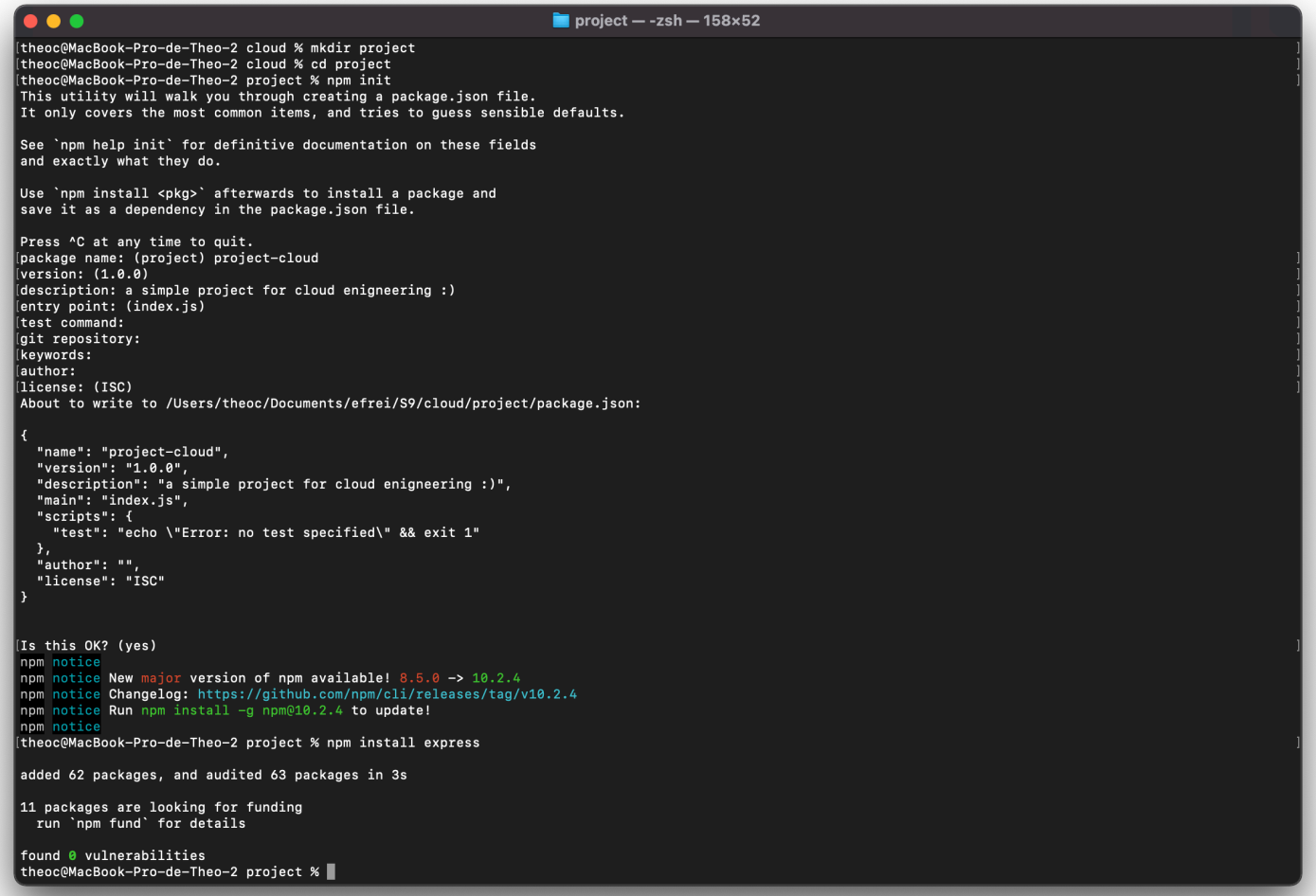
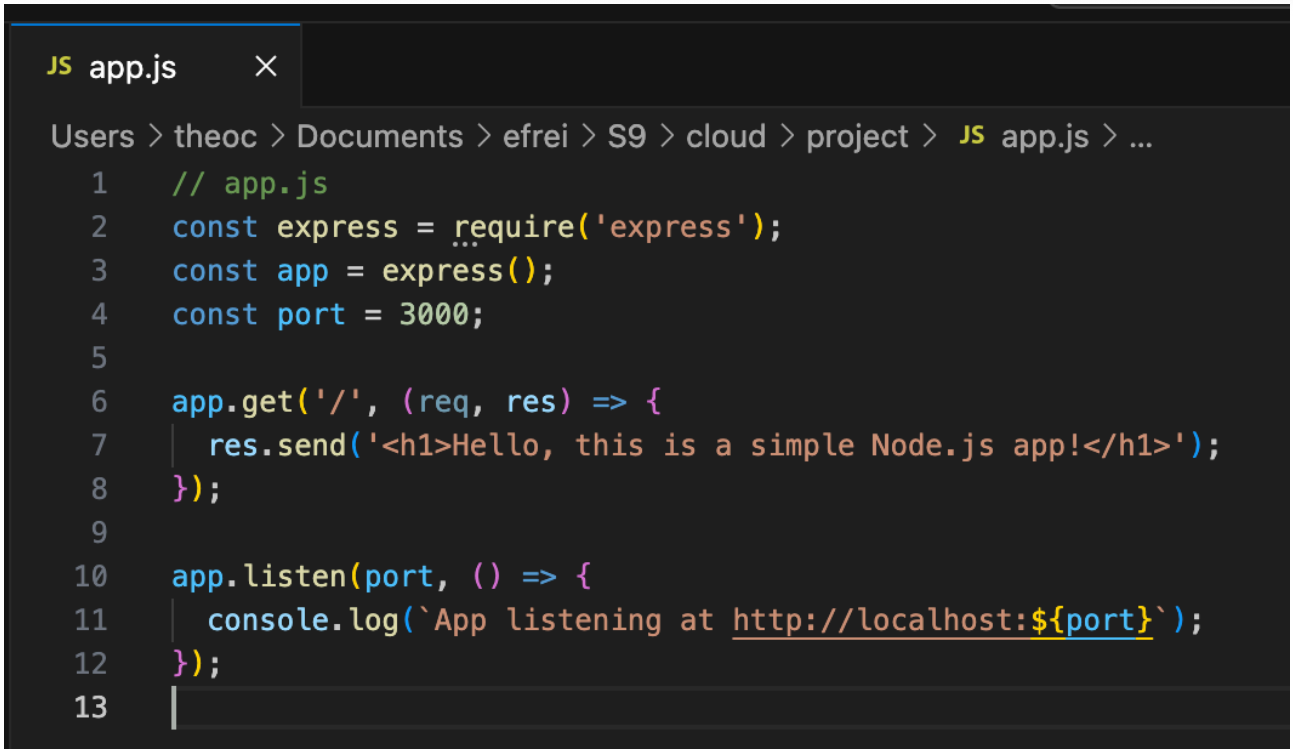
