

# #041 Kubernetes - Services 1

## Introduction



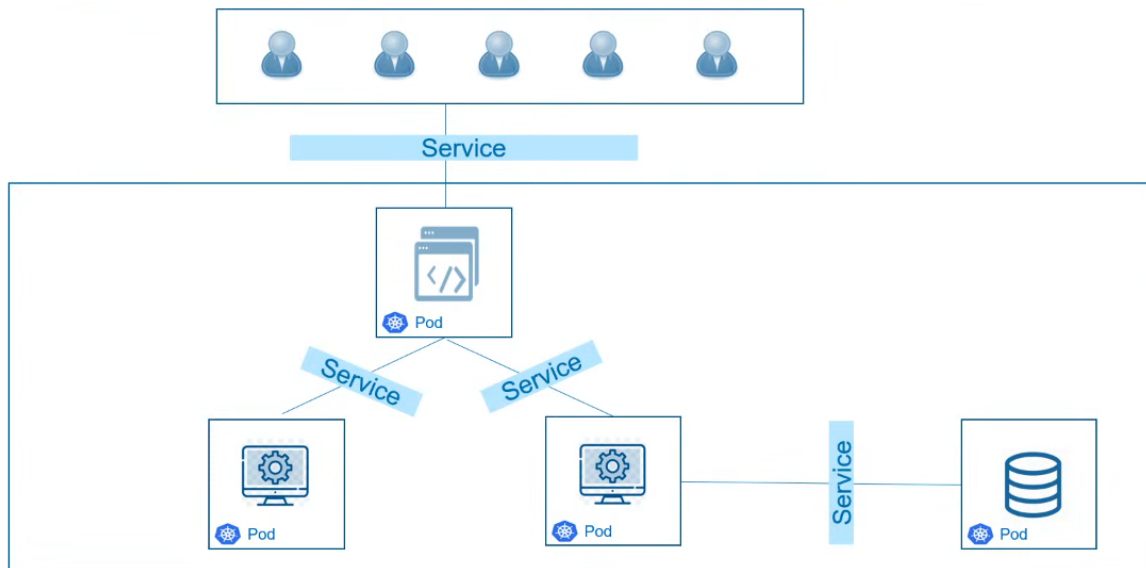
this is part 41 from the journey it's a long journey(360 day) so go please check previous parts , and if you need to walk in the journey with me please make sure to follow because I may post more than once in 1 Day but surely I will post daily at least one 😊.

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

---

## Services

Today we are going about services in Kubernetes and what they do.



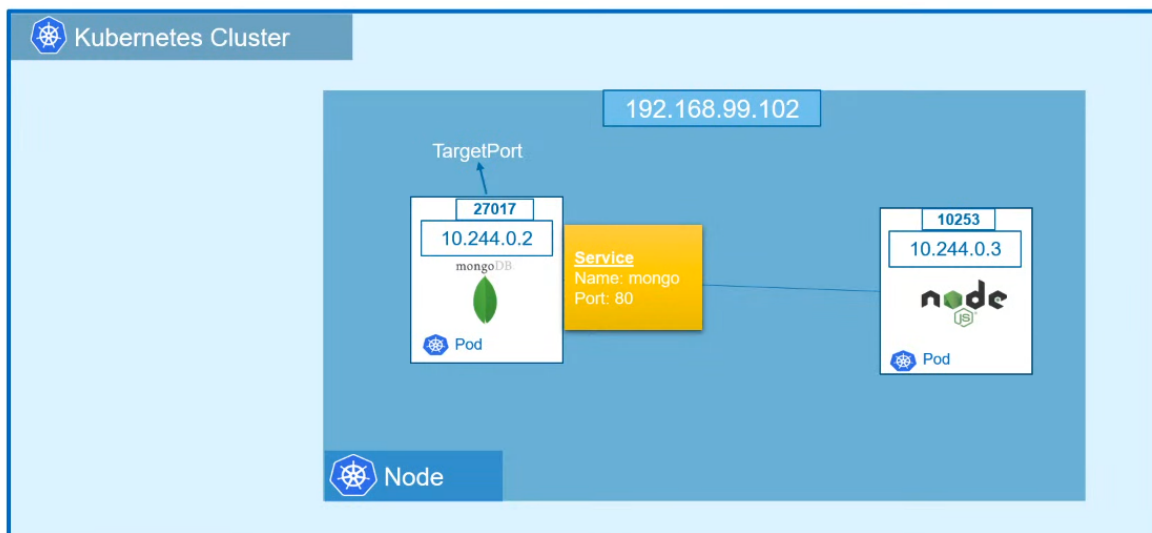
In this picture we have 4 pods , 1 for front end (icon terminal) , 2 for backend (icon gear) and 1 for database (icon DB...)

