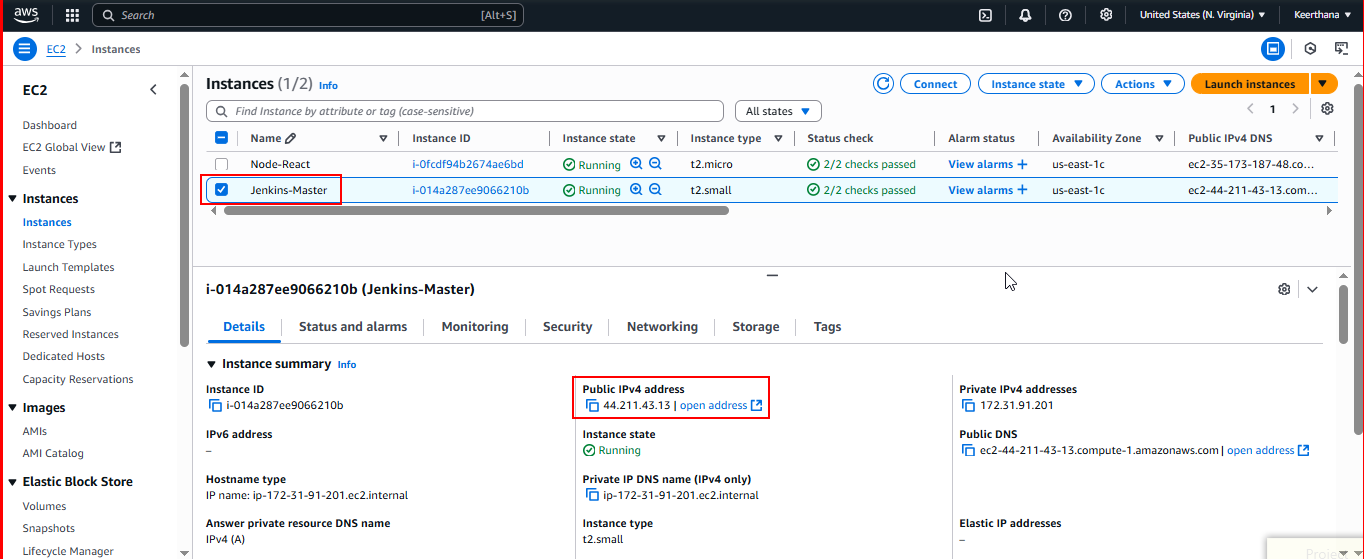
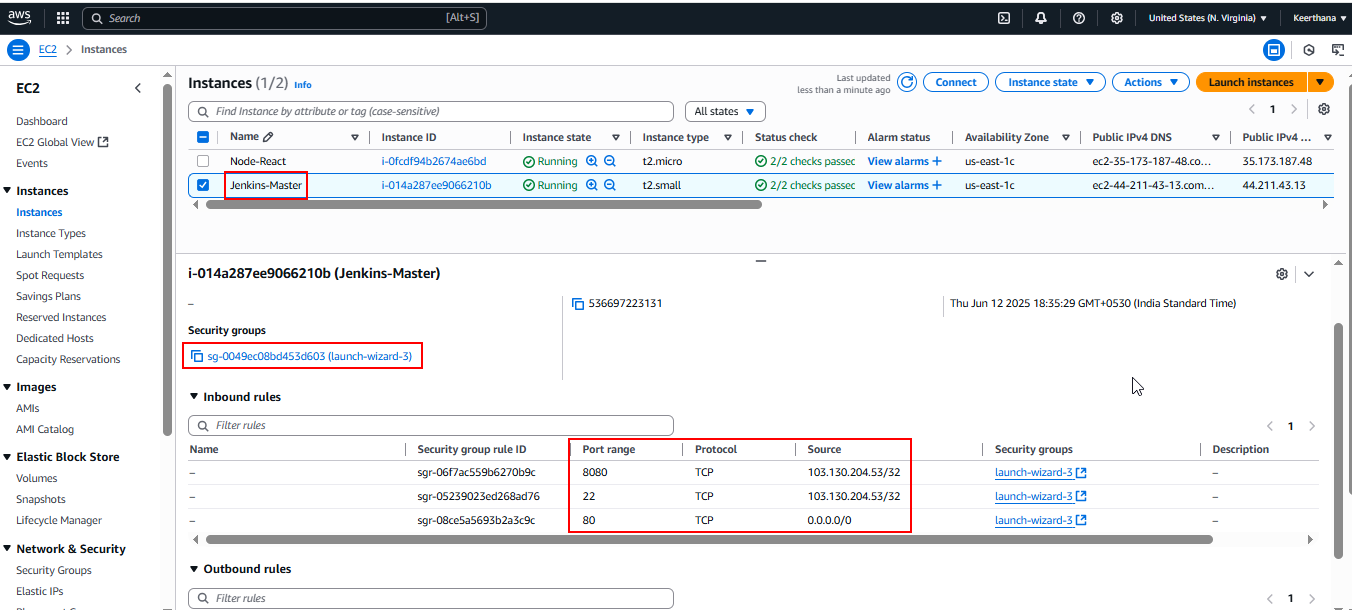
**React E-commerce Application Deployment**

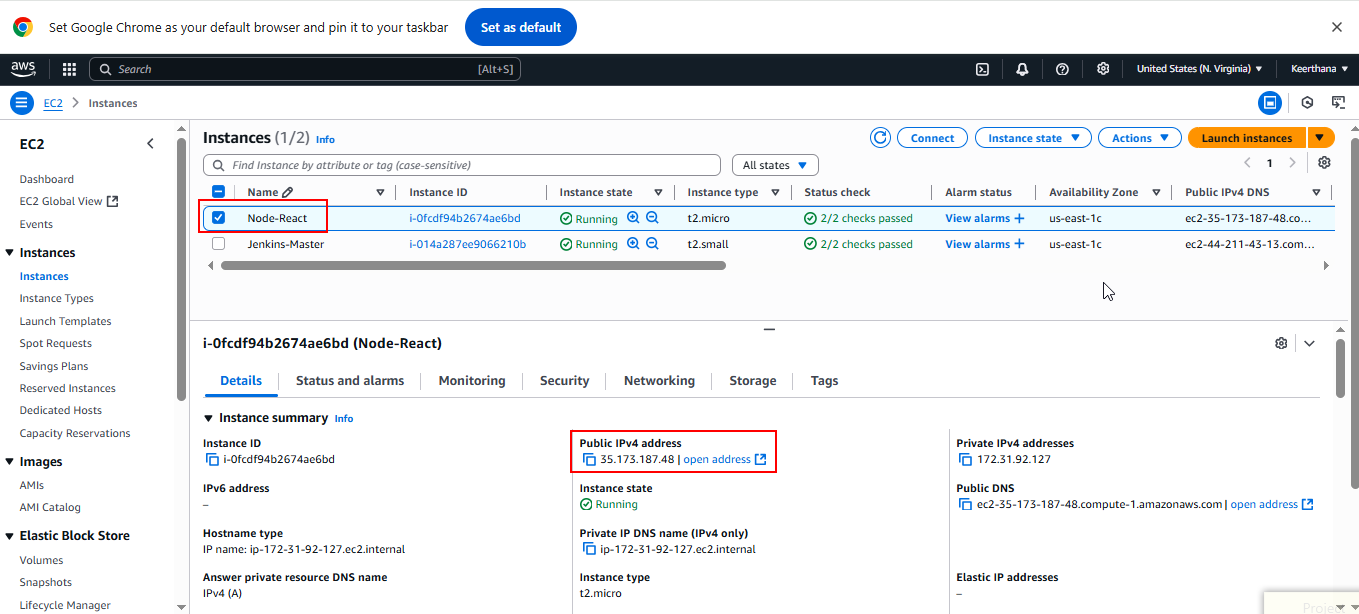
**Launch EC2 instance: Jenkins-Master**  
  


**Jenkins-Master - Security Group**



**Launch EC2 instance: Node-React**

* This machine is for deploying the application and working as a agent.
* Monitor the deployed react application in this instance.

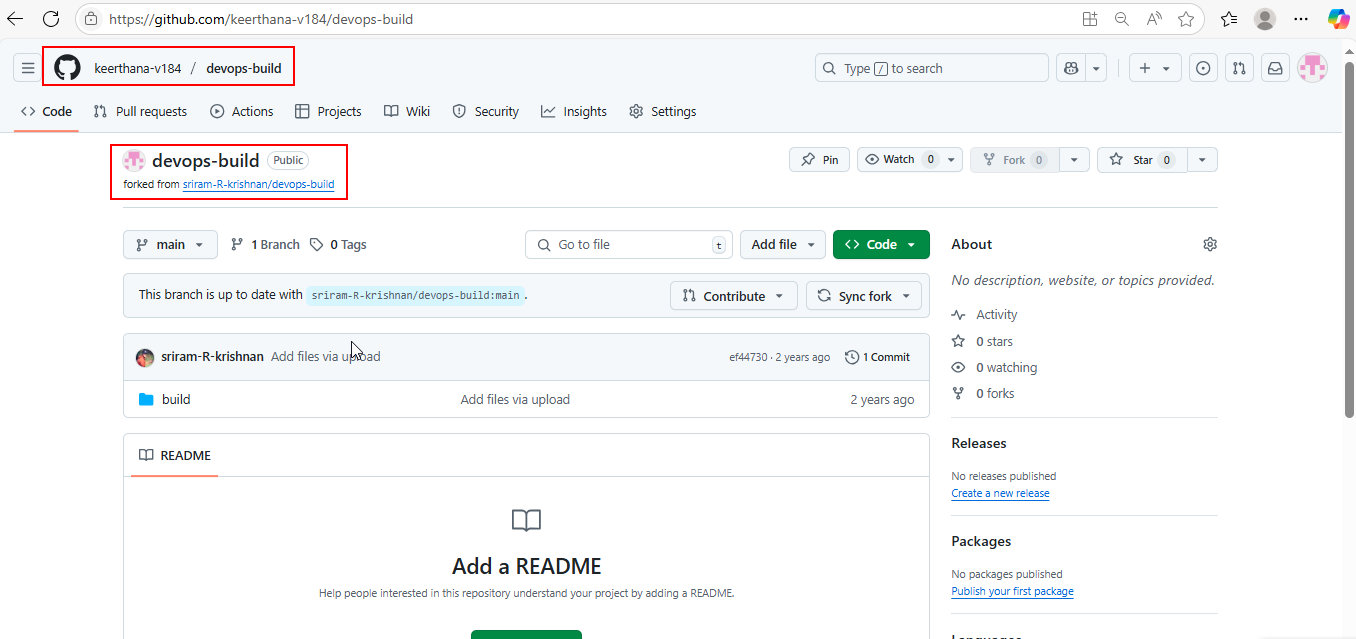


**Node-React – Security Group**



**Forked the given repo to my GitHub**

* Repo URL : <https://github.com/sriram-R-krishnan/devops-build> to keerthana-v184

