

Sommaire :

Sommaire :	1
Introduction.....	1
Configuration de DRBD :.....	2
Configuration de Pacemaker - Corosync :	7
Configuration de ProFTPD :	9
ANNEXES :.....	12

Introduction

Objectif : L'objectif de cette situation professionnelle est de mettre en place un service FTP hautement disponible pour les utilisateurs. C'est-à-dire qu'en cas de panne d'un des serveurs FTP, un second serveur doit pouvoir répondre aux requêtes des utilisateurs, en totale transparence. Pour cela, des logiciels de répartition de charge et de synchronisation de données seront installés et configurés.

Pré-requis : Il faut des connaissances en service FTP, en IP virtuelle et en RAID 1

Norme : Toutes les commandes issues d'une machine avec un système d'exploitation Debian ou Windows sont écrites ***en gras et en italique***.

Configuration de DRBD :

Pour commencer, nous devons attribuer des adresses IP à nos deux serveurs, et changer leurs noms d'hôtes, tout d'abord en accédant au fichier */etc/network/interfaces* :

```
# The primary network interface
allow-hotplug eth0
iface eth0 inet static
    address 192.168.1.118
    netmask 255.255.255.0
    gateway 192.168.1.254
```

```
# The primary network interface
allow-hotplug eth0
iface eth0 inet static
    address 192.168.1.119
    netmask 255.255.255.0
    gateway 192.168.1.254_
```

Puis aux fichiers */etc/hostname* et */etc/hosts* :

```
GNU nano 2.2.6      Fichier : /etc/hostname
noeud1_
```

```
GNU nano 2.2.6      Fichier : /etc/hosts
127.0.0.1    localhost
192.168.1.118 noeud1
192.168.1.119 noeud2_
```

```
GNU nano 2.2.6      Fichier : /etc/hostname
noeud2_
```

```
GNU nano 2.2.6      Fichier : /etc/hosts
127.0.0.1    localhost
192.168.1.118 noeud1
192.168.1.119 noeud2_
```

Et ajouter les sources :

```
GNU nano 2.2.6      Fichier : /etc/apt/sources.list
#
# deb cdrom:[Debian GNU/Linux 8.5.0 _Jessie_ - Official amd64 CD Binary-1 20160606]
deb cdrom:[Debian GNU/Linux 8.5.0 _Jessie_ - Official amd64 CD Binary-1 20160606]
deb http://ftp.fr.debian.org/debian/ jessie main
deb-src http://ftp.fr.debian.org/debian/ jessie main
deb http://security.debian.org/ jessie/updates main
deb-src http://security.debian.org/ jessie/updates main
# jessie-updates, previously known as 'volatile'
deb http://ftp.fr.debian.org/debian/ jessie-updates main
deb-src http://ftp.fr.debian.org/debian/ jessie-updates main
```

Après un redémarrage, nous pouvons ensuite installer DRBD avec la commande ***apt-get install drbd8-utils*** :

```
root@noeud1:~# apt install drbd8-utils_
```

Ensuite, il faut configurer le fichier de configuration principal de DRBD sur le nœud principal, qui est le fichier ***/etc/drbd.d/global_common.conf*** :

```
global {
    usage-count no;
}
common {
    protocol C;

    net {
        allow-two-primaries;
    }

    syncer {
        rate 650M;
    }
}
```

Il faut ensuite créer un fichier de ressource pour notre futur service FTP, nommé ***/etc/drbd.d/ftp.res***, sur le nœud principal :

```
resource ftp {
    device /dev/drbd0;
    disk /dev/sda8;

    on noeud1 {
        address 192.168.1.118:7788;
        meta-disk internal;
    }

    on noeud2 {
        address 192.168.1.119:7788;
        meta-disk internal;
    }
}
```

Pour obtenir le résultat suivant :

```
resource ftp {
    device /dev/drbd0;
    disk /dev/sda8;

    on noeud1 {
        address 192.168.1.118:7788;
        meta-disk internal;
    }

    on noeud2 {
        address 192.168.1.119:7788;
        meta-disk internal;
    }
}
```

