Objet : Mise en place d'un cluster et haute disponibilité avec deux serveurs LAMP Debian 11 avec Heartbeat.

Développement :

Table des matières

Définition

Haute disponibilité

La haute disponibilité est une approche stratégique visant à garantir la résilience des systèmes informatiques, assurant ainsi une disponibilité continue des services cruciaux pour les entreprises et les utilisateurs.

Cluster

Un cluster se réfère à un ensemble de plusieurs ordinateurs ou serveurs interconnectés et agissant comme une seule unité. Ces systèmes travaillent ensemble pour améliorer la performance et la disponibilité.

Heartbeat

Heartbeat est un service essentiel pour assurer la surveillance et la disponibilité continue des systèmes informatiques, en particulier dans des environnements critiques où la fiabilité est primordiale. Heartbeat contribue à maintenir la disponibilité des services en détectant

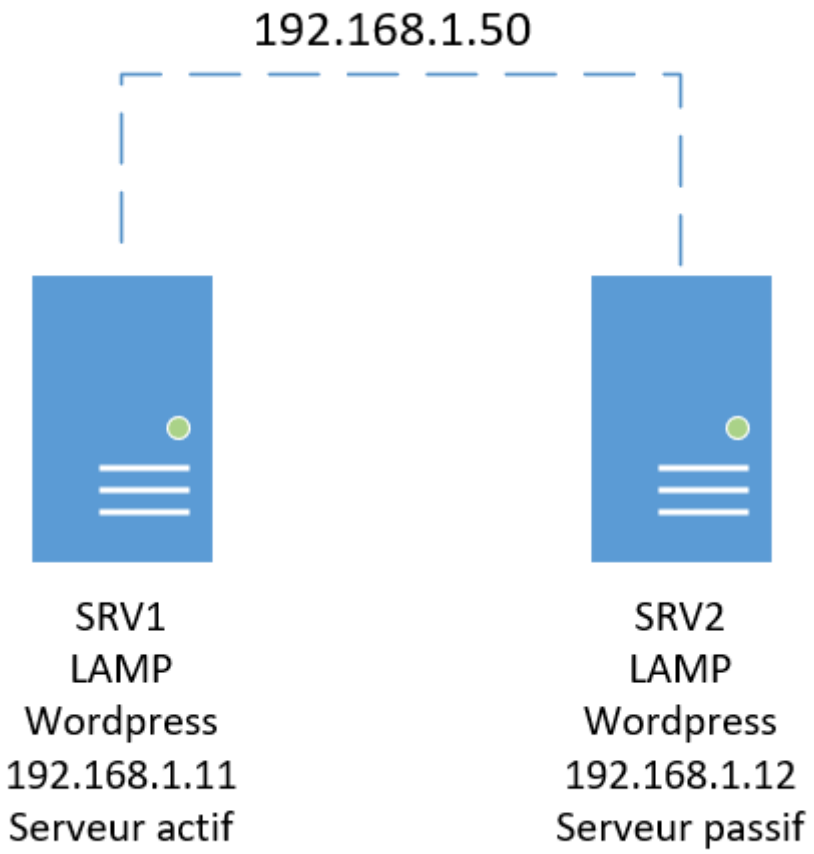
rapidement les pannes. En identifiant les défaillances, il permet aux clusters de basculer vers des nœuds fonctionnels pour éviter les interruptions de service.

Serveur LAMP

Un serveur LAMP est une solution pour l'hébergement web, offrant une combinaison robuste de composants open source pour le développement et l'exécution d'applications web. Les composants de cette configuration sont Linux, Apache, MySQL (ou MariaDB), et PHP.

Présentation

Nous aurons donc un serveur actif en **192.168.1.11** et un serveur passif **192.168.1.12** tout deux sont des serveurs LAMP sur lesquels sera installé le paquet **Heartbeat**. Nous souhaitons que les clients communiquent avec le serveur via l'IP virtuelle du cluster **192.168.1.50** et non pas vers 192.168.1.11 ou 192.168.1.12. Ce sera au cluster de passer les requêtes à tel ou tel serveur suivant la disponibilité du serveur "maitre".



