L'auto-hébergement facile

Créez l'internet que vous voulez

Xavier Cartron – 10 octobre 2015–

Table des matières

1 À propos...

Ce document est libre, mais pas gratuit. Si vous l'avez apprécié, n'hésitez pas à donner en échange un petit quelque chose :

— Avec flattr en cliquant sur ce bouton 

ou en allant à l'adresse suivante : https://flattr.com/submit/auto?user_id=Thuban&url=http://yeuxdelibad.net/DL/PDF/auto-h-facile.pdf
— Avec un petit merci
— De la façon qui vous plaît le mieux. Soyez originaux ☺

Vous pouvez toujours contacter l'auteur :
http://yeuxdelibad.net/Divers/Contact.html
Les sources sont disponibles sur le dépôt git suivant :
http://git.yeuxdelibad.net/auto-h-facile

2 L'auto-hébergement

2.1 C'est quoi l'auto-hébergement ?

La plupart des sites web que vous avez l'habitude de consulter, vos courriels, les réseaux sociaux, sont hébergés, quelque part dans le monde, sur des serveurs. C'est quoi un serveur ? C'est ni plus ni moins un ordinateur.

Le principe est simple : votre navigateur ne fait qu'échanger des données avec ces serveurs pour que vous puissiez utiliser les services qu'ils proposent.

+ SIMPLEMENT

Lorsque vous voulez consulter vos e-mails, votre navigateur va chercher sur un serveur quelque part dans le monde tous vos messages, qui sont alors téléchargés vers votre ordinateur.
C'est comme si pour lire votre courrier postal, vous deviez aller à la poste, (*attendre dans la file d'attente*), demander au facteur :
T'as du courrier pour moi ?
(*Attendre qu'il aille chercher les colis*) , lire votre courrier. C'est le bureau de poste qui l'avait, jusqu'à ce que vous le releviez.

2.2 Avantages de l'auto-hébergement

Héberger chez soi les services que l'on utilise, ou *s'auto-héberger* présente plusieurs avantages :

— Les données restent chez vous. Cela veut dire que vous gardez le contrôle de vos fichiers. C'est particulièrement intéressant si vous aviez l'habitude de partager des documents à l'aide de service tiers (les photos chez picasa, les vidéos sur youtube, sans parler de mega...). Ces données restent donc chez vous et ne sont pas sur un lointain serveur qui peut en faire on ne sait quoi.
— Votre vie privée est respectée Par exemple, vos courriels ne seront pas scannés pour revendre ensuite à des tiers vos préférences personnelles.
— Vous pouvez avoir à portée de main des services qui répondent exactement à vos besoins.
— Vous pouvez utiliser du matériel à faible consommation électrique et faire ainsi attention à la planète.
— S'auto-héberger, c'est amusant et instructif. Cela permet de mieux comprendre le fonctionnement d'internet.

2.3 Inconvénients

Oui, il y a aussi quelques inconvénients :
— Cela peut demander du temps.
— La bande passante est limitée. Les performances seront donc inférieures à celle d'un serveur grand public.

+ SIMPLEMENT

Imaginez un tuyau. Lorsque vous ouvrez le robinet, vous aurez beau le dévisser, il y a un moment où le débit d'eau n'augmente plus. La bande passante est la quantité maximale de données que votre connexion peut envoyer/recevoir par seconde, ou si vous préférez, le diamètre du tuyau.

— C'est vous qui vous chargez de la sécurité. Cela demande donc du soin.

3 Pré-requis

Vous êtes décidés à tenter l'aventure ? Alors c'est parti. Voyons voir de quoi nous allons avoir besoin :

3.1 Le matériel

Du point de vue matériel, plusieurs choix s'offrent à vous.

— Récupérer de l'ancien matériel. Par exemple, le vieil ordinateur qui traîne dans un coin. Ou bien le portable à l'écran cassé. C'est le moins onéreux, et c'est écologique.
— Acheter une carte comme le raspberry pi qui est bon marché et consomme très peu.
— Acheter les pièces pour concevoir un serveur basse consommation. Là, c'est vraiment si vous avez déjà de bonnes connaissances.

3.2 Le système d'exploitation

Vous êtes libres d'installer le système d'exploitation qui vous convient le mieux.

Dans la suite du document, je supposerai que vous utilisez debian, une distribution GNU/Linux que je vous conseille.

Pourquoi ?

Debian est stable, simple à mettre à jour, facile à installer. Niveau sécurité, en cas de faille découverte dans un logiciel, les mises à jour avec correctifs sont très rapidement disponibles. Enfin, en plus des pages de manuel habituelles, de la documentation et des exemples sont le plus souvent fournis dans le répertoire `/usr/share/doc/NOM-DU-LOGICIEL`.

3.3 Un nom de domaine

Vous voudrez certainement obtenir un nom de domaine, qui permettra à tous de retrouver simplement votre serveur. Cela vous permet aussi de mieux vous organiser avec des sous domaine, par exemple `mail.mondomaine.com`, `blog.mondomaine.com`...

+ SIMPLEMENT

M. Ali Gator vit au 5 rue du moulin à Picsouville. Pour aller lui rendre visite, c'est à cette adresse que vous allez vous rendre.
Sur l'internet, l'adresse de votre serveur, c'est une série de nombres. Par exemple *93.22.160.7*. C'est pratique pour les machines, pas pour les humains.
Un nom de domaine permet aux humains d'utiliser l'adresse *wikipedia.org*, qui est traduit par les ordinateurs en *91.198.174.192*.
Avouez que c'est plus facile à retenir.

Allons plus loin

Les séries de nombres indiquant “l’adresse” d’un serveur est ce qu’on appelle une adresse ip.
L’association d’une ip avec un nom de domaine s’appelle *DNS* (Domain Name System)

3.3.1 Achat d’un nom de domaine

Vous pouvez acheter un nom de domaine auprès de ce que l’on appelle un *registre* ou *registrar*. En voici quelques exemples :

— OVH
— Gandi
— bookmyname

Vous allez simplement pouvoir vérifier que le nom de domaine que vous souhaitez acheter est disponible (choisissez-en un qui vous correspond). Si oui, alors vous passez la commande. C’est très rapide, et la plupart du temps, votre adresse ip est déjà par défaut associée à votre nom de domaine.

3.3.2 Nom de domaine gratuit

Certains registres proposent des noms de domaine, avec moins de flexibilité pour vous cependant. Cela peut être intéressant le temps de se faire la main. Quelques exemples :

— FDN
— EU.org
— FreeDNS

3.3.3 Relier mon nom de domaine à mon adresse ip

Une fois que vous avez un nom de domaine, il faut le relier à l’adresse ip de votre serveur. Mais si, souvenez vous, cette série de nombres ressemblant à *91.198.174.192*.

Pour cela, enregistrez un champ de type **A** dans l’interface d’administration du registre. Par exemple :

```
mondomaine.com    A    34.121.124.123
```

Comment connaître mon adresse ip me direz-vous ? Rien de plus simple, il existe de nombreux services qui vous permettent de la retrouver. Quelques exemples :

— lehollandaisvolant.net/tout/ip

— http://who.is/
— http://www.whatsmyip.org/

Allons plus loin

Il existe plusieurs type d'enregistrement :
— Les champs **A** pour les adresses IPv4. ex : `12.123.123.12`
— Les champs **AAAA** pour les adresses IPv6. ex : `2001:db8:4212:4212:4212:4212:4212:4212`
— Les champs **MX**, **NS** et **CNAME**. Ces derniers seront particulièrement utiles pour définir des sous-domaines.
En effet, vous pouvez faire pointer un sous-domaines vers le domaine principal avec un CNAME au lien d'enregistrer un nouveau champ A. Exemple :
`blog.mondomaine.com CNAME mondomaine.com`
Cela permet d'organiser plusieurs sites sur son serveur proprement.

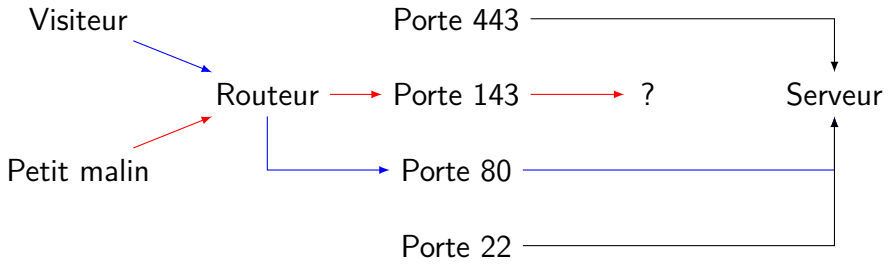
3.4 Apprenez à rediriger les ports de votre *box

Votre serveur, tout comme votre ordinateur actuellement, sera certainement connecté à internet par l'intermédiaire d'un modem (une *box).

Il faut s'assurer que lorsqu'un visiteur voudra accéder à votre serveur, la *box le redirige bien vers votre serveur, et non vers une autre machine du réseau local. On dit que l'on *configure le routeur*.

+ SIMPLEMENT

Imaginez votre *box comme un grand mur avec dedans plusieurs portes. Chaque porte est numérotée.
Quand quelqu'un veut accéder à votre serveur, il va venir frapper à l'une des portes, par exemple la numéro 80 pour un serveur http. Afin que tout fonctionne bien, il est nécessaire de savoir où mène la porte n° 80. Si la box ne le sait pas, alors la porte 80 reste fermée et votre serveur est inaccessible.
Bien sûr, pour plus de sécurité encore, une fois la porte 80 passée, votre serveur sera équipé d'un parefeu pour vérifier que vous avez bien le droit d'entrer.



Dans le schéma ci-dessus, seuls les ports 443, 80 et 22 sont associées au serveur.
Si le petit malin demande un port qui n'est pas associé au serveur (la porte est fermée), alors la requête ne peut pas aller jusqu'au bout. C'est comme si ce malin demandait à aller à une destination qui n'existe pas. En revanche, lorsque le visiteur demande à passer par la porte 80, il est bien renvoyé vers le serveur.

Cela se configure toujours de la même façon :

1. Vous accédez à l'interface de configuration du modem
2. Vous préciser le port d'écoute par lequel vont arriver les requêtes. Par exemple, le port 80 pour un site web.
3. Vous indiquez que ce qui est adressé à ce port doit être mis en relation avec le port 80 de votre serveur (et pas un autre ordinateur connecté à la *box).

Cependant, l'interface de configuration n'est pas la même selon si vous avez une livebox, freebox, modem OVH... Pas d'inquiétude, on peut trouver l'adresse à taper dans un navigateur web pour accéder à cette interface. Essayez dans l'ordre suivant :

— `192.168.0.1`
— `192.168.1.1`

— `192.168.1.254`
— Pour une freebox, cela se passe sur la page de gestion de votre compte

Bien sûr, cette "adresse" est à utiliser sur un ordinateur lui-même connecté à la *box.

Il est possible qu'un nom d'utilisateur et un mot de passe soient demandés. Essayez dans ce cas `admin/admin`, sinon, demandez directement à votre fournisseur d'accès à internet. Une fois connecté, allez dans la section "Configurer mon routeur".

Pour plus d'informations, cette page peut vous être utile : https://craym.eu/tutoriels/utilitaires/ouvrir_les_ports_de_sa_box.html.

4 Installation de services

Avant d'aller plus loin, je dois préciser quelques conventions utilisées par la suite.

— Certaines commandes seront précédées d'un symbole `$`. Cela signifie qu'elle ne nécessitent pas les privilèges superutilisateur.
— Les autres seront précédées par un symbole `#`. Ces dernières nécessitent d'avoir les droits d'administrateur pour les lancer. Pour devenir superutilsateur, il faut taper la commande `su`, puis votre mot de passe (c'est normal s'il ne s'affiche pas). On dit alors que vous êtes "*root*".

+ SIMPLEMENT

Afin de faciliter l'installation de services, vous voudrez peut-être utiliser **hostathome**. C'est un script qui se chargera de réaliser les étapes ci-dessous pour vous. Jetez un œil au 9 pour en savoir plus.

4.1 Préparer l'accès ssh

SSH, ou Secure Shell , vous permettra principalement d'accéder à votre serveur à distance de façon sécurisée, et ainsi de l'administrer.

Pour l'installer, il suffit d'ajouter le paquet `openssh-server` sur le serveur. Une fois ceci fait, pensez à faire deux choses :

1. Ouvrir le port 22 du parefeu du serveur s'il y en a un
2. Rediriger le port 22 de votre routeur vers le serveur

Pour utiliser ssh et se connecter, il faut le paquet `openssh-client`. La syntaxe pour se connecter au serveur est la suivante.

```
ssh -p PORT UTILISATEUR@SERVEUR
```

Quelques explications :
— PORT : il faut préciser le port sur lequel le serveur ssh écoute. Par défaut, c'est le 22, mais nous verrons ensuite comment le modifier pour plus de sécurité.
— UTILISATEUR : sur le serveur, il y a des utilisateurs. Il faut simplement préciser le login de l'utilisateur avec lequel on souhaite se connecter.
— SERVEUR : c'est soit l'adresse ip du serveur, soit le nom de domaine.

4.1.1 Configuration minimale pour la sécurité

La configuration du serveur ssh se déroule dans le fichier `/etc/ssh/sshd_config`. Veillez à préciser les quelques lignes suivantes, en précisant les noms des utilisateurs autorisés à se connecter :

```
Port 2221
PermitRootLogin no
X11Forwarding no
AllowUsers utilisateur_autorisé_à_se_connecter
```

Bien sûr, pour le port, mettez la valeur que vous souhaitez.

4.1.2 Connexion automatique à l'aide de clés

On peut préférer se connecter au serveur à l'aide d'une clé afin de ne pas avoir à entrer de mot de passe. Voici rapidement la marche à suivre.

"Client" désigne la machine qui veut se connecter au "serveur".

1. On crée d'une paire de clefs sur le client : `ssh-keygen -t dsa -f ~/.ssh/labas`
2. Configuration de ssh sur le client : éditer ou créer le fichier `~/.ssh/config` et y ajouter :

   ```
   Host labas
   HostName nomtreslong.lichtenstein.loin
   User bouvardetpecuchet.flaubert
   PasswordAuthentication no
   IdentityFile ~/.ssh/labas
   ```

   Il faudra bien sûr adapter "HostName" par le nom de domaine du serveur, "User" par le nom d'utilisateur se connectant au serveur.

3. Configuration du serveur : sur le client, exécutez ceci (remplacez “xxx” par le port utilisé par le serveur (22 par défaut))

```
ssh-copy-id -i ~/.ssh/labas.pub -p xxx\
 bouvardetpecuchet.flaubert@nomtreslong.lichtenstein.loin
```

4. Finalement, vous pouvez vous connecter simplement en tapant `ssh labas`

4.2 Vos sites web – serveur http

Un site web est le plus souvent un ensemble de pages .html ou .php. Ce contenu est fourni aux visiteurs de votre site par un “serveur http”.

Le plus connu des serveurs http est certainement apache. *Ce n'est pas celui dont on va parler ici.*

En effet, on va lui préférer nginx qui présente des avantages non-négligeables. Il est beaucoup plus léger, et s'adapte donc mieux à des serveurs aux configurations modestes, le cas le plus fréquent lorsqu'on s'auto-héberge.

4.2.1 Installation de nginx

Comme la plupart du temps sur une debian, l'installation d'un service est simplissime :

```
# apt-get install nginx
```

Et voilà ! ☺

Bon, il y a quand même quelques petites choses à savoir :

— Le fichier de configuration global se trouve dans `/etc/nginx/nginx.conf`,

— Le dossier `/etc/nginx/conf.d`, dans lequel vous mettrez des fichiers se terminant par “.conf” afin de configurer chacun de vos sites.

Exemple de site : Voici un fichier `/etc/nginx/conf.d/exemple.conf` qui montre un exemple de ce à quoi ressemble la configuration d'un site. Remarquez que les instructions sont dans une section `server`.

```
server {
listen 80;
server_name monsite.fr;
index index.html;
root /media/www/monsite;
access_log /var/log/nginx/monsite.log;
```

```
    location ~*  \.(jpg|jpeg|png|gif|ico|svg|mp4|ogg|ogv|webm|css|js)$ {
    expires 1M;
    }
}
```

Dans ce cas, toutes les pages html déposées dans /media/www/monsite seront disponibles à l'adresse monsite.fr .

Pour utiliser php , il sera nécessaire d'ajouter quelques lignes au fichier ci-dessus :

```
server {
listen 80;
server_name monsite.fr;
index index.html;
root /media/www/monsite;
access_log /var/log/nginx/monsite.log;

    location ~*  \.(jpg|jpeg|png|gif|ico|svg|mp4|ogg|ogv|webm|css|js)$ {
    expires 1M;
    }

    location ~ \.php$ {
        try_files $uri = 404;
        fastcgi_pass unix:/var/run/php5-fpm.sock;
        include fastcgi_params;
        fastcgi_intercept_errors on;
        fastcgi_param HTTPS on;
        fastcgi_param   SCRIPT_FILENAME  $document_root$fastcgi_script_name;
    }
}
```

Et bien sûr, il faudra installer php :

```
    # apt-get install php5-fpm php5-acpu
```

Allons plus loin

"Tu nous balances pleins de lignes bizarres, mais moi, j'ai envie de savoir ce que ça veut dire !"
Voyons rapidement ce que signifient les lignes dans ces fichiers de configuration :

- `listen 80;` : le site est accessible sur le port 80
- `server_name` : le domaine du site est monsite.fr . Ça permet à nginx de donner le bon site si, sur le même serveur, on a plusieurs sous-domaines. Par exemple, un site en blog.monsite.fr, un autre en webmail.monsite.fr ...
- `index index.html;` : Si le visiteur ne précise pas de page html , on cherche le fichier index.html. Exemple : monsite.fr/ est équivalent à monsite.fr/index.html
- `root /media/www/monsite;` : L'emplacement où se trouvent les fichiers du site sur le serveur
- `access_log` : L'emplacement des logs d'accès au site
- `location ...` : Cette instruction permet de réaliser certaines actions selon ce que le visiteur demande au serveur. Ici, on augmente le cache pour les fichiers de type image. Plus bas, on traite les fichiers en php de façon à ce qu'ils soient rendus par le serveur. Pour plus de détails sur ce sujet, il est conseillé de visiter le site de nginx

4.3 Courrier électronique

Cette partie explique l'installation d'un serveur mail le plus simple possible. Aucune base de donnée ne sera utilisée, car ça n'a aucune intérêt en auto-hébergement.

L'installation va se dérouler en 4 étapes :

1. Configuration d'enregistrement DNS particuliers (voir 3.3) et préparation du serveur,
2. Configuration de postfix, qui se chargera d'envoyer et recevoir votre courrier
3. Configuration de dovecot, qui permet la réception du courrier avec un client comme thunderbird
4. Faire le nécessaire pour que vos messages ne soient pas considérés comme spam.

On verra ensuite comment ajouter de nouveaux comptes mail sur votre serveur, et comment configurer votre client de messagerie (thunderbird) pour l'utiliser.

4.3.1 Configuration des DNS pour le courrier électronique et préparation du serveur

[1] Chez votre registrar, ajoutez deux nouveaux champs :
— Un champ de type A qui pointe vers votre ip :
`mail.votredomaine.net A 109.190.193.182`
— Un champ de type MX qui pointe vers le A précédent
`votredomaine.net. MX 1 mail.votredomaine.net.`

Sur votre serveur, modifiez le fichier `/etc/hosts` pour rajouter une ligne de ce type :

```
127.0.0.1       mail.votredomaine.net
```

Enfin, ouvrez et redirigez les ports suivants : TCP 25, 143 et 993 (voir 3.4).

