

Cloud Kubernetes Kevin Bütikofer, Charles-Lewis Jaggi # Task 1 ## Redis ###
Déploiement

Pod :

Voir redis-pod.yaml

Service :

Voir redis-svc.yaml

```
kubectl.exe create -f redis-svc.yaml --validate=false  
kubectl.exe create -f redis-pod.yaml --validate=false``
```

API

Déploiement

Pod :

Voir api-pod.yaml

Service :

Voir api-svc.yaml

```
```bash=  
kubectl.exe create -f api-svc.yaml --validate=false
kubectl.exe create -f api-pod.yaml --validate=false
```

## Frontend

### Déploiement

Pod :

Voir frontend-pod.yaml

**Question:** L'adresse locale du service svc/api-svc

Service :

Voir frontend-svc.yaml

```
kubectl.exe create -f frontend-svc.yaml --validate=false
kubectl.exe create -f frontend-pod.yaml --validate=false
```

## Résultat

```
$ kubectl.exe get all
NAME READY STATUS RESTARTS AGE
```

po/api	1/1	Running	0	16m
po/frontend	1/1	Running	0	15m
po/redis	1/1	Running	0	16m

NAME	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
svc/api-svc	10.100.48.138	<none>	8081/TCP	16m
svc/frontend-svc	10.100.115.90	af0cb46e1ef1b...	80:30032/TCP	15m
svc/redis-svc	10.100.163.85	<none>	6379/TCP	16m

## Describe

```
$ kubectl.exe describe svc/frontend-svc
Name: frontend-svc
Namespace: group-12-ns
Labels: component=frontend
Annotations: <none>
Selector: app=todo,component=frontend
Type: LoadBalancer
IP: 10.100.115.90
LoadBalancer Ingress: af0cb46e1ef1b11e9b8e706ae75a39b7-477009224.eu-west-1.elb.amazonaws.com
Port: frontend 80/TCP
NodePort: frontend 30032/TCP
Endpoints: 192.168.27.196:8080
Session Affinity: None
```

## Task 2 Resilience

### Task 2.1

#### Redis

**Question** Pour qu'il y ait une seule version des données.

Deployment : Voir redis-deploy.yaml

```
kubectl.exe create -f redis-svc.yaml --validate=false
kubectl.exe create -f redis-deploy.yaml --validate=false
```

#### API

Deployment : Voir api-deploy.yaml

```
kubectl.exe create -f api-svc.yaml --validate=false
kubectl.exe create -f api-deploy.yaml --validate=false
```

## Front-End

Deployment: voir frontend-deploy.yaml

```
kubectl.exe create -f frontend-svc.yaml --validate=false
kubectl.exe create -f frontend-deploy.yaml --validate=false
```

## Résultats

```
$ kubectl.exe get all
```

NAME	READY	STATUS	RESTARTS	AGE
po/api-deploy-8b899f8f5-6zt26	1/1	Running	0	10m
po/api-deploy-8b899f8f5-hgbj7	1/1	Running	0	10m
po/frontend-deploy-99fb8bb6c-mt6j4	1/1	Running	0	8m
po/frontend-deploy-99fb8bb6c-wjmbv	1/1	Running	0	8m
po/redis-deploy-6c8d7db57b-mks5s	1/1	Running	0	14m

NAME	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
svc/api-svc	10.100.118.191	<none>	8081/TCP	10m
svc/frontend-svc	10.100.50.223	a593dc9c5ef22...	80:30359/TCP	10m
svc/redis-svc	10.100.200.71	<none>	6379/TCP	19m

NAME	DESIRED	CURRENT	UP-TO-DATE	AVAILABLE	AGE
deploy/api-deploy	2	2	2	2	10m
deploy/frontend-deploy	2	2	2	2	8m
deploy/redis-deploy	1	1	1	1	14m

NAME	DESIRED	CURRENT	READY	AGE
rs/api-deploy-8b899f8f5	2	2	2	10m
rs/frontend-deploy-99fb8bb6c	2	2	2	8m
rs/redis-deploy-6c8d7db57b	1	1	1	14m

## Task 2.2

### Questions

**What happens if you delete a Frontend or API Pod? How long does it take for the system to react?** Une nouvelle instance est créée lorsque que le pod est en état terminating. Le nouveau pod est d'abord en état pending puis il crée le conteneur puis il passe dans l'état running.

```
$ kubectl.exe get pods --watch
```

NAME	READY	STATUS	RESTARTS	AGE
api-deploy-8b899f8f5-6zt26	1/1	Running	0	12m
api-deploy-8b899f8f5-hgbj7	1/1	Running	0	12m
frontend-deploy-99fb8bb6c-mt6j4	1/1	Running	0	10m
frontend-deploy-99fb8bb6c-wjmbv	1/1	Running	0	10m

frontend-deploy-99fb8bb6c-mt6j4	1/1	Terminating	0	11m
frontend-deploy-99fb8bb6c-jpzxv	0/1	Pending	0	1s
frontend-deploy-99fb8bb6c-jpzxv	0/1	Pending	0	1s
frontend-deploy-99fb8bb6c-jpzxv	0/1	ContainerCreating	0	1s
frontend-deploy-99fb8bb6c-mt6j4	0/1	Terminating	0	11m
frontend-deploy-99fb8bb6c-jpzxv	1/1	Running	0	4s
frontend-deploy-99fb8bb6c-mt6j4	0/1	Terminating	0	11m
frontend-deploy-99fb8bb6c-mt6j4	0/1	Terminating	0	11m

**What happens when you delete the Redis Pod?** La base de donnée sera supprimé. Le pod est recréé automatiquement mais avec une base de donnée vide. L'api n'arrive plus accéder à la base de données

```
$ kubectl.exe get pods --watch
```

NAME	READY	STATUS	RESTARTS	AGE
api-deploy-8b899f8f5-hgbj7	1/1	Running	0	17m
api-deploy-8b899f8f5-lrs9j	1/1	Running	0	2m
frontend-deploy-99fb8bb6c-jpzxv	1/1	Running	0	4m
frontend-deploy-99fb8bb6c-wjmbv	1/1	Running	0	15m
redis-deploy-6c8d7db57b-mks5s	1/1	Terminating	0	21m
redis-deploy-6c8d7db57b-mjgn7	0/1	Pending	0	2s
redis-deploy-6c8d7db57b-mjgn7	0/1	Pending	0	2s
redis-deploy-6c8d7db57b-mjgn7	0/1	ContainerCreating	0	2s
redis-deploy-6c8d7db57b-mks5s	0/1	Terminating	0	21m
redis-deploy-6c8d7db57b-mjgn7	1/1	Running	0	5s
redis-deploy-6c8d7db57b-mks5s	0/1	Terminating	0	21m
redis-deploy-6c8d7db57b-mks5s	0/1	Terminating	0	21m

**How can you change the number of instances temporarily to 3? Hint:** look for scaling in the deployment documentation

```
kubectl scale app --replicas=3
```

**What autoscaling features are available? Which metrics are used?**

Avec la commande autoscale on spécifie le nombre minimal et maximal de replicas puis le pourcentage d'occupation cpu :

```
kubectl autoscale app --min=10 --max=15 --cpu-percent=80
```

**How can you update a component? (see update in the deployment documentation)** On peut utiliser la commande set image en lui donnant la nouvelle image.:

```
kubectl set image app nginx=nginx:1.91 --record
```

On peut aussi modifier le fichier yaml et faire la commande:

```
kubectl edit app
```