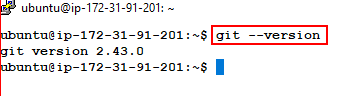


**Install Git on Jenkins-Master**

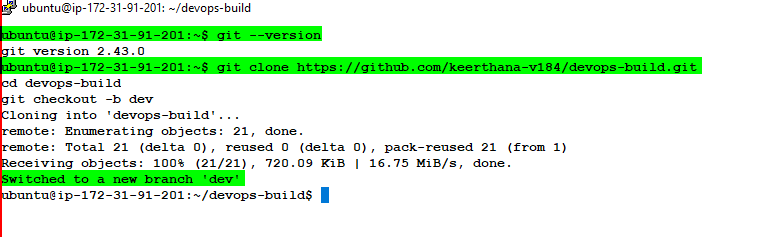
sudo apt-get update

sudo apt-get install git -y

git --version

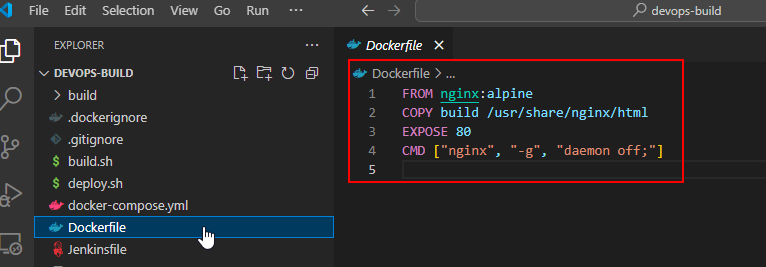


**Clone the repo and get into dev branch**

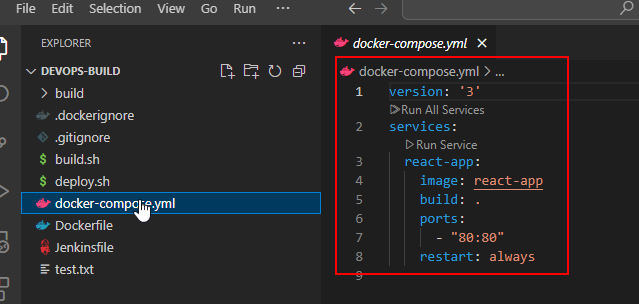


**Create the files for the Build & Deployment:**

**Create Dockerfile**

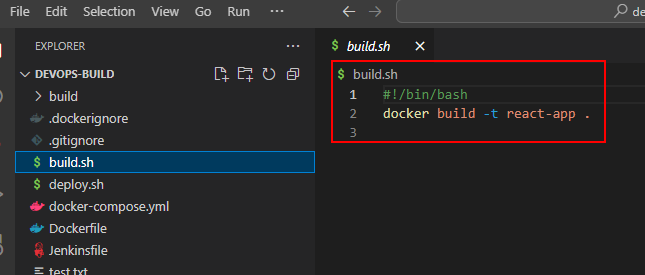


**Create docker-compose.yml**

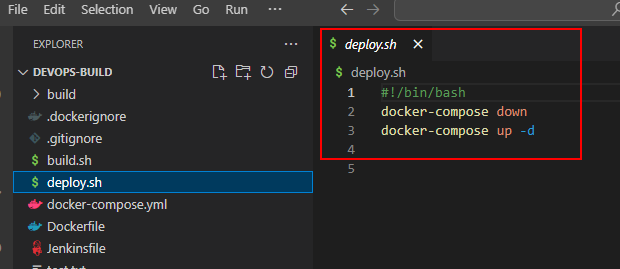


**Script files:**

**Create build.sh**



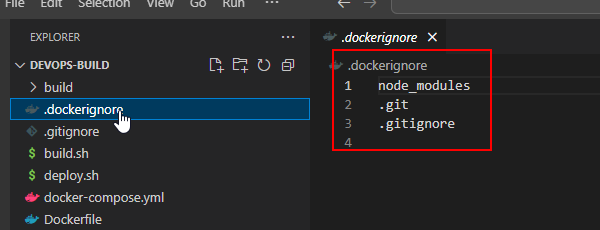
**Create deploy.sh**



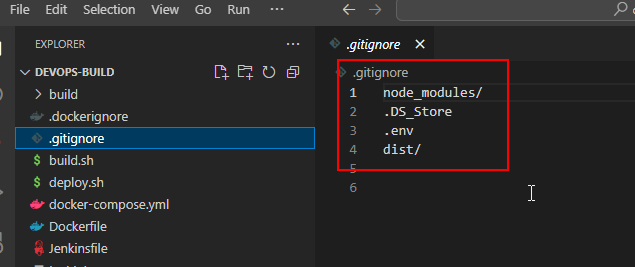
**Make them executable:**

* chmod +x build.sh deploy.sh

**Create .dockerignore**



**Create .gitignore**



**Install Docker:**

sudo apt update

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

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt update

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

sudo usermod -aG docker $USER

newgrp docker

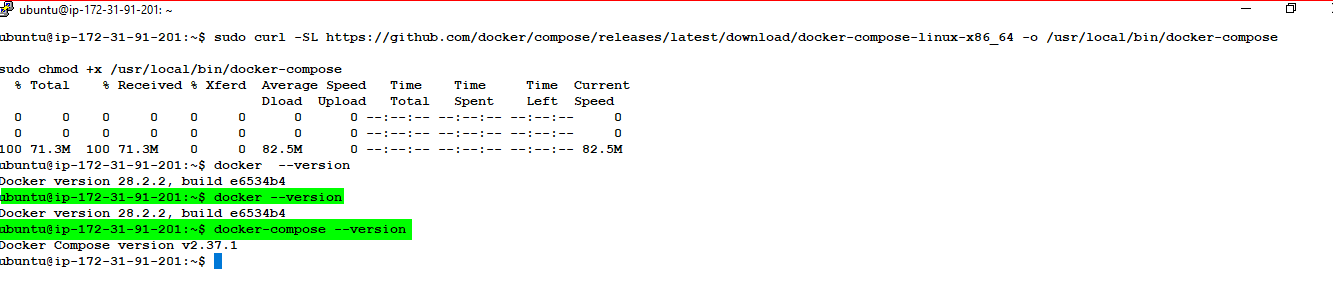
sudo systemctl enable docker

sudo systemctl start docker  
  
**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

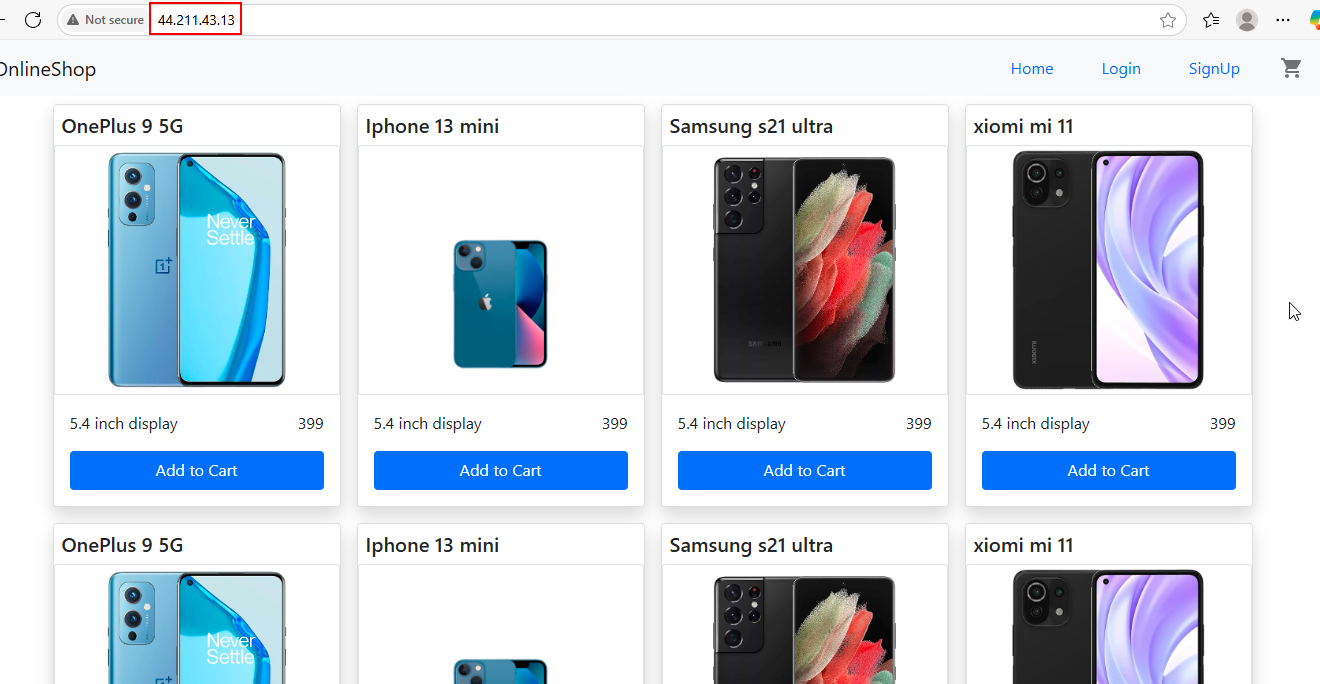
docker –version  
docker-compose ---version

  
  
**Push the files to dev branch:**

git add .

git commit -m "Files pushed"

git push origin dev

**Access the Application with Master’s Public IP:**  
  
  
  
 **Creating the Pipeline for the Automation of deployment to Node-React instance with Jenkins**

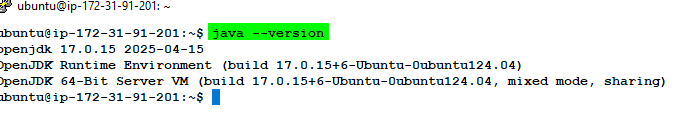
**Jenkins Setup in Master machine**

**Install Java**

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

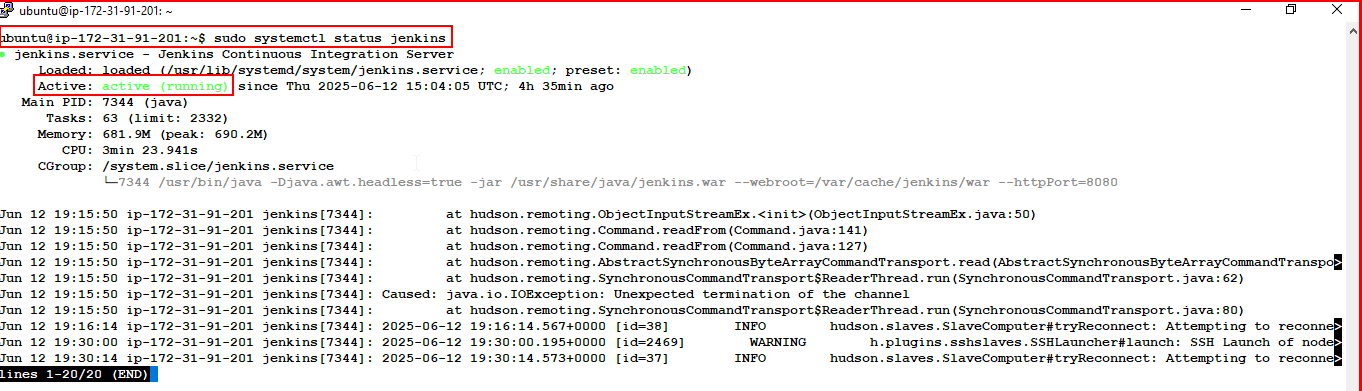
sudo apt-get update

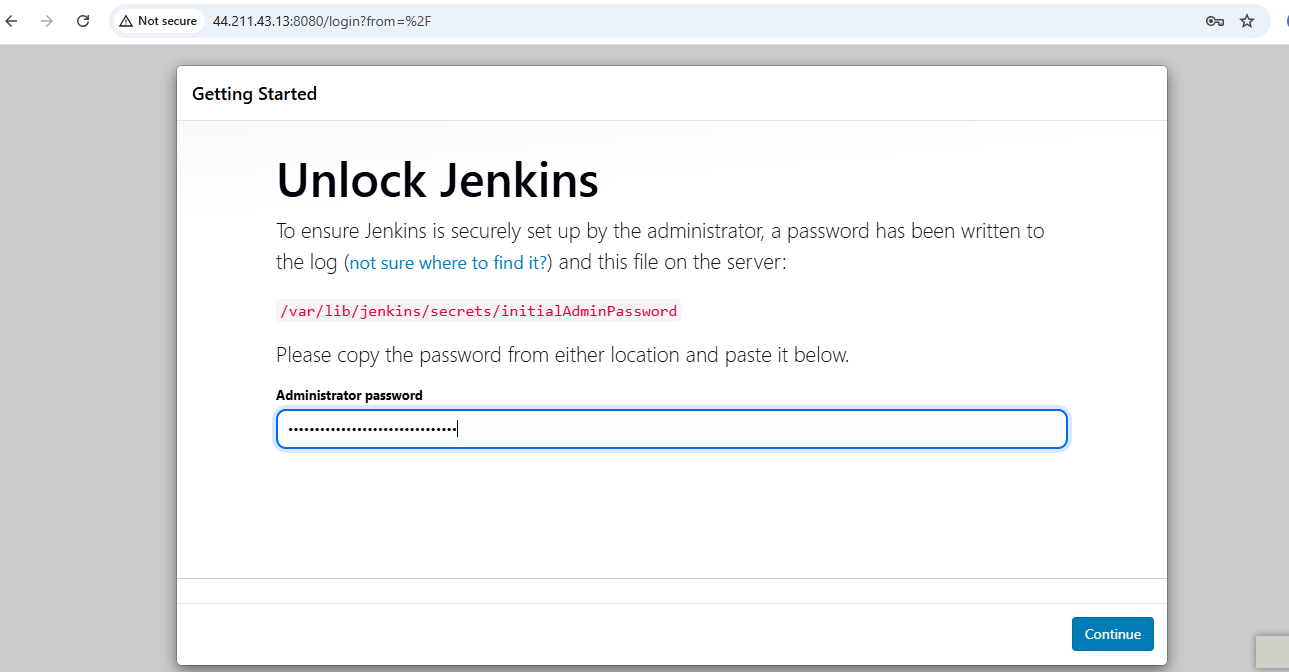
sudo apt-get install jenkins -y

sudo systemctl start jenkins

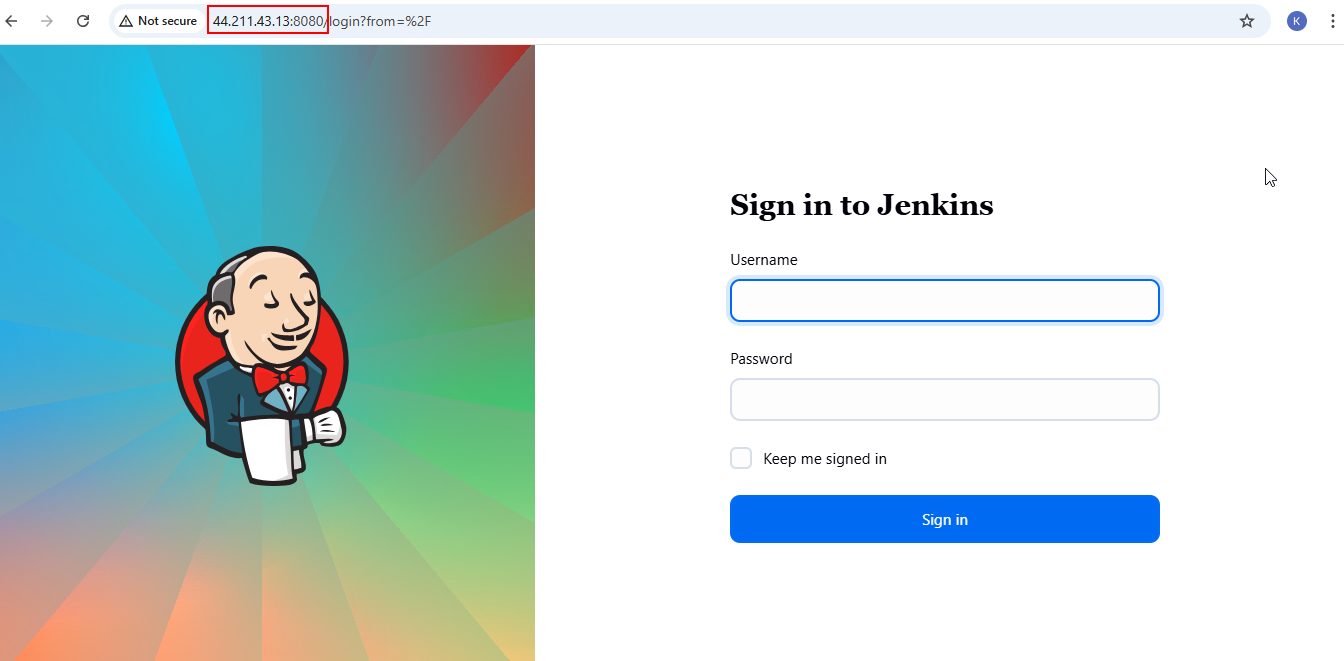
sudo systemctl enable jenkins

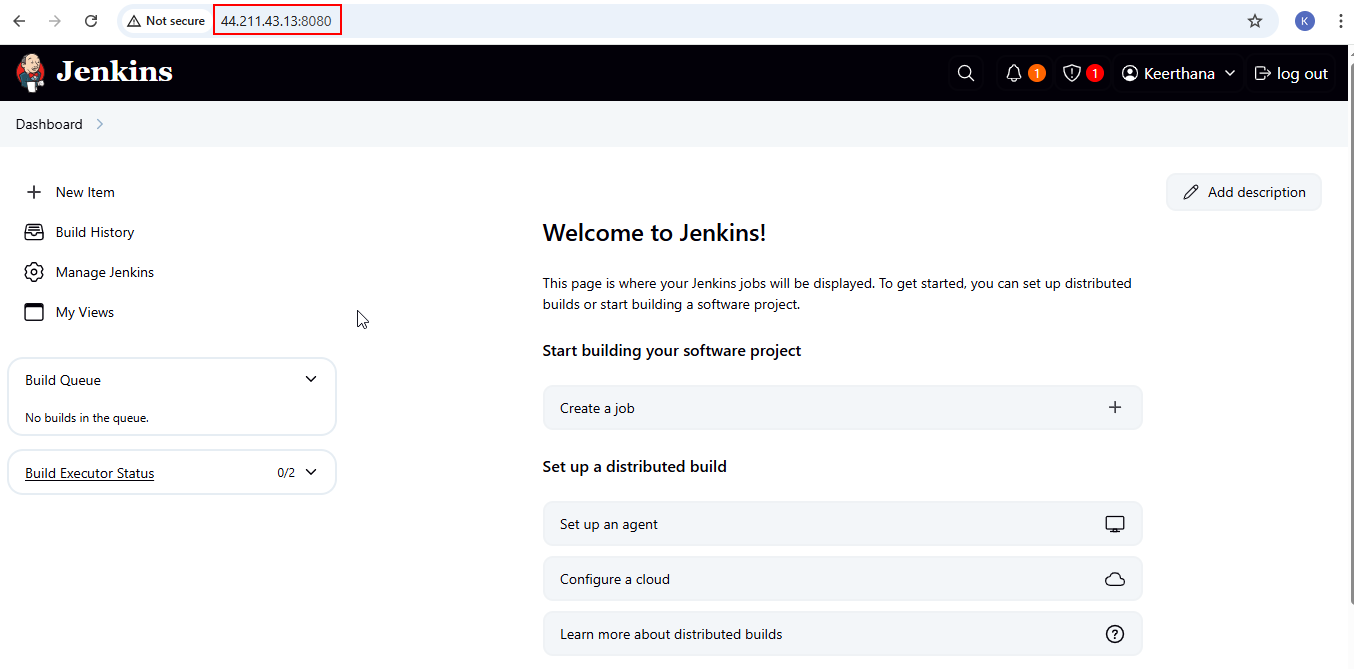
sudo systemctl status Jenkins

  
  
