



REVERSE ENGINEERING – Déploiement & Design - SAGNANE Saïdou (5SRC2)

Objectifs

- Déployer tous les services de l'application : `vote`, `result`, `worker`, `redis` et `postgres`
- Vérifier le bon fonctionnement de l'application en testant l'interaction entre les interfaces `vote` et `result`

Étapes de déploiement

```
git clone https://github.com/ssagnane1/example-voting-app.git
```

```
ssagnane@ubuntu:~/example-voting-app$ git clone https://github.com/ssagnane1/example-voting-app.git
Cloning into 'example-voting-app'...
remote: Enumerating objects: 1160, done.
remote: Total 1160 (delta 0), reused 0 (delta 0), pack-reused 1160 (from 1)
Receiving objects: 100% (1160/1160), 1.20 MiB | 9.73 MiB/s, done.
Resolving deltas: 100% (437/437), done.
ssagnane@ubuntu:~/example-voting-app$ ls
example-voting-app  minikube  mini-projet-5esgi  projet-esgi
ssagnane@ubuntu:~/example-voting-app$ cd example-voting-app/
ssagnane@ubuntu:~/example-voting-app$ |
```

Lancer Minkube

```
minikube start
```

```
ssagnane@ubuntu:~/example-voting-app$ minikube start
🕒 minikube v1.35.0 sur Ubuntu 24.04
💡 Utilisation du pilote docker basé sur le profil existant
👉 Démarrage du nœud "minikube" primary control-plane dans le cluster "minikube"
🌐 Extraction de l'image de base v0.0.46...
🕒 Mise à jour du container docker en marche "minikube" ...
🌐 Préparation de Kubernetes v1.32.0 sur Docker 27.4.1...
🌐 Vérification des composants Kubernetes...
  * Utilisation de l'image gcr.io/k8s-minikube/storage-provisioner:v5
* Modules activés: storage-provisioner, default-storageclass
🕒 Terminé ! kubectl est maintenant configuré pour utiliser "minikube" cluster et espace de noms "default" par défaut.
ssagnane@ubuntu:~/example-voting-app$ |
```

Lancement des services Kubernetes

Les fichiers de déploiement se trouvent dans le dossier **k8s-specifications**.

```
ssagnane@ubuntu:~/example-voting-app/k8s-specifications$ tree
.
├── db-deployment.yaml
├── db-service.yaml
├── redis-deployment.yaml
├── redis-service.yaml
├── result-deployment.yaml
├── result-service.yaml
├── vote-deployment.yaml
├── vote-service.yaml
└── worker-deployment.yaml
```

```
sudo kubectl create -f k8s-specifications/
```

```
ssagnane@ubuntu:~/example-voting-app$ sudo kubectl create -f k8s-specifications/
[sudo] password for ssagnane:
deployment.apps/db created
service/db created
deployment.apps/redis created
service/redis created
deployment.apps/result created
service/result created
deployment.apps/vote created
service/vote created
deployment.apps/worker created
ssagnane@ubuntu:~/example-voting-app$ |
```

Vérification des services

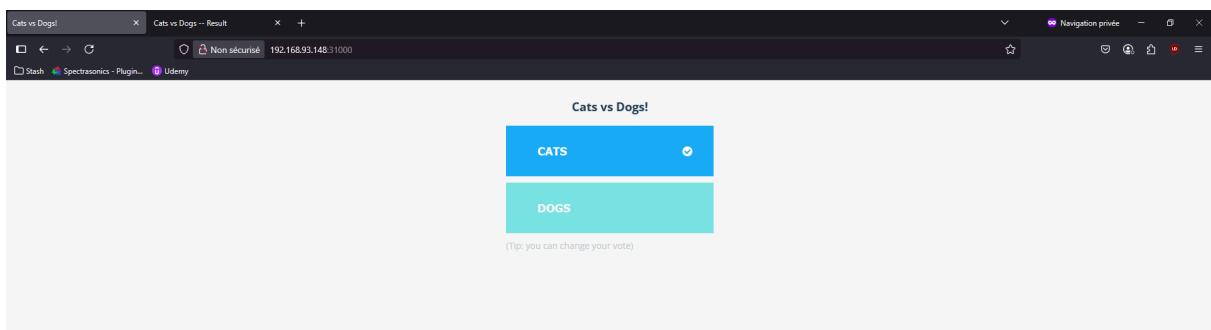
```
sudo kubectl get svc
```

```
sagnane@ubuntu:~/example-voting-app$ sudo kubectl get svc
NAME           TYPE      CLUSTER-IP   EXTERNAL-IP  PORT(S)        AGE
db             ClusterIP 10.43.54.20  <none>       5432/TCP     75s
kubernetes     ClusterIP 10.43.0.1    <none>       443/TCP      58d
paymybuddy-db-service ClusterIP 10.43.248.218 <none>       3306/TCP     58d
paymybuddy-service  NodePort   10.43.30.207  <none>       8080:30229/TCP 58d
redis          ClusterIP 10.43.13.29  <none>       6379/TCP     74s
result          NodePort   10.43.159.164 <none>       8081:31001/TCP 74s
vote            NodePort   10.43.121.84  <none>       8080:31000/TCP 74s
sagnane@ubuntu:~/example-voting-app$ |
```