Configuration & Installation

Mise à jour fichier Hosts

Tout d'abord, nous allons mettre à jours notre liste de paquet sur les **deux serveurs** :

```
sudo apt update
sudo apt upgrade
```

Ensuite, nous allons ajouter une ligne dans le fichier **/etc/hosts** permettant d'effectuer la liaison entre les machines avec leurs noms.
Sur le **SRV1** :

```
192.168.1.12 SRV2
```

Sur le **SRV2** :

```
192.168.1.11 SRV1
```

Maintenant, il faut tester la liaison avec les commandes :

```
ping srv2
ping srv1
```

Installation GlusterFS

Pour commencer, nous allons créer un répertoire pour stocker les données de GlusterFS sur les deux machines et ensuite télécharger ces paquets :

```
mkdir –p /gfsvolume/gv0
apt install –y glusterfs-server
```

Configuration GlusterFS

Tout d'abord, il faut activer le service GlusterFS sur les **deux serveurs** :

```
systemctl enable glusterd
systemctl start glusterd
systemctl status glusterd
```

Ensuite, nous allons ajouter les serveurs en tant que pair.
Sur **SRV1** :

```
Gluster peer probe srv2
```

```
Gluster peer status
```

Suite à ça, nous souhaitons configurer un volume "répliqué" qui stocke les fichiers créés sur chacune des machines, rendant ainsi le fichier disponible pour toute application ou conteneur s'exécutant sur ces VMs. Créez le volume répliqué nommé "gfs" avec 2 replica :
Sur **SRV1** :

```
gluster volume create gfs \
replica 2 \
srv1:/gfsvolume/gvo \
srv2:/gfsvolume/gvo force
```

Ensuite, nous allons démarrer le volume Gluster nommé "gfs" :
Toujours sur le **SRV1** :

```
gluster volume start gfs
gluster volume status gfs
```

Commande pour que ça s'active dès le démarrage des machines :
Sur les **deux serveurs** :

```
echo 'localhost:/gfs /mnt glusterfs defaults,_netdev,backupvolfile-server=localhost 0 0' >> /etc/fstab
```

Commande pour que ça s'active seulement pour un test, si la machine vient à s'éteindre le service ne sera plus en marche :
Sur les **deux serveurs** :

```
mount.glusterfs localhost:/gfs /mnt
```

Pour voir si cela fonctionne correctement, nous allons créer un dossier «test» avec le SRV1 et nous verrons sur le SRV2 si le dossier à bien était créé.
Sur **SRV1** :

```
touch /mnt/test
```

Sur **SRV2** :

```
ls /mnt/
```

Installation LAMP

Tout d'abord, nous allons installer les paquets de Apache, Mariadb et PHP :
Sur les **deux serveurs** :

```
apt install apache2
apt install mariadb-server mariadb-client
apt install php php-mysql
```

Ensuite, nous allons créer des répertoires répliqués pour WordPress et MySQL dans GlusterFS.
Sur **SRV1** :

```
mkdir /mnt/wordpress
mkdir /mnt/mysql
```

Redirection répertoire MySQL & Apache

Tout d'abord, il faut stopper le service Mariadb :
Sur les **deux serveurs** :

```
systemctl stop mariadb
```

Déplacer le répertoire /var/lib/mysql vers /mnt/mysql/ et donner les droits mysql au répertoire :
Sur **SRV1** :

```
mv /var/lib/mysql /mnt/mysql/
chown –R mysql :mysql /mnt/mysql/
```

Sur les **deux serveurs** :
Nous allons à présent modifier le chemin de la base de données dans le fichier de configuration de MariaDB dans le fichier **/etc/mysql/mariadb.conf.d/50-server.cnf**. Il faut donc changer le chemin de la base de données par :

```
datadir         = /mnt/mysql/mysql
```

