

Table des matières :

| | |
|---|----|
| Table des matières :..... | 1 |
| Objectif :..... | 2 |
| 1.Installation du serveur ProFTPD | 2 |
| 2.Accès en anonyme | 2 |
| 3.Configuration générale | 3 |
| Le mode FTP passif | 5 |
| Création de dossiers Web personnels sur le service Web: | 6 |
| 4.Analyse de trames FTP | 7 |
| 5.ProFTP avec MySQL | 7 |
| 6.Installation php5 et phpmyadmin | 13 |
| 7.Ajout 2eme serveur avec FTP | 16 |
| 8.TEST keepalived entre les serveurs..... | 17 |

Avant-Propos

Compétences :

- A1.1.1 Analyse du cahier des charges d'un service à produire
- A1.2.4 Déterminer des tests nécessaires à la validation d'un service (3)
- A4.1.9 Rédaction d'une documentation technique

Un serveur FTP permet de stocker des fichiers, des répertoires et de mettre ceux-ci à la disposition des clients. Deux catégories 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 leur machine.

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

Objectif :

Dans cette procédure, nous allons montrer comment installer et configurer un serveur FTP avec un équilibrage de charges au moyen du service Keepalived sous Debian.

| OS | Distribution | Version |
|--------|--------------|---------|
| Debian | Linux | 8.5 |

1. Installation du serveur ProFTPD

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

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>
```

2. Accès en anonyme

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

Il faut créer la section anonymous pour que les clients puissent se connecter sans authentification, en fait au nom de l'utilisateur ftp, dont le répertoire personnel est /home/ftp, et qui n'a pas de shell, comme le confirme l'examen de /etc/passwd.

Il faut aller décommenter les lignes suivantes :

```
root@debian8:/etc/proftpd# nano /etc/proftpd/proftpd.conf _
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 "f
  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>
# Include other conf
```

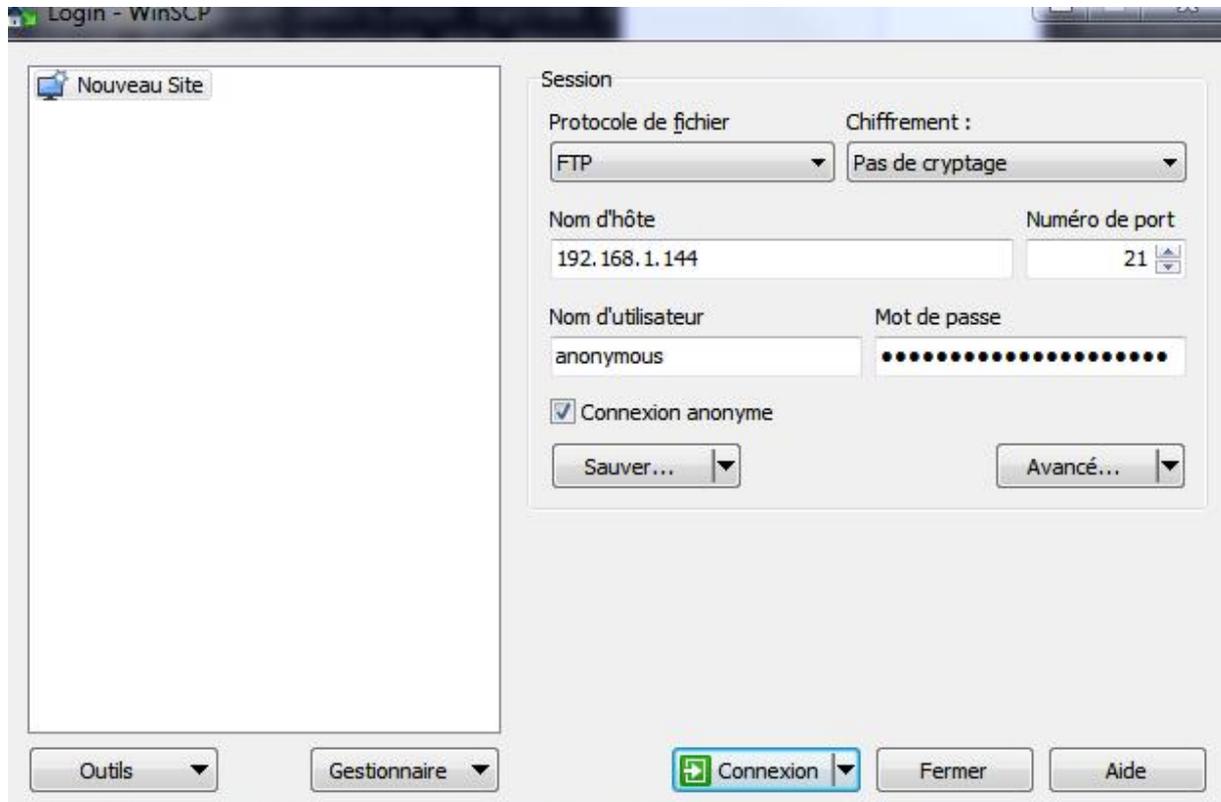
Il faut ensuite relancer le serveur

