

Table des matières

TABLE DES MATIERES	1
INSTALLATION PROFTP	2
ANALYSE DE TRAME	6
MYSQL	6
INSTALLATION DE KEePALIVED	13

Avant-Propos

Un serveur FTP permet de stocker des fichiers, des répertoires et de mettre ceux-ci à la disposition des clients. Deux catégories de clients sont à considérer : les utilisateurs authentifiés avec noms et mots de passe d'une part et les anonymes d'autre part. Les utilisateurs ont le droit de déposer et de charger des fichiers à partir de leur répertoire personnel sur le serveur. Les anonymes ne peuvent que charger des documents du serveur vers leurs machines.

ProFTP utilise une syntaxe similaire à celle d'Apache permettant ainsi d'homogénéiser les fichiers de configurations.

E6 :

Elaboration de documents relatifs à la production et à la fourniture de services

A1.1.1 , Analyse du cahier des charges d'un service à produire

A1.2.4 , Détermination des tests nécessaires à la validation d'un service

A1.3.4 , Déploiement d'un service

A4.1.9 , Rédaction d'une documentation technique

Installation ProFTP

Pour installer ProFTP, il suffit de taper la commande suivante :

```
root@debian8:~# apt install proftpd
```

Lors de l'installation, on va nous poser une question, il faut répondre Indépendamment :

```
Outil de configuration des paquets

ProFTPD configuration
ProFTPD peut être lancé soit en tant que service depuis inetd, soit
comme un serveur indépendant. Chaque méthode a ses avantages. Pour
quelques connexions par jour, il est suggéré de lancer ProFTPD depuis
inetd afin de préserver les ressources du système.

Au contraire, avec un trafic plus important, il est recommandé
d'exécuter ProFTPD indépendamment pour éviter de démarrer un nouveau
processus pour chaque connexion entrante.

Lancement de proftpd :

Depuis inetd
Indépendamment

<Ok>
```

Attention, tous les utilisateurs se connectant sur le serveur ProFTP doivent exister réellement sur le système (avec un UID)

Maintenant il faut créer et modifier le fichier de conf de proftpd qui est dans /etc/proftpd/proftpd.conf

