

TP – Utilisation de Kubectl

Kubectl est LA commande en ligne utilisée pour interagir avec les clusters Kubernetes. Il vous permet de gérer et de contrôler les ressources présentes dans un cluster Kubernetes.

Create (créer) : Cette action est utilisée pour créer une nouvelle ressource dans le cluster Kubernetes. Vous spécifiez le fichier de configuration YAML ou JSON contenant les détails de la ressource que vous souhaitez créer, et kubectl l'ajoute au cluster.

```
ludo@kubernetes:~$ kubectl create deployment nginx --image=nginx:latest
```

Get (obtenir) : Cette action vous permet d'obtenir des informations sur les ressources présentes dans le cluster. Par exemple, vous pouvez utiliser "kubectl get nodes" pour obtenir la liste des nœuds (nodes) du cluster, ou "kubectl get pods" pour obtenir la liste des pods en cours d'exécution.

```
ludo@kubernetes:~$ kubectl get deployments
```

```
ludo@kubernetes:~$ kubectl get deployments -o wide
```

Describe (décrire) : Cette action vous fournit des informations détaillées sur une ressource spécifique. Par exemple, "kubectl describe pod <nom-du-pod>" affiche des informations détaillées sur un pod spécifique, telles que son état, ses événements, ses conteneurs, etc.

```
ludo@kubernetes:~$ kubectl describe deployment nginx-deployment
```

Delete (supprimer) : Cette action vous permet de supprimer une ressource du cluster. Vous pouvez utiliser "kubectl delete" suivi du type de ressource et de son nom pour supprimer la ressource spécifiée. Par exemple, "kubectl delete pod <nom-du-pod>" supprime un pod spécifique.

```
ludo@kubernetes:~$ kubectl delete deployment nginx-deployment
```

Apply (appliquer) : Cette action est utilisée pour créer ou mettre à jour des ressources à partir d'un fichier de configuration. Vous spécifiez le fichier de configuration YAML ou JSON et kubectl applique les modifications au cluster en conséquence. Cela permet de créer des ressources ou de mettre à jour des ressources existantes de manière déclarative.

Quelques ressources spécifiques :

Node (nœud) : Un nœud est une machine physique ou virtuelle qui fait partie du cluster Kubernetes. Il exécute les pods et les services.

```
ludo@kubernetes:~$ kubectl get nodes  
  
ludo@kubernetes:~$ kubectl get node -o wide
```

Namespace (espace de noms) : Un espace de noms est une façon de diviser un cluster Kubernetes en plusieurs parties logiques. Il permet d'isoler les ressources et les objets au sein d'un cluster.

```
ludo@kubernetes:~$ kubectl create namespace formation  
  
ludo@kubernetes:~$ kubectl get namespaces  
  
ludo@kubernetes:~$ kubectl get ns
```

Pod : Un pod est l'unité de base dans Kubernetes. Il représente un ou plusieurs conteneurs qui partagent un environnement d'exécution commun. Les pods sont déployés sur les nœuds du cluster.

```
ludo@kubernetes:~$ kubectl run nginx --image=nginx:latest  
  
ludo@kubernetes:~$ kubectl get pods nginx  
  
ludo@kubernetes:~$ kubectl pods nginx -o wide  
ludo@kubernetes:~$ kubectl describe pod nginx  
ludo@kubernetes:~$ kubectl delete pod nginx
```

Deployment (déploiement) : Un déploiement est une ressource Kubernetes qui définit comment les pods doivent être créés et mis à l'échelle. Il facilite le déploiement et la gestion des applications dans le cluster.

```
ludo@kubernetes:~$ kubectl create deployment nginx --image=nginx:latest  
  
ludo@kubernetes:~$ kubectl get deployments  
  
ludo@kubernetes:~$ kubectl get deployments -o wide  
  
ludo@kubernetes:~$ kubectl describe deployment nginx-deployment  
ludo@kubernetes:~$ kubectl delete deployment nginx-deployment
```

Créer des alias

Vous pouvez créer des alias pour simplifier l'utilisation de la commande kubectl. Voici quelques exemples d'alias couramment utilisés

```
ludo@kubernetes:~$ alias kga='kubectl get all' >> .bashrc

ludo@kubernetes:~$ alias kapply='kubectl apply -f' >> .bashrc

ludo@kubernetes:~$ alias kdel='kubectl delete ' >> .bashrc

ludo@kubernetes:~$ source .bashrc
```