Ensuite, il faut modifier le service mariadb.service afin d'éviter l'erreur : mysql can't creat test files.lower-test. Il faut commenter la ligne ProtectHome=true dans le fichier : **/etc/systemd/system/multi-user.target.wants/mariadb.service**

```
#ProtectHome=true
```

Après ça, recharger le service :

```
systemctl daemon-reload
```

Redémarrer le serveur MariaDB :
Sur **SRV1** :

```
systemctl start mariadb
systemctl status mariadb
```

Sur les **deux serveurs** :
Nous allons configurer les autorisations et changer le répertoire spécifié du fichier **/etc/apache2/site-available/000-default.conf** par :

```
<Directory /mnt/wordpress/>
        Options Indexes FollowSymLinks
        AllowOverride None
        Require all granted
        Allow from all
</Directory>
```

Après cela, on redémarre le service Apache 2 :

```
service apache2 restart
service apache2 status
```

Ensuite, il faut mettre en place une base de donnée avec un utilisateurs pour Wordpress.
Sur **SRV1** :

```
mysql -u root –p
CREATE DATABASE wordpress_db;
CREATE USER 'wp_user'@'localhost' IDENTIFIED BY 'password';
GRANT ALL ON wordpress_db.* TO 'wp_user'@'localhost' IDENTIFIED BY 'password';
FLUSH PRIVILEGES;
exit;
```

Nous allons télécharger la dernière version compressée de WordPress dans un répertoire temporaire :

```
cd /tmp && wget https://wordpress.org/latest.tar.gz
```

Maintenant il faut extraire le contenu de l'archive :

```
tar -xvf latest.tar.gz
```

Puis on copie récursivement tout le contenu du répertoire "wordpress" (y compris les sous-répertoires et fichiers) vers le répertoire "/mnt/wordpress/" :

```
cp -R wordpress /mnt/wordpress/
```

Nous allons changer le propriétaire et le groupe d'un répertoire pour que le serveur web a les autorisations nécessaires pour accéder et manipuler les fichiers de WordPress.

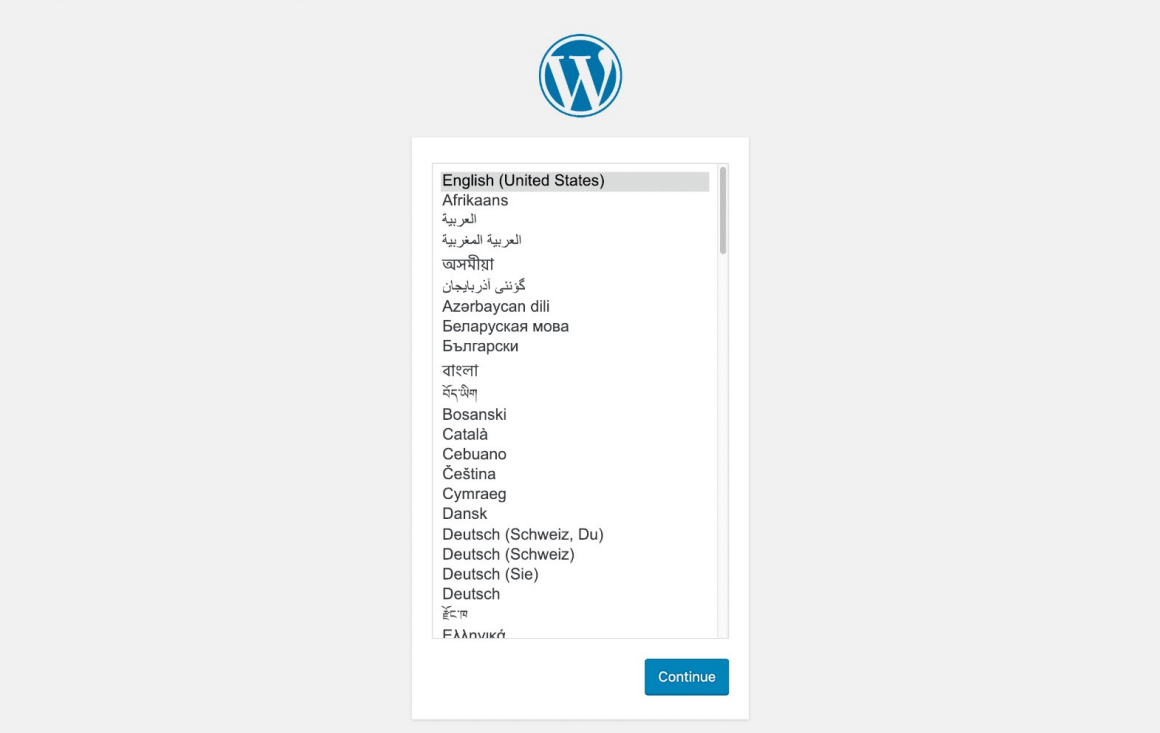
```
chown -R www-data:www-data /mnt/wordpress/
```

