

Installation Manuelle

DRBD et GFS2

DRBD est un système de réplication de disque au travers d'un réseau. Il permet de "recopier" le contenu d'un disque vers un serveur distant.

Ceci est particulièrement intéressant (et utilisé) dans les architectures hautement disponibles (Architectures "HA").

DRBD est un module noyau qui se comporte comme un driver de système de fichier. Toutes modification est "interceptée" et envoyée au miroir via le réseau.

Par défaut, DRBD fonctionne en mode Primaire / Secondaire. Ceci signifie que l'un des disques supporte les lectures / écritures, mais que le second n'accepte que les opérations en lecture. Dans les dernières versions de DRBD, il est possible d'obtenir des disques répliqués en mode primary / primary. Cela signifie que les 2 disques supportent les écritures. Encore faut-il un système de fichier qui le supporte. **Ce n'est pas le cas de systèmes de fichiers "classiques" tels que EXT2/3/4, ReiserFS, ...**

Néanmoins, ce système est intéressant dans le cas du fail-over IP: vous n'aurez pas besoin d'activer les écritures sur le disque de secours, c'est déjà fait.

**TOUT CE QUI EST DIT A PARTIR DE MAINTENANT DOIT ETRE IDENTIQUE SUR LES DEUX MACHINES,
LANCER LES COMMANDES EN MEME TEMPS :**

Pré-requis :

Hostname : pvm4.domain1.tetaneutral.net = **IP** : 91.224.149.38

Hosntame : pvm5.domain1.tetaneutral.net = **IP** : 91.224.149.30

Leur fichier /etc/hosts ressemble à cela :

```
127.0.0.1 localhost
91.224.149.38 pvm4.domain1.tetaneutral.net
91.224.149.30 pvm5.domain1.tetaneutral.net
```

Leur /etc/resolv.conf doivent être identiques également (domaine : domain1.tetaneutral.net) :

```
domain domain1.tetaneutral.net
search domain1.tetaneutral.net
nameserver 91.224.149.36
```

Installation :

```
# apt-get install drbd8-utils gfs2-tools cman libopenais-dev ntpdate
```

Remarquons que /dev/drbd0 a été créé. Nous allons commencer par mettre en place un second disque dur grâce à Virtualbox. Taille minimum : 2 G. Maintenant que c'est fait, au redémarrage de la machine il va apparaitre dans : /dev/sdb.

Nous le partitionnons :

```
# cfdisk /dev/sdb
NEW => LOGIQUE => Taille totale (faites entré) => WRITE => yes => "q" pour sortir
```

A la suite de cela, nous obtenons la partition : /dev/sdb5.

Maintenant nous chargeons le module dans le noyau :

```
# modprobe drbd
```

S'il y a un pare-feu, il faut ouvrir en entrée et en sortie le port 7788 tcp.

Configuration des fichiers de conf : DRBD :

vi /etc/drbd.d/global_common.conf :

```
global {  
    usage-count no;  
}  
common {  
    protocol C;  
    syncer {  
        rate 100M;  
    }  
}
```

Puis on va créer le fichier /etc/drbd.d/r0.res, ce fichier contiendra la ressource (ip, node, disque..)

vi /etc/drbd.d/r0.res :

```
resource r0 {  
    device /dev/drbd0;  
    disk /dev/sdb5;  
    meta-disk internal;  
    startup {  
        wfc-timeout 120;  
        degr-wfc-timeout 120;  
        become-primary-on both;  
    }  
    disk { on-io-error detach; }  
    net {  
        allow-two-primaries;  
        after-sb-0pri discard-zero-changes;  
        after-sb-1pri discard-secondary;  
        after-sb-2pri disconnect;  
    }  
    on pvm4 {  
        address 91.224.149.38:7788;  
    }  
    on pvm5 {  
        address 91.224.149.30:7788;  
    }  
}
```

This guide describes DRBD version 8.4 and above. For 8.3 please look [here](#).

11.2. Creating a DRBD resource suitable for GFS

Since GFS is a shared cluster file system expecting concurrent read/write storage access from all cluster nodes, any DRBD resource to be used for storing a GFS filesystem must be configured in [dual-primary mode](#). Also, it is recommended to use some of DRBD's [features for automatic recovery from split brain](#). And, it is necessary for the resource to switch to the primary role immediately after startup. To do all this, include the following lines in the resource configuration:

```
resource <resource> {
  startup {
    become-primary-on both;
    ...
  }
  net {
    allow-two-primaries yes;
    after-sb-0pri discard-zero-changes;
    after-sb-1pri discard-secondary;
    after-sb-2pri disconnect;
    ...
  }
  ...
}
```

Once you have added these options to [your freshly-configured resource](#), you may [initialize your resource as you normally would](#). Since the `allow-two-primaries` option is set to `yes` for this resource, you will be able to [promote the resource](#) to the primary role on both nodes.

