A bit of context: Kubectl?

"Kubectl is a command-line interface that lets you run commands on Kubernetes clusters." This is how we will be able to perform various operations on our cluster.

Kubectl depends on a kubeconfig. This is a configuration file for access to one or more clusters. We will talk about context, to know which cluster is configured our Kubectl command, we can use:

- **Kubectl config current-context**

Or if we want to change clusters in our config, we can use:

- **Kubectl config use-context**

Kubectl monitoring commands

- **Kubectl get pods**

This command lists pods on the Kubernetes cluster. This command works for all types of Kubernetes resources: pods, services, deployments, cronjobs, events, ingresses, etc. We can also add parameters:

--all-namespaces: List all resources of all namespaces.

-o wide: List all resources with more details.

- **Kubectl describe pod**

The describe command gives a verbose display of the pod unlike the get and basic display. This allows having the events, useful when a pod does not start.

e.g. Kubectl describe pods my-pod.

- **Kubectl logs [-f] POD [-c CONTAINER]**

This command displays the logs of your POD. We can add the -c container option when we want to display the logs of a multi-container pod. The -f command displays the output of the logs continuously (stream).

Example: Kubectl logs -f my_pod -c my_app

-> Stream the logs of the container my_app on the my_pod pod.

- **Kubectl top pod POD_NAME --containers**

Displays the metrics for a given pod and its containers within a Kubernetes cluster.

Kubectl Create / Delete Commands

- **Kubectl create -f FILE**

Create one or more resources from your file or folder.

- **Kubectl apply -f FILE**

Applies a configuration change to a resource from your file.

- **Kubectl delete (-f FILE | TYPE [PREFIX_NAME | NAME])**

Deletes one or more Kubernetes resources from a configuration file or directly from resource names.

e.g. Kubectl delete my_pod (destroy the pod on the cluster named my_pod)

- **Kubectl port-forward POD [LOCAL_PORT:]REMOTE_PORT**

Lets you expose a local port to the port of a POD that is running on the Kubernetes cluster. Useful to debug.

e.g. Kubectl port-forward my_pod 80:3000 (exposes the port 3000 of the pod my_pod on our local port 80)

- **Kubectl run NAME --image=image [--env="key=value"] [--port=port] [--replicas=replicas]**

Run a resource in the Kubernetes cluster.

e.g. Kubectl run -i --tty busybox --image=busybox -- sh

-> Run a pod as an interactive shell