Après ça on attribue des permissions spécifiques à tous les fichiers et sous-répertoires du répertoire "/mnt/wordpress/"

```
chmod -R 755 /mnt/wordpress/
```

Pour voir si tout est fonctionnel ouvrer votre navigateur web puis aller sur : https://192.168.1.11/

Vous arriverez donc sur cette page :



Sur les **deux serveurs** :
Nous allons définir dans le fichier **/etc/apache2/site-available/000-default.conf** le répertoire racine du site web par :

```
Documentroot /mnt/wordpress/wordpress
```

Ensuite en redémarre le service Apache 2 :

```
Service apache2 reload
```

Configuration Heartbeat

Sur les **deux serveurs** :
Tout d'abord, il faut télécharger les paquets nécessaires de hearbeat :

```
apt install heartbeat
```

Pour faire fonctionner Heartbeat, il faut créer 3 fichiers de configuration. Le premier est **/etc/ha.d/ha.cf** et il devra contenir ces éléments :

logfile (facultatif) : Fichier où loguer les évènements relatifs à heartbeat (lancement, arrêt, etc.). Si vous utilisez ce paramètre, faites attention à limiter la taille des logs via le mécanisme logrotate de Debian.

keepalive : Intervalle entre 2 battements de coeur. La valeur est en secondes par défaut. Pour la spécifier en millisecondes, on rajoutera 'ms' derrière. (Par exemple, 200ms).

deadtime :Temps nécessaire avant de considérer qu'un noeud est mort. Le temps est en secondes par défaut. On rajoutera 'ms' derrière pour l'avoir en millisecondes.

bcast : Spécifie l'interface réseau utilisée par Heartbeat pour envoyer les battements de coeur (en UDP).

node : Liste des machines utilisées pour la haute disponibilité, séparées par des espaces.

auto_failback : Comportement à adopter si la machine en panne revient sur le réseau. Si le paramètre est à 'Off', elle se met simplement en attente. Avec 'On', elle redevient la machine active, et celle qui fonctionne à l'heure actuelle repasse en passive.

```
logfile        /var/log/ha-log
keepalive      2
deadtime       10
bcast          eth0
node           SRV1 SRV2
auto_failback  no
```

Le deuxième fichier est **/etc/ha.d/haresources**, il indique les opérations à effectuer au démarrage de la haute disponibilité sur une machine. Il devra contenir ces éléments :

```
SRV1 IPaddr::192.168.1.50/24/eth0 apache2 mysql
```

Le troisième fichier est **/etc/ha.d/authkeys**, il détermine la clé et le protocole de protection utilisé. Il devra contenir ces éléments :

```
auth 3
3 md5 mot_de_passe
```