For sure in this picture the front end and backend are separated.

those pods need to communicate between them selves they need what we call a service.

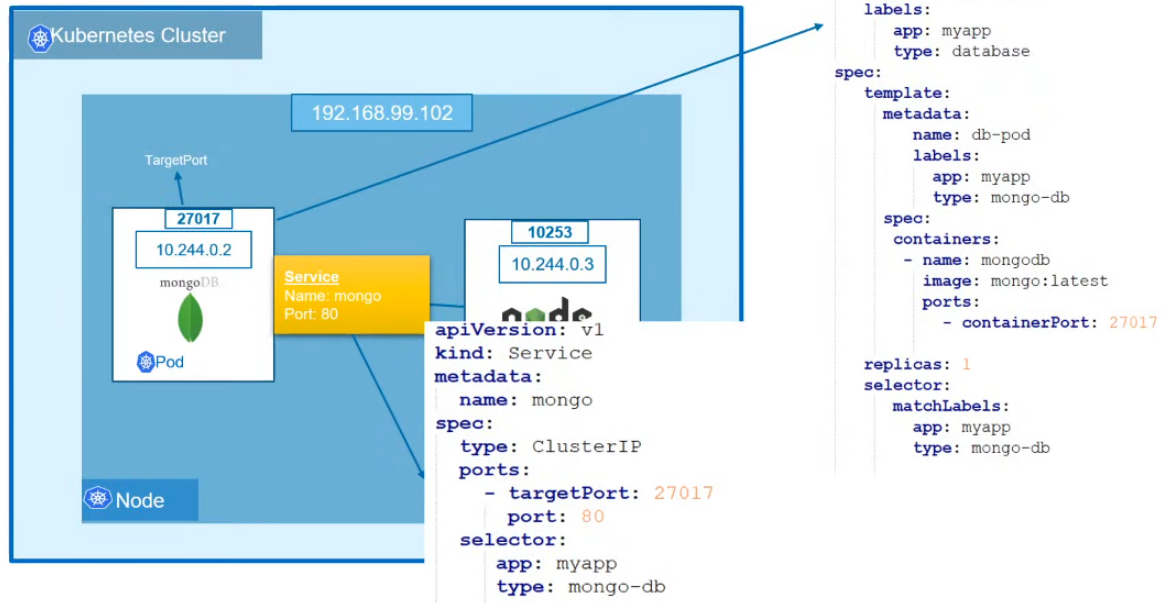
And we have a service to make the user communicate with the front-end.

