DETROZ Sébastien Version 1 18/10/2016

Sommaire :

Introduction..... 1
1 – Complétez la description de votre domaine pour prendre en compte votre serveur FTP : 2
2 – Décrivez la procédure employée pour effectuer un test complet du domaine DNS : 4
3 – Votre serveur fonctionne sous Debian, quelle commande utilisez-vous pour installer ProFTP : 6
4 – Paramétrer l'accès anonyme puis faire les tests : 7
5 – Configurer le mode passif. En quoi est-il intéressant : 9
6 – Installer le serveur SQL MariaDB (préciser les mots de passe) et les modules nécessaires : 9
7 – Créer la base de données usersftp puis les tables nécessaires pour la gestion des utilisateurs : .. 11
8 – Insérer un jeu de données : 13
9 – Faire les tests. Conclure : 15
10 – Installer l'interface graphique Adminer : 15
11 – Ajouter l'utilisateur Jean Bonneau : 18
12 – Réaliser une sauvegarde de la base usersftp avec ses données : 20

Introduction

Objectif : L'objectif de ce TP est de mettre en place un service FTP et un service DNS sur un domaine local, dont les utilisateurs seront créés via une base de données MariaDB.

Pré-requis : Il faut des connaissances en service FTP, en service DNS et en base de données MariaDB.

Nous disposons de deux machines Debian :

VM	Adresse IP	Nom
Debian FTP	192.168.1.116	sebftp
Debian DNS	192.168.1.117	sebdns

Le serveur FTP doit utiliser un nom générique ***sebftp.stseb.local*** !

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

1 – Complétez la description de votre domaine pour prendre en compte votre serveur FTP :

Pour compléter la description de notre domaine, nous avons besoin d'installer un service DNS. Nous allons donc mettre à jour nos machines virtuelles avec un ***apt update***, puis lancer ***apt install bind9*** sur une machine pour installer un service DNS.

Ensuite, il faut modifier les fichiers ***/etc/resolv.conf*** des deux Debian afin d'insérer les informations de notre nouveau DNS :

```
#domain sio.local
#search sio.local
#nameserver 192.168.1.49
#nameserver 192.168.1.50
#nameserver 8.8.8.8
#nameserver 81.253.149.6
#nameserver 80.10.246.136
#nameserver 192.168.1.254

domain stseb.local
search stseb.local
nameserver 192.168.1.117
```

Nous allons maintenant transformer notre serveur en un serveur maître pour un domaine que nous allons baptiser ***stseb.local.*** Nous accédons donc au fichier ***nano /etc/bind/named.conf.local*** pour commencer la configuration du serveur maître.

Il faut préciser à notre serveur que nous allons créer des fichiers de zones. On déclare notre domaine, puis notre domaine en inversé :

```
zone "stseb.local" IN {
        type master;
        file "/etc/bind/meszones/zone.stseb.local";
};

zone "1.168.192.in-addr.arpa" IN {
        type master;
        file "/etc/bind/meszones/1.168.192.in-addr.arpa";
};
```

Par défaut, les fichiers de zones sont à placer dans ***/var/cache/bind***, c'est pourquoi on peut directement écrire le chemin complet de l'emplacement de nos fichiers de zones, ou modifier le chemin par défaut de ***named.conf.options***.

Dans le répertoire, qu'il faut préalablement créer, ***/etc/bind/meszones***, on créer ensuite le fichier ***zone.stseb.local*** :

```
$TTL 86400

stseb.local. IN SOA sebdns.stseb.local. root.stseb.local. (
        2016101801      ; serial
        86400           ; refresh
        21600           ; retry
        3600000         ; expire
        3600            ; minimum
)

stseb.local. IN NS sebdns.

sebdns.stseb.local. IN A 192.168.1.117
sebftp.stseb.local. IN A 192.168.1.116_
```

On peut ensuite créer le fichier de zone inversée ***1.168.192.in-addr.arpa*** :

```
$TTL 86400

1.168.192.in-addr.arpa. IN SOA sebdns.stseb.local. root.stseb.local. (
        2016101801      ; serial
        86400           ; resfresh
        21600           ; retry
        3600000         ; expire
        3600            ; minimum
)

1.168.192.in-addr.arpa. IN NS sebdns.stseb.local.

117 IN PTR sebdns.stseb.local.
116 IN PTR sebftp.stseb.local._
```

Le serveur DNS est maintenant installé et configuré, il ne reste plus qu'à tester la validité des fichiers de configurations, et son fonctionnement.

2 – Décrivez la procédure employée pour effectuer un test complet du domaine DNS :

Pour vérifier le fonctionnement du DNS, nous pouvons tout d'abord utiliser la commande ***named-checkconf fichierdeconf*** qui permet de vérifier la validité du fichier de configuration ***named.conf.local*** :

```
root@sebdns:/etc/bind# named-checkconf named.conf.local
root@sebdns:/etc/bind# _
```

Si la commande ne renvoie rien, le fichier est correct. Avec la commande ***named-checkzone domaine fichier***, on peut tester la validité des fichiers de zone :

```
root@sebdns:/etc/bind/meszones# named-checkzone 1.168.192.in-addr.arpa 1.168.192
.in-addr.arpa
zone 1.168.192.in-addr.arpa/IN: loaded serial 2016101802
OK
root@sebdns:/etc/bind/meszones# named-checkzone stseb.local zone.stseb.local
zone stseb.local/IN: loaded serial 2016101801
OK
```

Le retour pour le numéro de série du fichier doit être « ***OK*** ». Ensuite, sur le serveur FTP, qui est client de notre DNS, nous pouvons essayer une série de ***ping*** :

- Sur la loopback :

```
root@sebftp:~# ping 127.0.0.1
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.043 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.044 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.049 ms
```