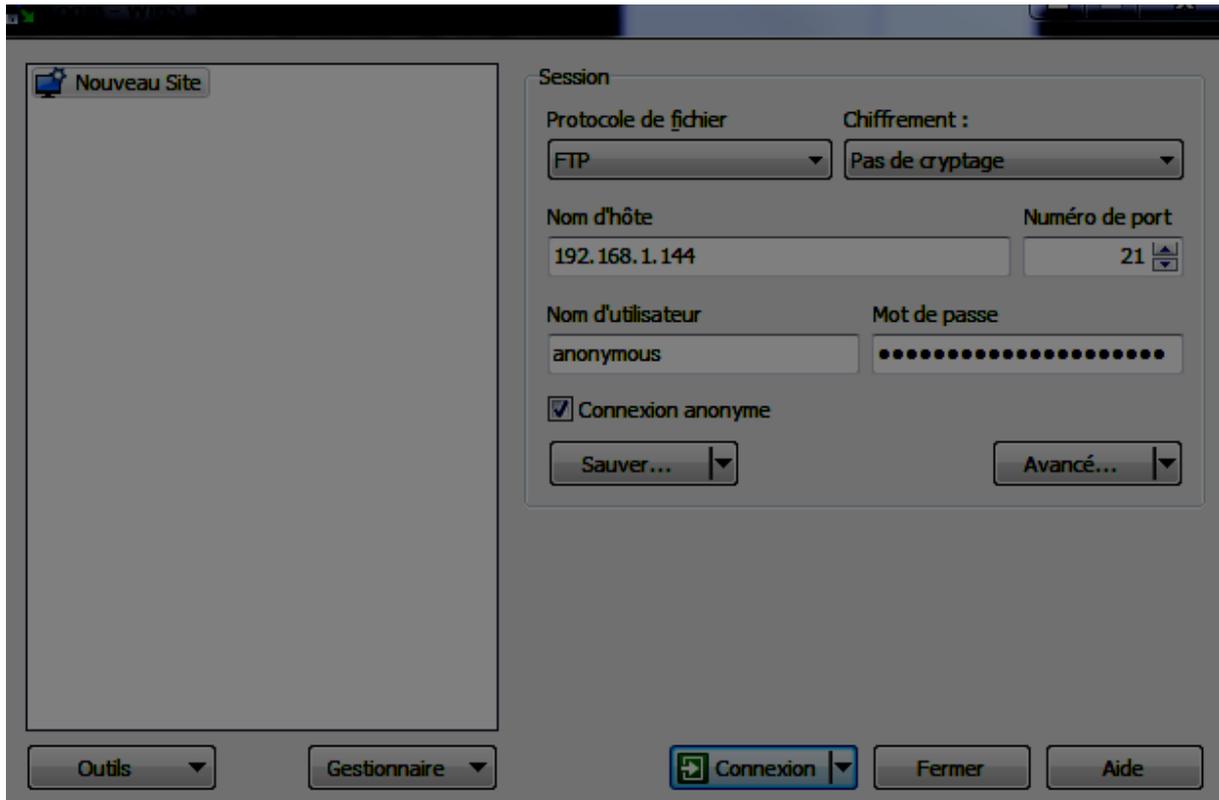
Il faut décommenter certaines lignes, et voici ce que cela donne :

```
GNU nano 2.2.6 Fichier : /etc/proftpd/proftpd.conf M
# A basic anonymous configuration, no upload directories.
<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
  DisplayChdir .message
#
# # Limit WRITE everywhere in the anonymous chroot
<Directory *>
  <Limit WRITE>
    DenyAll
  </Limit>
</Directory>
#
# # Uncomment this if you're brave.
# # <Directory incoming>
#
_</Anonymous>
```

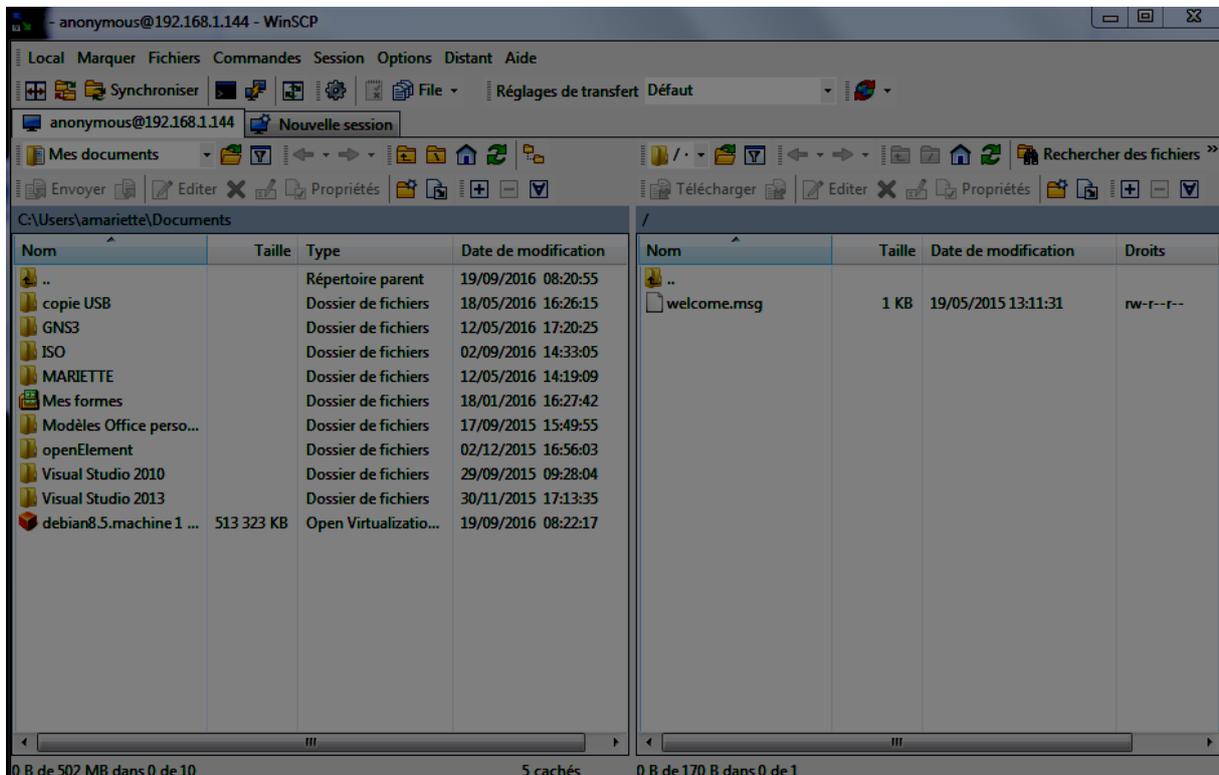
Une fois la configuration modifiée, il faut relancer le serveur.

