Partie 2

Pré-requis

□ Démarrer Docker□ Démarrer Minikube :		
minikube startdriver=docker		
minikube status		
☐ Tester kubectl :		
kubectl version		

L'arborescence du projet avec quelques exemples est disponible dans le repertoire TP

ChatGPT et autres outils sont autorisés, mais peu utiles. Le code YAML retourné peut inclure des objets Kubernetes non vus en cours pour le moment.

Je reste disponible pour toute question, n'hésitez pas!

MYSQL

Les fichiers YAML concernant MySQL sont à déposer dans le dossier TP/database.

Les ressources Kubernetes concernant MySQL sont à déployer dans le namespace database.

1. Créer un fichier YAML mysql-secret.yml, ajoutez-y une ressource de type Secret avec ces différentes clé/valeur:

Description	Key	Value
Mot de passe de votre utilisateur de la BDD MySQL	MYSQL_PASSWORD	Au choix (encodé en base64)
Mot de passe de l'utilisateur root de la BDD MySQL	MYSQL_ROOT_PASSWORD	Au choix (encodé en base64)

Depuis Linux:

2. Créer un fichier YAML mysql-configmap.yml, ajoutez-y une ressource de type ConfigMap avec ces différentes clé/valeur:

| Description | Key | Value |

Votre utilisateur pour la BDD MySQL | MYSQL_USER | Au choix |

Schéma de la base de données | MYSQL_DATABASE | wordpress |

- 3. Créer un fichier YAML nommé mysql-statefulset.yml . Dans ce fichier, définissez une ressource de type StatefulSet avec :
- L'image docker mysql:9.0
- 1 replica
- Les variables définies dans la ressource ConfigMap en tant que variables d'environnement
- Les secrets définis dans la ressource Secret en tant que variables d'environnement
- Selector et Labels qui ont pour clé-valeur app: mysql
- Le stockage persistant fourni dans le fichier d'exemple (/TP/database/mysql-statefulset.yml)
- 4. Créer un fichier YAML nommé mysql-service.yml. Dans ce fichier, définissez une ressource de type Service qui écoutera sur le port 3306 et redirigera les requêtes vers le port 3306 des pods ayant le label app: mysql
- 5. Connexion au pod MySQL et vérifier la présence des 4 variables d'environnement avec les valeurs définies dans les ressources ConfigMap et Secret

MYSQL_DATABASE

MYSQL_USER

MYSQL_PASSWORD

MYSQL_ROOT_PASSWORD

Livrables attendus : 4 fichiers YAML, la liste des commandes utilisées, 1 capture d'écran

WORDPRESS

Les fichiers YAML concernant Wordpress sont à déposer dans le dossier TP/middle.

Les ressources Kubernetes concernant Wordpress sont à déployer dans le namespace middle.

1. Créer un fichier YAML wordpress-pv.yml.

Dans ce fichier, ajoutez-y une ressource de type PersistentVolume qui a pour nom wordpress-pv . Ce volume persistant doit avoir une capacité de 100Mi , un mode d'accès en

lecture/écriture pour l'ensemble des noeuds du cluster et les données seront stockées dans /tmp/kubernetes/wordpress

La storageClassName est de type manual

2. Créer un fichier YAML wordpress-pvc.yml.

Dans ce fichier, ajoutez-y une ressource de type PersistentVolumeClaim qui a pour nom wordpress-pvc. Cette demande de stockage a pour exigences : une capacité de 100Mi et un mode d'accès en lecture/écriture pour l'ensemble des noeuds du cluster

La storageClassName est de type manual

Le control plane de Kubernetes va chercher un PersistentVolume qui respecte les exigences du PersistentVolumeClaim. Si le control plane trouve un PersistentVolume approprié avec la même StorageClass, il attache la demande au volume

- 3. Déployer la ressource Secret mysql-secret au sein du namespace middle
- 4. Déployer la ressource Secret mysql-configmap au sein du namespace middle
- 5. Créer un fichier YAML nommé wordpress-deployment.yml . Dans ce fichier, définissez une ressource de type Deployment
- Utilisation de l'image Docker wordpress:latest
- 1 replicas
- Les variables définies dans la ressource MySQL ConfigMap en tant que variables d'environnement

WORDPRESS_DB_USER = user
WORDPRESS_DB_NAME = database

Définir WORDPRESS_DB_HOST en tant que variables d'environnement avec la valeur suivante : <nom-de-votre-service-mysql>.<namespace-ou-se-situe-la-bdd-mysql>.svc.cluster.local

• Le secret défini dans la ressource MySQL Secret en tant que variables d'environnement

WORDPRESS_DB_PASSWORD = password

- Selector et Labels qui ont pour clé-valeur app: blog
- Monter la ressource PVC wordpress-pvc afin de stocker les données du blog Wordpress de façon persistante
- 6. Créer un fichier YAML nommé wordpress-service.yml. Dans ce fichier, définissez une ressource de type Service qui écoutera sur le port 80 et redirigera les requêtes vers le port

80 des pods ayant le label app: blog

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

WORDPRESS_DB_USER
WORDPRESS_DB_NAME
WORDPRESS_DB_HOST
WORDPRESS_DB_PASSWORD

Livrables attendus : 4 fichiers YAML, la liste des commandes utilisées, 1 capture d'écran

NGINX

Les fichiers YAML concernant Nginx sont à déposer dans le dossier TP/front.

Les ressources Kubernetes concernant Nginx sont à déployer dans le namespace front .

- 1. Créer un fichier YAML nginx-cm.yml, au sein de ce fichier ajoutez-y une ressource de type configMap qui a pour clé-valeur le fichier de configuration default.conf correctement parametré pour rediriger les requetes vers le pod Wordpress
- 2. Créer une redirection de port vers le service Wordpress
- 3. La Accéder à la page d'accueil de votre site depuis votre navigateur

Si la configuration et le déploiement des ressources sont corrects, le serveur web Nginx redirige le flux vers Wordpress qui affiche sa page d'installation. Félicitations !

Si un message d'erreur concernant la base de donnée apparait, cela signifie que Wordpress n'arrive pas à se connecter à MySQL. Vérifiez votre Deployment Wordpress, votre StatefulSet MySQL ainsi que les Secrets et ConfigMap

Si la page d'accueil ou d'installation n'apparait pas, cela signifie que les requetes n'arrivent pas jusqu'au pod Wordpress. Vérifiez vos configuration Nginx et le bon fonctionnement du pod Wordpress.

Livrables attendus : 1 fichier YAML, la liste des commandes utilisées, 1 capture d'écran

Vérifications

- ? Les différentes ressources sont déployées dans le bon namespace ? Quelle commande utilisezvous pour vérifier cela ?
- ? Les ressources MySQL ConfigMap et MySQL sont à déployer au sein des namespaces middle et database, pourquoi ?
- ? Les différentes pods sont-ils correctement démarrés ? Quelle commande utilisez-vous pour vérifier cela ?
- ? 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 ?

Livrables attendus : Réponse aux questions