

# Gestion de la haute disponibilité

## Travaux pratiques

**Contexte** : La société THOLDI offre un service web de gestion des conteneurs disponibles dans les ports d'activités.

Plusieurs dockers se sont plaints de ne pas pouvoir accéder au site (voir les tickets d'incident reçus).

**Problématique** : Le contrat de service prévoit une disponibilité du site web à 4 neuf.

Le service devra être disponible par basculement sur un serveur de secours en cas de panne du serveur principal.

### 1. GÉNÉRALITÉS SUR LE CONCEPT DE HAUTE DISPONIBILITÉ.

**« On appelle « haute disponibilité » toutes les dispositions visant à garantir la disponibilité d'un service et son bon fonctionnement 24h/24, 7 jours/7. ».**

Le 19 août 2013, une interruption de service de 40 mn aurait fait perdre à Amazon près de 5 millions \$.

Le terme « Haute disponibilité » couvre un grand nombre de domaines :

- Manipulations des serveurs « à chaud » : reconfiguration des services, sauvegarde des données, ...
- Redondance du matériel
- Répartition dynamique des données (RAID, etc.)
- Stockage des données à un emplacement physique différent
- Plan de secours (PCA, PRA)
- Fonctionnement en mode dégradé (fournir un service jugé indispensable sans ses ressources habituelles humaines ou matérielles)...

### LA HAUTE DISPONIBILITÉ DU SERVICE

Lorsque les performances d'une seule machine suffisent à répondre à tous les clients, il faut tolérer la panne de celle-ci (mode « failover ») :

1. Une machine configurée à l'identique : services arrêtés, surveillant la première en permanence...



2. Lorsqu'elle détecte une panne : elle lance ses propres services, et répond aux clients à la place de l'autre...



3. Si la première est réparée : soit elle se met à son tour en écoute d'une panne, soit elle demande à reprendre la main...

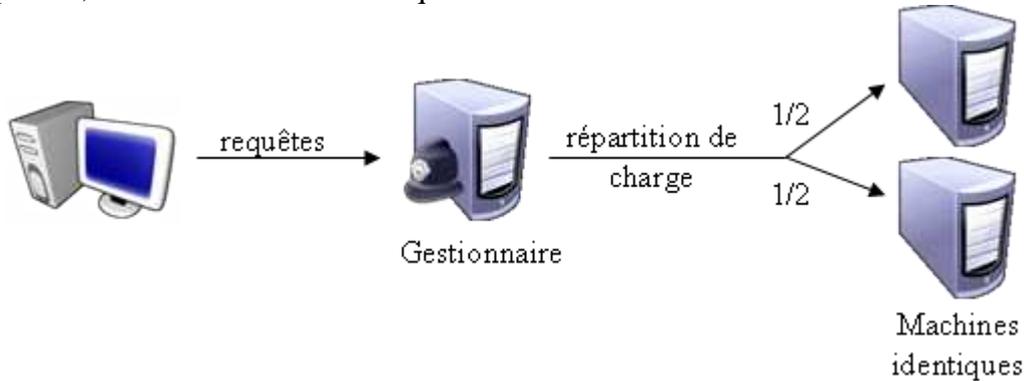


Une méthode de surveillance : la *tachycardie*<sup>1</sup> (HEARTBEAT)

<sup>1</sup> La tachycardie correspond à un rythme cardiaque plus rapide que la normale. Le contraire d'une tachycardie est une bradycardie, c'est-à-dire un rythme cardiaque lent.

Lorsque les performances d'une seule machine ne suffisent pas à répondre à tous les clients, il faut à la fois tolérer la panne et répartir la charge de travail :

1. Plusieurs serveurs proposant le même service ;
  2. Les requêtes des clients sont redirigées vers tous les serveurs...
- 1<sup>ère</sup> approche, alternance de manière équitable : « **Round Robin** »



- 2<sup>ème</sup> approche, alternance contrôlée : « **Load Balancer** »  
Cette méthode prend en compte la puissance des machines, le nombre d'utilisateurs déjà connectés, etc.



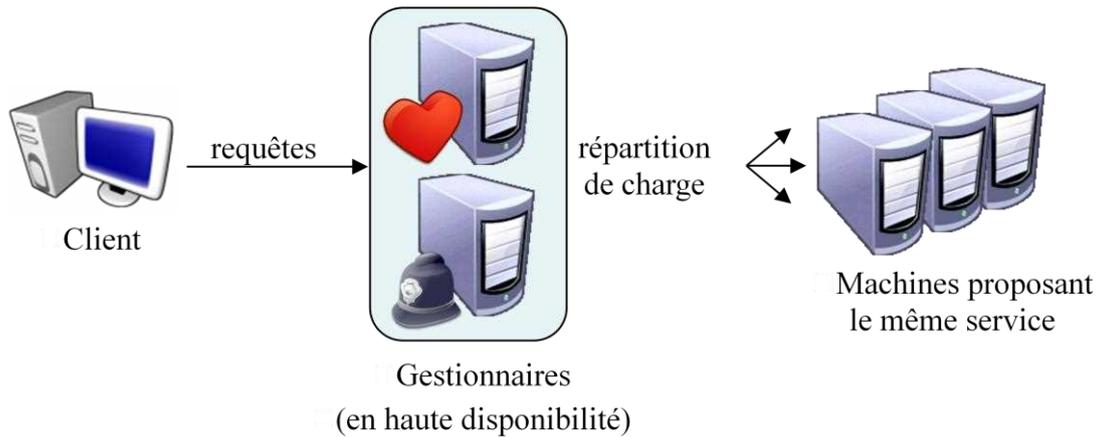
Les algorithmes de « *load balancing* » sont nombreux :

- Least-Connection
- Weighted Least Connection
- Round-Robin
- Weighted Round-Robin
- Locality-Based Least-Connection
- Destination-Hashing
- Source-Hashing
- Short Expected Delay
- Never Queue
- ...

Cependant cette méthode crée un nouveau point de faiblesse : le gestionnaire de répartition de charge de lui-même !



Pour surmonter cette faiblesse, on peut prévoir une solution mixte :



### LA HAUTE DISPONIBILITÉ DES DONNÉES

La plupart des services en réseau nécessite simultanément l'accès à des données :

- Un service web → les pages stockées dans le dossier de publication : /var/www/html
- Un service FTP → les fichiers stockés dans le dossier partagé
- Un service de messagerie → les boîtes aux lettres...

**Il faut donc distinguer :**



- **la haute disponibilité du service,**
- **la haute disponibilité des données.**

Habituellement, la disponibilité des données est assurée localement par un système de tolérance de panne sur les disques en RAID1 ou 5 ou 6.

Mais lorsque c'est la machine qui ne répond plus, le système RAID sur disques ne sert à rien !

L'usage d'un stockage en réseau (SAN) permet d'externaliser le stockage d'une machine.

Une autre idée est de « répliquer » en permanence les données entre 2 ou plusieurs machines à travers un réseau d'interconnexion. Une sorte de RAID en réseau ou RAID sur IP...