Pour tester que le FTP fonctionne, nous utiliserons pour la suite du TP WinSCP

Il faut donc choisir sur le logiciel la connexion en FTP, puis rentrer l'adresse IP de la machine sur laquelle on veut se connecter. Vu que l'on a activé la connexion anonyme, on peut cocher la case. Il suffit ensuite de cliquer sur connexion et voilà, nous sommes connectés.



Le logiciel est séparé en deux parties, à gauche on a le poste sur lequel on est, et à droite le serveur Debian. Maintenant il suffit de faire glisser les fichiers que l'on veut dans le répertoire que l'on veut. Cependant, lorsque l'on veut se déplacer dans l'arborescence, on ne peut pas, et on ne peut pas déplacer un fichier (pour mon test une image), car nous n'avons pas encore les droits.



Analyse de trame

Nous n'avons pas eu le temps de le faire.

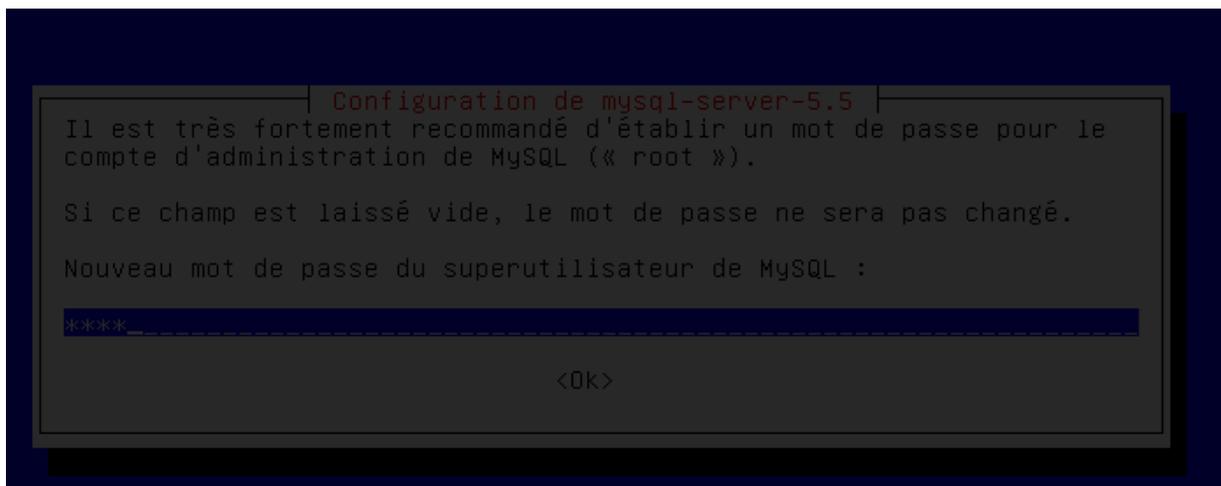
MySQL

Le but de cette installation sera de pouvoir créer des utilisateurs directement via une base de données, comme cela tous les utilisateurs créés seront dans le même groupe et auront le même ID.

