preparing

b8d7d1d22634: Waiting

e244aa659f61: Waiting

c56f134d3805: Waiting

d71eae0084c1: Waiting

08000c18d16d: Waiting

0d853d50b128: Layer already exists

811a4dbbf4a5: Layer already exists

947e805a4ac7: Layer already exists

320d0d9d1060: Layer already exists

0741406cbd2e: Layer already exists

b8d7d1d22634: Layer already exists

c56f134d3805: Layer already exists

d71eae0084c1: Layer already exists

e244aa659f61: Layer already exists

08000c18d16d: Layer already exists

63: digest: sha256:6fe52943a7b972cef9c2615d6d5a22220eb912ecfbe1ba77375f385a5da720a6 size: 2406

[Pipeline] }

[Pipeline] // retry

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // withCredentials

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Deploy to Agent-1)

[Pipeline] script

[Pipeline] {

[Pipeline] withCredentials

Masking supported pattern matches of $SSH\_KEY

[Pipeline] {

[Pipeline] sh

Warning: A secret was passed to "sh" using Groovy String interpolation, which is insecure.

Affected argument(s) used the following variable(s): [SSH\_KEY]

See <https://jenkins.io/redirect/groovy-string-interpolation> for details.

+ ssh -o StrictHostKeyChecking=no -i \*\*\*\* ubuntu@51.20.2.247

docker pull docker.io/prasanth0003/prod:18cdf26

docker stop react-app || true

docker rm react-app || true

docker run -d --name react-app -p 80:80 docker.io/prasanth0003/prod:18cdf26

18cdf26: Pulling from prasanth0003/prod

Digest: sha256:6fe52943a7b972cef9c2615d6d5a22220eb912ecfbe1ba77375f385a5da720a6

Status: Image is up to date for prasanth0003/prod:18cdf26

docker.io/prasanth0003/prod:18cdf26

react-app

react-app

416cc29d68d57f5c2dfcf681898a935722d9c7dcb851828839c8cb8f1fe51755

[Pipeline] echo

🚀 Successfully deployed docker.io/prasanth0003/prod:18cdf26 to 51.20.2.247

[Pipeline] }

[Pipeline] // withCredentials

[Pipeline] }

[Pipeline] // script

[Pipeline] }

[Pipeline] // stage

[Pipeline] stage

[Pipeline] { (Declarative: Post Actions)

[Pipeline] script

[Pipeline] {

[Pipeline] sh

+ docker logout

Removing login credentials for <https://index.docker.io/v1/>

[Pipeline] sh

+ docker rmi docker.io/prasanth0003/prod:18cdf26

Untagged: prasanth0003/prod:18cdf26

[Pipeline] sh

+ docker rmi docker.io/prasanth0003/prod:latest

Untagged: prasanth0003/prod:latest

[Pipeline] sh

+ docker rmi docker.io/prasanth0003/prod:main

Untagged: prasanth0003/prod:main

[Pipeline] sh

+ docker rmi docker.io/prasanth0003/prod:63

Untagged: prasanth0003/prod:63

[Pipeline] cleanWs

[WS-CLEANUP] Deleting project workspace...

[WS-CLEANUP] Deferred wipeout is used...

[WS-CLEANUP] done

[Pipeline] }

[Pipeline] // script

[Pipeline] echo

Successfully built and pushed images with tags: docker.io/prasanth0003/prod:18cdf26 docker.io/prasanth0003/prod:latest docker.io/prasanth0003/prod:main docker.io/prasanth0003/prod:63