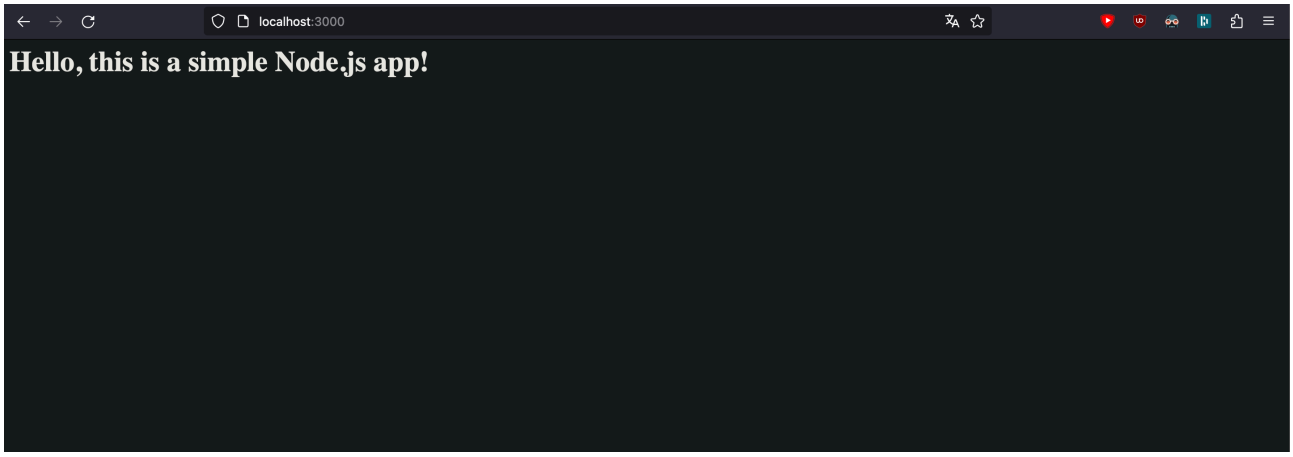
CANARIO Théo

