

Compte Rendu - TP2 Kubernetes

Étudiant : Jules DEBÉE

Introduction

Ce document présente les étapes suivies lors du TP Kubernetes, portant sur le déploiement de MySQL, WordPress et NGINX, ainsi que la vérification des différentes ressources et configurations. Le TP comprend la création de namespaces, l'utilisation de **ConfigMap**, **Secrets**, et la configuration de services pour ces applications.

Capture d'écran

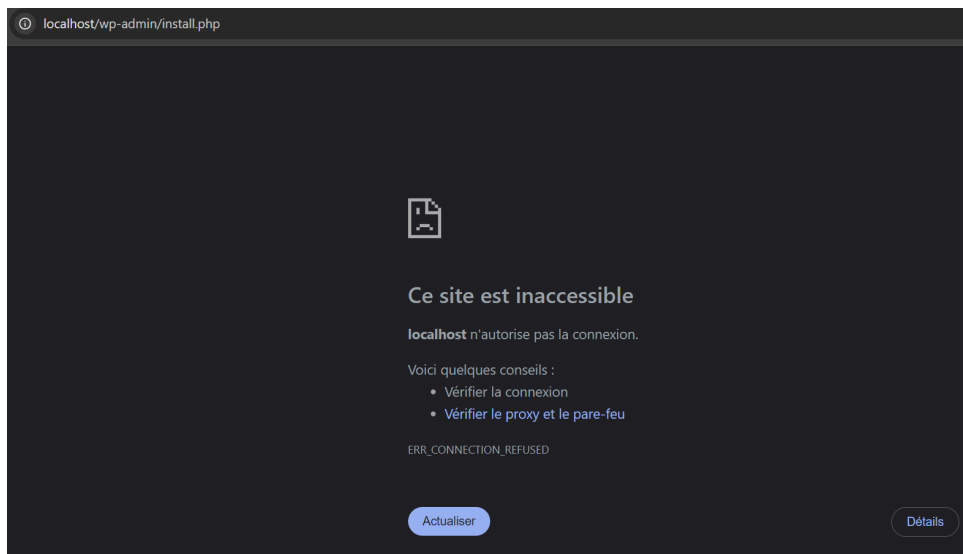
1. Connexion au pod MySQL et vérifier la présence des 4 variables d'environnement avec les valeurs définies dans les ressources ConfigMap et Secret :

```
jules@LEGIONJULES: /mnt/c/Users/debee/Cours/3eme_annee/Virtualisation/TP/TP-Kubernetes/TP2/TP$ kubectl exec -it mysql-statefulset-0 -n database -- /bin/bash
bash-5.1# env | grep MYSQL
MYSQL_MAJOR=innovation
MYSQL_SERVICE_PORT_3306_TCP_ADDR=10.109.97.174
MYSQL_ROOT_PASSWORD=123456
MYSQL_SERVICE_SERVICE_HOST=10.109.97.174
MYSQL_PASSWORD=123456
MYSQL_USER=Jules
MYSQL_VERSION=9.0.1-1.el9
MYSQL_SERVICE_PORT=tcp://10.109.97.174:3306
MYSQL_DATABASE=wordpress
MYSQL_SERVICE_PORT_3306_TCP=tcp://10.109.97.174:3306
MYSQL_SERVICE_PORT_3306_TCP_PORT=3306
MYSQL_SERVICE_SERVICE_PORT=3306
MYSQL_SHELL_VERSION=9.0.1-1.el9
MYSQL_SERVICE_PORT_3306_TCP_PROTO=tcp
bash-5.1#
```

2. Connexion au pod Wordpress et vérifier la présence des 4 variables d'environnement avec les valeurs définies :

```
jules@LEGIONJULES: /mnt/c/Users/debee/Cours/3eme_annee/Virtualisation/TP/TP-Kubernetes/TP2/TP$ kubectl get pods -n middle
NAME                                READY   STATUS    RESTARTS   AGE
wordpress-cd654d6b4-gfdcb          1/1     Running   0           4m50s
jules@LEGIONJULES: /mnt/c/Users/debee/Cours/3eme_annee/Virtualisation/TP/TP-Kubernetes/TP2/TP$ kubectl exec -it wordpress-cd654d6b4-gfdcb -n middle -- env | grep WORDPR
ESS
WORDPRESS_DB_HOST=mysql.database.svc.cluster.local
WORDPRESS_DB_USER=Jules
WORDPRESS_DB_NAME=wordpress
WORDPRESS_DB_PASSWORD=123456
WORDPRESS_SERVICE_PORT=80
WORDPRESS_PORT=tcp://10.102.3.191:80
WORDPRESS_PORT_80_TCP_PROTO=tcp
WORDPRESS_PORT_80_TCP_PORT=80
WORDPRESS_PORT_80_TCP_ADDR=10.102.3.191
WORDPRESS_SERVICE_HOST=10.102.3.191
WORDPRESS_PORT_80_TCP=tcp://10.102.3.191:80
jules@LEGIONJULES: /mnt/c/Users/debee/Cours/3eme_annee/Virtualisation/TP/TP-Kubernetes/TP2/TP$
```

3. Accéder à la page d'accueil de votre site depuis votre navigateur :



Note : Cela accède à wordpress mais demande les installations, vue avec le prof.