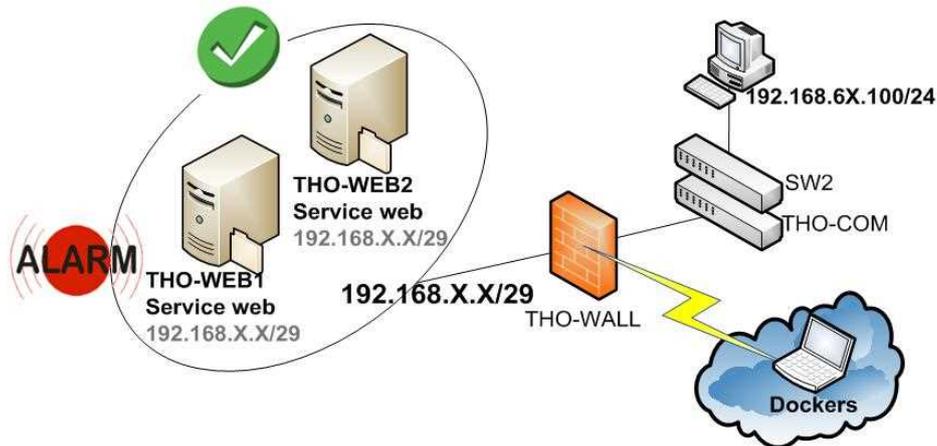
- **HEARTBEAT**, assure la surveillance d'un cluster de machines et la **basculement de services**.
- **DRBD**, permet de **répliquer/synchroniser les données** de deux disques à travers le réseau.

## 2. LA HAUTE DISPONIBILITÉ D'UN SERVICE AVEC « HEARTBEAT ».

L'idée générale pour assurer la disponibilité d'un service est de faire fonctionner **plusieurs machines** (deux au minimum) en même temps au sein de ce qu'on appelle un **cluster**. Chaque machine est un nœud (*node*) du cluster.



Comme son nom l'indique, HeartBeat utilise l'écoute permanente des « battements de cœur » entre les différents nœuds du cluster. Chaque nœud va vérifier si l'autre répond en prenant son « pouls ». Si l'un de ces battements ne reçoit pas de réponse, le processus de haute disponibilité sera déclenché et le service est basculé sur l'autre nœud.



Une fois le cluster configuré, on accède au service par une **seule et unique adresse IP** qui est celle du cluster.

*Sauf information précise,  
les mêmes manipulations seront à réaliser sur chaque nœud du cluster.*

### Travail à faire

#### Pré-requis : THO-WEB doit être upgradée vers une distribution Debian 10 « Buster »

1. Vérifier votre version de Debian : `lsb_release -a`
2. Démarrer la machine virtuelle « THO-WEB » et renommer en « THO-WEB1 ».
3. Vérifier votre connexion effective avec les ressources sur Internet...
4. Vérifier le fichier « sources.list » et mettre à jour le serveur.
5. Vérifier la présence d'un analyseur de trame : `aptitude search tcpdump`
6. Installer un gestionnaire de daemons : `aptitude install sysv-rc-conf`



La commande « `sysv-rc-conf` » vous permet de choisir les services ou daemons qui démarre à l'initialisation du système (en fonction du « runlevel »).

7. Installer les paquets pour synchroniser les machines : `aptitude install ntp ntpdate`

Par défaut les serveurs de temps sont configurés ainsi : `ntpq -p`

remote	refid	st	t	when	poll	reach	delay	offset	jitter
*wint.smersh.fr	193.204.114.233	2	u	1	64	377	63.527	-8.640	25.353
+ntp0.dillydally	192.93.2.20	2	u	5	64	377	49.592	-5.977	35.081
+obelix.gegeweb.	212.37.192.31	3	u	6	64	377	45.084	-7.332	28.052
+sysprod.libfy.c	192.93.2.20	2	u	7	64	177	53.038	-10.434	27.474

Si vous souhaitez changer de serveurs de temps, voir le fichier de configuration : `/etc/ntp.conf`

## 8. Renseigner le fichier hosts :

```
127.0.0.1      localhost
192.168.X.X   THO-WEB1
192.168.X.X   THO-WEB2
```

9. Installer HEARTBEAT : `apt-get install heartbeat`10. Vérifier le statut du service web sur les 2 nœuds du cluster : `/etc/init.d/apache2 status`11. Désactiver le démarrage et l'arrêt du service web par le système : `sysv-rc-conf`

*En effet, désormais c'est HeartBeat qui gèrera le daemon.*

12. Créer une simple page PHP « `tholdi.php` » (dans `/var/www` ou `/var/www/` : voir votre configuration « virtual host » dans apache2) :

```
<HTML>
<HEAD><TITLE>HeartBeat</TITLE></HEAD>
<BODY>
<?php
    echo date('d M Y');
    echo '<h1>Actuellement sur ' . strtoupper(gethostname()) . '</h1>';
    echo 'IP: </strong>' . $_SERVER['SERVER_ADDR'] . '</strong><br>';
    echo 'dossier: </strong>' . $_SERVER['DOCUMENT_ROOT'] . '</strong>';
?>
</BODY>
</HTML>
```

Les répertoires et fichiers de Heartbeat que nous allons utiliser ou modifier, sont les suivants :<sup>2</sup>

<code>/etc/ha.d/</code>	<code>ressources.d/</code>	<i>contient les scripts des services manipulés par HeartBeat</i>
<code>___ ha.cf</code>		<i>configuration générale</i>
<code>___ haresources</code>		<i>configuration des actions demandées à HeartBeat</i>
<code>___ authkeys</code>		<i>fichier contenant le mot de passe et son format de chiffrement</i>

13. Créer le fichier `/etc/ha.d/ha.cf`

```
bcast          eth0
debugfile      /var/log/ha-debug
logfile        /var/log/ha-log
logfacility    none
keepalive     2
deadtime      10
warntime      6
initdead      60
udpport       694
node          tho-web1
node          tho-web2
auto_failback on
```

Vérifier le nom de  
votre carte réseau :  
eth0 ? ens33 ?



Maintenant, nous allons définir le **nœud maître**, l'adresse **IP du cluster** et le **service** devant être assurés.

14. Mettre en place l'adresse IP du cluster : `nano /etc/ha.d/haresources`

```
tho-web1 IPAddr::192.168.0.X/29/eth0:0   apache2
tho-web1 MailTo::sip@tholdi.com
```

15. Mettre le chiffrement : `nano /etc/ha.d/authkeys`

```
auth 3
3 md5 azerty
```

16. Modifier les permissions sur le fichier : `chmod 600 /etc/ha.d/authkeys`

<sup>2</sup> Voir annexe 1 : HeartBeat.

17. Arrêter la machine virtuelle.

18. Cloner la machine vers la machine « THO-WEB2 ».

19. Démarrer la machine « THO-WEB2 » :

- Modifier son nom d'hôte : `/etc/hostname` et `/etc/hosts`
- modifier son interface réseau, par exemple `192.168.X.X/29`



Attention : la présence d'une nouvelle machine dans la DMZ peut remettre en cause votre stratégie d'adressage IP.