Résultat

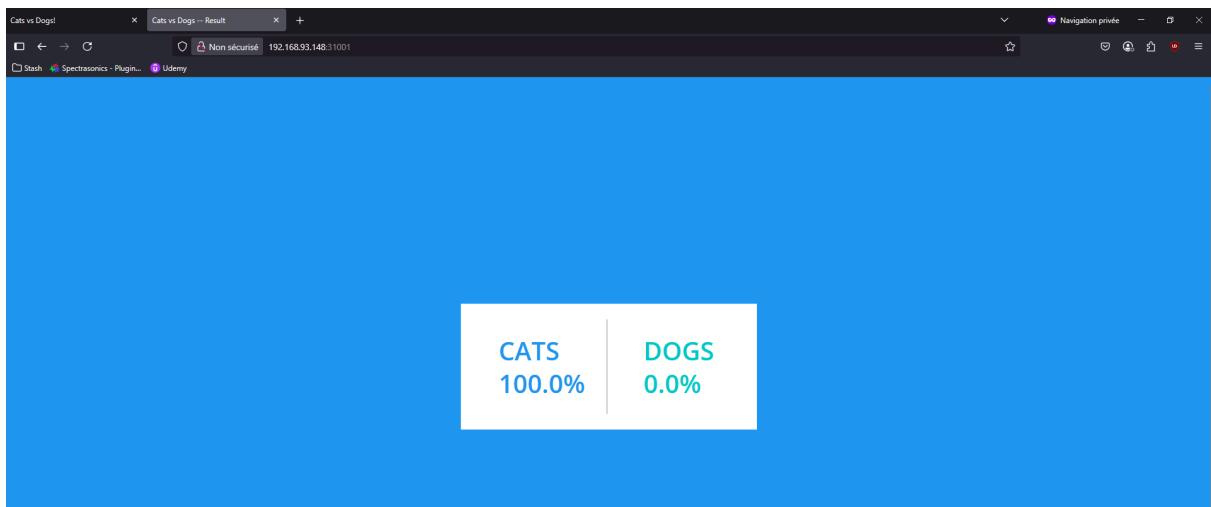
Frontend vote

192.168.93.148:31000



Le frontend **vote** permet de voter pour une option.

Frontend result



Le frontend **result** affiche les résultats en temps réel dès qu'un vote est validé.

Nettoyage ressources

Pour supprimer les ressources:

```
sudo kubectl delete -f k8s-specifications/
```

```
ssagnane@ubuntu:~/example-voting-app$ sudo kubectl delete -f k8s-specifications/
deployment.apps "db" deleted
service "db" deleted
deployment.apps "redis" deleted
service "redis" deleted
deployment.apps "result" deleted
service "result" deleted
deployment.apps "vote" deleted
service "vote" deleted
deployment.apps "worker" deleted
ssagnane@ubuntu:~/example-voting-app$ |
```

Conclusion

Ce projet m'a permis de comprendre comment déployer une application multi-conteneurs via Kubernetes, tester l'interopérabilité entre plusieurs services, et automatiser un déploiement complet avec des fichiers YAML.