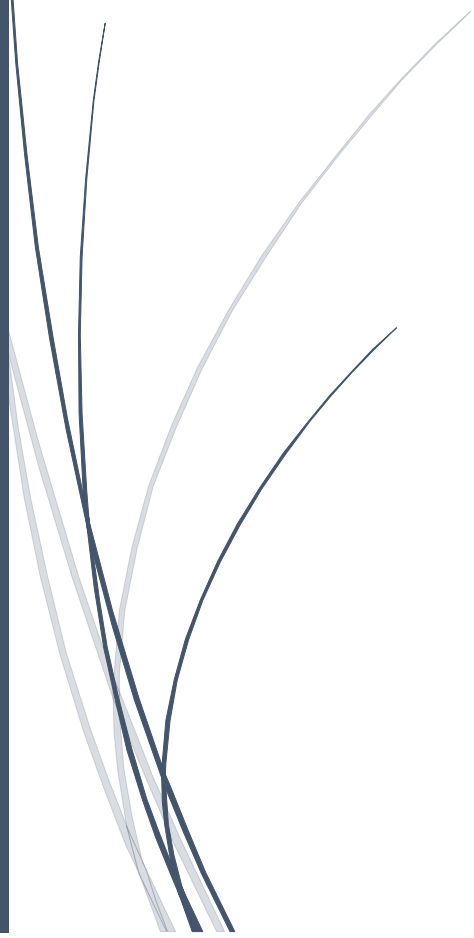
01/05/2016

HaFTP v.1

Debian 8.1



Thomas Lévêque
2SIO

Projet BTS

Mon second projet était de créer un serveur FTP avec proftpd qui grâce à un système de haute disponibilité permettrait de faire de la tolérance de panne. J'ai donc choisit d'utiliser le service Heartbeat.

Prérequis :

- Une machine sur Debian 8.1

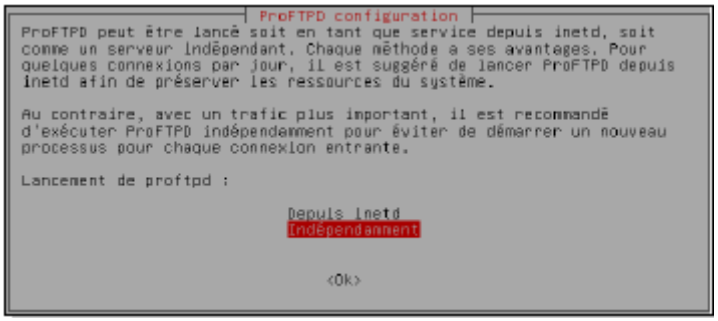
Configuration du serveur

Installation de Proftpd

Il faut d'abord installer le service proftpd.

apt-get install proftpd

On valide l'installation et on choisit Indépendamment.



Configuration de ProFTPD

Nous allons maintenant modifier ProFTPD dans son fichier de configuration.

nano /etc/proftpd/proftpd.conf

On peut changer le nom du serveur FTP en modifiant la ligne « ServeurName »

ServeurName « HaFTP1 »

On peut ensuite mettre les clients dans un dossier, pour cela il suffit de chercher la ligne « DefaultRoot ». Par défaut « ~ », mais il est possible de choisir un autre dossier et les utilisateurs qui seront cloisonné dedans.
Il faut donc mettre le chemin après « DefaultRoot » par exemple

DefaultRoot /var/www toto

Ainsi si toto ce connecte il se trouvera dans le dossier « www »

On peut aussi activer les ports en mode passif plutôt que le mode actif, c'est-à-dire :

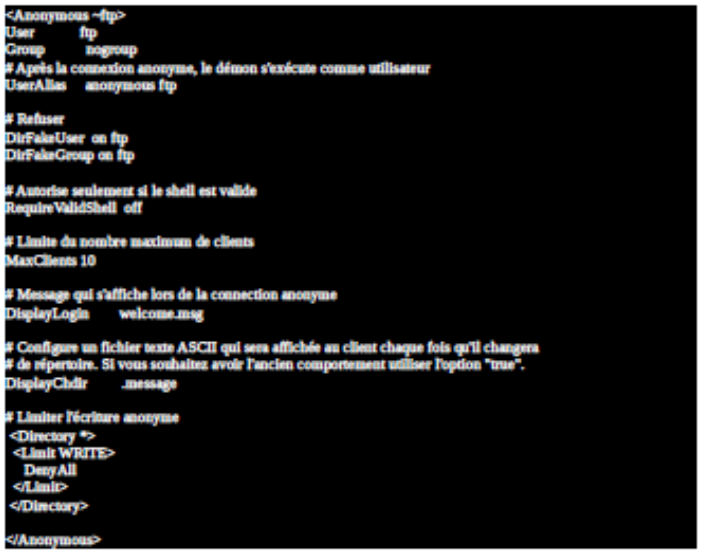
- Mode actif : Lors d'un échange c'est le client qui choisit le port.
- Mode Passif : Lors d'un échange c'est le serveur qui choisit le ports selon une plage définit.

PassivePorts 61000 62000

Ici le serveur choisira un port entre 61000 et 62000

Accès en anonyme

L'accès en anonyme permet à tout le monde de se connecter au serveur FTP sans mettre de mot de passe.



```
<Anonymous ~ftp>
User            ftp
Group           nogroup
# Après la connexion anonyme, le démon s'exécute comme utilisateur
UserAlias       anonymous ftp

# Refuser
DirFakeUser  on ftp
DirFakeGroup on ftp

# Autorise seulement si le shell est valide
RequireValidShell  off

# Limite du nombre maximum de clients
MaxClients 10

# Message qui s'affiche lors de la connection anonyme
DisplayLogin        welcome.msg

# Configure un fichier texte ASCII qui sera affichée au client chaque fois qu'il changera
# de répertoire. Si vous souhaitez avoir l'ancien comportement utiliser l'option "true".
DisplayChdir        .message

# Limiter l'écriture anonyme
 <Directory *>
  <Limit WRITE>
    DenyAll
  </Limit>
 </Directory>

</Anonymous>
```

Il suffira de redémarrer le service ProFTPD grâce à la commande :

service proftpd restart

Après avoir installé le serveur FTP il faut mettre un moyen de haute disponibilité.