Ces fichiers de configuration doivent également être présents sur le deuxième serveur. Pour éviter de les recopier, nous allons les transférer via la commande **scp**. Pour cela, il est nécessaire d'installer le service SSH sur le nœud 2, et d'autoriser la connexion avec l'utilisateur **root** :

```
root@noeud2:~# apt install ssh_
```

Puis dans **/etc/ssh/sshd_config** :

```
GNU nano 2.2.6      Fichier : sshd_config
SyslogFacility AUTH
LogLevel INFO

# Authentication:
LoginGraceTime 120
PermitRootLogin yes_
StrictModes yes
```

Après avoir redémarré le service SSH sur le nœud 2, nous pouvons transférer nos fichiers de configurations du nœud 1 au nœud 2 avec les commandes :

```
root@noeud1:~# scp /etc/drbd.d/global_common.conf root@noeud2:/etc/drbd.d/global_
_common.conf
root@noeud1:~# scp /etc/drbd.d/ftp.res root@noeud2:/etc/drbd.d/ftp.res_
```

Et nous vérifions la présence de ces fichiers sur le nœud 2 :

```
root@noeud2:~# ls -l /etc/drbd.d/
total 8
-rw-r--r-- 1 root root 193 juin  6 10:19 ftp.res
-rw-r--r-- 1 root root 2053 juin  6 10:17 global_common.conf
```

Pour la suite, nous devons créer des métadonnées sur la partition que nous voulons synchroniser, sur **les deux nœuds**. La partition utilisée ici sera la partition `/dev/sda8` qui contient le répertoire `/home`. Nous devons d'abord démonter cette partition et la remettre à zéro pour pouvoir créer les métadonnées, avec les commandes `umount /dev/sda8` et `shred -zvf -n 1 /dev/sda8` :

```
root@noeud1:~# umount /dev/sda8
root@noeud1:~# shred -zvf -n 1 /dev/sda8
shred: /dev/sda8 : passe 1/2 (random)...
shred: /dev/sda8 : passe 1/2 (random)...299MiB/3,7GiB 7 %
shred: /dev/sda8 : passe 1/2 (random)...447MiB/3,7GiB 11 %
shred: /dev/sda8 : passe 1/2 (random)...473MiB/3,7GiB 12 %
shred: /dev/sda8 : passe 1/2 (random)...495MiB/3,7GiB 13 %
```

Nous pouvons ensuite créer les métadonnées avec la commande `drbdadm create-md ftp` :

```
root@noeud1:~# drbdadm create-md ftp
initializing activity log
NOT initializing bitmap
Writing meta data...
New drbd meta data block successfully created.
```

Puis il faut activer notre ressource `ftp.res` avec la commande `drbdadm up ftp` :

```
root@noeud1:~# drbdadm up ftp
```

Ensuite, sur le nœud 1, nous pouvons lancer la synchronisation des données, en mode Primaire/Secondaire, afin que les données s'écrivent sur le nœud 1 soient transférables sur le nœud 2, avec la commande `drbdadm --overwrite-data-of-peer primary ftp`, et nous vérifions l'avancement de la synchronisation avec `cat /proc/drbd` :

```
root@noeud1:~# cat /proc/drbd
version: 8.4.3 (api:1/proto:86-101)
srcversion: 1A9F77B1CA5FF92235C2213
0: cs:SyncSource ro:Primary/Secondary ds:UpToDate/Inconsistent C r---n-
ns:60992 nr:0 dw:0 dr:62352 al:0 bm:3 lo:0 pe:3 ua:1 ap:0 ep:1 wo:f oos:3818
340
[>.....] sync'ed: 1.6% (3818340/3876708)K
finish: 0:02:10 speed: 29,184 (29,184) K/sec
```

