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

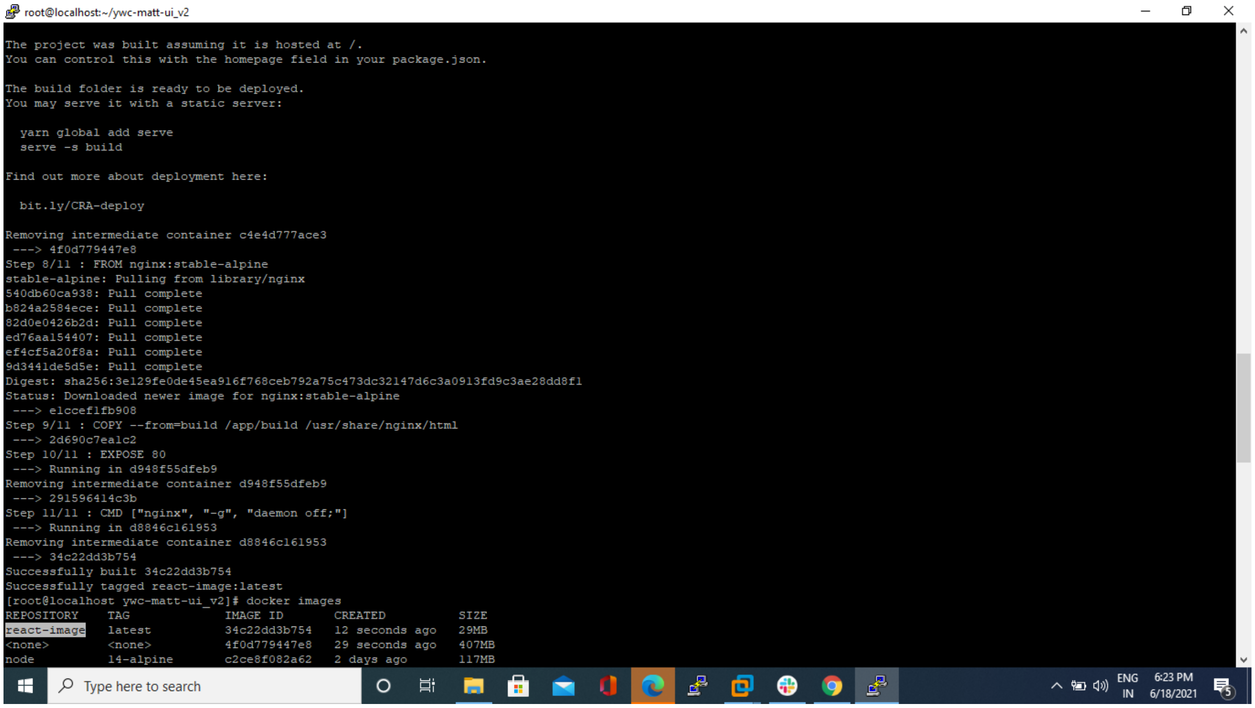
.dockerignore

Dockerfile

Dockerfile.prod

Step 5: create docker iamge

$ docker build –t react-image .

