**REALTIME FRONTEND APPLICATION**

**Step 1:** Install the git

$ sudo yum -y install <https://packages.endpoint.com/rhel/7/os/x86_64/endpoint-repo-1.7-1.x86_64.rpm>

$ sudo yum install git

**Step 2:** clone the url from gitlab

$ git clone (url)

**Step 3:** create Dockerfile

$ touch Dockerfile

$ vi Dockerfile

# pull the base image

FROM node:14-alpine as build

# set the working direction

WORKDIR /app

# copy package.json

COPY package.json ./

COPY package-lock.json ./

# install app dependencies

RUN npm install

# add app

COPY . ./

RUN npm run build

FROM nginx:stable-alpine

COPY --from=build /app/build /usr/share/nginx/html

EXPOSE 80

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

**Step 4:** create docker ignore file

$ touch .dockerignore

$ vi .dockerignore

node\_modules

build

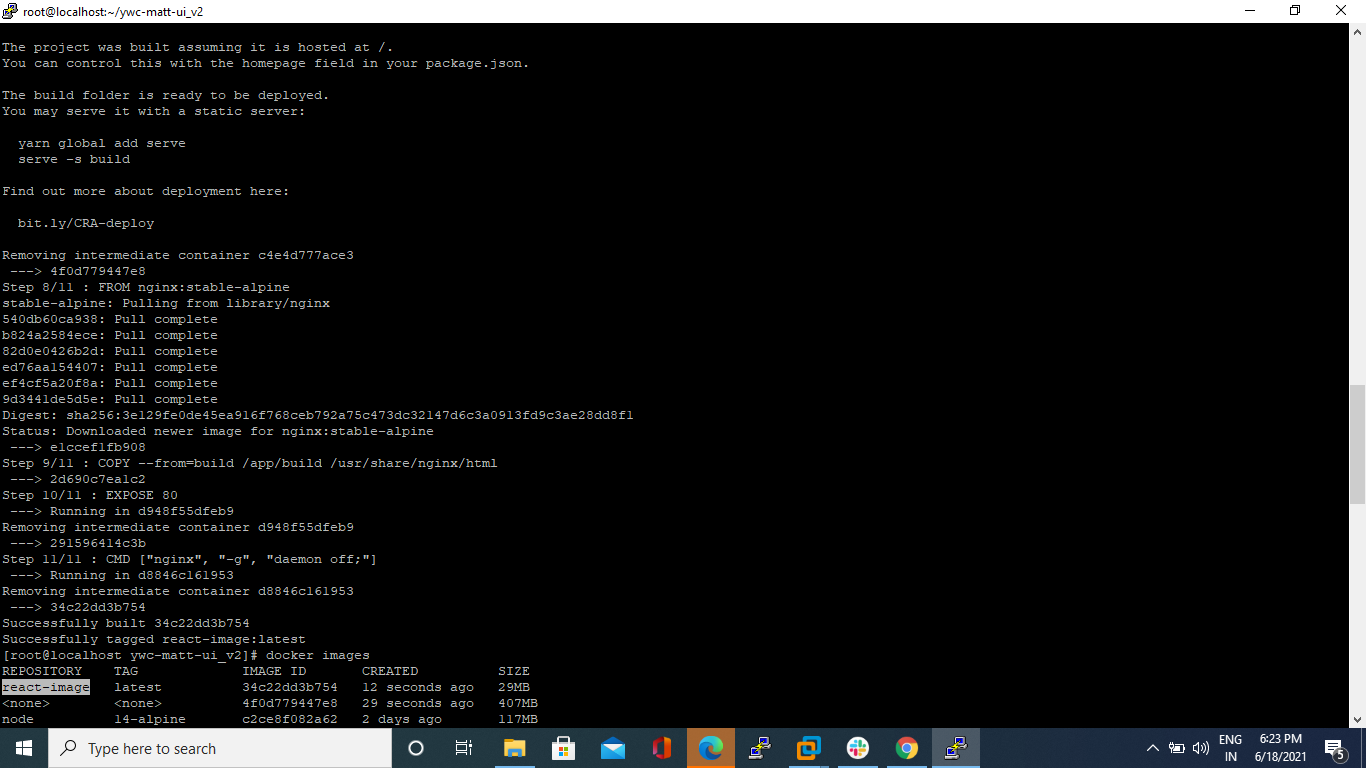
.dockerignore

Dockerfile

Dockerfile.prod

**Step 5:** create docker iamge

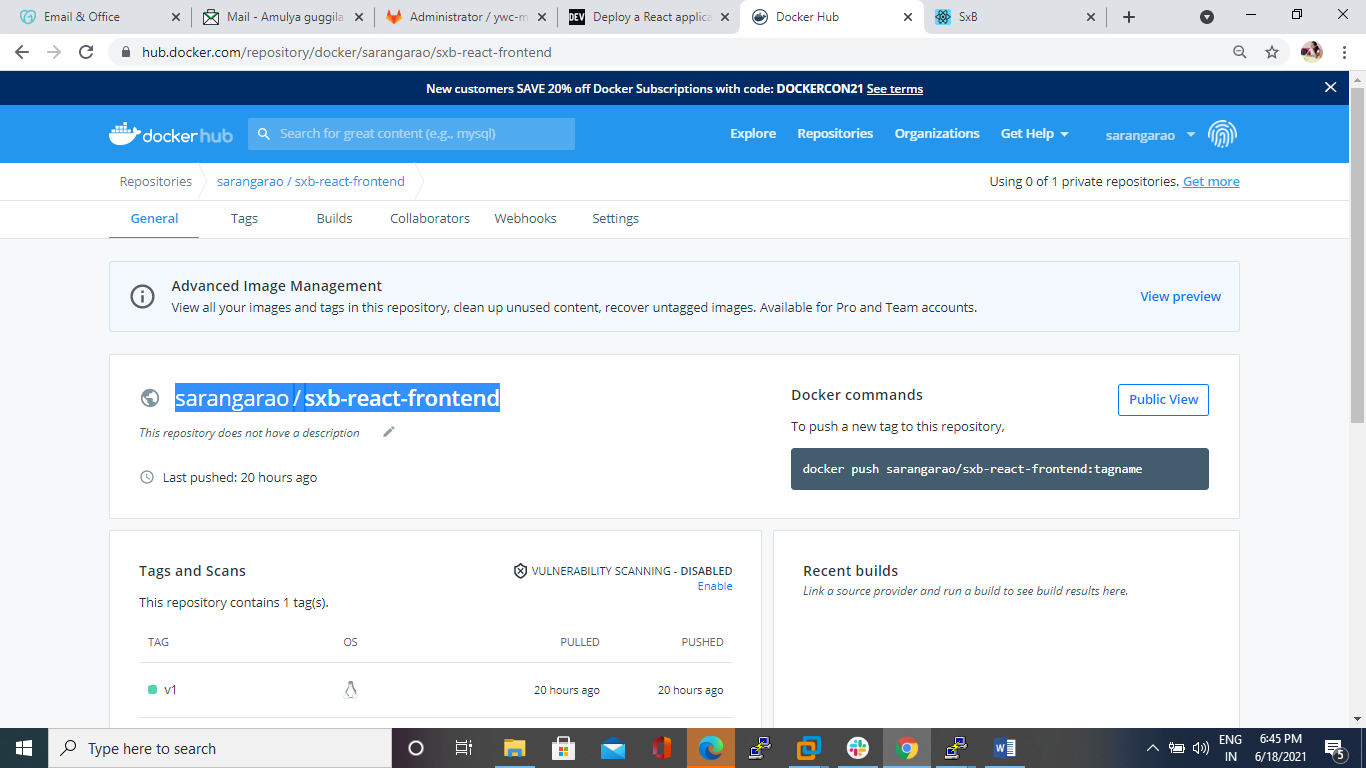
$ docker build –t react-image .



**Step 6:** push the docker image to docker hub

$ docker tag react-image dockerhub ID/new folder name

$ docker push dockerhub ID/new folder name



**Step 7:** create directory

$ mkdir Deployments

$ cd Deployments

**Step 8:** create a file in Deployments folder

$ touch deployment.yaml

$ vi deployment.yaml

kind: Deployment

apiVersion: apps/v1

metadata:

name: my-react-app

spec:

replicas: 2

selector:

matchLabels:

app: my-react-app

template:

metadata:

labels:

app: my-react-app

spec:

containers:

- name: my-react-app

image: sarangarao/sxb-react-frontend:v1

imagePullPolicy: Always

ports:

- containerPort: 80

restartPolicy: Always

---

kind: Service

apiVersion: v1

metadata:

name: my-react-app

spec:

type: NodePort

ports:

- port: 80

targetPort: 80

protocol: TCP

nodePort: 31000

selector:

app: my-react-app

**step 8:** apply in kubernetes

$ kubectl apply –f deployment.yaml

**Step 9:** check for the pods

$ kubectl get pods

Access in the browser with ip