- Sur le serveur DNS :

```
root@sebftp:~# ping 192.168.1.116
PING 192.168.1.116 (192.168.1.116) 56(84) bytes of data.
64 bytes from 192.168.1.116: icmp_seq=1 ttl=64 time=0.044 ms
64 bytes from 192.168.1.116: icmp_seq=2 ttl=64 time=0.049 ms
64 bytes from 192.168.1.116: icmp_seq=3 ttl=64 time=0.050 ms
```

Mais le plus significatif et le plus important sera la commande ***nslookup***, qui permet de résoudre des noms en adresses IP, et inversement, en utilisant notre DNS. Nous pouvons donc tester une recherche directe ou une recherche inversée sur les deux machines :

```
> sebdns
Server:         192.168.1.117
Address:        192.168.1.117#53

Name:   sebdns.stseb.local
Address: 192.168.1.117
```

```
> 192.168.1.117
Server:         192.168.1.117
Address:        192.168.1.117#53

117.1.168.192.in-addr.arpa      name = sebdns.stseb.local.
```

```
> sebftp
Server:         192.168.1.117
Address:        192.168.1.117#53

Name:   sebftp.stseb.local
Address: 192.168.1.116
```

```
> 192.168.1.116
Server:         192.168.1.117
Address:        192.168.1.117#53

116.1.168.192.in-addr.arpa      name = sebftp.stseb.local.
```

Notre serveur DNS fonctionne donc parfaitement. La commande ***dig*** permet la même chose que la commande nslookup, mais avec beaucoup plus de détails. En voici un exemple :

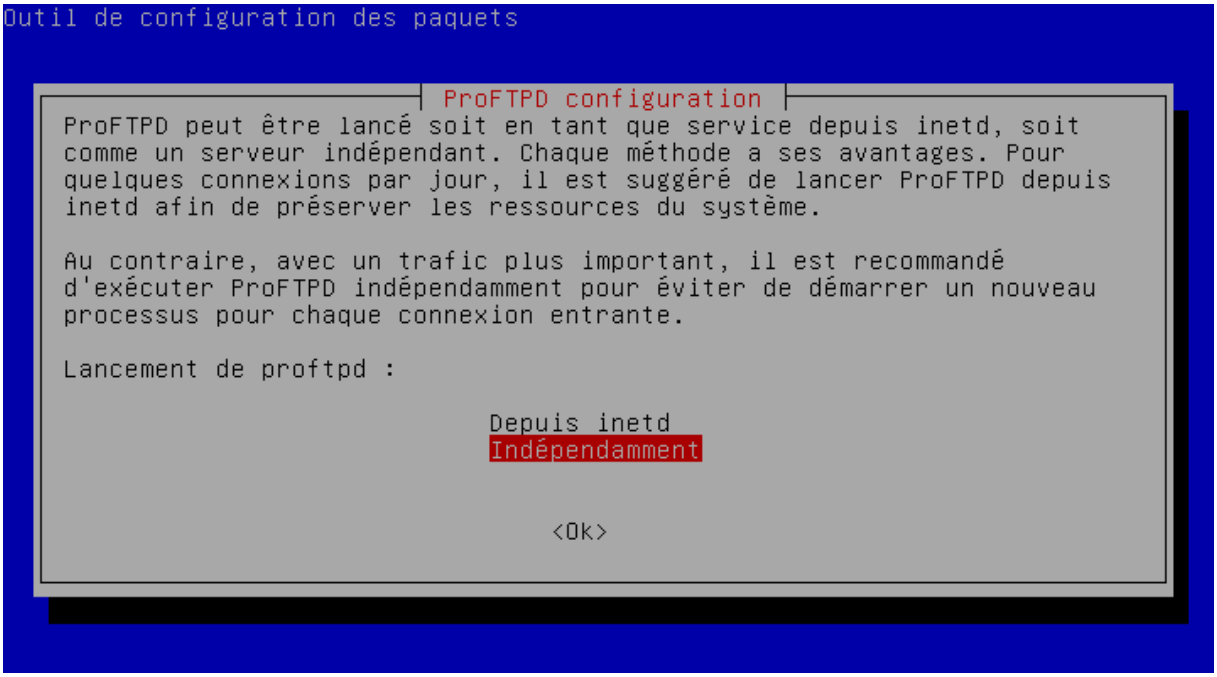
```
root@sebftp:~# dig sebdns

; <<>> DiG 9.9.5-9+deb8u6-Debian <<>> sebdns
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NXDOMAIN, id: 24575
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;sebdns.                                IN      A

;; AUTHORITY SECTION:
.                       10800   IN      SOA     a.root-servers.net. nstld.verisi
gn-grs.com. 2016101800 1800 900 604800 86400

;; Query time: 70 msec
;; SERVER: 192.168.1.117#53(192.168.1.117)
;; WHEN: Tue Oct 18 09:58:48 CEST 2016
;; MSG SIZE  rcvd: 110
```

3 – Votre serveur fonctionne sous Debian, quelle commande utilisez-vous pour installer ProFTP :

Nous utilisons la commande ***apt install proftpd*** pour installer le service FTP ProFTPD:

```
root@sebftp:~# apt-get install proftpd
```

Nous devons le paramétrer en mode « standalone » (soit « Indépendamment ») lors de l'installation :



Un ***systemctl status proftpd*** permet d'attester de la bonne installation du service :