Cela pourra également remettre en cause vos stratégies de règles de pare-feu sur PfSense !

**Il faut retenir que seule l'adresse IP du cluster restera connue des clients web.**

En revanche les 2 machines du cluster conservent leur propre adresse IP pour accéder aux ressources de THOLDI (DNS, GLPI, Nagios...).

20. Redémarrer les 2 machines.

21. Vérifier le statut des services sur chaque nœud : `/etc/init.d/heartbeat status`

22. Analyser les processus en cours : `ps axf | grep heartbeat`

```
root@tho-web2:~# ps axf | grep heartbeat
4098 pts/2    S+   0:00 |      \_ grep heartbeat
4048 ?        SLs  0:00 heartbeat: master control process
4052 ?        SL   0:00 \_ heartbeat: FIFO reader
4053 ?        SL   0:00 \_ heartbeat: write: bcast eth0
4054 ?        SL   0:00 \_ heartbeat: read: bcast eth0
```

23. Observer les interfaces réseau sur THO-WEB1 : `ifconfig`

```
eth0:0      Link encap:Ethernet  HWaddr 00:0c:29:2e:c9:35
            inet addr:192.168.X.X  Bcast:192.168.X.X  Mask:255.255.255.248
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
```

24. Réaliser une capture de trame sur THO-WEB1 : `tcpdump -c 1 -X -i eth0 port 694`

```
19:33:56.556352 IP tho-web2.tholdi.com.53932 > 192.168.0.7.694: UDP, length 298
0x0000:  4500 0146 0c0b 4000 4011 ac3f c0a8 0005  E..F..@.?.?....
0x0010:  c0a8 0007 d2ac 02b6 0132 1c92 2323 230a  ..1.....2..##
0x0020:  3131 3a28 3029 743d 7374 6174 7573 2c31  11:(0)t=status,1
0x0030:  323a 2830 2973 743d 6163 7469 7665 2c31  2:(0)st=active,1
0x0040:  303a 2830 2964 743d 3237 3130 2c31 333a  0:(0)dt=2710,13:
0x0050:  2830 2970 726f 746f 636f 6c3d 312c 3135  (0)protocol=1,15
0x0060:  3a28 3029 7372 633d 7468 6f2d 7765 6232  :(0)src=tho-web2
0x0070:  2c32 373a 2831 2973 7263 7575 6964 31fd  ,27:(1)srcuid=.
0x0080:  6c6a 8a9a 7845 498d 4ea6 184c 270a bf2c  1j...xEI.N..L'..,
0x0090:  393a 2830 2973 6571 3d35 652c 3134 3a28  9:(0)seq=5e,14:(
0x00a0:  3029 6867 3d35 3837 3637 3966 312c 3134  (0)hg=587679f1,14
0x00b0:  3a28 3029 7473 3d35 3837 3637 6139 342c  :(0)bc=58767a94,
0x00c0:  3331 3a28 3029 6c64 3d30 2e30 3020 302e  31:(0)ld=0.00.0.
0x00d0:  3031 2030 2e30 3520 312f 3130 3320 3336  01.0.05.1/103.36
0x00e0:  3432 2c38 3a28 3029 7474 6c3d 332c 3432  42,8:(0)ttl=3,42
0x00f0:  3a28 3029 6175 7468 3d33 2061 6239 6663  :(0)auth=3.ab9fc
0x0100:  3937 6335 3138 3764 3564 6233 3038 6664  97c5187d5db308fd
0x0110:  3363 3036 3239 3639 3661 612c 2525 250a  3c0629696aa,%%.
0x0120:  3334 3a33 2035 3634 3564 3732 3962 3830  34:3.5645d729b80
0x0130:  3061 3062 6138 3738 3164 3966 3032 6138  0a0ba8781d9f02a8
0x0140:  3862 3161 362c 8b1a6,
```

1 packet captured  
3 packets received by filter  
0 packets dropped by kernel

25. Depuis un poste client, accéder au site web par l'adresse virtuelle :

<http://192.168.X.X/tholdi.php>

Éventuellement avec une adresse FQDN, par l'alias <http://www.tholdi.com> si vous disposez de l'enregistrement auprès du service DNS.



26. Déclencher une panne sur le serveur primaire **THO-WEB1** : **arrêt de la machine** !

- Accéder à nouveau au site web : <http://192.168.X.X/tholdi.php>



- Observer les interfaces réseau sur THO-WEB2 : `ifconfig`
- Observer le statut du service sur THO-WEB1 et THO-WEB2 : `service apache2 status`
- Consulter les journaux « heartbeat » : `tail -30 /var/log/ha-log`

```
Feb 02 19:36:49 tho-web2 heartbeat: [1185]: info: Received shutdown notice from 'tho-web1'.
Feb 02 19:36:49 tho-web2 heartbeat: [1185]: info: Resources being acquired from tho-web1.
Feb 02 19:36:49 tho-web2 heartbeat: [3360]: info: acquire local HA resources (standby).
Feb 02 19:36:49 tho-web2 heartbeat: [3361]: info: No local resources [/usr/share/heartbeat/ResourceManager listkeys tho-web2] to acquire.
Feb 02 19:36:49 tho-web2 heartbeat: [3360]: info: local HA resource acquisition completed (standby).
Feb 02 19:36:49 tho-web2 heartbeat: [1185]: info: Standby resource acquisition done [foreign].
2020/02/02_19:36:49 harc(default)[3386]: info: Running /etc/ha.d/rc.d/status status
2020/02/02_19:36:49 mach_down(default)[3400]: info: Taking over resource group IPaddr::192.168.10.13/29/eth0:0
2020/02/02_19:36:49 ResourceManager(default)[3424]: info: Acquiring resource group: tho-web1 IPaddr::192.168.10.13/29/eth0:0 apache2
2020/02/02_19:36:49 /usr/lib/ocf/resource.d/heartbeat/IPaddr(IPaddr_192.168.10.13)[3450]: INFO: Resource is stopped
2020/02/02_19:36:49 ResourceManager(default)[3424]: info: Running /etc/ha.d/resource.d/IPaddr 192.168.10.13/29/eth0:0 start
2020/02/02_19:36:49 IPaddr(IPaddr_192.168.10.13)[3537]: INFO: Using calculated netmask for 192.168.10.13: 255.255.255.248
2020/02/02_19:36:49 IPaddr(IPaddr_192.168.10.13)[3537]: INFO: eval ifconfig eth0:0 192.168.10.13 netmask 255.255.255.248 broadcast 192.168.10.15
2020/02/02_19:36:49 /usr/lib/ocf/resource.d/heartbeat/IPaddr(IPaddr_192.168.10.13)[3513]: INFO: Success
2020/02/02_19:36:49 ResourceManager(default)[3424]: info: Running /etc/init.d/apache2 start
2020/02/02_19:36:49 mach_down(default)[3400]: info: /usr/share/heartbeat/mach_down: nice_failback: foreign resources acquired
2020/02/02_19:36:49 mach_down(default)[3400]: info: mach_down takeover complete for node tho-web1.
Feb 02 19:36:49 tho-web2 heartbeat: [1185]: info: mach_down takeover complete.
Feb 02 19:37:01 tho-web2 heartbeat: [1185]: WARN: node tho-web1: is dead
Feb 02 19:37:01 tho-web2 heartbeat: [1185]: info: Dead node tho-web1 gave up resources.
Feb 02 19:37:01 tho-web2 heartbeat: [1185]: info: Link tho-web1:eth0 dead.
```