```
root@debian8:~# reboot_
```

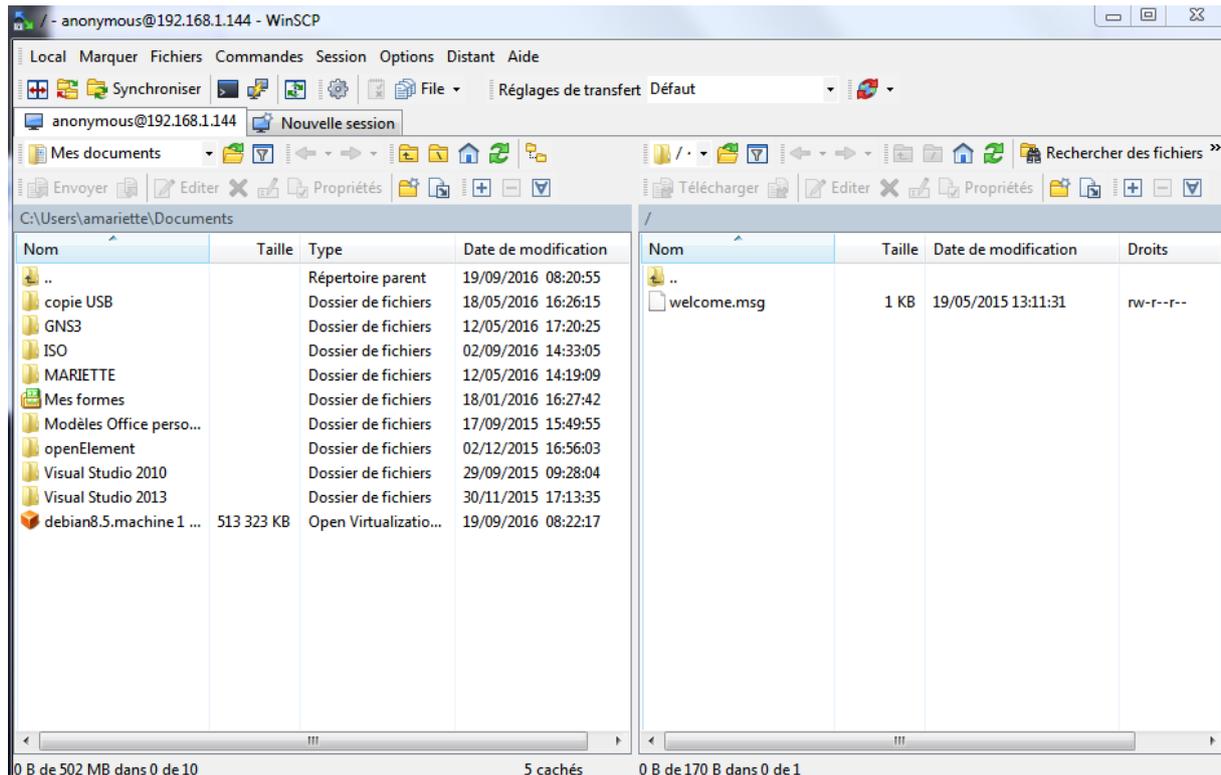
3. [Configuration générale](#)

Il faut ensuite effectuer des tests d'accès au serveur FTP : (prendre le logiciel 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.



Le mode FTP passif

Mais ce n'est pas ça qui nous intéresse pour le moment. De base, les serveurs FTP sont configurés en mode Actif. C'est le mode par défaut des clients FTP. Le client établit dans un premier temps une session TCP sur le port 21 (FTP) du serveur. Une fois la session établie et l'authentification FTP acceptée, c'est le serveur qui établit une session TCP (avec le port source 20, FTP-DATA) vers un port dynamique du client.

Le mode passif est conseillé à partir du moment où les clients peuvent se trouver derrière un Firewall/NAT. Comme le FTP actif, le client établit une première session TCP sur le port 21 (FTP) du serveur. Une fois la session établie et l'authentification FTP acceptée, on demande au serveur de se mettre en attente de session TCP grâce à la commande PASV. Alors le client peut établir une seconde session TCP sur un port dynamique vers le serveur. Le numéro de port dynamique est transmis du serveur vers le client suite à la commande PASV.

Pour activer le mode passif, il suffit simplement d'aller dans `proftpd.conf` et de décommenter la ligne suivante et de renseigner les ports que l'on veut utiliser :