Pour ce faire nous allons d'abord installer mysql-server avec la commande suivante.

```
root@debian8:~# apt install mysql-server_
```

Lors de l'installation, on nous demande de rentrer le mot de passe pour pouvoir se connecter sur la base de données.



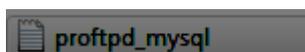
J'ai mis root comme mot de passe.

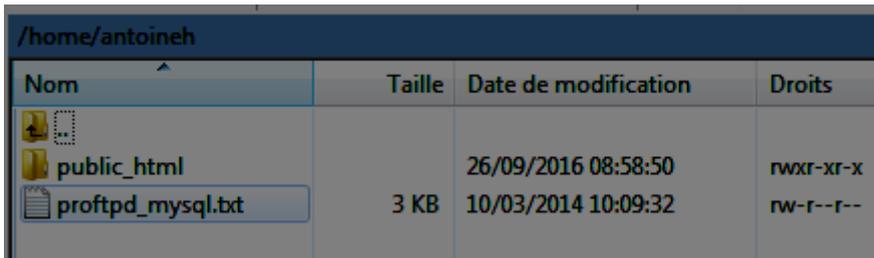
Maintenant il faut installer un paquet qui va nous servir pour le ftp.

```
root@debian8:~# apt install proftpd-mod-mysql_
```

Maintenant, nous allons devoir créer des bases de données. Cependant, cela serait trop long de le faire, notre professeur nous a donc donné un script à mettre en FTP sur le home de l'user. En l'utilisant, la base sera remplie directement.

Il faut mettre le script que notre prof nous a donné dans le home de l'user.





Nom	Taille	Date de modification	Droits
public_html		26/09/2016 08:58:50	rwxr-xr-x
proftpd_mysql.txt	3 KB	10/03/2014 10:09:32	rw-r--r--

Une fois que c'est fait on peut se connecter sur la base pour voir que tout est bien fait :

```
root@debian8:~# mysql -u root -p
```

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
+-----+
3 rows in set (0.00 sec)
```

Il nous faut maintenant créer la table, qui va contenir nos users.

```
mysql> create database proftpd;
Query OK, 1 row affected (0.01 sec)
```

Il faut ensuite utiliser la commande suivante pour pouvoir l'utiliser :

```
mysql> grant select, insert, update, delete on proftpd.* to 'proftpd'@'localhost'
identified by 'root';
Query OK, 0 rows affected (0.00 sec)
```

Il faut utiliser ensuite la commande suivante :

```
mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)
```

Maintenant, nous pouvons quitter mysql. Il faut maintenant remplir la table précédemment créée des champs, c'est là que nous allons utiliser le script de notre professeur. Il faut rentrer la commande suivant afin de remplir les champs.

```
root@debian8:~# mysql -u proftpd -p proftpd < /home/antoineh/proftpd_mysql.txt
Enter password:
```

Il faut ensuite ce connecter en superutilisateur, puis aller dans la base proftpd

```
mysql> use proftpd;_
```

Maintenant quand on liste ce qu'on à dedans :

```
mysql> desc ftpgroup;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| groupname  | varchar(16)   | NO   | MUL |          |       |
| gid        | smallint(6)   | NO   |     | 5500     |       |
| members    | varchar(16)   | NO   |     |          |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

Maintenant on veut lister ce qui est présent dans la table des utilisateurs, mais on peut constater que celle-ci est vide.

```
mysql> select * from ftpuser;
Empty set (0.00 sec)

mysql> quit
Bye
```