[Pipeline] }

[Pipeline] // stage

[Pipeline] }

[Pipeline] // withEnv

[Pipeline] }

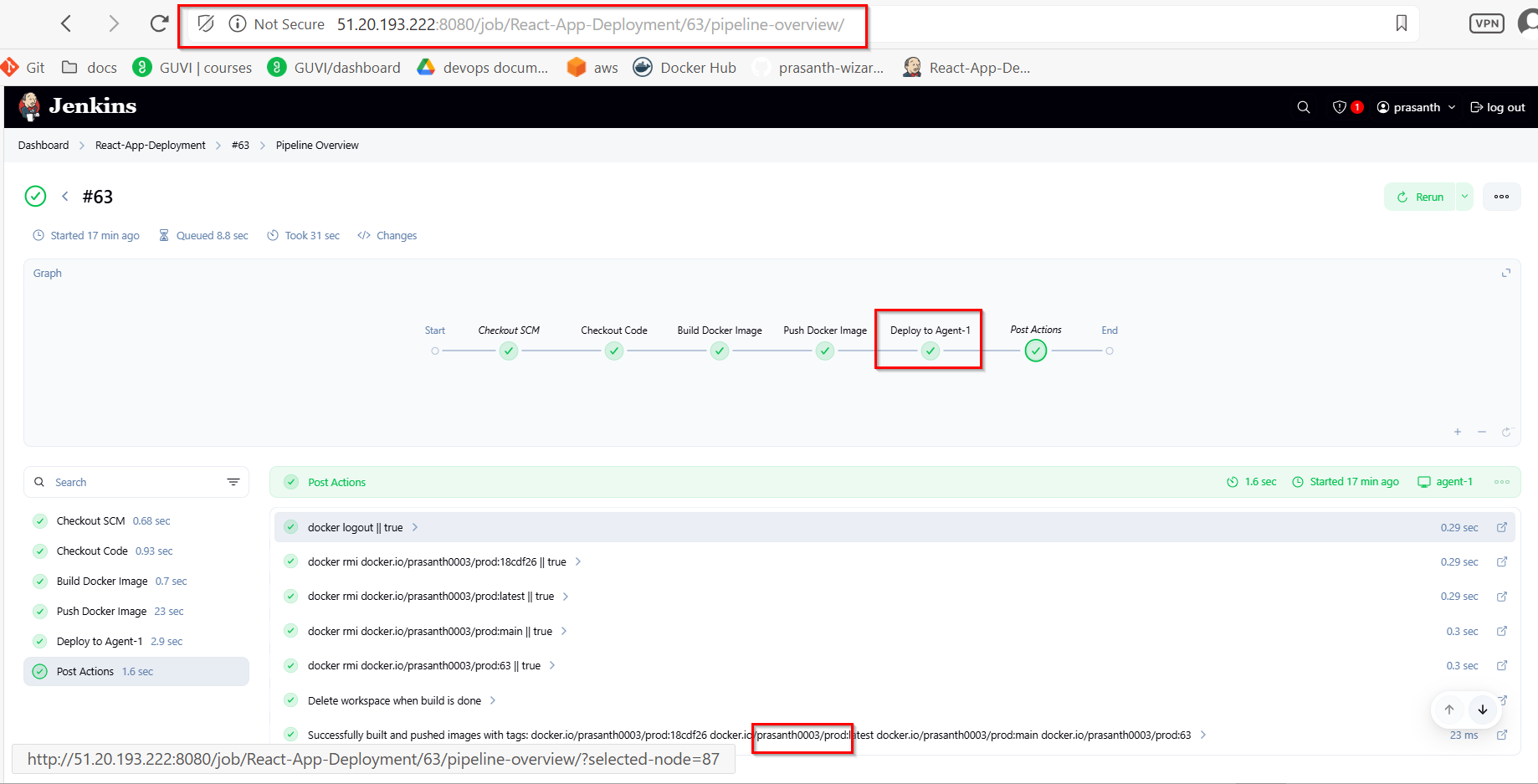
[Pipeline] // withEnv

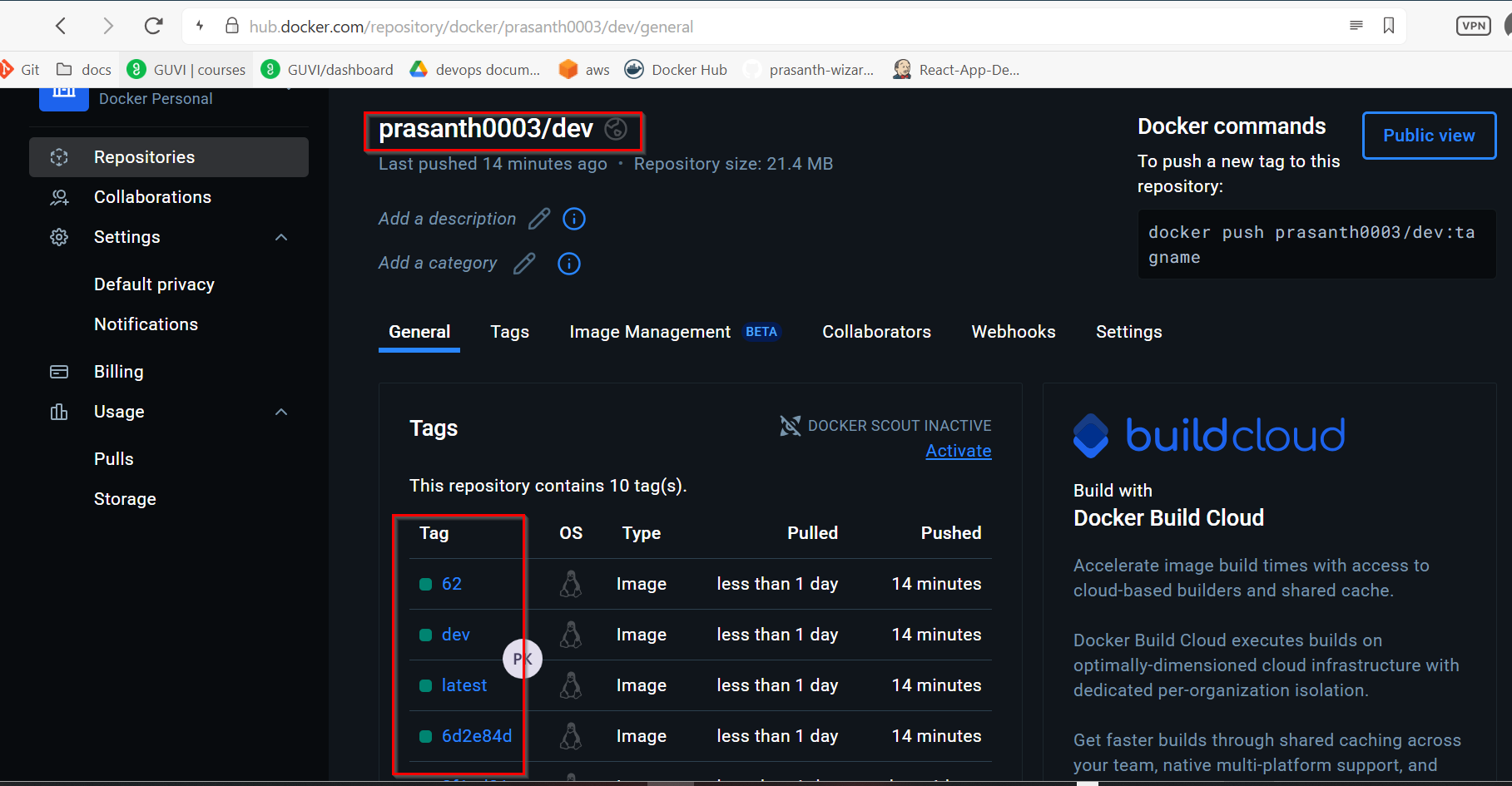
[Pipeline] }

[Pipeline] // node

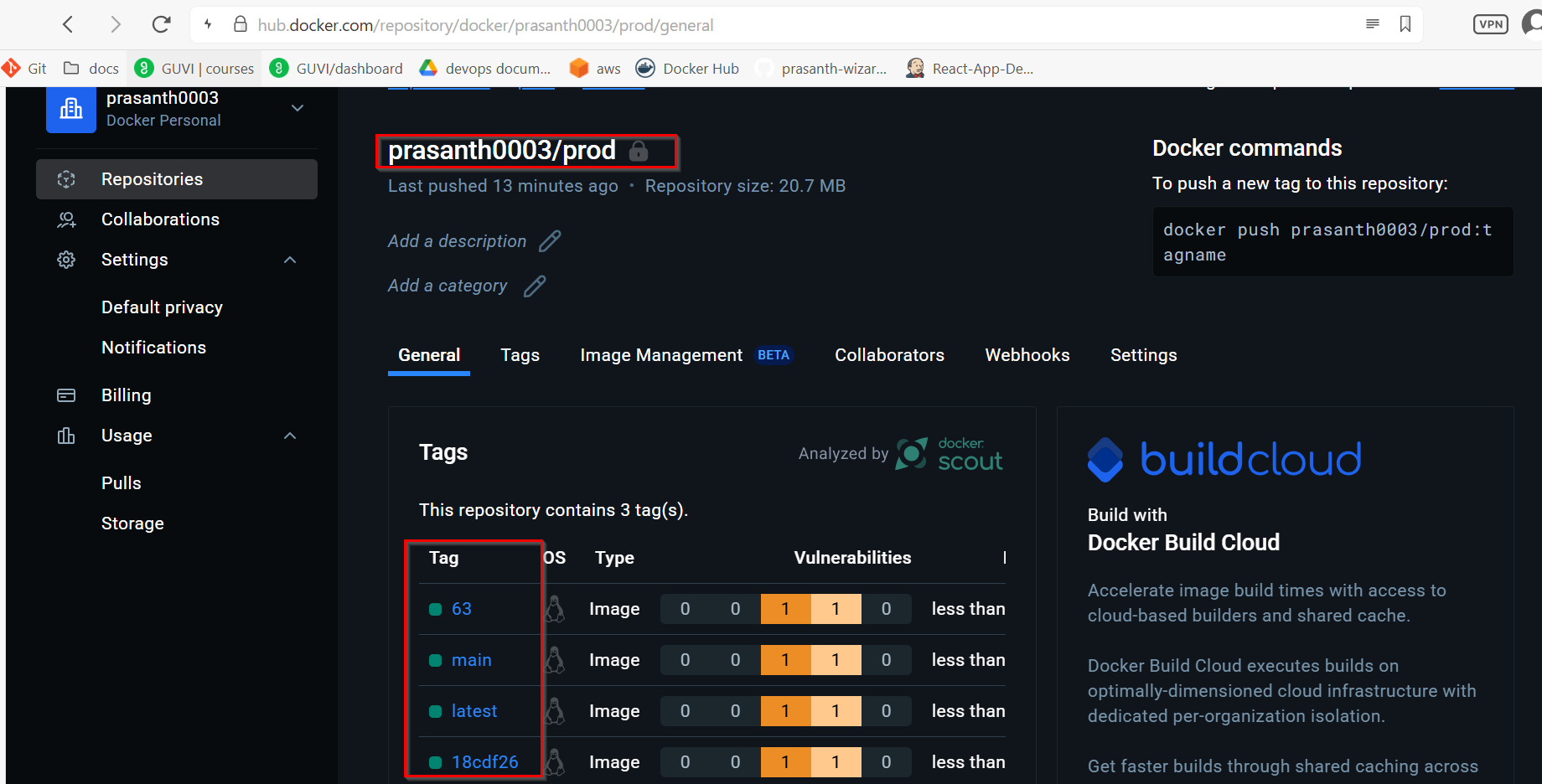
[Pipeline] End of Pipeline

Finished: SUCCESS

