

Deploiement de la VotingApp

Exercice

Dans cet exercice vous allez déployer la Voting App, une application de vote très souvent utilisée pour les démos et présentation

1. Récupération du projet

Cloner le repository avec la commande suivante

```
$ git clone https://github.com/docker-samples/example-voting-app
$ cd example-voting-app
```

2. Création des ressources

Dans le répertoire *k8s-specifications* se trouvent l'ensemble des spécifications des ressources.

Examinez chacun des fichiers de spécification, quelles sont les ressources en jeu pour chaque micro-service ?

Avec *kubectl* créer l'ensemble de ces ressources en une seule fois.

Note: il vous faudra créer le namespace nommé "vote" au préalable avec la commande suivante:

```
kubectl create namespace vote
```

Ceci est nécessaire car l'application sera lancée dans ce contexte d'isolation (nous reviendrons sur les namespaces dans un prochain chapitre).

3. Liste des ressources

Listez les différentes ressources créées.

4. Accès à l'application

Lancez un navigateur sur l'interface de vote.

Note: l'IP est celle de minikube, le port est défini dans la spécification du Service *vote*

Sélectionnez une option et visualisez le résultat dans l'interface *result*.

Correction

2. Création des ressources

Pour chaque micro-service de l'application, il y a un Deployment et un Service. Seul le micro-service *worker* n'a pas de Service associé, c'est le seul micro-service qui n'est pas exposé dans le cluster.

La commande suivante permet de créer l'ensemble des ressources:

```
$ kubectl create -f ./k8s-specifications
deployment "db" created
service "db" created
deployment "redis" created
service "redis" created
deployment "result" created
service "result" created
deployment "vote" created
service "vote" created
deployment "worker" created
```

3. Liste des ressources

La commande suivante permet de lister les Deployments, Pods et Services créés.

```
$ kubectl get deplpy,pod,svc
```

NAME	DESIRED	CURRENT	UP-TO-DATE	AVAILABLE	AGE
deploy/db	1	1	1	1	1m
deploy/redis	1	1	1	1	1m
deploy/result	1	1	1	1	1m
deploy/vote	1	1	1	1	1m
deploy/worker	1	1	1	1	1m

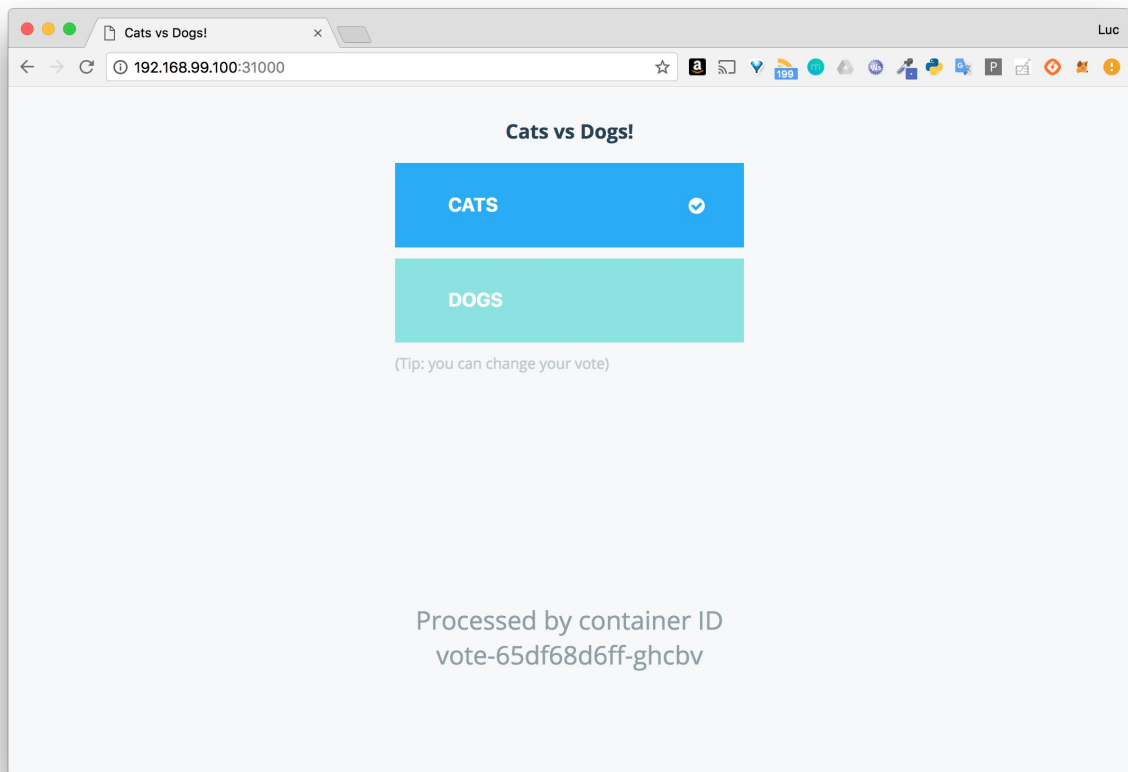
NAME	READY	STATUS	RESTARTS	AGE
po/db-549c4694d9-td9gj	1/1	Running	0	1m
po/redis-5ff865c7d-bxhd9	1/1	Running	0	1m

po/result-76784c98fb-4mq57	1/1	Running	0	1m
po/vote-65df68d6ff-ghcbv	1/1	Running	0	1m
po/worker-8875fdcc8-zbx12	1/1	Running	0	1m

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
svc/db	ClusterIP	10.99.192.60	<none>	5432/TCP	1m
svc/redis	ClusterIP	10.111.62.16	<none>	6379/TCP	1m
svc/result	NodePort	10.107.254.26	<none>	5001:31001/TCP	1m
svc/vote	NodePort	10.99.171.171	<none>	5000:31000/TCP	1m

4. Accès à l'application

L'interface de vote est disponible sur le port *31000*



L'interface de result est disponible sur le port *31001*