On va donc faire un clone de notre machine et installer sur les deux le service heartbeat.

Prérequis :

- Ne pas configurer l'IP virtuelle sur votre système Linux. C'est Heartbeat qui s'en occupera lorsque le besoin s'en fera sentir.

- Les services qui seront lancés par Heartbeat ne doivent pas être chargés automatiquement au démarrage de Linux. C'est également Heartbeat qui se chargera de cela le moment venu.

- Enfin, Les fichiers de paramètres doivent être identiques sur les deux serveurs pour fonctionner.

On commence par installer le service

apt-get install heartbeat

Il faut après créer 3 fichiers de configuration.

/etc/heartbeat/ha.cf :

Ce fichier détermine la liste des machines à utiliser et la manière de dialoguer entre elles. Une liste exhaustive des paramètres pourra être trouvée ici : http://www.linux-ha.org/ha.cf.

```
Logfile         /var/log/ha-log
logfacility     local0
keepalive       2
deadtime        10
bcast           eth0
node            nom_serveur1 nom_serveur2
auto_failback   no
respawn         hacluster /usr/lib/heartbeat/ipfail
apiauth         ipfail gid=haclient uid=hacluster
```

logfile (facultatif)
Fichier où loguer les évènements relatifs à heartbeat (lancement, arrêt, etc.). Si vous utilisez ce paramètre, faites attention à limiter la taille des logs via le mécanisme logrotate de Debian.

logfacility
Indique quelle "facility" le syslog devra utiliser pour loguer les évènements. Les valeurs peuvent changer selon les systèmes, mais local0 reste une valeur sûre.

keepalive
Intervalle entre 2 battements de coeur. La valeur est en secondes par défaut. Pour la spécifier en millisecondes, on rajoutera 'ms' derrière. (Par exemple, 200ms).

deadtime
Temps nécessaire avant de considérer qu'un noeud est mort. Le temps est en secondes par défaut. On rajoutera 'ms' derrière pour l'avoir en millisecondes.
Attention avec cette valeur : si elle est trop courte, le système risque de s'auto-déclarer mort. Si elle est trop grande, l'autre machine mettra un temps conséquent avant de s'en apercevoir et de reprendre la main.

bcast
Spécifie l'interface réseau utilisée par Heartbeat pour envoyer les battements de coeur (en UDP).

node
Liste des machines utilisées pour la haute disponibilité, séparées par des espaces.

auto_failback
Comportement à adopter si la machine en panne revient sur le réseau. Si le paramètre est à 'Off', elle se met simplement en attente. Avec 'On', elle redevient la machine active, et celle qui fonctionne à l'heure actuelle repasse en passive.

respawn
Permet de lancer un programme au démarrage de heartbeat, qui sera tout le temps actif (Le premier paramètre est l'id de l'utilisateur qui lancera le programme. Le second, le programme lui-même).
On instancie ici ipfail, qui est à peu près le seul logiciel intéressant à lancer. Il permet d'accélérer la détection d'erreur, en regardant la disponibilité des liens réseaux en plus d'attendre un certain nombre de battements de coeur. Ceci permet au serveur passif de prendre la main dés qu'une connexion réseau est coupée, au lieu d'attendre la non réception de X battements de coeur.

apiauth
(Utilisé conjointement avec respawn). Indique quels groupes et utilisateurs ont le droit de dialoguer avec les programmes lancés par respawn. Ces programmes étant normalement utilisés pour des besoins internes, il semble logique d'interdire les accès extérieurs.

/etc/heartbeat/haresources :

Ce fichier indique les opérations à effectuer au démarrage de la haute disponibilité sur une machine

```
NOM_1ER_NODE proftpd action2 action3 etc.…
```

/etc/heartbeat/authkeys :

Ce fichier détermine la clé et le protocole de protection utilisé. Voici un exemple du fichier :

```
auth 3
3 md5 mot_de_passe
```

On notera qu'il y a 3 méthodes d'authentification :

- crc (réseaux sécurisés, comme un câble croisé par exemple)
- md5 (bonne alternative de sécurité)
- sha1 (si vous pensez vraiment que quelqu'un va essayer de vous attaquer et d'envoyer de fausses informations à heartbeat. Cette méthode est la plus sûre mais consomme plus de temps processeur)

Quoi qu'il en soit, pensez à protéger ce fichier pour qu'il ne soit plus visible par n'importe qui (si vous ne le faites pas, le programme ne se lancera pas) :

Pensez à protéger ce fichier pour qu'il ne soit plus visible par n'importe qui (si vous ne le faites pas, le programme ne se lancera pas) :

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

Avant de continuer, assurez-vous que les fichiers de configuration des deux serveurs sont identiques, et que les services sont arrêtés :

/etc/init.d/nom_service stop

Il faut ensuite faire en sorte que les services gérés par Heartbeat ne soient plus lancés automatiquement au démarrage de Linux :

update-rc.d -f nom_service remove

Vous pouvez maintenant taper (sur les 2 serveurs) :

/etc/init.d/heartbeat start

Au bout de quelques instants, les services se sont normalement lancés sur la première machine, pendant que l'autre est en attente. Voici quelques manières simples de le contrôler.