sudo cat /var/lib/jenkins/secrets/initialAdminPassword 🡪 to get password

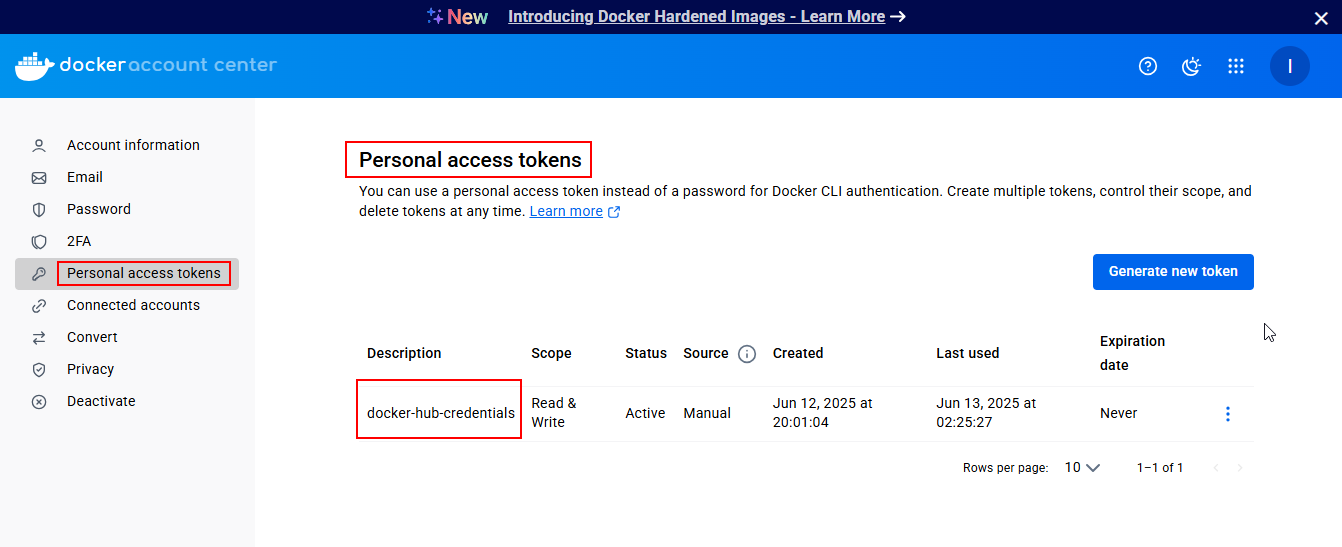


**Access with Public IP 44.211.43.13:8080**Jenkins home page



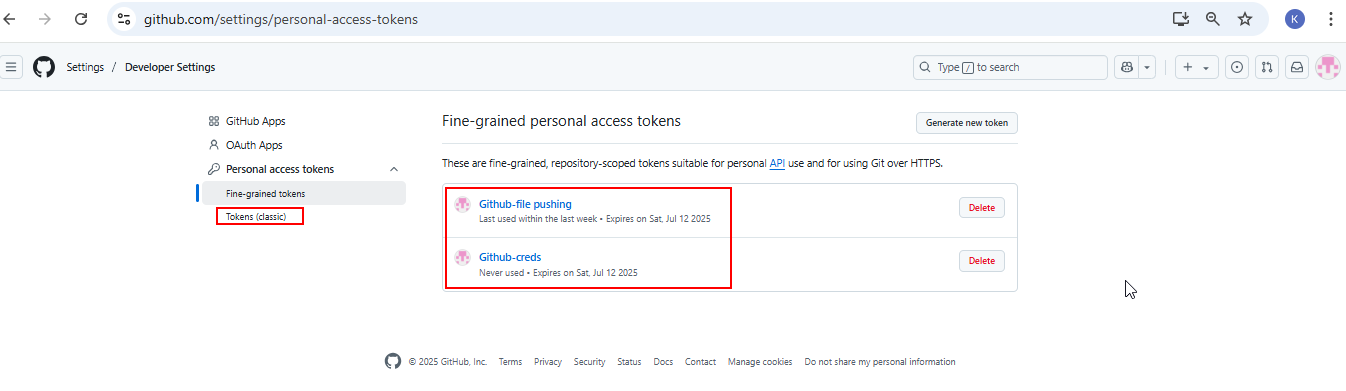
  
  
  
**Create Docker Hub Token:**

* Create access token credential



**Create Github Token:**

* Create access token credential



**Add Credentials in Jenkins:**

**Git Hub credentials (Username with password)**

* ID: github-credentials
* Username: keerthana-v184
* Password: Access Token

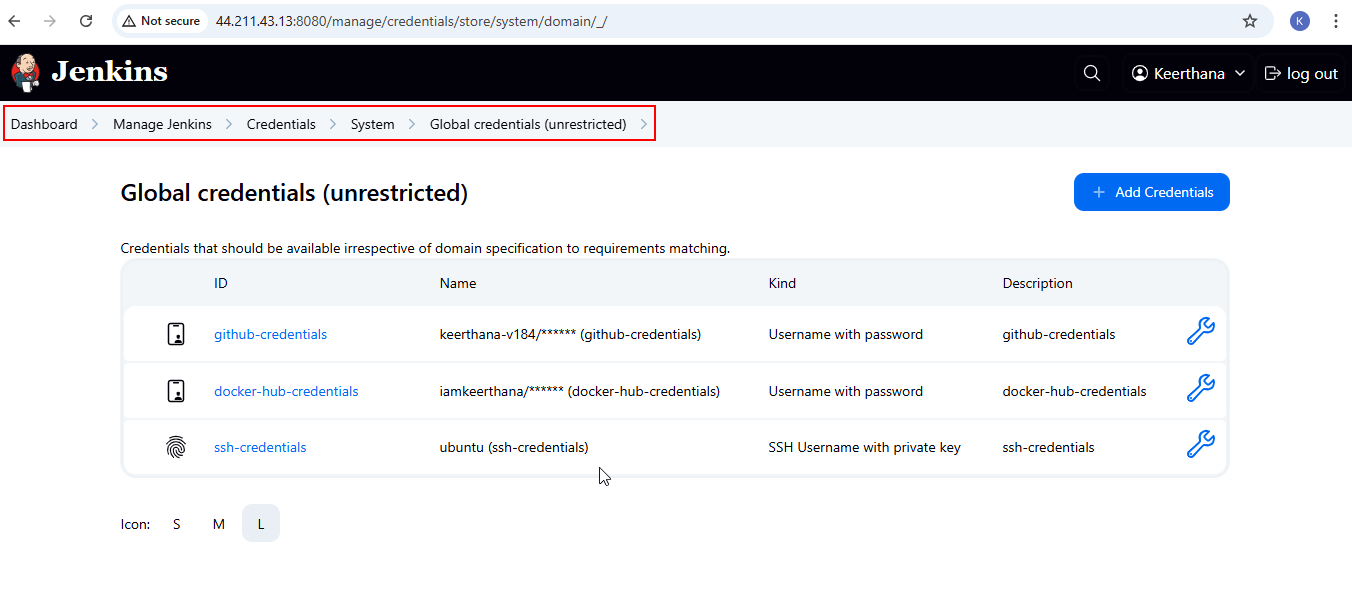
**Docker Hub credentials**

* ID: docker-hub-credentials
* Username: iamkeerthana
* Password: Access Token

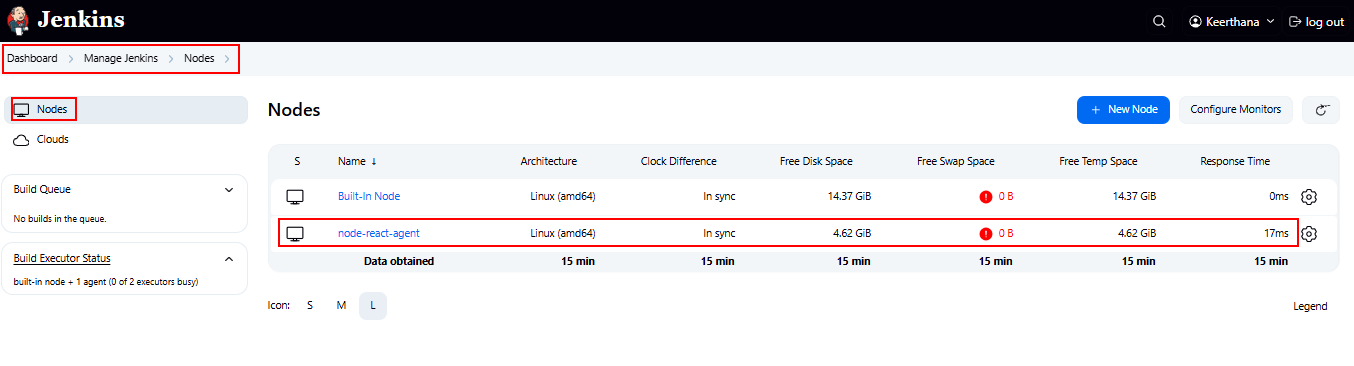
**SSH credentials for Node machine**

* ID: ssh-credentials
* Username: ubuntu
* Private key: SSH private key

**Credentials added**

  
  
**Adding Node**

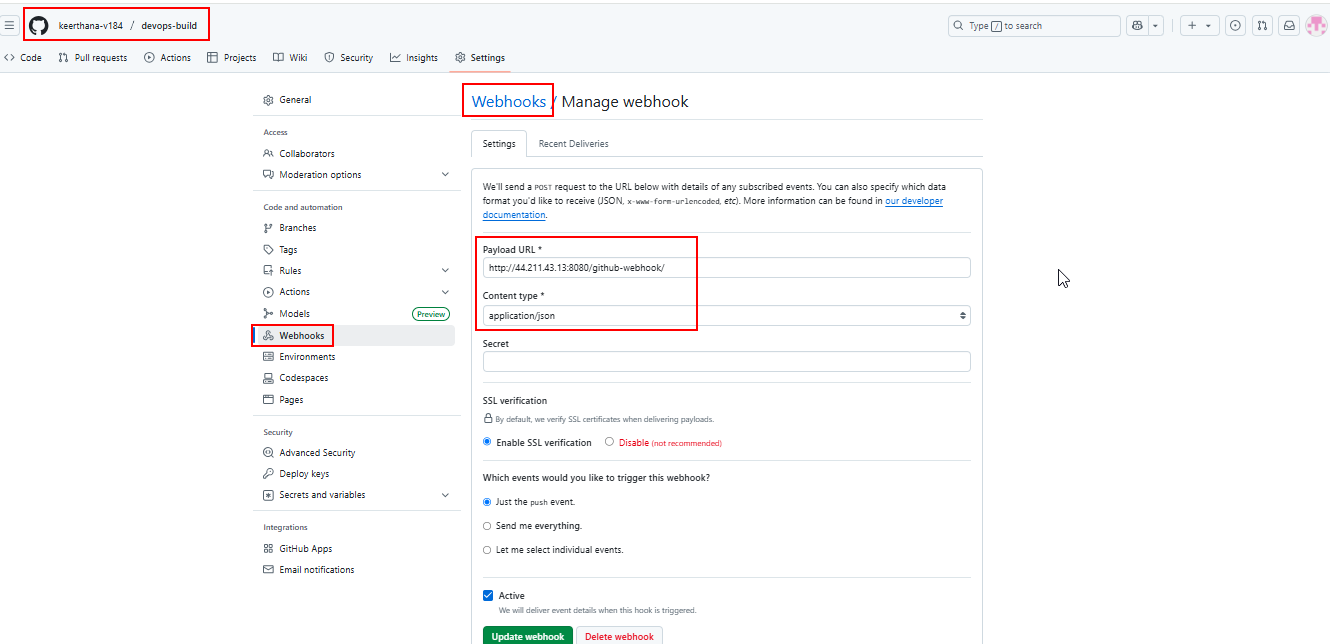
* Node-React instance added as agent for deployment.



