Université de Picardie Jules Verne

Informatique – Master CCM

INSSET – Saint-Quentin

Conteneurs Applicatifs et Micro-Services M2

C. Drocourt

cyril.drocourt@u-picardie.fr

Cours 5.2 : Kubernetes – Les pods

V2023.2

Table des matières

1 - Lister les Pod

On peut consulter la liste des Pods qui doit être vide :

```
root@maitre:~# kubectl get pods
No resources found.
```

La commande n'affiche que les « pods » contenus dans le « namespace » par défaut, pour afficher tous les « pods » :

```
root@maitre:~# kubectl get pods --all-namespaces
NAMESPACE     NAME                                READY STATUS  RESTARTS AGE
kube-system   coredns-576cbf47c7-csd87            1/1   Running 0        45m
kube-system   coredns-576cbf47c7-rvjt2            1/1   Running 0        45m
kube-system   etcd-lemaitre                       1/1   Running 0        45m
kube-system   kube-apiserver-lemaitre             1/1   Running 0        45m
kube-system   kube-controller-manager-lemaitre 1/1   Running 1        45m
kube-system   kube-flannel-ds-amd64-nvpmt         1/1   Running 0        16m
kube-system   kube-flannel-ds-amd64-q5bb7         1/1   Running 0        29m
kube-system   kube-flannel-ds-amd64-rdjq4         1/1   Running 0        37m
kube-system   kube-flannel-ds-kjlb4               1/1   Running 0        16m
kube-system   kube-flannel-ds-kxqdk               1/1   Running 0        37m
kube-system   kube-flannel-ds-r8s7j               1/1   Running 0        29m
kube-system   kube-proxy-2lrz5                    1/1   Running 0        45m
kube-system   kube-proxy-nlhjg                    1/1   Running 0        16m
kube-system   kube-proxy-ztqcj                    1/1   Running 0        29m
kube-system   kube-scheduler-lemaitre             1/1   Running 1        44m
```

2 - Créer un pod

Il est possible de déployer directement une image sur la ligne de commande :

```
root@maitre:~# kubectl run my-nginx --image nginx --restart Never
```

On peut exécuter des commandes dans le conteneur :

```
root@maitre:~# kubectl exec my-nginx -- ls
```

Il est aussi possible de s'attacher au terminal comme avec docker si l'on souhaite exécuter un Shel lpar exemple :

```
root@maitre:~# kubectl exec -it my-nginx -- bash
```

Il faudra simplement sortir avec « exit ».

On vérifie maintenant les pods :

```
root@maitre:~# kubectl get pod
NAME                        READY   STATUS    RESTARTS   AGE
my-nginx                    1/1     Running   0          12s
```

Pour obtenir plus d'informations, on ajoute l'option « -o wide » :

```
root@maitre:~# kubectl get pod -o wide
NAME     READY STATUS  RESTARTS AGE IP          NODE ...
my-nginx  1/1  Running 0        19s 10.244.1.7 kb2  ...
```

Pour obtenir les « logs » d'un pod :

```
root@maitre:~# kubectl logs my-nginx
...
```

On peut directement tester une connexion réseau :

```
root@maitre:~# curl -i 10.244.1.7
HTTP/1.1 200 OK
Server: nginx/1.17.6
Date: Thu, 21 Nov 2019 06:16:50 GMT
Content-Type: text/html
Content-Length: 612
Last-Modified: Tue, 19 Nov 2019 12:50:08 GMT
Connection: keep-alive
ETag: "5dd3e500-264"
Accept-Ranges: bytes

<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
...
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

Pour avoir des informations sur le pod :

```
root@maitre:~# kubectl describe pod my-nginx
Name:             my-nginx
Namespace:        default
Priority:         0
Service Account:  default
Node:             slave1/10.3.134.150
Start Time:       Tue, 15 Nov 2021 08:25:52 +0000
Labels:           <none>
Annotations:      <none>
Status:           Running
IP:               10.244.2.2
IPs:
  IP:  10.244.2.2
Containers:
  my-nginx:
    Container ID:   containerd://4fce5411496cb5d2f6378952fdc201279ecc24ee3e6
    Image:          nginx
    Image ID:       docker.io/library/nginx@sha256:d08d96402375f39d7c6a209d
    Port:           <none>
    Host Port:      <none>
    State:          Running
      Started:      Tue, 15 Nov 2022 08:25:59 +0000
    Ready:          True
    Restart Count:  0
    Environment:    <none>
    Mounts:
      /var/run/secrets/kubernetes.io/serviceaccount from kube-api-access-64pn9 (ro)
```

3 - Fichier de description

Il est plus pratique de passer par un fichier de description YAML qui :

- Peut débuter par les caractères « --- », qui ne sont pas nécessaires dans le cas d'une unique déclaration,
- Contient la version de l'API, ici **v1** : `apiVersion : v1`,
- Un champ « kind » précisant le type d'objet, ici un **Pod** : `kind: Pod`

Chaque description d'élément d'un fichier YAML contient deux sections :

- `metadata` : Pour donner des informations comme un nom (name) ou des labels,
- `spec` : Comme « specification », pour décrire l'entité,

Il est possible de demander à Kubernetes de ne pas exécuter la commande mais d'afficher le fichier yaml correspondant :

```
root@maitre:~# kubectl run my-nginx --image nginx --restart
Never --dry-run -o yaml
```

Nous pouvons maintenant supprimer le pod et vérifier ensuite :

```
root@maitre:~# kubectl delete pod my-nginx
pod "my-nginx" deleted
root@maitre:~# kubectl get pod -o wide
No resources found in default namespace.
```

On peut donc utiliser le fichier suivant :

```
apiVersion: v1
kind: Pod
metadata:
  name: my-nginx
spec:
  containers:
  - image: nginx
    name: my-nginx
```

Et de l'appliquer à l'aide de la commande :

```
root@maitre:~# kubectl apply -f pod.yaml
```

Pour ajouter les authentifications d'autorisation de téléchargement depuis le DockerHub :

```
apiVersion: v1
kind: Pod
metadata:
  name: my-nginx
spec:
  containers:
  - image: nginx
    name: my-nginx
  imagePullSecrets:
  - name: regcred
```

4 - Variables d'environnement

Il est possible de spécifier des variables d'environnement dans une sous-section « env » du container :

```
apiVersion: v1
kind: Pod
metadata:
  name: my-nginx
spec:
  containers:
  - image: nginx
    name: my-nginx
    env:
    - name: MAVAR
      value: "super"
```

5 - Credential

Pour utiliser un « credential » déclaré précédemment :

```
apiVersion: v1
kind: Pod
metadata:
  name: private-reg
spec:
  containers:
  - name: private-reg-container
    image: <your-private-image>
  imagePullSecrets:
  - name: regcred
```