Lorsque la première synchronisation est effectuée, il faut maintenant formater le disque virtuel `/dev/drbd0` avec la commande `mkfs.ext4 /dev/drbd0` :

```
root@noeud1:~# mkfs.ext4 /dev/drbd0
mke2fs 1.42.12 (29-Aug-2014)
En train de créer un système de fichiers avec 969177 4k blocs et 242400 i-noeuds
.
UUID de système de fichiers=c06dd1f4-21f1-4e77-8100-bf388a026a22
Superblocs de secours stockés sur les blocs :
    32768, 98304, 163840, 229376, 294912, 819200, 884736

Allocation des tables de groupe : complété
Écriture des tables d'i-noeuds : complété
Création du journal (16384 blocs) : complété
Écriture des superblocs et de l'information de comptabilité du système de
fichiers : complété
```

Puis à monter le disque virtuel sur le répertoire **/home** avec la commande **mount /dev/drbd0 /home**, afin de pouvoir récupérer les données de ce répertoire :

```
root@noeud1:~# mount /dev/drbd0 /home
```

Pour tester notre configuration, nous pouvons créer des documents dans le répertoire **/home**, puis effectuer la bascule, avec :

Pour pouvoir tester la synchronisation, il est nécessaire sur le nœud primaire de :

- démonter la partition : **umount /dev/drbd0** ;
- changer le statut de la ressource pour « secondaire » : **drbdadm secondary resftp** ;

Et sur le nœud secondaire :

- changer le statut de la ressource pour « primaire » : **drbdadm primary resftp** ;
- monter la partition : **mount /dev/drbd0 /home**

Et les documents sont bien transférés sur le nœud 2.

Il est également possible de positionner les deux nœuds en primaire avec **drbdadm primary ftp** sur les deux nœuds, et de monter/démonter la partition lorsque nous voulons synchroniser les données.

Configuration de Pacemaker - Corosync :

Nous passons ensuite à l'installation de Pacemaker – Corosync. Ce logiciel n'étant plus disponible, il faut ajouter les backports sur les deux nœuds pour pouvoir le télécharger, soit :

```
cat > /etc/apt/sources.list.d/jessie-backports.list << "EOF"
deb http://http.debian.net/debian jessie-backports main
EOF
```

Nous pouvons ensuite mettre à jour les machines et installer les logiciels :

```
# apt-get update
# apt-get install -t jessie-backports pacemaker crmsh
```

Il faut ensuite configurer Corosync sur le nœud 1, dans le fichier */etc/corosync/corosync.conf* :

```
# Please read the corosync.conf.5 manual page
# Debian-HA ClustersFromScratch sample config
totem {
    version: 2

    cluster_name: debian

    token: 3000
    token_retransmits_before_loss_const: 10

    clear_node_high_bit: yes

    crypto_cipher: aes256 # default was 'none'
    crypto_hash: sha1 # default was 'none'

    interface {
        ringnumber: 0

        # set address of the network here; default was '127.0.0.1'
        bindnetaddr: 192.168.122.0

        mcastaddr: 239.255.1.1
        mcastport: 5405

        ttl: 1
    }
}

logging {
    fileline: off

    to_stderr: no
    to_logfile: no
    to_syslog: yes

    syslog_facility: daemon
    debug: off

    timestamp: on
    logger_subsys {
        subsys: QUORUM
        debug: off
    }
}

quorum {
    provider: corosync_votequorum
    two_node: 1 # value added
    expected_votes: 2
}
```

Il faut ensuite créer une clé d'authentification sur le nœud 1 :

```
root@node01:~# corosync-keygen
```

Puis transférer le fichier de configuration */etc/corosync/corosync.conf* et la clé générée du nœud 1 vers le nœud 2 avec la commande *scp* :

```
root@node01:~# scp /etc/corosync/corosync.conf root@node02:/etc/corosync/corosync.conf
root@node01:~# scp /etc/corosync/authkey root@node02:/etc/corosync/authkey
```

Nous pouvons ensuite stopper et démarrer les deux services sur les deux nœuds :

```
service corosync start
service pacemaker start
```

Et vérifier la configuration :

```
crm status
```

Ça fonctionne :

```
2 nodes and 0 resources configured
Online: [ noeud1 noeud2 ]
Full list of resources:
```