27. Redémarrer la machine THO-WEB1.

- Accéder à nouveau au site web : <http://192.168.X.X/tholdi.php>
- Observer le statut du service web sur THO-WEB1 et THO-WEB2
- Consulter les journaux « heartbeat ».
- Consulter les emails de l'utilisateur déclaré dans `haresources`

### 3. LA RÉPLICATION DES DONNÉES EN RÉSEAU AVEC DRBD.

« **Distributed Replicated Block Device** » est une architecture de stockage distribuée permettant la réplication de disques (ou d'autres blocs : partitions, volumes logiques, etc...) entre 2 machines en réseau.

DRBD est un logiciel libre pour GNU/Linux, réalisé par la société Linbit.

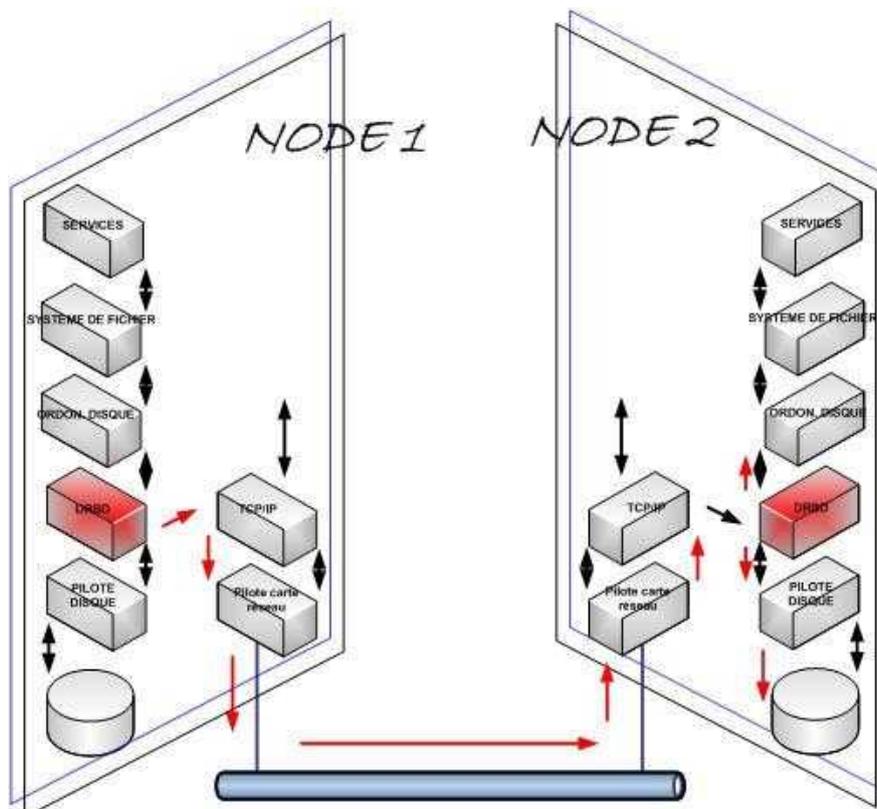


DRBD est composé d'un module noyau, d'outils d'administration en espace utilisateur ainsi que de scripts shell.

La **réplication** des données se fait:

- **En temps réel** : en permanence, pendant que les applications modifient les données présentes sur le périphérique
- De façon **transparente** : les applications qui stockent leur données sur le périphérique répliqué n'ont pas conscience que ces données sont en fait stockées sur plusieurs ordinateurs
- De façon **synchrone**, ou **asynchrone** : en fonctionnement synchrone, une application qui déclenche une écriture de donnée est notifiée de la fin de l'opération seulement après que l'écriture a été effectuée sur tous les serveurs, alors qu'en fonctionnement asynchrone, la notification se fait après que la donnée a été écrite localement, mais avant la propagation de la donnée.

DRBD ajoute une couche logique de périphériques (nommée /dev/drbdX, ou X est le numéro de périphérique mineur) au dessus de la couche logique locale des périphériques existante sur les machines participantes au cluster.



Les écritures sur la machine primaire sont transférées sur le périphérique de bas niveau et sont simultanément propagées à la machine secondaire.

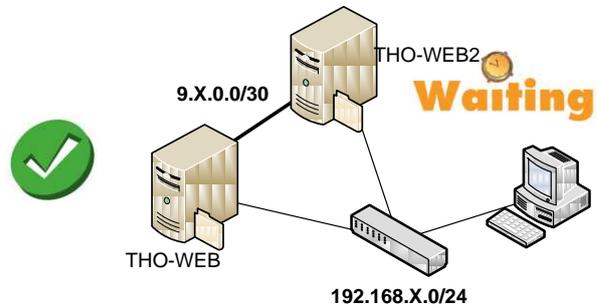
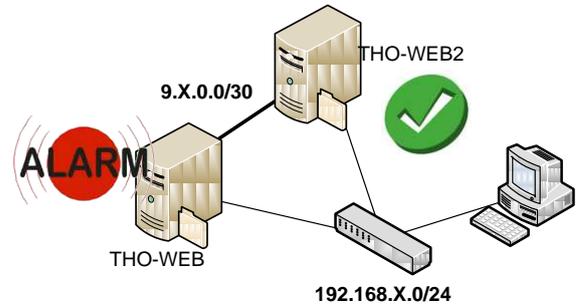
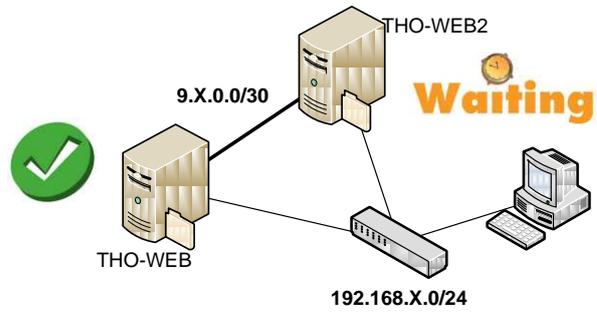
Le nœud secondaire transfère ensuite les données à son périphérique de bas niveau correspondant. Toutes les lectures sont effectuées localement.

En cas de défaillance du nœud primaire, un processus de gestion de cluster promeut le nœud secondaire dans un état primaire.

Cette transition peut requérir une vérification subséquente de l'intégrité du système de fichier, empilé au sommet de DRBD, par le biais d'une vérification du système de fichiers ou d'un rejeu de journal.

Quand l'ancien nœud primaire défaillant revient, le système peut (ou peut ne pas) l'élever à nouveau à un niveau primaire, après une synchronisation des données du périphérique.