Software engineering for the cloud

1. Creation de la mini app (powered by ChatGPT)







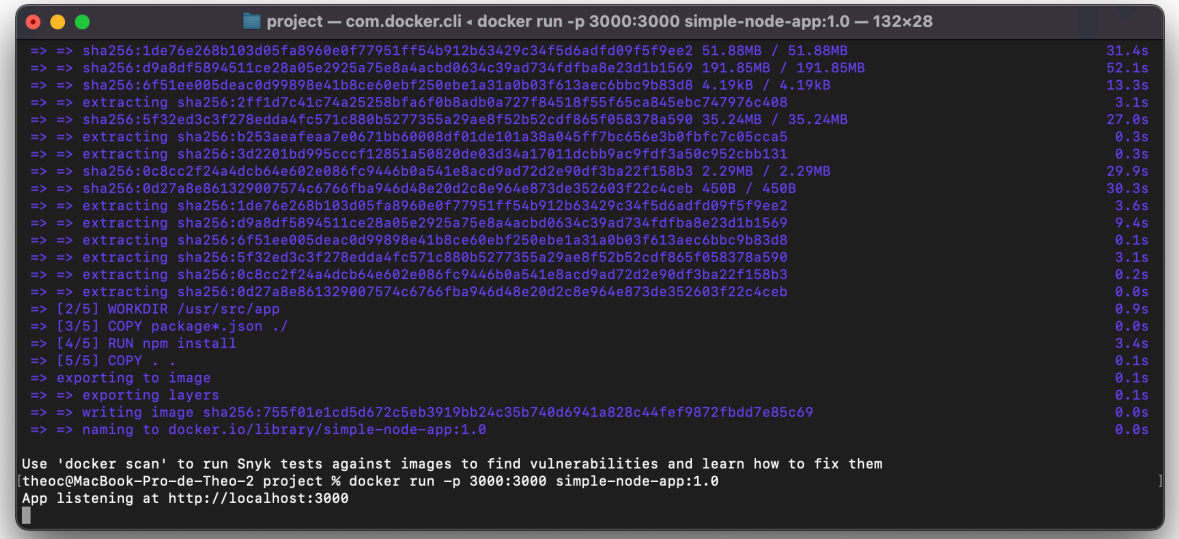
2. Creation of the docker file

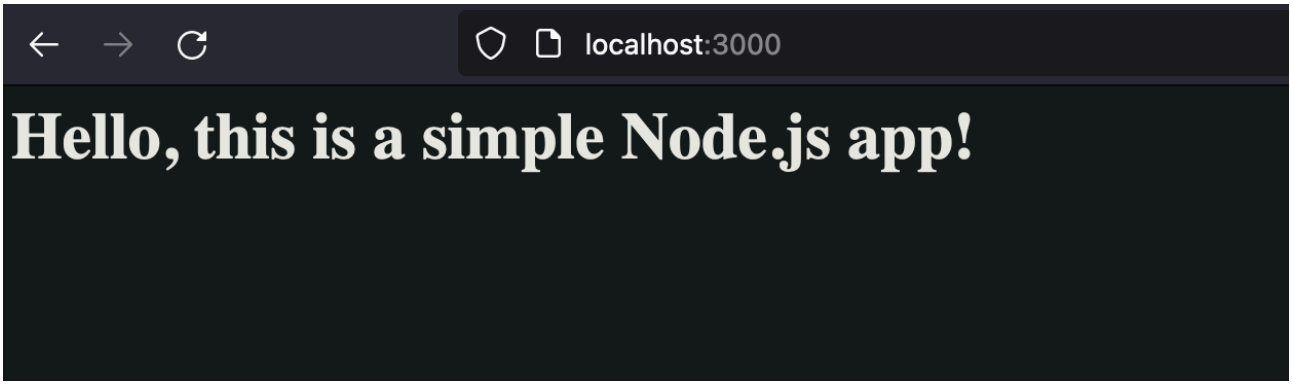
app.js Dockerfile

Users > theoc > Documents > efrei > S9 > cloud > project > Dockerfile > ...

```
# Use the official Node.js image as the base image
FROM node:14

# Set the working directory in the container
WORKDIR /usr/src/app

