

# Failover avec Heartbeat, rsync et réplication mysql

Pour mieux expliquer ce TP, j'ai pris un exemple concret: une installation de Wordpress, qui nous permettra de tester la réplication des fichiers et de la BDD !

## Prérequis :

- Installer Debian 10 et ses mises à jour
- Cloner la VM pour en créer une autre, identique
- Renommer le clone /etc/hostname
- Configurer les IP statiques sur les 2 VM
- Choisir une IP virtuelle pour le cluster

## Installation d'Heartbeat

### (a faire sur chaque node)

```
apt install heartbeat
```

Création du fichier de configuration:

```
nano /etc/heartbeat/ha.cf
```

Ajoutez les paramètres suivants:

```
# Indication du fichier de log
logfile /var/log/heartbeat.log

# Les logs heartbeat ne seront pas gérés par syslog
logfacility none

# On liste tous les membres de notre cluster heartbeat (par les noms de préférences)
node NOM_DU_NODE1
node NOM_DU_NODE2

# On définit la périodicité de contrôle des nœuds entre eux (en seconde)
keepalive 1

# Au bout de combien de seconde un nœud sera considéré comme "mort"
deadtime 10
```

```
# Quelle carte réseau utiliser pour les broadcasts Heartbeat (ens33 dans mon cas)
bcast ens18

# Adresse du routeur pour vérifier la connexion au net
ping 10.10.1.254

# Rebascule-t-on automatiquement sur le primaire si celui-ci redevient vivant ? oui*
auto_failback yes
```

Création du fichier de clé:

```
nano /etc/heartbeat/authkeys
```

Ajoutez les paramètres suivants:

```
auth 1
1 sha1 UneCle
```

Édition des droits sur le fichier:

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

Création du fichier haressources:

```
nano /etc/heartbeat/haresources
```

On indique le nom du node primaire et son IP

```
NODE1 10.10.1.127
```

Démarrage d'Heartbeat:

```
systemctl start heartbeat
```

Si tout se passe bien, sur le premier node l'IP virtuel apparaît

```
2: ens18: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 56:95:75:f5:13:84 brd ff:ff:ff:ff:ff:ff
    inet 10.10.1.158/24 brd 10.10.1.255 scope global ens18
        valid_lft forever preferred_lft forever
    inet 10.10.1.127/24 brd 10.10.1.255 scope global secondary ens18:0
        valid_lft forever preferred_lft forever
```

## Installation de Wordpress

(Vous pouvez passer cette étape si vous avez votre propre site)

Idem, à faire sur les deux nodes

## Installation du serveur LAMP:

```
apt install apache2 mariadb-server php libapache2-mod-php php-cli php-common php-curl php-gd php-json php-mbstring php-mysql php-xml php-xmlrpc php-soap php-intl php-zip
```

## Création de la base de donnée:

```
mysql -u root -p
CREATE DATABASE wordpress DEFAULT CHARACTER SET utf8 COLLATE utf8_unicode_ci;
GRANT ALL ON wordpress.* TO 'wordpress_user'@'localhost' IDENTIFIED BY 'password';
FLUSH PRIVILEGES;
EXIT;
```

## Téléchargement et décompression de WordPress:

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

## Création du fichier .htaccess

```
touch /tmp/wordpress/.htaccess
```

## Copie du fichier config exemple en fichier config :

```
cp /tmp/wordpress/wp-config-sample.php /tmp/wordpress/wp-config.php
```

## Copie du dossier temporaire vers le dossier définitif :

```
cp -a /tmp/wordpress/. /var/www/html/wordpress
```

## Donner les permissions au dossier /wordpress :

```
chown -R www-data:www-data /var/www/html/wordpress
```

## Configurer le fichier de config Wordpress :

```
nano /var/www/html/wordpress/wp-config.php
```

## Remplacer :

- database\_name\_here
- user\_name\_here

- password\_here
- régler la collation de la base de donnée comme configurée plus tôt

Redémarrer Apache, se connecter via navigateur, vérifier que le site wordpress est accessible. (Sur les deux Nodes)

**Ne finissez pas la configuration de WordPress !**

## Réplication du dossier de l'application avec Rsync

Synchronisation des dossiers wordpress avec Rsync

Installation de Rsync (par défaut il y est déjà, au cas ou)

```
apt install rsync
```

Test de Rsync à partir du Node secondaire :

```
rsync --delete -avzhe ssh root@10.10.1.167:/var/www/html/wordpress/ /var/www/html/wordpress/
```

Mise en place du login SSH sans mot de passe, par certificat, sur le node secondaire :

```
ssh-keygen -t rsa -b 2048  
ssh-copy-id -i /root/.ssh/id_rsa.pub root@10.10.1.158
```

Test de la connexion sans mot de passe

```
ssh root@10.10.1.158
```

Vous avez accès directement au shell sans rentrer de mot de passe

Automatisation de la synchro entre les deux Nodes :

Créer un script */scripts/rsync.sh*

Y coller la commande rsync précédente

Si besoin on ajoute les droits d'exécution:

```
chmod +x rsync.sh
```

On ajoute la cron:

```
crontab -e
```

```
# m h dom mon dow command
* * * * * /etc/rsync.sh
```

On teste en créant un fichier dans le dossier wordpress sur le Node principal, il devrait être créé sur le Node secondaire moins d'une minute après.

On le supprime et on vérifie qu'il soit bien supprimé.

## Réplication des bases de données :

Sur le Node principal (base Master) :

On active les binary logs

```
mysql -u root -p
SET sql_log_bin = 1;
exit
```

A la fin du fichier `/etc/mysql/my.cnf` on ajoute les paramètres suivants :

```
[mariadb]
log-bin
server_id=1
log-basename=master1
```

On redémarre le service MySQL

```
systemctl restart mysql
```

On crée un utilisateur pour la réplication :

```
CREATE USER 'replication_user'@'%' IDENTIFIED BY 'repli_password';
GRANT REPLICATION SLAVE ON *.* TO 'replication_user'@'%';
```

Sur le Node secondaire :

Même principe, à la fin du fichier `/etc/mysql/my.cnf` on ajoute les paramètres suivants (pensez à changer l'ID):

```
[mariadb]
log-bin
server_id=2
log-basename=master1
```

On redémarre le service MySQL

```
systemctl restart mysql
```

On configure la base en esclave :

```
mysql -u root -p
CHANGE MASTER TO
  MASTER_HOST='10.10.1.158',
  MASTER_USER='replication_user',
  MASTER_PASSWORD='repli_password',
  MASTER_PORT=3306,
  MASTER_CONNECT_RETRY=10;
```

On démarre le mode esclave :

```
START SLAVE;
```

On vérifie l'état de la synchronisation :

```
SHOW SLAVE STATUS;
```

Si la synchronisation est fonctionnelle, on peut tester en terminant la configuration sur le Node primaire. Celle ci devrait s'appliquer automatiquement sur le Node secondaire.

On crée un article Wordpress, il devrait être dupliqué sur le second serveur.

## Voila, votre site web est répliqué !

---

Revision #2

Created 23 March 2020 15:28:05 by Cécile

Updated 23 March 2020 16:16:23 by Cécile