#044 Kubernetes - Services 4

Introduction

this is part 44 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 😍.

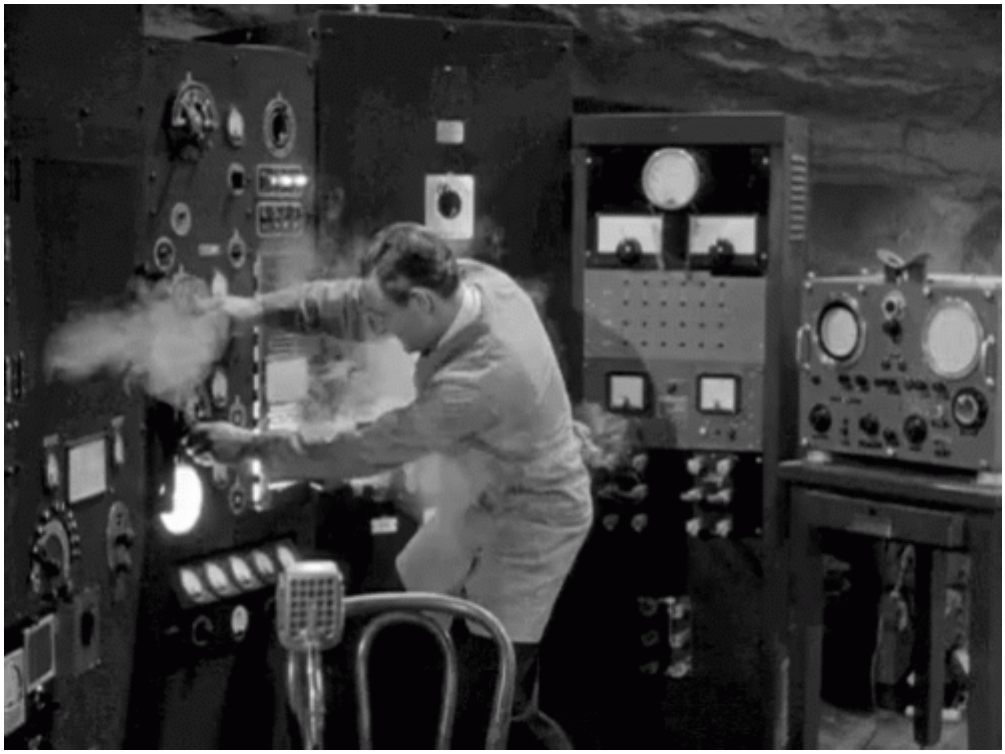
And I will cover lot of tools as we move on.

Prepare file

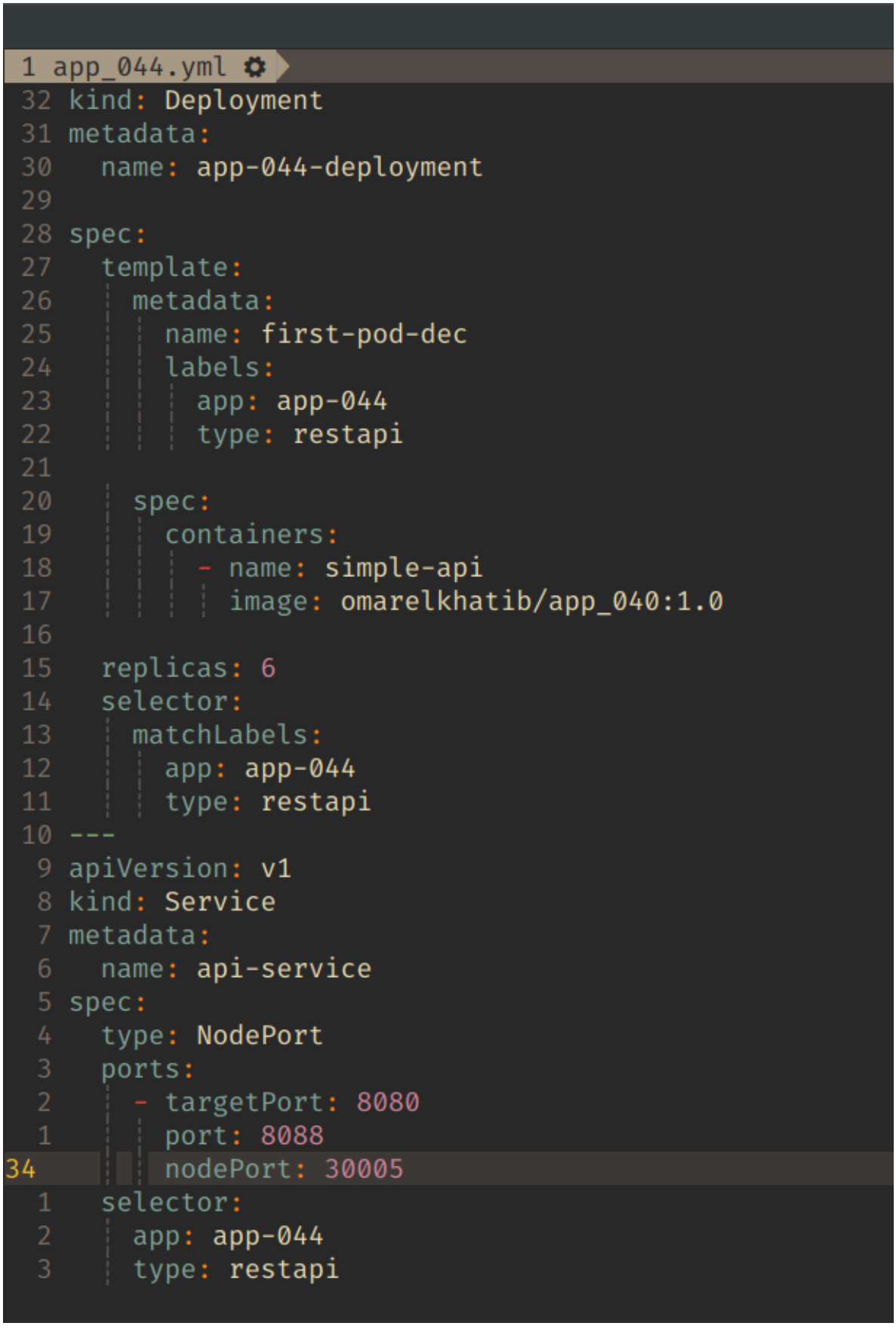
as always all the files used are in my github repo here
{% github OmarElKhatibCS/DevOpsJourney no-readme %}
if you already have it just pull , if not clone it.
the source code usually hold the same chapter number we are in 044 so app_044 is what we are looking for.
I am using the same image of app_040:1.0 here because it's the same if this part but we need to access it without trick of port-forward

