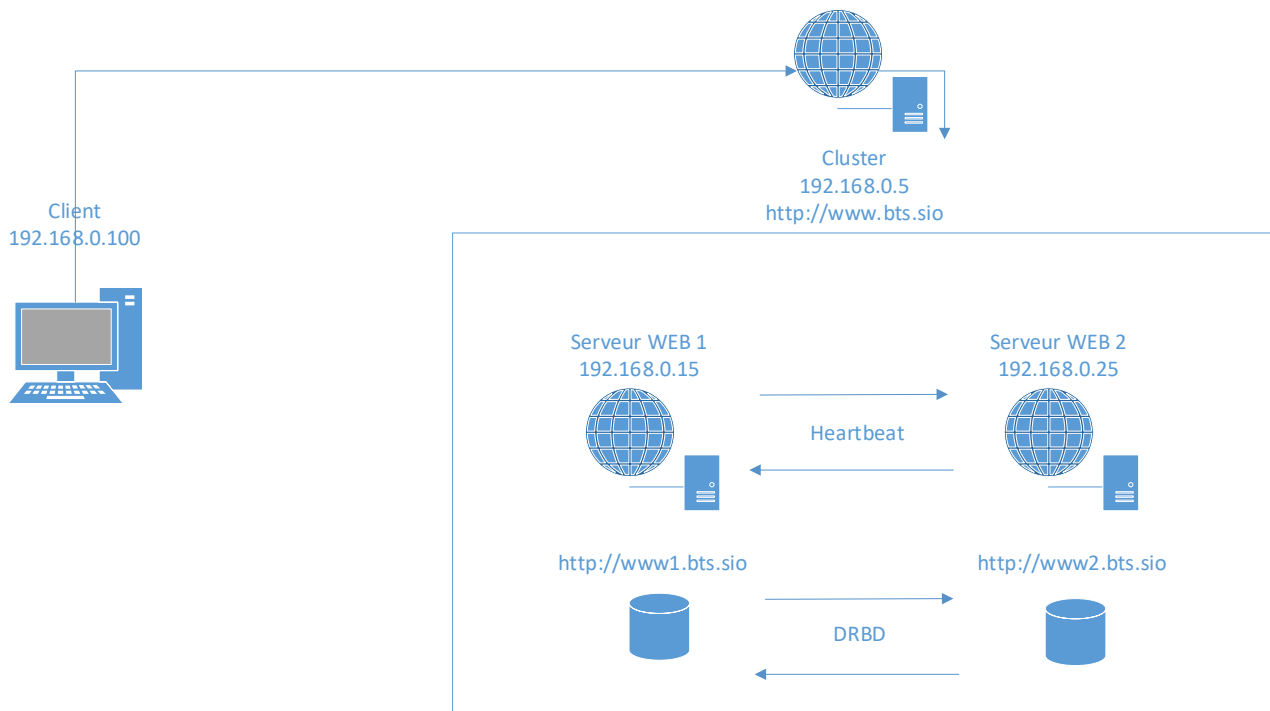


MISE EN PLACE D'UNE SOLUTION HAUTE DISPONIBILITE AVEC REPLICATION DES DONNEES



Contexte :

Votre client est pleinement satisfait de la solution mise en place pour pallier la panne d'un serveur web. Fonctionnelle pour les sites web « vitrine » qui n'accèdent à aucune base de données, votre client souhaite dorénavant que vous assuriez également la continuité de service de son serveur de base de données, en assurant la réplication des données.

Vous êtes chargé de mettre en place une plateforme de test, afin de vérifier le fonctionnement de la solution de haute disponibilité HeartBeat + DRBD.

Préparation de la plateforme de test

Nous allons avoir besoin de machines Debian avec apache et mariadb installés.

Pour ne pas perdre de temps, clonez une machine Debian fonctionnelle dont vous disposez d'un instantané « juste après l'installation » sans aucun paquet supplémentaire installé. Restaurez l'instantané « juste après l'installation », configurez votre machine pour qu'elle puisse communiquer avec l'extérieur.

Vous pouvez également utiliser vos machines du TP précédent, dans ce cas, il vous faudra désactiver le démarrage automatique de Heartbeat, et stopper le processus :

```
# systemctl disable heartbeat
# systemctl stop heartbeat
```

Pour fonctionner, DRBD a besoin d'une partition non formatée, nous allons donc ajouter un disque dur à notre serveur.

