

**POLYTECHNIQUE
MONTREAL**



INF8480 - SYSTÈMES RÉPARTIS ET INFONUAGIQUE

TP3 - SYSTÈMES DE FICHIERS DISTRIBUÉS

Chargés de laboratoire :

Pierre-Frederick DENYS pierre-frederick.denys@polymtl.ca

Automne 2020 - V4.0

1 Introduction

1.1 Prérequis

- **Services de fichiers répartis et poste à poste** : composantes et interfaces, mécanismes pour l'implémentation. Exemples de Sun NFS, AFS, GFS, Ceph, Napster, Gnutella et BitTorrent

1.2 But du TP

- Découverte de docker et déploiement de containers
- Mise en place d'une architecture de stockage distribuée avec GlusterFS

2 Système de fichiers distribués GlusterFS

2.1 Introduction

Le but du TP est de mettre en place une architecture de stockage haute disponibilité avec glusterfs. L'architecture est composée de deux serveurs de stockage simulés par deux containers, et un poste client simulé par un troisième container. Ensuite, un volume répliqué et un volume distribué seront déployés sur les deux noeuds. Le déploiement va s'effectuer sur une machine virtuelle comme cela se fait dans l'industrie. La machine virtuelle est en local sur vos machines, mais vous ne pouvez y accéder que par SSH.

2.2 Procédure

2.2.1 Installation de GlusterFS

Suivre les instructions du TP2 pour déployer la VM sur virtualbox et vous y connecter en SSH.

2.2.2 Installation de GlusterFS

1. Créer les trois containers avec les commandes suivantes : (les ports ouverts sont documentés ici : <https://www.jamescoyle.net/how-to/457-glusterfs-firewall-rules>)

```
sudo docker run -t -d --privileged=true --name gluster1 --  
  hostname gluster1 -p 24007 -p 49152-49160 ubuntu  
sudo docker run -t -d --privileged=true --name gluster2 --  
  hostname gluster2 -p 24007 -p 49152-49160 ubuntu  
sudo docker run -t -d --privileged=true --name client --hostname  
  client -p 24007 -p 49152-49160 ubuntu
```

2. Connectez vous au premier container avec docker exec.
3. Récupérer l'IP du container avec la commande `ifconfig` (installer le paquet `net-tools`).

4. Dans une autre console, connectez vous au second container. Editer le fichier `/etc/hosts/` afin d'y ajouter le nom et l'IP du premier container. Faire de même sur le premier container. Au final, les fichiers `/etc/hosts/` devront contenir les noms et IP des deux containers.
5. Sur les **deux** containers, installer gluster-fs avec les commandes suivantes :

```
apt-get install software-properties-common
add-apt-repository ppa:gluster/glusterfs-6
apt-get update
apt-get install glusterfs-server
/usr/sbin/glusterd -p /var/run/glusterd.pid
mkdir /home/disk1/
```



Conseil

Le logiciel **terminator** vous permet de lancer des commandes identiques dans plusieurs terminaux avec les groupes.

6. Sur le container **gluster1**, ajouter le second container au *trusted pool*, et créer un volume répliqué.

```
gluster peer probe gluster2
gluster volume create replicated1 replica 2 gluster1:/home/disk1
/ gluster2:/home/disk1/ force
gluster volume start replicated1
```

7. Sur le container **gluster2**, vérifiez le bon fonctionnement du volume avec :

```
gluster volume status
```

8. Vous connecter au container **client** et exécuter les commandes suivantes pour installer glusterfs client, et monter le volume réseau. N'oubliez pas de modifier le fichier `/etc/hosts` afin d'y ajouter les noms et adresses des deux serveurs.

```
apt-get update
apt-get install software-properties-common
add-apt-repository ppa:gluster/glusterfs-6
apt-get update
apt-get install glusterfs-client
mkdir /mnt/replicated1
mount.glusterfs gluster1:/replicated1 /mnt/replicated1
```

9. Sur les **deux serveurs**, lancez la commande qui permet de voir le contenu du disque du volume en temps réel.

```
watch ls /home/disk1/
```

10. Sur le **client**, créer un fichier `bob.txt` dans le dossier `/mnt/replicated1/`. Vous devriez le voir apparaître sur les deux serveurs. Créer un fichier de grande taille dans `/root/` avec la commande suivante :

```
head -c 200M </dev/urandom >bigfile1
```