On peut maintenant passer à l'installation du serveur mail

4.3.2 Postfix

Hop, on installe postfix à la manière debian, ainsi que postgrey qui permettra d'éviter de nombreux spams :

```
# apt-get install postfix postgrey
```

Quelques questions vont vous être posées. Choisissez la configuration de type "Site Internet". Le reste est assez compréhensible. Vous pouvez relancer le questionnaire avec la commande

```
# dpkg-reconfigure postfix
```

Pour configurer postfix, on va mettre le contenu suivant dans le fichier `/etc/postfix/main.cf`. Attention, c'est long, mais pas de panique ☺

```
# See /usr/share/postfix/main.cf.dist for a commented,
# more complete version

# Debian specific:  Specifying a file name will cause the first
# line of that file to be used as the name.  The Debian default
# is /etc/mailname.
#myorigin = /det/mailname
```

1. Pour quelques explications sur les DNS, voir le §3.3

```
smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
biff = no

# appending .domain is the MUA's job.
append_dot_mydomain = no

# Uncomment the next line to generate "delayed mail" warnings
#delay_warning_time = 4h

readme_directory = no

# TLS parameters
smtp_use_tls=yes
smtpd_use_tls = yes
smtp_tls_note_starttls_offer = yes
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_tls_session_cache_database = btree:/smtpd_scache
smtp_tls_session_cache_database = btree:/smtp_scache

# See /usr/share/doc/postfix/TLS_README.gz in the postfix-doc package for
# information on enabling SSL in the smtp client.

mydomain = votreserveur.net
myhostname = votreserveur.net
myorigin = $mydomain
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
mydestination = $myhostname, $mydomain, localhost.$mydomain,\
    localhost, localhost.$myhostname
relayhost =
recipient_delimiter = +
inet_interfaces = all
home_mailbox = Maildir/
### Limit the info given to outside servers
show_user_unknown_table_name = no

# Basics Restrictions
smtpd_helo_required = yes
strict_rfc821_envelopes = yes
```

```
smtpd_sasl_auth_enable = yes

# Utiliser le service d'identification de Dovecot
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
# Noter dans les en-tête des messages l'identifiant de l'utilisateur.
smtpd_sasl_authenticated_header = yes

smtpd_sasl_local_domain = votreserveur.net
smtpd_sasl_security_options = noanonymous


# DKIM
milter_default_action = accept
milter_protocol = 2
smtpd_milters = inet:localhost:8891
non_smtpd_milters = inet:localhost:8891

## ANTISPAM
# Wait until the RCPT TO command before evaluating restrictions
smtpd_delay_reject = yes
### Tarpit those bots/clients/spammers who send errors or scan for accounts
smtpd_error_sleep_time = 20
smtpd_soft_error_limit = 1
smtpd_hard_error_limit = 3
smtpd_junk_command_limit = 2

smtpd_helo_required = yes
strict_rfc821_envelopes = yes

# Allow connections from trusted networks only.
smtpd_client_restrictions = permit_mynetworks,
    permit_sasl_authenticated,
    reject_rbl_client bl.spamcop.net,
    reject_rbl_client cbl.abuseat.org,
    reject_rbl_client sbl-xbl.spamhaus.org,
    reject_rbl_client list.dsbl.org,
    permit

# Don't talk to mail systems that don't know their own hostname.
# With Postfix < 2.3, specify reject_unknown_hostname.
```

```
    smtpd_helo_restrictions = reject_unknown_helo_hostname

    # Don't accept mail from domains that don't exist.
    smtpd_sender_restrictions = reject_unknown_sender_domain,
        permit_mynetworks,
        permit_sasl_authenticated,
        reject_non_fqdn_sender

    # Relay control (Postfix 2.10 and later): local clients and
    # authenticated clients may specify any destination domain.
    smtpd_relay_restrictions = permit_mynetworks,
        permit_sasl_authenticated,
        reject_unauth_destination

    # Spam control: exclude local clients and authenticated clients
    # from DNSBL lookups.
    smtpd_recipient_restrictions = permit_mynetworks,
        permit_sasl_authenticated,
        check_policy_service inet:127.0.0.1:10023,
        reject_rbl_client zen.spamhaus.org,
        reject_rhsbl_reverse_client dbl.spamhaus.org,
        reject_rhsbl_helo dbl.spamhaus.org,
        reject_rhsbl_sender dbl.spamhaus.org

    # Block clients that speak too early.
    smtpd_data_restrictions = reject_unauth_pipelining
```

Vous n'avez quasiment rien à modifier dans ce fichier, mis à part remplacer `votreserveur.net` par votre nom de domaine.

Pour terminer, on va faire en sorte de pouvoir utiliser le port 587 pour envoyer des courriels, car le port 25 est souvent bloqué par les fournisseurs d'accès à internet.

Décommentez les lignes suivantes (retirez le `#`) dans le fichier `/etc/postfix/master.cf` :

```
    submission inet n       -       -       -       -       smtpd
    -o syslog_name=postfix/submission
    -o smtpd_tls_security_level=encrypt
    -o smtpd_sasl_auth_enable=yes
    -o smtpd_client_restrictions=permit_sasl_authenticated,reject
    -o milter_macro_daemon_name=ORIGINATING
```

Voilà pour postfix.

Allons plus loin

On a installé postgrey, qui permet d'éviter de recevoir la majorité des spams. Son principe de fonctionnement est très simple : la première fois qu'un serveur tente de vous envoyer un mail, il est mis en attente.
S'il s'agit d'un spam, cela s'arrête là, et vous ne le recevez pas.
Si c'est un serveur légitime, il va retenter d'envoyer le même message quelques minutes plus tard. Dans ce cas, postgrey ajoute ce serveur à la liste des autorisés, et vous recevrez les messages en provenance de ce serveur sans délai par la suite.

4.3.3 Dovecot

Dovecot va être utilisé comme serveur imap. Pour pouvoir s'identifier de façon sécurisée, installons tout d'abord sasl :

```
# apt-get install libsasl2-2 sasl2-bin
```

Modifiez le fichier `/etc/default/saslauthd` pour y mettre :

```
START=yes
```

On peut maintenant installer dovecot :

```
# apt-get install dovecot-imapd
```

Ajoutez maintenant ces lignes à la fin du fichier `/etc/dovecot/dovecot.conf`

```
mail_location = maildir:~/Maildir

protocols = imap

disable_plaintext_auth = yes
ssl = yes

ssl_cert = </etc/ssl/certs/ssl-cert-snakeoil.pem
ssl_key = </etc/ssl/private/ssl-cert-snakeoil.key

service auth {
    unix_listener /var/spool/postfix/private/auth {
        mode = 0660
        # Assuming the default Postfix user and group
```

```
            user = postfix
            group = postfix
        }
    }
```

Pour terminer cette partie, on peut relancer les services qui sont maintenant configurés :

```
# service saslauthd restart
# service postfix restart
# service dovecot restart
```

4.3.4 Ne pas être mis dans les spams

Plusieurs dispositifs permettent de montrer que les mails sortant de votre serveur ne sont pas des spams. Prenons-les un par un.

SPF : Ajoutez un champ DNS [2] auprès de votre registrar de type SPF tel que celui-ci :
`votredomaine.net. SPF ‘‘v=spf1 a mx ~all’’`
ou bien sous forme de champ TXT :
`votredomaine.net. TXT ‘‘v=spf1 a mx ~all’’`

DKIM : On installe opendkim comme d'habitude :

```
# apt-get install opendkim opendkim-tools
```

Ensuite, il faut créer une clé : Nous allons les créer dans un répertoire fait exprès pour. Par exemple, nous allons créer /etc/dkim :

```
# mkdir -p /etc/dkim
```

Générons les clés dans ce répertoire :

```
# opendkim-genkey -D /etc/dkim/ -d votreDomaine.net -s mail
```

Vous pouvez voir dans ce répertoire 2 fichiers : mail.private et mail.txt
On rend l'utilisateur "opendkim" propriétaire de ce dossier :

```
# chown opendkim:opendkim -R /etc/dkim
```

2. voir 3.3

Il faut maintenant éditer le fichier /etc/opendkim.conf. Pour un domaine unique ajoutez à la fin du fichier `/etc/opendkim.conf` les lignes suivantes :

```
UserID opendkim:opendkim
Domain      votreDomaine.net
KeyFile     /etc/dkim/mail.private
Selector        mail
AutoRestart     yes
DNSTimeout              5
```

Il faut que postfix et opendkim puissent communiquer. On ajoute alors dans /etc/default/opendkim ceci :

```
SOCKET="inet:8891:localhost"
```

Il n'y a rien d'autre à modifier pour postfix, car la configuration du paragraphe 4.3.2 est déjà prête.

Finalement, il faut ajouter un nouveau champ dans vos DNS [3] auprès de votre registrar. Eh oui, encore !

Il s'agira d'un champ DKIM ou TXT selon ce qui est disponible. Remplissez-le ainsi :

— Nom de domaine : `mail._domainkey.votredomaine.net`
— Contenu : `v=DKIM; k=rsa; p=.....` Recopiez en fait dans le contenu ce qui est dans le fichier `/etc/dkim/mail.txt`

Pour vérifier que vous n'êtes pas en spam, suivez les indications de ce site https://www.mail-tester.com/

4.3.5 Ajouter un nouveau compte mail

Un compte mail est en fait un simple compte d'utilisateur. Vous pouvez donc en créer un nouveau ainsi :

```
# adduser --shell /bin/false nouvel_utilisateur
```

Et voilà ! ☺

Allons plus loin

Notez que l'on précise le shell qui sera celui du nouvel utilisateur. Ce n'est pas n'importe lequel, puisqu'il s'agit de `/bin/false`. Cela permet de s'assurer que l'utilisateur ne pourra lancer aucune commande sur le serveur.
C'est un plus pour la sécurité.

3. voir 3.3

4.3.6 Configurer son client de messagerie

Pour consulter vos mails sur le serveur, vous pouvez utiliser un client de messagerie comme l'excellent thunderbird, logiciel-libre respectueux de votre vie privée.

Voici les paramètres qu'il faudra indiquer au logiciel pour envoyer et recevoir des courriels. Notez que tous ne vous seront peut-être pas demandés :

- Nom du serveur : mail.votredomaine.net
- Adresse électronique : votre_identifiant@votredomaine.net
- Nom d'utilisateur : l'identifiant choisi à la création d'un compte mail (voir 4.3.5)
- Serveur entrant : IMAP
  - port : 143
  - SSL : STARTTLS
- Serveur sortant : SMTP
  - port : 587
  - SSL : STARTTLS

Vous souhaiterez peut-être plutôt utiliser un webmail, afin d'accéder à votre messagerie à partir d'un navigateur web. Cela est expliqué au paragraphe suivant : 5.4

4.4 Serveur sftp

Sftp, ou "Secure File Transfert Protocol" est une variante du protocole FTP nettement plus sécurisée puisqu'elle passe par un tunnel ssh.

Suite à cette installation, vous pourrez partager des fichiers avec ceux que vous voulez, en toute sécurité. Eh oui, c'est ssh qui s'en charge ! Il sera possible de déposer des fichiers sur le serveur, et d'en télécharger.

Dans la suite, je suppose que les fichiers déposés sur le serveur sont stockés dans `/media/sftp`

4.4.1 Configuration du serveur sftp

Un serveur sftp est disponible lorsqu'on installe le paquet `openssh-server`. Donc si vous avez déjà un accès ssh sur votr serveur, c'est déjà fait.

Il faut toutefois modifier la configuration de ssh. Souvenez-vous, c'est dans le fichier `/etc/ssh/sshd_config` :

```
Match Group sftpusers
ChrootDirectory /media/sftp
ForceCommand internal-sftp -f LOCAL7 -l INFO
AllowTcpForwarding no
```

+ SIMPLEMENT

Tous les utilisateurs du groupe sftpusers seront coincés dans le dossier `/media/sftp`.
Utiliser un groupe sftpusers exprès pour l'accès en sftp permet de facilement rajouter des utilisateurs, tout en gardant le tout sécurisé
On appelle cela faire un chroot

Le dossier de stockage `/media/sftp` doit appartenir à root pour plus de sécurité. Lancez la commande suivante pour que l'on ne puisse pas remonter plus haut dans l'arborescence :

```
# chown root:root /media/sftp
```

Créez un dossier `/home` dans le dossier `/media/sftp` pour que les utilisateurs y soient automatiquement placés à leur connexion. Dans `/media/sftp/home`, il y aura les dossiers portant le nom des utilisateurs.

+ SIMPLEMENT

En fait, on fait comme si le répertoire /media/sftp était la nouvelle racine / pour les utilisateurs.

4.4.2 Ajouter un compte sftp

Ajouter un compte pour se connecter revient à créer un nouvel utilisateur, et ajouter cet utilisateur au groupe sftpusers.

Il faut quand même faire attention à plusieurs points :

— L'utilisateur ne doit pouvoir faire QUE du sftp, on va donc changer son shell
— L'utilisateur n'a pas besoin de dossier personnel dans `/home`, mais aura besoin de son dossier dans `/media/sftp`
— Il faut s'assurer que seul l'utilisateur a des droits d'écriture dans son dossier.

Je vous propose donc un script ci-dessous qui explique la mise en place d'un nouvel utilisateur en faisant attention aux points ci-dessus :

```
#!/bin/sh
# $1 : username
# $2 : hostname
```

```
if [ $# -ne 2 ]; then
    echo "need two argument"
    exit 1
fi

### Création d'un utilisateur, qui n'a pas de shell
# et ne peut donc nuire au serveur
# On ne lui crée pas de /home/utilisateur non plus.
adduser --shell /bin/false --no-create-home "$1"

### On ajoute l'utilisateur au groupe sftpusers
usermod -a -G sftpusers "$1"

### On crée le répertoire utilisateur dans le chroot
mkdir -p "/media/sftp/home/$1"

### Accès seulement (pas d'ecriture) au dossier de l'utilisateur
chmod 555 "/media/sftp/home/$1"

### Répertoire seulement accessible en lecture/écriture
# par l'utlisateur
mkdir -p "/media/sftp/home/$1/prive"
chmod 700 "/media/sftp/home/$1/prive"

### On rend l'utilisateur propriétaire de son répertoire
chown -R $1:$1 "/media/sftp/home/$1"

# ajout de l'utilisateur pour sshd_config
sed -i "s/AllowUsers.*/& $1/" /etc/ssh/sshd_config

service ssh restart

exit 0
```

4.4.3 Utiliser le serveur sftp

Les utilisateurs appartenant au groupe *sftpusers* peuvent télécharger des fichiers sur le serveur (ou en récupérér) avec un client sftp. Par exemple, filezilla est un client multi plateforme qui supporte le protocole sftp.

4.4.4 Accéder aux fichiers via un navigateur web

Afin de rendre les fichiers déposés par les utilisateurs disponibles au public, ajoutez ces lignes dans la configuration de nginx :

```
location ~ ^/~(.+?)(/.*)?$ {
        alias /media/sftp/home/$1/$2;
        index  index.html index.htm;
        autoindex on;
    }
```

Les documents seront accessible à l'adresse http://votreserveur.com/~utilisateur , ce qui est plus pratique que de devoir télécharger un client sftp pour un utilisateur moins averti.

Pour qu'un dossier ne soit pas accessible sur le net (en http) (utilisateur privé), il suffit de changer les droits sur son dossier avec la commande :

```
chmod 700 /media/home/sftp/home/utilisateur
```

4.5 Messagerie instantanée

Pour la messagerie instantanée, quoi de mieux qu'un serveur xmpp ? C'est un protocole libre qui permet à la fois la communication en texte, mais aussi en audio et vidéo, échanger des fichiers et bien plus.

Le programme qui fera office de serveur xmpp sera prosody simple à configurer et très léger.

4.5.1 Installation

L'installation sera très simple avec apt :

```
# apt-get install prosody liblua5.1-sec
```

Ouvrez ensuite les ports[4] 5222 et 5269 en TCP dans votre routeur et votre parefeu.

Enfin, ajoutez deux champs DNS[5] chez votre registrar :

— Champ de type A : xmpp.domaine.net

— champs de type SRV vers xmpp.domaine.net :

— `_xmpp-client._tcp.domaine.net. 18000 IN SRV 0 5 5222 xmpp.domaine.net.`

— `_xmpp-server._tcp.domaine.net. 18000 IN SRV 0 5 5269 xmpp.domaine.net.`

On peut maintenant passer à la configuration du serveur xmpp.

4. voir 3.4
5. voir 3.3

4.5.2 Configuration

La configuration de prosody se passe dans le fichier `/etc/prosody/prosody.cfg.lua`. Il s'agit d'un fichier écrit en langage lua. Pour décommenter une ligne, il faudra retirer les double tirets `--` en début de ligne pour par exemple activer un module.

Par défaut, l'inscription au serveur est désactivée. Pour l'activer, changez cette ligne en :

```
allow_registration = true;
```

Pour votre serveur, indiquez le nom de domaine que vous avez créé dans vos DNS pour la communication xmpp :

```
VirtualHost "xmpp.votreDomaine.net"
```

Tout ce que l'on écrira en dessous cette ligne sera la configuration pour ce domaine uniquement.

En l'occurence, on va activer ce domaine :

```
enabled = true;
```

Enfin, redémarrez prosody :

```
service prosody restart
```

Vous pouvez maintenant commencer à discuter avec votre client favori.

4.5.3 Ajouter un nouveau compte xmpp

Pour ajouter un nouveau compte de messagerie sur votre serveur, vous pouvez utiliser un client comme gajim, ou bien lancer une commande directement sur votre serveur :

```
# prosodyctl adduser utilisateur@xmpp.votreDomaine.net
```

4.5.4 Utiliser une connexion SSL

On se place dans /etc/prosody/certs, puis on lance les commandes :

```
# openssl req -new -x509 -nodes \
    -out votreDomaine.cert -keyout votreDomaine.key
# chown prosody:prosody votreDomaine.*
```

Ou alors, allez dans le dossier /etc/prosody/certs, vous y trouverez un fichier Makefile contenant des explications pour créer facilement des certificats en utilisant juste make.

Ensuite, précisez bien les certificats dans le fichier de configuration de prosody :

```
ssl = {
    key = "/etc/prosody/certs/votreDomaine.key";
    certificate = "/etc/prosody/certs/votreDomaine.cert";
    }
```

Vous pouvez exiger l'encryption en décommentant ces lignes (la seconde ne marche pas toujours :

```
c2s_require_encryption = true
s2s_require_encryption = true
```

Une fois les modifications effectuées, pensez à redémarrer prosody :

```
service prosody restart
```

4.5.5 Ajouter des modules

Prosody propose de nombreux modules. Pour les activer, ajouter leurs noms dans la section `modules_enabled = {`. Vous trouverez la liste des modules disponibles dans /usr/lib/prosody/modules. Remarquez que les modules ont un nom commençant par "mod_". Vous n'avez pas à préciser cette partie lorsque vous chargez un module.

4.5.6 Salons de discussion

Vous pouvez héberger des salons de discussion facilement avec prosody en indiquant dans le fichier de configuration :

```
Component "conf.votreDomaine.net" "muc"
```

4.6 Seedbox

Une seedbox est en fait un serveur qui partage en continu les torrents. De quoi partager ses distributions linux favorites à toutes heures ☺.

4.6.1 Installation de transmission

Tout d'abord, installez le paquet transmission-daemon :

```
# apt-get install transmission-daemon
```

Ensuite, éditez le fichier `/etc/transmission-daemon/settings.json` selon vos préférences.