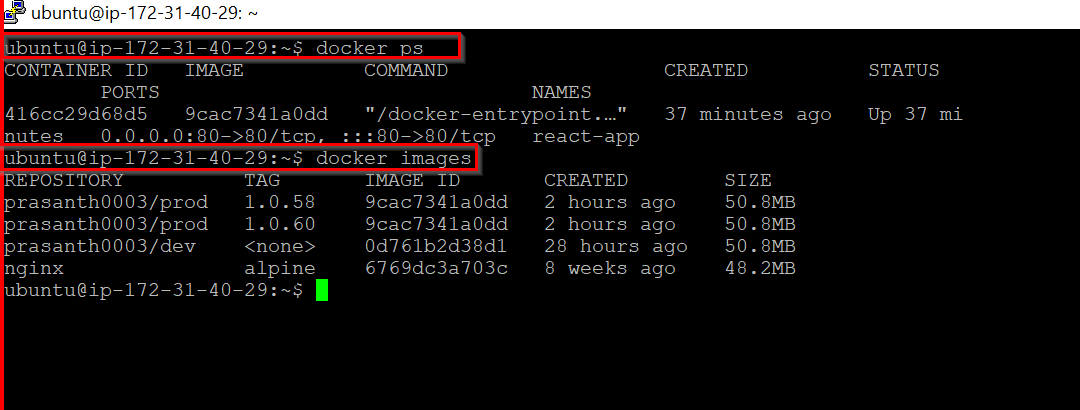
****

**Dockerhub “dev” public:**

**Dockerhub “prod” private:**

****

**Agent\_1 Container Status:**

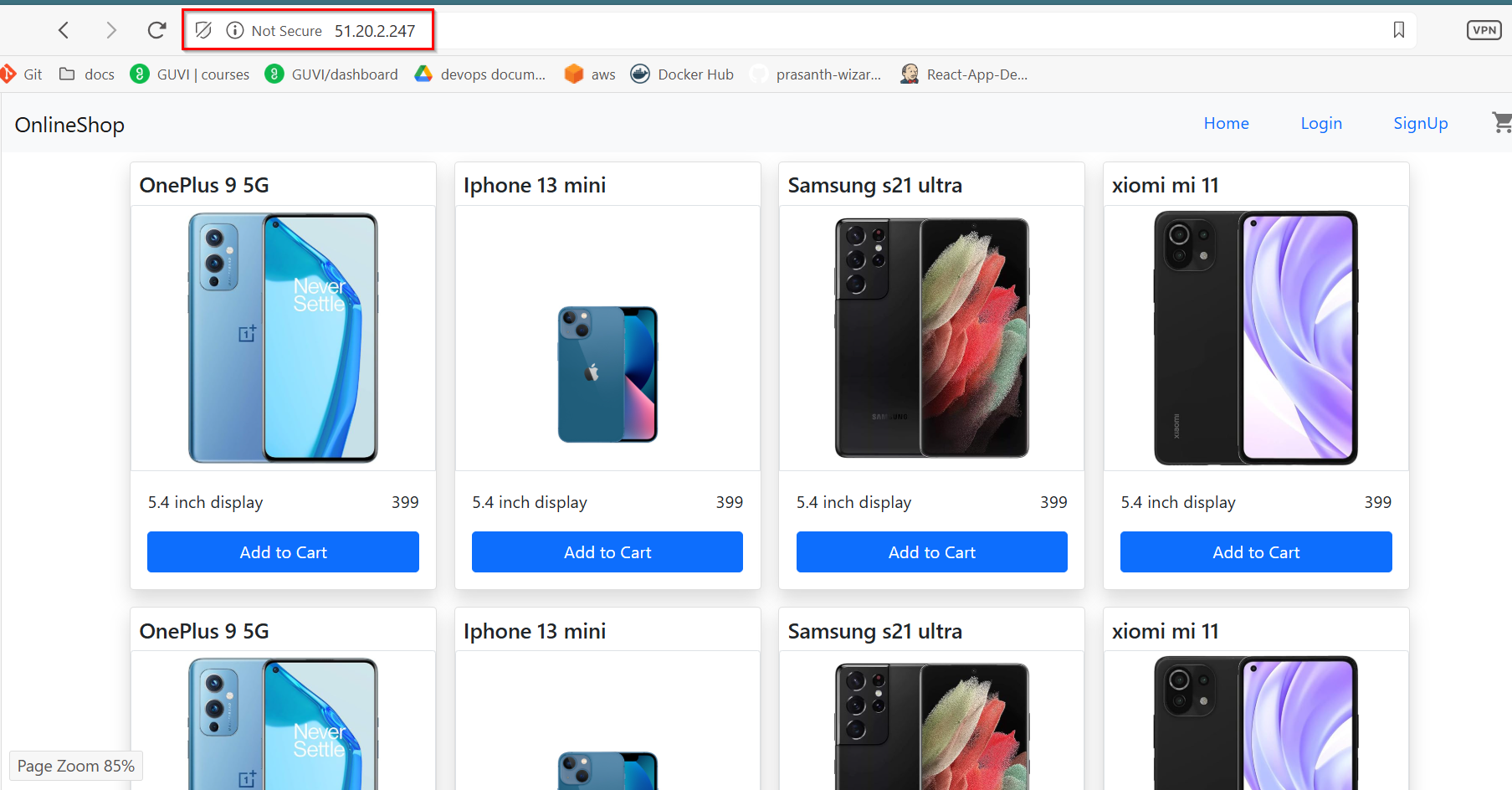
****

**React application running in Jenkins server IP and worker node IP:**

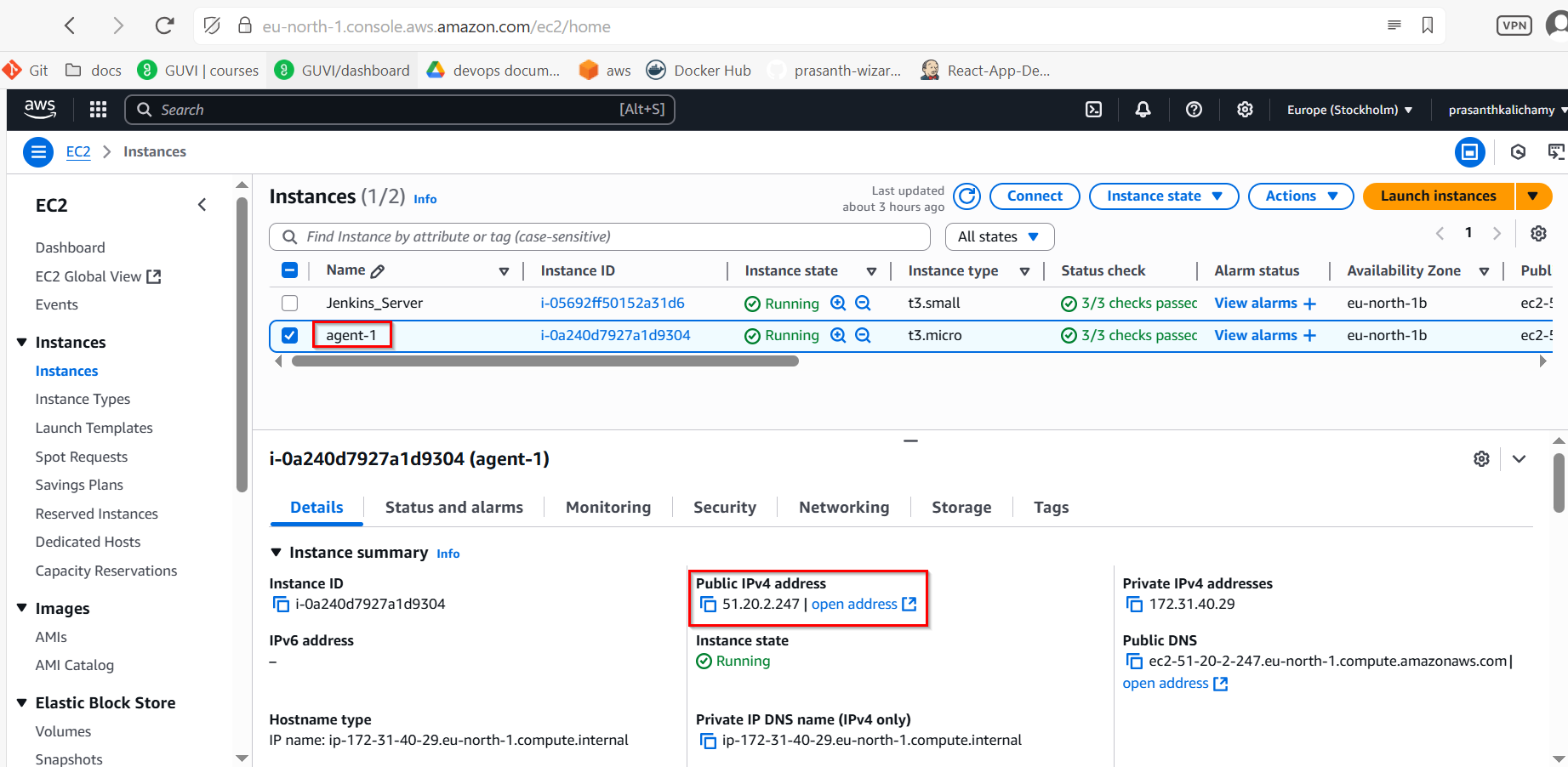
