Application Deployment

Deploy the React Application to a production ready state.

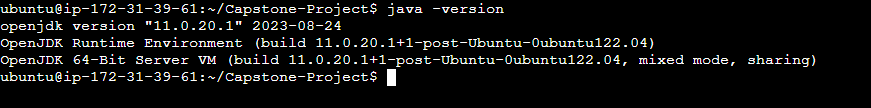
**Github link**: <https://github.com/shreegs123/Capstone-Project>

**Note:** I have tried to deploy the react application on kubernetes cluster. I have used Prometheus and grafana to monitor the ec2 instance on which docker and Jenkins was installed and not the cluster. I tried to monitor the cluster using helm but I could not achieve the desired result. Also the ip address of the machine has changed as I stopped and started the machine several time.

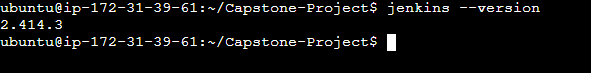
**Set up Docker-Kubernetes CI CD Jenkins pipeline on AWS step by step**

**Prerequisites**

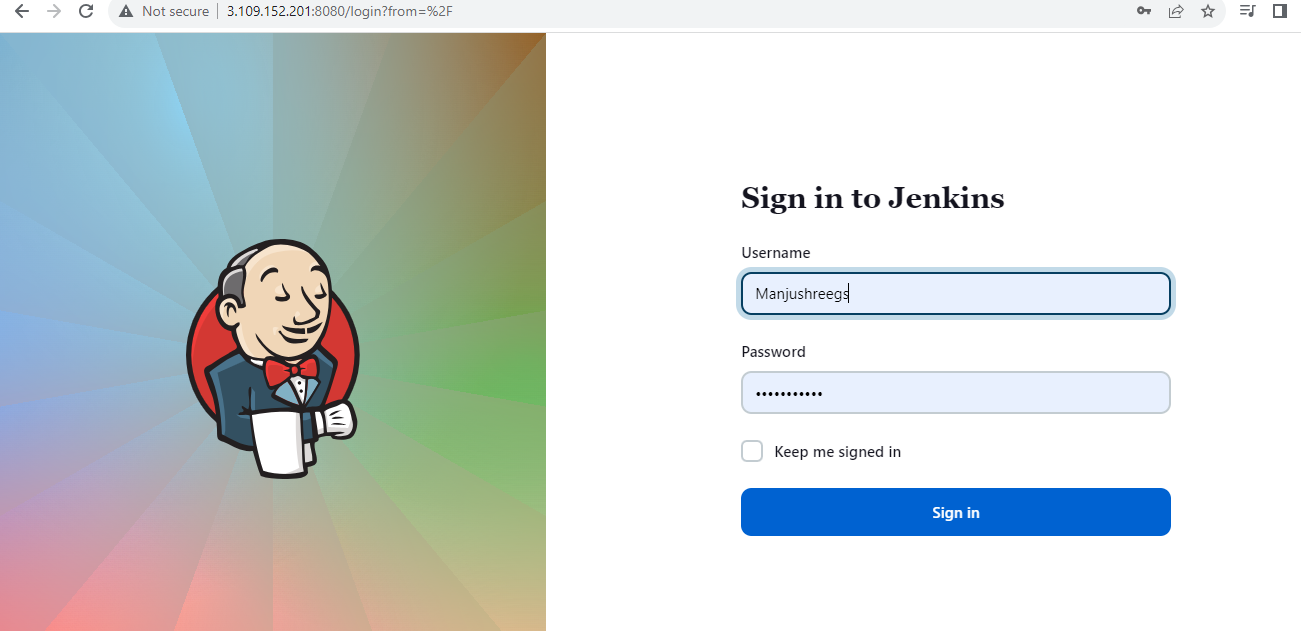
**Install Java**



**Install Jenkins**

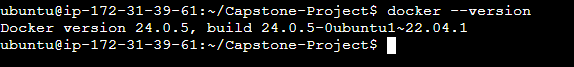


Jenkins login page

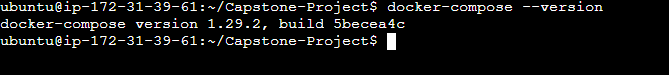


Now switch to the Jenkins user and install docker and docker-compose

**Install Docker**



**Install Docker-Compose**

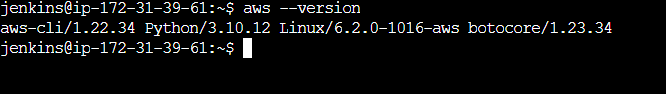


Add Jenkins user and Ubuntu machine to docker group

sudo usermod -aG docker Jenkins

sudo usermod -aG docker Ubuntu

**Set up AWS CLI on the EC2 machine to use *eksctl* .**



**Configure the AWS CLI so that it can authenticate and communicate with the AWS environment.**

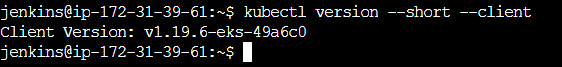
AWS Access Key ID [None]:

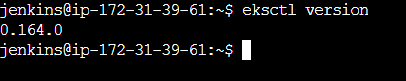
AWS Secret Access Key [None]:

Default region name [None]:

Default output format [None]:

**Install and Setup Kubectl and eksctl**

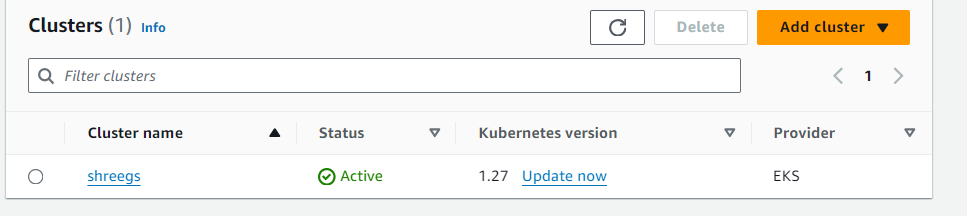




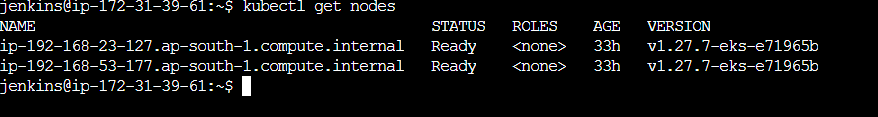
**Create EKS cluster using eksctl command**

eksctl create cluster --name shreegs --region ap-south-1 --node-type t2.micro

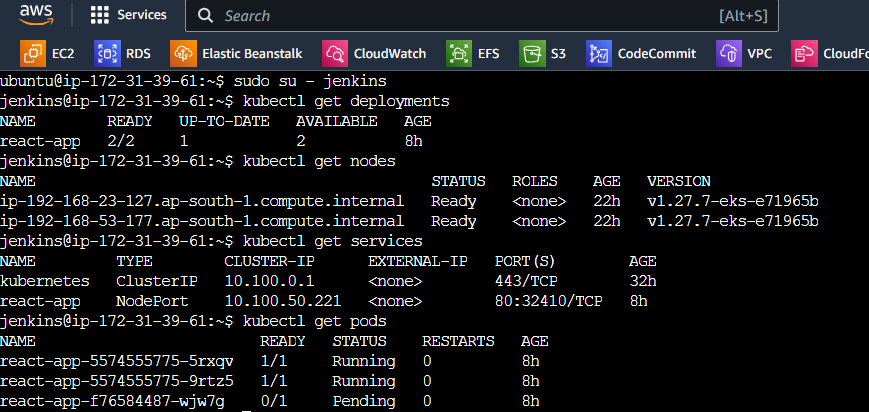
Verify the EKS kubernetes cluster from AWS