Une fois vos modifications effectuées, relancez transmission avec la commande

```
# service transmission-daemon reload
```

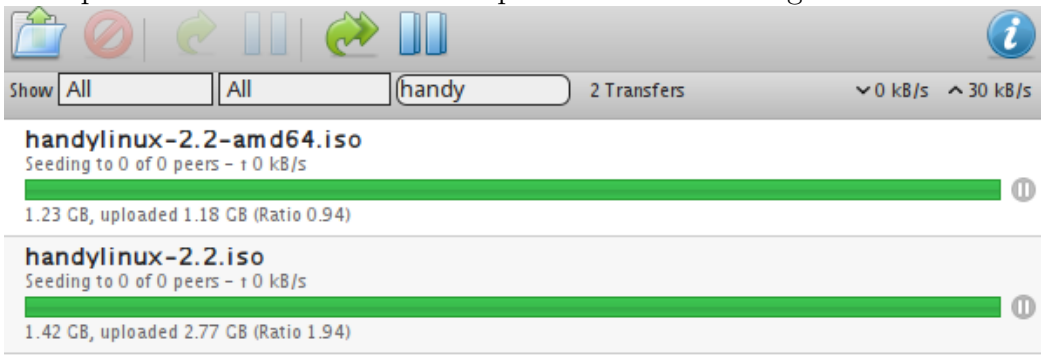
4.6.2 Ajout automatique des torrents dans un dossier

Pour que dès que vous copiez un torrent dans un dossier, ils soit mis en téléchargement, vous devez modifier le fichier `/etc/transmission-daemon/settings.json`, pour y mettre ces deux lignes :

```
"watch-dir": "/chemin/vers/le/dossier/qui/contient/les/torrents",
"watch-dir-enabled": true
```

4.6.3 Accéder à l'interface web avec un navigateur

Vous pouvez contrôler transmission par le biais d'un navigateur.



La suite suppose que vous avez déjà installé et configuré nginx (voir 4.2) En plus, cet accès peut être limité par mot de passe ! Dans le fichier `/etc/transmission-daemon/settings.json`, modifiez ces lignes à votre convenance :

```
"rpc-authentication-required": true,
"rpc-bind-address": "0.0.0.0",
"rpc-enabled": true,
```

```
    "rpc-password": motdepassedelamort,
    "rpc-port": 9091,
    "rpc-url": "/transmission/",
    "rpc-username": "nomdutilisateur",
    "rpc-whitelist": "127.0.0.1",
    "rpc-whitelist-enabled": true,
```

Et pour la configuration de nginx, copiez ce qui suit dans par exemple `/etc/nginx/conf.d/transmission.conf`

```
    server {
            listen 443 ssl;
            server_name votredomaine.com;

            location / {
                proxy_pass_header X-Transmission-Session-Id;
                proxy_set_header X-Forwarded-Host $host;
                proxy_set_header X-Forwarded-Server $host;
                proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
                proxy_pass http://127.0.0.1:9091/transmission/web/;
            }

            # Also Transmission specific
            location /rpc {
                proxy_pass         http://127.0.0.1:9091/transmission/rpc;
            }

    }
```

4.7 Relai TOR

Tor est un logiciel libre permettant de se protéger des surveillances réseau. Lorsqu'on l'utilise, les communications sont réparties à travers une maille de serveurs, afin d'obtenir un onion router. En gros, ce que vous demandez sur internet est réparti entre pleins de serveurs (les couches de l'oignon), ce qui rend très difficile de savoir d'où viennent les paquets, et donc de vous localiser !

Il vous est cependant possible de participer à ce réseau en étant un serveur relais. Qui plus est, cela rendra d'autant plus anonyme vos propres activités, puisqu'il sera difficile de cibler vos centres d'intérêt.

Après avoir installé le paquet `tor`, voici comment configurer un relais.

Assurez-vous d'ouvrir dans votre pare-feu, ainsi que dans votre routeur [6], le port 9001.

Ensuite, éditez le fichier `/etc/tor/torrc` , afin d'obtenir ces quelques lignes :

```
ORPort 9001
Nickname Surnom
RelayBandwidthRate 50 KB
RelayBandwidthBurst 100 KB
ContactInfo votrenom <adresse AT email dot fr>
ExitPolicy reject *:* # no exits allowed
```

Augmentez les valeurs pour `RelayBandwidthRate` et `RelayBandwidthBurst` selon votre connexion internet. C'est la bande passante que vous laissez disponible pour tor.

Enfin, lorsque vous redémarrez tor avec `/etc/init.d/tor restart`, vous devez voir apparaître dans le fichier de log `/var/log/tor/log`

```
Aug 28 14:09:08.000 [notice] Self-testing indicates your
ORPort is reachable from the outside. Excellent. Publishing
server descriptor. Aug 28 14:09:12.000 [notice] Performing
bandwidth self-test...done.
```

4.8 Streaming avec Icecast

Vous pouvez diffuser de la musique ou des vidéos avec icecast. Si si ! De quoi mettre en place votre propre radio ou encore héberger son podcast.

Comment ça marche ? En gros, vous envoyez vos musiques en direct sur le serveur, ou même votre voix, et lui il s'occupe de le rendre accessible sur le réseau.

4.8.1 Installation d'icecast

Hop, on passe comme d'habitude par apt :

```
# apt-get install icecast2
```

6. Voirs 3.4

Lors de l'installation, il est possible que vous ayez à répondre à quelques questions. Répondez que oui, vous le voulez, ou sinon, passez directement à la partie suivante 4.8.2.

Dans le cas où vous avez répondu "Oui" :

— Le nom d'hôte : c'est le nom de domaine de votre serveur. Par exemple, "iloverocknroll.net". Si vous souhaitez utiliser icecast sur un réseau local, laissez "localhost".

— Le mot de passe pour la source : c'est le mot de passe qui servira à envoyer du son sur le serveur, qui le redistribuera.

— Mot de passe pour le relai et pour l'administration : le mot de passe qu'il faudra utiliser pour relayer le flux ou pour administrer icecast.

Si vous utilisez icecast pour diffuser sur internet et non sur un réseau local, pensez à rediriger dans votre box[7] le port 8000 (TCP) vers votre serveur et aussi l'ouvrir dans le pare-feu.

4.8.2 Configuration manuelle de icecast

Votre configuration doit maintenant se retrouver dans le fichier `/etc/icecast2/icecast.xml`. Les lignes importantes à modifier sont toutes celles contenant "hackme" ci-dessous :

```
<authentication>
    <!-- Sources og in with username 'source' -->
    <source-password>hackme</source-password>
    <!-- Relays log in username 'relay' -->
    <relay-password>hackme</relay-password>

    <!-- Admin logs in with the username given below -->
    <admin-user>admin</admin-user>
    <admin-password>hackme</admin-password>
</authentication>

<hostname>localhost(hackme)</hostname>
```

4.8.3 Accéder à l'interface

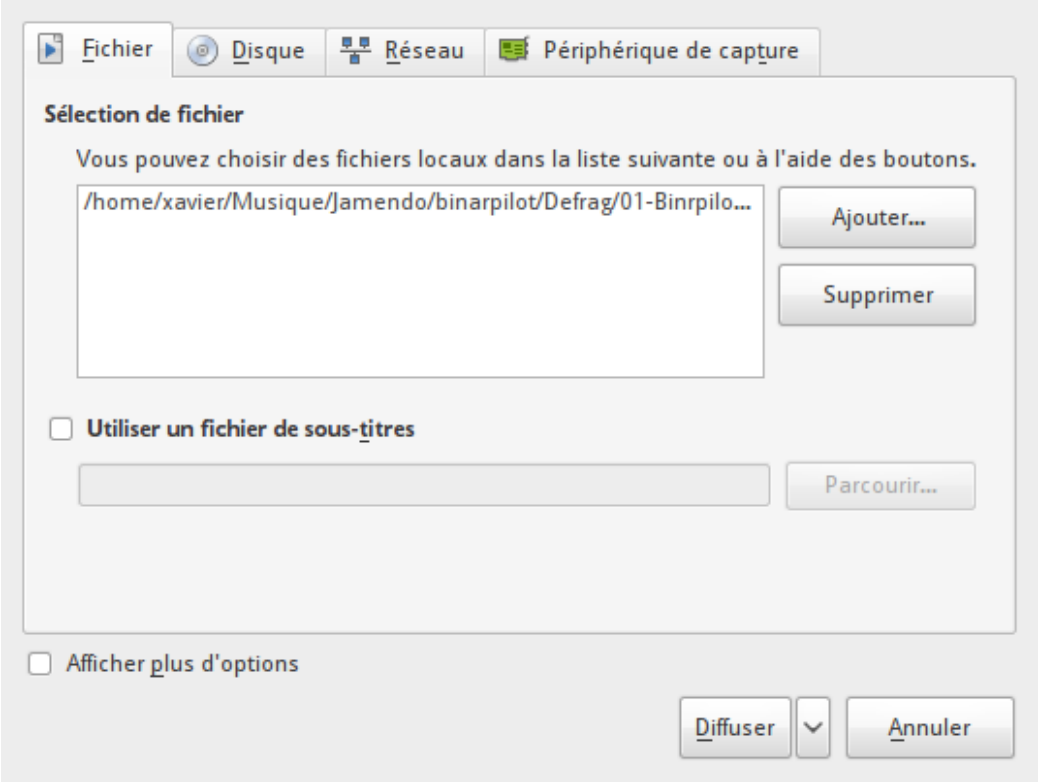
Dans un navigateur, ouvrez l'adresse http ://localhost :8000 ou bien http ://votredomaine.com :8000

7. voir 3.4

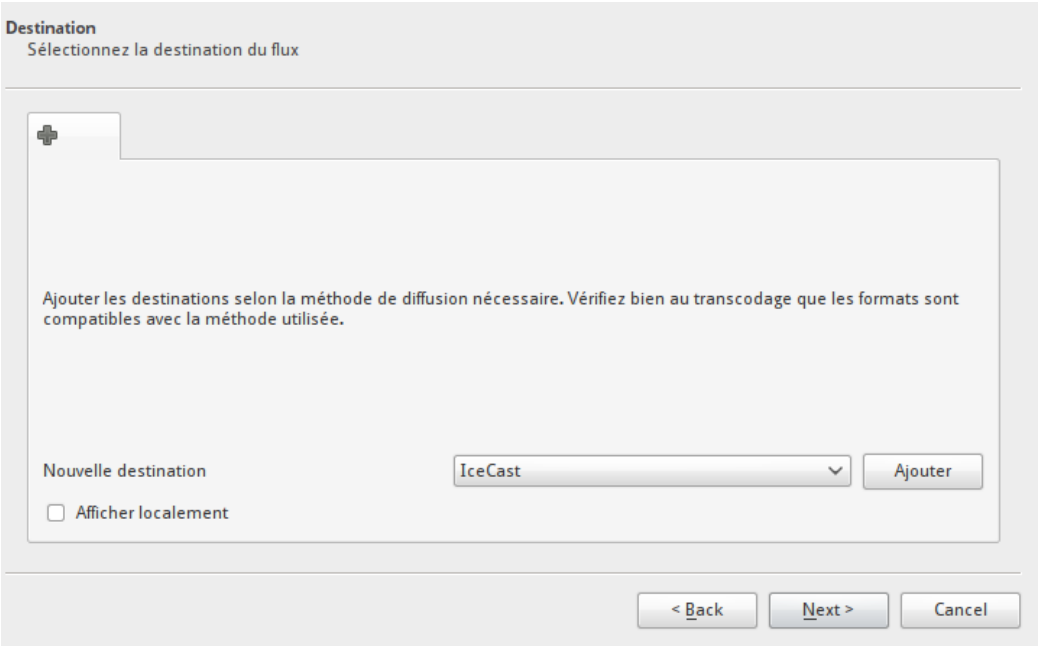
4.8.4 Diffuser un flux

Avec vlc : La solution de facilité, c'est avec vlc.

Une fois vlc ouvert, cliquez sur "Media" -> "Flux". Ajoutez un fichier puis cliquez sur diffuser.



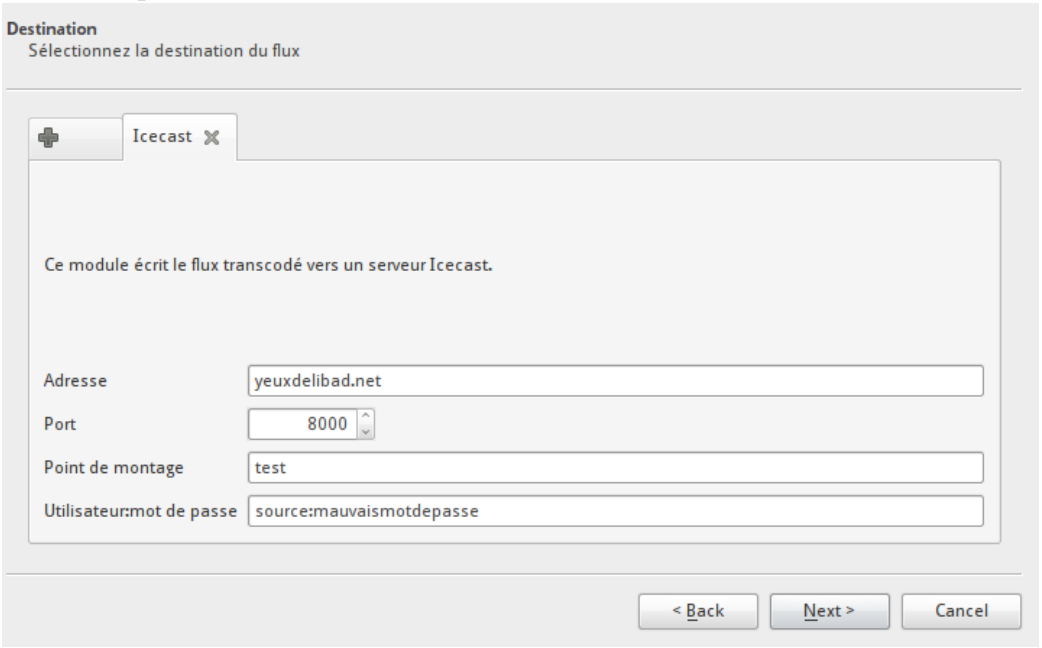
Cliquez sur "Next", puis choisissez dans "Nouvelle destination" "IceCast". Cliquez sur "Ajouter".



Remplissez ensuite les différents champs :

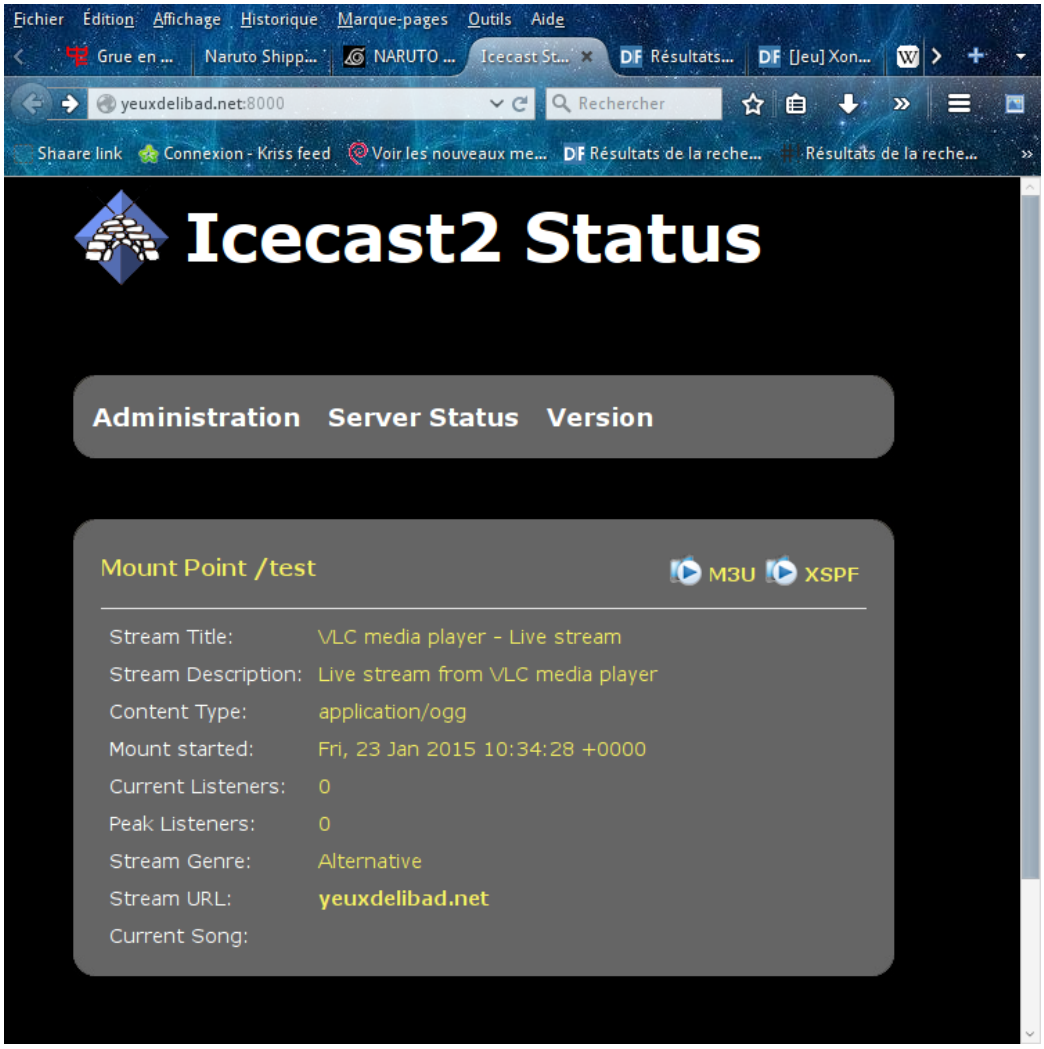
— Adresse : localhost ou bien votredomaine.com
— Port utilisé : 8000 par défaut
— Point de montage : le nom qu'aura votre flux
— Utilisateur :mot de passe : `source:votremotdepasse`

L'utilisateur ici est "source", à moins d'avoir modifié le fichier de configuration sur ce point.



Vous pouvez activer le transcodage si vous le souhaitez. Audio Vorbis (OGG) fonctionne bien en général. Il vous reste à valider la suite.

Vous pouvez alors aller dans votre navigateur à l'interface de icecast pour découvrir votre flux :



En cliquant sur "M3u", vous pouvez écouter votre flux ! Donnez cette adresse à n'importe qui pour qu'il vous écoute.

Diffuser avec ices2 : Ices2 (paquet du même nom à installer) s'utilise pour sa part en ligne de commande. Debian fournit des exemples de configuration dans îtexttt/usr/share/doc/ices2/examples.