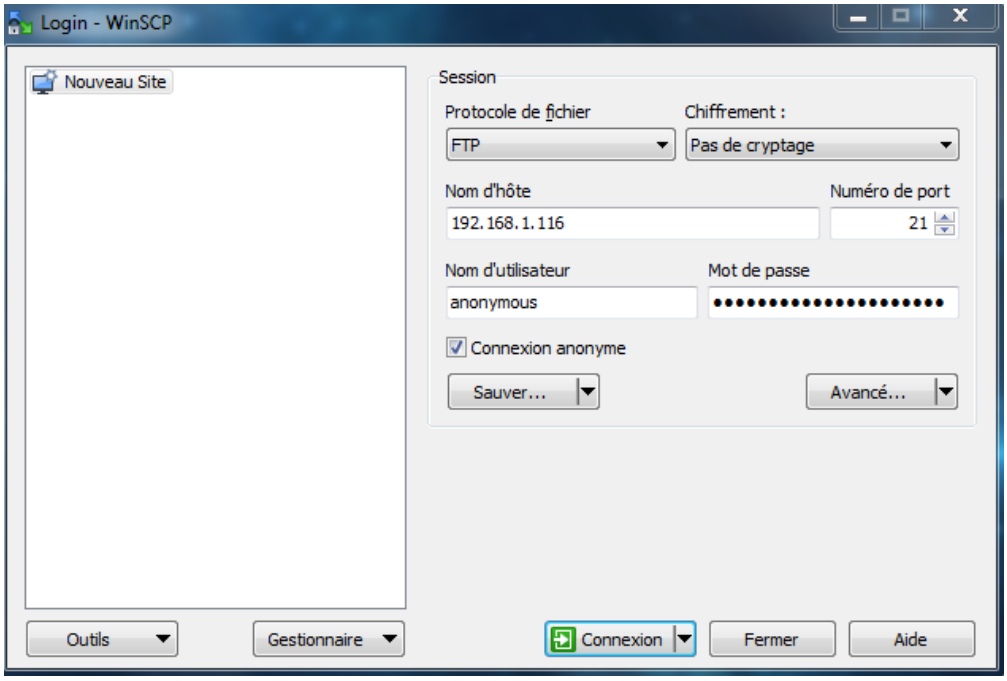
```
root@sebftp:~# systemctl status proftpd
● proftpd.service - LSB: Starts ProFTPD daemon
   Loaded: loaded (/etc/init.d/proftpd)
   Active: active (running) since mar. 2016-10-18 11:30:25 CEST; 34min ago
  Process: 20240 ExecStop=/etc/init.d/proftpd stop (code=exited, status=0/SUCCES
S)
  Process: 20248 ExecStart=/etc/init.d/proftpd start (code=exited, status=0/SUCC
ESS)
   CGroup: /system.slice/proftpd.service
           └─20255 proftpd: (accepting connections)

oct. 18 11:30:25 sebftp proftpd[20248]: Starting ftp server: proftpd.
oct. 18 11:30:25 sebftp systemd[1]: Started LSB: Starts ProFTPD daemon.
```

4 – Paramétrer l'accès anonyme puis faire les tests :

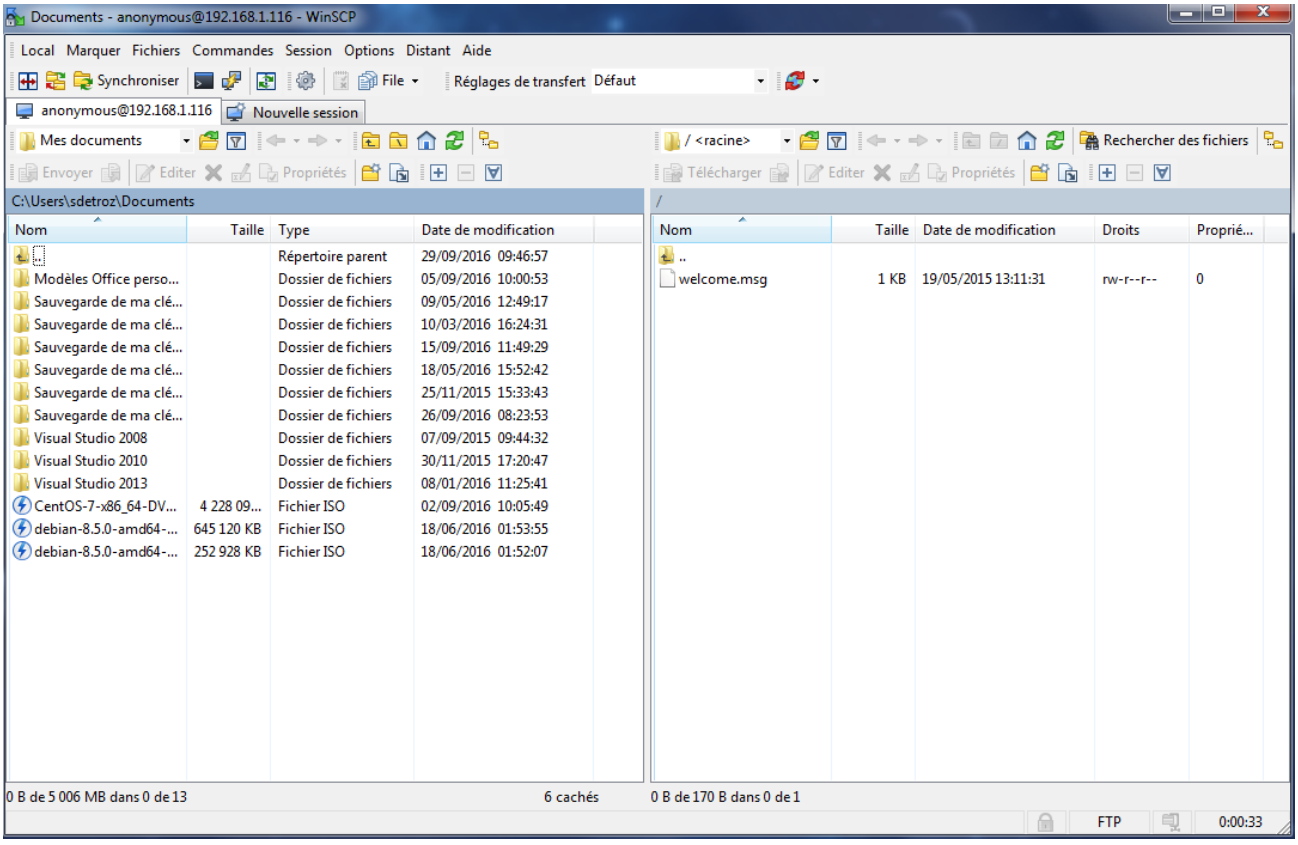
Pour autoriser les anonymes à se connecter, nous devons utiliser la section ***anonymous*** pour que les clients puissent se connecter sans authentification. Nous allons donc modifier le fichier ***/etc/proftpd/proftpd.conf***. Il n'y a que quelques lignes à dé-commenter, car une section ***anonymous*** existe déjà :

```
  GNU nano 2.2.6                          Fichier : 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
   DisplayChdir               .message

#    # Limit WRITE everywhere in the anonymous chroot
   <Directory *>
     <Limit WRITE>
       DenyAll
     </Limit>
   </Directory>
#
#    # Uncomment this if you're brave.
#    # <Directory incoming>
#    #    # Umask 022 is a good standard umask to prevent new files and dirs
#    #    # (second parm) from being group and world writable.
#    #    Umask                           022  022
#    #             <Limit READ WRITE>
#    #             DenyAll
#    #             </Limit>
#    #             <Limit STOR>
#    #             AllowAll
#    #             </Limit>
#    # </Directory>
#
 </Anonymous>

# Include other custom configuration files
Include /etc/proftpd/conf.d/
```