Nous pouvons ensuite ajouter une IP virtuelle afin que les utilisateurs l'utilisent pour se connecter au serveur FTP, et pour qu'ils soient automatiquement redirigés vers le serveur principal :

```
# crm configure property stonith-enabled=false
# crm configure property no-quorum-policy=ignore
# crm configure primitive VIP-1 ocf:heartbeat:IPaddr2 params ip="192.168.14.35" nic="eth0:0" cidr_netmask="24" op monitor interval="30s"
timeout="20s"
```

Un **crm status** permet de vérifier le bon fonctionnement :

```
2 nodes and 1 resource configured
Online: [ noeud1 noeud2 ]
Full list of resources:
VIP (ocf::heartbeat:IPaddr2): Started noeud1
```

Nous pouvons utiliser **crm node standby noeud1** et **crm node online noeud1** pour vérifier le basculement.

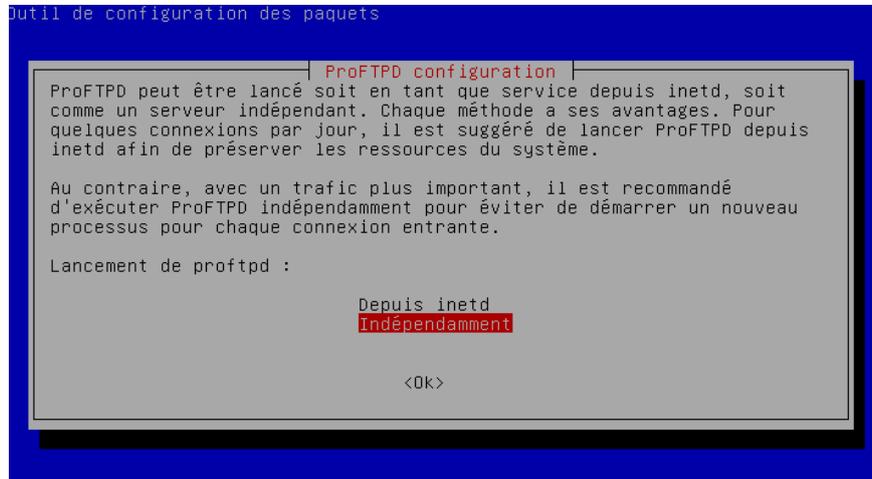
Configuration de ProFTPD :

Avant tout, nous allons créer des comptes utilisateurs pour nos futurs clients FTP, avec les commandes **adduser sebastien**, et **chown -R sebastien /home/sebastien** pour ajouter les bons droits.

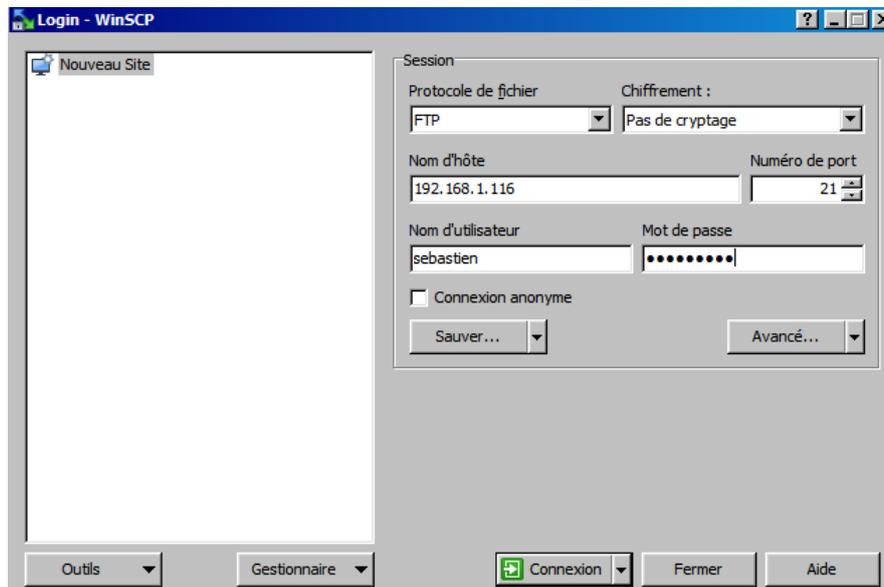
Nous pouvons ensuite installer le service ProFTPD :

```
root@noeud1:~# apt install proftpd
```