Verify the cluster nodes

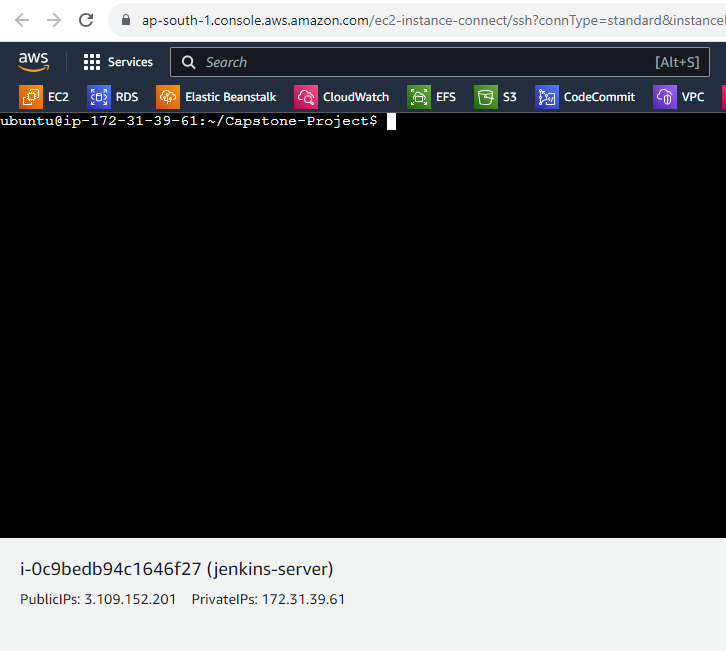


Verify the Kubernetes deployment and service

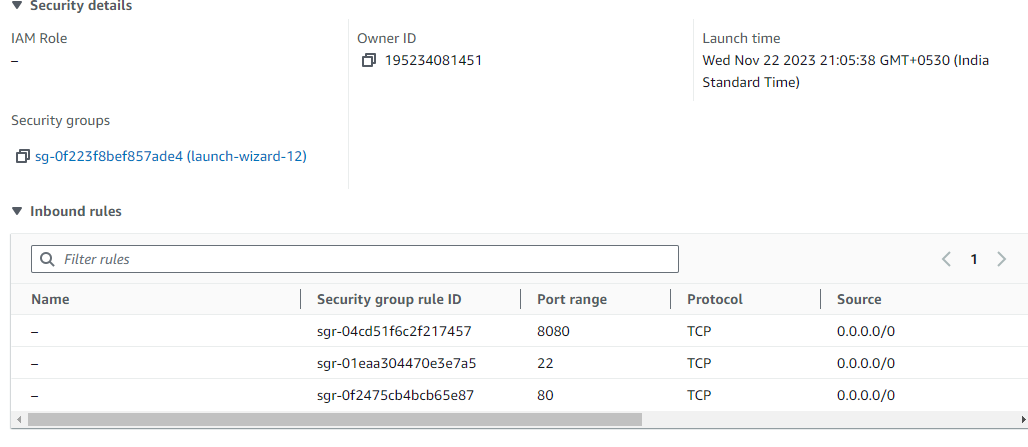


**Screenshots of Project**

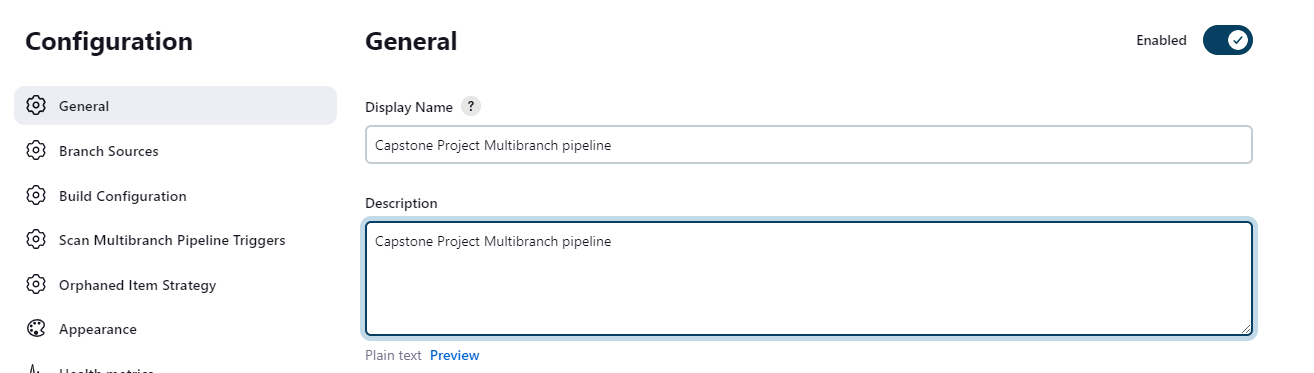
AWS console



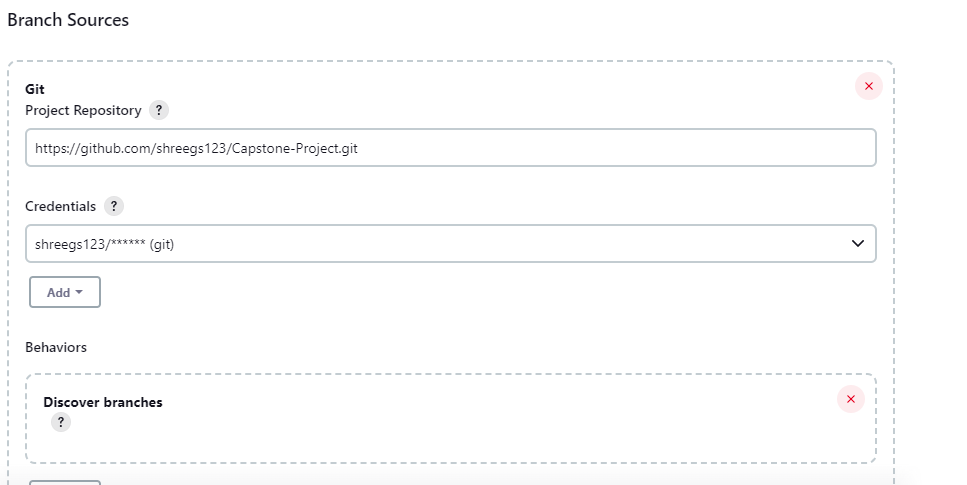
Security Group Configuration



Jenkins Configuration Setting