Copiez par exemple `/usr/share/doc/ices2/examples/ices2-alsa.xml` et éditez les champs à l'intérieur selon votre configuration, pour obtenir par exemple :

```
<?xml version="1.0"?>
<ices>

    <!-- run in background  -->
```

```
    <background>0</background>
    <!-- where logs go. -->
    <logpath>/tmp/</logpath>
    <logfile>ices.log</logfile>
    <!-- size in kilobytes -->
    <logsize>2048</logsize>
    <!-- 1=error, 2=warn, 3=infoa ,4=debug -->
    <loglevel>4</loglevel>
    <!-- logfile is ignored if this is set to 1 -->
    <consolelog>1</consolelog>

    <!-- optional filename to write process id to -->
    <!-- <pidfile>/home/ices/ices.pid</pidfile> -->

    <stream>
        <!-- metadata used for stream listing -->
        <metadata>
            <name>My amazing stream</name>
            <genre>Variable</genre>
            <description>Un peu de tout et n'importe quoi</description>
            <url>http://votredomaine.net</url>
        </metadata>

        <!--    Input module.

            This example uses the 'alsa' module. It takes input from the
            ALSA audio device (e.g. line-in), and processes it for live
            encoding.  -->
        <input>
            <module>alsa</module>
            <param name="rate">44100</param>
            <param name="channels">2</param>
            <param name="device">pulse</param>
            <!-- Read metadata (from stdin by default, or -->
            <!-- filename defined below (if the latter, only on SIGUSR1) -->
            <param name="metadata">1</param>
            <param name="metadatafilename">test</param>
        </input>

        <!--    Stream instance.
```

```
        You may have one or more instances here.  This allows you to
        send the same input data to one or more servers (or to different
        mountpoints on the same server). Each of them can have different
        parameters. This is primarily useful for a) relaying to multiple
        independent servers, and b) encoding/reencoding to multiple
        bitrates.

        If one instance fails (for example, the associated server goes
        down, etc), the others will continue to function correctly.
        This example defines a single instance doing live encoding at
        low bitrate.  -->

    <instance>
        <!--    Server details.

            You define hostname and port for the server here, along
            with the source password and mountpoint.  -->

        <hostname>votredomaine.net</hostname>
        <port>8000</port>
        <password>c'estunsecret</password>
        <mount>/Thubanstream.ogg</mount>
        <yp>0</yp>   <!-- allow stream to be advertised on YP, default 0 -->

        <!--    Live encoding/reencoding:

            channels and samplerate currently MUST match the channels
            and samplerate given in the parameters to the alsa input
            module above or the remsaple/downmix section below.  -->

        <encode>
            <quality>0</quality>
            <samplerate>22050</samplerate>
            <channels>1</channels>
        </encode>

        <!-- stereo->mono downmixing, enabled by setting this to 1 -->
        <downmix>1</downmix>

        <!-- resampling.
```

```
                Set to the frequency (in Hz) you wish to resample to, -->

            <resample>
                <in-rate>44100</in-rate>
                <out-rate>22050</out-rate>
            </resample>
        </instance>

    </stream>
</ices>
```

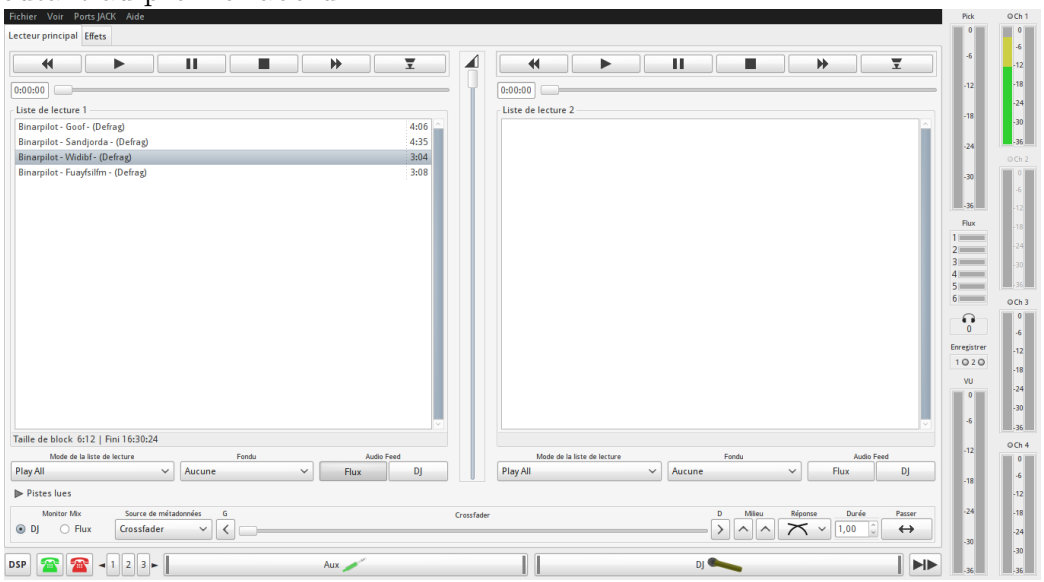
Modifiez la partie `<param name="device">pulse</param>` si vous voulez diffuser ce qu'entend votre micro. (par exemple `<param name="device">hw:0,0</param>`).

Enfin, pour diffuser avec ices2, c'est tout simple. Il faut juste lancer cette commande :

```
ices2 fichier_de_config.xml
```

4.8.5 Créer son podcast

Un logiciel incontournable sur linux pour créer votre podcast est idjc. Il vous permettra de définir une liste de lecture, des sons à jouer ponctuellement (type jingle), d'enregistrer vos émissions, de mixer des appels VoIP avec votre flux. Bref, tout y est pour faire un podcast. Par contre, il peut paraitre déroutant au premier abord.



Vous disposez de 2 panneaux, dans lesquels vous pouvez ajouter des musiques à jouer (via un clic-droit). En dessous des panneaux, vous avez deux

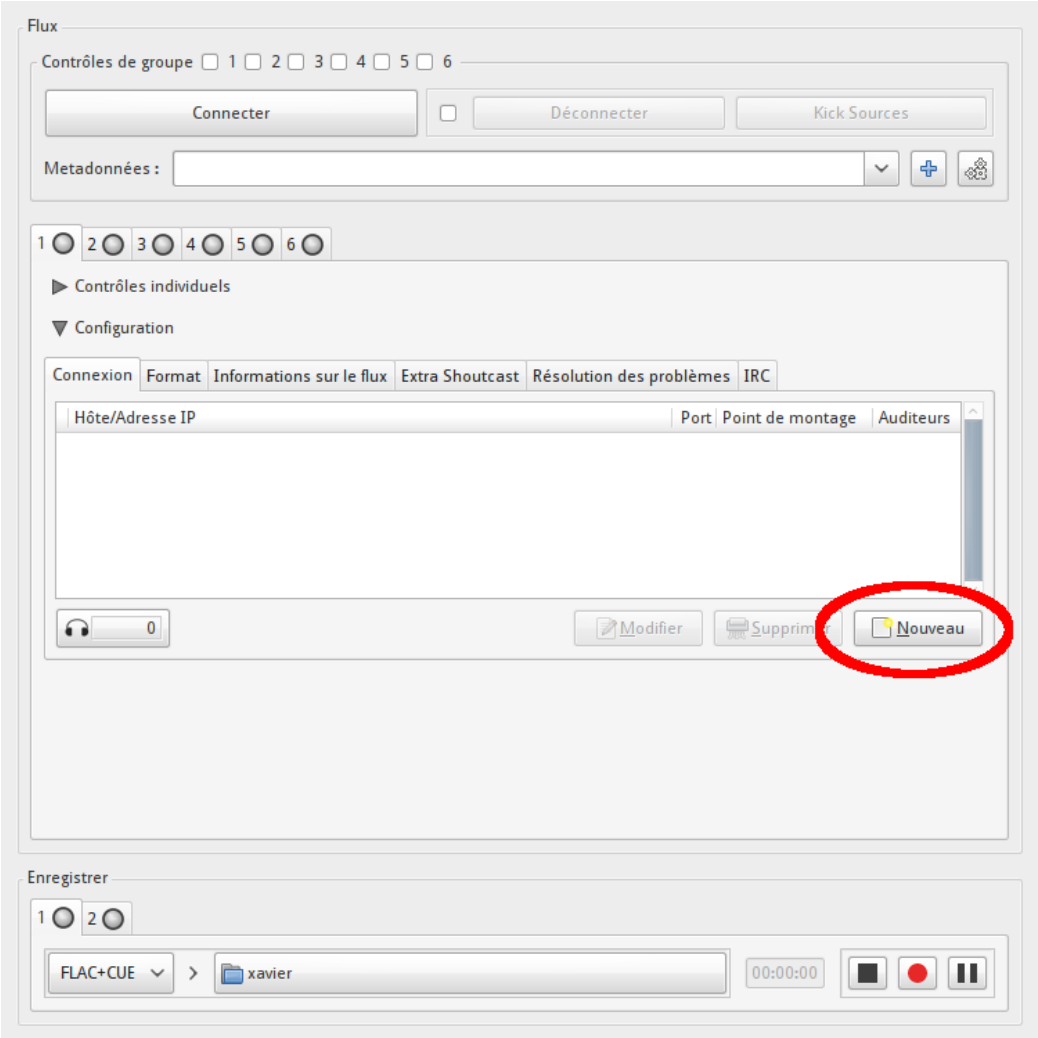
boutons : "Flux" et "DJ". Si "Flux" est appuyé, cela envoie au serveur icecast la musique jouée. Si "DJ" est activé, vous pouvez entendre ce qui est envoyé comme flux.

Une barre de "crossfade" permet de passer d'une playlist à une autre.

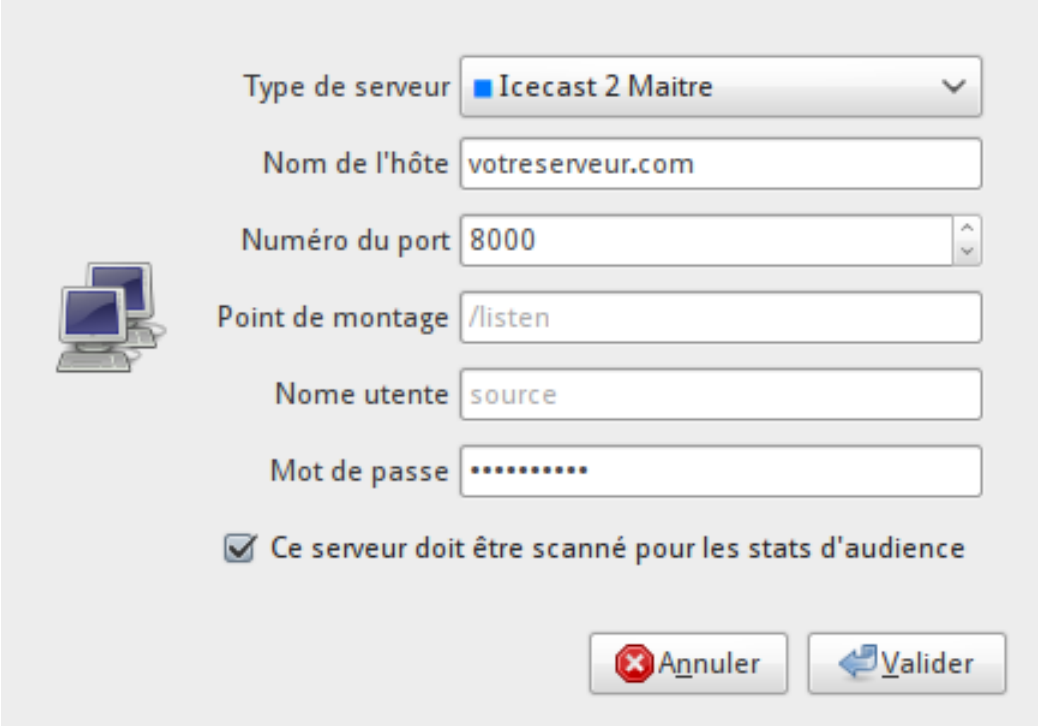
À droite, vous avez tous les niveaux sonores.

En haut, il y a un onglet "Effet", qui permet de définir des jingles.

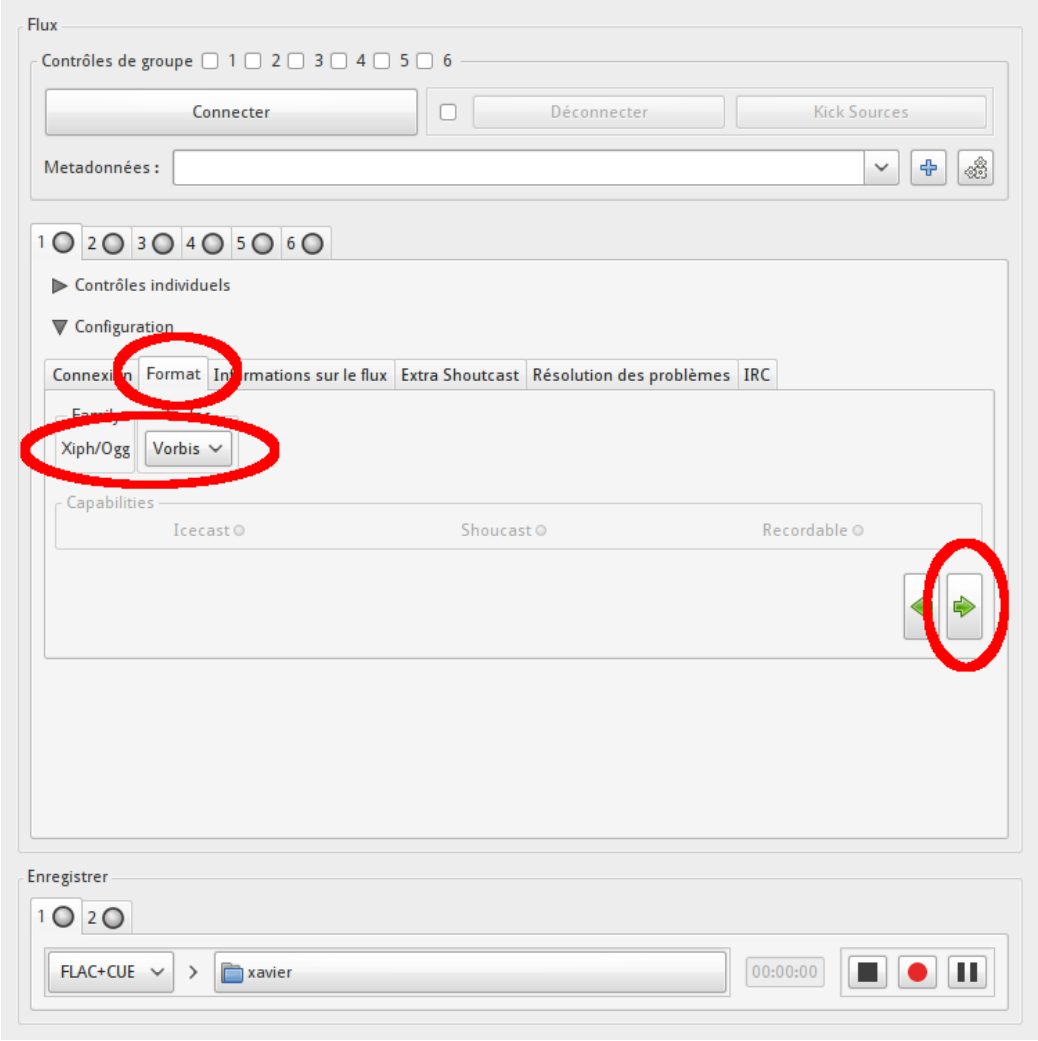
Bon, ce n'est pas tout ça, mais il faut se connecter à notre serveur Ice-Cast. Pour cela, cliquez sur le menu "Voir" -> "Sortie". Une nouvelle fenêtre apparaît.



Dans l'onglet 1, cliquez sur "Configuration", puis sur le bouton "Nouveau", afin de définir les options pour se connecter au serveur.

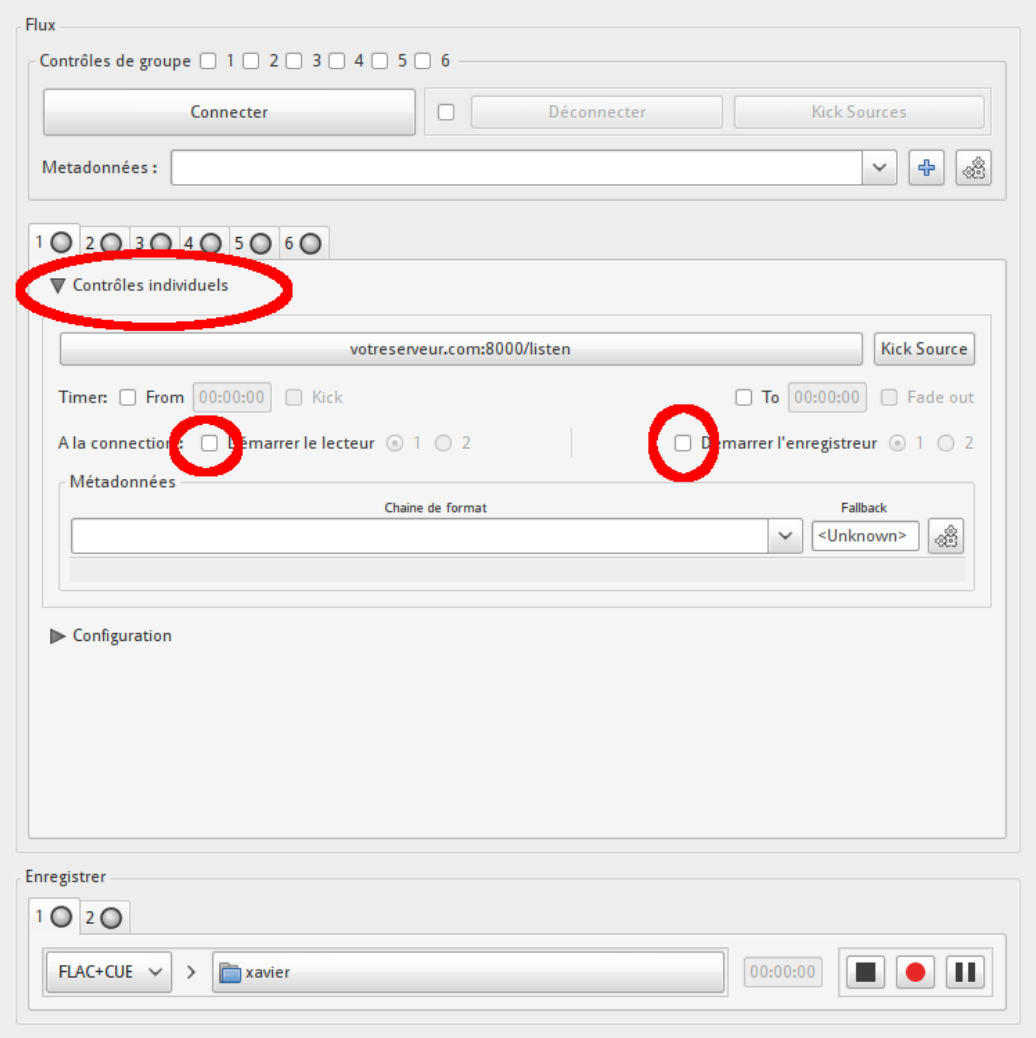


Choisissez ensuite l'onglet "format". Choisissez le paramètre "Family", par exemple Xiph/Ogg, puis cliquez sur la flèche droite de façon à définir les autres paramètres.



Enfin, vous pouvez compléter les derniers paramètres des autres onglets.

Déroulez maintenant "Contrôles individuels". Vous pouvez cocher éventuellement "Démarrer le lecteur" et "Démarrer l'enregistreur" à la connexion. Il vous reste à cliquer sur le bouton "votreserveur.com/listen".



Si quelqu'un vous écoute, vous verrez dans la fenêtre principale en dessous du casque à droite (à côté des niveaux sonores) le nombre d'auditeurs.

Amusez-vous bien ! ☺

5 Services web

Cette partie va présenter l'installation de quelques services web, comme un blog, un webmail ou un forum.

On supposera que nginx et php sont déjà installés comme expliqué au 4.2 , et que vous avez généré des certificats SSL comme indiqué plus loin (voir 5.1).

Afin de décrire la difficulté de l'installation, on va mettre des petites étoiles $\star$ comme pour les exercices de maths.

— ⋆ : installation facile. Pas de base de donnée, très peu de manipulations à faire
— ⋆⋆ : difficulté intermédiaire.
— ⋆⋆⋆ : installation avancée. Une base de donnée sera certainement à mettre en place.

5.1 Générer un certificat SSL

Afin que les données ne circulent pas en clair sur votre site, il faut utiliser le chiffrement ssl. C'est important lorsque des mots de passes sont utilisés pour se connecter (webmail, forum...) Nous allons ici auto-signer le certificat. Les visiteurs de votre site risquent juste d'avoir un avertissement de ce type :

Sachez qu'il est possible d'acheter une autorité de certification. Mais dépenser votre argent n'est pas forcément nécessaire, et un certificat auto-signé ne retire en rien la protection du chiffrement ssl.

Tout d'abord, il faut installer quelques paquets :

```
# apt-get install openssl ssl-cert
```

Pour créer un certificat et le signer, il faut ensuite lancer la commande suivante. Bien sûr, remplacez le nom du fichier certificat.pem à votre convenance :

```
# openssl req -x509 -sha256 -nodes -days 365 -newkey rsa:2048 \
    -keyout /etc/ssl/private/certificat.pem \
    -out /etc/ssl/private/certificat.pem
```

Quelques questions vous seront posées. Vous n'êtes pas obligé de remplir tous les champs.