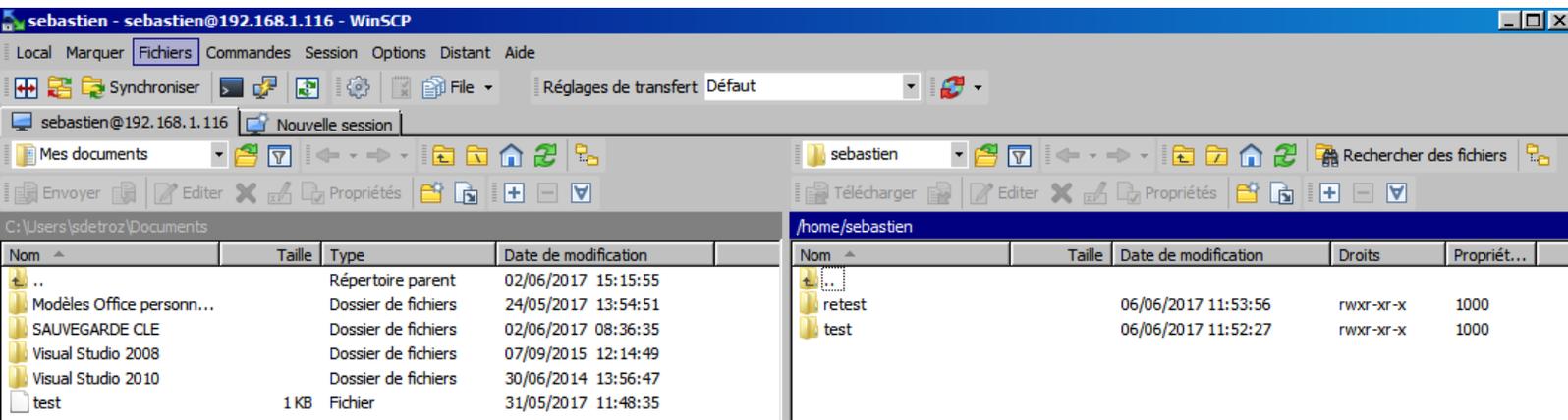
Nous le configurons « *indépendamment* » :



Nous pouvons ensuite essayer de nous connecter avec notre utilisateur, en utilisant l'IP virtuelle de nos serveurs :



Et tout fonctionne convenablement :



ANNEXES :

Documentation :

<https://www.sebastien-han.fr/blog/2011/07/04/introduction-au-cluster-sous-linux/>

<https://doc.ubuntu-fr.org/pacemaker>

<https://www.yanx.eu/pacemaker-et-corosync/>

<https://wiki.debian.org/Debian-HA/ClustersFromScratch>

https://doc.ubuntu-fr.org/tutoriel/pacemaker_configuration_ip_virtuelle_plus_script_lsb

<https://www.hisyl.fr/ip-failover-avec-hearbeat-et-pacemaker/>

<http://www.reseaucerta.org/sites/default/files/haServiceFTP-v2.pdf>

http://wiki.tuxunix.com/index.php/HA_%2B_DRBD

https://wikit.firewall-services.com/doku.php/tuto/nas_san/openfiler_ha

<https://www.howtoforge.com/openfiler-2.99-active-passive-with-corosync-pacemaker-and-drbd>

<https://wiki.debian.org/fr/DrBd>

<http://www.adminpasbete.fr/debian-cluster-drbd-2-noeuds-cible-iscsi/>

https://wiki.kogite.fr/index.php/R%C3%A9plication_de_syst%C3%A8mes_de_fichiers_avec_DRBD