Question

? Les différentes ressources sont déployées dans le bon namespace ? Quelle commande utilisez-vous pour vérifier cela ?

kubectl get all -n front (ou middle ou database)

```
jules@LEGIONJULES: /mnt/c/Users/debee/Cours/3eme_annee/Virtualisation/tp-kubernetes-main/TP$ kubectl get all -n front
NAME                                READY   STATUS    RESTARTS   AGE
pod/nginx-deployment-656fd46f57-gb9fm 1/1     Running   1 (10m ago) 13d
pod/nginx-deployment-656fd46f57-hpq8x 1/1     Running   1 (10m ago) 13d
pod/nginx-deployment-656fd46f57-lxkw2 1/1     Running   1 (10m ago) 13d

NAME                TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)    AGE
service/nginx-svc   ClusterIP   10.101.8.91  <none>        8383/TCP    13d

NAME                                READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/nginx-deployment  3/3     3             3           13d

NAME                                DESIRED   CURRENT   READY   AGE
replicaset.apps/nginx-deployment-656fd46f57 3         3         3       13d
replicaset.apps/nginx-deployment-6884d88685 0         0         0       13d
```

On peut également lister les ConfigMaps et Secrets pour vérifier leur existence dans le bon namespace avec :

kubectl get configmap -n middle

kubectl get secret -n middle

? Les ressources MySQL ConfigMap et MySQL sont à déployer au sein des namespaces middle et database, pourquoi ?

- **Namespace database** : Les ressources MySQL, comme le **StatefulSet** et le **Service**, sont déployées dans le namespace **database** car ce namespace est dédié à la gestion des ressources liées aux bases de données. Cela permet d'isoler les composants liés à MySQL dans un espace spécifique.
- **Namespace middle** : Les ConfigMaps et Secrets MySQL sont également déployés dans le namespace **middle** car WordPress (qui est déployé dans ce namespace) a

besoin de ces informations pour se connecter à la base de données. Cela permet à WordPress d'avoir accès aux configurations et secrets nécessaires sans interférer directement avec les composants MySQL dans le namespace **database**.

? Les différentes pods sont-ils correctement démarrés ? Quelle commande utilisez-vous pour vérifier cela ?

kubectl get pods -n front (ou database ou middle)

```
jules@LEGIONJULES:/mnt/c/Users/debee/Cours/3eme_annee/Virtualisation/tp-kubernetes-main/TP$ kubectl get pods -n front
NAME                                READY   STATUS    RESTARTS   AGE
nginx-deployment-656fd46f57-gb9fm   1/1     Running   1 (31m ago)  13d
nginx-deployment-656fd46f57-hpq8x   1/1     Running   1 (31m ago)  13d
nginx-deployment-656fd46f57-lxkw2   1/1     Running   1 (31m ago)  13d
```

Cette commande nous montrera l'état de chaque pod. Un état **Running** indique que le pod fonctionne correctement.

? Lorsqu'un pod est en erreur comment accédez-vous aux events du namespace et aux logs du pod ? Quelles commandes utilisez-vous pour vérifier cela ?