# Copy package.json and package-lock.json to the container
COPY package*.json ./

# Install app dependencies
RUN npm install

# Copy the rest of the application code to the container
COPY . .

# Expose the port on which the app will run
EXPOSE 3000

# Command to run the application
CMD ["node", "app.js"]
```

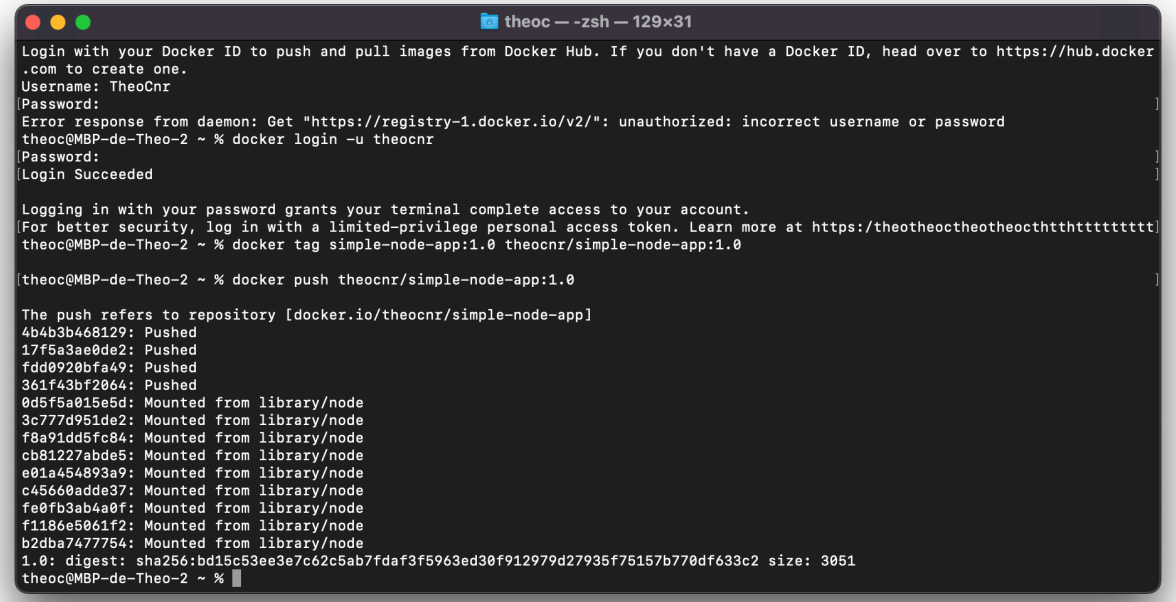
3. Build docker app : docker build -t simple-node-app:1.0 .
4. Run docker app : docker run -p 3000:3000 simple-node-app:1.0