**Jenkins server:**

****

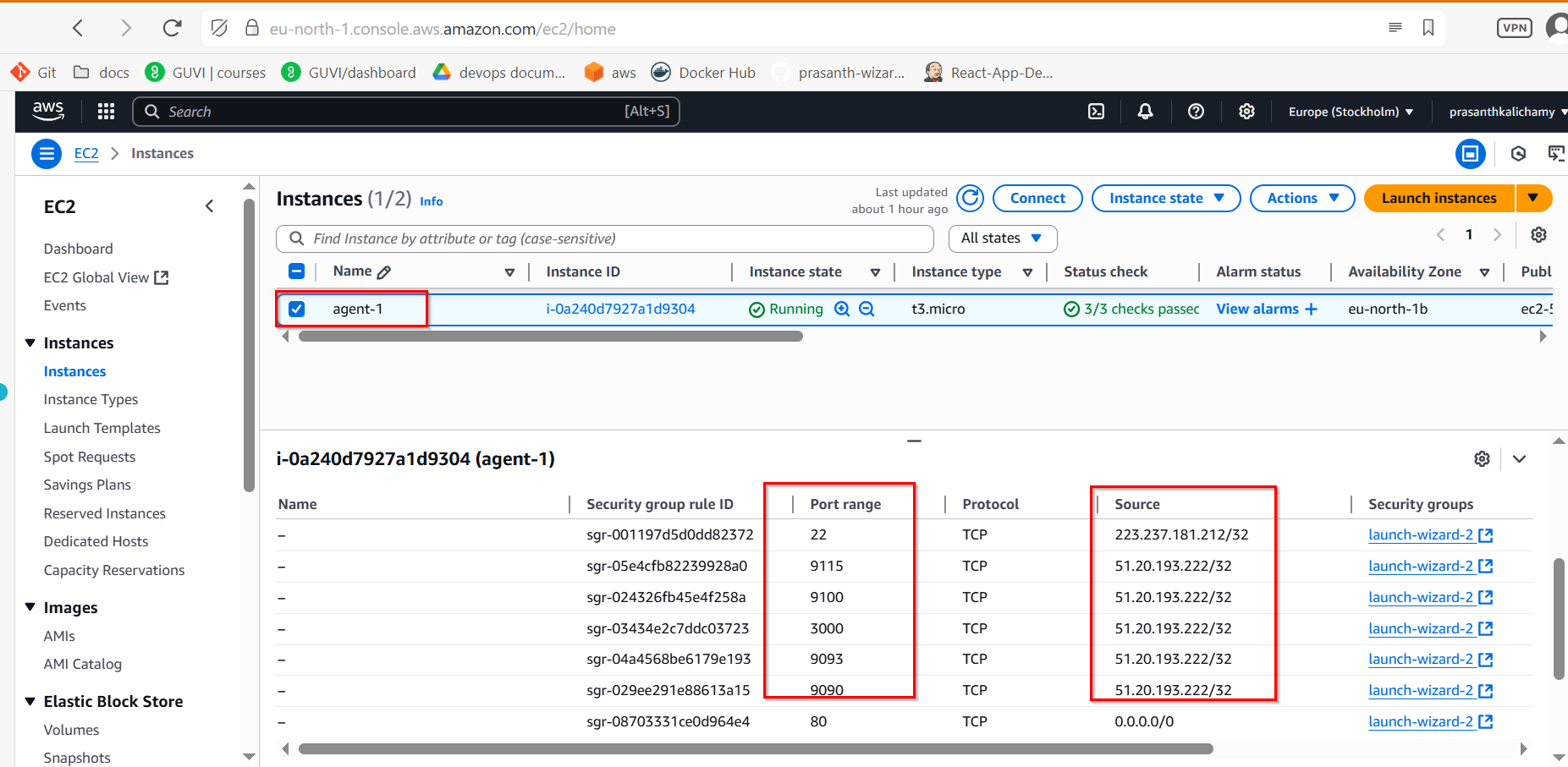
**Agent-1 IP:**

****

**AGENT-1:**



**Agent-1 Security\_Group Inbound rules:**

****

**Docker Install**

sudo apt update

sudo apt install apt-transport-https ca-certificates curl gnupg-agent software-properties-common -y

