# #046 Kubernetes - Services 6

# Introduction


this is part 46 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 files

this part was supposed to be an aws or azure lab , but due current situation of my country (big explosion happen due chemical products) and I don't have a visa to activate free account. So I will use an Nginx server instead .

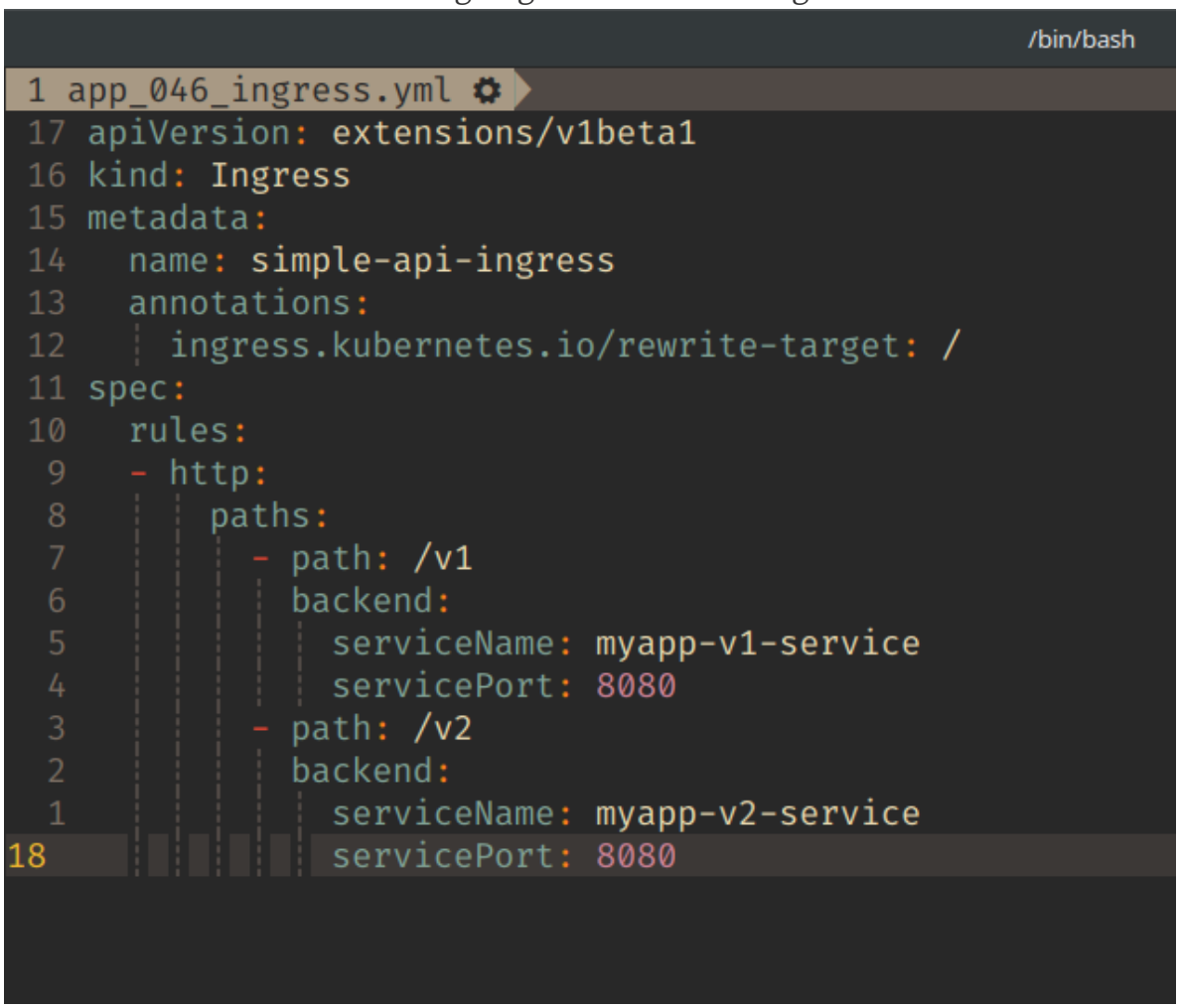
as always all the files used are in my github repo here
{% github OmarElKhatibCS/DevOpsJourney no-readme %}
if you already have it just pull , if not clone it.
the source code usually hold the same chapter number we are in 046 so app_046 is what we are looking for.