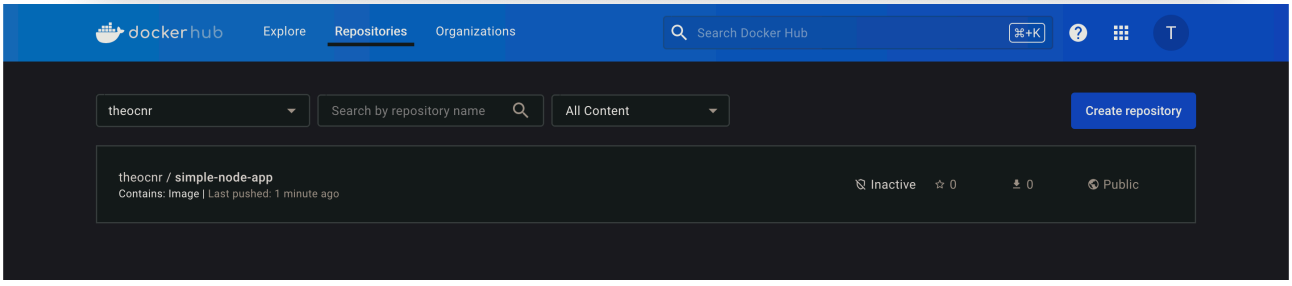




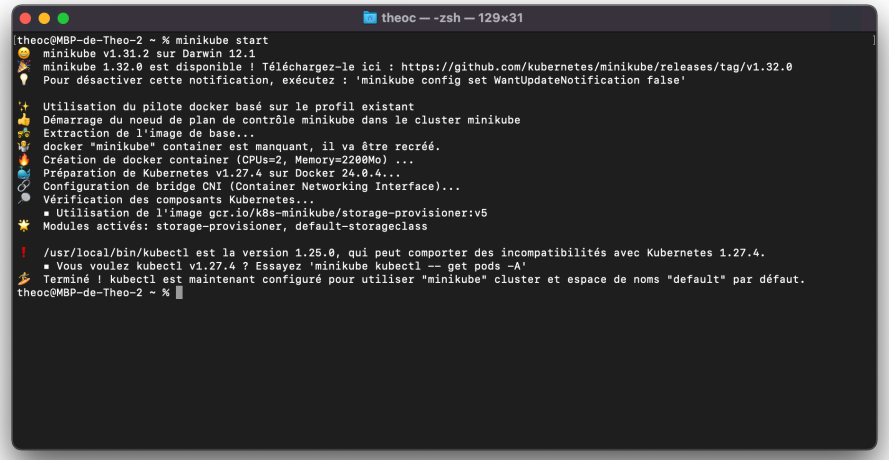
— —

5. Publish the docker image to the docker hub



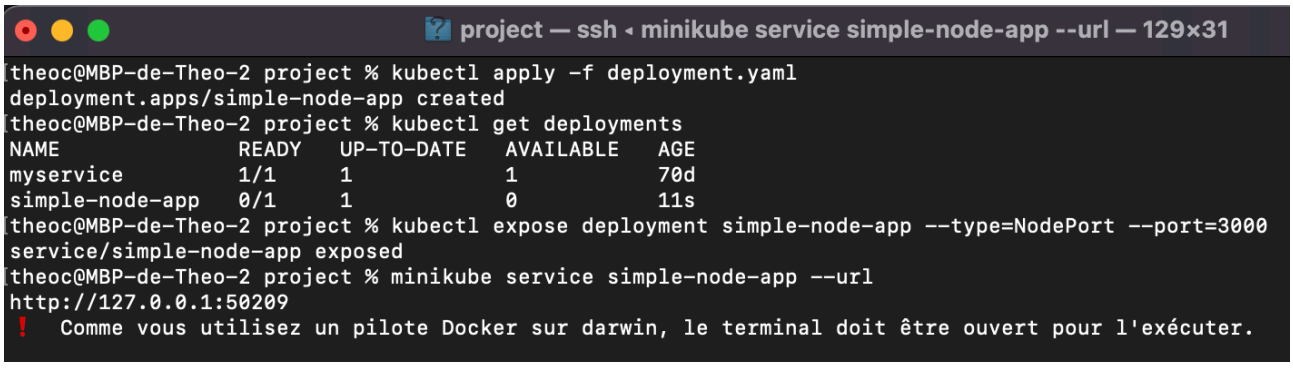


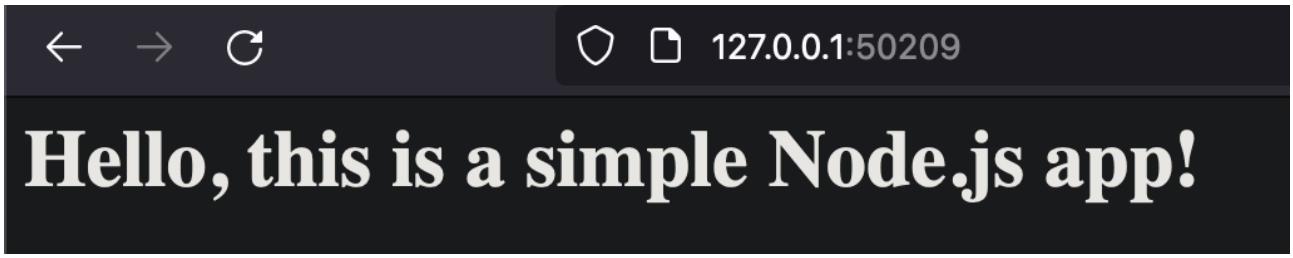
6. Create a kubernetes deployment



deployment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: simple-node-app
spec:
  replicas: 1
  selector:
    matchLabels:
      app: simple-node-app
  template:
    metadata:
      labels:
        app: simple-node-app
    spec:
      containers:
        - name: simple-node-app
          image: theocnr/simple-node-app:1.0
          ports:
            - containerPort: 3000
```



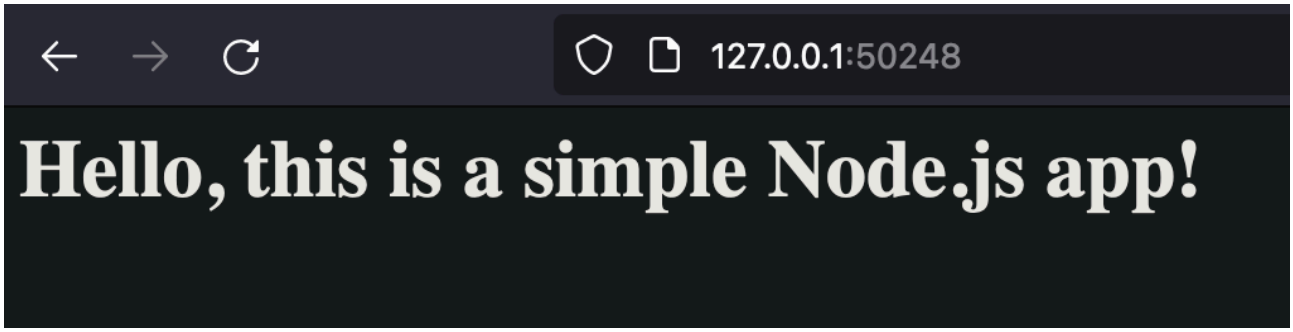


7.Create kubernetes service

service.yaml

```yaml
apiVersion: v1
kind: Service
metadata:
  name: simple-node-app-service
spec:
  selector:
    app: simple-node-app
  ports:
    - protocol: TCP
      port: 80
      targetPort: 3000
  type: LoadBalancer
```

```
theoc@MBP-de-Theo-2 project % kubectl apply -f service.yaml
service/simple-node-app-service created
theoc@MBP-de-Theo-2 project % kubectl get services