Je configure l'accès en anonyme dans /etc/proftpd/proftpd.conf :

```
<Anonymous ~ftp>
  User                      ftp
  Group                      nogroup
#   # We want clients to be able to login with "anonymous" as well as "ftp"
  UserAlias                  anonymous ftp
#   # Cosmetic changes, all files belongs to ftp user
  DirFakeUser  on ftp
  DirFakeGroup on ftp
#
  RequireValidShell         off
#
#   # Limit the maximum number of anonymous logins
  MaxClients                 10
#
#   # We want 'welcome.msg' displayed at login, and '.message' displayed
#   # in each newly chdired directory.
  DisplayLogin               welcome.msg
```

(

Après avoir redémarré et éteint le service FTP, j'ajoute cette ressource à Corosync :

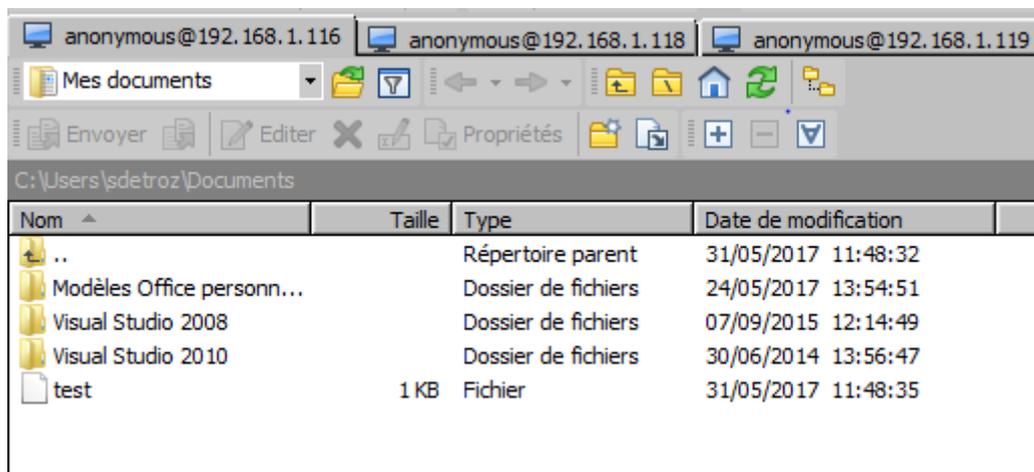
```
crm configure primitive FTP ocf::heartbeat:proftpd op monitor interval="30s"
```

Et je constate son bon fonctionnement :

```
2 nodes and 2 resources configured
Online: [ noeud1 noeud2 ]
Full list of resources:
VIP-1 (ocf::heartbeat:IPaddr2):      Started noeud1
FTP   (ocf::heartbeat:proftpd):      Stopped
Failed Actions:
* FTP_start_0 on noeud2 'not configured' (6): call=18, status=complete, exitreas
on='none',
  last-rc-change='Wed May 31 10:51:14 2017', queued=0ms, exec=6ms
```

)

Et ça fonctionne :



```

root@noeud1:~# mkfs -t ext3 /dev/sdb1
mke2fs 1.42.12 (29-Aug-2014)
En train de créer un système de fichiers avec 261888 4k blocs et 65536 i-noeuds.
UUID de système de fichiers=5bfb50d4-6040-46e5-8731-56710756de09
Superblocs de secours stockés sur les blocs :
    32768, 98304, 163840, 229376

Allocation des tables de groupe : complété
Écriture des tables d'i-noeuds : complété
Création du journal (4096 blocs) : complété
Écriture des superblocs et de l'information de comptabilité du système de
fichiers : complété

```

J'installe ensuite DRBD sur les deux nœuds :

```

root@noeud2:~# apt install drbd8-utils_

```

On crée ensuite une ressource dans le fichier `/etc/drbd.d/ftp.res` :

```

resource ftpdata {
    meta-disk internal;
    device /dev/drbd1;
    syncer {
        verify-alg sha1;
    }
    net {
        allow-two-primaries;
    }
    on noeud1 {
        disk /dev/sdb1;
        address 192.168.1.118:7788;
    }
    on noeud2 {
        disk /dev/sdb1;
        address 192.168.1.119:7788;
    }
}