Nous allons maintenant nous rendre dans le fichier proftpd.conf et décommenter puis modifier certaines lignes, que voici.

```
GNU nano 2.2.6      Fichier : /etc/proftpd/proftpd.conf

ControlsSocket      /var/run/proftpd/proftpd.sock
</IfModule>

<IfModule mod_ctrls_admin.c>
AdminControlsEngine off
</IfModule>

#
# Alternative authentication frameworks
#
#Include /etc/proftpd/ldap.conf
Include /etc/proftpd/sql.conf
```

```
GNU nano 2.2.6      Fichier : modules.conf

ModuleControlsACLs insmod,rmmod allow user root
ModuleControlsACLs lsmod allow user *

LoadModule mod_ctrls_admin.c
LoadModule mod_tls.c

# Install one of proftpd-mod-mysql, proftpd-mod-pgsq
# SQL backend engine to use this module and the requ
# This module must be mandatory loaded before anyone
# the existent SQL backedds.
LoadModule mod_sql.c
```

```
GNU nano 2.2.6 Fichier : modules.conf
# Install proftpd-mod-ldap to use this
#LoadModule mod_ldap.c
#
# 'SQLBackend mysql' or 'SQLBackend postgres' (or any other valid b
# are required to have SQL authorization working. You can also comm
# unused module here, in alternative.
#
# Install proftpd-mod-mysql and uncomment the previous
# mod_sql.c module to use this.
LoadModule mod_sql_mysql.c
```

```
GNU nano 2.2.6 Fichier : sql.conf
#
# Proftpd sample configuration for SQL-based authentica
#
# (This is not to be used if you prefer a PAM-based SQL
#
<IfModule mod_sql.c>
#
# Choose a SQL backend among MySQL or PostgreSQL.
# Both modules are loaded in default configuration, so
# or comment out the unused module in /etc/proftpd/modu
# Use 'mysql' or 'postgres' as possible values.
#
SQLBackend      mysql
#
SQLEngine on
SQLAuthenticate users groups_
#
# Use both a crypted or plaintext password
SQLAuthTypes Crypt Plaintext
```

```
GNU nano 2.2.6 Fichier : sql.conf
SQLAuthTypes Backend Crypt
#
# Connection
SQLConnectInfo proftpd@localhost proftpd root_
#
# Describes both users/groups tables
#
```

Le mot de passe root est celui que l'on utilise d'habitude.

```
GNU nano 2.2.6 Fichier : sql.conf
#SQLAuthTypes Backend Crypt
#
# Connection
SQLConnectInfo proftpd@localhost proftpd root
#
# Describes both users/groups tables
#
SQLUserInfo ftpuser userid passwd uid gid homedir shell
#SQLUserWhereClause "loginallowed = 'true'"
SQLGroupInfo ftpgroup groupname gid members
CreateHome on
#
</IfModule>
```

```
GNU nano 2.2.6 Fichier : /etc/proftpd/sql.conf
#SQLAuthTypes Backend Crypt
#
# Connection
SQLConnectInfo proftpd@localhost proftpd root
#
# Describes both users/groups tables
#
SQLUserInfo ftpusers userid passwd uid gid homedir shell
#SQLUserWhereClause "loginallowed = 'true'"
SQLGroupInfo ftpgroups groupname gid members
CreateHome on
#
</IfModule>
```

Une fois toutes les modifications faites, on redémarre le service proftpd.

Il nous fait maintenant ajouter un groupe avec un gid précis, comme ça quand on créera les utilisateurs directement depuis la base de donnée ils seront ajoutés dans le bon groupe et auront directement le bon id.

```
root@debian8:/etc/proftpd# groupadd -g 5500 ftpgroup
```

```
root@debian8:/etc/proftpd# useradd -u 5500 -s /bin/false -d /bin/null -g ftpgroup ftpuser_
```

Une fois cela fait, nous allons lister les groupes et vérifier que tout a bien été ajouté.

```
root@debian8:/etc/proftpd# cat /etc/group_
```

```
utmp:x:43:
video:x:44:antoineh
sasl:x:45:
plugdev:x:46:antoineh
staff:x:50:
games:x:60:
users:x:100:
nogroup:x:65534:
input:x:101:
systemd-journal:x:102:
systemd-timesync:x:103:
systemd-network:x:104:
systemd-resolve:x:105:
systemd-bus-proxy:x:106:
crontab:x:107:
netdev:x:108:antoineh
Debian-exim:x:109:
messagebus:x:110:
mlocate:x:111:
ssh:x:112:
antoineh:x:1000:
ssl-cert:x:113:
mysql:x:114:
ftpgroup:x:5500:
```