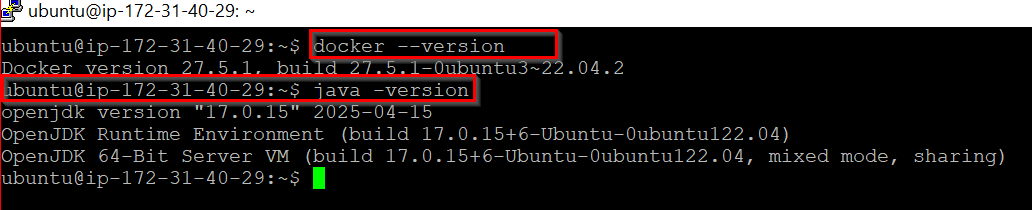
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add –

sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable"

sudo apt update

sudo apt install docker-ce docker-ce-cli containerd.io -y

sudo usermod -aG docker $USER



**MONITORING**

├── alertmanager

│   └── alertmanager.yml

├── blackbox

│   └── blackbox.yml

├── docker-compose.yml

└── prometheus

├── alert\_rules.yml

└── prometheus.yml

**docker-compose.yml**

version: '3.8'

services:

react-app:

image: prasanth0003/dev:latest

container\_name: react-app

ports:

- "80:80"

prometheus:

image: prom/prometheus

container\_name: prometheus

volumes:

- ./prometheus/prometheus.yml:/etc/prometheus/prometheus.yml

- ./prometheus/alert\_rules.yml:/etc/prometheus/alert\_rules.yml

command:

- '--config.file=/etc/prometheus/prometheus.yml'

ports:

- "9090:9090"

alertmanager:

image: prom/alertmanager

container\_name: alertmanager

volumes:

- ./alertmanager/alertmanager.yml:/etc/alertmanager/alertmanager.yml

command:

- '--config.file=/etc/alertmanager/alertmanager.yml'

ports:

- "9093:9093"

grafana:

image: grafana/grafana

container\_name: grafana

ports:

- "3000:3000"

depends\_on:

- prometheus

node-exporter:

image: prom/node-exporter

container\_name: node-exporter

ports:

- "9100:9100"

volumes:

- /proc:/host/proc:ro

- /sys:/host/sys:ro

- /:/rootfs:ro

command:

- '--path.procfs=/host/proc'

- '--path.sysfs=/host/sys'

- '--path.rootfs=/rootfs'

blackbox:

image: prom/blackbox-exporter

container\_name: blackbox

ports:

- "9115:9115"

volumes:

- ./blackbox:/config

command:

- '--config.file=/config/blackbox.yml'

**prometheus.yml**

global:

scrape\_interval: 15s

evaluation\_interval: 15s

alerting:

alertmanagers:

- static\_configs:

- targets: ['alertmanager:9093']

rule\_files:

- alert\_rules.yml

scrape\_configs:

- job\_name: 'prometheus'

static\_configs:

- targets: ['localhost:9090']

- job\_name: 'node\_exporter'

static\_configs:

- targets: ['node-exporter:9100']

- job\_name: 'react\_http\_probe'

metrics\_path: /probe

params:

module: [http\_2xx]

static\_configs:

- targets:

- http://51.20.2.247

relabel\_configs:

- source\_labels: [\_\_address\_\_]

target\_label: \_\_param\_target

- source\_labels: [\_\_param\_target]

target\_label: instance

- target\_label: \_\_address\_\_

replacement: blackbox:9115

**alert\_rules.yml**

groups:

- name: react\_app\_alerts

rules:

- alert: ReactAppContainerDown

expr: up{job="react\_app"} == 0

for: 15s

labels:

severity: critical

annotations:

summary: "React container unreachable"

description: "Prometheus can't connect to react-app on port 80."

- alert: ReactHTTPProbeFailed

expr: probe\_success{job="react\_http\_probe"} == 0

for: 15s

labels:

severity: critical

annotations:

summary: "React HTTP health check failed"

description: "React app did not return HTTP 200."

**alertmanager.yml**

global:

smtp\_smarthost: 'smtp.gmail.com:587'

smtp\_from: 'prasanthkalichamy30@gmail.com'

smtp\_auth\_username: 'prasanthkalichamy30@gmail.com'

smtp\_auth\_password: 'aqsa nfqv kkhz rloa'

smtp\_require\_tls: true

route:

receiver: 'email-alert'

receivers:

- name: 'email-alert'

email\_configs:

- to: 'prasanthkalichamy30@gmail.com'

send\_resolved: true

**blackbox.yml**

modules:

http\_2xx:

prober: http

timeout: 5s

http:

valid\_http\_versions: ["HTTP/1.1", "HTTP/2"]

valid\_status\_codes: [200]

