KUBECTL CHEAT SHEET

LA COMMANDE GET

`get namespaces` permet d'obtenir la liste des namespaces
`get pods` permet d'obtenir la liste des pods
`get services` permet d'obtenir la liste des services
`get deployments` permet d'obtenir la liste des déploiements

L'argument `-n <ns>` permet de se concentrer sur un namespace

La commande DELETE fonctionne de la même façon que la commande GET, sauf qu'elle supprime et qu'elle est suivie du nom de la cible

LES COMMANDES "D'INSTANCES"

`apply -f <filepath>` déploie les applications configurées dans le fichier indiqué en paramètre
`apply -f <path>` déploie l'ensemble des fichiers de configuration dans le dossier indiqué en paramètre
`apply -f <filepath> --dry-run=server|client` permet de simuler le déploiement de l'application
`create deployment <nom> --image=<image>` permet de déployer une image dans le cluster
`expose deployment <nom> --port=<port> --type=NodePort` permet d'exposer un déploiement dans le cluster

LA COMMANDE CONFIG

`config set <prop> <value>` permet de définir une valeur spécifique pour un paramètre dans le fichier Kube-config
`config get-clusters` permet d'obtenir la liste de clusters enregistrés dans la configuration
`config set-clusters` permet d'enregistrer un nouveau cluster Kubernetes dans la configuration kube
`config get-contexts` permet d'obtenir la liste de contexts enregistrés dans la configuration Kube
`config set-context` permet d'enregistrer un nouveau context dans la configuration Kubernetes
`config use-context <ctx>` configure l'environnement Kubernetes avec lequel la CLI interagit.
`config unset <prop>` supprime une valeur de la configuration Kubernetes
`config view` affiche la configuration Kubernetes dans la CLI (sans les informations sensibles)

COMMANDES UTILES

`port-forward pod/mypod <port ext.>:<port int.>` permet de faire un port-forward de la machine vers le pod
`exec <pod> -- <cmd>` permet d'exécuter une commande dans le pod

OUTILS SYMPAS POUR COMPLETER KUBECTL

K9S, une TUI pour Kubernetes

kubectx & kubens, un outil pour switch de context/namespaces rapidement

Kubernetes (for VSCode), une interface Kubernetes directement dans VSCode

POINTS DE VIGILANCE / BONNES PRATIQUES

Quand on travaille sur plusieurs contexts (ex: Dev/Preprod/Prod), il est préférable de s'assurer qu'on est sur le contexte souhaite, surtout si on réalise des actions qui pourrait avoir des effets de bord.
Faire un "dry run" avant un apply peut permettre de s'assurer que l'application va bien se déployer sur le cluster
Il faut essayer de mettre le moins de container possible dans un pod, si possible, pas plus d'un seul.
Configurer les routes de readyness & liveness permet d'assurer la haute disponibilité de l'application
Mettre en place le RBAC (Role-Based Access Control) dès la mise en place du cluster
Définir des limites de ressources (mémoire et cpu) pour éviter une sur-consommation de ressources
Monitorer tout le plus possible avec un outil adapté (avec prometheus et grafana, par exemple)

● = critique

Par Jules Pillot