Une fois le fichier de configuration modifié, nous relançons le serveur avec un ***systemctl restart proftpd.*** Nous essayons ensuit de nous connecter en anonyme :



Et cela fonctionne :



5 – Configurer le mode passif. En quoi est-il intéressant :

Nous allons ensuite paramétrer le mode passif, afin que le serveur FTP fournisse lui-même le numéro de port à utiliser aux clients. Sans cela, l'utilisateur serait bloqué par les pare-feu. Nous modifions donc la ligne suivante dans le fichier de configuration ***/etc/proftpd/proftpd.conf :***

```
PassivePorts                  63000 65000
```

Il devient alors intéressant d'analyser les trames, et de vérifier que le serveur FTP octroie bien un port aux utilisateurs anonymes, situé entre 63000 et 65000 (ici, 64143) :

```
192.168.1.116    192.168.1.58    FTP     106 Response: 227 Entering Passive Mode (192,168,1,116,250,143).
192.168.1.116    192.168.1.58    TCP      60 64143→49576 [ACK] Seq=352 Ack=2 Win=29312 Len=0
```

6 – Installer le serveur SQL MariaDB (préciser les mots de passe) et les modules nécessaires :

Nous avons besoin d'installer les trois programmes suivants : le serveur de base de données ***MariaDB***, le serveur web ***Apache 2*** et le langage ***PHP 5***. Nous pouvons les télécharger et les installer l'un après l'autre avec les commandes : ***apt install mariadb-server apache2 php5***

Lors de l'installation de ***MariaDB***, il faut définir le mot de passe de l'utilisateur ***root***, ici nous utiliserons « ***root*** » :

Lorsque les installations sont terminées, pour vérifier si tout fonctionne correctement, nous pouvons faire les commandes ***systemctl status nomduservice***, par exemple :

```
root@sebdb:~# systemctl status mysql
● mysql.service - LSB: Start and stop the mysql database server daemon
   Loaded: loaded (/etc/init.d/mysql)
   Active: active (running) since lun. 2016-10-03 09:06:02 CEST; 15min ago
   CGroup: /system.slice/mysql.service
           ├─2359 /bin/bash /usr/bin/mysqld_safe
           ├─2360 logger -p daemon.err -t /etc/init.d/mysql -i
           ├─2498 /usr/sbin/mysqld --basedir=/usr --datadir=/var/lib/mysql --
           └─2499 logger -t mysqld -p daemon.error
```

/!\ ATTENTION : ***MariaDB*** est installé sous le nom de ***MySQL***.

Nous pouvons également accéder aux différents services via un navigateur, pour ***Apache 2*** :

Ainsi qu'après la création d'une page ***info.php*** contenant la variable ***phpinfo()*** :

/!\ ATTENTION : Pour que ProFTP fonctionne avec MariaDB, il faut aussi installer le paquet ***proftpd-mod-mysql***.

7 – Créer la base de données usersftp puis les tables nécessaires pour la gestion des utilisateurs :

Pour créer la base de données ***usersftp***, nous nous connectons à notre service ***MariaDB*** avec la commande : ***mysql –u root –p*** (u pour user, p pour password) Nous avons ensuite accès à MariaDB :

```
root@sebftp:/etc/proftpd# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 42
Server version: 10.0.27-MariaDB-0+deb8u1 (Debian)

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> _
```

Un ***show databases ;*** permet de voir les bases initialement présentent :

```
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
+--------------------+
3 rows in set (0.00 sec)
```

Nous créons ensuite la base ***usersftp*** avec la commande : ***create database usersftp ;*** Un ***show databases ;*** peut attester de sa création :

```
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| usersftp           |
+--------------------+
4 rows in set (0.00 sec)
```

Nous donnons maintenant des droits en sélection, insertion, mises à jour, et suppression à l'utilisateur ***usersftp*** sur la base ***usersftp***, car nous en aurons besoin plus tard pour insérer des données autrement qu'en ***root*** :
grant select, insert, update, delete on usersftp.* to 'usersftp'@'localhost' identified by 'usersftp' ; flush privileges ;

quit ;

```
MariaDB [(none)]> grant select, insert, update, delete on usersftp.* to 'usersft
p'@'localhost' identified by 'usersftp';
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> quit;
Bye
```