NAME                     TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
kubernetes               ClusterIP      10.96.0.1       <none>        443/TCP          70d
simple-node-app          NodePort       10.110.153.17   <none>        3000:31637/TCP   7m55s
simple-node-app-service  LoadBalancer   10.96.121.199   <pending>     80:31320/TCP     9s
theoc@MBP-de-Theo-2 project % kubectl get services
NAME                     TYPE           CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
kubernetes               ClusterIP      10.96.0.1       <none>        443/TCP          70d
simple-node-app          NodePort       10.110.153.17   <none>        3000:31637/TCP   8m17s
simple-node-app-service  LoadBalancer   10.96.121.199   <pending>     80:31320/TCP     31s
theoc@MBP-de-Theo-2 project % minikube service simple-node-app-service --url
http://127.0.0.1:50248
❗  Comme vous utilisez un pilote Docker sur darwin, le terminal doit être ouvert pour l'exécuter.
```



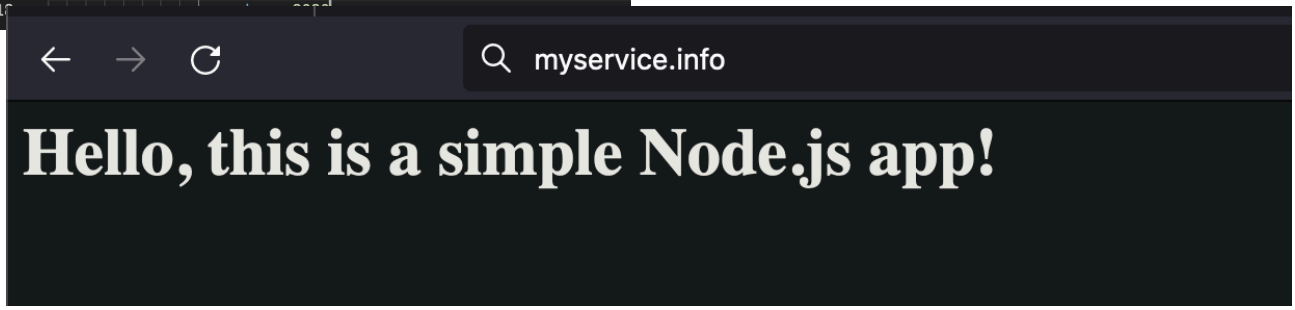
8. Add service

We use the ingress.yaml file provided in the GitHub.