11. Comparer le temps de copie en local et sur le réseau et noter le rapport entre les deux temps.

```
time cp bigfile1 /mnt/replicated1  
time cp bigfile1 copie
```

12. Maintenant que vous avez un volume de stockage répliqué, mettez en place un volume de stockage distribué `distributed1` selon la même démarche (ne pas oublier de créer le dossier `/home/distributed1` sur les deux serveurs avant de créer le volume. Monter le sur le client dans le dossier `/mnt/distributed1` puis exécuter la commande suivante sur le client dans `/mnt/distributed1` qui crée 20 fichiers. et observer la répartition des fichier avec la commande `watch` du dossier `/home/distributed1` sur les deux serveurs.

```
for (( i=1; i <= 20; i++ )); do touch $i; done
```

le serveur 1 devrait stocker 10 fichiers, et le second les 10 autres fichiers.

3 Partie 2 : Création d'un Dockerfile

3.1 Introduction

Pour "containeriser" une application, une bonne méthode est de la déployer manuellement (comme dans la partie 1) dans un container pour tester, puis d'écrire un *Dockerfile* pour créer une image générique. Ensuite, on utilise `docker-compose` ou Kubernetes ou Docker swarm pour orchestrer les containers.

Dans cette seconde partie, on va se contenter de créer une image d'un serveur de stockage glusterfs, de créer un container avec et de l'ajouter à notre grappe (*trusted pool*) de la partie 1.

3.2 Procédure

- Créer un dockerfile à partir de la documentation https://docs.docker.com/develop/develop-images/dockerfile_best-practices/. Le fichier devra contenir :
 - Image de base `ubuntu`
 - Installation de GlusterFS
 - Exposer les bon ports
 - Pour simplifier, l'édition du fichier `/etc/hosts/` sera effectuée à la main hors du Dockerfile.
- L'image finale `glusterfs_tp3` devra être construite, et lancée avec :

```
sudo docker run -t -d --privileged=true --name gluster3 --  
hostname gluster3 -p 24007 -p 49152-49160 glusterfs_tp3
```

- Se connecter à l'intérieur de ce nouveau container, et demarrer Glusterfs. N'oubliez pas de modifier le fichier `/etc/hosts/` des quatre containers.

```
/usr/sbin/glusterd -p /var/run/glusterd.pid
```

- Sur le container `gluster2`, ajouter le noeud et un réplica au premier volume.

```
gluster peer probe gluster3  
gluster volume add-brick replicated1 replica 3 gluster3:/home/  
disk1/ force
```

- Tester la copie de fichiers comme pour la partie 1.

3.3 Remise

Exécuter **SUR LA VM** le script de correction (que vous pouvez copier à l'aide de la commande `scp`) avec le code header obtenu dans la question 1 du quiz. Si le fonctionnement de votre serveur de fichier est correct, le hash obtenu permet de valider la seconde question du quiz.

```
./correct.sh.x code_header_moodle
```

4 Conclusion

On remarque que l'ajout de noeuds devient fastidieuse avec cette méthode, car il faut déclarer le nom du noeud sur tous les autres noeuds. Cela est dû au fait que l'on utilise une résolution de nom locale (le fichier `/etc/hosts/`). Dans un système réel, on utilise un DNS (domain name server), qui permet de faire la résolution de nom sur le réseau. Ainsi, pour l'ajout d'un noeud, il suffit d'ajouter une entrée sur un seul fichier. Ce type de service sera abordé dans le prochain TP.

Contrairement à certains systèmes de stockage distribué, GlusterFS n'utilise pas de serveur de métadonnées, il n'y a donc pas de serveur central. GlusterFS est une solution très fiable et opensource, utilisée massivement en IA et Big Data et par de grandes entreprises comme Amazon et Red Hat.



5 Dépannage

5.1 Installation des paquets dans les containers

Il se peut que la VM ne dispose pas de la bonne date. Comme c'est celle-ci qui sert de référence de temps aux containers, il se peut qu'un message "is not valid yet" s'affiche suite à l'échec de l'installation des paquets dans le container. Sur la VM, vérifier la date avec la commande `date`. Si celle-ci est en avance ou en retard, lancez la commande suivante :

```
sudo systemctl restart systemd-timesyncd.service
```