L'algorithme de synchronisation de DRBD est efficace dans le sens où seuls les blocs qui ont changé durant la panne doivent être resynchronisés, plutôt que le périphérique entièrement.



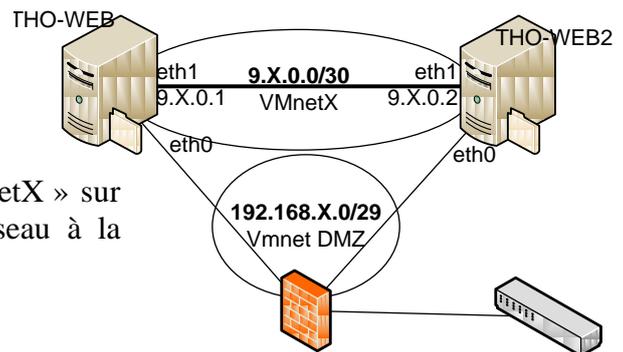
## Travail à faire



**Sauf information contraire, les étapes sont à réaliser sur les 2 machines !**

1. Arrêter les machines.
2. Ajouter un nouveau disque dur de 1GB sur chaque machine.
3. Installer une seconde carte réseau en « VMnetX » sur chaque machine pour dédier une liaison réseau à la réplication DRBD (9.X.0.0/30).
4. Redémarrer les machines.
5. Configurer les nouvelles interfaces « eth1 » sur le réseau « VMnetX » en 9.X.0.0/30 et ajouter les associations (THO-WEB1 et THO-WEB2) dans les fichiers /etc/hosts.
6. Créer une nouvelle partition dédiée à DRBD : `fdisk /dev/sd[x]` (par exemple sdb pour un 2<sup>ème</sup> disque) avec le système de fichiers « ext4 » (83) :

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1		1	130	1044193+	83	Linux



7. Installer le paquet pour drbd : `# aptitude install drbd-utils`
8. Charger le module drbd : `# modprobe drbd`  
`# lsmod | grep drbd`
9. Ajouter le module à charger à chaque démarrage dans le fichier `/etc/modules`  
`# echo drbd >> /etc/modules`  
`# sysv-rc-conf`

SysV	Runlevel	Config	-: stop service	=/+: start service	h: help	q: quit		
service	1	2	3	4	5	0	6	S
drbd	[ ]	[X]	[X]	[X]	[X]	[ ]	[ ]	

## 10. Configurer DRBD:

`/etc/drbd.conf :`

```
include "drbd.d/global_common.conf";
include "drbd.d/*.res";
```

`/etc/drbd.d/global_common.conf : trouver et modifier les lignes suivantes...`

```
startup {
    wfc-timeout 1;
    degr-wfc-timeout 1;
}
```

`/etc/drbd.d/data.res : créer le fichier pour déclarer la ressource « webstore »`

```
resource webstore { ❶
    device /dev/drbd0; ❷
    disk /dev/sdb1; ❸
    flexible-meta-disk internal; ❹
    on tho-web1 { ❺
        address 9.X.0.1:7788; ❻
    }
    on tho-web2 { ❺
        address 9.X.0.2:7788; ❻
    }
    syncer {
        rate 500M; ❼
    }
}
```

❶	Nom de la ressource. Il est recommandé d'utiliser des noms de ressources comme <code>data</code> , <code>r1</code> , etc.
❷	Le nom du périphérique pour DRBD et son numéro mineur. Dans l'exemple ci-dessus, le nom de nœud de périphérique, tel que créé avec <code>udev</code> , est référencé ( <code>/dev/drbd0</code> , où <code>data</code> représente le nom de la ressource). Pour cet usage, vous devez avoir le paquet <code>drbd-udev</code> installé. Alternativement, omettez le nom du nœud périphérique dans la configuration et l'utilisation de la ligne suivante à la place <code>:device minor 0</code>
❸	Le dispositif qui est répliqué entre les nœuds. Dans cet exemple, les dispositifs sont les mêmes sur les deux nœuds. Si vous avez besoin de différents périphériques, déplacer le paramètre de <code>disk</code> dans la section <code>on</code> .
❹	Le paramètre <code>meta-disk</code> contient généralement la valeur interne, mais il est possible de spécifier un périphérique explicite pour contenir les métadonnées.
❺	La section <code>on</code> contient le nom d'hôte d'un nœud.
❻	L'adresse IP et le numéro de port du nœud correspondant. Chaque ressource a besoin d'un port individuel, en commençant généralement par <code>7788</code> .
❼	Le taux de synchronisation. Réglez-le à 1/3 de votre bande passante. Il ne limite que la resynchronisation, pas le miroir.

## 11. Initialiser les métadonnées, création de la ressource « webstore » :

```
root@tho-web1:~# drbdadm -- --ignore-sanity-checks create-md webstore
initializing activity log
initializing bitmap (32 KB) to all zero
Writing meta data...
New drbd meta data block successfully created.
```

## 12. Redémarrer le démon drbd sur les deux serveurs : service drbd restart

## 13. Déclarer le serveur « maitre » : **uniquement sur THO-WEB1** (n°1)

```
root@THO-WEB1:~# drbdadm -- --overwrite-data-of-peer primary webstore
```

## 14. Vérifier en continu l'état de DRBD sur THO-WEB2 : commande « watch »<sup>3</sup>

La synchronisation (réplication) peut être longue, en fonction de la taille du volume...

```
Every 2.0s: cat /proc/drbd          tho-web2: Wed Jan 29 20:03:11 2020

version: 8.4.10 (api:1/proto:86-101)
srcversion: 15055BDD6F0D23278182874
0: cs:SyncTarget ro:Secondary/Primary ds:Inconsistent/UpToDate C r-----
ns:0 nr:57456 dw:57456 dr:0 al:8 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:990028
[>.....] sync'ed: 5.9% (990028/1047484)K
finish: 0:03:26 speed: 4,788 (4,788) want: 9,480 K/sec

(...)

version: 8.4.10 (api:1/proto:86-101)
srcversion: 15055BDD6F0D23278182874
0: cs:SyncTarget ro:Secondary/Primary ds:Inconsistent/UpToDate C r-----
ns:0 nr:186096 dw:186096 dr:0 al:8 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:861388
[==>.....] sync'ed: 18.0% (861388/1047484)K
finish: 0:01:29 speed: 9,580 (8,456) want: 17,160 K/sec

(...)

version: 8.4.10 (api:1/proto:86-101)
srcversion: 15055BDD6F0D23278182874
0: cs:SyncTarget ro:Secondary/Primary ds:Inconsistent/UpToDate C r-----
ns:0 nr:797784 dw:797784 dr:0 al:8 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:249700
[=====>.....] sync'ed: 76.6% (249700/1047484)K
finish: 0:00:08 speed: 27,464 (17,340) want: 35,640 K/sec

(...)

version: 8.4.10 (api:1/proto:86-101)
srcversion: 15055BDD6F0D23278182874
0: cs:Connected ro:Secondary/Primary ds:UpToDate/UpToDate C r-----
ns:0 nr:1047484 dw:1047484 dr:0 al:8 bm:0 lo:0 pe:0 ua:0 ap:0 ep:1 wo:f oos:0
```