Finalement, il faut protéger ce certificat. Il faudra alors lancer ces deux dernières commandes afin d'en restreindre les droits d'accès :

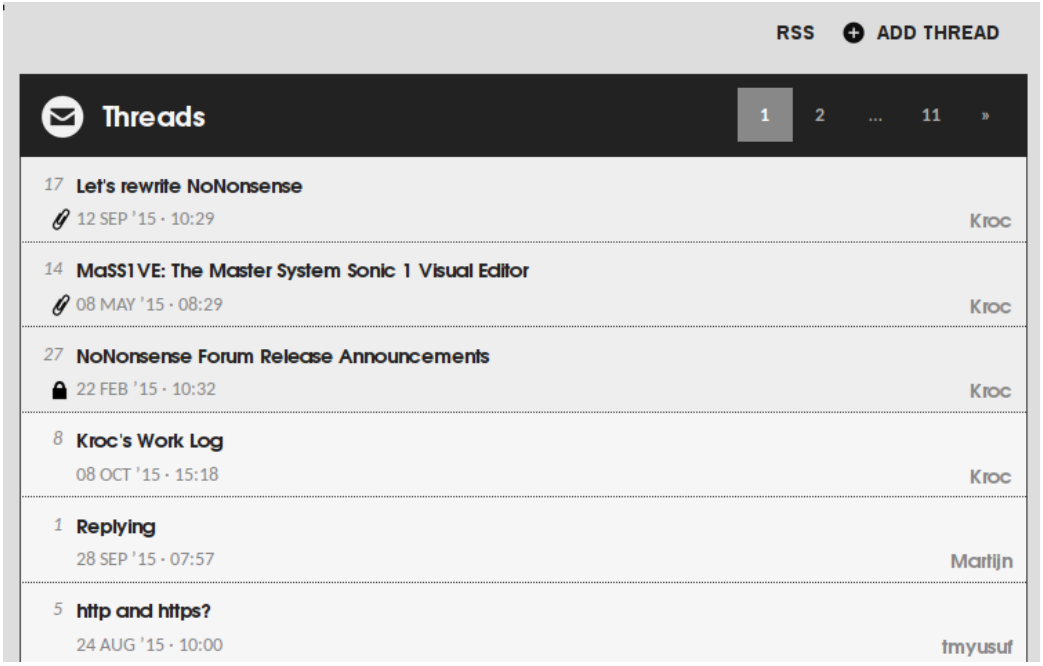
```
# chown root:root /etc/ssl/private/certificat.pem
# chmod 600 /etc/ssl/private/certificat.pem
```

5.2 Exemple simple : NoNonsense Forum ⋆

On va commencer par décrire l'installation d'un petit bijou : NoNonsenseForum, un forum minimaliste, sans base de données, et pourtant génial.

Avec ce forum, pas de compte à créer avec confirmation par mail. Vous choisissez un pseudo lorsque vous envoyer un message, puis vous le réservez avec un mot de passe. Ainsi pas d'inscription, juste un mot de passe à retenir.

Pour l'utiliser et l'installer, vous n'aurez besoin que d'installer le paquet php5 et d'avoir un serveur http fonctionnel. Suivez les indications au 4.2 et 4.2.1.



5.2.1 Installation du forum

Passons à l'installation du forum, qui est une méthode assez classique comme vous le constaterez pour d'autres services :

— Tout d'abord, on télécharge l'archive à cette adresse https://github.com/Kroc/NoNonsenseForum/archive/master.zip

```
wget https://github.com/Kroc/NoNonsenseForum/archive/master.zip
```

— On la décompresse avec unzip : `$ unzip master.zip`. Un dossier NoNonsenseForum-master est créé.

— On s'assure que ce dossier appartient au bon utilisateur :
`# chown -R www-data:www-data NoNonsenseForum-master`

— Dans le dossier NoNonsenseForum-master, on copie le fichier de configuration car on n'utilisera pas le .htaccess avec nginx :
`cp config.default.php config.php` Dans ce fichier, on précise le dossier contenant les informations des utilisateurs :

```
@define ('FORUM_USERS',          '../forum_infos');.
```

— On crée le dossier `forum_infos` puis on change le propriétaire :

```
# mkdir forum_infos
# chown www-data:www-data forum_infos
```

5.2.2 Configuration de nginx pour le forum

Voici ce qu'il faudra mettre dans un nouveau fichier de configuration pour nginx, par exemple dans `/etc/nginx/conf.d/forum/conf`

```
# Forum
server {
listen 80;
server_name forum.mondomaine.net;
index index.php;
root /chemin/vers/NoNonsenseForum-master;
        location ~ \.php$ {
                try_files $uri = 404;
                fastcgi_pass unix:/var/run/php5-fpm.sock;
                include fastcgi_params;
        }
}
```

N'oubliez pas de modifier les variables `server_name` et `root`, puis relancez nginx avec `# service nginx restart`.

Et voilà, votre forum est accessible sur http ://forum.mondomaine.net.

5.3 Un blog

Il existe une multitude de moteurs de blog, plus ou moins complets, et donc plus ou moins faciles à installer.

5.3.1 Blogotext ⋆

Blogotext est un moteur de blog très léger et pourtant très puissant. Il s'avère être très bien pensé et constitue à lui seul un outil complet pour qui veut publier sur le web. En effet, sa description indique ses nombreuses fonctionnalités. En moins de 1 Mo, il vous permet de :

— publier un blog
— partager/sauvegarder des liens
— partager des fichiers et des images
— faire un micro-blogging
— lire vos flux RSS

Pour la suite, nous supposerons que vous voulez avoir votre blog sur le sous-domaine `blog.mondomaine.com`.

Installation de Blogotext Pour installer blogotext, vous aurez besoin d'installer les paquets suivant. Notez que certains sont normalement déjà installés suite aux paragraphes 4.2 et 4.2.1.

```
# apt-get install nginx openssl ssl-cert
php5 php-apcu php5-gd unzip sqlite php5-sqlite
```

On peut maintenant télécharger blogotext :

```
wget "http://lehollandaisvolant.net/blogotext/blogotext.zip"
```

Ensuite, on décompresse l'archive :

```
unzip blogotext.zip
```

Vous voilà avec un dossier `blogotext`. Je vous propose de le déplacer dans le dossier `/var/messites` :

```
mkdir -p /var/messites
mv blogotext /var/messites/
```

Avant d'aller plus loin, modifions les droits du dossier de blogotext :

```
chown -R www-data:www-data /var/messites
```

Allons plus loin

La commande ci-dessus restreint le site à l'utilisateur www-data. C'est un utilisateur un peu spécial, qui est en fait nginx. Cela renforce la sécurité de votre serveur

Nous pouvons désormais passer à la configuration de nginx. Créez un nouveau fichier de configuration, par exemple dans `/etc/nginx/conf.d/blog.conf` :

```
server {
  listen 80;
  server_name blog.mondomaine.com;
  return 301 https://$server_name$request_uri;  # enforce https
}
server {
    listen 443 ssl;
    ssl_certificate /etc/ssl/private/mondomaine.pem;
    ssl_certificate_key /etc/ssl/private/mondomaine.pem;"
    server_name blog.mondomaine.com;
    index index.php;
    root /var/messites/blogotext;
    client_max_body_size 1500M;

    location ~ \.php$ {
        try_files $uri = 404;
        fastcgi_pass unix:/var/run/php5-fpm.sock;
        include fastcgi_params;
        fastcgi_intercept_errors on;
        fastcgi_param HTTPS on;
        fastcgi_param  SCRIPT_FILENAME $document_root$fastcgi_script_name;
    }
}
```

Comme toujours, quelques modifications seront nécessaires. Adaptez à votre cas les variables `server_name`, `ssl_certificate`, `ssl_certificate_key` (voir 5.1) et `root`.

Une fois ceci fait, vous pourrez aller terminer l'installation de blogotext à l'adresse https ://blog.mondomaine.com après avoir rechargé nginx :

```
    service nginx restart
```

Allons plus loin

Petite astuce pour renforcer la sécurité de votre blog après l'installation de blogotext. Renommez le dossier `admin` situé dans le dossier `blogotext` avec ces deux commandes

```
cd /var/messites/blogotext
mv admin nouveau-nom
```

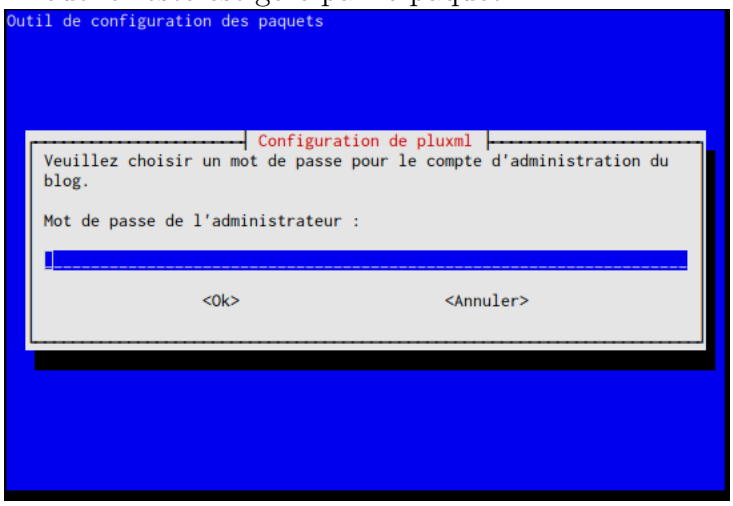
Retenez-bien son nouveau nom.
C'est maintenant à l'adresse https ://blog.mondomaine.com/nouveau-nom que vous pourrez administrer votre blog.

5.3.2 PluXML ⋆

PluXML est un autre moteur de blog. On va utiliser ici le paquet debian pour l'installer.

```
# apt-get install pluxml
```

On vous demande à l'installation le mot de passe de l'administrateur du site. Tout le reste est géré par le paquet.



Ensuite, il n'y a plus qu'à configurer nginx avec le fichier suivant à enregistrer dans `/etc/nginx/conf.d/pluxml.conf`

```
server {
    listen      443 ssl;
    ssl_certificate /etc/ssl/private/votreserveur.pem;
    ssl_certificate_key /etc/ssl/private/votreserveur.pem;
```

```
    server_name blog.monsite.com;
    root /usr/share/pluxml;
    index index.php;
    location ~ \.php$ {
        try_files $uri = 404;
        fastcgi_pass unix:/var/run/php5-fpm.sock;
        include fastcgi_params;
        fastcgi_intercept_errors on;
        fastcgi_param HTTPS on;
        fastcgi_param   SCRIPT_FILENAME  $document_root$fastcgi_script_name;
    }
    # On cache le fichier version:
        location /version {
        return 404;
    }
    # Ligne tres importante pour eviter le vol de mot de passe
        location /data/configuration/users.xml {
        return 403;
    }
    # On cache le dossier update
        location /update {
        return 404;
    }
    # URL Rewriting
        if (!-e $request_filename) {
        rewrite ^/([^feed\/].*)$ /index.php?$1 last;
    }
    rewrite ^/feed\/(.*)$ /feed.php?$1 last;
        # Interdir l'acces au repertoire contenant un fichier .htaccess
        location ~ /\.ht {
        deny  all;
    }
}
```

Modifiez les variables `server_name` et `ssl_` pour les adapter à votre cas.
Rechargez nginx :

```
    # service nginx restart
```

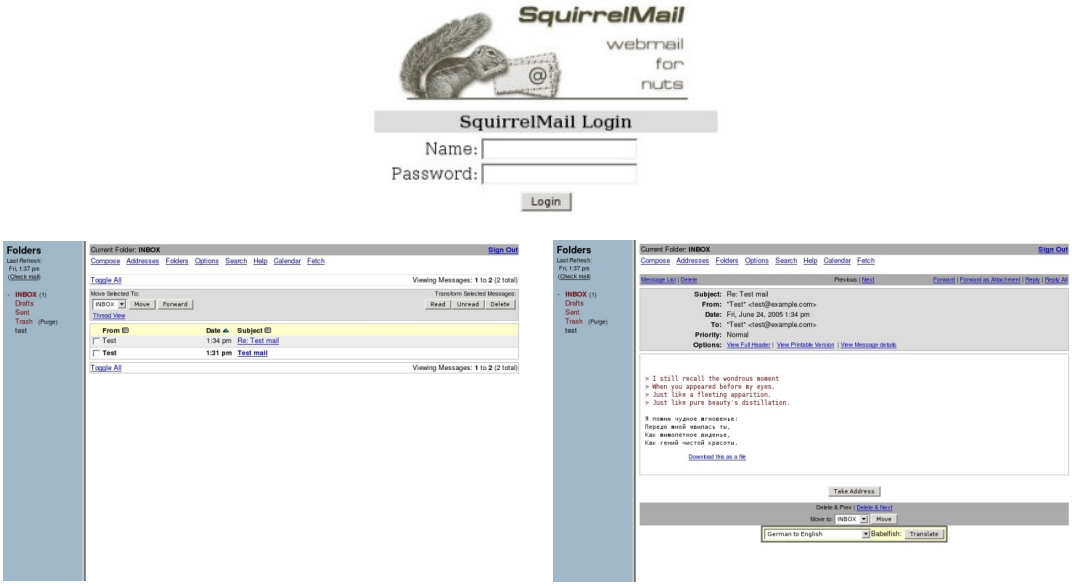
Et voilà, pluxml est maintenant accessible ☺. Pour votre première connexion, le login est “admin” avec le mot de passe définit au-dessus.

5.4 Webmail

Un webmail est une interface que l'on ouvre dans un navigateur web, qui permet de consulter sa messagerie. Il en existe plusieurs, c'est pourquoi les exemples présentés ensuite ne sont pas exhaustifs.

5.4.1 Squirrelmail ⋆

Squirrelmail est un webmail dont le look peut paraître démodé. Cependant, il fonctionne très bien et sait se montrer léger tout en restant facile à utiliser.



Sur debian, squirrelmail et ses dépendances s'installent ainsi :

```
# apt-get install squirrelmail squirrelmail-locales php5-fpm php5 php5-common
```

Vous pouvez compléter l'installation de squirrelmail avec les plugins. Des paquets sont disponibles sur debian. Pour les voir, utilisez la commande `apt-cache search squirrelmail`.

Maintenant, il faut configurer squirrelmail. Pour nous, pas grand chose à faire.

— Lancez `# squirrelmail-configure`
— Tapez `D`, puis entrée.
— Tapez `dovecot` puis entrée.
— Tapez `S` puis entrée.
— (facultatif) Tapez `10` puis entrée : `Default language : fr_FR` et `Default Charset : UTF-8`.

— Le menu `8` permet d'activer d'éventuels plugins.
— Toujours terminer par `S`, puis quittez avec `Q`.

Il reste à ajouter un site à nginx. Copiez la configuration ci-dessous dans `/etc/nginx/conf.d`.

N'oubliez pas de modifier :
— `server_name` en y mettant l'adresse de votre webmail à taper dans un navigateur. Cette adresse sera en https ://webmail.votredomaine.net.
— Le chemin vers votre certificat ssl (voir 5.1) aux variables `ssl_certificate` et `ssl_certificate_key`

```
# Squirrelmail
server {
    listen 443 ssl;
    server_name webmail.votredomaine.net;
    index index.php index.html;
    root /usr/share/squirrelmail;

    ssl_certificate /etc/ssl/private/votreserveur.pem;
    ssl_certificate_key /etc/ssl/private/votreserveur.pem;

    location / {

    ## All regex locations are nested for easier maintenance.

    ## Static files are served directly.
    location ~* .(?:css|gif|jpe?g|js|png|swf)$ {
        expires max;
        log_not_found off;
        ## No need to bleed constant updates. Send the all shebang in one
        ## fell swoop.
        tcp_nodelay off;
        ## Set the OS file cache.
        open_file_cache max=500 inactive=120s;
        open_file_cache_valid 45s;
        open_file_cache_min_uses 2;
        open_file_cache_errors off;
    }

    ## Keep a tab on the 'big' static files.
    location ~* ^.+.(?:m4a|mp[34]|mov|ogg|flv|pdf|ppt[x]*)$ {
        expires 30d;
```

```
        ## No need to bleed constant updates. Send the all shebang in one
        ## fell swoop.
        tcp_nodelay off;
    }

    ## All PHP files that are to be directly processed by the FCGI
    ## upstream are on the src subdirectory of the squirrelmail
    ## distribution.
    location ^~ /src/ {
        location ~* ^/src/[[:alnum:]_]+.php$ {
        fastcgi_pass unix:/var/run/php5-fpm.sock;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
        }
    }

    ## Plugins directory. This is used in the case you're using plugins.
    location ^~ /plugins/ {
        location ~* ^/plugins/[[:alnum:]]+/[[:alnum:]_]+.php$ {
        fastcgi_pass unix:/var/run/php5-fpm.sock;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
        }
    }

    ## The default index handler.
    location = /index.php {
        fastcgi_pass unix:/var/run/php5-fpm.sock;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
        include fastcgi_params;
    }

    ## Protect the web root configure.
    location = /configure {
        return 404;
    }

    ## If no favicon exists return a 204 (no content error).
    location = /favicon.ico {
        try_files $uri @empty;
        log_not_found off;
```

```
        access_log off;
    }

    ## Disable all robots crawling.
    location = /robots.txt {
        return 200 "User-agent: *nDisallow: /n";
    }

    ## Protect the documents directory.
    location ^~ /doc/ {
        return 404;
    }

    ## Protect the translation directory.
    location ^~ /po/ {
        return 404;
    }

    ## Protect the attach and data directories. You can comment
    ## out this if these directories are out of the web root in
    ## your setup. That is the default squirrelmail configuration.
    location ^~ /attach/ {
        return 404;
    }

    location ^~ /data/ {
        return 404;
    }

    ## All files/directories that are protected and unaccessible from
    ## the web.
    location ~* ^.*(.(?:git|htaccess|pl|svn|txt))$ {
        return 404;
    }

    ## All READMEs are off limits.
    location ~* README {
        return 404;
    }

    ## All other PHP files are off limits.
```

```
        location ~* .php$ {
            return 404;
        }

        ## Fallback to index.php if no file is found.
        try_files $uri $uri/ /index.php;
    }
    ## Return a 1x1 in memory GIF.
        location @empty {
            empty_gif;
        }
    }
```

5.4.2 Roundcube avec sqlite – debian ⋆

Debian dispose de paquets pour roundcube, ce qui rend son installation nettement plus simple. Cependant, à l'heure où j'écris ces lignes, ils sont présents dans les backports seulement.

Afin d'ajouter les backports à debian, lancez les commandes suivantes :

```
# echo 'deb http://ftp.debian.org/debian/ jessie-backports main' | \
    tee -a /etc/apt/sources.list.d/backports.list
# apt-get update
```

C'est désormais facile d'installer roundcube avec les métapaquets prévus à cet effet :

```
# apt-get install roundcube-sqlite3 rondcube-plugins roundcube
```

Lors de cette installation, il vous sera demandé si vous voulez configurer la base de données pour roundcube. Répondez oui, et choisissez sqlite3.

Et voilà, il ne vous reste plus qu'à configurer nginx pour roundcube en vous dirigeant à la partie 5.4.6.

5.4.3 Roundcube avec sqlite – archive ⋆⋆

On peut aussi installer roundcube à la main, en récupérant l'archive directement sur le site de roundcube.

Tout d'abord, il faut installer les dépendances

```
    # apt-get install php-pear php5-sqlite php5-fpm php5-apcu \
    php5-mcrypt php5-intl php5-dev php5-gd aspell libmagic-dev sqlite
```

Ensuite, on télécharge roundcube :

```
wget "https://downloads.sourceforge.net/project/roundcubemail/\
roundcubemail/1.1.3/roundcubemail-1.1.3.tar.gz"
```

On extrait roundcube :

```
tar xvf roundcubemail*.tar.gz
```

On déplace roundcube à l'endroit souhaité

```
mv roundcubemail-1.1.3 /var/www/roundcube
```

Enfin, on s'assure que les droits sont corrects :

```
chown -R www-data:www-data /var/www/roundcube
```

Et voilà, vous pouvez passer à la configuration de nginx pour roundcube (voir 5.4.6). Il faudra juste penser à modifier dans la configuration de nginx la variable `root` pour mettre l'emplacement des fichiers de roundcube. Ci-dessus, on avait choisi `/var/www/roundcube`.

