

# CORRECTION TP 2 KUBERNETES

---

## PARTIE 2: l'interaction avec le cluster

1. Vérifiez l'état du cluster.

```
vagrant@kmaster:~$ kubectl cluster-info
Kubernetes control plane is running at https://172.42.42.100:6443
KubeDNS is running at
https://172.42.42.100:6443/api/v1/namespaces/kube-system/services/kube-dns:
dns/proxy
To further debug and diagnose cluster problems, use 'kubectl cluster-info
dump'.
```

2. créez un pods qui lance un connecteur ubuntu avec la commande bash.

```
vagrant@kmaster:~$ kubectl run myubuntu -it --image ubuntu -- bash
If you dont see a command prompt, try pressing enter.
root@myubuntu:/# ls
bin boot dev etc home lib lib32 lib64 libx32 media mnt opt proc
root run sbin srv sys tmp usr var
root@myubuntu:/#
```

3. récupérez l'adresse ip du pods en ouvrant un nouvel onglet.

```
vagrant@kmaster:~$ kubectl get pods -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP            NODE       NOMINATED NODE
READINESS GATES
myubuntu      1/1     Running   0           6m22s  192.168.77.130 kworker2   <none>         <none>
vagrant@kmaster:~$
```

- 
- connectez-vous au worker sur lequel tourne votre pods et lancer la commande `docker ps`

```
vagrant@kworker2:~$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
5270d8b958bf	ubuntu	"bash"	6 minutes ago	Up 6 minutes		
k8s_myubuntu_myubuntu_default_f431a08e-6e07-4cf5-8cf4-272b77b91b2f_0						
d24386970d11	k8s.gcr.io/pause:3.2	"/pause"	6 minutes ago	Up 6 minutes		
k8s_POD_myubuntu_default_f431a08e-6e07-4cf5-8cf4-272b77b91b2f_0						
c9a39136abfc	15f795b449d2	"start_runit"	25 minutes ago	Up 25 minutes		
k8s_calico-node_calico-node-jtj69_kube-system_c509cf76-d4b4-4023-8b8c-fa63ad22f09a_2						
abfe957df70e	e3f6fcd87756	"/usr/local/bin/kube..."	26 minutes ago	Up 26 minutes		
k8s_kube-proxy_kube-proxy-5bcjs_kube-system_9386a9f3-50f2-46dc-8510-c81e155e7066_2						
b8c8bb8806e7	k8s.gcr.io/pause:3.2	"/pause"	26 minutes ago	Up 26 minutes		
k8s_POD_calico-node-jtj69_kube-system_c509cf76-d4b4-4023-8b8c-fa63ad22f09a_2						
f9c62bcb25de	k8s.gcr.io/pause:3.2	"/pause"	26 minutes ago	Up 26 minutes		
k8s_POD_kube-proxy-5bcjs_kube-system_9386a9f3-50f2-46dc-8510-c81e155e7066_2						

- supprimez le pods

```
vagrant@kmaster:~$ kubectl delete pods myubuntu
pod "myubuntu" deleted
```

- déployez un serveur nginx

```
vagrant@kmaster:~$ kubectl create deployment mynginx --image nginx
deployment.apps/mynginx created
```

- afficher les pods

```
vagrant@kmaster:~$ kubectl get pods -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATES
mynginx-5b686ccd46-zcz4v	1/1	Running	0		2m38s	192.168.77.131	kworker2	<none>

- supprimer le pods, que remarquez-vous en affichant les pods

```
vagrant@kmaster:~$ kubectl get pods -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE	READINESS GATES
mynginx-5b686ccd46-zcz4v	1/1	Running	0		2m38s	192.168.77.131	kworker2	<none>

```
vagrant@kmaster:~$ kubectl delete pod mynginx-5b686ccd46-zcz4v
```

```
pod "mynginx-5b686ccd46-zcz4v" deleted
vagrant@kmaster:~$ kubectl get pods -o wide
NAME                                READY   STATUS             RESTARTS   AGE   IP          NODE   NOMINATED NODE   READINESS GATES
mynginx-5b686ccd46-drzlk            0/1     ContainerCreating   0          24s   <none>      kworker1  <none>           <none>
```

lors de la suppression du pods kubernetes recrée automatiquement un nouveau pods

9. Pour pouvoir accéder au serveur nginx, il faudra créer un service, créer un service relié au déploiement que vous avez fait.