Nous importons ensuite le fichier de données ***proftpd_mysql.txt***, que nous avons précédemment inséré dans notre Debian via FTP, afin de remplir la base de données ***usersftp*** :

mysql –u root –p usersftp < /root/ proftpd_mysql.txt

Nous nous connectons pour vérifier la création de la base, et lister son contenu :

mysql –u usersftp –p
show databases ;

```
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| usersftp           |
+--------------------+
2 rows in set (0.00 sec)
```

use usersftp ;
show tables ;

```
MariaDB [usersftp]> show tables;
+--------------------+
| Tables_in_usersftp |
+--------------------+
| ftpgroup           |
| ftpquotalimits     |
| ftpquotatotal      |
| ftpuser            |
+--------------------+
4 rows in set (0.00 sec)
```

8 – Insérer un jeu de données :

Pour insérer un utilisateur, nous devons modifier un certain nombre de choses pour autoriser son authentification sur le serveur FTP. Tout d'abord, dans ***/etc/proftpd/modules.conf***, on dé-commente :

LoadModule mod_sql.c
LoadModule mod_sql_mysql.c

Puis, dans **/etc/proftpd/sql.conf**, on dé-commente :

SQLBackend mysql
SQLEngin on
SQLAuthenticate on
SQLAuthTypes Crypt Plaintext (et on enlève Plaintext)

SQLConnectInfo proftpd@localhost proftpd proftpd

SQLUserInfo ftpuser userid passwd uid gid homedir shell
SQLGroupInfo ftpgroup groupname gid members

CreateHome on
RequireValidShell off
Include /etc/proftpd/sql.conf

Nous passons maintenant à la création du groupe Debian qui accueillera les utilisateurs que nous allons créer dans la base de données :

groupadd –g 5500 ftpgroup
useradd –u 5500 –s /bin/false –d /bin/null –g ftpgroup ftpuser
gpasswd –a ftpuser ftpgroup

```
root@sebftp:/etc/proftpd# groupadd -g 5500 ftpgroup
root@sebftp:/etc/proftpd# useradd -u 5500 -s /bin/false -d /bin/null -g ftpgroup
 ftpuser
root@sebftp:/etc/proftpd# gpasswd -a ftpuser ftpgroup
Ajout de l'utilisateur ftpuser au groupe ftpgroup
```

Nous nous connectons à la base avec l'utilisateur ***usersftp*** afin d'insérer l'utilisateur « ***seb*** » dans le groupe ***ftpgroup*** :

mysql –u usersftp –p usersftp
insert into ftpgroup values ('ftpgroup',5500,'ftpuser') ;
select * from ftpgroup ;

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

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

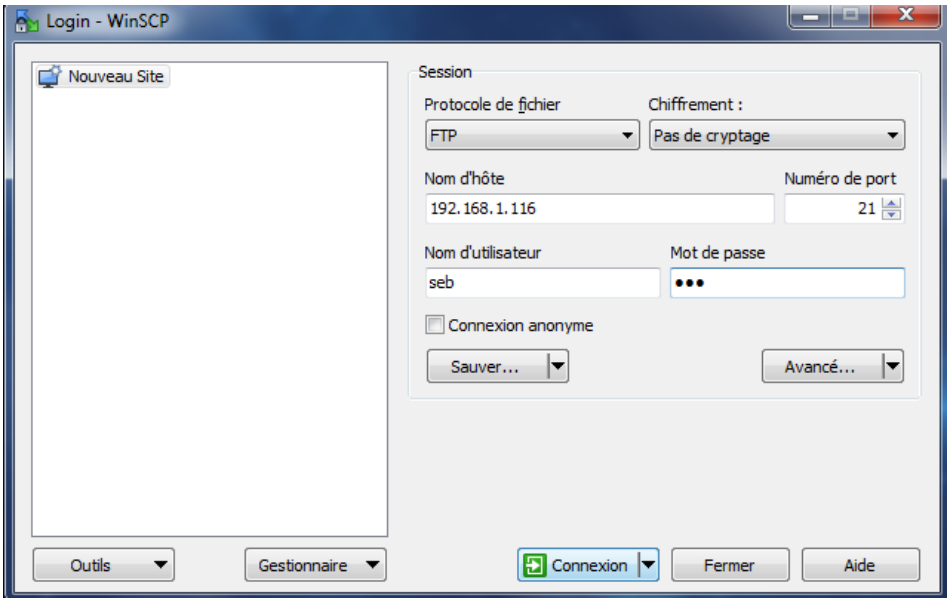
insert into ftpuser values (1,'seb',encrypt('seb'),5500,5500,'/home/seb','/sbin/nologin','','','','') ; select * from ftpuser ;

```
MariaDB [usersftp]> insert into ftpuser values (1,'seb',encrypt('seb'),5500,5500
,'/home/seb','/sbin/nologin','','','','');
Query OK, 1 row affected, 4 warnings (0.00 sec)

MariaDB [usersftp]> select * from ftpuser;
+----+--------+--------------+------+------+-----------+---------------+-------
+---------------------+---------------------+--------------+
| id | userid | passwd       | uid  | gid  | homedir   | shell         | count
| accessed            | modified            | LoginAllowed |
+----+--------+--------------+------+------+-----------+---------------+-------
+---------------------+---------------------+--------------+
|  1 | seb    | yJr49PvYB2YXk | 5500 | 5500 | /home/seb | /sbin/nologin |     0
| 0000-00-00 00:00:00 | 0000-00-00 00:00:00 |              |
+----+--------+--------------+------+------+-----------+---------------+-------
+---------------------+---------------------+--------------+
1 row in set (0.00 sec)
```

L'utilisateur ***seb*** est donc bien créé.