---

Lab



first let's take a look at our yml configs



We can see I use an --- to separate 2 yml files , I can separate them each one on file but since they are depends on each other it's good practice to have them in 1 file.

the key feature here is :
1 . type: NodePort , so we can access our app outside the cluster
2 . Ports:
target port is 8080 which is the node js port that I choose for backend in my configs.
nodePort: 30005 is the port of my node which I will use to access the node.
port: 8088 is the port of my service.

```
✔ ~/Documents/DevOpsJourney/app_044 [master|…1]
23:43 $ kubectl apply -f app_044.yml
deployment.apps/app-044-deployment created
service/api-service created
```

```
kubectl apply -f app_044.yml
```

again and again I like to repeat it notice I am in the same folder as myfile so I can pass it directly.

```
✔ ~/Documents/DevOpsJourney/app_044 [master|…1]
23:43 $ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
app-042-5554954c96-hl7qz            1/1     Running   1          3d21h
app-044-deployment-b975c9494-9p7wg  1/1     Running   0          6s
app-044-deployment-b975c9494-khs8b  1/1     Running   0          6s
app-044-deployment-b975c9494-nqwdw  1/1     Running   0          6s
app-044-deployment-b975c9494-nxr4n  1/1     Running   0          6s
app-044-deployment-b975c9494-sw8w2  1/1     Running   0          6s
app-044-deployment-b975c9494-tsbdx  1/1     Running   0          6s
```

```
kubectl get pods
```

we have our 6 pods created , I like to mention we can create shortcuts for us in Linux (spoiler : I will make an entire LinuxJourney to discover linux from kernel and up :) after devops or in parallel) , Linux have aliases so we can tell linux to recognize kgp as shortcut for 'kubectl get pods' , I have them but I like to keep it simple for those who are not familiar with aliases yet.

```
✔ ~/Documents/DevOpsJourney/app_044 [master|…1]
23:44 $ kubectl get svc
NAME          TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
api-service   NodePort    10.110.14.113   <none>        8088:30005/TCP   47s
```

```
kubectl get svc
```

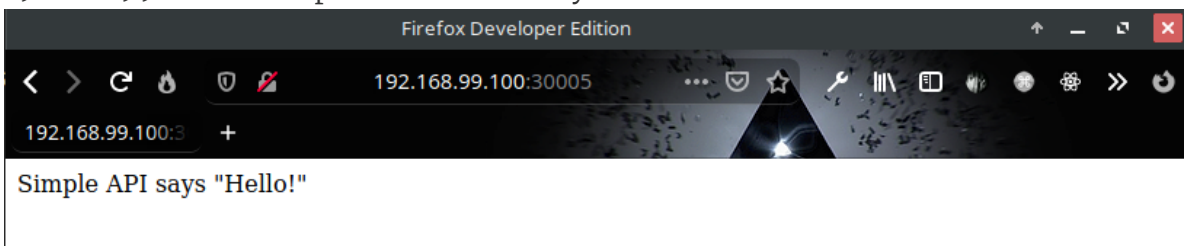
svc shortcut for services , we seen before. We can see our service is ready.

to get the node ip , basically it's the minikube ip because minikube is the test node for us locally.

```
✔ ~/Documents/DevOpsJourney/app_044 [master|…1]
23:44 $ minikube ip
192.168.99.100
```

```
minikube ip
```

192.168.99.100 is the ip of the node in my case.



now we have the ip , go to browser and type 192.168.99.100:30005
now we can access our app directly without the port-forward trick :)