

--Pull image existante de node--
docker pull node

--J'ai crée mon Dockerfile et je la build--
docker build -t thaldarim/react-web-app .

--je démarre mon container--
docker run --name my-react-app -d -p 8989:3000 thaldarim/react-web-app

--j'efface mon container--
docker stop my-react-app
docker rm my-react-app

--je push mon image--
docker login
docker push thaldarim/react-web-app

--lien du repo--
<https://hub.docker.com/repository/docker/thaldarim/react-web-app>

Kubernetes

gcloud container clusters get-credentials tp-docker --zone europe-west1-b --project
durable-tracer-296115
-- récupérer les infos du cluster et du projet --

gcloud config set project durable-tracer-296115
-- Lier le projet --

gcloud container clusters list
-- lister les containers --

kubectl get all --all-namespaces
-- infos de config diverses --

kubectl apply -f deployment.yml
-- appliquer le délpoiement --

kubectl get deployments
-- voir les déploiements en cours --

kubectl get pod
-- lister les pods --

kubectl port-forward app-deployment-6567d9dcfc-69wfz 8989:3000
-- se connecter au port du pod --

kubectl apply -f service.yml

-- appliquer le service --

kubectl get service

-- lister les services --

kubectl apply -f ingress.yml

-- appliquer l'ingress --

```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: service-tp-docker
5    namespace: default
6  spec:
7    ports:
8      - port: 8989
9        protocol: TCP
10       targetPort: 3000
11    selector:
12      run: service-tp-docker
13    type: NodePort
```

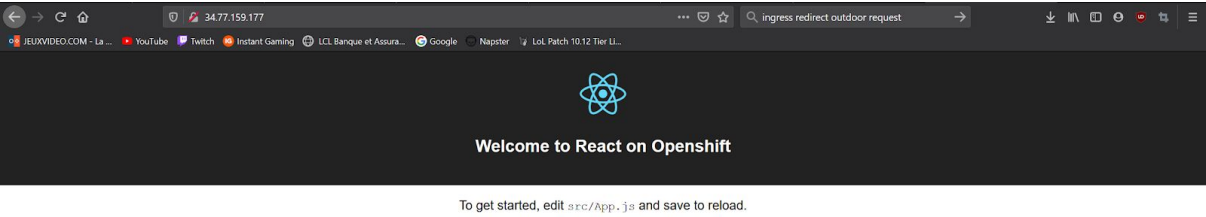
Version du service.yaml utilisée jusque là puis remplacée par celle-ci :

```
1  apiVersion: v1
2  kind: Service
3  metadata:
4    name: service-tp-docker
5    namespace: default
6  spec:
7    ports:
8      - port: 8989
9        protocol: TCP
10       targetPort: 3000
11    selector:
12      run: service-tp-docker
13    type: NodePort
14
15  apiVersion: v1
16  kind: Service
17  metadata:
18    name: service-tp-docker
19  spec:
20    selector:
21      pod: react
22    ports:
23      - protocol: TCP
24        port: 80
25        targetPort: 3000
26    clusterIP: 10.124.3.25
27    type: LoadBalancer
```

kubectl apply -f service.yaml

kubectl get service

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.124.0.1	<none>	443/TCP	92m
service-tp-docker	LoadBalancer	10.124.3.25	34.77.159.177	80:32193/TCP	58m



kubectl apply -f hpa.yml

NAME	REFERENCE	TARGETS	MINPODS	MAXPODS	REPLICAS	AGE
hpa	Deployment/app-deployment	0%/65%	3	50	3	43s