**Set Up GitHub Webhook**

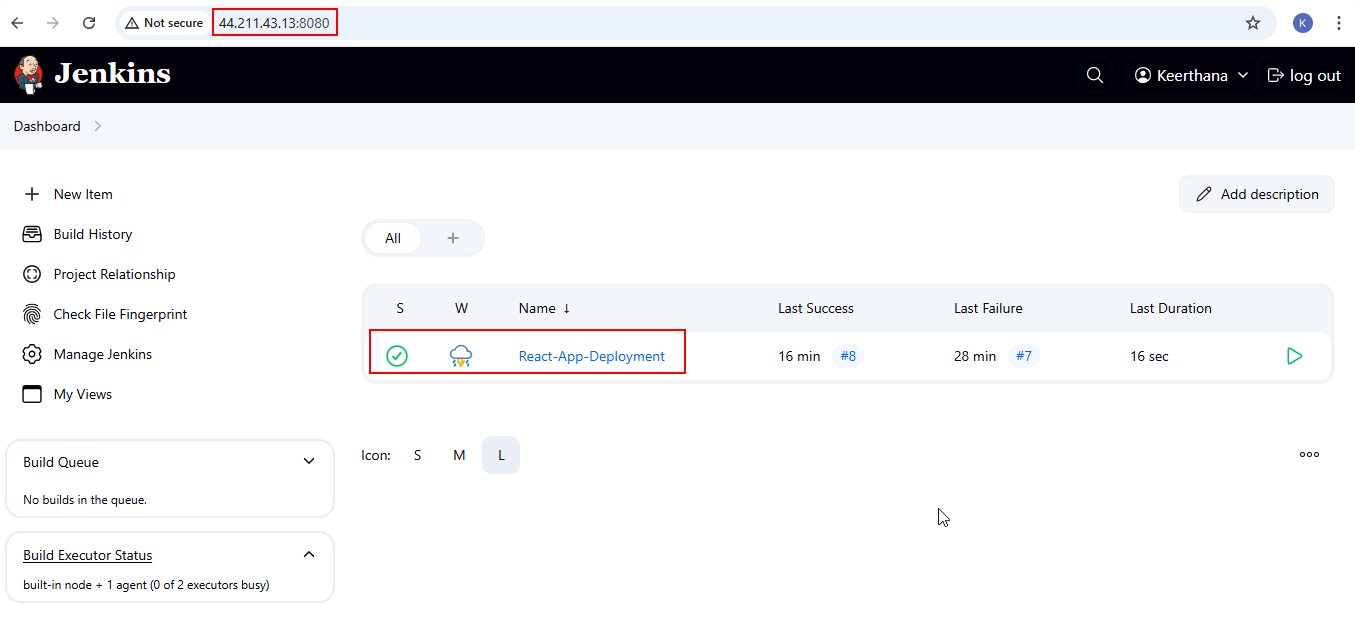
1. GitHub repo > Settings > Webhooks
2. Add webhook:

* Payload URL: <http://44.211.43.13:8080/github-webhook/>
* Content type: application/json
* Which events: Just the push event 🡪 save

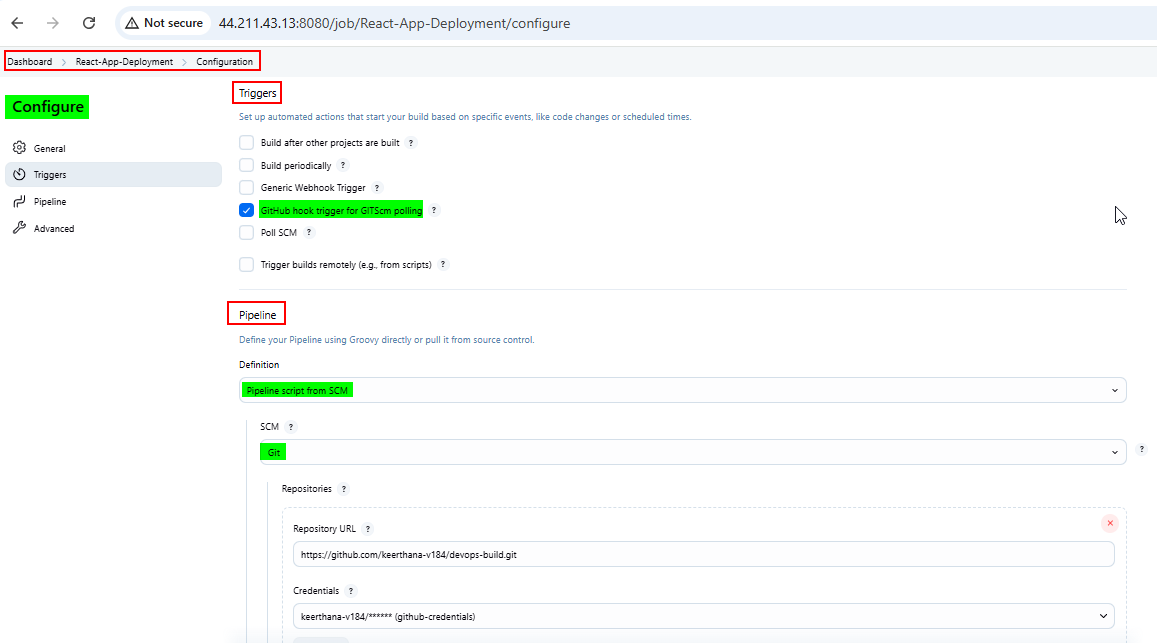


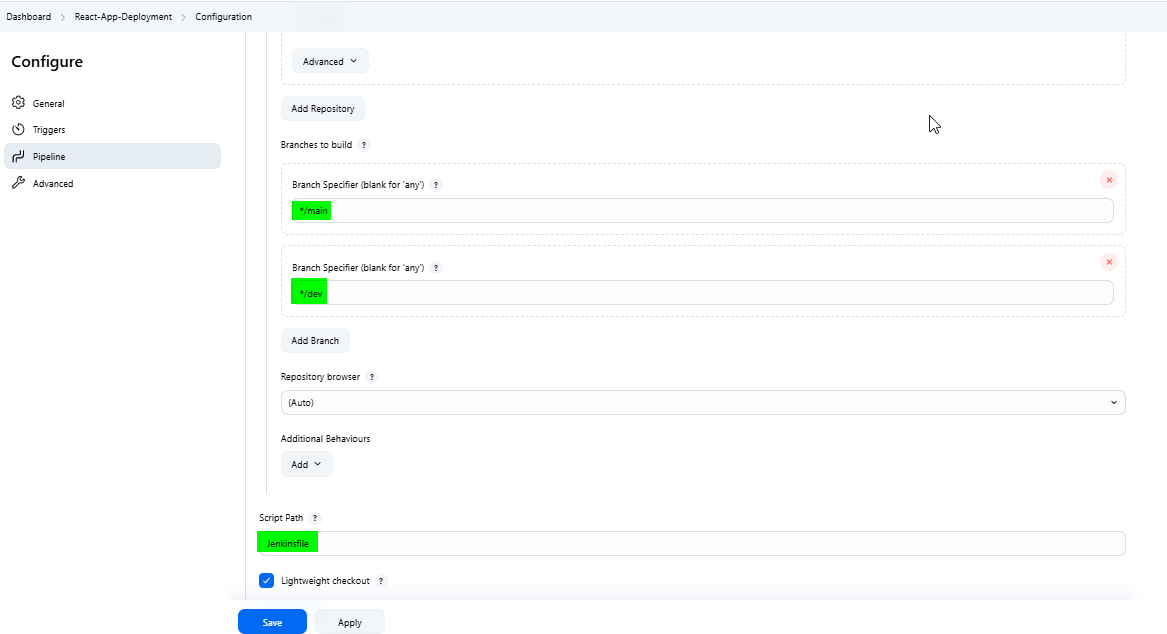
**Create Pipeline in Jenkins**

1. New Item > Pipeline
2. Name: React-App-Deployment
3. Pipeline > Definition: Pipeline script from SCM
4. SCM: Git
5. Repository URL: <https://github.com/keerthana-v184/devops-build.git>
6. Credentials: GitHub credentials
7. Branch Specifier: \*/dev and \*/main
8. Script Path: Jenkinsfile
9. Save



**Pipeline Configuration:**

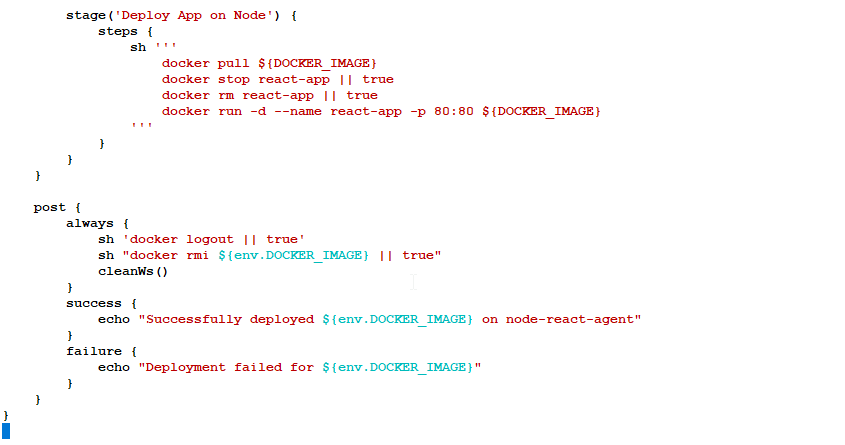




**Create Jenkinsfile**





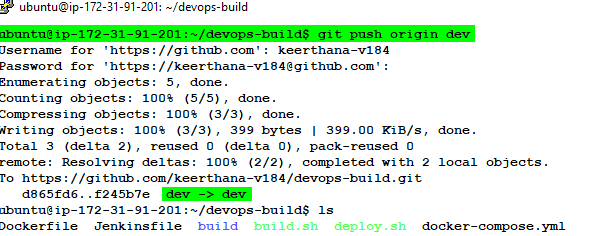


**Push the files to dev branch:**

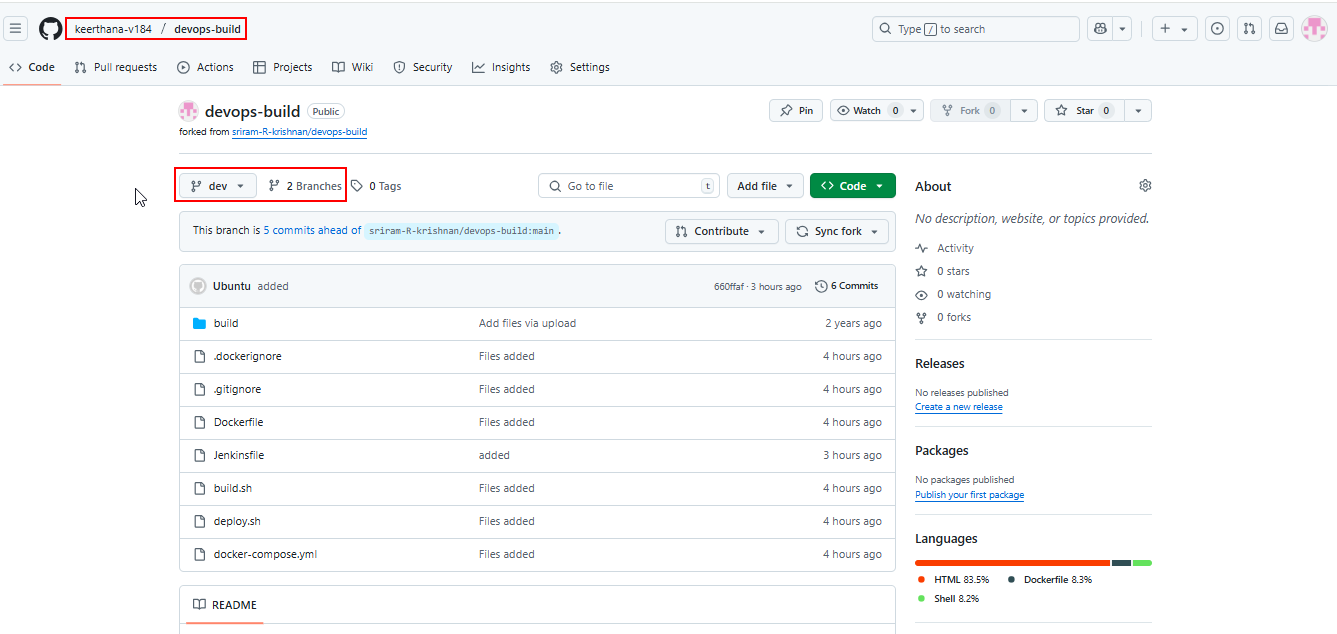
git add .

git commit -m "Your commit message"