```
GNU nano 2.2.6      Fichier : /etc/proftpd/proftpd.conf      Mc
# DefaultRoot          ~
# Users require a valid shell listed in /etc/shells to login.
# Use this directive to release that constrain.
# RequireValidShell    off
# Port 21 is the standard FTP port.
Port                  21
# In some cases you have to specify passive ports range to by-pass
# firewall limitations. Ephemeral ports can be used for that, but
# feel free to use a more narrow range.
# PassivePorts          49152 65534
PassivePorts          63000 65000
# If your host was NATted, this option is useful in order to
# allow passive tranfers to work. You have to use your public
```

On redemarre le serveur :

```
root@debian8:~# service proftpd restart
```

On install ensuite un apache :

```
root@debian8:~# apt install apache2
```

Pour le test on a l'utilisateur « antoineh »

On active ensuite les autorisations pour l'utilisateur antoineh qui sera le seul a accéder à la racine du serveur web (/var/www)

[Création de dossiers Web personnels sur le service Web:](#)

Le mod **UserDir** est un module installé par défaut avec Apache 2. Il permet, une fois configuré, de faire correspondre une URL de type **http://192.168.0.100/~utilisateur** à un dossier **/home/utilisateur**, par exemple.

Il faut tout d'abord activer le modèle userdir à l'aide de la commande « a2enmod userdir »

Il ne faut pas oublier de redémarrer les services après.

```
root@debianlinux:~# a2enmod userdir
Enabling module userdir.
To activate the new configuration, you need to run:
  service apache2 restart
root@debianlinux:~# service apache2 restart
root@debianlinux:~#
```

Après on peut créer le répertoire et y aller

```
root@debian8:~# mkdir /home/antoineh/public_html_
```

```
root@debian8:~# chgrp www-data /home/antoineh/public_html_
```

Il faut de nouveau redémarrer apache

```
root@debianlinux:~# service apache2 restart_
```

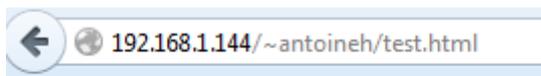
```
root@debian8:~# cd /home/antoineh/public_html/_
```

Lorsqu'on est dans le répertoire public_html, on peut créer un fichier html pour tester