9 – Faire les tests. Conclure :

Nous essayons donc de nous connecter avec l'utilisateur ***seb*** dont le mot de passe est ***seb*** au serveur FTP :



Sur cet exemple, l'authentification ne fonctionne pas, car une ligne doit être manquante dans les fichiers de configuration. Sur les TP réalisés en cours, ce problème a été corrigé seulement après avoir installé ***KeepAlived***, qui a rendu l'authentification de notre utilisateur possible, sans modification de la configuration de ***ProFTP***. Avec notre configuration actuelle, sans ***KeepAlived***, il semble donc impossible, ou très difficile de faire fonctionner l'authentification d'un utilisateur présent dans une base de données.

10 – Installer l'interface graphique Adminer :

Adminer est une application web offrant une interface graphique pour le système de gestion de base de données ***MySQL***, réalisée en ***PHP*** et distribué sous licence ***Apache***. Il se présente comme une alternative légère à ***phpMyAdmin*** et a pour particularité d'être entièrement contenu dans un seul fichier ***PHP***. On peut toutefois ajouter un fichier ***CSS***, pour modifier la présentation ; il y en a de nombreux à télécharger gratuitement sur le site.

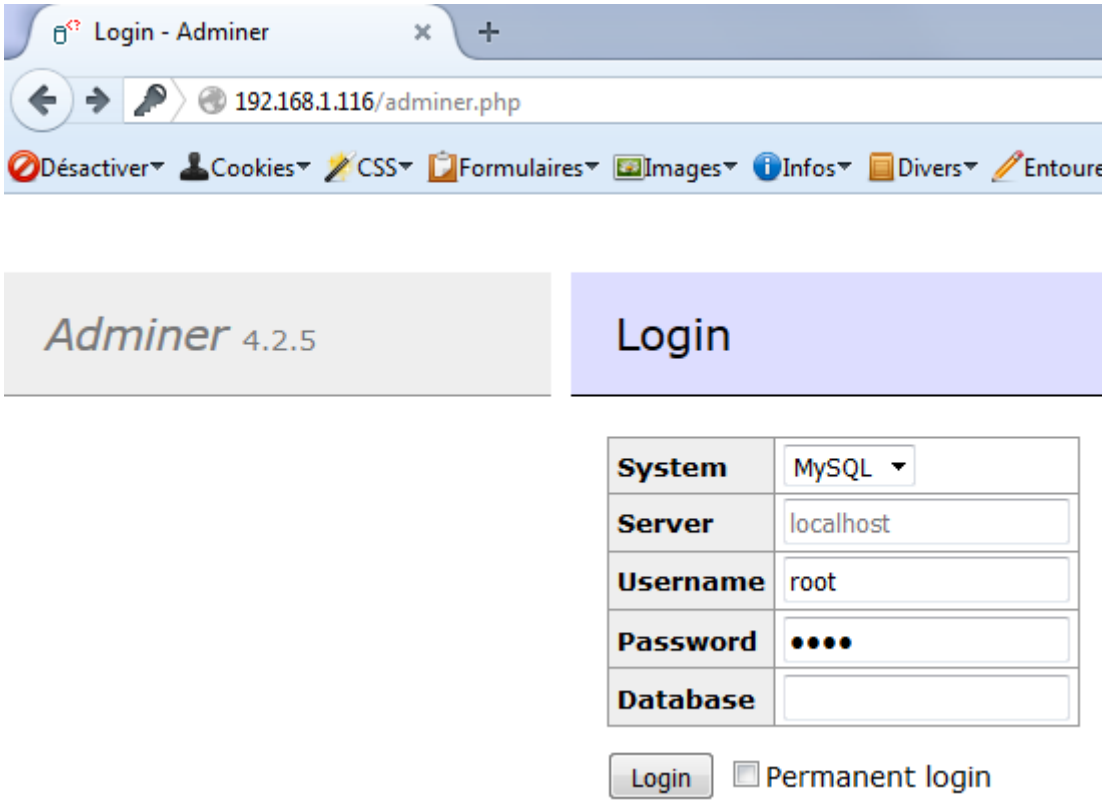
Nous allons donc installer ce logiciel en le téléchargeant directement sur le site officiel avec un :
wget https://www.adminer.org/static/download/4.2.5/adminer-4.2.5-mysql-en.php

```
root@sebftp:/etc/proftpd# wget https://www.adminer.org/static/download/4.2.5/adm
iner-4.2.5-mysql-en.php
--2016-10-18 11:32:47--  https://www.adminer.org/static/download/4.2.5/adminer-4
.2.5-mysql-en.php
Résolution de www.adminer.org (www.adminer.org)… 54.79.123.29
Connexion à www.adminer.org (www.adminer.org)|54.79.123.29|:443… connecté.
```

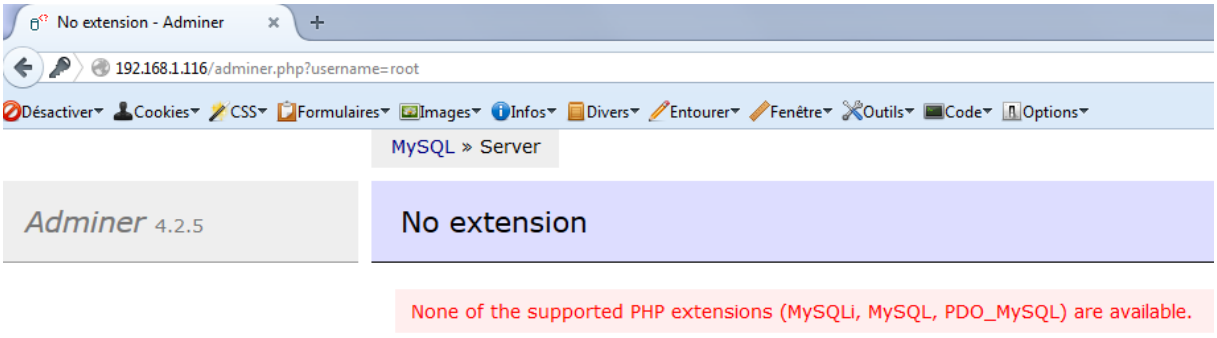
Nous déplaçons ensuite le fichier tout en le renommant dans le répertoire ***/var/www/html*** avec un ***mv adminer-4.2.5-mysql-en.php /var/www/html/adminer.php*** :

```
root@sebftp:~# mv adminer-4.2.5-mysql-en.php /var/www/html/adminer.php
root@sebftp:~# ls /var/www/html
adminer.php  index.html
```