git push origin dev

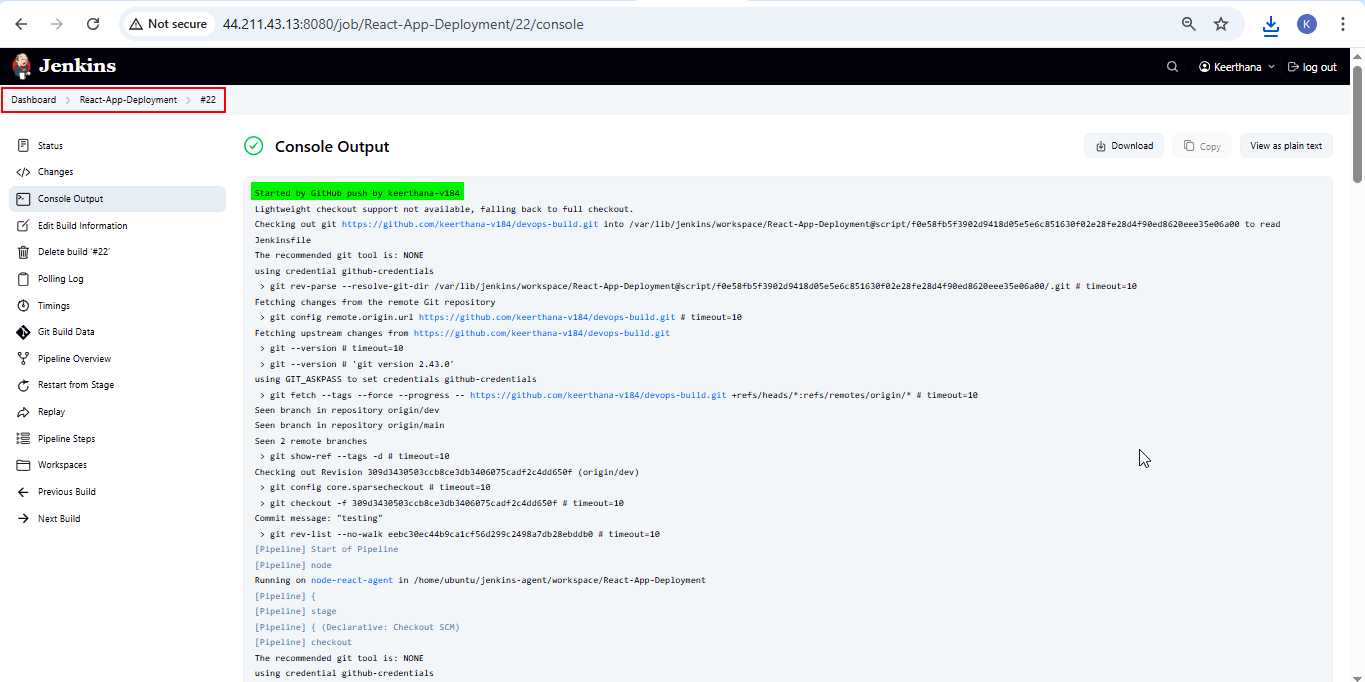
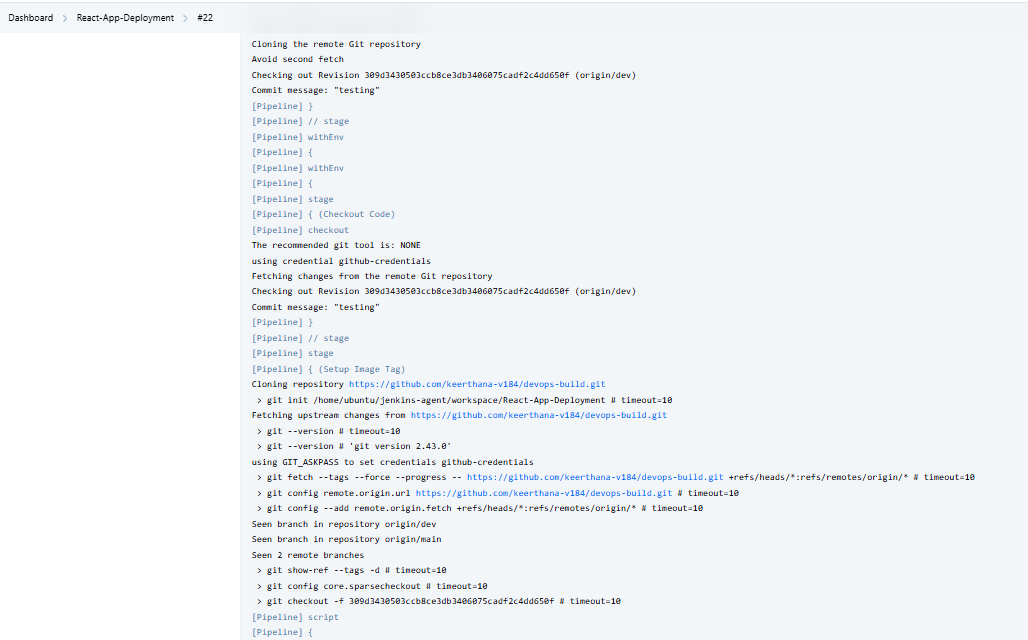
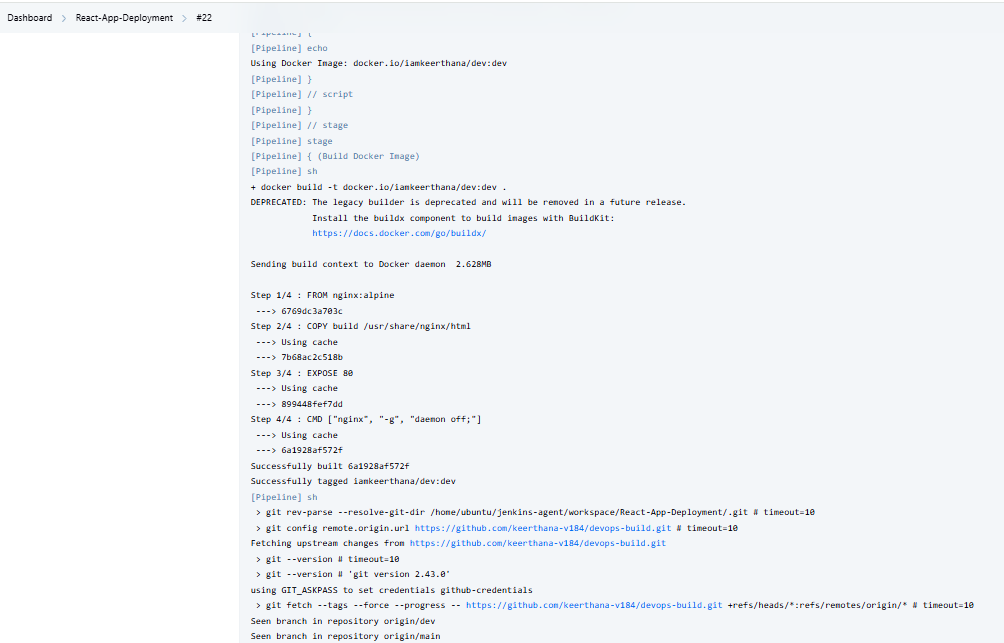
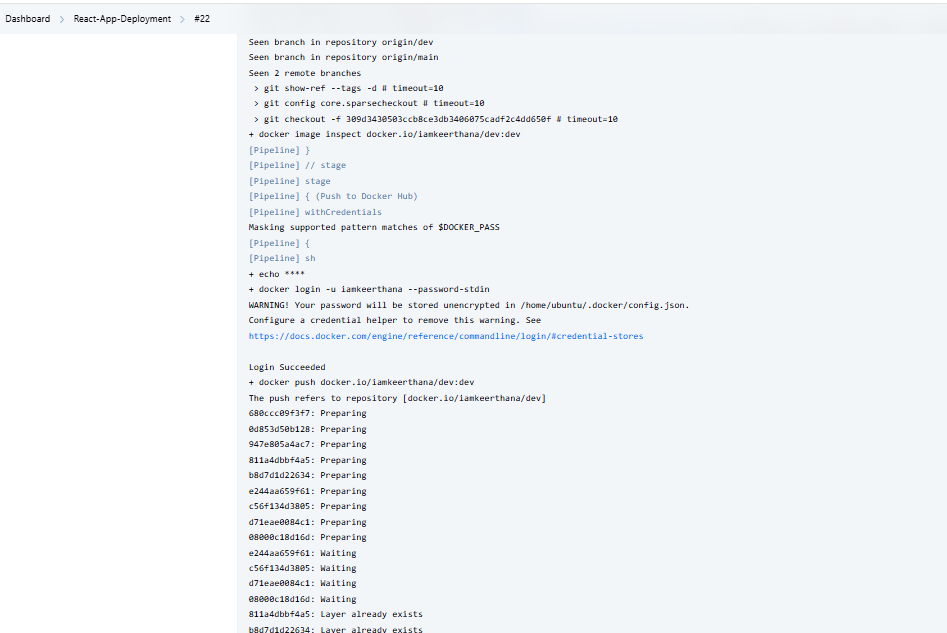
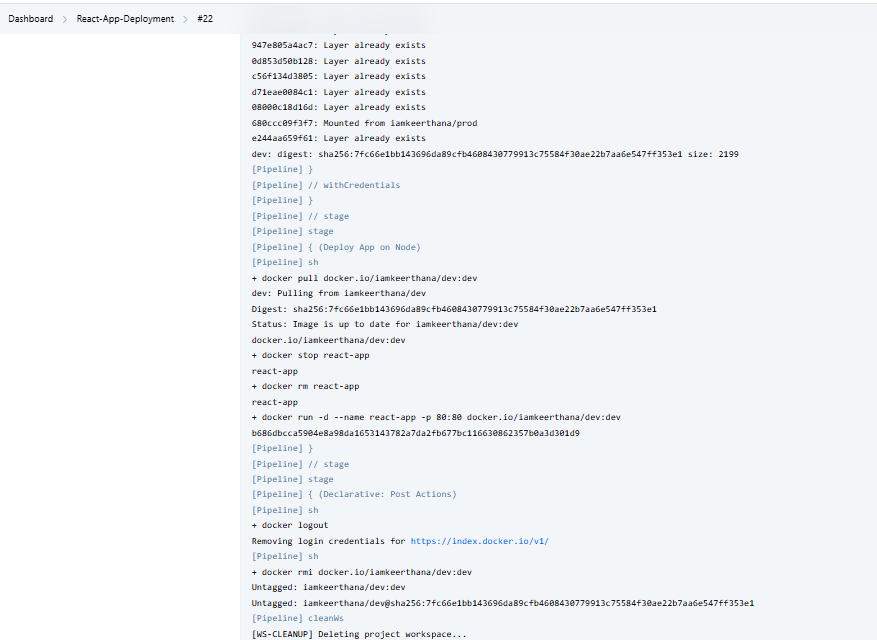
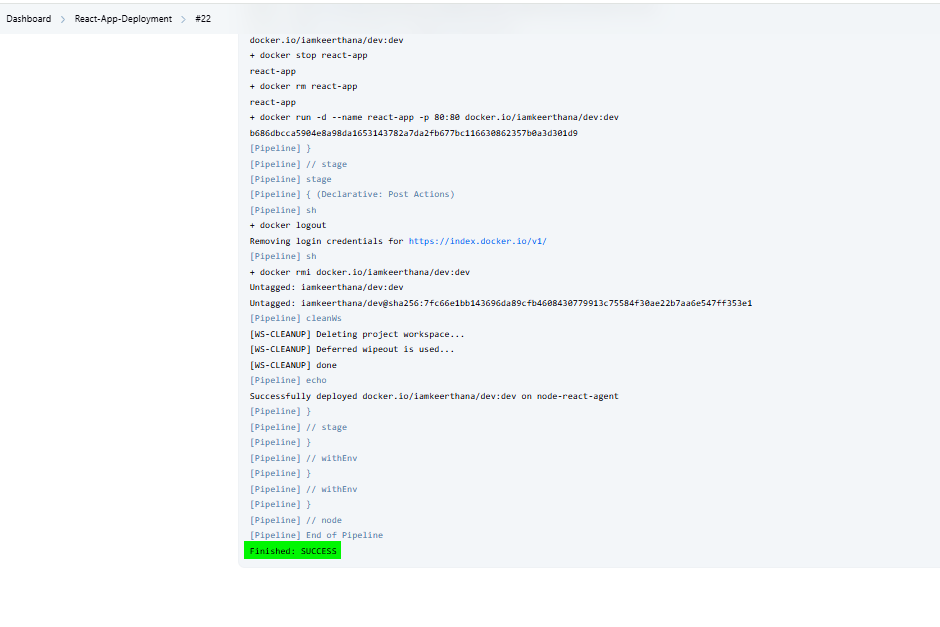


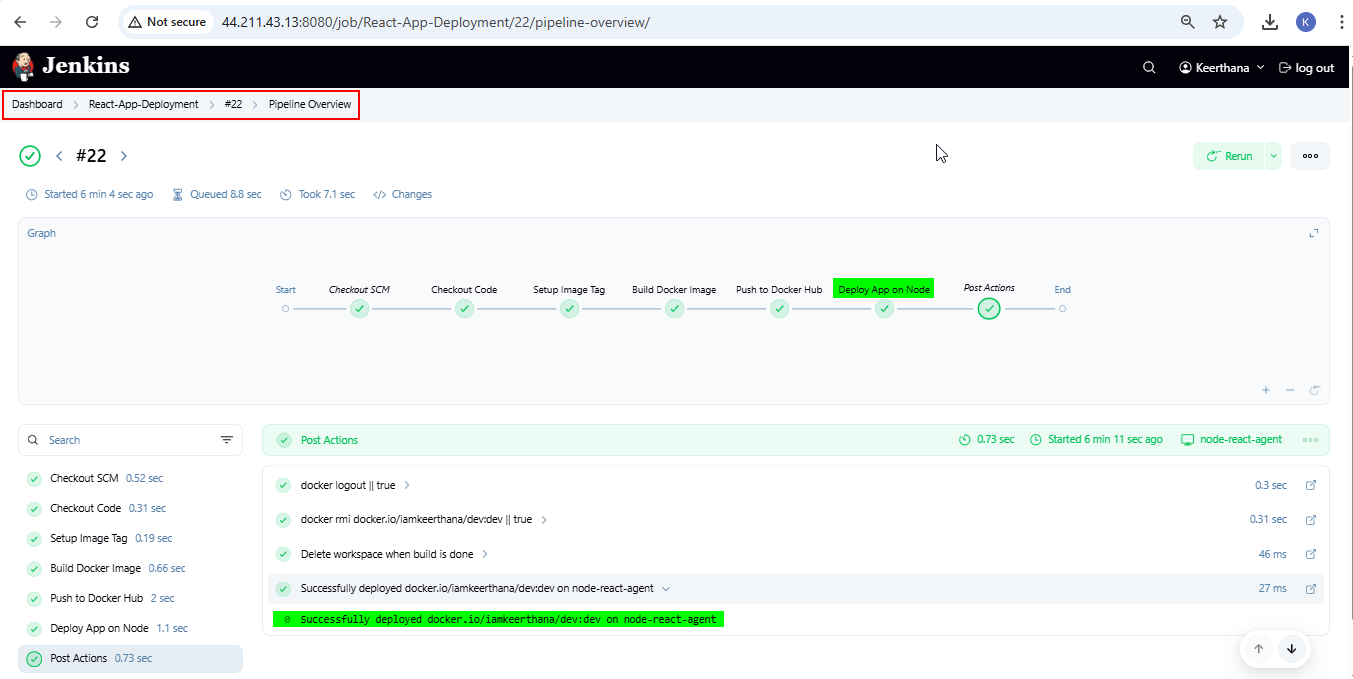
**GitHub - Dev branch**



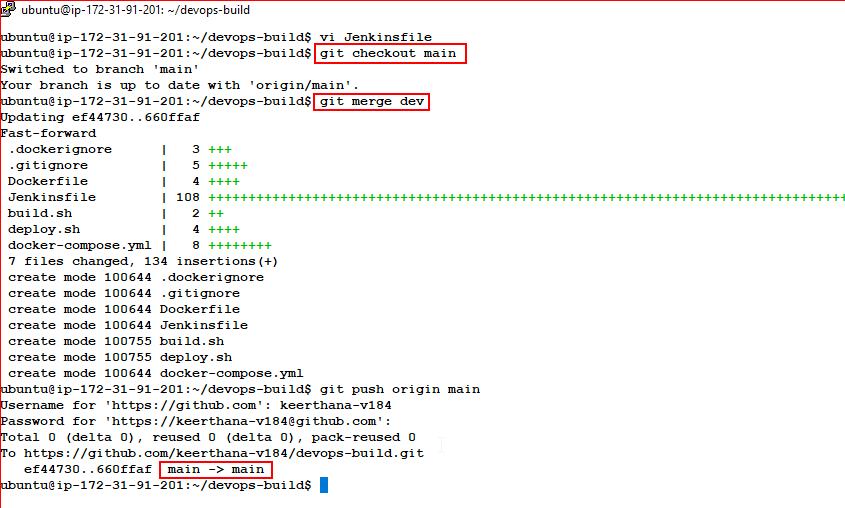
**Console Output – dev branch**

* [Console Output - dev branch #22.txt](Console%20Output%20-%20dev%20branch%20#22.txt)