## 15. Vérifier l'état de DRBD :

```
root@tho-web1:~# drbd-overview
0:webstore/0 Connected Primary/Secondary UpToDate/UpToDate

root@tho-web2:~# drbd-overview
0:webstore/0 Connected Secondary/Primary UpToDate/UpToDate
```

<sup>3</sup> La commande watch permet d'exécuter une commande en boucle tout en observant sa sortie standard

## 16. Vérifier l'état de DRBD à la fin de la synchronisation :

```

root@tho-web2:~# service drbd status
• drbd.service - LSB: Control DRBD resources.
  Loaded: loaded (/etc/init.d/drbd; generated)
  Active: active (exited) since Thu 2020-01-30 10:20:53 CET; 17s ago
  Docs: man:systemd-sysv-generator(8)
  Process: 534 ExecStart=/etc/init.d/drbd start (code=exited, status=0/SUCCESS)

Jan 30 10:20:52 tho-web2 drbd[534]: Starting DRBD resources: [
Jan 30 10:20:52 tho-web2 drbd[534]:   create res: webstore
Jan 30 10:20:52 tho-web2 drbd[534]:   prepare disk: webstore
Jan 30 10:20:52 tho-web2 drbd[534]:   adjust disk: webstore
Jan 30 10:20:52 tho-web2 drbd[534]:   adjust net: webstore
Jan 30 10:20:52 tho-web2 drbd[534]: ]

```

## 17. Créer le système de fichiers sur le périphérique DRBD de THO-WEB1 (primaire) :



```

root@tho-web1:~# mkfs.ext4 /dev/drbd0
mke2fs 1.42.12 (29-Aug-2014)
Creating filesystem with 261871 4k blocks and 65536 inodes
Filesystem UUID: 1055e6bd-104c-4a08-842d-d22adc317a45
Superblock backups stored on blocks:
    32768, 98304, 163840, 229376
Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
Writing superblocks and filesystem accounting information: done

root@tho-web1:~# fsck /dev/drbd0
fsck from util-linux 2.33.1
e2fsck 1.44.5 (15-Dec-2018)
/dev/drbd0: clean, 11/65536 files, 8859/261871 blocks

```

18. Créer, sur chaque serveur, le point de montage pour DRBD : `mkdir /var/webstore`19. Monter le système de fichier à utiliser : `mount /dev/drbd0 /var/webstore`

## 20. Vérifier le montage de l'arborescence sur le support disque :

```

root@tho-web1:~# mount
/dev/drbd0 on /var/webstore type ext4 (rw,relatime,data=ordered)

root@tho-web1:~# df -h | grep drbd
/dev/drbd0          991M  1.3M  923M   1% /var/webstore

```

## 21. Vérifier à nouveau le statut du service :

```

root@tho-web1:~# drbd-overview
NOTE: drbd-overview will be deprecated soon.
Please consider using drbdtop.

0:webstore/0 Connected Primary/Secondary UpToDate/UpToDate /var/webstore
ext4 991M 2.6M 922M 1%

```

## 22. Modifier le Virtual Host de apache pour indiquer le nouveau répertoire de stockage par défaut des pages web (DocumentRoot) vers /var/webstore

```
# nano /etc/apache2/sites-available/000-default.conf
```



*Selon votre configuration Apache2, vous devrez peut être modifier les permissions d'accès au répertoire des pages web :*

```
# nano /etc/apache2/apache2.conf
```

```
[...]

# Sets the default security model of the Apache2 HTTPD server. It does
# not allow access to the root filesystem outside of /usr/share and /var/www.
# The former is used by web applications packaged in Debian,
# the latter may be used for local directories served by the web server. If
# your system is serving content from a sub-directory in /srv you must allow
# access here, or in any related virtual host.
<Directory />
    Options FollowSymLinks
    AllowOverride None
# à modifier
#     Require all denied
#     Require all granted
</Directory>
```

## 23. Réaliser une modification du site web uniquement sur le serveur primaire THO-WEB1 :

```
echo '<br>dernière modifcation: </strong>'.date("d F Y H:i:s.",filemtime("index.php")).'</strong>';
```



*La modification est faite sur THO-WEB1,*

*Il faudra constater que la réplication s'est bien faite sur THO-WEB2*

## 24. Copier la page de test de /var/www/html/tholdi.php vers /var/webstore/.

## 25. Modifier le fichier /etc/ha.d/haresources de HeartBeat pour prendre en compte le montage du volume DRBD (sur chaque serveur[nœud]) :

```
tho-web1 IPaddr::192.168.X.X/29/eth0:0 drbddisk::webstore Filesystem::/dev/drbd0::/var/webstore:ext4 apache2
tho-web1 MailTo::sip@tholdi.com
```

#### 4. RÉALISATION DES TESTS DE HAUTE DISPONIBILITÉ DES SERVICES ET DES DONNÉES

1. Redémarrer les 2 nœuds du cluster.

2. Afficher les volumes montés : # df -h

• **Sur THO-WEB1 :**



Filesystem	Size	Used	Avail	Use%	Mounted on
udev	467M	0	467M	0%	/dev
tmpfs	97M	4.3M	93M	5%	/run
/dev/sda1	4.0G	1.1G	2.8G	28%	/
tmpfs	482M	53M	429M	11%	/dev/shm
tmpfs	5.0M	0	5.0M	0%	/run/lock
tmpfs	482M	0	482M	0%	/sys/fs/cgroup
/dev/sda7	352M	2.1M	327M	1%	/tmp
/dev/sda5	1.7G	371M	1.2G	24%	/var
/dev/sda8	13G	41M	12G	1%	/home
tmpfs	97M	0	97M	0%	/run/user/0
<b>/dev/drbd0</b>	<b>991M</b>	<b>9.7M</b>	<b>915M</b>	<b>2%</b>	<b>/var/webstore</b>



• **Sur THO-WEB2 :**



Filesystem	Size	Used	Avail	Use%	Mounted on
udev	467M	0	467M	0%	/dev
tmpfs	97M	4.3M	93M	5%	/run
/dev/sda1	4.0G	1.1G	2.8G	28%	/
tmpfs	482M	53M	429M	11%	/dev/shm
tmpfs	5.0M	0	5.0M	0%	/run/lock
tmpfs	482M	0	482M	0%	/sys/fs/cgroup
/dev/sda8	13G	41M	12G	1%	/home
/dev/sda7	352M	2.1M	327M	1%	/tmp
/dev/sda5	1.7G	371M	1.2G	24%	/var
tmpfs	97M	0	97M	0%	/run/user/0

3. Vérifier l'état de la réplication DRBD : # drbd-overview

• **Sur THO-WEB1 :**



```
0:webstore/0 Connected Primary/Secondary UpToDate/UpToDate /var/webstore ext4
991M 1.3M 923M 1%
```