```

Je crée une méta-donnée sur les deux nœuds :

```

root@noeud1:/etc/drbd.d# drbdadm create-md ftpdata_
root@noeud2:/etc/drbd.d# drbdadm create-md ftpdata_

```

- Configurer les adresses IP.
- Changer les hostnames.
- Ajouter le backports pour pacemaker.
- Télécharger pacemaker.
- Configurer corosync sur Nœud 1.
- Créer la clé d'authentification sur Nœud 1.
- Installer et activer SSH sur Nœud 2.

Copier la conf et la clé de Nœud 1 à Nœud 2.
Démarrer les deux services sur les deux nœuds.
Créer une IP virtuelle failover sur le Nœud 1.

DRBD :

lsblk

Ajouter deux disques.

Configurer les deux nœuds.

Synchroniser les nœuds.

<https://www.system-linux.eu/index.php?post/2010/06/01/Installation-de-Drbd-pour-de-la-haute-disponibilit%C3%A9>

https://wiki.deimos.fr/Installation_et_configuration_de_DRBD

<https://www.morot.fr/cluster-debian-avec-drbd-et-gfs2-glfm-n135/>

DRBD :

J'attribue des adresses IP à mes machines :

```
# The primary network interface
allow-hotplug eth0
iface eth0 inet static
    address 192.168.1.118
    netmask 255.255.255.0
    gateway 192.168.1.254
```

```
# The primary network interface
allow-hotplug eth0
iface eth0 inet static
    address 192.168.1.119
    netmask 255.255.255.0
    gateway 192.168.1.254_
```

Je donne les bons noms d'hôtes à mes machines :

```
GNU nano 2.2.6 Fichier : /etc/hostname
noeud1_
GNU nano 2.2.6 Fichier : /etc/hosts
127.0.0.1 localhost
192.168.1.118 noeud1
192.168.1.119 noeud2_
```

```
GNU nano 2.2.6 Fichier : /etc/hostname
noeud2_
GNU nano 2.2.6 Fichier : /etc/hosts
127.0.0.1 localhost
192.168.1.118 noeud1
192.168.1.119 noeud2_
```

Un reboot.

Un apt update.

J'installe ensuite DRBD sur les deux nœuds :

```
root@noeud2:~# apt install drbd8-utils_
```

Ensuite, dans /etc/drbd.d/global_common.conf sur les deux nœuds :

```
global {
    usage-count no;
    # minor-count dialog-refresh disable-ip-verification
}
common {
    protocol C;

    startup {
        become-primary-on both; # POUR PRIMARY/PRIMARY
    }
}
```

```

    }
    net {
        allow-two-primaries; # Uniquement pour PRIMARY/PRIMARY
    }
    syncer {
        rate 650M;
    }
}

```

Et dans /etc/drbd.d/ftp.res sur les deux nœuds :

```

resource ftp {
    device /dev/drbd0;
    disk /dev/sda8;

    on noeud1 {

        address 192.168.1.118:7788;
        meta-disk internal;
    }

    on noeud2 {
        address 192.168.1.119:7788;
        meta-disk internal;
    }
}

```

Ensuite, on créer les métadonnées sur les deux nœuds :

```

Umount /dev/sda8
Shred -zvf -n 1 /dev/sda8

```

SAUVEGARDE AVANT METADATA

```

Drbdadm create-md ftp
Drbdadm up ftp

```

Ensuite, sur le noeud 1 :

```

Cat /proc/drbd
Drbdadm -- --overwrite-data-of-peer primary ftp
Cat /proc/drbd

```

SAUVEGARDE AVANT GFS EN PRIMARY/SECONDARY

Sur le noeud 2 :

```

Drbdadm primary ftp

```

Cat /proc/drbd

On test :

Et sur le nœud secondaire :

- changer le statut de la ressource pour « primaire » : ***drbdadm primary resftp*** ;
- monter la partition : ***mount /dev/drdb0 /home***