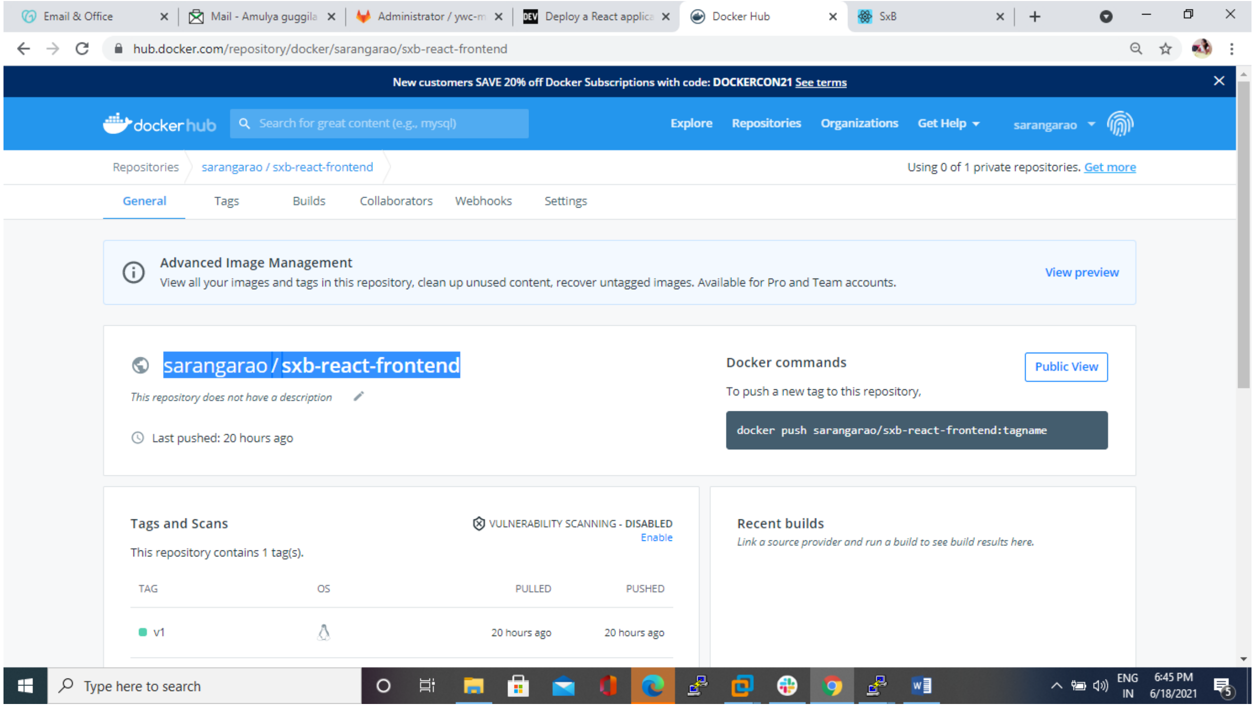


```
The project was built assuming it is hosted at /.
You can control this with the homepage field in your package.json.

The build folder is ready to be deployed.
You may serve it with a static server:

  yarn global add serve
  serve -s build

Find out more about deployment here:

  bit.ly/CRA-deploy

Removing intermediate container c4e4d777ace3
 ---> 4f0d779447e8
Step 8/11 : FROM nginx:stable-alpine
stable-alpine: Pulling from library/nginx
540db60ca938: Pull complete
b824a2584ece: Pull complete
82d0e0426b2d: Pull complete
ed76aa154407: Pull complete
ef4cf5a20f8a: Pull complete
9d3441de5d5e: Pull complete
Digest: sha256:3e129fe0de45ea916f768ceb792a75c473dc32147d6c3a0913fd9c3ae28dd8f1
Status: Downloaded newer image for nginx:stable-alpine
 ---> e1ccef1fb908
Step 9/11 : COPY --from=build /app/build /usr/share/nginx/html
 ---> 2d690c7ea1c2
Step 10/11 : EXPOSE 80
 ---> Running in d948f55dfeb9
Removing intermediate container d948f55dfeb9
 ---> 291596414c3b
Step 11/11 : CMD ["nginx", "-g", "daemon off;"]
 ---> Running in d8846c161953
Removing intermediate container d8846c161953
 ---> 34c22dd3b754
Successfully built 34c22dd3b754
Successfully tagged react-image:latest
[root@localhost ywc-matt-ui_v2]# docker images
REPOSITORY    TAG            IMAGE ID       CREATED          SIZE
react-image   latest         34c22dd3b754   12 seconds ago   29MB
<none>        <none>         4f0d779447e8   29 seconds ago   407MB
node          14-alpine      c2ce8f082a62   2 days ago       117MB
```

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



New customers SAVE 20% off Docker Subscriptions with code: DOCKERCON21 See terms

Repositories / sarangarao / sxb-react-frontend

Using 0 of 1 private repositories. Get more

Advanced Image Management
View all your images and tags in this repository, clean up unused content, recover untagged images. Available for Pro and Team accounts.

sarangarao / sxb-react-frontend

This repository does not have a description

Last pushed: 20 hours ago

Docker commands

To push a new tag to this repository,

```
docker push sarangarao/sxb-react-frontend:tagname
```

Tags and Scans

VULNERABILITY SCANNING - DISABLED

This repository contains 1 tag(s).