```
theoc@macbook-pro-de-theo project % kubectl apply -f ingress.yaml
ingress.networking.k8s.io/example-ingress created
theoc@macbook-pro-de-theo project % kubectl get ingress example-ingress
NAME              CLASS    HOSTS              ADDRESS   PORTS   AGE
example-ingress   <none>   myservice.info               80      86s
theoc@macbook-pro-de-theo project %
```

Users > theoc > Documents > efrei > S9 > cloud > project > ! ingress.yaml

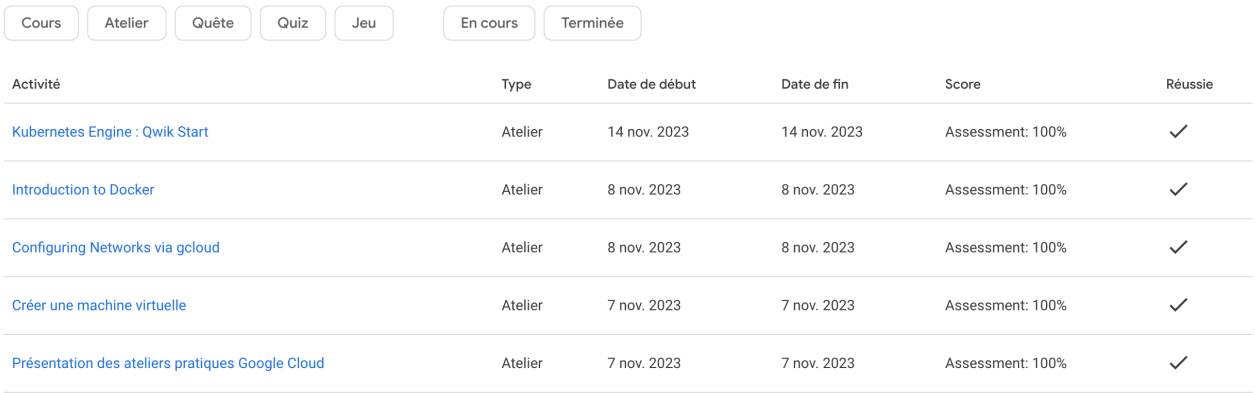
```yaml
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: example-ingress
  annotations:
#    nginx.ingress.kubernetes.io/rewrite-target: /$1
spec:
  rules:
    - host: myservice.info
      http:
        paths:
          - path: /
            pathType: Prefix
            backend:
              service:
                name: myservice
                port:
```



9. Add a second service

I used the file front-back-app.yaml, I didn’t really understood this part so it is not completed but this is my work so far.

Then I used :
kubectl apply -f front-back-app.yaml
But I can’t access it on browser, I’m still wondering why



Activité	Type	Date de début	Date de fin	Score	Réussie
Kubernetes Engine : Qwik Start	Atelier	14 nov. 2023	14 nov. 2023	Assessment: 100%	✓
Introduction to Docker	Atelier	8 nov. 2023	8 nov. 2023	Assessment: 100%	✓
Configuring Networks via gcloud	Atelier	8 nov. 2023	8 nov. 2023	Assessment: 100%	✓
Créer une machine virtuelle	Atelier	7 nov. 2023	7 nov. 2023	Assessment: 100%	✓
Présentation des ateliers pratiques Google Cloud	Atelier	7 nov. 2023	7 nov. 2023	Assessment: 100%	✓