```
root@debian8:/home/antoineh/public_html# nano test.html
```

```
GNU nano 2.2.6 Fichier : test.html
<h1>test public_html</h1>
```

Résultat : il ne faut oublier de mettre le caractère « ~ » avant l'utilisateur



← 192.168.1.144/~antoineh/test.html

test public_html

Autoriser seulement l'utilisateur antoineh à accéder à la racine du serveur web

(non fait)

[4. Analyse de trames FTP](#)

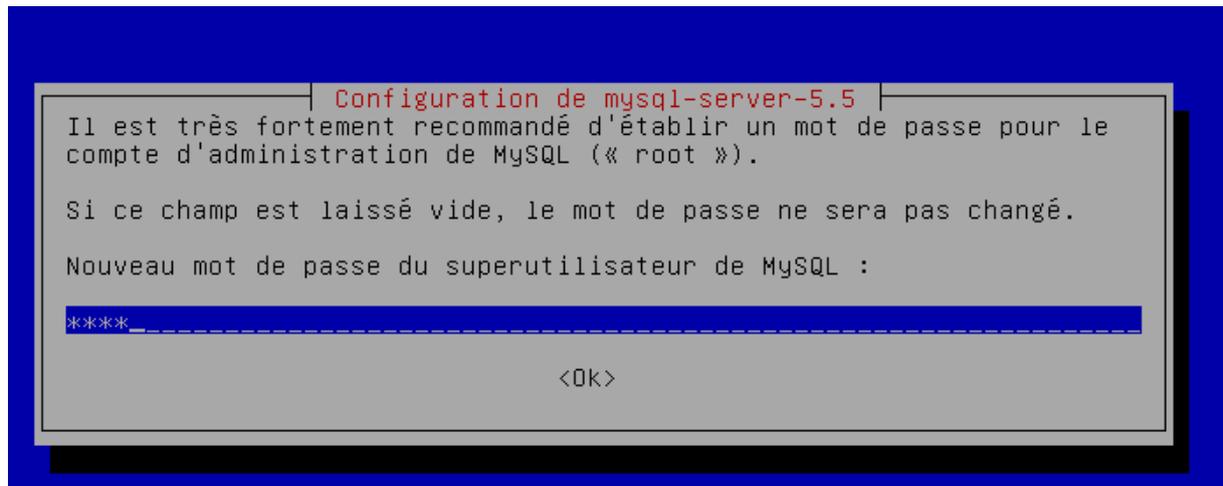
Manque de temps pour l'effectuer

[5. ProFTP avec 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.

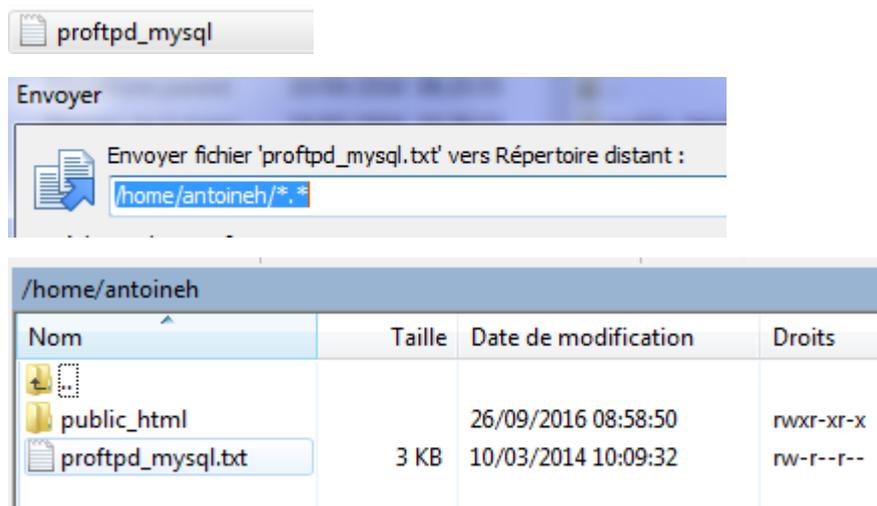
```
root@debian8:~# apt install mysql-server_
```

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



Maintenant, nous allons devoir créer des bases de donnée. 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'utilisateur. En l'utilisant, la base sera remplie directement.

Avec le fichier suivant qu'il faut mettre dans le répertoire de notre utilisateur



On peut aller ensuite se connecter sur la base de donnée :

```
mysql -u root -p
```

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

Permet de voir les bases

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

Pour cree la table

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

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

Le dernier root correspond au mot de passe

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

```
mysql> quit
Bye
```

Maintenant on va se connecter en tant que utilisateur root et non proftpd

```
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;_
```

```
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
```

On decommente la ligne

```
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 backed.  
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 decomment 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 authentication
#
# (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
#
```

Root → mot de passe de tout à l'heure

```
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
#
</TfModule>
```

On redémarre le service

```
root@debian8:/etc/proftpd# systemctl restart proftpd.service_
```

Il nous faut maintenant ajouter un groupe avec un gid précis, comme ça quand on créera les utilisateurs directement depuis la base de données 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_ Pour regarde le groupe
```

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

On tape la commande suivant pour vérifier le contenu de la table

```
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.
Création utilisateur titi avec mot de passe secret

```
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)
```

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

```
mysql> update ftpuser set shell='/bin/nologin' where id=1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

```
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
```