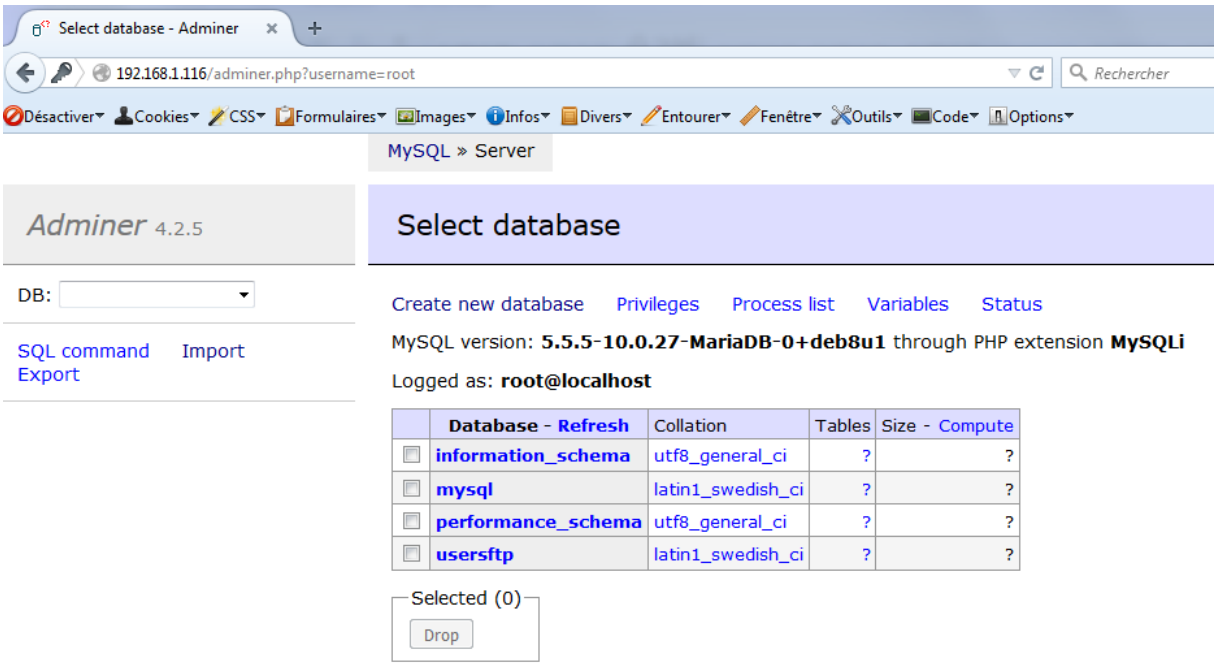
Lorsque c'est fait, nous pouvons accéder à ***http://192.168.1.116/adminer.php*** sur un navigateur :



Si une erreur de ce type apparaît :