kubectl get events -n front (ou database ou middle)

```
jules@LEGIONJULES:/mnt/c/Users/debee/Cours/3eme_annee/Virtualisation/tp-kubernetes-main/TP$ kubectl get events -n front
LAST SEEN   TYPE      REASON              OBJECT                                                                 MESSAGE
13d         Normal    Scheduled            pod/nginx-deployment-656fd46f57-gb9fm   Successfully assigned front/nginx-deployment-656fd46f57-gb9fm to minikube
13d         Normal    Pulled               pod/nginx-deployment-656fd46f57-gb9fm   Container image "nginx:1.14.2" already present on machine
13d         Normal    Created              pod/nginx-deployment-656fd46f57-gb9fm   Created container nginx
13d         Normal    Started              pod/nginx-deployment-656fd46f57-gb9fm   Started container nginx
33m         Normal    SandboxChanged       pod/nginx-deployment-656fd46f57-gb9fm   Pod sandbox changed, it will be killed and re-created.
33m         Normal    Pulled               pod/nginx-deployment-656fd46f57-gb9fm   Container image "nginx:1.14.2" already present on machine
33m         Normal    Created              pod/nginx-deployment-656fd46f57-gb9fm   Created container nginx
33m         Normal    Started              pod/nginx-deployment-656fd46f57-gb9fm   Started container nginx
13d         Normal    Scheduled            pod/nginx-deployment-656fd46f57-hpq8x   Successfully assigned front/nginx-deployment-656fd46f57-hpq8x to minikube
13d         Normal    Pulled               pod/nginx-deployment-656fd46f57-hpq8x   Container image "nginx:1.14.2" already present on machine
13d         Normal    Created              pod/nginx-deployment-656fd46f57-hpq8x   Created container nginx
13d         Normal    Started              pod/nginx-deployment-656fd46f57-hpq8x   Started container nginx
33m         Normal    SandboxChanged       pod/nginx-deployment-656fd46f57-hpq8x   Pod sandbox changed, it will be killed and re-created.
33m         Normal    Pulled               pod/nginx-deployment-656fd46f57-hpq8x   Container image "nginx:1.14.2" already present on machine
33m         Normal    Created              pod/nginx-deployment-656fd46f57-hpq8x   Created container nginx
33m         Normal    Started              pod/nginx-deployment-656fd46f57-hpq8x   Started container nginx
13d         Normal    Scheduled            pod/nginx-deployment-656fd46f57-lxkw2   Successfully assigned front/nginx-deployment-656fd46f57-lxkw2 to minikube
13d         Normal    Pulled               pod/nginx-deployment-656fd46f57-lxkw2   Container image "nginx:1.14.2" already present on machine
13d         Normal    Created              pod/nginx-deployment-656fd46f57-lxkw2   Created container nginx
13d         Normal    Started              pod/nginx-deployment-656fd46f57-lxkw2   Started container nginx
33m         Normal    SandboxChanged       pod/nginx-deployment-656fd46f57-lxkw2   Pod sandbox changed, it will be killed and re-created.
33m         Normal    Pulled               pod/nginx-deployment-656fd46f57-lxkw2   Container image "nginx:1.14.2" already present on machine
33m         Normal    Created              pod/nginx-deployment-656fd46f57-lxkw2   Created container nginx
33m         Normal    Started              pod/nginx-deployment-656fd46f57-lxkw2   Started container nginx
13d         Normal    SuccessfulCreate      replicaset/nginx-deployment-656fd46f57   Created pod: nginx-deployment-656fd46f57-gb9fm
13d         Normal    SuccessfulCreate      replicaset/nginx-deployment-656fd46f57   Created pod: nginx-deployment-656fd46f57-lxkw2
13d         Normal    SuccessfulCreate      replicaset/nginx-deployment-656fd46f57   Created pod: nginx-deployment-656fd46f57-hpq8x
13d         Normal    Killing               pod/nginx-deployment-6884d88685-7gq4s   Stopping container nginx
13d         Normal    Killing               pod/nginx-deployment-6884d88685-cql82   Stopping container nginx
13d         Normal    Killing               pod/nginx-deployment-6884d88685-fccd8   Stopping container nginx
13d         Normal    SuccessfulDelete      replicaset/nginx-deployment-6884d88685   Deleted pod: nginx-deployment-6884d88685-cql82
13d         Normal    SuccessfulDelete      replicaset/nginx-deployment-6884d88685   Deleted pod: nginx-deployment-6884d88685-7gq4s
13d         Normal    SuccessfulDelete      replicaset/nginx-deployment-6884d88685   Deleted pod: nginx-deployment-6884d88685-fccd8
13d         Normal    ScalingReplicaSet     deployment/nginx-deployment              Scaled up replica set nginx-deployment-656fd46f57 to 1
13d         Normal    ScalingReplicaSet     deployment/nginx-deployment              Scaled down replica set nginx-deployment-6884d88685 to 2 from 3
13d         Normal    ScalingReplicaSet     deployment/nginx-deployment              Scaled up replica set nginx-deployment-656fd46f57 to 2 from 1
13d         Normal    ScalingReplicaSet     deployment/nginx-deployment              Scaled down replica set nginx-deployment-6884d88685 to 1 from 2
13d         Normal    ScalingReplicaSet     deployment/nginx-deployment              Scaled up replica set nginx-deployment-656fd46f57 to 3 from 2
13d         Normal    ScalingReplicaSet     deployment/nginx-deployment              Scaled down replica set nginx-deployment-6884d88685 to 0 from 1
```