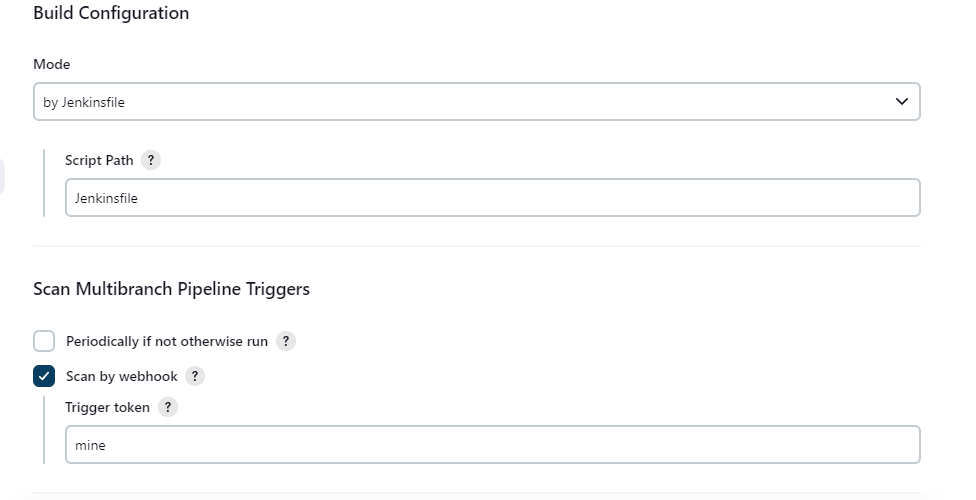


Github repo link as Source

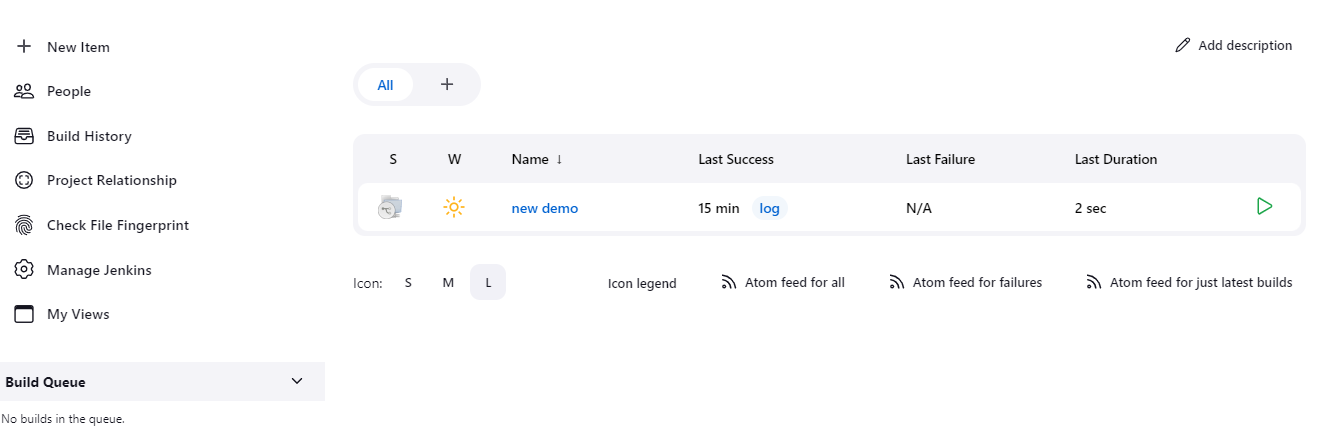


Jenkins file path and Webhook Configuration

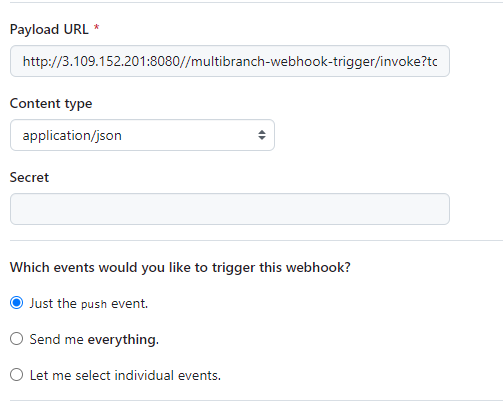
Apply and Save



Jenkins Multibranch Pipeline Job

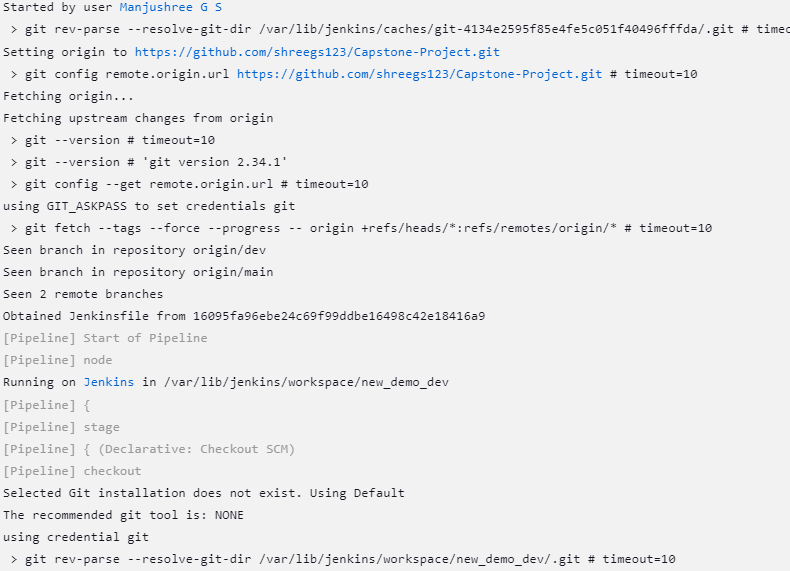


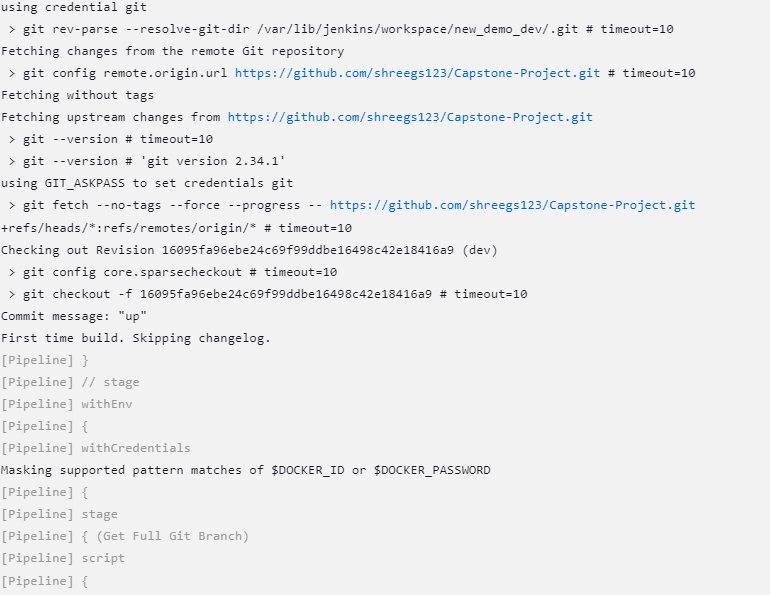