TAG	OS	PULLED	PUSHED
v1		20 hours ago	20 hours ago

Recent builds

Link a source provider and run a build to see build results here.

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
```

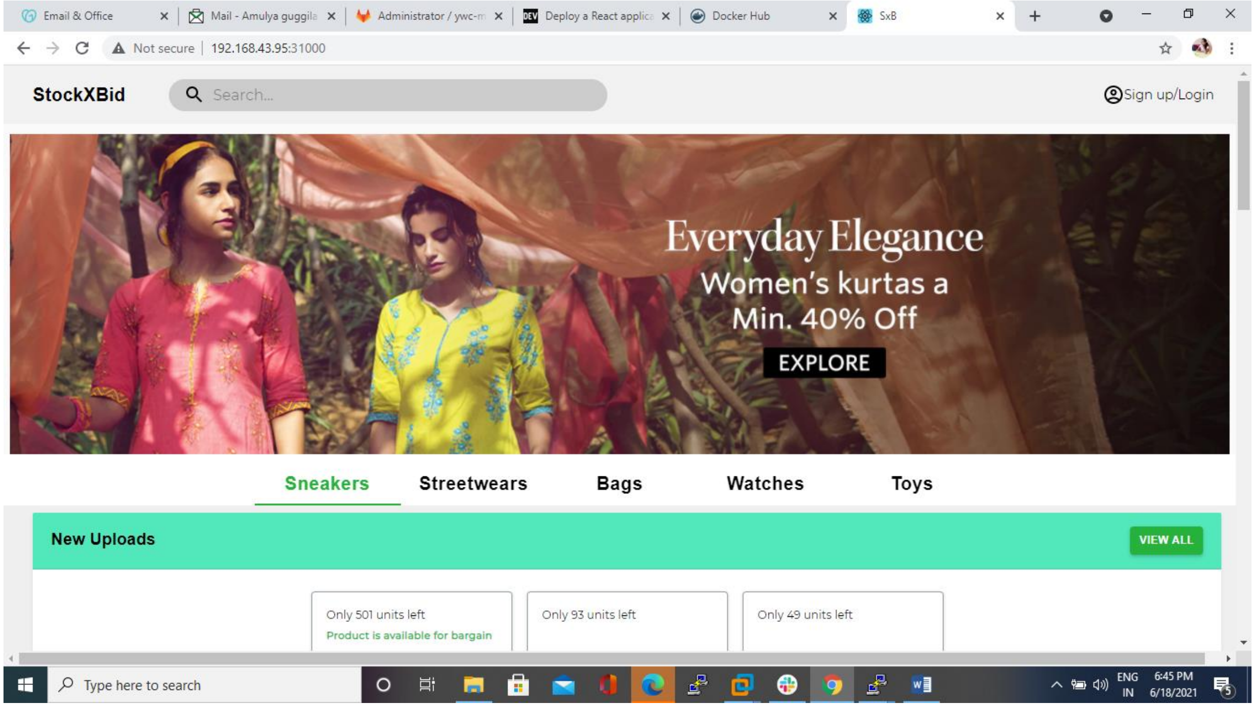
step 8: apply in kubernetes

```
$ kubectl apply –f deployment.yaml
```

Step 9: check for the pods

```
$ kubectl get pods
```

Access in the browser with ip


Email & Office
Mail - Amulya guggila
Administrator / ywc-m
Deploy a React applic
Docker Hub
SxB
Not secure
192.168.43.95:31000
StockXBid
Search...
Sign up/Login
Everyday Elegance
Women's kurtas a
Min. 40% Off
EXPLORE
Sneakers
Streetwears
Bags
Watches
Toys
New Uploads
VIEW ALL
Only 501 units left
Product is available for bargain
Only 93 units left
Only 49 units left
Type here to search
ENG
IN
6:45 PM
6/18/2021