Vérifiez que le disque est bien vu par le système :

```
# fdisk -l
```

Créez une partition

```
# cfdisk /dev/sdb
```

 choisir New puis Logical (lui affecter toute la taille) puis Write

```
# fdisk -l
```

 renvoie maintenant sur sdb deux partitions sdb1 et sdb5 qui servira à la réplication des données

Installation des paquets :

```
# apt update
# apt install apache2 php7.4      ou php8.2 si debian 12
# apt install mariadb-server      choisir un mot de passe simple...
# apt install phpmyadmin          cocher la case « reconfigurer apache2 »
# apt install heartbeat
# apt install drbd-utils
```

Reconfigurez la machine dans votre domaine bts.sio avec IP 192.168.0.15

Modifiez son hostname (/etc/hostname) : www1

Pour garantir le bon fonctionnement de la solution, dans le cas d'un dysfonctionnement du serveur DNS, vous ajouterez au fichier /etc/hosts les lignes :

```
192.168.0.15    www1
192.168.0.25    www2
```

Effectuez un clone de cette machine (en réinitialisant l'adresse MAC) ayant pour IP 192.168.0.25 et pour nom www2

Nous utiliserons 192.168.0.5 comme IP virtuelle

Installation de la solution

SUR CHAQUE SERVEUR

Heartbeat a été installé lors de la préparation de la plateforme

DRBD fait partie du noyau Debian, il suffit de l'activer :

```
# modprobe drbd
# cat /proc/drbd
```

```
root@www1:~# cat /proc/drbd
version: 8.3.11 (api:88/proto:86-96)
srcversion: F937DCB2E5D83C6CCE4A6C9
```

Il faut que le service `drbd` démarre, sinon le script de démarrage de `drbd` lancé par `heartbeat` ne fonctionne pas.

Il faut donc lancer la commande :

```
# systemctl start drbd
# systemctl enable drbd
```

Finalisation pour faire en sorte que les outils `heartbeat` fonctionnent :

# chgrp haclient /sbin/drbdsetup	changement de groupe du fichier
# chmod o-x /sbin/drbdsetup	retrait du droit d'exécution pour « le reste du monde »
# chmod u+s /sbin/drbdsetup	accès aux droits du propriétaire lors de l'exécution du fichier
# chgrp haclient /sbin/drbdmeta	changement de groupe du fichier
# chmod o-x /sbin/drbdmeta	retrait du droit d'exécution pour « le reste du monde »
# chmod u+s /sbin/drbdmeta	accès aux droits du propriétaire lors de l'exécution du fichier

Configuration de DRBD (sur chaque serveur)

Editez le fichier de configuration `/etc/drbd.conf`

```
# You can find an example in /usr/share/doc/drbd.../drbd.conf.example

include "drbd.d/global_common.conf";
include "drbd.d/*.res";
resource r0 {
    device          /dev/drbd0;
    disk            /dev/sdb5;
    meta-disk       internal;

    disk {
        on-io-error detach;
    }
    startup {
        wfc-timeout 60;
        degr-wfc-timeout 30;
        become-primary-on www1;
    }

    net {
        after-sb-0pri discard-older-primary;
        after-sb-1pri call-pri-lost-after-sb;
        after-sb-2pri call-pri-lost-after-sb;
    }
    on www1 {
        address      192.168.0.15:7788;
    }
    on www2 {
        address      192.168.0.25:7788;
    }
}
```

A faire sur chaque serveur en même temps

En même temps signifie simplement que la commande doit être lancée sur les 2 serveurs et que la suivante ne peut être lancée sur quelque serveur que ce soit que si la précédente a terminé sans erreur sur les 2 serveurs

```
# drbdadm create-md r0
```

à faire sur le primaire puis le secondaire

```
--== Thank you for participating in the global usage survey ==--
The server's response is:

Writing meta data...
initializing activity log
NOT initialized bitmap
New drbd meta data block successfully created.
```

ENSUITE

```
# drbdadm up r0
```

à faire sur le primaire puis le secondaire

La commande suivante doit donner des informations semblables à cela sur les deux serveurs :

```
# cat /proc/drbd
```

```
root@www1:~# cat /proc/drbd
version: 8.3.11 (api:88/proto:86-96)
srcversion: F937DCB2E5D83C6CCE4A6C9
0: cs:Connected ro:Secondary/Secondary ds:Inconsistent/Inconsistent C r-----
   ns:0 nr:0 dw:0 dr:0 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:1048444
```

Puisque drbd est lancé mais aucun serveur *primaire* n'est déclaré. On le voit car après `ro:`, il est indiqué `Secondary/Secondary` ce qui signifie que le nœud sur lequel la commande est exécutée est *secondaire* (le premier avant le /) et que l'autre l'est aussi (le second après le /).

Pour ne connaître que l'état de la grappe drbd, la commande suivante suffit :

```
# drbdadm role r0
```

A faire seulement sur le serveur primaire (www1)

Il s'agit maintenant de déclarer un des deux serveurs comme primaire (le serveur secondaire se synchronisera automatiquement).

Lancez la commande :

```
drbdadm -- --overwrite-data-of-peer primary r0
```

```
cat /proc/drbd
```

```
root@www1:~# cat /proc/drbd
version: 8.3.11 (api:88/proto:86-96)
srcversion: F937DCB2E5D83C6CCE4A6C9
0: cs:SyncSource ro:Primary/Secondary ds:UpToDate/Inconsistent C r-----
   ns:1792 nr:0 dw:0 dr:2456 al:0 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:104665
2
   [>.....] sync'd: 0.4% (1046652/1048444)K
   finish: 0:08:43 speed: 1,792 (1,792) K/sec
```

On voit que le serveur secondaire se synchronise avec une barre de progression. Le temps indiqué après `finish:` est le temps indicatif restant avant la fin de la première synchronisation, en général c'est assez long, soyez patients ! Vous pouvez de temps en temps la relancer sur les deux serveurs pour voir où en est la synchronisation...