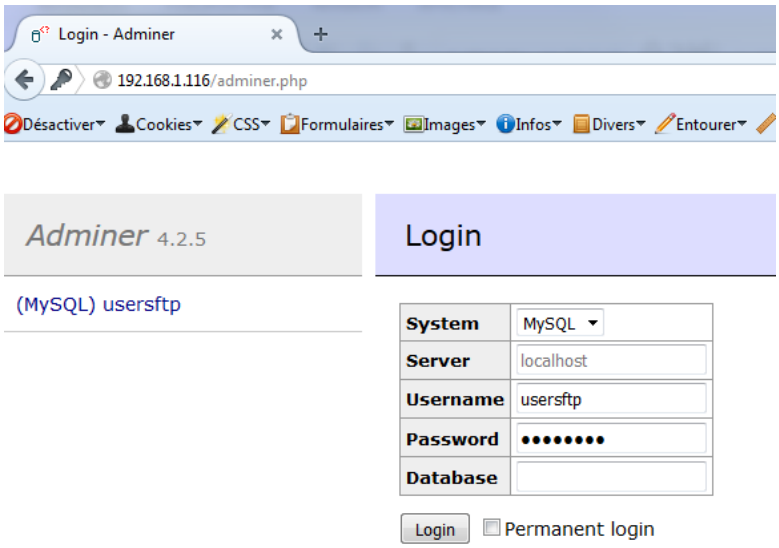


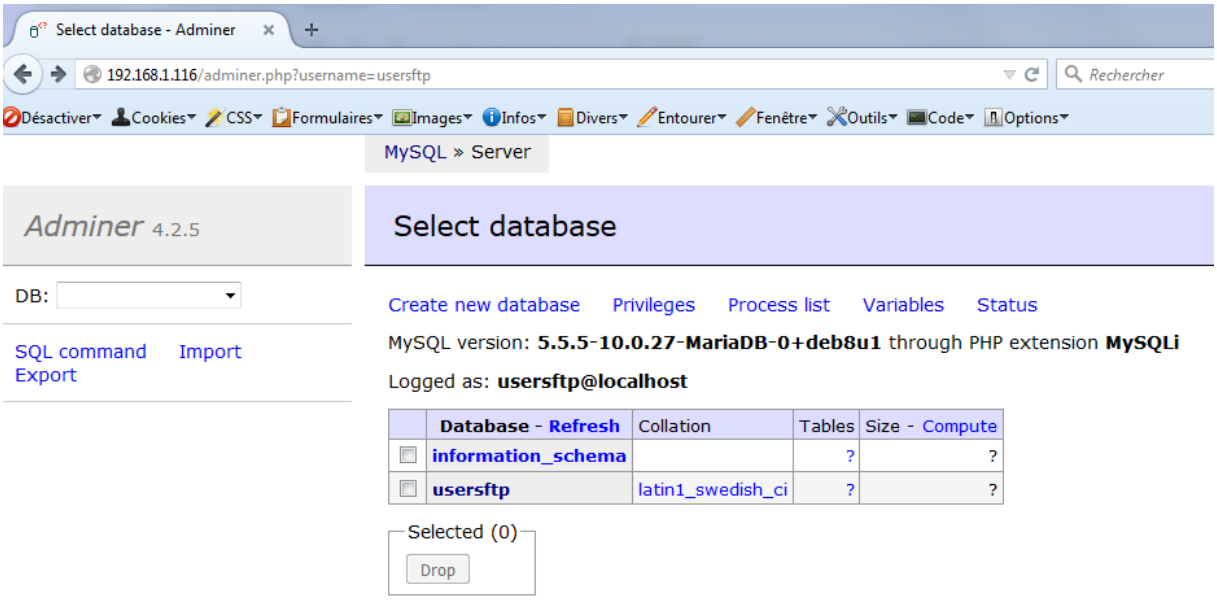
Il faut alors installer le paquet manquant avec un ***apt install php5-mysql***, et redémarrer le service ***Apache 2***. Une fois ceci fait, nous pouvons accéder à l'interface graphique :



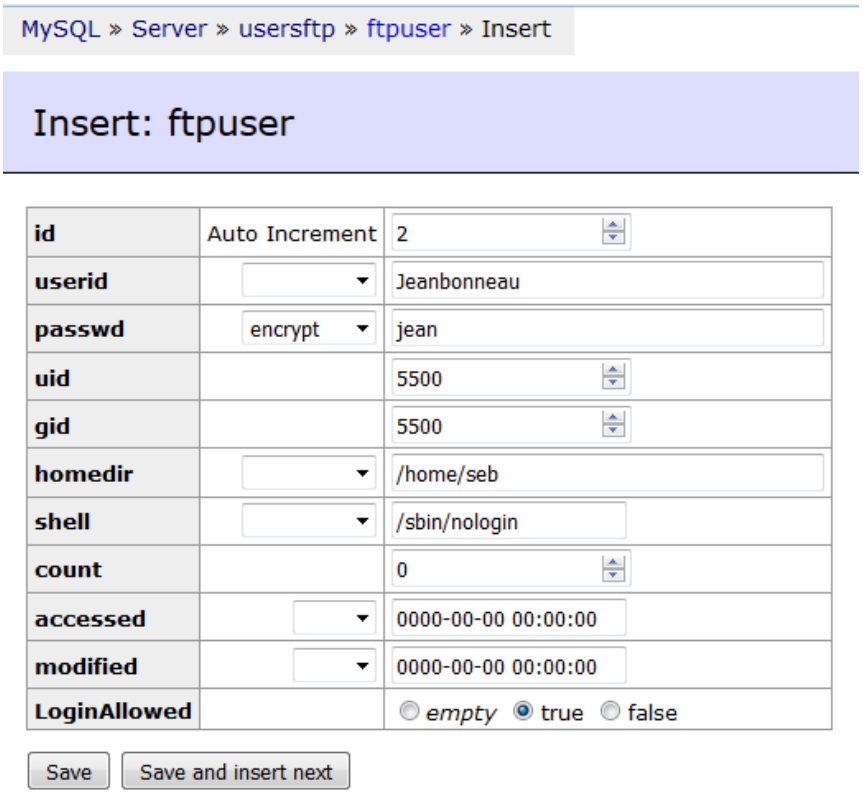
11 – Ajouter l'utilisateur Jean Bonneau :

Nous nous connectons avec l'utilisateur ***usersftp*** créé précédemment, et apte à insérer de nouveaux utilisateurs :





Pour ajouter un utilisateur, il faut ensuite suivre le chemin ***usersftp > ftpuser > New item***. Nous pouvons ensuite entrer les informations requises :



Et cliquer sur « ***Save and insert next*** » :

L'utilisateur a donc bien été créé.

12 – Réaliser une sauvegarde de la base usersftp avec ses données :

Nous devons maintenant en faire une sauvegarde non compressée. Pour cela, nous utilisons la commande ***mysqldump –u root –p MaBase > MaSauvegarde.sql***

```
root@sebftp:~# mysqldump -u root -p usersftp > usersftp_dump.sql
Enter password:
root@sebftp:~# ls
proftpd_mysql.txt  usersftp_dump.sql
```

Nous pouvons vérifier son contenu avec un ***cat usersftp_dump.sql*** :

```
--
-- Dumping data for table `ftpuser`
--

LOCK TABLES `ftpuser` WRITE;
/*!40000 ALTER TABLE `ftpuser` DISABLE KEYS */;
INSERT INTO `ftpuser` VALUES (1,'seb','yJr49PvYB2YXk',5500,5500,'/home/seb','/sb
in/nologin',0,'0000-00-00 00:00:00','0000-00-00 00:00:00',''),(2,'Jeanbonneau','
npJdyohQXs35w',5500,5500,'/home/seb','/sbin/nologin',0,'0000-00-00 00:00:00','00
00-00-00 00:00:00','true');
/*!40000 ALTER TABLE `ftpuser` ENABLE KEYS */;
UNLOCK TABLES;
/*!40103 SET TIME_ZONE=@OLD_TIME_ZONE */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
/*!40014 SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;

-- Dump completed on 2016-10-18 11:48:01
```

La sauvegarde a donc bien eu lieu !