On redémarre le service

```
root@debian8:~# systemctl restart proftpd.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.

Maintenant on peut se connecter avec les identifiants titi et secret

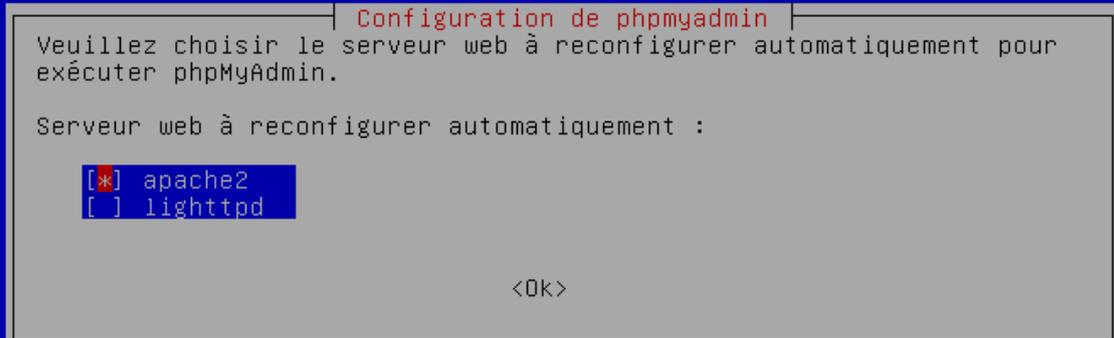
```
titi@192.168.1.144
```

[6. Installation php5 et phpmyadmin](#)

Il faut faire apt install php5

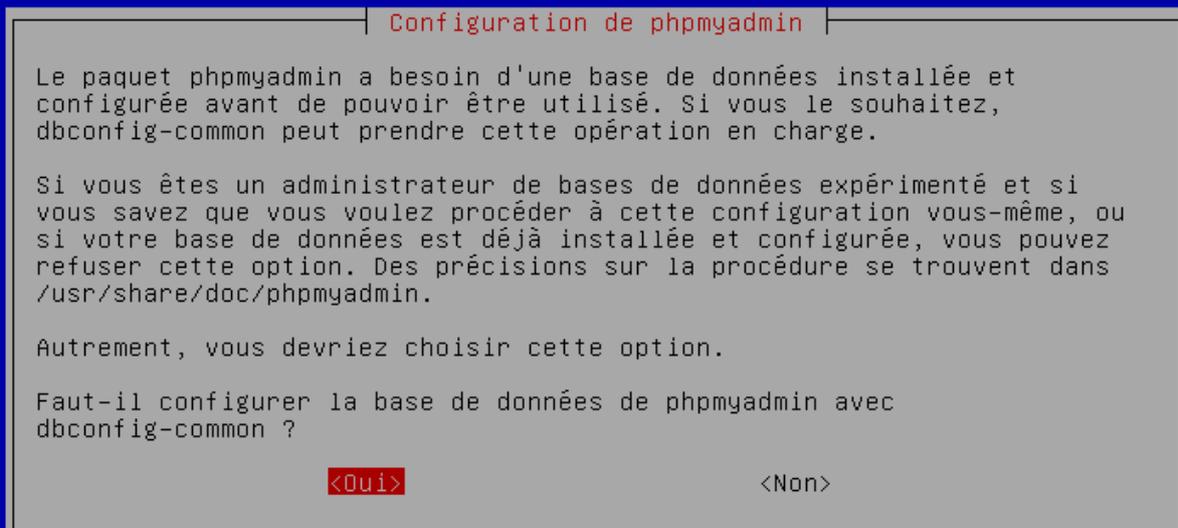
Il faut aussi faire apt install phpmyadmin

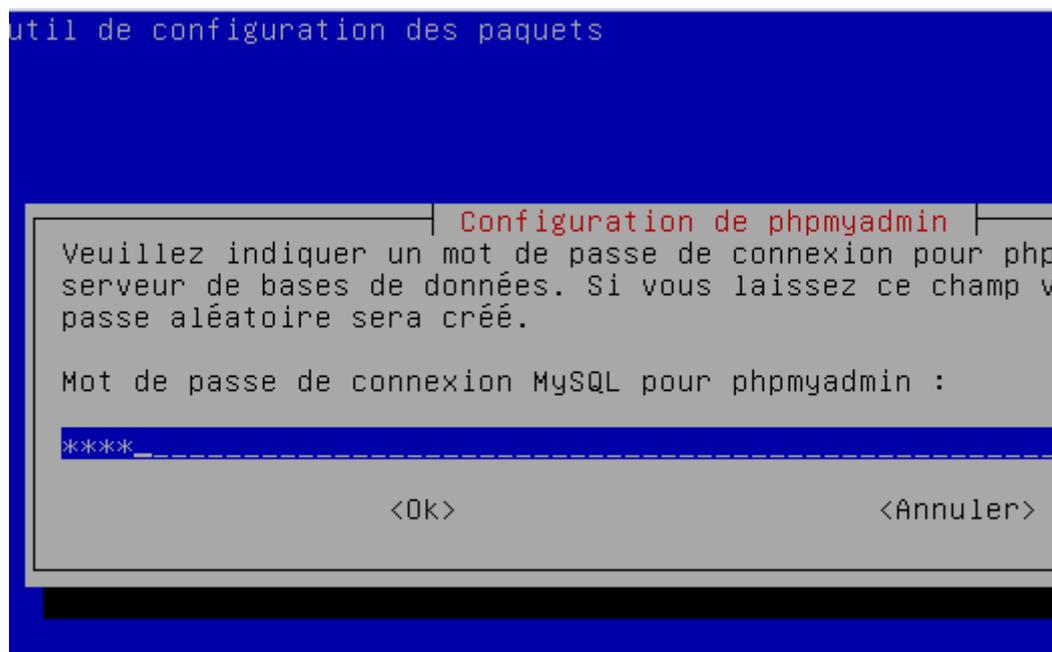
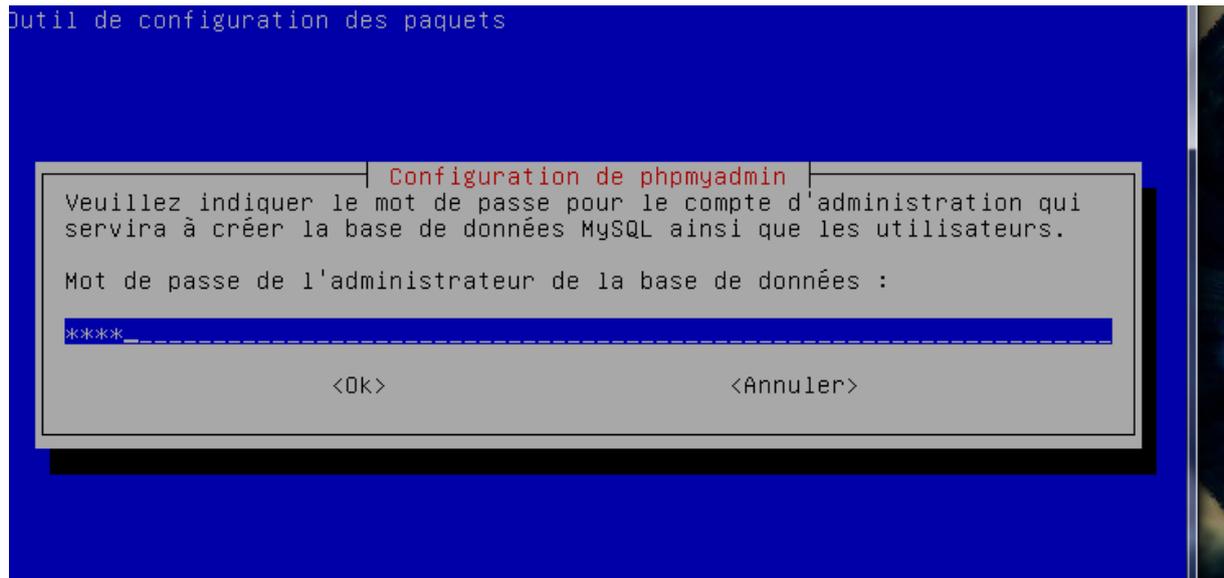
Outil de configuration des paquets



FAIRE BARRE D'ESPACE POUR VALIDER APACHE2

Outil de configuration des paquets





On peut aller ensuite se connecter avec l'identifiant « proftpd » et mdp « root »



On ajoute ensuite un utilisateur avec phpmyadmin

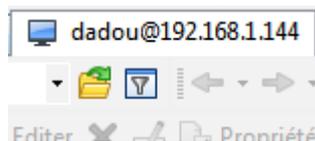
The screenshot shows the phpMyAdmin interface for the 'ftpuser' table. The table structure is as follows:

| Colonne | Type | Fonction | Null | Valeur |
|--------------|------------------|----------|------|---|
| id | int(10) unsigned | | | 2 |
| userid | varchar(32) | | | dadou |
| passwd | varchar(32) | | | root |
| uid | smallint(6) | | | 5500 |
| gid | smallint(6) | | | 5500 |
| homedir | varchar(255) | | | /home/dadou |
| shell | varchar(16) | | | /bin/false |
| count | int(11) | | | 0 |
| accessed | datetime | | | 0000-00-00 00:00:00 |
| modified | datetime | | | 0000-00-00 00:00:00 |
| LoginAllowed | enum | -- | | <input checked="" type="radio"/> true <input type="radio"/> false |

At the bottom of the interface, there is a table with the following data:

| | | | | | | | | | | |
|---|-------|------|------|------|-------------|------------|---|---------------------|---------------------|------|
| 2 | dadou | root | 5500 | 5500 | /home/dadou | /bin/false | 0 | 0000-00-00 00:00:00 | 0000-00-00 00:00:00 | true |
|---|-------|------|------|------|-------------|------------|---|---------------------|---------------------|------|

On se connecte ensuite sur winscp



7. Ajout 2eme serveur avec FTP

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.

On clone la 1^{er} ftp, avec la 2^{eme}, on change l'adresse ip, son hosts et hostname

On va installer keepalived sur les 2 ftp

