**kubkubectlLes commandes essentielles sur Kubernetes**

**Reduire le nombre des nodes à zéro pour sauvegarder les crédits**

*gcloud container clusters resize adme-cluster --zone us-central1-c --node-pool=pool-1 --num-nodes=0*

*gcloud container clusters resize adme-cluster --zone us-central1-c --node-pool=default-pool --num-nodes=0*

**Pour redemarrer le cluster**

gcloud container clusters resize adme-cluster --zone us-central1-c --node-pool=default-pool --num-nodes=3

gcloud container clusters resize adme-cluster --zone us-central1-c --node-pool=pool-1 --num-nodes=5

**Créer un deploiement**

kubectl create deployment <name\_of\_deployment> --image=<docker\_hub\_image>

**Exposer un deploiement pour qu'il soit visible ailleurs**

kubectl expose <name\_deployment> --type=LoadBalancer --port=<port\_number>

**Pour voir les evenements pour quand on crée un deploiement**

kubectl get events

**Pour arranger les events par date d’execution**

kubectl get events --sort-by=.metadata.creationTimestamp

**Pour voir les deploiements**

kubectl get deployments

**Pour voir les replicasets**

kubectl get replicaset

**Pour voir les services**

kubectl get services

**Pou avoir une explication sur les commandes**

kubectl explain <name\_of\_responsibility>

**Pour scale**

kubectl scale deployment <name\_of\_deployment> --replicas=<number\_of\_rs> (***Le nombre d’instances***)

**Pour deployer un nouveau image**

kubectl set image deployment <name\_of\_deployment> <name\_of\_container>=<image\_from\_docker\_hub>

**Pour deployer une nouvelle image et enregistre les raisons (commandes)**

kubectl set image deployment <name\_of\_deployment> <name\_of\_container>=<image\_from\_docker\_hub> --record

**Pour inspecter l'etat d'un deploiement en cours d'execution, de suspendre et de reprendre un déploiement, d'annuler une mise à jour et d'afficher l'historique de déploiement d'un objet.**

* ***Historique***
* kubectl rollout history deployment <name\_of\_deployment>
* ***voir le statut***
* Kubectl rollout history deployment <name\_of\_deployment>
* **Undo deployment**
* Kubectl rollout undo deployment <name\_of\_deployment>
* **Pause**
* Kubectl rollout pause deployment <name\_of\_deployment>

**Pour voir les logs d’un pod**

* Kubectl logs <pod\_id>

**Pour voir le fichier. yaml pour un deploiement ou un service**

* Kubectl get <deployment\_or\_service> <name\_of\_service\_or\_deployment> -o yaml
* **Pour les enregistrer dans un fichier. yaml**
* Kubectl get <deployment\_or\_service> <name\_of\_service\_or\_deployment> -o yaml > <service\_or\_deployment>.yaml

**Pour supprimer tous les composants (pods, deployment, service, replicaset) d’un deploiement en utilisant un label**

* Kubectl delete all –l app=<name\_of\_deploiement>

**Pour deployer une application à partir d’un fichier deployment.yaml**

* **Kubectl apply –f deployement.yaml**

**ConfigMap**

Enregistrer des valeurs non sensibles de format Cle-valeur dans des fichiers centralisés comme le nom de la BD, le port de connexion et autres informations pouvant etre partagées par différents pods

***Cmd*** : kubectl create configmap <name\_of\_configmap> --from-literal=<name\_of\_key>=value

Pour modifier un configmap

Cmd: kubectl edit configmap/<name\_of\_configmap>

Pour afficher un configmap

Cmd: kubectl get configmap <name\_of\_config>

Pour afficher les informations des cles-valeurs

Cmd : kubectl describe configmap <name\_of\_configmap>

**Secret**

Enregistrer des informations sensibles (Mot de passe, token, etc…) dans un cluster kubernetes sous format cles-valeurs.

***Cmd :*** kubectl create secret generic <name\_of\_secret> --from-literal=<name\_of\_key>=<value>

**ClusterIP**

Autoriser la connexion des IP qui font partie du cluster. Par exemple, autoriser seulement la connexion avec la base de données à l’intérieur du cluster

Changer le Type : LoadBalancer to ClusterIP

**CPU and Memories**

Pour voir le nombre de CPUs et de memoires

***Cmd*** : kubectl top pods

**AutoScaling**

Pour activer le scale horizontal

Cmd : ***kubectl autoscale deployment <name of deployment> --min=1 –max=3 (Number of pods) --cpu-percent=10 (Pourcentage du CPU)***. Si le pourcentage est atteint, Kubernetes cree automatiquement un nouveau pod.

Pour tester si votre application Spring Boot est en execution et prêt, il faut utiliser un Probe (Faire diagnostique du container par kubelet)