#032 Kubernetes - Pods lab 2

Introduction

this is part 32 from the journey it's a long journey(360 day) so go please check previous parts , and if you need to walk in the journey with me please make sure to follow because I may post more than once in 1 Day but surely I will post daily at least one 😊.

And I will cover lot of tools as we move on.

---

lab

in last part we create a pod . we are going to delete this pod using

```
(base) in ~
> kubectl delete pods first-pod
pod "first-pod" deleted
(base) in ~
>
```

```
kubectl delete first-pod
```

As we see the pod got deleted , now I am going to use another random container from docker hub , I found an container called simple-api you can take a look at it here

let's create a pod from it

```
(base) in ~
> kubectl run first-pod --image=emondek/simple-api:latest --restart=Never
pod/first-pod created
```

```
kubectl run first-pod --image=emondek/simple-api:latest --restart=Never
```

let's take a look

```
(base) in ~
> kubectl get pods
NAME        READY   STATUS            RESTARTS   AGE
first-pod   0/1     ContainerCreating 0          11s
```

```
kubectl get pods
```

we can see status creating container , if we need more info about it we type

```
(base) in ~
> kubectl describe pods/first-pod
Name:          first-pod
Namespace:     default
Priority:      0
Node:          minikube/192.168.99.100
Start Time:    Fri, 10 Jul 2020 01:03:44 +0300
Labels:        run=first-pod
Annotations:   <none>
Status:        Pending
```

```
kubectl describe pods/first-pod
```

if we scroll down we can see events

```
Events:
  Type    Reason     Age   From                Message
  ----    ------     ----  ----                -------
  Normal  Scheduled  42s   default-scheduler   Successfully assigned default/first-pod to minikube
  Normal  Pulling    41s   kubelet, minikube   Pulling image "emondek/simple-api:latest"
(base) in ~
>
```

it's now on image pulling phase

after pulling we can see now that status is RUNNING
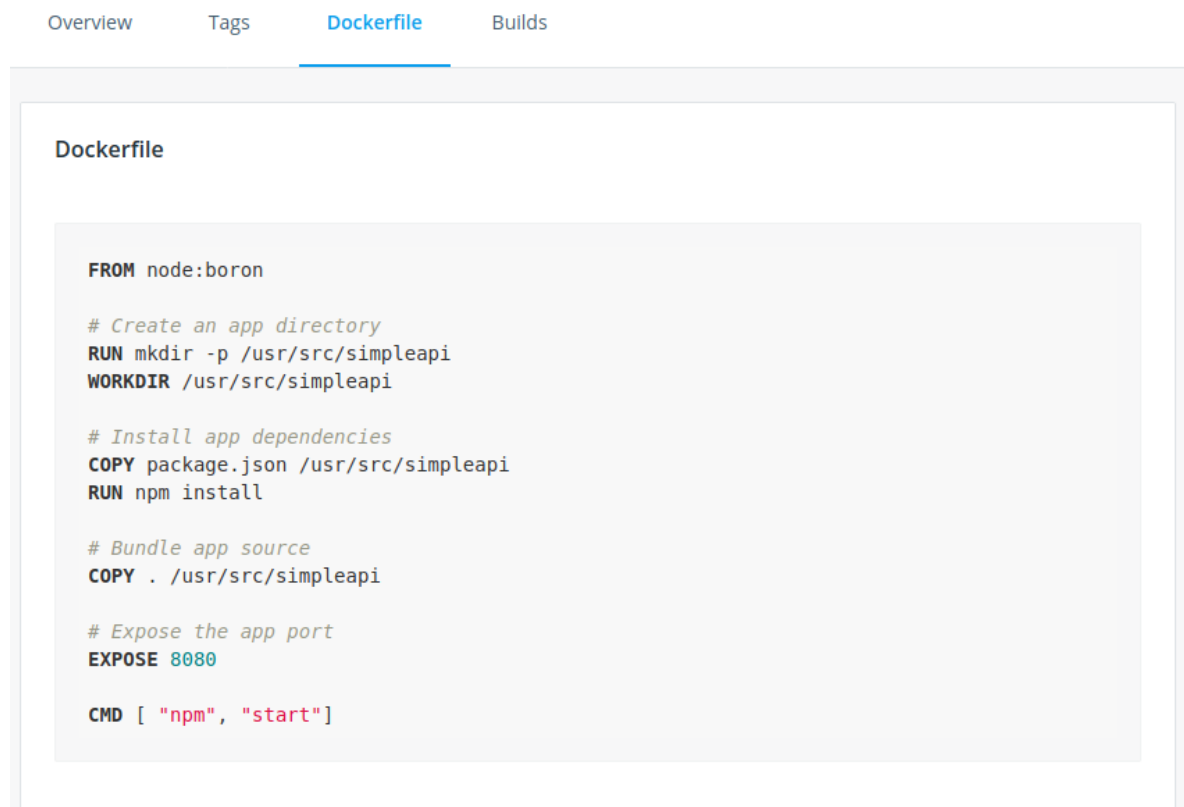
```
(base) in ~
> kubectl get pods
NAME         READY     STATUS     RESTARTS     AGE
first-pod    1/1       Running    0            21m
(base) in ~
```

now to run this pod in browser we need services(will talk about them later) but we have a nice hack to run it for now

```
(base) in ~
> kubectl port-forward pods/first-pod 8080
Forwarding from 127.0.0.1:8080 -> 8080
Forwarding from [::1]:8080 -> 8080
Handling connection for 8080
```
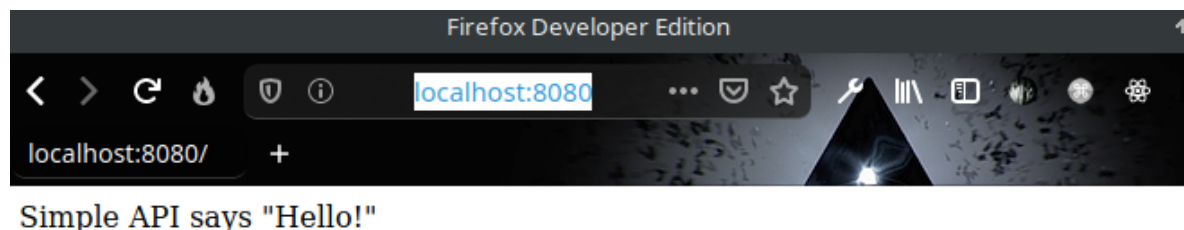
```
kubectl port-forward pods/first-pod 8080
```

8080 is the port the we decide to use in dockerfile

Dockerfile

```
FROM node:boron

# Create an app directory
RUN mkdir -p /usr/src/simpleapi
WORKDIR /usr/src/simpleapi

# Install app dependencies
COPY package.json /usr/src/simpleapi
RUN npm install

# Bundle app source
COPY . /usr/src/simpleapi

# Expose the app port
EXPOSE 8080

CMD [ "npm", "start"]
```

if we go to the browser and type http://localhost:8080/

we have our pod running

we can also access this pod using

```
(base) in ~
> kubectl exec -it first-pod -- bash
root@first-pod:/usr/src/simpleapi# ls
Dockerfile  LICENSE  README.md  index.js  node_modules  package.json
root@first-pod:/usr/src/simpleapi#
```

```
kubectl exec -it first-pod -- bash
```

-- it separate the commands that related to kubectl and linux.
And every change we make to this pod will be discarded just like container so also
we need volumes here :D