**Pipeline Overview:  
  
**

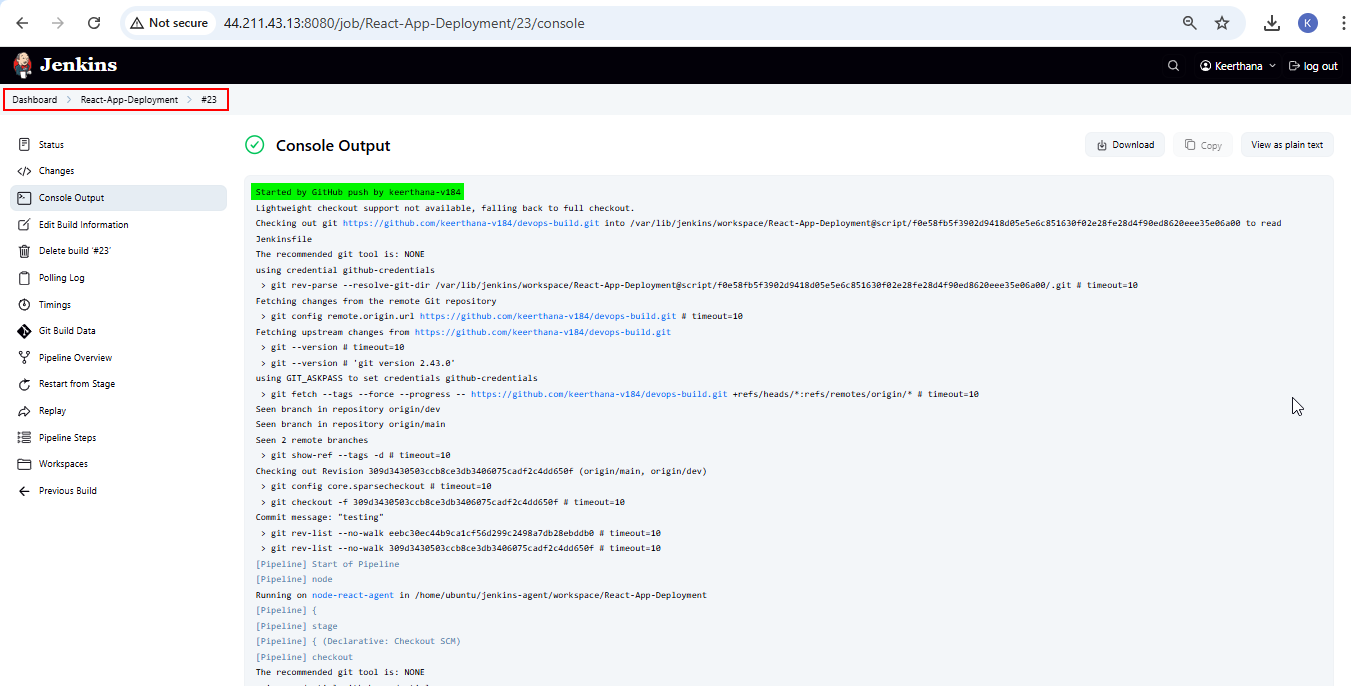
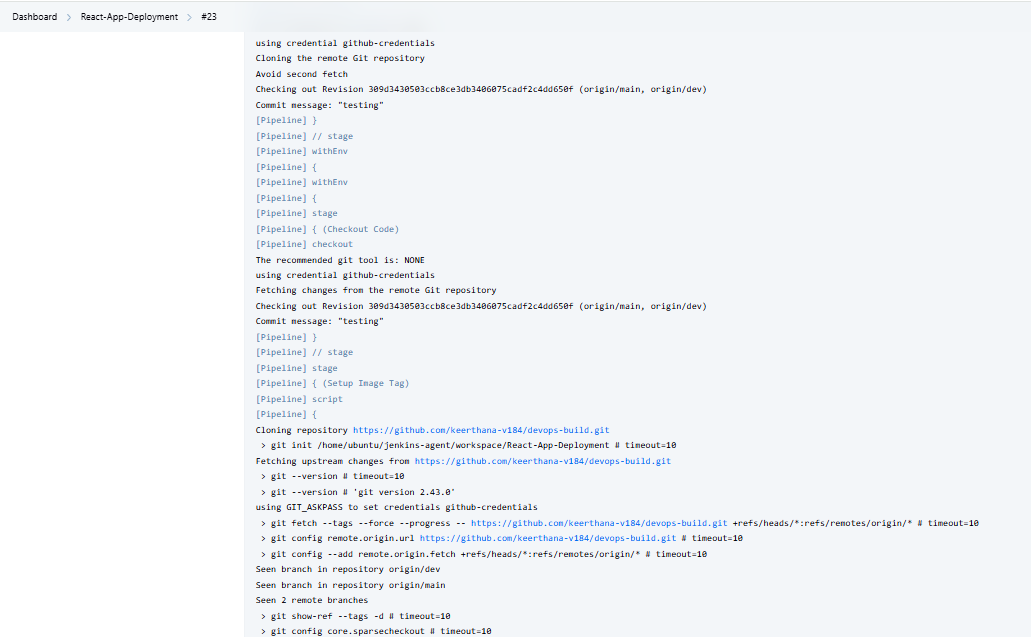
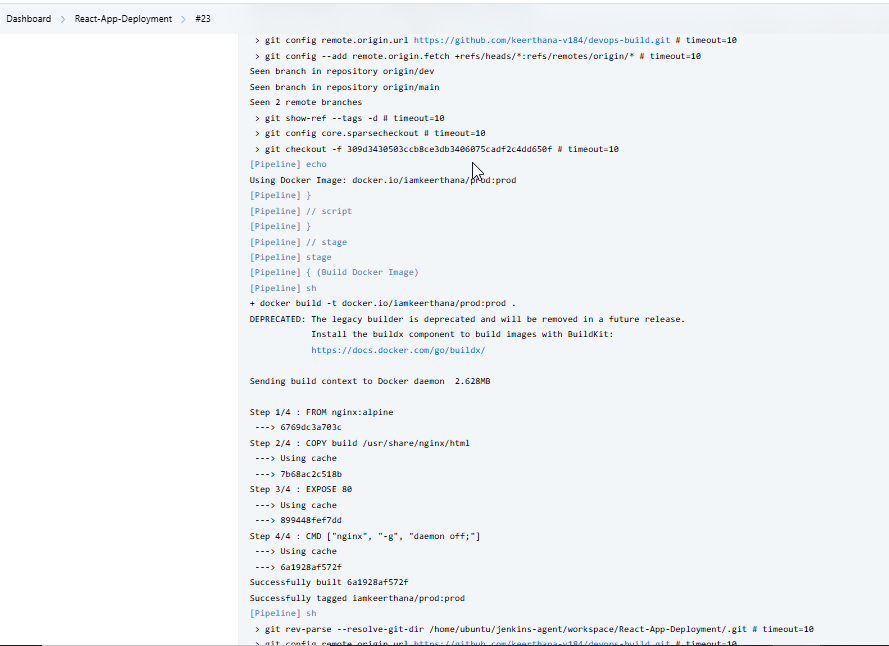
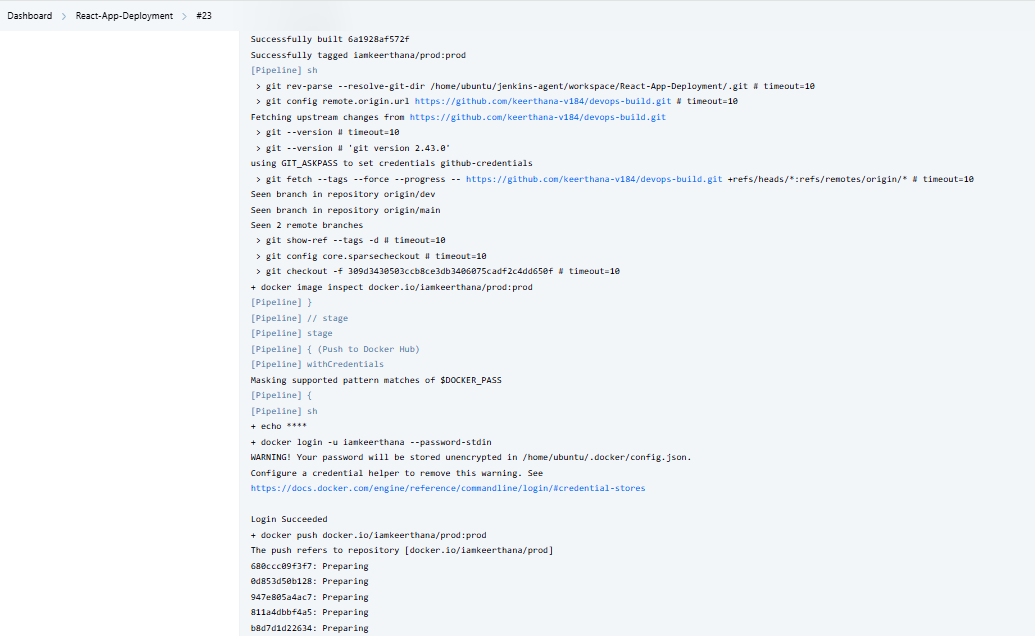
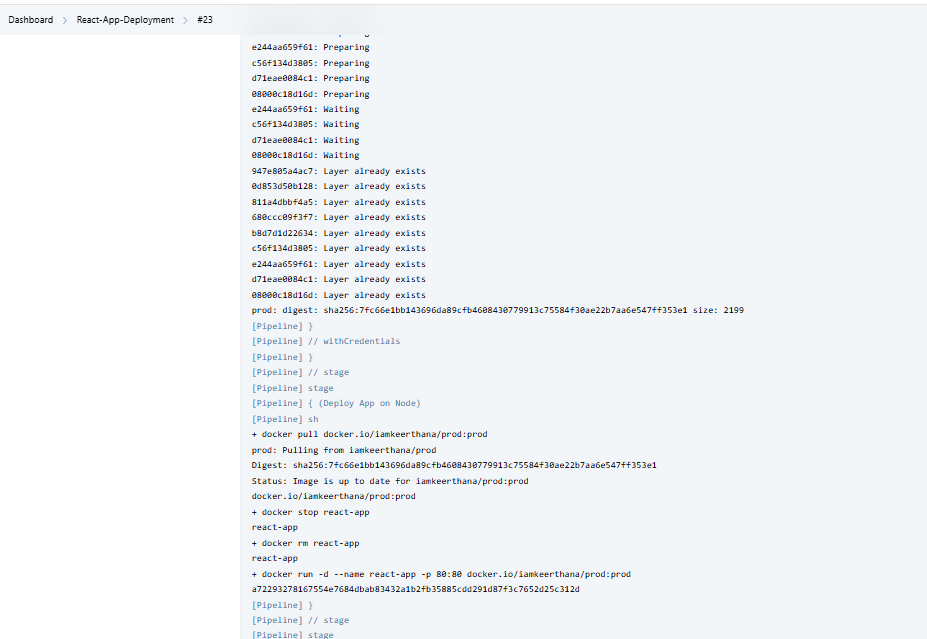
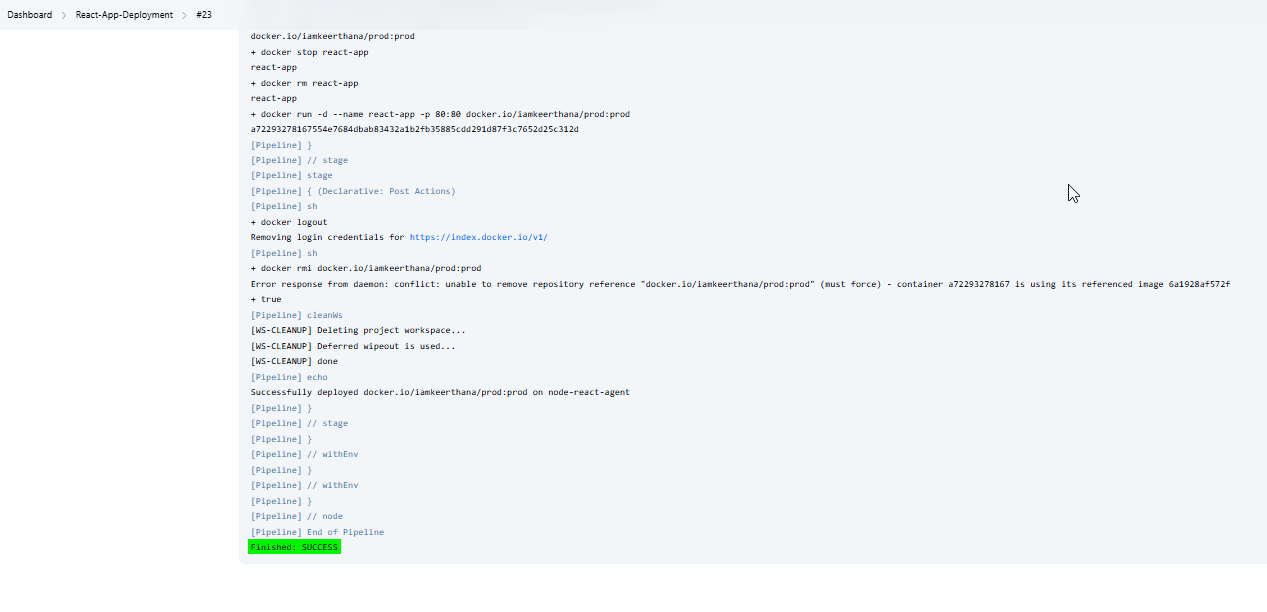
**Merge dev branch to main branch:**



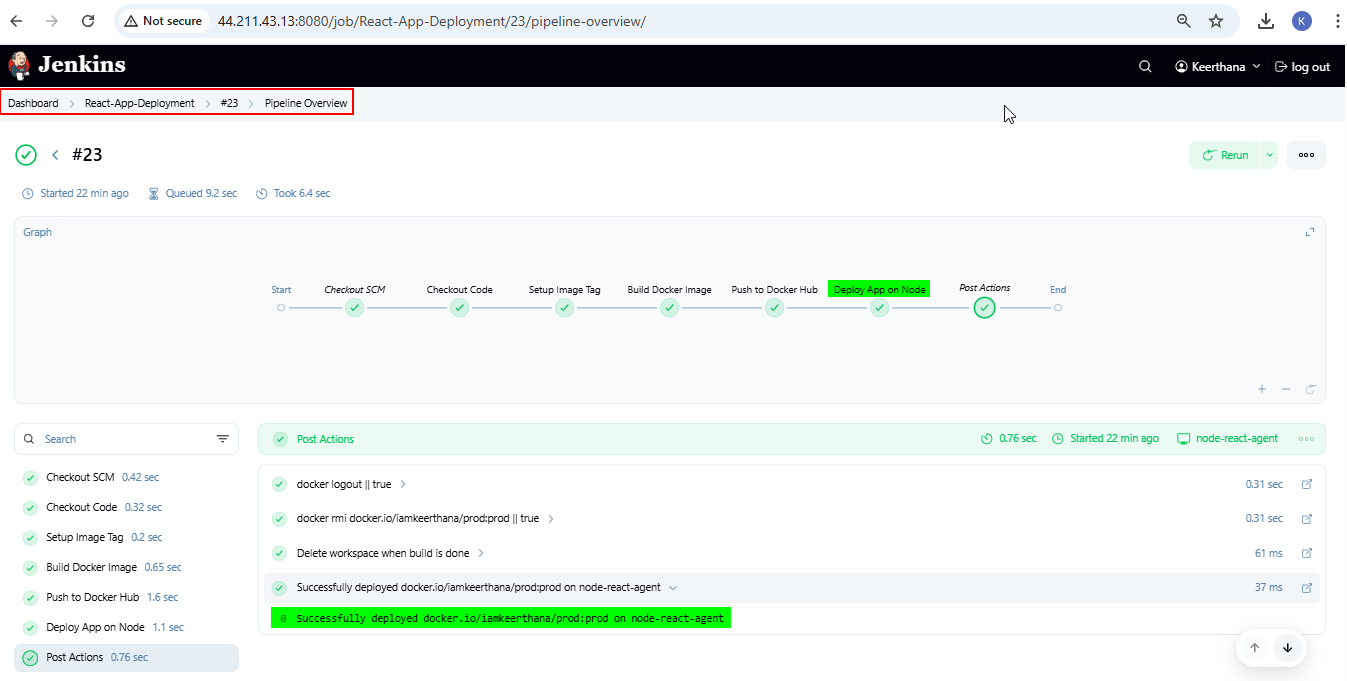
**GitHub - main branch**  
  


**Console Output – main branch**

* [Console Output - main branch #23.txt](Console%20Output%20-%20main%20branch%20#23.txt)