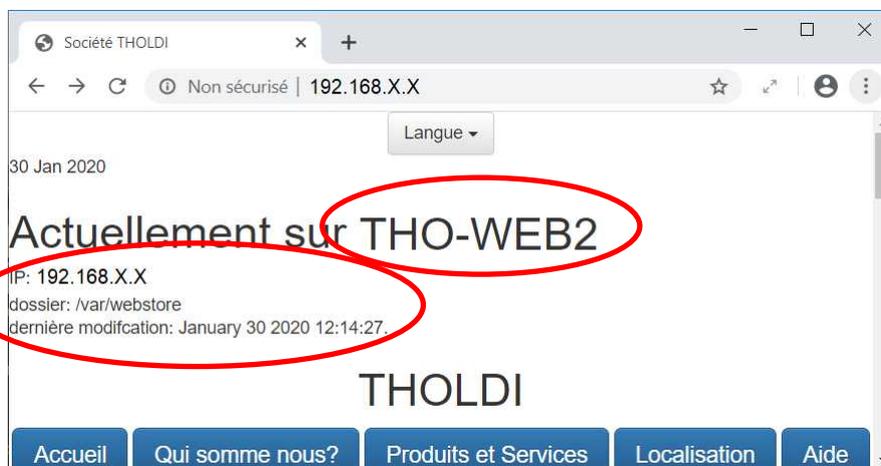


• **Sur THO-WEB2 :**



```
0:webstore/0 Connected Secondary/Primary UpToDate/UpToDate
```

4. Accéder au site web, à partir d'un poste client (adresse virtuelle) : <http://192.168.X.X/tholdi.php>

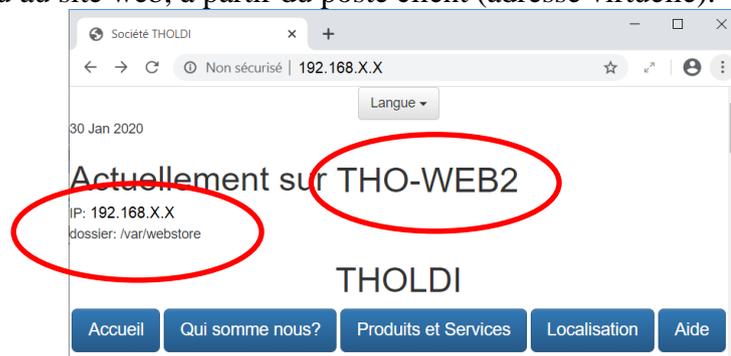


## 5. Observer les échanges entre les 2 nœuds :

```
root@tho-web2:~# tcpdump -c 1 -X -i eth0 port 7788
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 262144 bytes
11:59:12.354243 IP tho-web1.48945 > tho-web2.7788: Flags [P.], seq
423432836:423432852, ack 1245156631, win 229, options [nop,nop,TS val 1914501277 ecr
1737407188], length 16
    0x0000:  4500 0044 d12d 4000 4006 d424 c0a8 0a08  E..D.-@.@..$.
    0x0010:  c0a8 0a09 bf31 1e6c 193d 1284 4a37 9517  .....1.l.=..J7..
    0x0020:  8018 00e5 f83e 0000 0101 080a 721c f89d  .....>.....r...
    0x0030:  678e bad4 8620 ec20 0000 0013 0000 0000  g.....
    0x0040:  0000 0000                                     ....
1 packet captured
3 packets received by filter
0 packets dropped by kernel
```

## 6. Provoquer une panne sur THO-WEB1 (primaire) : # service heartbeat stop

## 7. Accéder à nouveau au site web, à partir du poste client (adresse virtuelle).



## 8. Afficher les volumes montés sur THO-WEB2 : # df -h | grep "drbd"

```

/ dev/drbd0          1004M  18M  936M   2% /var/webstore

```

## 9. Vérifier l'état de la réplication DRBD sur THO-WEB2: # drbd-overview

```
0:webstore/0 Connected Primary/Secondary UpToDate/UpToDate /var/webstore ext4 991M 1.3M 923M 1%
```

## 10. Consulter les log :

```
tail /var/log/messages
tail /var/log/ha-debug
tail /var/log/ha-log
```



```
edon@THO-WEB1:~$ mail
```

**Pour la réplication DRBD du site « officiel » de THOLDI, il suffira donc de copier le contenu de /var/www/html vers /var/webstore sur le serveur primaire THO-WEB1 (n°1).**

**Les paramètres d'accès FTP doivent être éventuellement reconfigurés sur THO-WEB2**

## ANNEXE 1 : HEARTBEAT.

### Les directives du fichier *ha.cf* :

Élément	Explications
bcast	indique l'interface réseau par laquelle on va effectuer la prise de pouls.
debugfile	indique le fichier de débogage à utiliser.
logfile	indique le log d'activité à utiliser (à consulter en cas d'erreurs).
logfacility	indique que l'on utilise la facilité syslog en plus.
keepalive	indique le délai entre deux battements de pouls en secondes.
deadtime	indique le temps nécessaire avant de considérer un nœud comme étant mort.
warntime	indique le délai avant d'envoyer un avertissement pour les pouls en retard.
initdead	indique un deadtime spécifique pour les configurations où le réseau met un certain temps à démarrer. initdead est normalement deux fois plus grand que deadtime (au minimum).
udpport	indique le port à utiliser pour la prise de pouls.
node	renseigne le nom des machines faisant partie du cluster. Ce nom doit être identique à celui retourné par la commande <b>hostname</b> .
auto_failback	indique le comportement à adopter si le <i>node</i> maître revient dans le cluster. Si on met la valeur on, lorsque le <i>node</i> maître revient dans le cluster, tout est transféré sur ce dernier. Si on met la valeur off, les services continuent à tourner sur l'esclave même lorsque le maître revient dans le cluster. La valeur off permet de faire un retour à la normale manuellement lorsque la charge de production est moins importante

### Les directives du fichier *haresources*

Le fichier *haresources* nous permet de déclarer, à HeartBeat, les diverses actions à faire si l'autre nœud tombe. Son format est le suivant : *Nom\_1<sup>ère</sup>\_machine action1 action2 ... action n*

Le nom de la 1<sup>ère</sup> machine est celui du nœud qui est considéré comme le principal.

Les actions sont codifiées, on peut voir les différentes possibilités dans le fichier exemple */usr/share/doc/heartbeat/haresources.gz* (fichier à décompresser !)

Voici trois actions principales :

<i>Machine</i>	<i>Action</i>	<i>Explications</i>
Nom_1 <sup>ère</sup> _machine	<b>service</b>	<b>pour ordonner le démarrage et la surveillance d'un service : par exemple apache2... !</b>
Nom_1 <sup>ère</sup> _machine	<b>IPaddr::w.x.y.z/mask/interface</b>	<b>pour gérer une ip virtuelle. Exemple : IPaddr ::172.31.0.200/26/eth0</b>
Nom_1 <sup>ère</sup> _machine	<b>Filesystem::/device ::point_de_montage ::format_fichier</b>	<b>permet de monter des répertoires réseaux partagés. Exemple : Filesystem ::/dev/sda6 ::mnt ::ext3</b>
Nom_1 <sup>ère</sup> _machine	<b>MailTo::root</b>	<b>permet d'envoyer des mail à un utilisateur. Exemple : MailTo::toto@free.fr</b>

### Le fichier *authkeys*

Il s'agit ici de créer un mot de passe entre les deux éléments du cluster, il peut être chiffré en en sha1 ou en md5 par exemple. Par contre le mot de passe est en clair dans le fichier d'où la nécessité de le protéger avec des droits très limités.