Ensuite, on modifie les permissions de ce fichier pour qu'il ne soit plus visible par n'importe qui (si vous ne le faites pas, le programme ne se lancera pas) :

```
chmod 600 /etc/heartbeat/authkeys
```

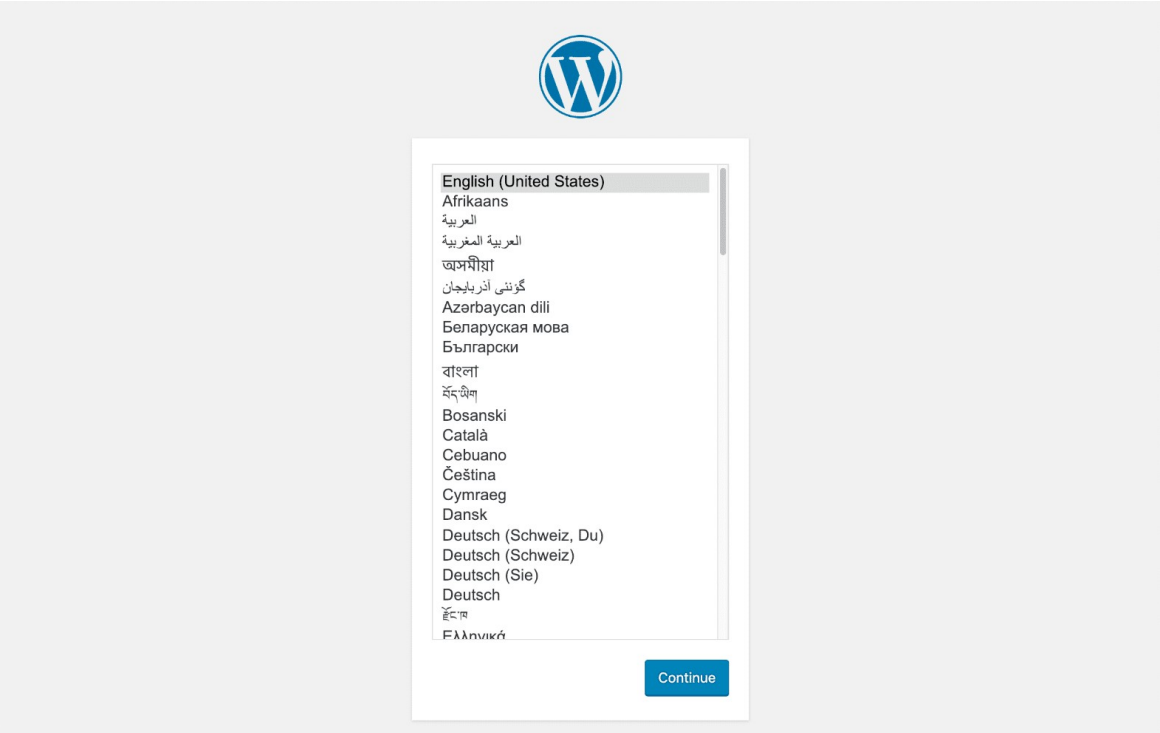
Assurez-vous que les fichiers de configuration des deux serveurs sont identiques puis vous pouvez démarrer le service Hearbeat :

```
/etc/init.d/heartbeat start
```

Une nouvelle interface virtuelle devrait apparaître, elle devrai se nommée eth0:0 avec l'adresse 192.168.1.50.

Simulation final

Pour finir, afin de vérifié si tout est fonctionnel ouvrer votre navigateur web et rendez-vous sur 192.168.1.50 (adresse IP de votre cluster). Celui-ci devrai vous renvoyez cette page (SRV1) :



Pour vérifier la présence de la haute disponibilité il suffit d'éteindre le SRV1 et voir si le SRV2 prend le relai sur l'adresse 192.168.1. Si cela fonctionne vous avez donc réussi à mettre en place un cluster haute disponibilité avec deux serveurs LAMP Debian 11 et Heartbeat.

Lucas MAIGNE