Github webhook configuration

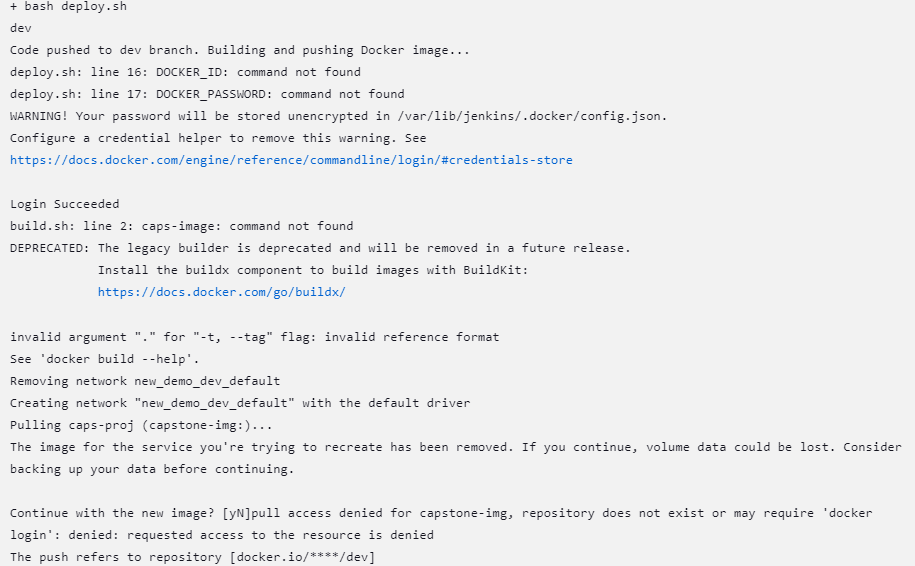


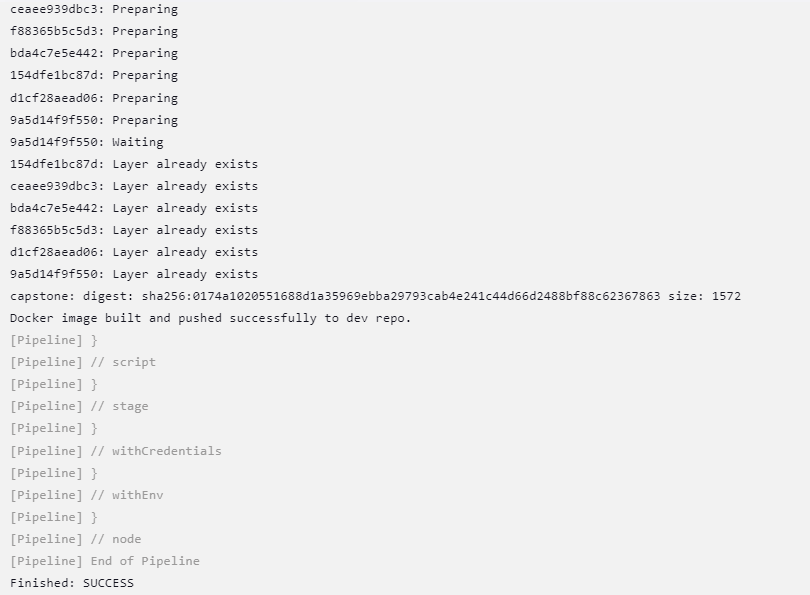
**Console Output 1**

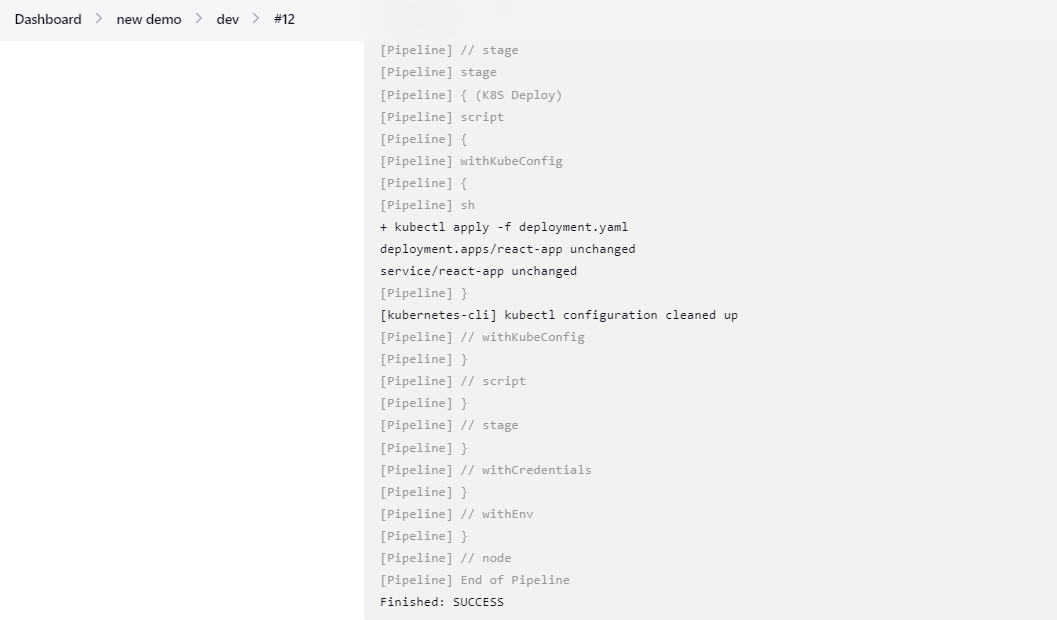
When the code is pushed to the dev branch on github then build and push the image to the public dev repo on dockerhub and also deployed on kubernetes cluster.