Ensuite, ouvrez dans un navigateur https ://votredomaine.com/installer pour terminer l'installation

Après l'installation, supprimez le dossier `installer` dans le dossier de roundcube avec la commande :

```
rm -r /var/www/roundcube/installer
```

5.4.4 Roundcube avec postgresql – debian ⋆⋆

La méthode est quasi-identique à l'installation de roundcube avec sqlite (voir 5.4.2), à ceci près que l'on va installer le paquet `roundcube-pgsql` à la place de `roundcube-sqlite3`.

```
# apt-get install postgresql
# apt-get install roundcube-sqlite3 roundcube-plugins roundcube
```

Ensuite, lorsque dbconfig vous demandera la base de donnée, choisissez bien sûr postgreSQL.

Il vous sera alors demandé un mot de passe qu'il faudra retenir.

Tout le reste est identique ☺.

5.4.5 Roundcube avec postgresql – archive ⋆⋆⋆

Dans ce cas, toutes les manipulations sont identiques au 5.4.3 sauf qu'il est inutile d'installer sqlite. Tout ce qu'on a à voir, c'est comment configurer postgreSQL.

Voici la commande pour installer toutes les dépendances :

```
# apt-get install php5 php-pear php5-fpm php5-apcu\
    php5-mcrypt php5-intl php5-dev php5-gd aspell \
    libmagic-dev php5-pgsql postgresql\
    postgresql-client postgresql-client-common
```

Ensuite, il faut modifier le mot de passe de l'utilisateur postgres qui sert à configurer postgresql, et créer un nouvel utilisateur pour roundcube.

Connectez-vous à postgresql avec la commande :

```
    # su postgres -c psql
```

puis tapez :

```
    postgres=# ALTER USER postgres WITH PASSWORD 'mot_de_passe';
    postgres=# CREATE USER "www-data" WITH PASSWORD 'mot_de_passe';
```

Ensuite, on crée la base de donnée pour roundcube :

```
    postgres=# \connect template1
    postgres=# CREATE DATABASE "_roundcube" WITH ENCODING 'UTF-8';
    postgres=# GRANT ALL PRIVILEGES ON DATABASE "_roundcube" TO "www-data";
    postgres=# ALTER DATABASE "_roundcube" OWNER TO "www-data";
```

Puis quittez en tapant `\q`.

```
    /etc/init.d/postgresql restart
```

Et voilà, ça sera tout pour la configuration de postgresql.

Lorsque vous configurerez roundcube à la première connexion, il faudra alors lui indiquer la base de donnée `_roundcube` avec l'utilisateur `www-data` selon l'exemple donné ci-dessus.

5.4.6 Configuration de nginx pour roundcube

Voici un fichier de configuration pour nginx, à enregistrer par exemple dans `/etc/nginx/conf.d/roundcube.conf`

```
server {
    listen 80;
    server_name webmail.votreserveur.net;
    return 301 https://$server_name$request_uri;  # enforce https
}

server {
    listen 443 ssl;
    server_name webmail.votreserveur.net;
    root /var/lib/roundcube;
    index index.php index.html;

    location ~ ^/favicon.ico$ {
        root /var/lib/roundcube/skins/default/images;
        log_not_found off;
        access_log off;
        expires max;
    }

    location = /robots.txt {
        allow all;
        log_not_found off;
        access_log off;
    }

    location ~ ^/(README|INSTALL|LICENSE|CHANGELOG|UPGRADING)$ {
        deny all;
    }
    location ~ ^/(bin|SQL)/ {
        deny all;
    }

    location ~ \.php$ {
            try_files $uri = 404;
            fastcgi_pass unix:/var/run/php5-fpm.sock;
            include fastcgi_params;
            fastcgi_intercept_errors on;
```

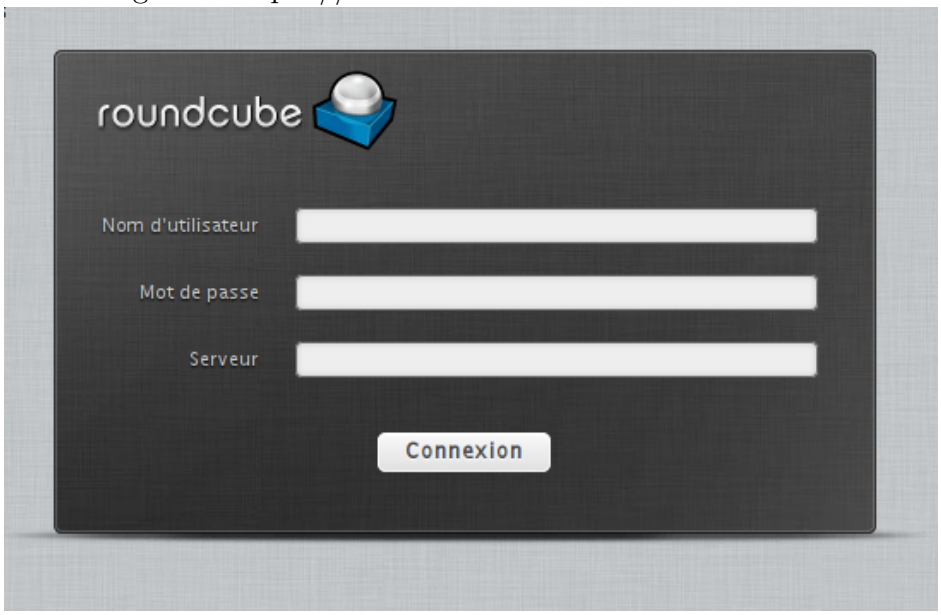
```
            fastcgi_param HTTPS on;
            fastcgi_param   SCRIPT_FILENAME  $document_root$fastcgi_script_name;
        }

}
```

Modifiez la variable `server_name` selon l'adresse prévue pour votre webmail (voir 3.3), puis relancez nginx :

```
# service nginx restart
```

Une fois nginx relancé, vous pourrez alors accéder à roundcube en tapant dans un navigateur https ://webmail.votredomaine.com.



Vous verrez alors un écran de connexion. Mettez votre nom d'utilisateur pour le compte mail et le mot de passe associé. Pour la ligne "Serveur", il faut simplement mettre `votredomaine.com`.

Pour ne plus avoir à préciser le serveur, éditez le fichier `/var/lib/roundcube/config/config.inc.php` puis remplacez

```
$config['default_host'] = '';
```

par

```
$config['default_host'] = 'votredomaine.com';
```

5.5 Exemple détaillé : CMS Spip ⋆⋆⋆

Spip est un gestionnaire de contenu simplifié, qui permet notamment de déployer facilement son site web.

5.5.1 Installation de spip et des dépendances

Prêt pour l'installer ? C'est parti !

Puisqu'on utilise debian, on va profiter de sa gestion intelligente des paquets. On installe tout d'abord le serveur http qui délivrera les pages web, puis spip.

```
# apt-get install nginx php5-cgi php5-fpm
# apt-get install spip
# rm /etc/nginx/sites-enabled/default
```

En procédant par étape, on évite normalement l'installation d'apache. Sinon, comme on a déjà nginx, on peut le supprimer (apache) sans inquiétudes. De la même façon, on supprime le site par défaut qui permet de tester si nginx fonctionne bien.

N'oubliez pas d'ouvrir le port 80 et/ou 443 si vous utilisez le chiffrement ssl (voir page 8).

5.5.2 Choix d'une base de donnée

Selon ce que vous préférez, vous pouvez choisir votre base de donnée : MySQL ou SQLite. MySQL est complet et conviendra pour un site conséquent, mais est plus difficile à mettre en place. SQLite est extrêmement simple, tout en restant performant. Pour de l'auto-hébergement, SQLite me semble le choix le plus approprié.

Avec SQLite, installez les paquets sqlite et php5-sqlite :

```
# apt-get install sqlite php5-sqlite
```

Et c'est tout ! Vous pouvez passer à la partie suivante.

Pour MySQL, on va installer les paquets nécessaires, puis créer une base donnée.

```
# apt-get install mysql-server php5-mysql
```

On vous demandera un mot de passe pour la base de données. Retenez-le bien. Afin de donner les droits nécessaires à la base de donnée, il faut maintenant taper ces commandes :

```
$ mysql -u root -p
    Enter password: * entrez le mot de passe choisi à l'étape précédente*
mysql> CREATE DATABASE spipdb;
mysql> CREATE user 'spip'@'localhost' IDENTIFIED BY 'mot_de_passe_pour_spip';
mysql> GRANT ALL privileges ON spipdb.* TO 'spip'@'localhost';
```

Pour revenir à l'invite de commande normal, appuyez simultanément sur Ctrl et d.

Quelques explications :

1. On crée la base de donnée pour spip, appelée *spipdb*.
2. On crée un utilisateur *spip* avec son mot de passe.
3. On accorde à *spip* les droits suffisants sur la base *spipdb*.

5.5.3 Configuration du serveur http pour spip

Bon, si vous tapez `dpkg -S spip`, vous remarquez que les fichiers sont installés dans `/usr/share/spip`. Il ne nous reste donc qu'à configurer nginx de façon adaptée.

On crée donc un fichier nginx-spip.conf dans `/etc/nginx/conf.d/`, pour y mettre le contenu suivant :

```
server {
    server_name votredomaine.net www.votredomaine.net;
    client_max_body_size 10m;
    root /usr/share/spip;
    index index.php;

    location / {
        try_files $uri $uri/ /spip.php?q=$uri&$args;
    }

    location ~^/(tmp|config|local)/{
        deny all;
        return 403;
    }

    location ~ \.php$ {
```

```
        try_files $uri =404;
        fastcgi_pass unix:/var/run/php5-fpm.sock;
        fastcgi_split_path_info ^(.+\.php)(/.+)$;
        fastcgi_index  index.php ;
        fastcgi_buffers 16 16k;
        fastcgi_buffer_size 32k;
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    }

}
```

Remplacez `votredomaine.net` par votre nom de domaine, et hop, le tour est joué ! Relancez nginx en tapant : `#service nginx restart`.

Voyons voir ce qu'il en est à l'adresse http ://votredomaine.net. Pour l'instant, on voit quelque chose comme ça :

On peut donc maintenant passer à la configuration de spip.

Je vous conseille vivement d'ajouter le chiffrement ssl à votre site. Pour cela, référez vous à la page 43, puis complétez le fichier `nginx-spip.conf` ci-dessus ainsi, sans oublier de préciser le nom de `votrecle.pem` :

```
server {
    listen 80;
    server_name ${NOMDHOTE};
    server_name votredomaine.net www.votredomaine.net;
    return 301 https://$server_name$request_uri;  # enforce https
}
server {
    server_name ${NOMDHOTE};
    ssl_certificate /etc/ssl/private/votrecle.pem;
    ssl_certificate_key /etc/ssl/private/votrecle.pem;"
    client_max_body_size 10m;
```

```
    root /usr/share/spip;
    index index.php;

    location / {
        try_files $uri $uri/ /spip.php?q=$uri&$args;
    }

    location ~^/(tmp|config|local)/{
        deny all;
        return 403;
    }

    location ~ \.php$ {
        try_files $uri =404;
        fastcgi_pass unix:/var/run/php5-fpm.sock;
        fastcgi_split_path_info ^(.+\.php)(/.+)$;
        fastcgi_index  index.php ;
        fastcgi_buffers 16 16k;
        fastcgi_buffer_size 32k;
        include fastcgi_params;
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;
    }

}
```

5.5.4 Configuration de spip

Comme il est indiqué dans le fichier `/usr/share/doc/spip/README.Debian`, que bien sûr, vous avez lu, il faut ajouter le site spip en tapant cette commande :

```
    # spip_add_site votredomaine.net
```

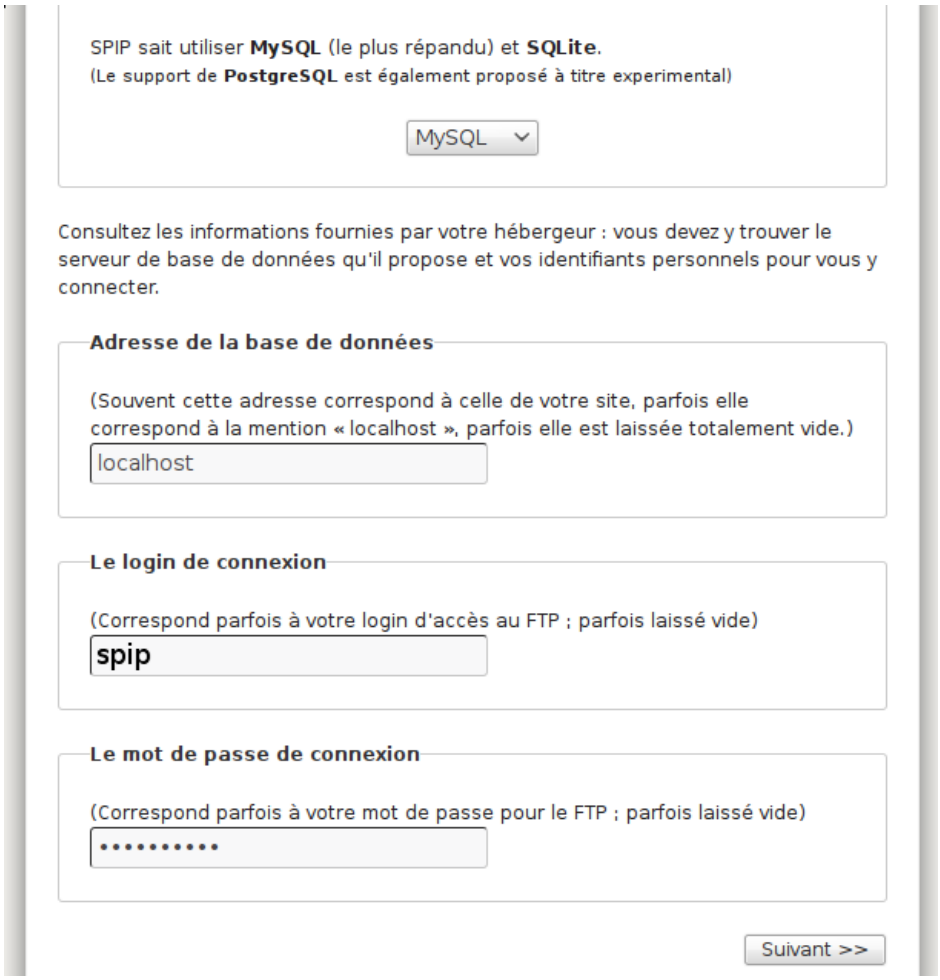
Vous obtenez alors une réponse qui ressemble à :

```
Creating site votredomaine.net...
Directories and files created, you may add hosts for this site in:
   /etc/spip/sites/votredomaine.net.php
```

Dirigez-vous maintenant à l'adresse http ://votredomaine.net/ecrire pour terminer l'installation.

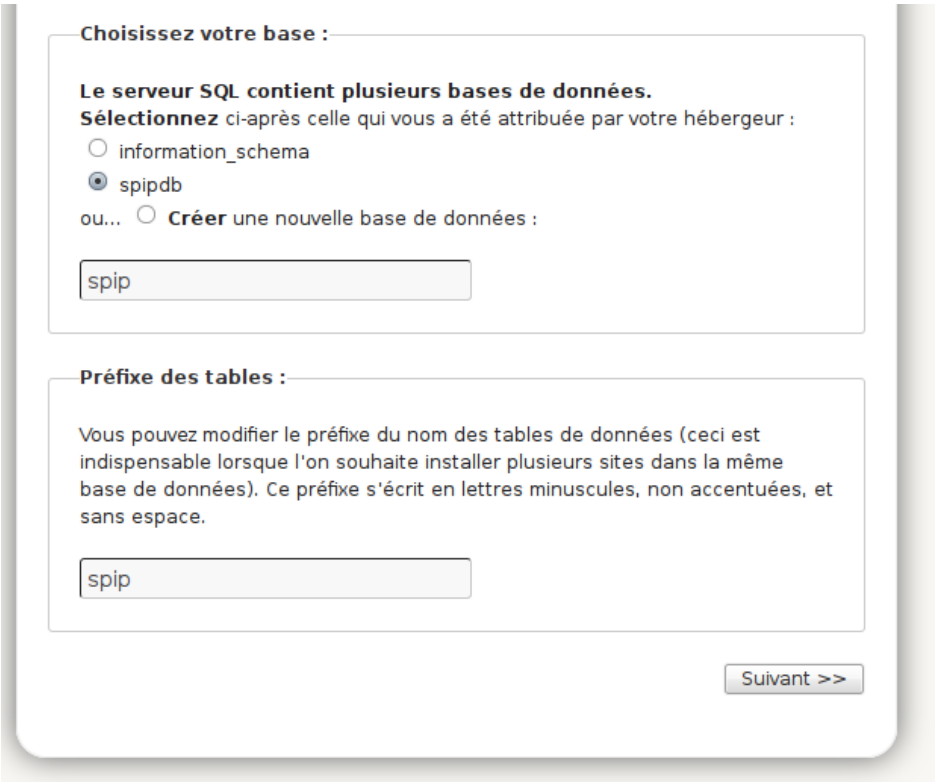
Si vous choisissez une base de données SQLite, c'est extrêmement simple, les choix par défaut sont compréhensibles et suffisants.

Si vous choisissez une base MySQL, quelques explications s'imposent :

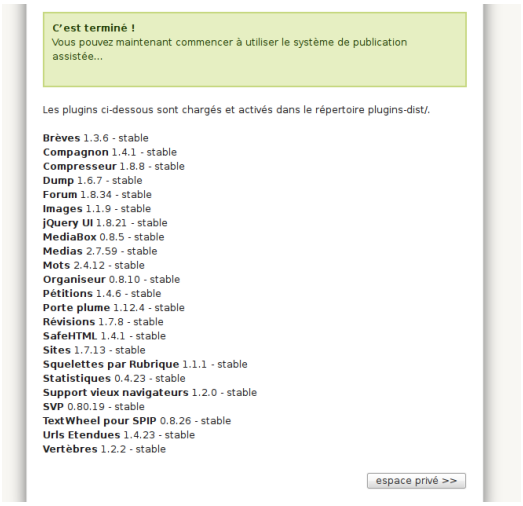


— Adresse de la base de données : localhost
— Login de connexion : c'est l'utilisateur créé tout à l'heure. Dans l'exemple, il s'appelait *spip*.
— Le mot de passe de connexion : il s'agit du mot de passe pour l'utilisateur *spip*. Dans l'exemple, c'était `mot_de_passe_pour_spip`.

Ensuite, choisissez de sélectionner la base de données déjà créée (on l'avait appelée spipdb).



La suite est très classique :



Ne vous inquiétez pas de l'avertissement au sujet du fichier htaccess. On s'en est en fait déjà occupé dans la configuration de nginx.

Il ne reste plus qu'à vous connecter, puis d'administrer votre site.

5.6 Un Wiki avec Dokuwiki

Dokuwiki est un moteur de wiki libre, facile à utiliser qui ne nécessite pas beaucoup de ressources et aucune base de données.
Il dispose en plus de nombreux greffons qui permet de lui ajouter tout un tas de fonctionnalités.

5.6.1 Méthode debian ⋆

Nous allons ici décrire l'installation de dokuwiki à l'aide des paquets debian.

Rien de bien compliqué, il faut juste installer le paquet `dokuwiki` :

```
# apt-get install dokuwiki
```

Quelques questions vont vous être posées, comme par exemple le mot de passe de l'administrateur du wiki.

Vous pouvez maintenant passer à la configuration de nginx pour dokuwiki au 5.6.3.

Lors de votre première connexion, le nom d'utilisateur est “admin” et le mot de passe celui définit précédemment.

5.6.2 Méthode avec l'archive ⋆⋆

Cette méthode un peu plus contraignante présente l'avantage de pouvoir installer plusieurs wiki sur son serveur.