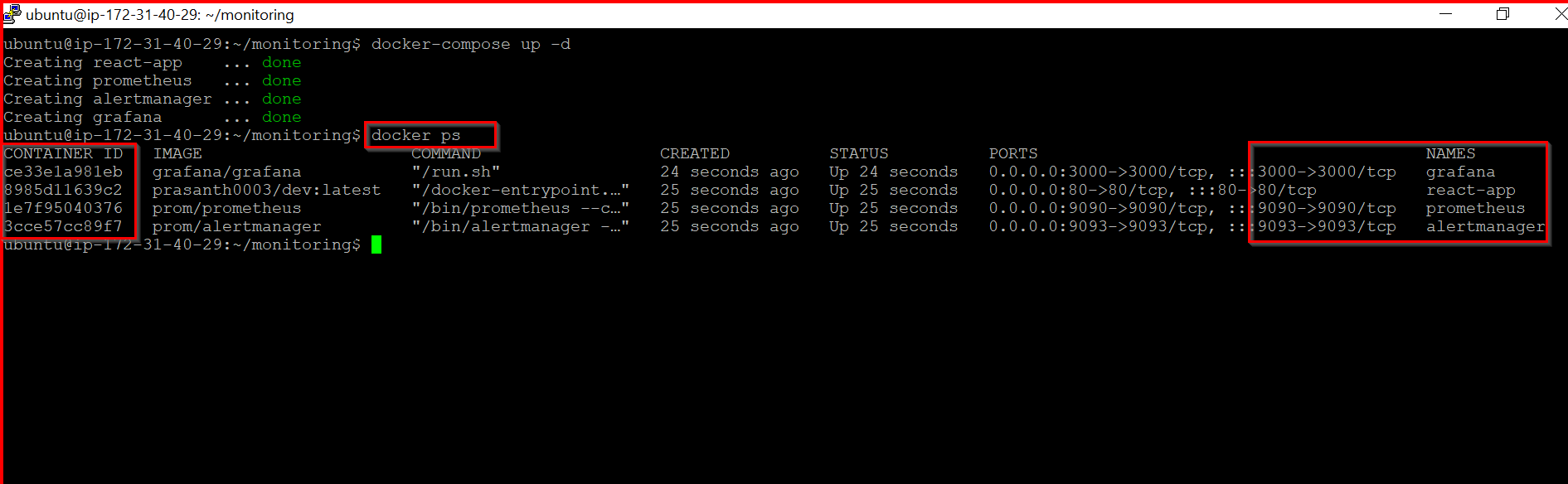
method: GET

**Run Everything**

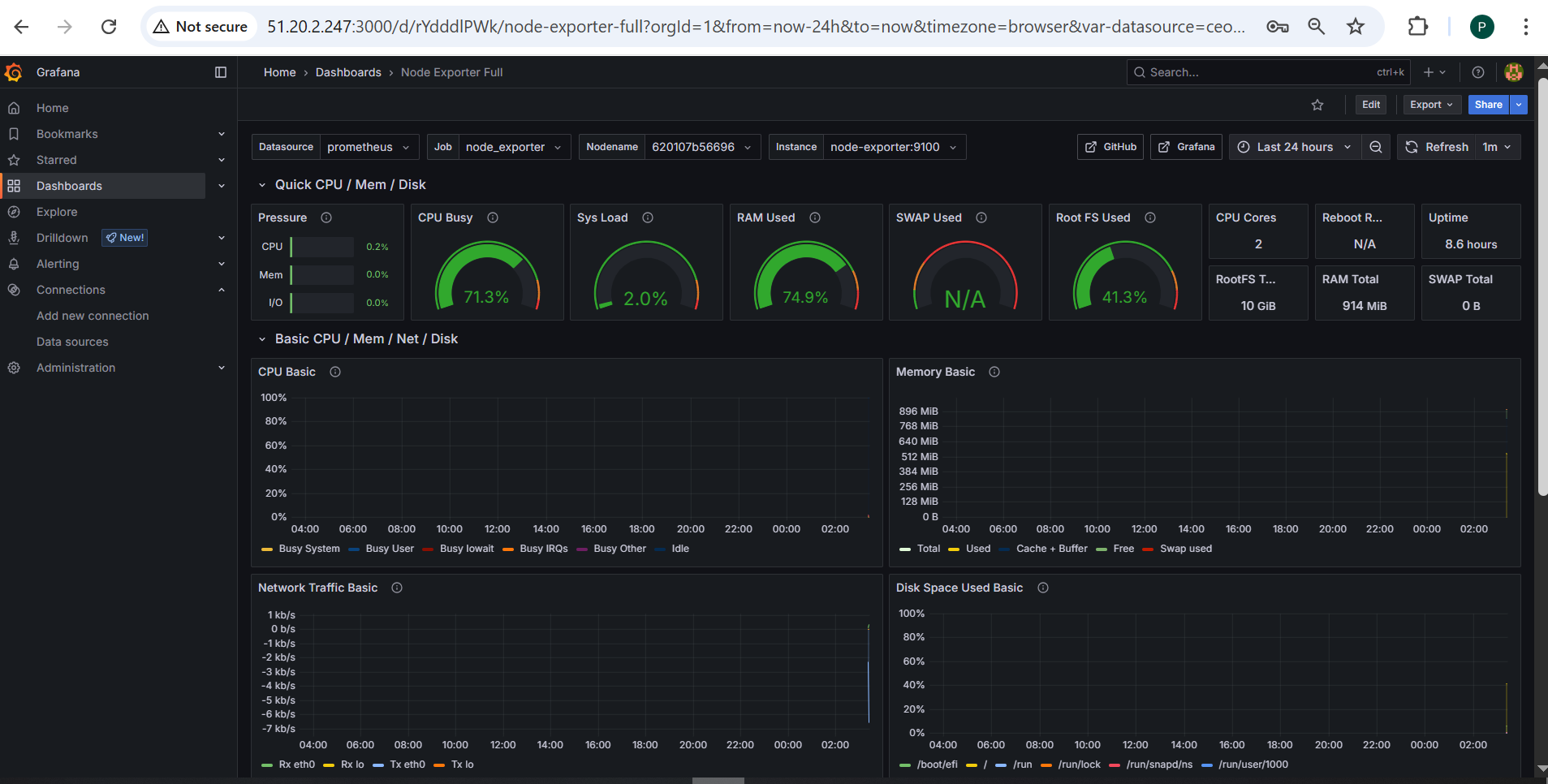
cd devops-monitoring/

docker-compose up -d

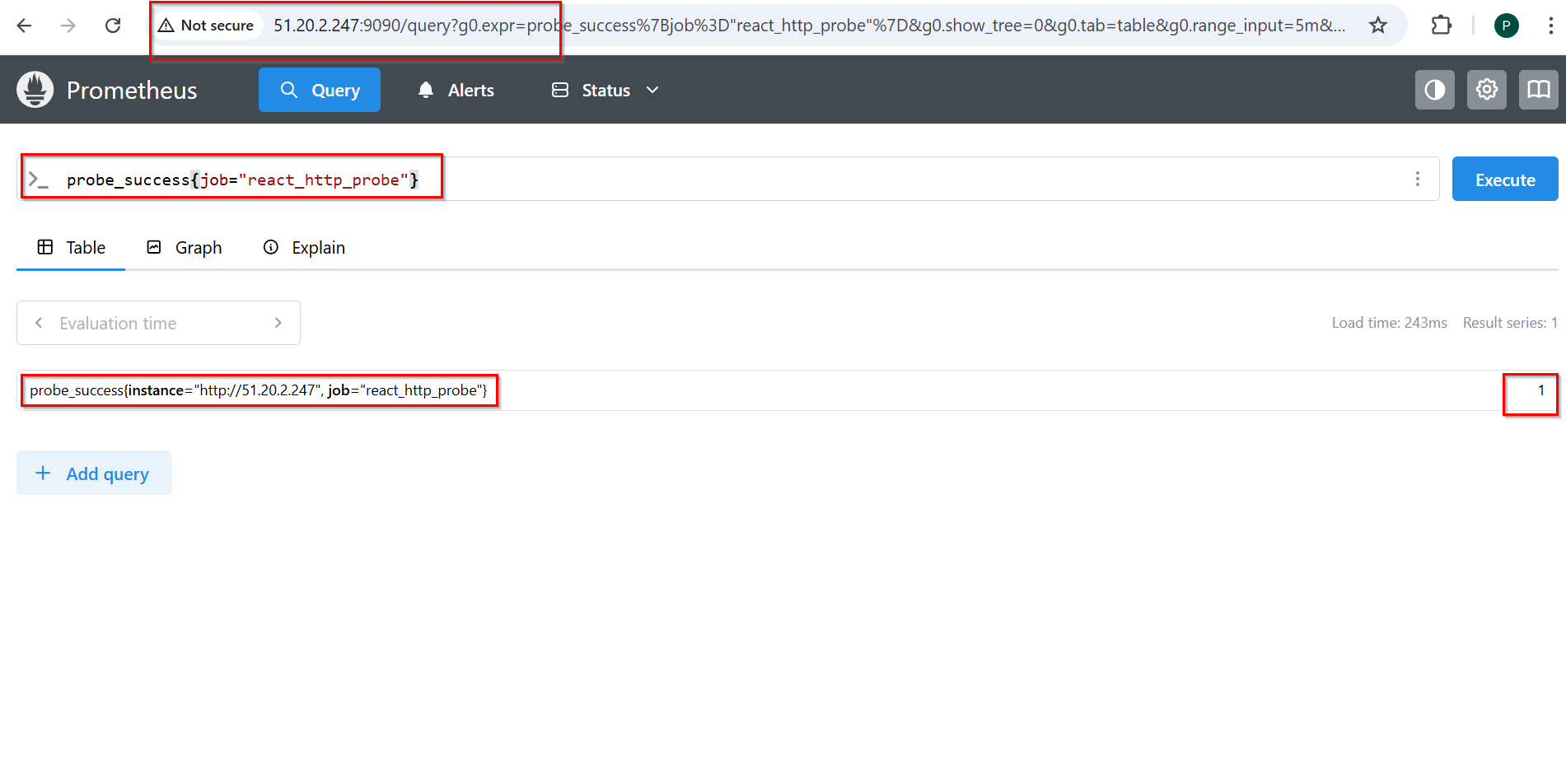