GFS2 :

Tout d'abord connaitre la version de gfs2 :

```
# apt-cache policy gfs2-tool
```

Ici il faut créer le fichier :

```
# vi /etc/cluster/cluster.conf
```

```

<?xml version="1.0"?>
<cluster name="maumau_cluster" config_version="3">
  <cman two_node="1" expected_votes="1"/>
  <clusternodes>
    <clusternode name="pvm4" votes="1" nodeid="1">
      <fence>
        <method name="single">
          <device name="manual" ipaddr="91.224.149.38"/>
        </method>
      </fence>
    </clusternode>
    <clusternode name="pvm5" votes="1" nodeid="2">
      <fence>
        <method name="single">
          <device name="manual" ipaddr="91.224.149.30"/>
        </method>
      </fence>
    </clusternode>
  </clusternodes>
  <fence_daemon clean_start="1" post_fail_delay="0" post_join_delay="3"/>
  <fencedevices>
    <fencedevice name="manual" agent="fence_manual"/>
  </fencedevices>
</cluster>

```

Synchronisation du temps :

```
# ntpdate -u ntp.ovh.net
```

Si sur votre machine vous avez le service network-manager, il faut qu'il soit stoppé :

```
# service network-manager stop
```

Puis on lance la création du cluster :

```
# service cman start
```

Pour vérifier :

```
# cman_tool nodes
```

Il faut que l'on voit les deux machines, avec l'heure de la synchronisation et il faut impérativement qu'elles aient le statut à M (membre du cluster), s'il y a X (non membre du cluster), alors vous devez vérifier votre fichier /etc/hosts.

Création du RAID primaire/primaire :

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

MAINTENANT QUE TOUTES CES MANIPULATIONS ONT ETE REALISEES SUR LES DEUX MACHINES, NOUS ALLONS MANIPULER AU CAS PAR CAS, TOUT D'ABORD :

Sur un premier node (prenons pvm4) on va démarrer le service drbd :

```
# service drbd start
```

Il faut maintenant aller sur le second node (pvm5) et démarrer le service :

```
# service drbd start
```

Voilà, le démarrage a pu se terminer vu que l'autre machine a lancé la commande.

Il y a un beau message d'erreur, c'est tout à fait normal, il n'y a pas encore de primaire encore.

Il faut choisir un node (prenons pvm4) et commencer par lui "dire" qu'elle sera primaire :

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

Une synchronisation a lieu, pour la visualiser :

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

Vous pouvez faire cette visualisation (commande watch) sur l'autre node (pvm5), vous verrez la même chose. Une fois cette synchronisation terminée, nous obtenons les statuts :

Primary/Secondary Inconsistent/Inconsistent.

Il faut alors aller sur le seconde node (pvm5) et lui "dire" qu'il est primaire, mais pas avec la même commande que la précédente :

```
# drbdadm primary r0
```

On vérifie :

```
# service drbd status
```

Nous voyons que nous sommes à Primary/Primary UpToDate/UpToDate

Ici les machines sont en attentes d'un montage.

Application du cluster :

Nous allons formater le périphérique DRBD /dev/drbd0 non pas en ext3 mais en gfs2 (cluster) :

Sur 1 seul node (pvm4) :

```
# mkfs.gfs2 -t maumau_cluster:gfs2 -p lock_dlm -j 2 /dev/drbd0
```

"y" pour valider

Maintenant il ne reste plus qu'à créer un répertoire, sur la pvm4:

```
# mkdir -p /media/cluster/drbd
```

```
# mount -t gfs2 /dev/drbd0 /media/cluster/drbd
```

Et, sur la pvm5 :

```
# mkdir -p /media/clusterdata/drbd
```

```
# mount -t gfs2 /dev/drbd0 /media/clusterdata/drbd
```

Vérification :

On vérifie :

```
# service drbd status
```

Il faut que l'on voit nos point de montage.

Sur une machine, n'importe laquelle, écrivez sur votre point de montage, par exemple pvm4 :

```
# touch /media/cluster/drbd/fichier.txt
```

```
# ls /media/cluster/drbd
```

On voit notre fichier : fichier.txt

Sur l'autre node du cluster (pvm5) :

```
# ls /media/clusterdata/drbd
```

On voit notre fichier : fichier.txt

Grâce à GFS2, il n'est pas nécessaire de démonter le point de montage sur le node (pvm5) puis de le remonter afin de voir les écritures, en effet l'écriture se fait mais on ne peut la visualiser tant que le umount/mount n'a pas été réalisé.

BIBLIOGRAPHIE

http://doc.ubuntu-fr.org/tutoriel/mirroring_sur_deux_serveurs
<http://www.howtoforge.com/howtos/high-availability>
<http://www.howtoforge.com/setting-up-network-raid1-with-drbd-on-ubuntu-11.10-p2>
<http://www.howtoforge.com/setting-up-network-raid1-with-drbd-on-debian-squeeze-p2>
http://publications.ibfavre.org/system/drbd_introduction
<http://www.crium.univ-metz.fr/docs/system/drbd/>
http://wiki.kogite.fr/index.php/R%C3%A9plication_de_syst%C3%A8mes_de_fichiers_avec_DRBD
<http://www.system-linux.eu/index.php?post/2010/06/01/Installation-de-Drbd-pour-de-la-haute-disponibilit%C3%A9>
http://www.piemontewireless.net/Storage_on_Cluster_DRBD_and_GFS2
<http://blog.inforeseau.com/2011/03/installation-gfs-pour-utilisation-avec-volume-drbd-sous-centos-5-5-ou-redhat>
<http://gcharriere.com/blog/?p=73>
<http://radiobytes.wordpress.com/2012/01/03/nfs-haute-disponibilite-avec-drbd-et-gfs2/>
http://www.clusterlabs.org/wiki/Validation_FAQ
<http://www.howtoforge.com/setting-up-an-active-active-samba-ctdb-cluster-using-gfs-and-drbd-centos-5.5-p2>