Le groupe est bien créé, mais on peut constater que l'utilisateur n'est pas dans le groupe. Il faut utiliser la commande suivante afin de l'ajouter.

```
root@debian8:/etc/proftpd# gpasswd -a ftpuser ftpgroup
Ajout de l'utilisateur ftpuser au groupe ftpgroup
```

Maintenant nous allons nous connecter sur mysql avec l'utilisateur proftpd, avec le mot de passe proftpd, directement dans la base proftpd

```
root@debian8:/etc/proftpd# mysql -u proftpd -p proftpd
Enter password:
```

Nous allons maintenant insérer dans la base proftpd ftpuser.

```
mysql> insert into ftpgroup values ('ftpgroup',5500,'ftpuser');
Query OK, 1 row affected (0.00 sec)
```

Pour vérifier que l'entrée est bien ajoutée, on utilise la requête SQL `select * from frpgroup`

```
mysql> select * from ftpgroup;
+-----+-----+-----+
| groupname | gid | members |
+-----+-----+-----+
| ftpgroup  | 5500 | ftpuser  |
+-----+-----+-----+
1 row in set (0.00 sec)
```

Maintenant que la base est prête, nous pouvons ajouter des utilisateurs directement via celle-ci. Les utilisateurs seront directement créés dans le bon groupe et avec le bon gid.

```
mysql> insert into ftpuser values (1,'titi',encrypt('secret'),5500,5000,'/home/titi', '/sbin/nologin', '', '', '', '');
Query OK, 1 row affected, 4 warnings (0.00 sec)
```

Il faut modifier la ligne suivante dans /etc/proftpd/proftpd.conf

```
GNU nano 2.2.6      Fichier : /etc/proftpd/proftpd.conf
# DefaultRoot          ~
# Users require a valid shell listed in /etc/shells to login.
# Use this directive to release that constrain.
RequireValidShell     off
```

Une fois cela fait on redémarre le service.

Pour faire le test, nous allons tenter de nous connecter en FTP avec le logiciel WinSCP sur le serveur, avec l'utilisateur titi et le mot de passe secret. C'est l'utilisateur que nous avons créé précédemment directement dans le mysql.

```
titi@192.168.1.144
```

Installation de Keepalived

Pour la suite nous allons avoir besoin de deux serveurs avec FTP d'installé dessus. Nous avons alors cloné le serveur que nous utilisons précédemment. Les opérations qui vont suivre doivent être répétées sur les deux serveurs.

Pour installer il suffit de faire `apt-get install keepalived.conf`, puis créer le fichier de configuration suivant :

```
root@debian8:~# apt-get install keepalived_
```

Pour le moment on le fait simple, il suffit de le remplir avec ça :

```
GNU nano 2.2.6      Fichier : /etc/keepalived/keepalived.conf
vrrp_instance FTP {
    state BACKUP
    interface eth0
    virtual_router_id 50
    priority 50
    advert_int 5
    virtual_ipaddress {
        192.168.1.146
    }
}
```

Maintenant il suffit de redémarrer le service et de se connecter sur WinSCP sur l'adresse IP virtuelle avec un utilisateur quelconque. Nous avons utilisé titi pour faire le test.

Maintenant que l'on a vu que l'adresse virtuelle fonctionne, il faut tester si le service keepalive en lui-même fonctionne aussi.