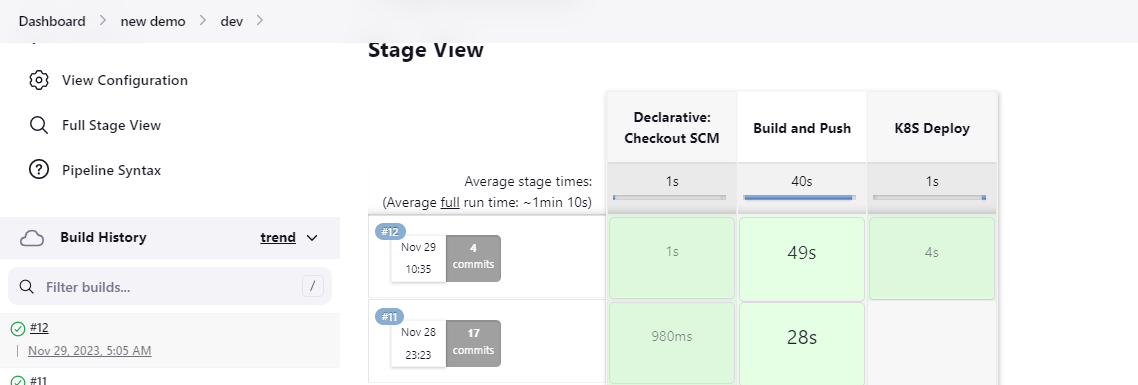






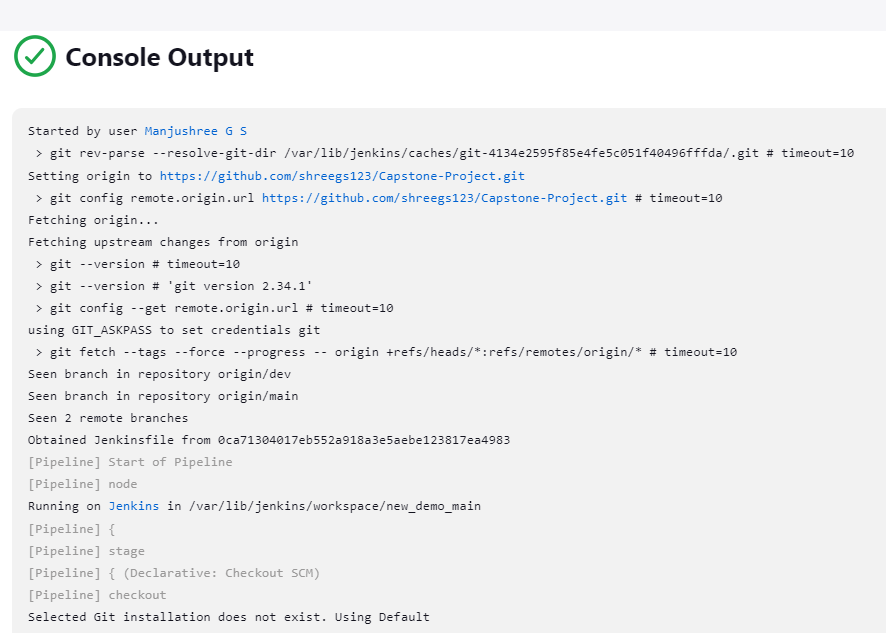


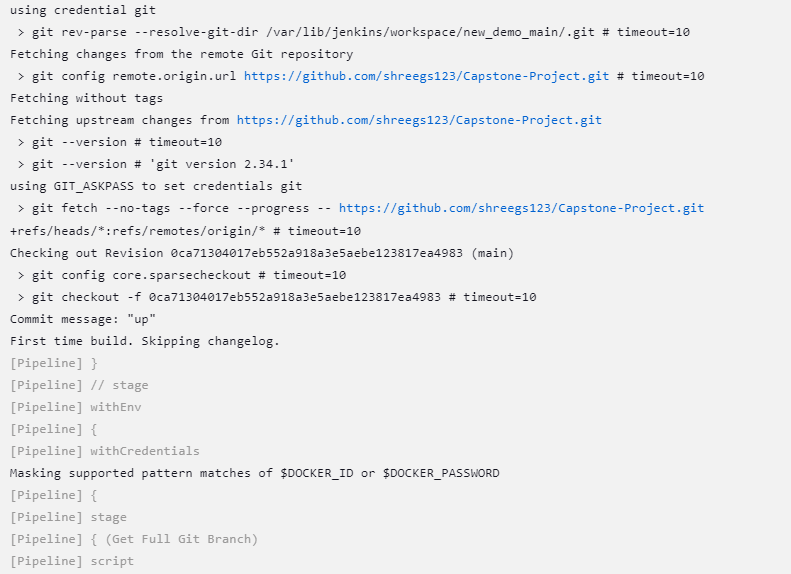


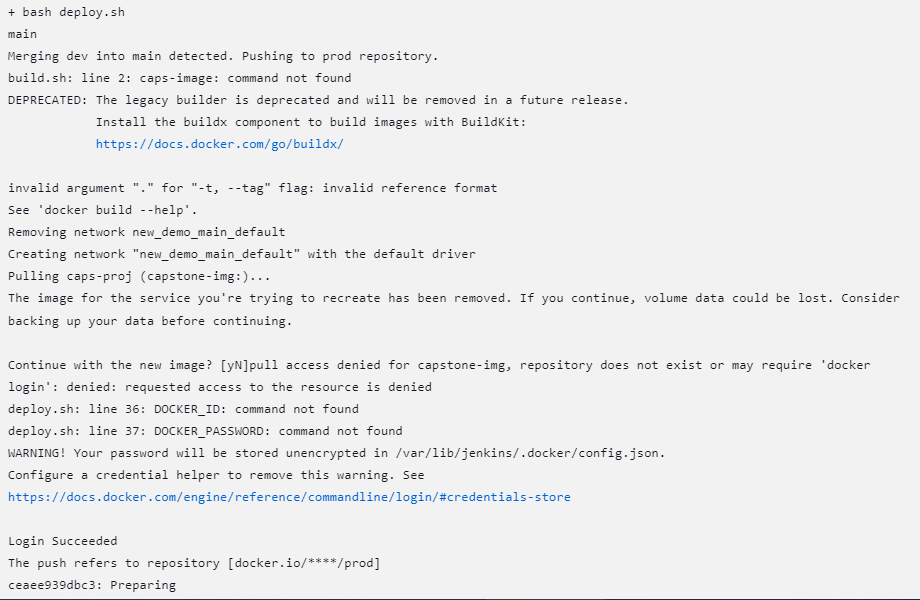


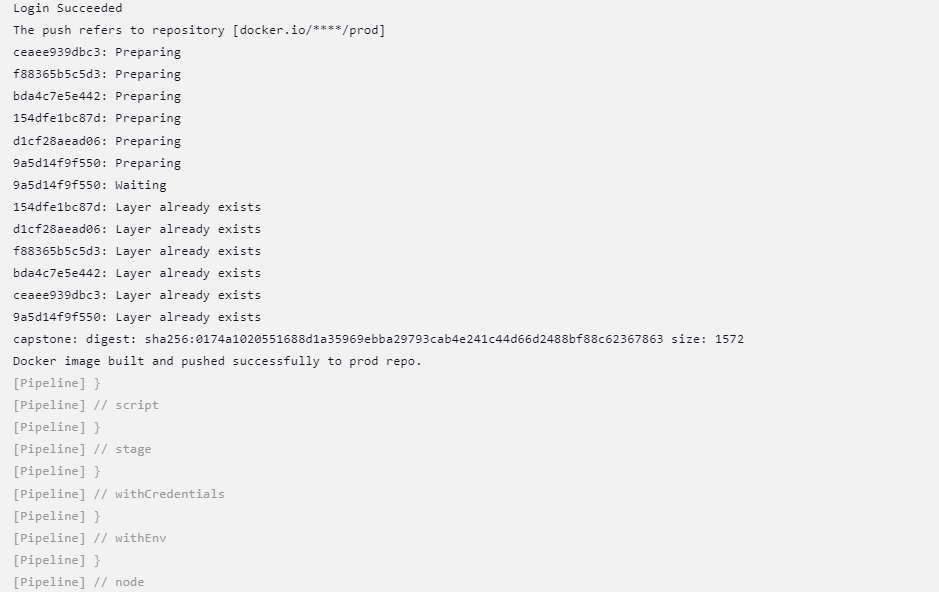
**Console Output 2**

When the code is merged from dev to main branch on github, then build and push the image to private prod repo on dockerhub, also deployed on kubernetes cluster.