---

# Understand files


this is the app_046_v1.yml it's an typical deployment it use the google samples hello app. and we have service here of type clusterIp (we can use also NodePort) , this port 8080 is the default port of the hello image and the name of app is myapp-v1 and service myapp-v1 . Same for the v2 (go and take a look). What I care here

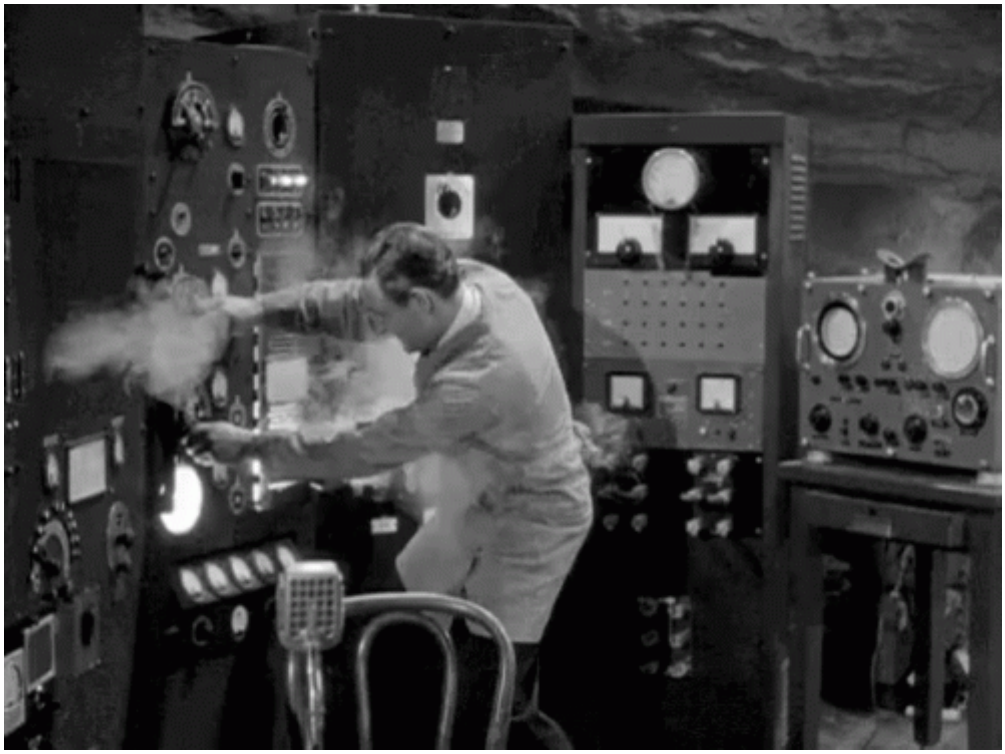
about is the service because it's going to talk with the ingress.




we can see here annotations it's like javascript object , it's here to configure the nginx server , please go to the docs and read it.
we can notice it's take a path /v1 and /v2 , and port same as the service , and name same of service name.

---

# Lab



as we discuss on the #045 when we expose our app to external ip we need Load balancer or Ingress .
Ingress it's mean different routes to same hosts , example test.app.example and test2.app.example or app.example/test and app.example/test2
every route connect to an web service.