Pour ce faire, on va éteindre le service sur le serveur maitre et voir si la bascule ce fait correctement sur le backup.

```
Sep 30 14:20:56 debian8 /etc/mysql/debian-start[963]: /usr/bin/mysql_upgrade: the '--basedir' option is always ignored
Sep 30 14:20:56 debian8 /etc/mysql/debian-start[963]: Looking for 'mysql' as: /usr/bin/mysql
Sep 30 14:20:56 debian8 /etc/mysql/debian-start[963]: Looking for 'mysqlcheck' as: /usr/bin/mysqlcheck
Sep 30 14:20:56 debian8 /etc/mysql/debian-start[963]: This installation of MySQL is already upgraded to 5.5.52, use --force if you still need to run mysql_upgrade
Sep 30 14:20:56 debian8 /etc/mysql/debian-start[1213]: Checking for insecure root accounts.
Sep 30 14:20:56 debian8 /etc/mysql/debian-start[1218]: Triggering myisam-recover for all MyISAM tables
Sep 30 14:20:57 debian8 Keepalived_vrrp[515]: VRRP_Instance(FTP) forcing a new MASTER election
Sep 30 14:20:57 debian8 Keepalived_vrrp[515]: VRRP_Instance(FTP) forcing a new MASTER election
Sep 30 14:21:02 debian8 Keepalived_vrrp[515]: VRRP_Instance(FTP) Transition to MASTER STATE
Sep 30 14:21:07 debian8 Keepalived_vrrp[515]: VRRP_Instance(FTP) Entering MASTER STATE
```

Une fois donc que le service est arrêté sur le serveur, on regarde les logs sur l'autre serveur pour voir s'il est bien passé en maitre.

On peut voir dans les logs que le serveur est passé en MASTER, ce qui confirme le fonctionnement de Keepalive.

Le fonctionnement du service est donc bien confirmé.

Cependant, cela n'a énormément d'intérêt. En effet, ce que l'on veut ce n'est pas particulièrement de savoir si le service keepalive est tombé en panne ou non, mais de savoir si le service FTP fonctionne toujours ou non. Nous allons modifier les fichiers de conf afin que lorsque le service FTP tombe, on bascule sur le second serveur automatiquement.

```
GNU nano 2.2.6      Fichier : /etc/keepalived/keepalived.conf
wrrp_instance FTP {
    state BACKUP
    interface eth0
    virtual_router_id 59
    priority 50
    advert_int 5

    authentication {
        auth_type PASS
        auth_pass mdpsecret
    }
    virtual_ipaddress {
        192.168.1.146
    }
}
}
virtual_server 192.168.1.146 21 {
    delay_loop 4
    lb_algo rr
    lb_kind DR
```

```
virtual_server 192.168.1.146 21 {
    delay_loop 4
    lb_algo rr
    lb_kind DR
    persistence_timeout 120
    protocol TCP

    real_server 192.168.1.145 21 {
        weight 1
        TCP_CHECK {
            connect_port 21
            connect_timeout 5
        }
    }
}
real_server 192.168.1.144 21 {
```

```
    real_server 192.168.1.144 21 {
        weight 1
        TCP_CHECK {
            connect_port 21
            connect_port 5
        }
    }
}
```

Petite erreur dans les captures, il faut mettre le port 80 à la place du port 21 ! En effet, faire du round robin avec le FTP n'est pas possible, nous l'avons découvert lors des tests. Lorsque l'on se connecte sur un serveur FTP, on reste connecté dessus, et il est impossible de basculer directement sur un autre sans authentification de la part du client. Nous avons donc décidé de faire le changement de serveur si le service web (qui utilise le port 80) tombe en panne, tant pis pour le 21 qui est celui du FTP

Après avoir redémarré le service keepalive, on peut effectuer les tests.

Pour voir si l'on arrive à se connecter maintenant, il suffit de lancer WinSCP et de se connecter sur l'adresse IP virtuelle.

Si l'on arrive à se connecter, alors c'est que le service fonctionne. Maintenant nous allons tester la continuité de service.

Il suffit pour effectuer le test d'éteindre le serveur comme précédent et de voir dans les logs si l'autre serveur est bien passé en master. Nous pouvons aussi rafraichir une page web qui est contenue sur le serveur et voir si celle-ci bascule bien d'un serveur à l'autre.