```
root@debian8:~# apt-get install keepalived_
root@proftpd2:~# apt-get install keepalived_
```

Puisqu'il faut le faire sur les 2 serveurs,

On peut directement transférer le fichier vers une autre machine avec la commande suivante :

```
root@debian8:~# scp /etc/keepalived/keepalived.conf root@192.168.1.145:/etc/keepalived/keepalived.conf_
```

On change la priority

```
GNU nano 2.2.6 Fichier : /etc/keepalived/keepalived.conf
vrp_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.

8. TEST keepalived entre les serveurs

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 keepalived en lui-même fonctionne aussi.

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

Test de la réponse des deux serveurs entre lorsqu'on en éteint un, l'autre serveur doit prendre la suite, il est alors MASTER.

Pour vérifier cela, on éteint un des deux serveurs, pour celui qui reste allumer, on vérifie c'est log :

```
root@debian8:~# tail /var/log/syslog _
```

```

Sep 30 14:20:56 debian8 /etc/mysql/debian-start[963]: /usr/bin/mysql_upgrade: the
e '--basedir' option is always ignored
Sep 30 14:20:56 debian8 /etc/mysql/debian-start[963]: Looking for 'mysql' as: /u
sr/bin/mysql
Sep 30 14:20:56 debian8 /etc/mysql/debian-start[963]: Looking for 'mysqlcheck' a
s: /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_upgra
de
Sep 30 14:20:56 debian8 /etc/mysql/debian-start[1213]: Checking for insecure roo
t 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 M
ASTER election
Sep 30 14:20:57 debian8 Keepalived_vrrp[515]: VRRP_Instance(FTP) forcing a new M
ASTER election
Sep 30 14:21:02 debian8 Keepalived_vrrp[515]: VRRP_Instance(FTP) Transition to M
ASTER STATE
Sep 30 14:21:07 debian8 Keepalived_vrrp[515]: VRRP_Instance(FTP) Entering MASTER
STATE

```

Au vue de la capture, on voit bien que le serveur est passé master.

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.

Maintenant on va configurer les fichiers de keepalived pour que lorsqu'on arrête proftpd sur un serveur, le proftpd du 2eme prend la suite.

On commence par modifier le fichier keepalived, il faut le faire sur les 2 serveurs :

```

root@proftpd2:~# nano /etc/keepalived/keepalived.conf

```

```

GNU nano 2.2.6      Fichier : /etc/keepalived/keepalived.conf
vrrp_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 keepalived, 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.

On redémarre enfin les services de keepalived

```
root@proftp2:~# systemctl restart keepalived.service
```

```
root@debian8:~# systemctl restart keepalived.service  
root@debian8:~#
```