Tout d'abord, on installe le nécessaire au bon fonctionnement de dokuwiki :

```
# apt-get install php5-apcu php5-gd imagemagick php-geshi php-seclib
```

Ensuite, on récupère l'archive de dokuwiki :

```
wget "http://download.dokuwiki.org/src/dokuwiki/dokuwiki-stable.tgz"
```

On décompresse l'archive

```
tar xvf dokuwiki-stable.tgz
```

Maintenant, on déplace le dossier de dokuwiki à un emplacement qui nous convient mieux, par exemple `/var/www/dokuwiki`

```
mv dokuwiki-* /var/www/dokuwiki
```

Enfin, on règle les droits sur le dossier de dokuwiki

```
chown -R www-data:www-data /var/www/dokuwiki
```

Et voilà, vous pouvez maintenant configurer nginx pour dokuwiki en passant au 5.6.3.

5.6.3 Configuration de nginx pour dokuwiki

Voici la configuration de nginx pour dokuwiki, à enregistrer par exemple dans `/etc/nginx/conf.d/dokuwiki.conf`.

Si vous utilisez le paquet debian, laissez la variable `root` à `/usr/share/dokuwiki`. Sinon, modifiez ce chemin vers l'emplacement où vous avez décompressé l'archive de dokuwiki.

De même, la variable `server_name` est à modifier selon le domaine choisi pour accéder à votre wiki.

Enfin, adaptez les variables pour le certificat ssl selon la configuration effectuée au 5.1.

```
server {
    listen 80;
    server_name monwiki.com;
    return 301 https://$server_name$request_uri;  # enforce https
}
server {
    listen      443 ssl;
```

```
        ssl_certificate /etc/ssl/private/mondomaine.pem;
        ssl_certificate_key /etc/ssl/private/mondomaine.pem;"
        server_name monwiki.com;
        root /usr/share/dokuwiki;
        client_max_body_size 1500M;
        index index.html index.php doku.php;
        include /etc/nginx/conf.d/php;

        location ~ /(data|conf|bin|inc)/ {
            deny all;
        }

    # serve static files
        location ~ ^/dokuwiki/lib/^((?!php).)*$  {
        root         /usr/share/dokuwiki/lib;   #adapt if needed
        expires 30d;
        }
    }
```

Recharger nginx pour accéder à votre site :

```
# service nginx restart
```

5.7 Un autre forum : FluxBB ⋆⋆⋆

FluxBB est un moteur de forum complet, qui nécessite une base de données. On va décrire ici l'installation avec PostgreSQL, bien qu'il soit tout à fait possible d'utiliser MySQL en suivant les indications du 7.

Installons tout d'abord les dépendances pour fluxbb :

```
# apt-get install php5-fpm php5-apcu php-pear php-db\
php5-pgsql postgresql postgresql-client postgresql-client-common
```

On configure ensuite postgresql. Il faut modifier le mot de passe de l'utilisateur postgres qui sert à configurer postgresql, et créer un nouvel utilisateur pour fluxbb.

Connectez-vous à postgresql avec la commande :

```
# su postgres -c psql
```

puis tapez :

```
postgres=# ALTER USER postgres WITH PASSWORD 'mot_de_passe';
postgres=# CREATE USER "fluxbbuser" WITH PASSWORD 'mot_de_passe';
```

Ensuite, on crée la base de donnée pour fluxbb :

```
postgres=# \connect template1
postgres=# CREATE DATABASE "_fluxbbdb" WITH ENCODING 'UTF-8';
postgres=# GRANT ALL PRIVILEGES ON DATABASE "_fluxbbdb" TO "fluxbbuser";
postgres=# ALTER DATABASE "_fluxbbdb" OWNER TO "fluxbbuser";
```

Puis quittez en tapant `\q`.

Relancez postgresql :

```
/etc/init.d/postgresql restart
```

Et voilà, ça sera tout pour la configuration de postgresql.

Lorsque vous configurerez fluxbb, il faudra alors lui indiquer la base de donnée `_flubxxdb` avec l'utilisateur `fluxbbuser` selon l'exemple donné ci-dessus.

Vous pouvez récupérer l'archive de fluxbb en consultant cette page https://fluxbb.org/downloads/. À l'heure où j'écris ces lignes, c'est la version 1.5.8 qui est disponible.

```
wget https://fluxbb.org/download/releases/1.5.8/fluxbb-1.5.8.tar.gz
```

Ensuite, on décompresse l'archive, et on la déplace dans `/var/www/fluxbb`. On n'oubliera pas de modifier les droits sur ce dossier :

```
tar xvf fluxbb-1.5.8.tar.gz
mv fluxbb-1.5.8 /var/www/fluxbb
chown -R www-data:www-data /var/www/fluxbb
```

On peut finalement ajouter ce fichier de configuration à nginx, à enregistrer par exemple dans `/etc/nginx/conf.d/fluxbb.conf`. N'oubliez pas de modifier les variables `root`, `server_name` et `ssl_*`.

```
server {
listen 80;
server_name forumquidechire.com;
return 301 https://$server_name$request_uri;  # enforce https
}
```

```
server {
    listen 443 ssl;
    ssl_certificate /etc/ssl/private/mondomaine.pem;
    ssl_certificate_key /etc/ssl/private/mondomaine.pem;"
    index index.php;
    root /var/www/fluxbb;

    location ~ \.php$ {
        try_files $uri = 404;
        fastcgi_pass unix:/var/run/php5-fpm.sock;
        include fastcgi_params;
        fastcgi_intercept_errors on;
        fastcgi_param HTTPS on;
        fastcgi_param   SCRIPT_FILENAME  $document_root$fastcgi_script_name;
    }
}
```

Redémarrez nginx avec `service nginx restart`, puis digirez-vous à l'adresse https ://votreforum.com/install.php pour terminer l'installation.

5.8 Un pastebin chiffré : ZeroBin ⋆

ZeroBin permet de partager du texte et d'en discuter tout en chiffrant le contenu afin de garder l'ensemble le plus discret possible.

Il est nécessaire d'avoir php et nginx d'installés pour l'utiliser :

```
# apt-get install nginx php5 php5-fpm php5-gd unzip
```

Ensuite, on va récupérer l'archive de zerobin et la décompresser :

```
wget "https://github.com/sebsauvage/ZeroBin/archive/master.zip"
unzip master.zip
```

On peut maintenant déplacer les fichiers de zerobin à un emplacement prévu pour votre serveur, par exemple `/var/www/zerobin`. On corrigera les droits par la même occasion.

```
mv ZeroBin-master /var/www/zerobin
chown -R www-data:www-data /var/www/zerobin
```

Il ne reste plus qu'à ajouter la configuration de nginx dans par exemple `/etc/nginx/conf.d/zerobin.conf`.

```
    server {
    listen 80 ;
    server_name pastebin.monserveur.net;
    root /var/www/zerobin;
    index index.php;
    location ~ \.php$ {
        try_files $uri = 404;
        fastcgi_pass unix:/var/run/php5-fpm.sock;
        include fastcgi_params;
        fastcgi_intercept_errors on;
        fastcgi_param HTTPS on;
        fastcgi_param   SCRIPT_FILENAME  $document_root$fastcgi_script_name;
    }
}
```

Comme d'habitude, pensez à adapter à votre cas les variables `root` et `server_name`.

Relancez nginx puis profitez de votre nouveau pastebin.

```
    # service nginx restart
```

5.9 Partage de liens avec shaarli ⋆

Shaarli est une application qui permet de partager et mettre de côté des liens que l'on trouve intéressant. Certains s'en servent même de blog.

Vous allez voir que l'installation ressemble beaucoup à celle de zerobin (5.8), puisque c'est le même développeur.

Il est aussi nécessaire d'avoir php et nginx d'installés pour l'utiliser :

```
    # apt-get install nginx php5 php5-fpm php5-gd unzip
```

Ensuite, on va récupérer l'archive de shaarli et la décompresser :

```
    wget "https://github.com/shaarli/Shaarli/archive/master.zip"
    unzip master.zip
```

On peut maintenant déplacer les fichiers de shaarli à un emplacement prévu pour votre serveur, par exemple `/var/www/shaarli`. On corrigera les droits par la même occasion.

```
    mv Shaarli-master /var/www/shaarli
    chown -R www-data:www-data /var/www/shaarli
```

Il ne reste plus qu'à ajouter la configuration de nginx dans par exemple `/etc/nginx/conf.d/shaarli.conf`.

```
server {
    listen 80;
    server_name links.monserveur.net;
    return 301 https://$server_name$request_uri;  # enforce https
}

server {
    listen      443 ssl;
    ssl_certificate /etc/ssl/private/mondomaine.pem;
    ssl_certificate_key /etc/ssl/private/mondomaine.pem;"
    server_name links.monserveur.net;
    root /var/www/shaarli;
    index index.php;
    location ^~ /cache {
        deny all;
        return 403;
    }
    location ^~ /data {
        deny all;
        return 403;
    }
    location ~ \.php$ {
        try_files $uri = 404;
        fastcgi_pass unix:/var/run/php5-fpm.sock;
        include fastcgi_params;
        fastcgi_intercept_errors on;
        fastcgi_param HTTPS on;
        fastcgi_param   SCRIPT_FILENAME  $document_root$fastcgi_script_name;
    }
}
```

Comme d'habitude, pensez à adapter à votre cas les variables `root`, `server_name` et celles concernant le certificat ssl `ssl_*`.

Relancez nginx puis profitez de votre nouveau shaarli.

```
    # service nginx restart
```

5.10 Statistiques sur votre site

5.10.1 Avec Webalizer ⋆⋆

webalizer est un outil pour obtenir des statistiques sur votre site. Il suffit de l'installer avec apt-get, puis d'ajouter ces lignes dans le fichier de configuration de nginx, afin d'accéder aux statistiques à la page http ://nom-du-serveur/Divers/stats/index.html.

```
# Pour Webalizer
location /Divers/stats {
    alias /var/www/webalizer;
}
```

Vous pouvez surveiller plusieurs domaines. Pour cela, copier le fichier de configuration de webalizer par défaut, 1 fichier par nom de domaine à surveiller :

```
cp /etc/webalizer/webalizer.conf.sample /etc/webalizer/domaine1.conf
cp /etc/webalizer/webalizer.conf.sample /etc/webalizer/domaine2.conf
cp /etc/webalizer/webalizer.conf.sample /etc/webalizer/domaine3.conf
```

Créez ensuite les dossiers pour accueillir les pages générées par webalizer, et donnez l'accès à ces dossiers à nginx :

```
mkdir -p /var/www/webalizer-domaine1
mkdir -p /var/www/webalizer-domaine2
mkdir -p /var/www/webalizer-domaine3
chown -R www-data:www-data /var/www/webalizer-domaine*
```

Adaptez ces nouveaux fichiers de configuration, en particulier changez les variables suivantes :

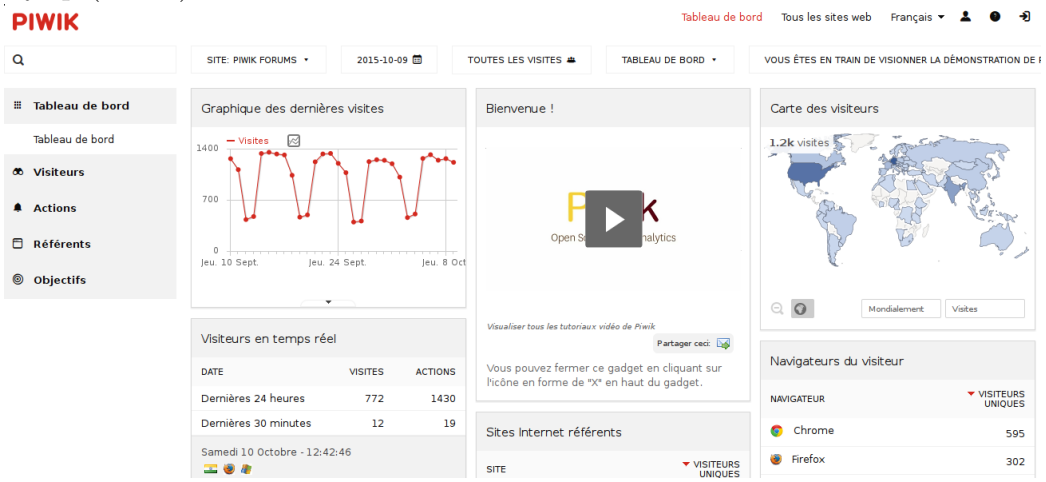
— `LogFile` : Le chemin vers le fichier de log pour ce domaine : `/var/log/nginx/domaine1-ac`
— `OutputDir` : mettez le dossier correspondant : `/var/www/webalizer-domaine1` par exemple
— `HostName` : le nom de domaine concerné.

Finalement, ajoutez ces lignes dans le fichier `/etc/logrotate.d/nginx` juste avant "prerotate"

```
webalizer -c /etc/webalizer/domaine1.conf >/dev/null 2>&1
webalizer -c /etc/webalizer/domaine2.conf >/dev/null 2>&1
webalizer -c /etc/webalizer/domaine3.conf >/dev/null 2>&1
```

5.10.2 Avec Piwik ⋆⋆⋆

Piwik va vous permettre de mettre un mouchard sur les pages de votre site afin d'obtenir des statistiques sur les visites reçues. Il nécessite l'utilisation de mysql (voir 7).



Piwik aura besoin des paquets suivants :

```
nginx php5-curl php5-gd php5-cli
php5-geoip php5-fpm php5-mysql mysql-server-5.5
```

Créez une nouvelle base de donnée pour piwik comme indiqué au 7.2.2.

Récupérez l'archive de piwik, puis déplacez-là où vous souhaitez, par exemple dans `/var/www/piwik`. On modifie les droits comme il faut pour finir :

```
wget 'http://builds.piwik.org/piwik.zip'
unzip piwik.zip
mv piwik /var/www/piwik
chown -R www-data:www-data /var/www/piwik
```

Visitez ensuite cette page pour obtenir une configuration de nginx complète et sécurisée : https://github.com/perusio/piwik-nginx/.

6 Sécuriser son serveur

6.1 Parefeu

+ SIMPLEMENT

Un parefeu, c'est un programme qui va faire la douane à l'entrée de votre serveur. Selon les règles que vous lui avez donné, il ne laissera passer les visiteurs que par les ports que vous avez ouvert.

Un parefeu bien configuré est important pour la sécurité de votre serveur. Le paquet arno-iptables-firewall est parfait pour faciliter cette opération. Lors de l'installation, il vous posera quelques questions pour configurer le parefeu. En cas de modifications ultérieures, modifiez le fichier

`/etc/etc/arno-iptables-firewall/conf.d/00debconf.conf`

Ce fichier ressemble à ceci :

```
EXT_IF="eth0"
EXT_IF_DHCP_IP=1
OPEN_TCP="22 443 587 80 25 143 993‘‘
OPEN_UDP=""
INT_IF="192.168.1.66"
NAT=0
INTERNAL_NET=""
NAT_INTERNAL_NET=""
OPEN_ICMP=0
```

La partie qui nous intéresse principalement, c'est les variables `OPEN_TCP` et `OPEN_UDP`, dans lesquelles on peut rajouter des numéros de ports.

Pensez juste à relancer le parefeu une fois vos modifications terminées :

```
# service arno-iptables-firewall restart
```

Allons plus loin

Par défaut, *iptables*, le parefeu de linux (enfin la surcouche à netfilter pour les puristes) laisse tout passer. C'est pourquoi il est important de tout fermer en entrée du serveur, puis d'ouvrir que ce que vous voulez rendre disponible au monde.
Il existe 2 types de ports : TCP et UDP. Il n'est pas nécessaire de bien comprendre la différence entre les 2. Il s'agit très grossièrement de la façon dont le serveur et le visiteur vont se parler.

6.2 Fail2ban

Fail2ban est sans doute le logiciel le plus important à avoir pour protéger votre serveur. C'est une sorte de gardien intelligent. Il va surveiller qui tente d'accéder à votre serveur, et si cette personne échoue plusieurs fois (par exemple à cause d'un mauvais mot de passe), alors elle est bannie et on ne l'écoute plus.

Fail2ban fonctionne avec ssh, mais aussi le serveur mail postfix et dovecot, ainsi que d'autres services comme le ftp.

Pour l'installer, c'est toujours aussi simple :

```
# apt-get install fail2ban
```

6.2.1 Mieux comprendre la configuration

La configuration de fail2ban se déroule dans le fichier `/etc/fail2ban/jail.local`.

Ce fichier est découpé en plusieurs sections que l'on repère facilement puisqu'elles sont entre crochets : `[section]`.

La première section est `[DEFAULT]` et permet de définir des valeurs qui seront utilisée par la suite, sauf si vous les modifiez dans les sections.

Voyons ce que signifient certaines options :

— `port` : permet de préciser sur quel port le service écoute. Par exemple, pour nginx, c'est le 80 ou `http`.

Allons plus loin

Pour connaître les ports utilisés par les services, vous pouvez consulter le fichier `/etc/services`

— `bantime` : C'est le nombre de secondes pendant lequel lest éventuels attaquants seront bannis. Évitez les chiffres ronds, puisque ce sont souvent des robots qui réalisent les attaques.

— `maxretry` : Nombre d'essais autorisés pendant le temps `findtime`

— `findtime` : Nombre de secondes pendant lequel on regarde si l'attaquant tente un nouvel assaut.

— `logpath` : l'emplacement du fichier de log (fichier qui enregistre l'activité du service)

Vous avez bien sûr la possibilité de créer vos propres filtres =, mais cela relève d'une documentation propre à fail2ban qui ferait l'objet d'une autre documentation. Vous pouvez toutefois visiter le site de fail2ban si le sujet vous intéresse.

Dans tous les cas, je vous conseille de jeter un œil au fichier `/etc/fail2ban/jail.conf` pour d'autres exemples.

6.2.2 Configuration rapide

Je vous propose ci-dessous une configuration déjà toute prête. Pour le fichier `/etc/fail2ban/jail.local` :

```
[DEFAULT]
ignoreip = 127.0.0.1/8
ignorecommand =
bantime  = 610
findtime = 600
maxretry = 3
backend = auto
usedns = warn
destemail = root@localhost
sendername = Fail2Ban
sender = fail2ban@localhost
banaction = iptables-multiport
mta = sendmail
protocol = tcp
action = %(action_mwl)s

[ssh]

enabled  = true
port  = ssh
filter   = sshd
logpath  = /var/log/auth.log
maxretry = 3

[pam-generic]

enabled  = true
filter   = pam-generic
port     = all
banaction = iptables-allports
port     = anyport
logpath  = /var/log/auth.log
maxretry = 3

[xinetd-fail]

enabled   = true
```

```
filter    = xinetd-fail
port      = all
banaction = iptables-multiport-log
logpath   = /var/log/daemon.log
maxretry  = 2

[ssh-ddos]

enabled  = true
port  = ssh
filter   = sshd-ddos
logpath  = /var/log/auth.log
maxretry = 3

[ssh-route]

enabled = false
filter = sshd
action = route
logpath = /var/log/sshd.log
maxretry = 6

[ssh-iptables-ipset4]

enabled  = false
port  = ssh
filter   = sshd
banaction = iptables-ipset-proto4
logpath  = /var/log/sshd.log
maxretry = 6

[ssh-iptables-ipset6]

enabled  = false
port  = ssh
filter   = sshd
banaction = iptables-ipset-proto6
logpath  = /var/log/sshd.log
maxretry = 6

[php-url-fopen]
```

```
enabled = false
port    = http,https
filter  = php-url-fopen
logpath = /var/www/*/logs/access_log

[nginx-http-auth]

enabled = true
filter  = nginx-http-auth
port    = http,https
logpath = /var/log/nginx/error.log

[vsftpd]

enabled  = false
port     = ftp,ftp-data,ftps,ftps-data
filter   = vsftpd
logpath  = /var/log/vsftpd.log
maxretry = 6

[postfix]

enabled  = true
port     = smtp,ssmtp,submission
filter   = postfix
logpath  = /var/log/mail.log

[sasl]

enabled  = true
port     = smtp,ssmtp,submission,imap2,imap3,imaps,pop3,pop3s
filter   = postfix-sasl
logpath  = /var/log/mail.log

[dovecot]

enabled = true
port    = smtp,ssmtp,submission,imap2,imap3,imaps,pop3,pop3s
filter  = dovecot
logpath = /var/log/mail.log
```

```
[ssh-blocklist]

enabled  = false
filter   = sshd
action   = iptables[name=SSH, port=ssh, protocol=tcp]
           sendmail-whois[name=SSH, dest="%(destemail)s", sender="%(sender)s", s
           blocklist_de[email="%(sender)s", apikey="xxxxxx", service="%(filter)s
logpath  = /var/log/sshd.log
maxretry = 20

[nginx-404]
enabled  = true
filter   = nginx-404
action   = iptables-multiport[name=nginx-404, port="http,https", protocol=tcp]
logpath = /var/log/nginx*/*error*.log
maxretry = 2
findtime  = 6
bantime  = 1200

[nginx-auth]
enabled = true
filter = nginx-auth
action = iptables-multiport[name=NoAuthFailures, port="http,https"]
logpath = /var/log/nginx*/*error*.log
bantime = 630
maxretry = 3

[nginx-login]
enabled = true
filter = nginx-login
action = iptables-multiport[name=NoLoginFailures, port="http,https"]
logpath = /var/log/nginx*/*error*.log
bantime = 630
maxretry = 3

[nginx-badbots]
enabled  = true
filter = apache-badbots
action = iptables-multiport[name=BadBots, port="http,https"]
logpath = /var/log/nginx*/*error*.log
```

```
bantime  = 87000
maxretry = 1

[nginx-noscript]
enabled = true
action = iptables-multiport[name=NoScript, port="http,https"]
filter = nginx-noscript
logpath = /var/log/nginx*/*error*.log
maxretry = 6
bantime  = 87000

[nginx-proxy]
enabled = true
action = iptables-multiport[name=NoProxy, port="http,https"]
filter = nginx-proxy
logpath = /var/log/nginx*/*error*.log
maxretry = 0
bantime  = 87000
```

La fin de cette configuration nécessite des filtres pour nginx qu'il faut ajouter dans `/etc/fail2ban/filter.d`.