```
vagrant@kmaster:~$ kubectl create service nodeport mynginx --tcp 8080:80
service/myginx created
```

10. nginx point vers un fichier qui se trouve /usr/share/nginx/html/index.html, accéder au pods qui correspond au déploiement et modifier ce fichier en écrivant " i am worker 1"

```
vagrant@kmaster:~$ kubectl exec -ti mynginx-5b686ccd46-drzlk /bin/bash
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future
version. Use kubectl exec [POD] -- [COMMAND] instead.
root@mynginx-5b686ccd46-drzlk:/# echo "i am woker 1" >
/usr/share/nginx/html/index.html
```

11. récupérer le port ouvert par le service

```
vagrant@kmaster:~$ kubectl get service
NAME            TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)          AGE
kubernetes      ClusterIP   10.96.0.1       <none>        443/TCP          33d
myginx          NodePort    10.102.108.203  <none>        8080:32495/TCP   23m
```

12. Récupérer l'IP du master

```
vagrant@kmaster:~$ kubectl get nodes -o wide
NAME      STATUS   ROLES                  AGE   VERSION   INTERNAL-IP   EXTERNAL-IP   OS-IMAGE           KERNEL-VERSION   CONTAINER-RUNTIME
kmaster   Ready    control-plane,master   33d   v1.20.1   172.42.42.100 <none>        Ubuntu 20.04.1 LTS  5.4.0-58-generic  docker://20.10.1
kworker1   Ready    <none>                 33d   v1.20.1   172.42.42.101 <none>        Ubuntu 18.04.5 LTS  4.15.0-128-generic docker://20.10.1
kworker2   Ready    <none>                 33d   v1.20.1   172.42.42.102 <none>        Ubuntu 18.04.5 LTS  4.15.0-128-generic docker://20.10.1
```

---

13. Requête le serveur nginx en lançant la commande curl [master ip address: service port]

```
vagrant@kmaster:~$ curl 172.42.42.100:32495
i am worker 1
```

14. Nous voulons que notre serveur puisse supporter un grand nombre de requêtes pour selon je vous propose de faire un scale horizontal sur notre déploiement .

```
vagrant@kmaster:~$ kubectl scale deployment mynginx --replicas=3
deployment.apps/mynginx scaled
```

15. Vérifier le nombre de pods actif

```
vagrant@kmaster:~$ kubectl get pods -o wide
```

NAME	READY	STATUS	RESTARTS	AGE	IP	NODE	NOMINATED NODE
mynginx-5b686ccd46-2hw6r	1/1	Running	0	4m45s	192.168.41.132	kworker1	<none>
<none>							
mynginx-5b686ccd46-drzlk	1/1	Running	0	37m	192.168.41.131	kworker1	<none>
<none>							
mynginx-5b686ccd46-q2lpf	1/1	Running	0	4m45s	192.168.77.132	kworker2	<none>
<none>							

on remarque que 2 autres pods ont été créer

16. accéder aux nouveau pods qui correspond et modifier ce fichier en écrivant " i am worker 2" et " i am worker 3"

```
vagrant@kmaster:~$ kubectl exec -ti mynginx-5b686ccd46-q2lpf /bin/bash
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future
version. Use kubectl exec [POD] -- [COMMAND] instead.
root@mynginx-5b686ccd46-q2lpf:/# echo "i am worker 2" >
/usr/share/nginx/html/index.html
root@mynginx-5b686ccd46-q2lpf:/# exit
exit
vagrant@kmaster:~$ kubectl exec -ti mynginx-5b686ccd46-2hw6r /bin/bash
kubectl exec [POD] [COMMAND] is DEPRECATED and will be removed in a future
```

---

```
version. Use kubectl exec [POD] -- [COMMAND] instead.
root@mynginx-5b686ccd46-2hw6r:/# echo "i am worker 3" >
/usr/share/nginx/html/index.html
root@mynginx-5b686ccd46-2hw6r:/# exit
exit
```

17. Lancer plusieurs fois la commande curl sur l'adress précédente, que remarquez vous ?

```
vagrant@kmaster:~$ while true;do curl 172.42.42.100:32495;done
i am worker 3
i am worker 1
i am worker 1
i am worker 1
i am worker 1
i am worker 3
i am worker 2
i am worker 3
i am worker 2
i am worker 2
```

on remarque qu'il y'a un effet de load balancing ce qui permet de diminuer la charge sur le serveur

18. faite en sorte qu'il y'ai au minimum 3 réplicas et au maximum 10.

```
vagrant@kmaster:~$ kubectl autoscale deployment mynginx --min=3 --max=10
horizontalpodautoscaler.autoscaling/mynginx autoscaled
```