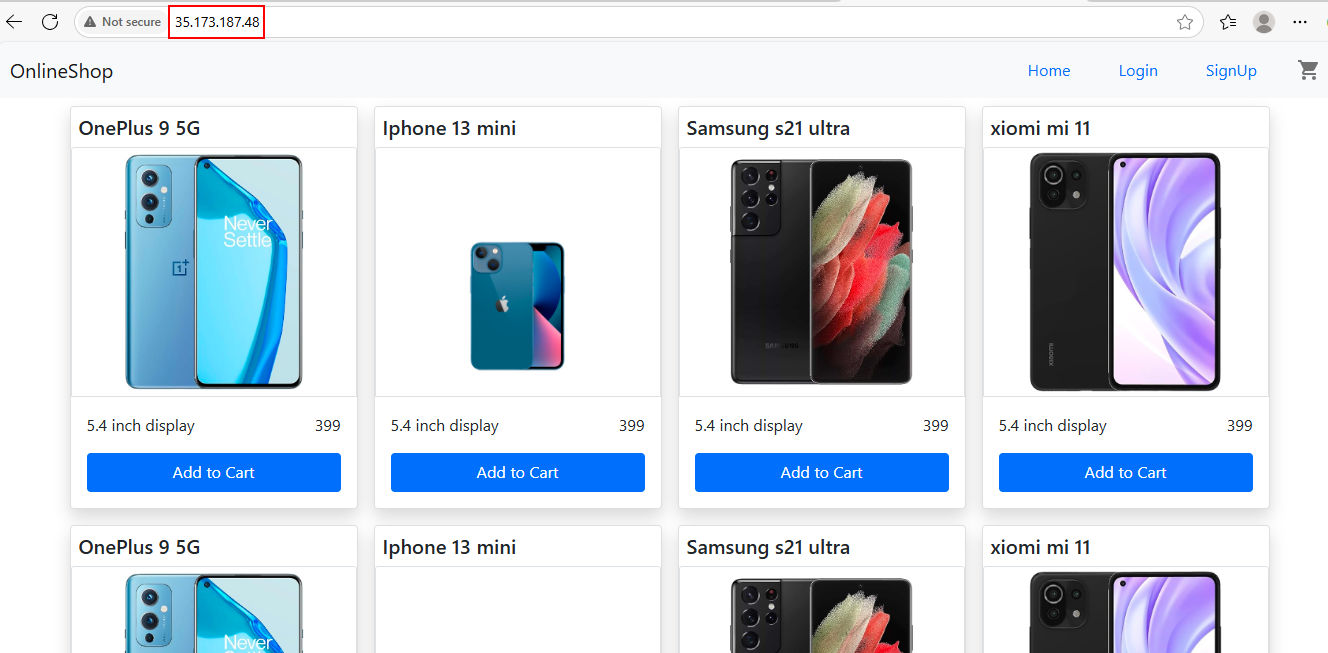
  
  
  
  
  
  
  
  
  
  


**Pipeline Overview:**

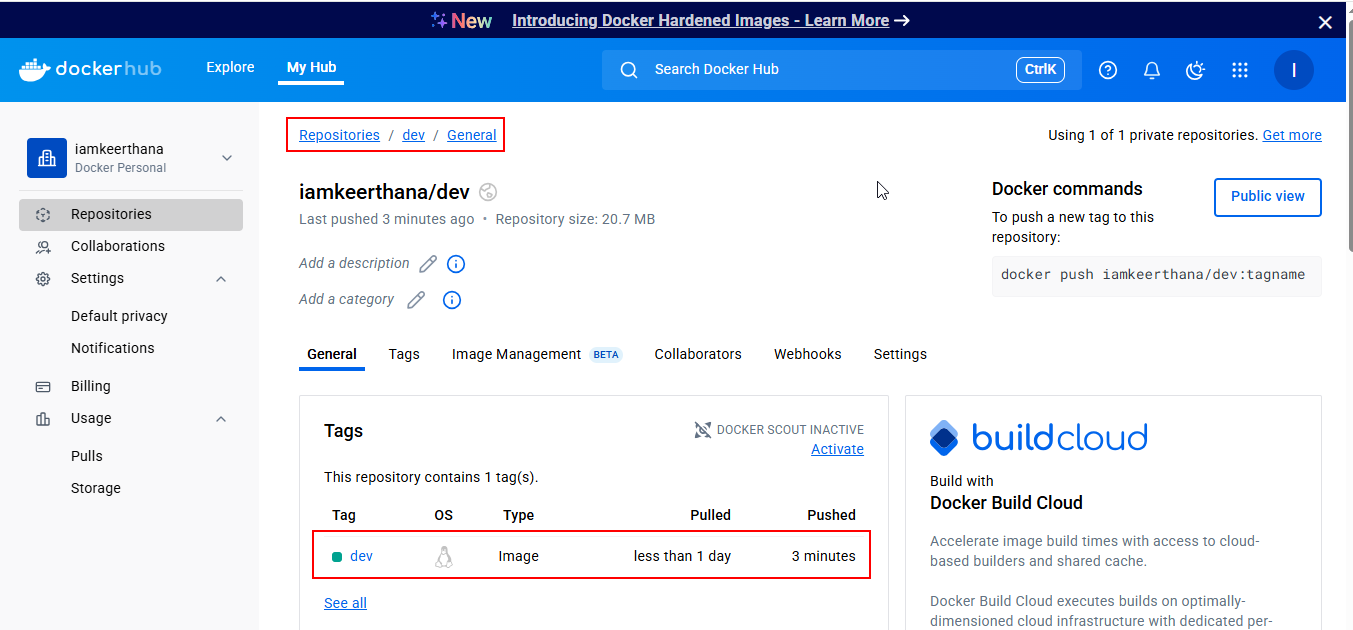


**Access the React Application via browser with Public IP**

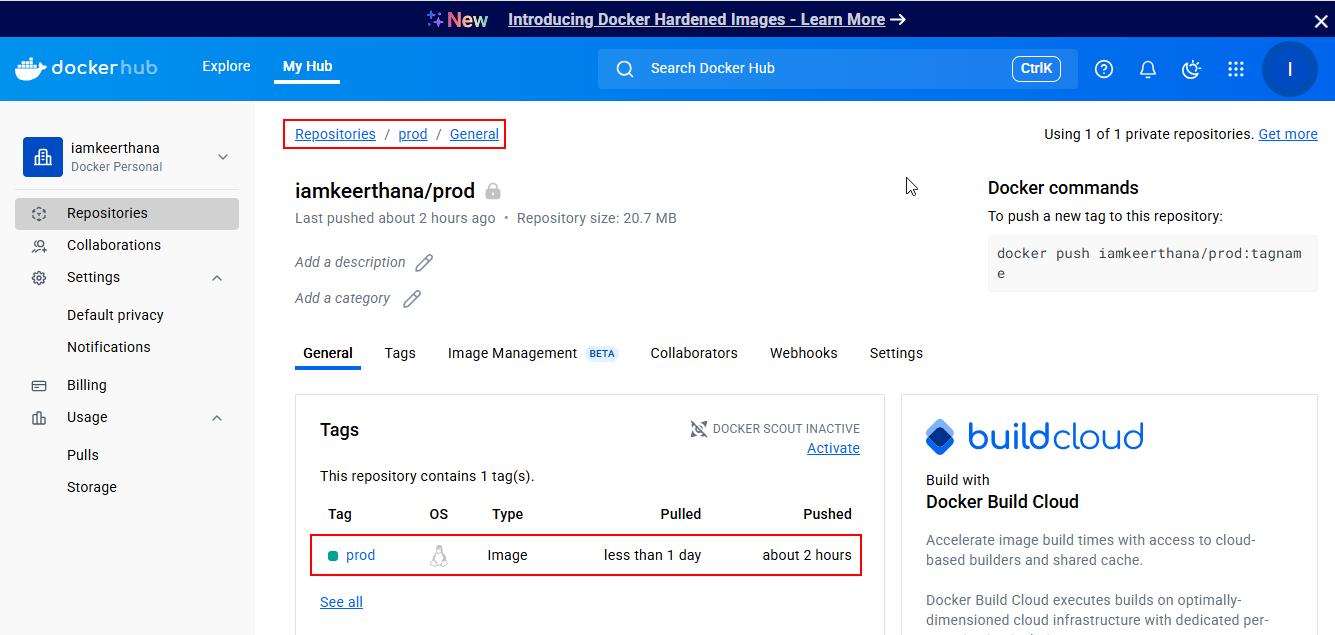
* Deployed site URL - <http://35.173.187.48>

  
  
**Images pushed to Docker Hub**

**Public repo – dev repo (dev branch)**



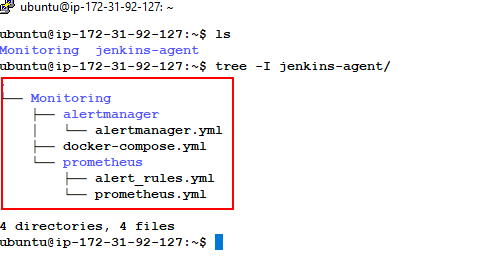
**Private repo – prod repo (main branch)**



