Pod

Si vous avez un problème avec un pod, regardez les logs via :
kubectl describe pod -n namespace nom_du_pos
kubectl logs -n namespace nom_du_pos

Le but de ce TP est de vous familiariser avec les notions de Pod et Contrôleur de k8s. Je vous invite à créer l'ensemble des objets dans un namespace «development».
Pour éviter des problèmes, utiliser la création et la mise à jour déclarative d'objet via des fichiers yaml et la commande

```
$ kubectl apply -f fichier.yaml
```

et pour la destruction

```
$ kubectl delete -f fichier.yaml
```

1. Création du namespace

Créer un fichier namespace.yaml qui contient la description du namespace «development»

```
$ kubectl apply -f namespace.yaml
```

Vérifier que «development» existe bien

```
$ kubectl get namespaces --show-labels
```

2. Création d'un pod

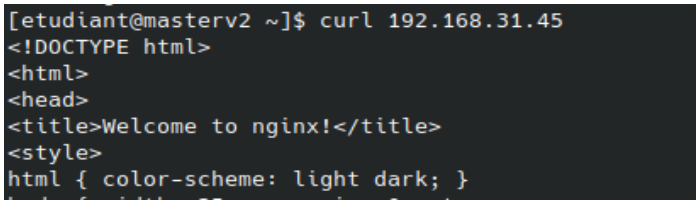
Pour votre premier pod, nous allons utiliser une image Docker déjà crée, en l'occurrence nginx.
Créer un fichier pods.yaml qui contient la description du pods avec le namespace «development»

```
$ kubectl apply -f pods.yaml
```

Vérifiez que le pods est bien créé, et déterminez son adresse IP (192.168.XXX.YYY).
Connectez-vous sur le master ou bien un nœud et envoyez une requête au pod

```
$ curl 192.168.XXX.YYY
```

Vous devez avoir une réponse du type



Supprimer maintenant le pods.yaml

```
$ kubectl delete  -f pods.yaml
```

3. Création d'un déploiement.

Créez maintenant un fichier deployment.yaml qui contient la description d'un déploiement avec 3 réplicas de nginx dans le namespace «development».
Comme précédemment, regardez la sortie de la commande curl sur un des nœuds.

Maintenant, modifiez le fichier pour que httpd soit lancé au lieu de nginx et faire

```
$ kubectl apply -f deployment.yaml
```

Vérifiez que la sortie de curl indique qu'apache est maintenant lancé.
Regardez l'historique des déploiements

```
$ kubectl rollout history  -n development deployment.v1.apps/frontend
```

Pour revenir à la version précédente

```
$ kubectl rollout undo -n development deployment.v1.apps/frontend
```

Supprimer maintenant le déploiement

```
$  kubectl delete  -f  deployment.yaml
```

4. Autre contrôleur.

De la même manière, je vous laisse découvrir les autres contrôleurs mis a disposition par k8s.

5. Back-end.

Pour la suite du cours, nous aurons besoins d'un déploiement avec 3 réplicas dans le namespace «development» d'un serveur REST qui donne l'ip du pod grâce à l'api suivante :

GET /api/v0/ip

Le nom des pods sera «ms-ip»
En utilisant le langage que vous voulez, créer une image docker à partir de votre serveur d'IP.

Je vous conseille :

- pour java d'utiliser le framework SPARK.
- pour go d'utiliser le framework chi

Pour pouvoir utiliser cette image docker, le plus simple est de l'exporter vers le registre privé situé en 10.54.56.39 du cluster.
Voici une partie d'un script que j'utilise pour automatiser le build et le déploiement

```
#!/bin/bash

NOM=tmp-serveur
VERSION=0.0.11
REGISTRY="10.54.56.39:5000"

TAG=$REGISTRY/$NOM:$VERSION

INFO="$VERSION build `date`"
echo $INFO

docker build --rm -t $TAG  --build-arg VERSION="$INFO" .
docker push $TAG
```

Pour utiliser cette image dans un pod, il est nécessaire de renseigner les paramètres de connexion dans un objet de type secret

```
apiVersion: v1
kind: Secret
metadata:
  name: private-registry-key
  namespace: development
data:
  .dockerconfigjson:
eyJhdXRocyI6IHsgIjEwLjU0LjU2LjM5OjUwMDAiOiB7ICJhdXRoIjogIlpYUjFaR2xoYm5RNlpYUjFaR2xoYm5RPSJ9fX0=
type: kubernetes.io/dockerconfigjson
```

La variable .dockerconfigjson contient le fichier
/home/etudiant/snap/docker/current/.docker/config.json
encodé sous le format base64, sans espace.

Dans la spec « containers » du pod, rajouter

```
spec:
  containers:
  - name: pod-ms-ip
    image: 10.54.56.39:5000/ms-ip:0.0.1
    imagePullPolicy: Always
  imagePullSecrets:
  - name: private-registry-key
```

Ceci est décrit ici :
https://kubernetes.io/fr/docs/tasks/configure-pod-container/pull-image-private-registry/

6. Health probe

Pour aller plus loin, vous pouvez implémenter le pattern « Heath probe » décrit dans le chapitre 4 du livre