

Formation Kubernetes

-

Pratique



kubernetes

Atelier 11

-

Stockage persistant

Table des matières

1. Le stockage persistant	3
1.1. Création d'un volume de stockage partagé.....	3
1.1.1. Création d'un GCP Persistent Disk.....	3
1.1.2. Initialisation du volume Kubernetes.....	4
1.1.3. Validation du fonctionnement.....	5

1. Le stockage persistant

Les « Stockages Persistants » sont comme leur nom l'indique des espaces de stockage persistants, qui peuvent être utilisés par des pods **Kubernetes**. Leur cycle de vie est indépendant de celui du pod qui les utilise. Autrement dit, si le pod est supprimé les données ne le sont pas.

Avec des volumes dits « basiques », de type *hostPath*, *local* ou *emptyDir*, les données sont propres à un pod. De plus, si le pod est détruit, son volume l'est aussi et dans le cas d'un volume *emptyDir*, les données sont supprimées. En règle générale, ces volumes sont utilisés pour partager de la donnée temporaire entre plusieurs containers d'un même pod.

Sachant que ces volumes sont situés sur un nœud **Kubernetes** spécifique, le problème est que si **Kubernetes** déplace le pod sur un autre nœud, alors celui-ci perd l'accès à ses données.

Pour pallier ce problème, il faut utiliser des volumes de stockage partagés entre plusieurs nœuds **Kubernetes** et qui ont donc des cycles de vie indépendants de celui des pods.

1.1. Création d'un volume de stockage partagé

Dans cet atelier vous allez créer un volume de stockage partagé afin de stocker de manière persistante la base données *test* de l'application « simpleweb ». Pour rappel, cette base de données contient tous les utilisateurs créés par l'application « simpleweb ».

Le stockage utilisé sera un « GCP Persistent Disk ».

1.1.1. Création d'un GCP Persistent Disk

Pour des questions de performance de stockage, vous allez créer un « GCP Persistent Disk » dans la même zone de déploiement que votre cluster **Kubernetes**.

Pour rappel, une zone de disponibilité équivaut à un Datacenter.

- 1) Identifiez votre zone disponibilité grâce à la commande suivante :

```
gcloud container clusters list
```

- 2) Créez votre disque persistant dans votre zone de disponibilité grâce à la commande suivante :

```
gcloud compute disks create --size=1GiB --zone=[ZONE_DE_DISPO] simpleweb-bdd
```

- 3) Listez vos disques GCP et vérifiez la présence du disque « simpleweb-bdd » grâce à la commande suivante :

```
gcloud compute disks list
```

À ce stade, votre « GCP Persistent Disk » est bien créé sur votre infrastructure GCP.

1.1.2. Initialisation du volume Kubernetes

Maintenant que le disque persistant est créé sur GCP, vous allez devoir l'affilier à un volume **Kubernetes** au sein de votre déploiement « simpleweb-bdd ».

Le contenu (squelette des tables) de la base de données **test** est localisé dans le conteneur de base de données, à l'emplacement « `/var/lib/mysql/test/` ».

Monter directement votre volume sur le chemin « `/var/lib/mysql/test/` » serait une erreur, car initialement, le volume est vide et en le montant vous supprimeriez le contenu du dossier « `/var/lib/mysql/test/` ». Autrement dit, le volume « prend l'avantage » par rapport au filesystem des conteneurs.

Pour pallier ce problème, vous devez utiliser un **initContainer** qui va ajouter au volume les fichiers initiaux de la base **test** uniquement s'ils n'existent pas déjà.

Une fois le volume initialisé, il pourra être monté sur le dossier « `/var/lib/mysql/test/` » du conteneur de base de données.

- 1) Modifiez votre déclaration de déploiement « simpleweb-bdd » comme suit :

```
1  apiVersion: apps/v1beta1
2  kind: Deployment
3  metadata:
4    name: simpleweb-bdd
5  spec:
6    replicas: 1
7    template:
8      metadata:
9        labels:
10         app: simpleweb-bdd
11      spec:
12        volumes:
13         - name: simpleweb-bdd-data
14           gcePersistentDisk:
15             pdName: simpleweb-bdd
16             fsType: ext4
17        containers:
18         - name: simpleweb-bdd
19           image: ilkiinformation/simpleweb_bdd:v1
20           ports:
21             - containerPort: 3306
22           volumeMounts:
23             - name: simpleweb-bdd-data
24               mountPath: /var/lib/mysql/
25         initContainers:
26         - name: init-bdd
27           image: ilkiinformation/initdbvolume:v1
28           volumeMounts:
29             - name: simpleweb-bdd-data
30               mountPath: /tmp/volume/
```

- 2) Lancer le provisionnement de votre déploiement.
- 3) Vérifiez que le provisionnement s'est bien terminé.

1.1.3. Validation du fonctionnement

Dans cette partie, vous allez valider le bon fonctionnement du volume persistant créé précédemment.

- 1) Connectez-vous à votre application web et saisissez des données utilisateurs.
- 2) Détruisez le pod de base de données. Vérifiez que le déploiement a bien reprovisionné un nouveau pod.
- 3) Retournez sur l'interface web de votre application et listez les utilisateurs contenus dans la base de données. Que remarquez-vous ?

Comme vous avez pu le voir, l'utilisation de ces volumes persistants est un moyen efficace pour s'assurer la persistance des données des pods.

Il existe de nombreux types de stockages permettant de stocker les données de manière persistante, tels que :

- GCEPersistentDisk
- AWSElasticBlockStore
- AzureFile
- AzureDisk
- FC (Fibre Channel)
- Flexvolume
- Flocker
- NFS
- iSCSI

- RBD (Ceph Block Device)
- CephFS
- Cinder (OpenStack block storage)
- Glusterfs
- vSphereVolume
- Quobyte Volumes
- Portworx Volumes
- ScaleIO Volumes
- StorageOS
- ...