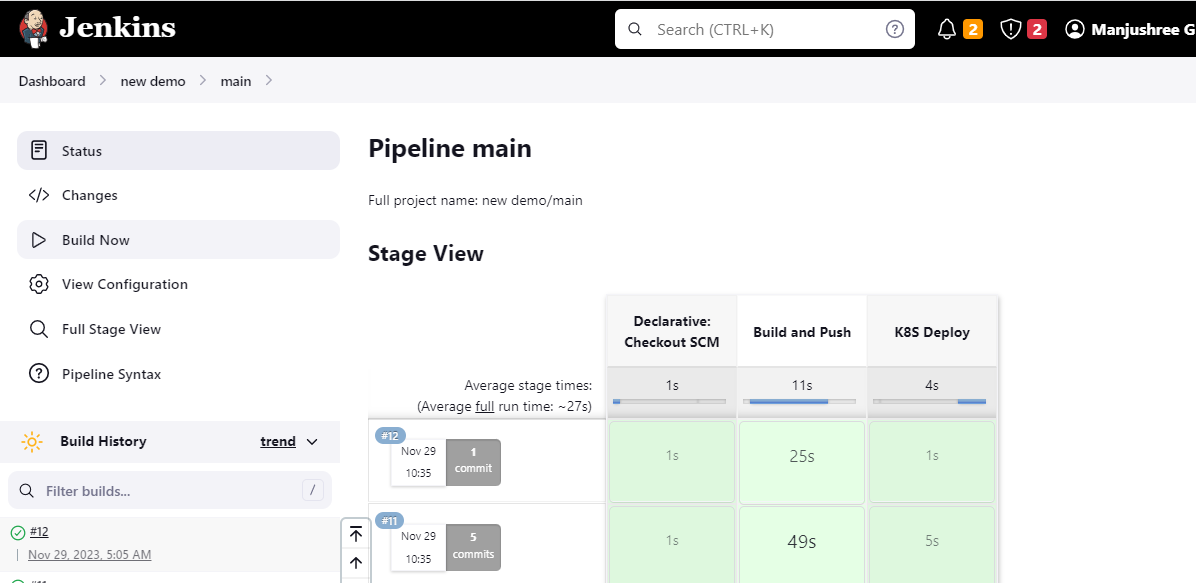






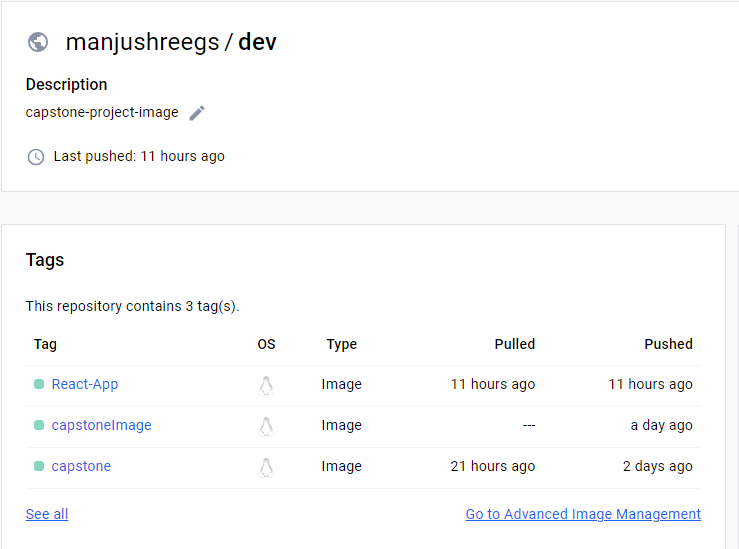


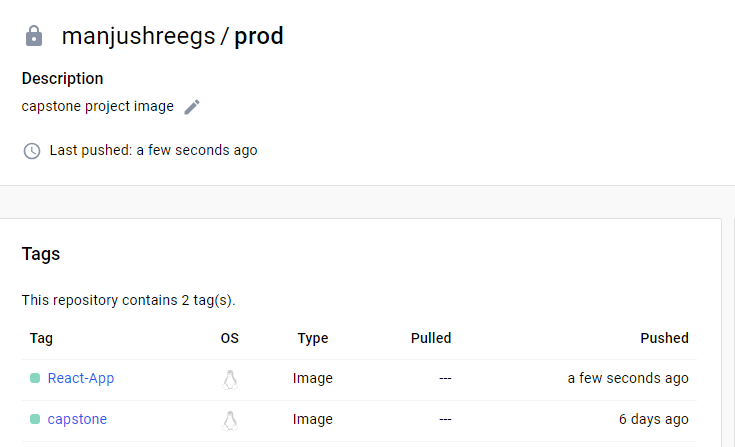




Docker hub Screenshoot

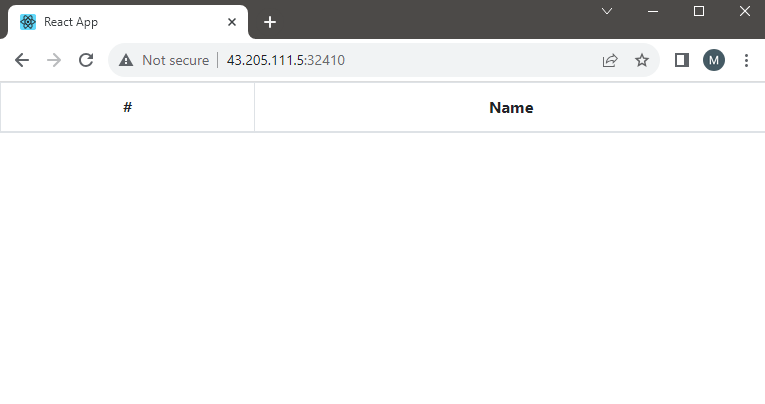
Dockerhub Public Repo : dev



Docker hub Private Repo: prod 

Output 3

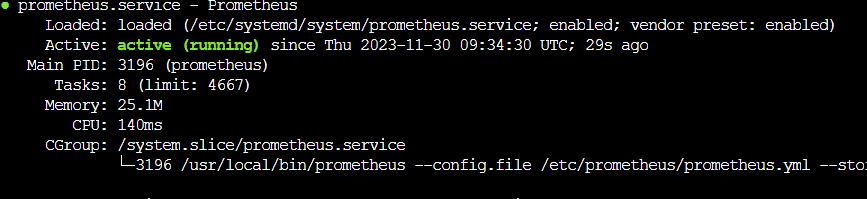
Application Access page

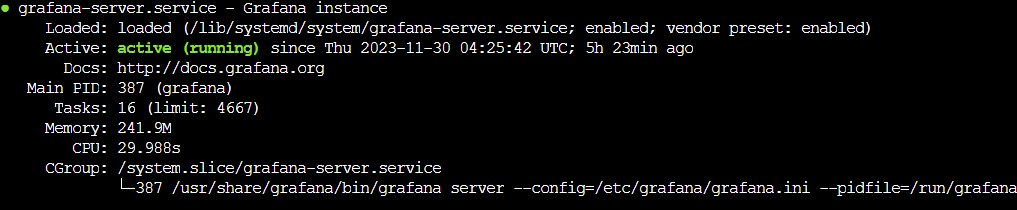


-------------------------------------------------------------------------------------