**Monitoring for Node-React instance**

**Using Prometheus and Grafana**

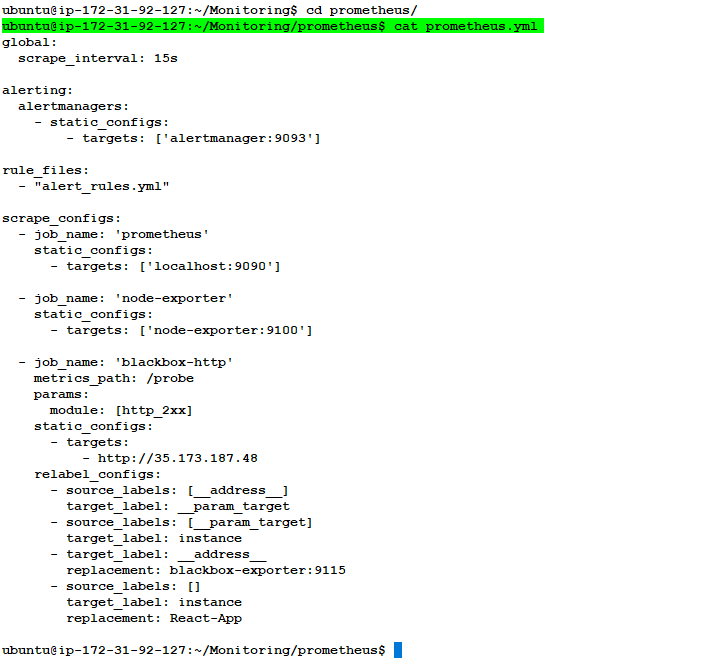
**File structure:**

  
**Create files for Monitoring:**  
  
mkdir Monitoring && cd Monitoring

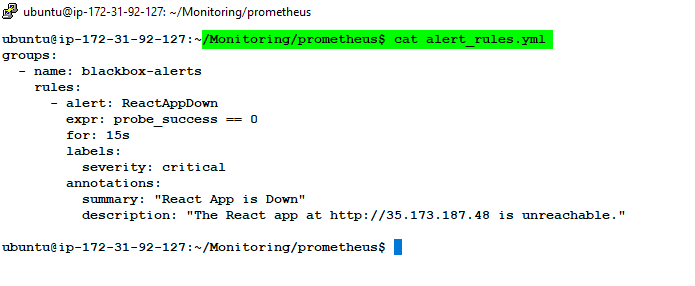
**Create docker-compose.yml file**



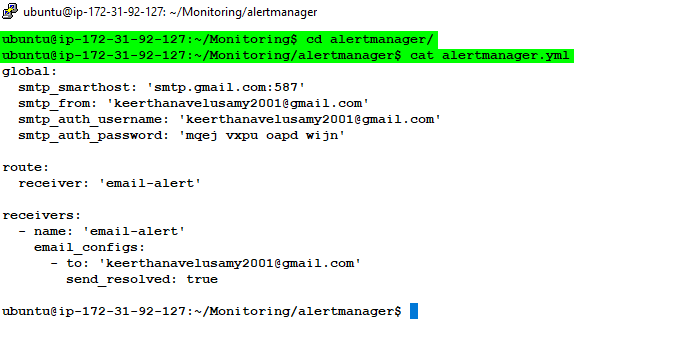
**Create prometheus.yml**



**Create alrt\_rules.yml**



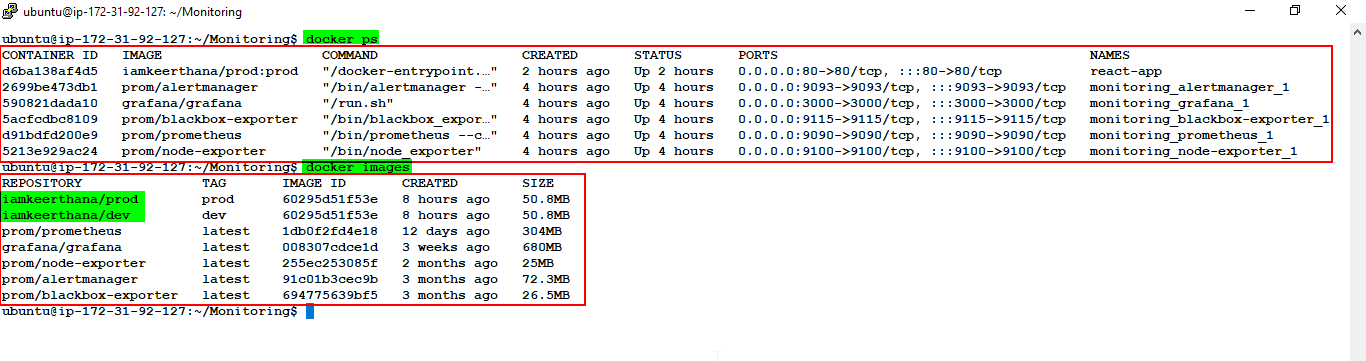
**Create alertmanager.yml**

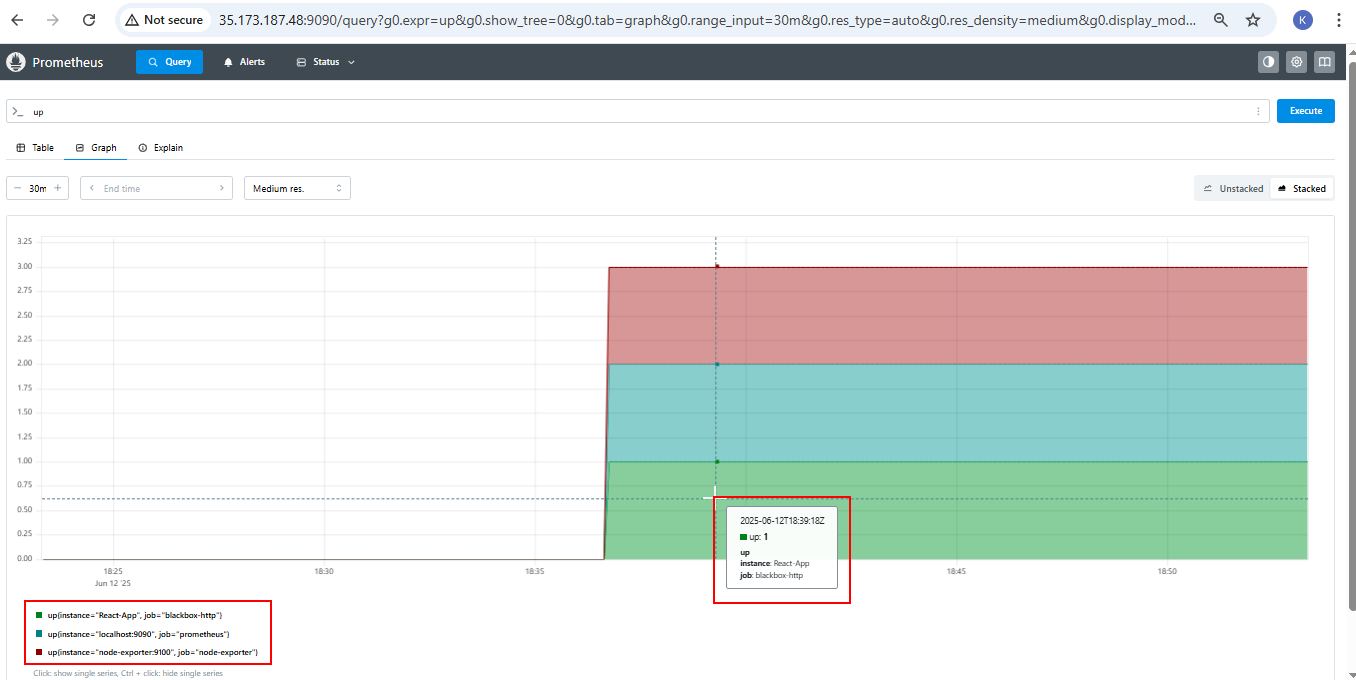


**Run Everything**

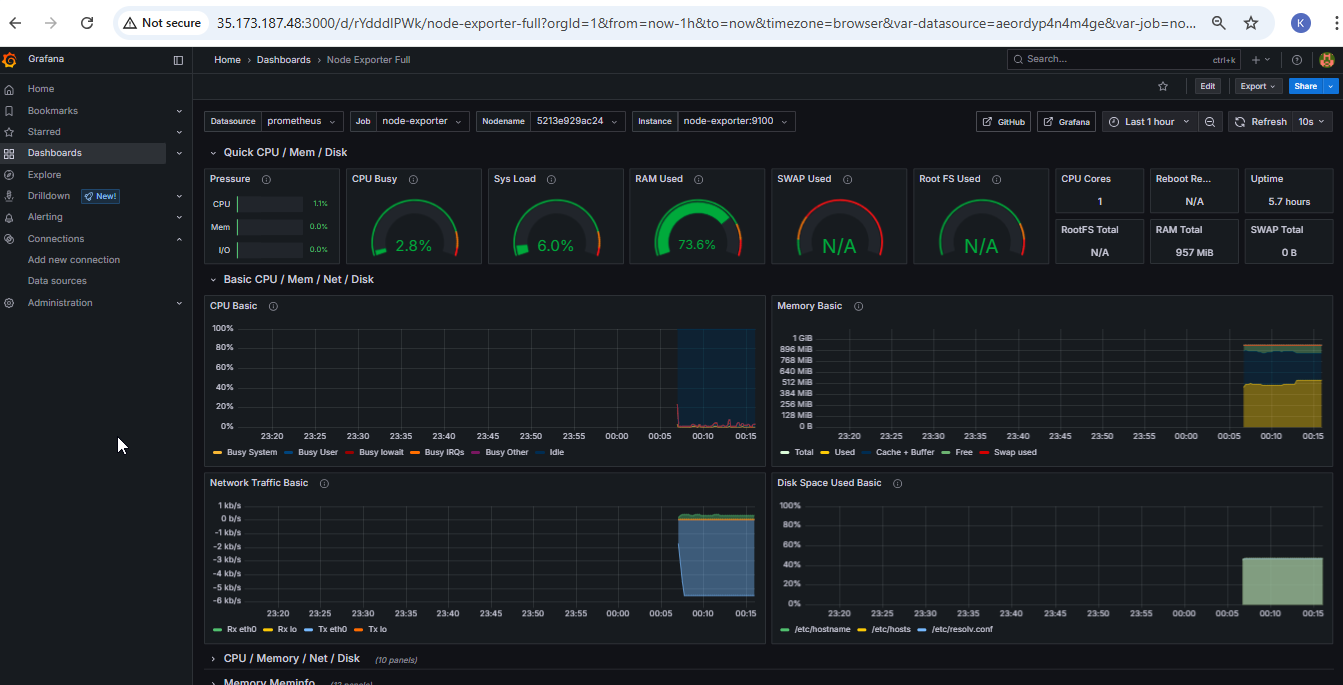
docker-compose up -d

**Containers Created with images:**

  
  
  
**Health Check in Prometheus Tools**

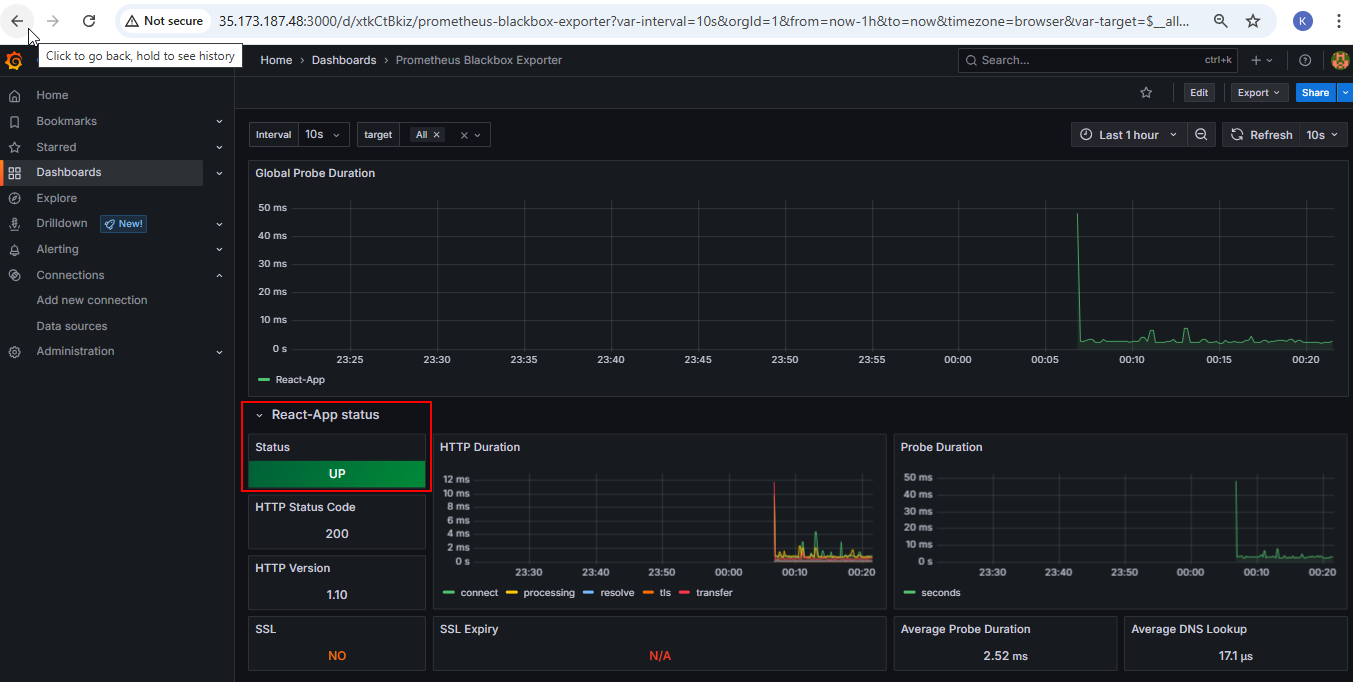


**All Metric Dashboard in Grafana:**

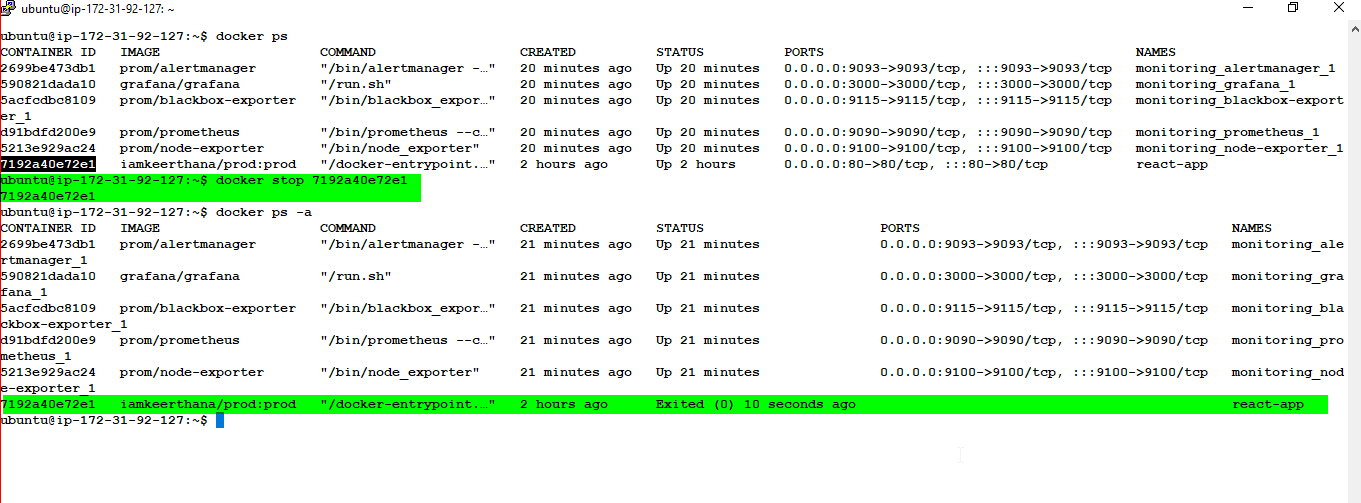
  
  
**Application health dashboard in Grafana:**

* Access with http://35.173.187.48:3000
* In Dashboard 🡪 Create→ Import
* In the Import page: Dashboard ID: 7587 🡪 Load
* Select Prometheus data source and import

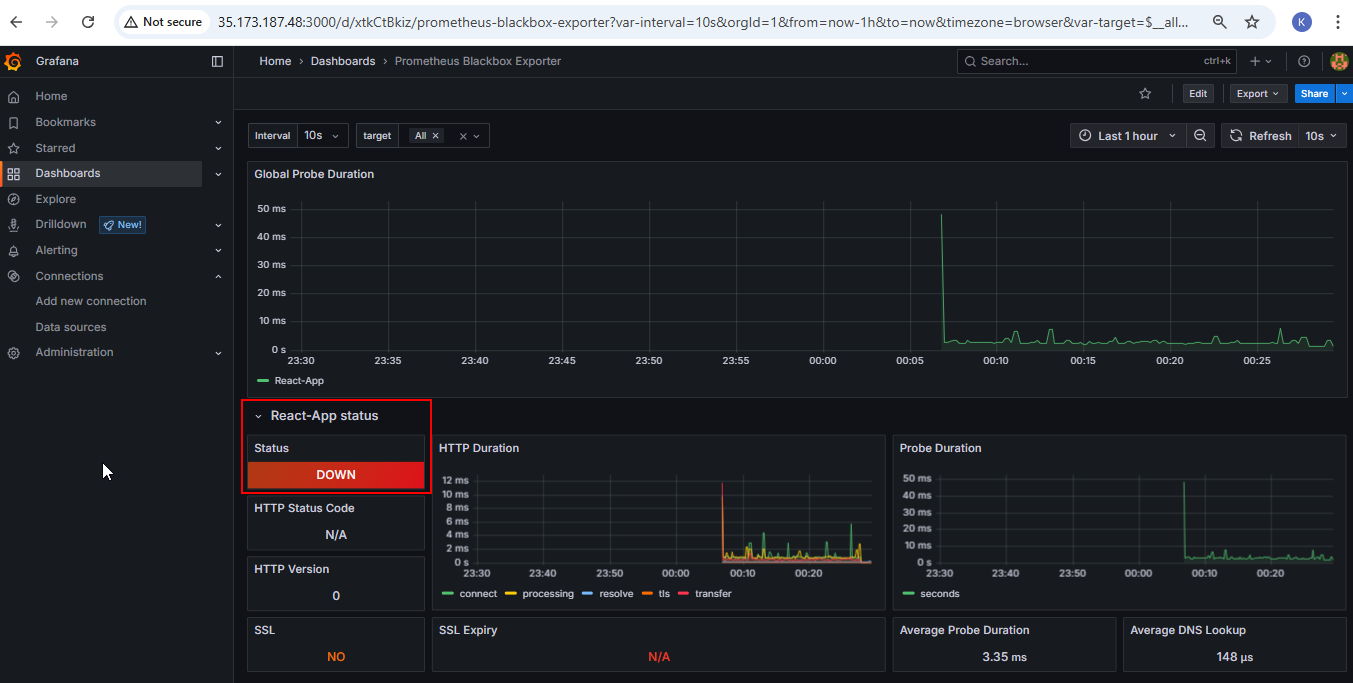
**React-Application Status – UP**



**Manually Stopped the React-App running container to check the Alert:**



**React-Application Status - DOWN**

  
  
  
**Email Alert received when Application container goes down:**