— Fichier `nginx-404.conf`

```
[Definition]

failregex = <HOST> - - [.*?] ".*?" 4(0[0-9]|1[0-5])

ignoreregex =
```

— Fichier `nginx-proxy.conf`

```
# Proxy filter /etc/fail2ban/filter.d/nginx-proxy.conf:
#
# Block IPs trying to use server as proxy.
#
# Matches e.g.
# 192.168.1.1 - - "GET http://www.something.com/
#
[Definition]
failregex = ^<HOST> -.*GET http.*
ignoreregex =
```

— Fichier `nginx-noscript.conf`

```
# Noscript filter /etc/fail2ban/filter.d/nginx-noscript.conf:
#
# Block IPs trying to execute scripts such as .php, .pl, .exe and other fu
#
# Matches e.g.
# 192.168.1.1 - - "GET /something.php
#
[Definition]
failregex = ^<HOST> -.*GET.*(\.php|\.asp|\.exe|\.pl|\.cgi|\scgi)
ignoreregex =
```

— Fichier `nginx-auth.conf`

```
#
# Auth filter /etc/fail2ban/filter.d/nginx-auth.conf:
#
# Blocks IPs that fail to authenticate using basic authentication
#
[Definition]

failregex = no user/password was provided for basic \
authentication.*client: <HOST>
            user .* was not found in.*client: <HOST>
            user .* password mismatch.*client: <HOST>

ignoreregex =
```

— Fichier `nginx-login.conf`

```
# Login filter /etc/fail2ban/filter.d/nginx-login.conf:
#
# Blocks IPs that fail to authenticate
# using web application's log in page
#
# Scan access log for HTTP 200 + POST /sessions => failed log in
[Definition]
failregex = ^<HOST> -.*POST /sessions HTTP/1\.." 200
ignoreregex =
```

À la fin de la configuration, il faut recharger fail2ban :

```
# service fail2ban restart
```

6.3 Portsentry

Portsentry permet de se munir contre le scan des ports vulnérables sur votre serveur.

Installez le paquet `portsentry`.

Ensuite, on va procéder à la configuration. Au lieu de coller ci-dessous une grande quantité de texte, vérifier simplement que le contenu écrit est bien présent dans les fichiers donnés :

— Fichier `/etc/portsentry/portsentry.conf`

```
TCP_PORTS="1,11,15,79,111,119,143,540,635,1080,1524,2000,5742,6667,\
  12345,12346,20034,27665,31337,32771,\
  32772,32773,32774,40421,49724,54320"
UDP_PORTS="1,7,9,69,161,162,513,635,640,641,700,37444,34555,\
  31335,32770,32771,32772,32773,32774,31337,54321"
ADVANCED_PORTS_TCP="1024"
ADVANCED_PORTS_UDP="1024"
ADVANCED_EXCLUDE_TCP="113,139"
ADVANCED_EXCLUDE_UDP="520,138,137,67"
IGNORE_FILE="/etc/portsentry/portsentry.ignore"
HISTORY_FILE="/var/lib/portsentry/portsentry.history"
BLOCKED_FILE="/var/lib/portsentry/portsentry.blocked"
RESOLVE_HOST = "0"
BLOCK_UDP="1"
BLOCK_TCP="1"
KILL_ROUTE="/sbin/route add -host $TARGET$ reject"
KILL_HOSTS_DENY="ALL: $TARGET$ : DENY"
KILL_RUN_CMD="/sbin/iptables -I INPUT -s $TARGET$ -j DROP && /sbin/ipt
SCAN_TRIGGER="0"
```

— Fichier `/etc/default/portsentry` :

```
TCP_MODE="atcp"
UDP_MODE="audp"
```

Et pour finir, on recharge portsentry :

```
# service portsentry restart
```

7 Les bases de données

Une base de donnée permet à une application de retrouver rapidement des informations. Cela est particulièrement important lorsqu'il y a beaucoup de données reliées entre elles.

Par exemple, si on prend le cas d'un blog, alors les commentaires peuvent être stockés dans une base de donnée. Chaque commentaire est fait sur un certain article, par un visiteur donné, à une date précise. Le commentaire comme l'article ont un lien bien précis. L'utilisateur peut avoir donné son adresse e-mail pour être averti de nouveaux messages...

Vous l'aurez compris, toutes ces données s'entrecroisent, et il est plus efficace d'utiliser une base de donnée.

Cependant, ce n'est pas forcément obligatoire. Surtout sur un serveur auto-hébergé, où vous n'aurez sans doute pas des milliers d'utilisateurs.

Comprenez donc bien que si vous pouvez choisir des applications qui n'ont pas besoin de base de donnée, c'est un avantage pour vous car ça fait ça de moins à administrer, et ça de moins à sécuriser. Eh oui, car une base de donnée peut elle aussi subir des attaques.

Une alternative est d'utiliser dans ce cas SQLite, puisque cette base de donnée ne nécessite pas d'administration particulière, c'est l'application qui se chargera de tout.

7.1 SQlite

SQLite est un moteur de base de donnée tout simplement génial.

Vous n'avez rien de particulier à faire pour l'administrer, c'est l'application qui en a besoin qui se chargera de créer la base. En plus, c'est très facile à sauvegarder puisque c'est dans ce cas un simple fichier. Enfin, ce moteur sait se montrer léger et fonctionne bien même sur du matériel moins puissant.

Alors certains diront que ce n'est pas le moteur le plus performant. C'est vrai. Mais à moins d'avoir des milliers de visiteurs sur votre site, vous ne verrez pas la différence et eux non plus. N'hésitez pas, il y a plus d'avantages que d'inconvénients à utiliser SQLite en auto-hébergement.

7.2 MySQL

MySQL est un autre moteur de base données, sans doute le plus répandu. Il est distribué sous une licence libre et sous une licence propriétaire (vous êtes prévenu).

Puisqu'une fois installé, MySQL sera lancé en arrière-plan, il faudra vous renseigner sur la sécurisation de ce service, en particulier ajouter à fail2ban la surveillance de MySQL (voir 6.2)

7.2.1 Installation de MySQL

Pour installer MySQL, il faut installer le paquet `mysql-server-5.5`. Si c'est une application en php qui a besoin de mysql, alors il faudra aussi le paquet `php5-mysql`.

Lors de son installation, il vous sera demandé un mot de passe administrateur. Ce mot de passe est à bien retenir, puisqu'il vous sera nécessaire ensuite pour créer de nouvelles bases.

7.2.2 Gérer MySQL

Pour la suite, vous pourrez administrer mysql après avoir entré la commande suivante :

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

Une fois vos modifications effectuées, vous pourrez quitter avec la combinaison de touches ctrl+d.

Créer une nouvelle base de données

```
mysql> CREATE DATABASE nom_de_la_nouvelle_base;
```

Créer un utilisateur de mysql

```
mysql> CREATE USER 'utilisateur'@'localhost' IDENTIFIED BY 'motdepasse';
```

Remplacez bien sûr `utilisateur` et `motdepasse`.

Modifier les droits

```
mysql> GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, ALTER,
    CREATE TEMPORARY TABLES, LOCK TABLES ON nom_de_la_base.* TO
    'utilisateur'@'localhost';
```

Quelques explications :

- `GRANT` : donne les droits suivants
- `SELECT, INSERT...` : différentes actions sur la base de donnée
- `ON nom_de_la_base` : ces droits sont données sur la base qui porte le nom `nom_de_la_base`
- `TO 'utilisateur'@'localhost'` : les droits sont données à l'utilisateur sur la machine locale

7.3 PostgreSQL

PostgreSQL est un autre moteur de base de données, entièrement libre.

7.3.1 Installation de PostgreSQL

Pour installer PostgreSQL, il faut installer les paquets suivants :

```
# apt-get install postgresql postgresql-client postgresql-client-common
```

Et pour qu'une application php puisse accéder au moteur postgreSQL, installez le paquet `php5-pgsql`

7.3.2 Gérer PostgreSQL

Pour se connecter à postgreysql, on utilise la commande `su postgres -c psql`
.

Voici quelques commandes permettant de gérer PostgreSQL.

Modifier le mot de passe administrateur

```
su postgres -c psql ALTER USER postgres WITH PASSWORD 'mot_de_passe';
```

Ajouter un utilisateur à la base

```
su postgres -c psql CREATE USER 'nouvelutilisateur'
WITH PASSWORD 'mot_de_passe_de_l_utilisateur';
```

Créer une nouvelle base

```
su postgres -c psql << EOF
\connect template1
CREATE DATABASE "nom_de_base" WITH ENCODING 'UTF-8';
GRANT ALL PRIVILEGES ON DATABASE "nom_de_base" TO "utilisateur";
ALTER DATABASE "nom_de_base" OWNER TO "utilisateur";
\q
```

8 Divers

8.1 Script d'installation

Vous pourrez trouver un script appelé hostathome qui permet de faciliter l'installation des différents services pour s'autohéberger. Pour cela, il suffit de suivre le lien suivant :

http://yeuxdelibad.net/Programmation/Hostathome.html

8.2 Recevoir un mail lorsque quelqu'un se connecte

Pour être averti dès que quelqu'un se connecte en ssh, créez le script suivant à enregistrer sous `/etc/ssh/sshrc` :

```
#!/bin/sh
# source: http://blog.uggy.org/post/2009/06/05/...
DATE=‘date "+%d.%m.%Y--%Hh%Mm"‘
IP=‘echo $SSH_CONNECTION | awk ’{print $1}’‘
REVERSE=‘dig -x $IP +short‘
echo "Connexion de $USER sur $HOSTNAME
IP: $IP
ReverseDNS: $REVERSE
Date: $DATE

" | mail -s "Connexion de $USER sur $HOSTNAME" moi@laposte.net
```

8.3 Foire aux questions

— *Il paraît que pour un serveur, on peut utiliser un vieux PC. Mais quelle taille de disque dur est nécessaire au minimum ?*
Ça dépend. La réponse ne sera pas la même selon ce que vous voulez faire du serveur.
Pour un petit site web, quelques Go suffiront amplement.
Si c'est un site contenant images et vidéos à gogo, alors il faudra un espace plus conséquent.
Si c'est pour faire une seedbox ou un mediacenter, alors il faut compter encore plus.
Dans la majorité des cas, entre 10G et 20G seront bien assez.

— *Et pour le processeur, quelle puissance au moins ?*
Là aussi, ça dépend ☺. Pour un simple site avec quelques visites par jour, peu de puissance est nécessaire.
S'il y a du php, il faut alors une puissance un peu plus élevée.
Enfin, si le serveur propose de la messagerie instantanée, un site avec php, des mails... Vous l'aurez compris, il faudra encore une puissance plus grande.
Il en va de même pour la mémoire vive d'ailleurs.
Notez qu'un simple raspberry pi, donc très peu puissant, est souvent assez pour un serveur auto-hébergé.

— *Admettons que on site soit sur serveur-à-moi , comment est-il visible sur internet par d'autres personnes ?* Lorsque quelqu'un tape l'adresse

de votre site dans son navigateur, c'est traduit en une série de chiffres qui permet de retrouver votre serveur. Il s'agit du fonctionnement des DNS, expliqué au paragraphe 3.3.

8.4 Notes à propos du raspberry pi

Un raspberrypi est une minuscule machine très peu gourmande en électricité.. C'est certainement une des solutions les plus économiques, que ce soit du point de vue matériel, prix, que consommation.

Pour l'utiliser comme serveur, vous aurez besoin :

— Un raspberry pi
— Une carte SD
— Un cable ethernet.
— Un disque dur externe qui a sa propre alimentation électrique (si votre serveur sert de stockage)

8.4.1 Raspbian

Raspbian est une distribution linux dérivée de debian et optimisée pour le pi. Je vous la recommande. Vous pouvez télécharger l'image de raspbian sur ce site : http://www.raspberrypi.org/downloads.

À partir d'une distribution linux, elle se copie sur la carte SD avec cette simple commande :

```
dd if=/chemin/vers/raspbian.img of=/dev/sdb
```

À condition que /dev/sdb représente bien votre carte SD. Pour être sûr, tapez la commande `dmesg` juste après avoir branché la carte SD, ou bien `fdisk -l` pour lister tous les périphériques branchés.

Voyons maintenant comment obtenir une installation minimale de raspbian.

8.4.2 Installateur minimal ua-netinst (méthode 1)

Un installateur est disponible pour avoir une raspbian minimale. C'est la méthode que je trouve la meilleure pour avoir une installation de raspbian la plus légère possible.

L'installateur se trouve ici : https://github.com/debian-pi/raspbian-ua-netinst/releases/latest.

Une fois une des archive téléchargée, voici la marche à suivre pour l'installer :

1. Décompresser l'archive : `unzx raspbian-ua-netinst-v1.0.7.Img.xz`

2. Copier l'image sur la carte SD : `dd if=raspbian-ua-netinst-v1.0.7.Img of=/dev/sdX` . Remplacez /dev/sdX par le nom de votre carte SD comme indiqué dans la section 8.4.1.
3. Insérez la carte SD dans le raspberry pi, puis branchez le cable ethernet. Enfin, allumez-le. L'installation se déroule tranquilement.
4. Au premier démarrage, modifiez le mot de passe du compte root avec la commande `passwd`.

   Par défaut, le mot de passe de **root** est **raspbian**.
5. Changez la langue par défaut : `dpkg-reconfigure locales`
6. Changez la zone horaire : `dpkg-reconfigure tzdata`
7. Lisez le README https://github.com/debian-pi/raspbian-ua-netinst/blob/master/READ

Et voilà !

8.4.3 Préparation de l'image avec qemu (méthode 2)

Si on souhaite utiliser le raspberry pi comme serveur, il faut se débarrasser de tout ce qui est inutile.

Si vous n'avez pas d'écran pour configurer raspbian après l'avoir mis sur la carte SD, ou si vous voulez préparer la distribution avant de la copier, vous pouvez utiliser *qemu*.

Cet outil va démarrer sur l'image de raspbian, comme si elle était installée sur le pi, avant de copier l'image modifiée sur la carte SD du RPi.

Vous aurez besoin de télécharger un kernel spécifique à qemu.

Installez qemu-system sur votre distribution linux, puis lancez l'image de raspbian avec cette commande :

```
qemu-system-arm -kernel kernel-3.6.8-armhf-qemu -cpu \
    arm1176 -m 512 -M versatilepb -no-reboot -serial stdio\
    -append "root=/dev/sda2 panic=1" -hda NOMDELIMAGE.img
```

Si ça ne fonctionne pas, essayer avec celle-ci, qui ne lance pas d'interface graphique :

```
qemu-system-arm -M versatilepb -cpu arm1176 -hda NOMDELIMAGE.img \
-nographic -no-reboot -kernel kernel-3.6.8-armhf-qemu \
-append "root=/dev/sda2 panic=1 console=ttyAMA0"
```

Connectez-vous avec les identifiants suivants :
— login : pi
— password : raspberry

Ensuite, vous pourrez alors vous débarasser de tout ce qui vous est inutile (lxde, pcmanfm, dillo, openbox, idle ...), et installer ssh avec la commande `apt-get` bien sûr.

Pour arrêter, tapez `halt`.

Vous obtenez ainsi une image de rasbian personnalisée, qui peut être copiée sur la carte SD.

8.4.4 Faire le ménage

Pour retirer une bonne partie des paquets inutiles pour un serveur, ces commandes feront l'affaire :

```
# apt-get remove --purge desktop-base \
x11-common midori omxplayer scratch dillo xpdf galculator \
netsurf-common netsurf-gtk lxde-common lxde-icon-theme \
hicolor-icon-theme libpoppler19 ed lxsession lxappearance lxpolkit \
lxrandr lxsession-edit lxshortcut lxtask lxterminal xauth \
debian-reference-common fontconfig fontconfig-config \
fonts-freefont-ttf wolfram-engine dbus-x11 desktop-file-utils \
libxmuu1
# apt-get autoremove
# rm -rf /opt/* \
/usr/share/icons/* \
/usr/games \
/usr/share/squeak \
/usr/share/sounds \
/usr/share/wallpapers \
/usr/share/themes \
/usr/share/kde4 \
/usr/share/images/* \
/home/pi/python_games
```

8.4.5 Augmenter la taille de l'image avec qemu-img

Si vous ajoutez de nombreux paquets à l'image ci-dessus, elle sera sûrement trop petite. Pour résoudre ce problème, cette commande fera l'affaire :

```
qemu-img resize raspbian.img +2GB
```

Ensuite, démarrer l'image avec qemu, puis connectez-vous avec le compte pi. On va maintenant copier l'utilitaire raspi-config pour le modifier légèrement. Lancez ces quelques commandes :

```
cp /usr/bin/raspi-config ~
sed -i 's/mmcblk0p2/sda2/' ~/raspi-config
sed -i 's/mmcblk0/sda/' ~/raspi-config
sudo ~/raspi-config
```

Choisissez `expand_rootfs`. Redémarrez, et regardez le résultat en tapant `df`.

8.5 Surveiller votre serveur avec Logwatch

Logwatch vous permettra de recevoir des mails quotidiens sur l'état de votre serveur. Il se configure par le fichier `/usr/share/logwatch/default.conf/logwatch.conf`

9 Références

Voici quelques références ayant servi à la rédaction de ce document, que je vous conseille de lire en supplément.

— Hostathome, script pour faciliter l'auto-hébergement `http://yeuxdelibad.net/Programmation/Hostathome.html`
— Le wiki sur l'auto-hébergement `http://www.auto-hebergement.fr/`
— Un annuaire d'applications et de services à auto-héberger `http://waah.quent1.fr/doku.php`
— Une distribution pour l'auto-hébergement `https://yunohost.org`
— calomel.org
— Screenshots de squirrelmail https://squirrelmail.org/screenshots.php
— icecast `http://icecast.org/`
— idjc pour faire des podcasts `http://idjc.sourceforge.net/`