**Docker Containers:**

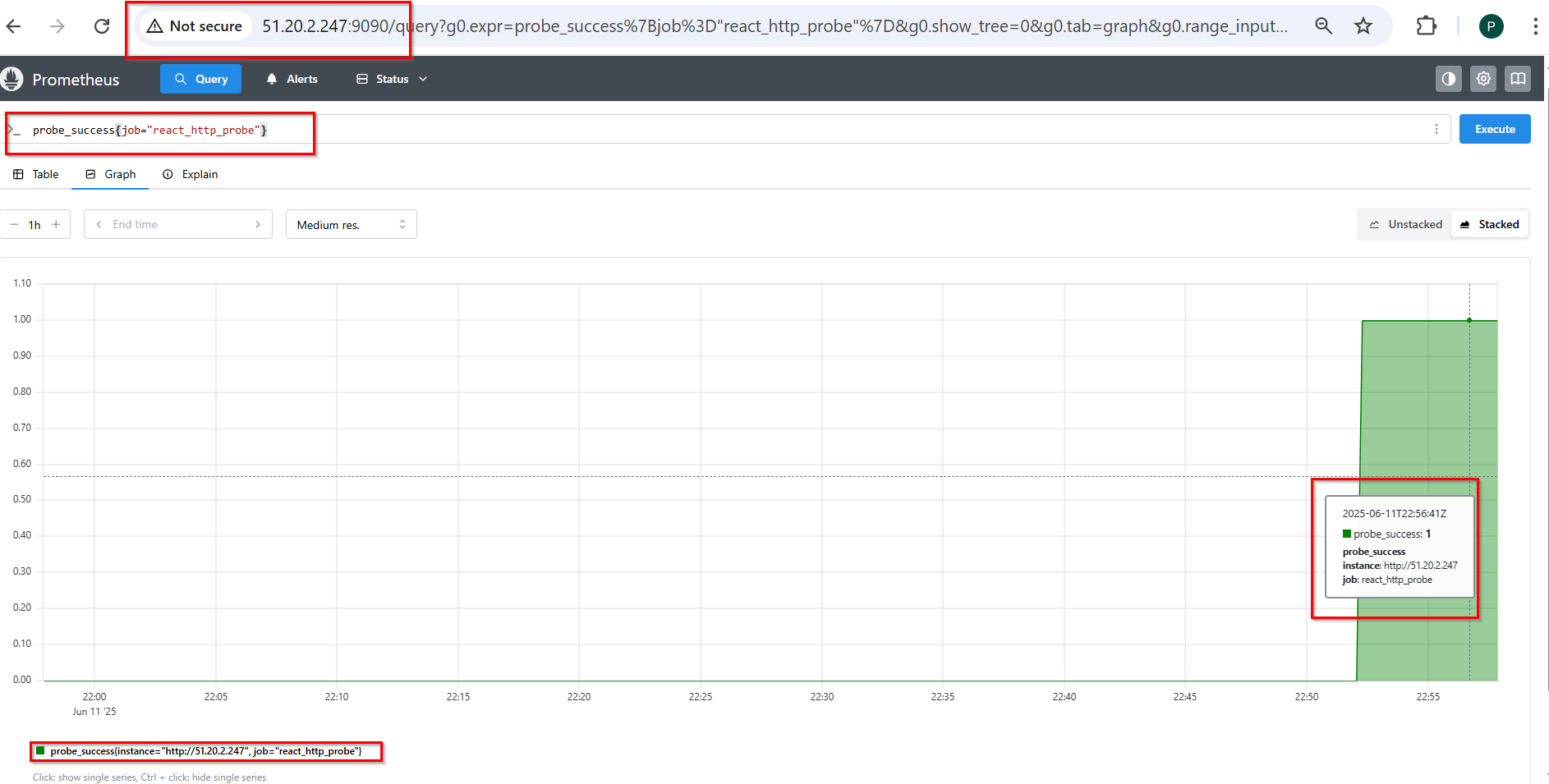


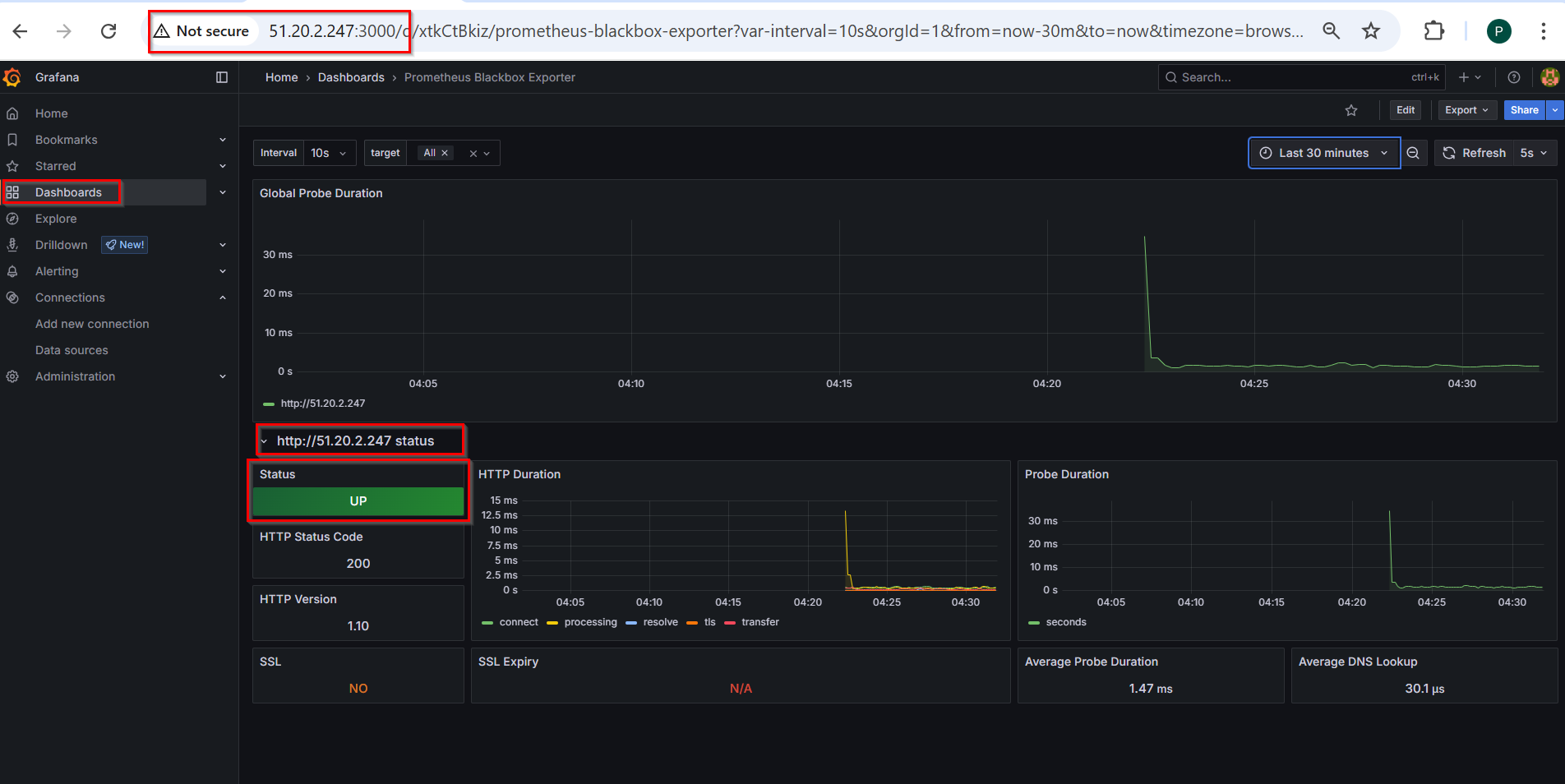
**All\_Metric\_Agent-1:**



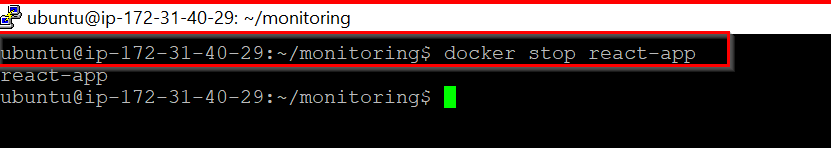