we are going to use the Ingress method in this lab.

```
in ~
> minikube addons enable ingress
🔎  Verifying ingress addon...
🌟  The 'ingress' addon is enabled
```

```
minikube addons enable ingress
```

first we need to enable the ingress add on inside minikube because we are going to use nginx as server for this lab.

```
✔ ~/Documents/DevOpsJourney/app_046 [master|…1]
01:12 $ l
v1/  v2/  app_046_ingress.yml  app_046_v1.yml  app_046_v2.yml
✔ ~/Documents/DevOpsJourney/app_046 [master|…1]
01:12 $ kubectl create -f app_046_v1.yml
deployment.apps/myapp-deployment-v1 created
✔ ~/Documents/DevOpsJourney/app_046 [master|…1]
01:12 $ kubectl create -f app_046_v2.yml
deployment.apps/myapp-deployment-v2 created
✔ ~/Documents/DevOpsJourney/app_046 [master|…1]
01:12 $
```

now let's create our deployments with name of deployments-v1 and v2.

```
kubectl create -f app_046_v1.yml
kubectl create -f app_046_v2.yml
```

```
✔ ~/Documents/DevOpsJourney/app_046 [master|…1]
01:47 $ kubectl get pods
NAME                                  READY   STATUS    RESTARTS   AGE
myapp-deployment-v1-5698d7c595-84ccd  1/1     Running   0          13s
myapp-deployment-v1-5698d7c595-dtkkw  1/1     Running   0          13s
myapp-deployment-v1-5698d7c595-dv96q  1/1     Running   0          13s
myapp-deployment-v1-5698d7c595-l6b97  1/1     Running   0          13s
myapp-deployment-v1-5698d7c595-wlfkb  1/1     Running   0          13s
myapp-deployment-v1-5698d7c595-xbk7j  1/1     Running   0          13s
myapp-deployment-v2-5bdf686664-d6nf9  1/1     Running   0          10s
myapp-deployment-v2-5bdf686664-dq49v  1/1     Running   0          10s
myapp-deployment-v2-5bdf686664-j9qw9  1/1     Running   0          10s
myapp-deployment-v2-5bdf686664-mk9r8  1/1     Running   0          10s
myapp-deployment-v2-5bdf686664-qd22f  1/1     Running   0          10s
myapp-deployment-v2-5bdf686664-tz82x  1/1     Running   0          10s
```

```
kubectl get pods
```

all our pods are created , now time to create our ingress service.

```
✔ ~/Documents/DevOpsJourney/app_046 [master|…1]
01:48 $ kubectl create -f app_046_ingress.yml
ingress.extensions/simple-api-ingress created
```

```
kubectl create -f app_046_ingress.yml
```

we can see our service successfully created.

```
✔ ~/Documents/DevOpsJourney/app_046 [master|…1]
01:55 $ kubectl get ingress
NAME                 CLASS    HOSTS   ADDRESS          PORTS   AGE
simple-api-ingress   <none>   *       192.168.99.100   80      6m16s
```

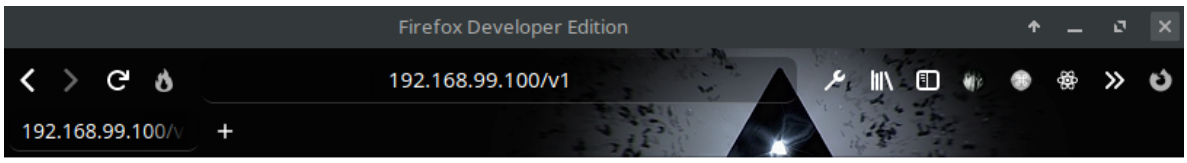
```
kubectl get ingress
```

now we have our ingress ready

```
✔ ~/Documents/DevOpsJourney/app_046 [master|…1]
02:40 $ minikube ip
192.168.99.100
```

```
minikube ip
```

192.168.99.100 is our local ip , if our azure setup is online we will get a real external ip that we can access any time , anywhere , and you also can access it!




```
Hello, world!
Version: 1.0.0
Hostname: myapp-deployment-v1-754d67d968-f9mb9
```


```
Hello, world!
Version: 2.0.0
Hostname: myapp-deployment-v2-55fb8ccd95-lhhgk
```

head to 192.168.99.100/v1 and 192.168.99.100/v2 the ip may vary for you so just change the ip.

we can see now we can access 2 different apps (microservices) using same ip with different routes !