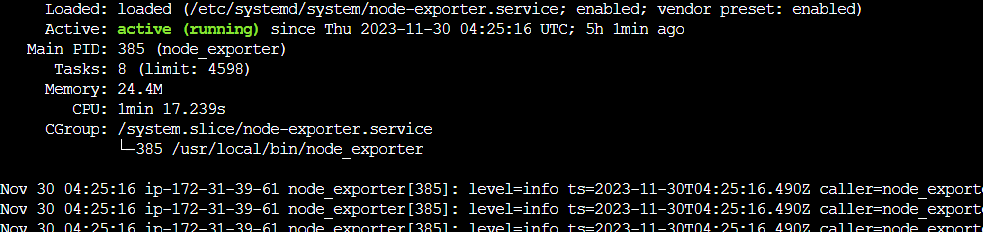
Monitoring the ec2 instance using Prometheus and grafana

Configure Prometheus and Grafana on different ec2 machine

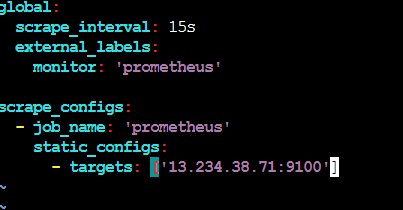




Configure Node-Exporter on the machine which has to be monitored



Configure Prometheus and add target machine ip address statically in the Prometheus.yaml file and restart Prometheus server



Grafana dashboard using node exporter dashboard