Format :

**auth [N]** → type d'authentification [N] avec 1 (crc), 2 (sha1), ou 3 (md5)  
**[N] md5 azerty** → on donne le mot de passe azerty avec le type choisi.

## ANNEXE 2 : COMPLÉMENT DRBD.

### Activation et désactivation des ressources

- *Activation des ressources : drbdadm up [nom\_ressource]*
- *Désactivation des ressources : drbdadm down [nom\_ressource]*

*Il est possible d'utiliser "all" à la place du nom de ressource pour tout activer ou tout désactiver.*

### Reconfigurer les ressources

Vous pouvez valider les paramètres qui ont été changés dans les fichiers de configuration des ressources sans redémarrer le service.

Il faut effectuer cette commande sur les deux nœuds :

- *drbdadm adjust [nom\_ressource]*

*"all" pour appliquer sur toutes les ressources.*

### Changer le rôle d'une ressource

Passer manuellement le rôle d'une ressource de secondaire en primaire (promotion) ou vice versa (rétrogradation), faite ceci en utilisant les commandes suivantes:

- *drbdadm primary [nom\_ressource]*
- *drbdadm secondary [nom\_ressource]*

### Liens qui peuvent vous aider:

- *DRBD - Réplication de disque par le réseau :*  
 <http://www.octetmalin.net/linux/tutoriels/drbd-distributed-replicated-block-device.php>
- *DRBD - Test de simulation de crash système du Primary*  
 <http://www.octetmalin.net/linux/tutoriels/drbd-test-simulation-crash-systeme-fonctionnement-et-reparation.php>
- *DRBD - Afficher et comprendre les différents états des disques*  
 <http://www.octetmalin.net/linux/tutoriels/drbd-afficher-et-comprendre-les-differents-etats-des-disques.php>
- *DRBD - Message d'erreur "Split-Brain"*  
 <http://www.octetmalin.net/linux/tutoriels/drbd-erreur-message-split-brain.php>
- *DRBD - Erreur - mount: you must specify the filesystem type*  
 <http://www.octetmalin.net/linux/tutoriels/drbd-erreur-message-mount-you-must-specify-the-filesystem-type.php>

### Encore de la veille technologique ?

-  [http://doc.ubuntu-fr.org/tutoriel/mirroring\\_sur\\_deux\\_serveurs#configuration1](http://doc.ubuntu-fr.org/tutoriel/mirroring_sur_deux_serveurs#configuration1)
-  [http://www.linux-ha.org/wiki/Main\\_Page](http://www.linux-ha.org/wiki/Main_Page)
-  <http://doc.ubuntu-fr.org/heartbeat>
-  [http://www.igm.univ-mlv.fr/~dr/XPOSE2006/JEREMIE\\_LEGRAND\\_HAUTE\\_DOSPO/index.htm](http://www.igm.univ-mlv.fr/~dr/XPOSE2006/JEREMIE_LEGRAND_HAUTE_DOSPO/index.htm)
-  <http://wapiti.enic.fr/Commun/ens/peda/options/ST/RIO/pub/exposes/exposesrio2007/legrand-playez/ha.htm>
-  <http://www.silverlake.fr/index.php?post/2011/05/25/Cluster-%C3%A0-deux-noeuds-%C3%A0-basculement>
-  [http://www.dalibo.org/hs45\\_drbd\\_la\\_replication\\_des\\_blocs\\_disques](http://www.dalibo.org/hs45_drbd_la_replication_des_blocs_disques)
-  [http://hpomeon.free.fr/spip.php?article24&calendrier\\_mois=6&calendrier\\_annee=2011](http://hpomeon.free.fr/spip.php?article24&calendrier_mois=6&calendrier_annee=2011)

#### Apache Cluster

-  [http://www.howtoforge.com/high\\_availability\\_loadbalanced\\_apache\\_cluster](http://www.howtoforge.com/high_availability_loadbalanced_apache_cluster)

#### MySQL cluster

-  <http://dev.mysql.com/doc/refman/5.0/fr/ndbcluster.html>
-  [http://www.howtoforge.com/loadbalanced\\_mysql\\_cluster\\_debian](http://www.howtoforge.com/loadbalanced_mysql_cluster_debian)

## ANNEXE 3 : COMPLÉMENT MON.

"mon" est un outil de suivi de la disponibilité des services, et l'envoi d'alertes sur des événements prévus.

Les services sont testé par un "moniteur", qui peut être quelque chose d'aussi simple que de pinger un système, ou aussi complexe que l'analyse des résultats d'une transaction de niveau application.

Les alertes sont des actions telles que l'envoi d'e-mails, faire des représentations à des systèmes de billetterie, ou le déclenchement de ressources de basculement dans un cluster haute disponibilité.

- **Installation**

aptitude install mon

- **Configuration de Mon**

- *Script de démarrage*

Configurer la surveillance de l'état de Heartbeat avec un script, permettant ainsi un passage des ressources en cas de non réponse d'un service.

Le dossier **mon.alert** contient un ensemble de scripts permettant à **Mon** de générer des alertes en cas de défaillance des services.

Créez le fichier `ha-up-down.alert` dans `/etc/ha.d/mon/mon.alert`.

```
#!/bin/bash
# arrêt/démarrage du service heartbeat
HEARTBEAT="/etc/rc.d/init.d/heartbeat"
if [ "$9" = "-u" ]
then
    $HEARTBEAT start
else
    $HEARTBEAT stop
Fi
# envoi d'un mail aux administrateurs (argument passé par mon.cf)
/etc/ha.d/mon/alert.d/mail.alert $*
```

- *Mon.cf*

Le fichier `/etc/ha.d/mon/mon.cf` contient la liste des services que "mon" doit surveiller et les actions résultantes de divers évènements. L'exemple suivant montre comment surveiller l'état d'un serveur http et basculer les ressources en cas de non réponse.

```
# emplacement des fichiers de configuration/d'alerte/logs
Cfbasedir = /etc/ha.d/mon/etc
Alertdir = /etc/ha.d/mon/alert.d
Mondir = /etc/ha.d/mon/mon.d
Statedir = /etc/ha.d/mon/state.d
logdir = /var/log/
# type d'authentification
authtype = getpwnam
# le serveur à surveiller
hostgroup server 127.0.0.1
# les services à surveiller sur le serveur
watch server
    service http
        interval 30s
        monitor http.monitor
        period wd {Mon-Sun}
# les actions en cas de panne/retour à la normale
    alert ha-up-down.alert -S "service web KO !" support@swiss-galaxy.com
    upalert mail.alert -S "service web OK !" support@swiss-galaxy.com
# délai entre chaque alerte
    alertevery 600s
```