**React Applicaiotn Health-check:**



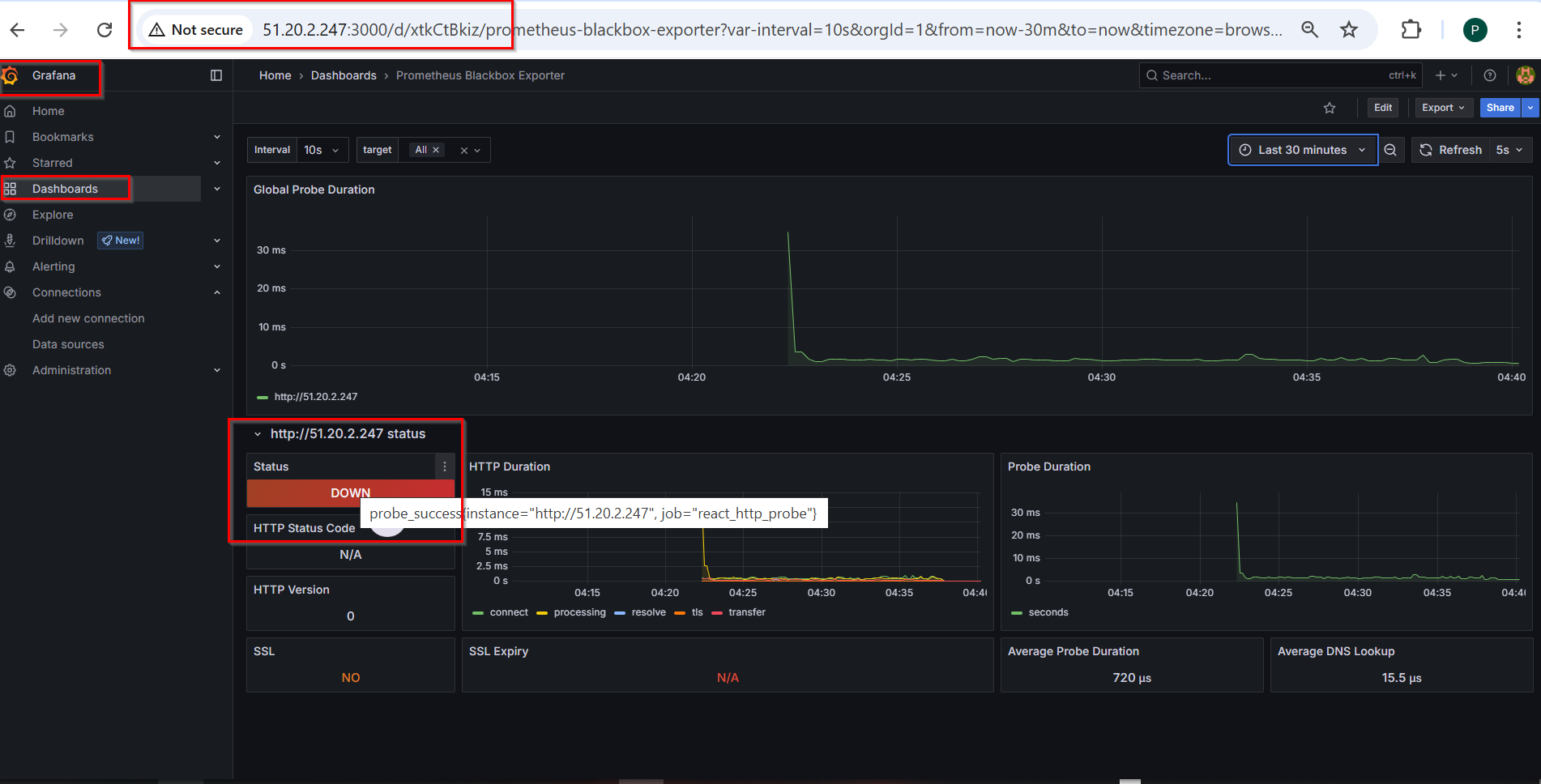




**Stopping the React container:**



**React App is Down:**



**Email Notification:**