Après synchronisation des serveurs, vous devriez avoir des informations semblables :

```
root@www1:~# cat /proc/drbd
version: 8.3.11 (api:88/proto:86-96)
srcversion: F937DCB2E5D83C6CCE4A6C9
0: cs:Connected ro:Primary/Secondary ds:UpToDate/UpToDate C r-----
   ns:1048444 nr:0 dw:0 dr:1049108 al:0 bm:64 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:
:0
```

Si ce n'est pas le cas, il est probable qu'il faille resynchroniser les serveurs (après avoir vérifié les journaux d'activités, les journaux d'erreurs et corrigé l'erreur) :

```
# drbdadm disconnect r0
# drbdadm connect r0
```

Il est temps de formater la partition `drbd` **uniquement sur le serveur primaire** (www1)

```
# mkfs.ext3 /dev/drbd0
```

il vous faudra de nouveau un peu de patience...

Et de monter votre partition pour vérifier que tout va bien. Le montage de la partition doit utiliser `/dev/drbd0` et ne doit se faire **que sur le serveur primaire**. La création de `/data` doit se faire également sur le secondaire sans montage.

```
# mkdir /data
# mount /dev/drbd0 /data
```

à faire également sur le secondaire
uniquement sur le primaire

Vous pouvez lister le contenu de `/data` avec un

```
# ls /data
```

uniquement sur le primaire

Copiez ou créez des fichiers dans `/data` pour faire des tests de réplication par la suite

HEARTBEAT

Configuration IDENTIQUE SUR CHAQUE SERVEUR

Créez (ou modifiez) le fichier `/etc/ha.d/ha.cf` :

```
logfile          /var/log/ha-log
logfacility       local0
keepalive        2
deadtime         10
bcast            eth0
node             www1 www2
auto_failback    on
respawn          hacluster /usr/lib/heartbeat/ipfail
apiauth          ipfail gid=haclient uid=hacluster
```

Attention, adaptez le nom de votre interface réseau en fonction de votre système (`eth0`, `enp0s3`, `ens192`...)

Créez (ou modifiez) le fichier `/etc/ha.d/authkeys` :

```
auth 3
3 md5 password
```

Changez (ou non si déjà effectué) les droits de `authkeys` :

```
# chmod 600 /etc/ha.d/authkeys
```

Créez (ou modifiez) le fichier `/etc/ha.d/haresources` : (attention l'intégralité du texte s'écrit sur une seule ligne)

```
www1 IPaddr::192.168.0.5/24/eth0 apache2 drbdisk::r0 Filesystem::/dev/drbd0::/data::ext3 mysql
```

Attention, adaptez le nom de votre interface réseau en fonction de votre système (`eth0`, `enp0s3`, `ens192`...)

Tests

N'oubliez pas sur chaque serveur de :

Stopper les services mariadb et apache2

```
# systemctl stop mariadb
# systemctl stop apache2
```

Désactiver le démarrage automatique des services mariadb et apache2

```
# systemctl disable mariadb
# systemctl disable apache2
```

Activer le démarrage automatique du service heartbeat

```
# systemctl enable heartbeat
```

Redémarrez ensuite le service heartbeat

```
# systemctl restart heartbeat
```

- Pour réaliser les tests vous pouvez arrêter le serveur principal et vérifier que le dossier `/data` du secondaire est bien devenu accessible, le lister, puis copier un fichier dans `/data`. Remettre en service le serveur principal et vérifier que le nouveau fichier s'y trouve.
- Pour effectuer un test en situation réelle d'utilisation d'une base de données, vous pouvez (**après avoir stoppé le service mariadb sur www1**) utiliser le dossier `/data` pour contenir les dossiers et fichiers utiles à mariadb :
 - `/var/lib/mysql/ :`

```
# mv /var/lib/mysql/ /data/
# ln -s /data/mysql/ /var/lib/
# rm -rf /var/lib/mysql/
# ln -s /data/mysql/ /var/lib/
```

sur www1
sur www1
sur www2
sur www2

- `/etc/mysql/debian.cnf`

```
# mv /etc/mysql/debian.cnf /data/mysql/
# ln -s /data/mysql/debian.cnf /etc/mysql/
# rm -f /etc/mysql/debian.cnf
# ln -s /data/mysql/debian.cnf /etc/mysql/
```

sur www1
sur www1
sur www2
sur www2

Expliquez le rôle de chacune de ces 8 commandes.

Vous pourrez ensuite redémarrer le service mariadb sur www1, créer une base de données puis provoquer la bascule vers www2 et vérifier la présence de la base de données sur www2.