## ClusterIP

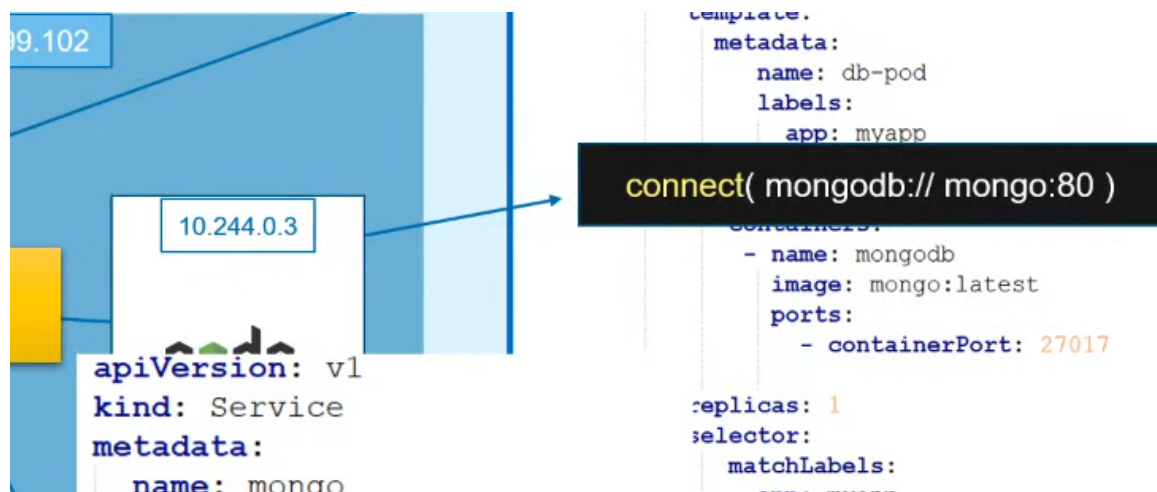


Let's say this is our DB pod and backend pods , in able to communicate between those pods we need an service it's called ClusterIP , this service take a port and a name , the port here is 80 you can choose any port that is available , the target port is the port of the container , in our case it's mongoDB . MongoDB use this port as default one (you can look at dockerhub image docs).

# Services (ClusterIP)



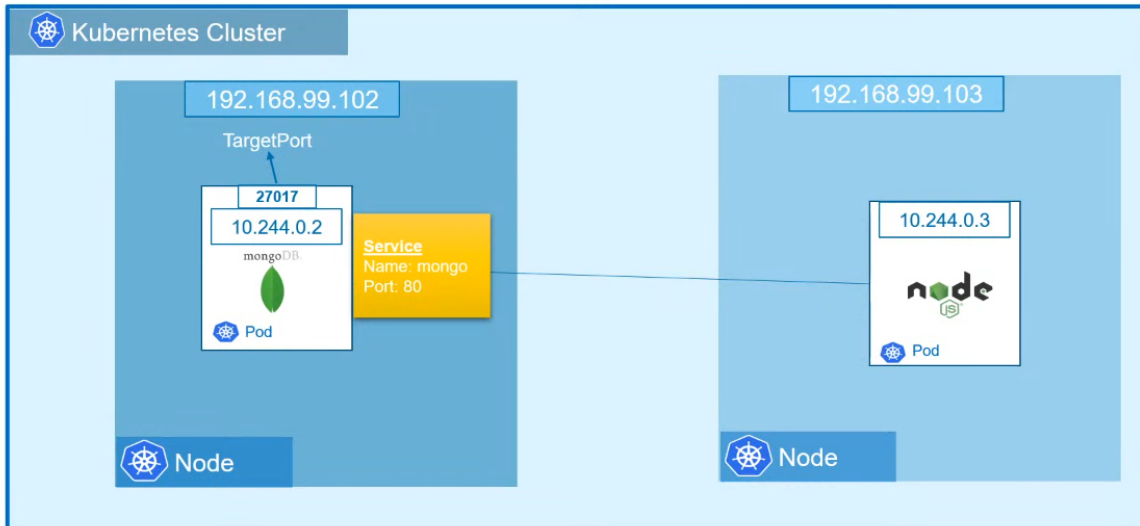
here a look at the config file for service and the mongo-db pod.  
for the service the selector app is to specify what pod I need to provide a service for it. targetPort is the containerPort of mongo-db , and 80 is the port of service.  
So the service is now to serve the mongo-db pod.



in the backend we call the function

```
connect (mongodb://mongo:80)
```

mongo is the name of our service and 80 is the port of our service.



now we have our pods in 2 separated nodes. This is a problem? well no , because the master handle this for us. This is true just inside the same Kubernetes cluster.