Installation et configuration d’un serveur mail sur raspberry pi3

Sommaire

Pré -requis :

Pour réaliser cette installation, j'ai acheté le nom de domaine « jmichaud.eu » sur OVH.
De plus, ce tuto est très fortement basé sur celui de Sam Hobbs
https://samhobbs.co.uk/raspberry-pi-email-server
Pour mon cwebmail, j'ai utilisé ce tutorial https://www.linuxbabe.com/mail-server/install-rainloop-webmail-ubuntu-16-04

J'utilise une raspberry pi3 avec une carte sd de 32go et la distribution raspbian 4.9.56

1- Installation et configuration de Postfix

Postfix est un serveur de messagerie électronique et un logiciel libre développé par Wietse Venema et plusieurs contributeurs.
Il se charge de la livraison de courriers électroniques (courriels) et a été conçu comme une alternative plus rapide, plus facile à administrer et plus sécurisée que l'historique Sendmail.
Il est le serveur de courriel par défaut dans plusieurs systèmes de type UNIX, comme Mac OS X, NetBSD2, diverses distributions GNU/Linux, etc. Via Wikipédia

On part du principe que vous effectuez l'installation sur un système vierge et à jour.

Pour installer postfix :

```
sudo apt-get update
sudo apt-get install postfix
```

Un menu avec des choix va apparaître, sélectionnez « **Internet Site** » et indiquez votre domaine, dans mon cas « **jmichaud.eu** ».

Déplacez vous dans le dossier **/etc/postfix** et éditez le fichier **main.cf** pour y ajouter `inet_protocols = ipv4` à la fin.

Assurez-vous que votre hostname est bien paramétré dans **/etc/postfix/main.cf**.
Regardez la ligne `myhostname =` et assurez vous que votre FQDN est indiqué.

Redémarrez postfix `sudo service postfix restart`

Pour la configuration des boites mails, j'ai choisi (ou plutôt Sam Hobbs)d'utiliser la configuration MailDir qui permet à chaque compte UNIX présent sur ma raspberry de bénéficier d'une boite Mail.
Maildir va créer un dossier dans le dossier **home** de chaque utilisateur pour y mettre ces mails. Un mail sera représenté par un fichier.

Pour cela, dans le fichier /etc/postfix/main.cf, il faudra modifier les lignes suivantes :

Il faut maintenant créer le dossier mail et ces sous dossiers pour les utilisateurs existants mais aussi rentre leurs créations automatiques.
Pour faire ça, il faut utiliser des commandes présentes dans le paquet Dovecot. (J'explique Dovecot dans la deuxième partie de ce tuto)

Pour installer Dovecot :

```
sudo apt-get install dovecot-common dovecot-imapd
```

Pour créer les dossiers et rendre leurs créations automatiques :

```
sudo maildirmake.dovecot /etc/skel/Maildir
sudo maildirmake.dovecot /etc/skel/Maildir/.Drafts
sudo maildirmake.dovecot /etc/skel/Maildir/.Sent
sudo maildirmake.dovecot /etc/skel/Maildir/.Spam
sudo maildirmake.dovecot /etc/skel/Maildir/.Trash
sudo maildirmake.dovecot /etc/skel/Maildir/.Templates
```

Maintenant, copiez ces fichiers vers les répertoires home des utilisateurs déjà crées et changez les permissions :

```
sudo cp -r /etc/skel/Maildir /home/USER/
sudo chown -R USER:USER /home/USER/Maildir
sudo chmod -R 700 /home/USER/Maildir
```

Pour tester ce que nous avons fait, on va utiliser telnet.

```
sudo apt-get install telnet
```

Depuis votre rasp, connectez-vous au port 25 :

```
telnet localhost 25
```

Pour tester l'envoi de mail, voici les étapes :

1. Envoyez un **ehlo** pour dire au serveur qui vous êtes.
2. Utilisez **mail** from pour dire de qui proviens le mail. Si c'est un utilisateur qui est sur la rasp, rentrez juste son nom.
3. La commande **rcpt** indique à qui envoyer le mail.
4. **Data** indique au serveur que vous vous apprêter à envoyer un message.
5. **Subject :** indique…le sujet !
6. Ensuite entrez le message à envoyer
7. **Quit** permet de quitter.

Un exemple :

```
admin@samhobbs /etc/postfix $ telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 samhobbs.co.uk ESMTP Postfix (Debian/GNU)
ehlo samhobbs.co.uk
250-samhobbs
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
mail from: USER
250 2.1.0 Ok
rcpt to: me@externalemail.com
554 5.7.1 <me@externalemail.com>: Relay access denied
quit
221 2.0.0 Bye
Connection closed by foreign host.
```

Pour restreindre les personnes qui peuvent envoyer des mails à des serveurs mails externes, ajoutez les lignes suivantes au fichier **/etc/postfix/main.cf** :

```
smtpd_recipient_restrictions =
  permit_sasl_authenticated,
  permit_mynetworks,
  reject_unauth_destination
```

Rechargez Postfix :

```
sudo service postfix reload
```

Maintenant, il faut empêchez les spammers de nous envoyer des mails.

Ajoutez ces lignes au fichier **/etc/postfix/main.cf:**

```
smtpd_helo_required = yes

smtpd_helo_restrictions =

      permit_mynetworks,

      permit_sasl_authenticated,

      reject_invalid_helo_hostname,

      reject_non_fqdn_helo_hostname,

      reject_unknown_helo_hostname
```

Pour savoir exactement ce à quoi servent ces lignes, je vous invite à visiter le site de Sam Hobbs pour plus d'explications.

Maintenant, il faut bloquer les personnes utilisant notre nom de domaine.

Ajoutez ces lignes toujours au même fichier :

```
smtpd_helo_restrictions =
        permit_mynetworks,
        permit_sasl_authenticated,
        reject_invalid_helo_hostname,
        reject_non_fqdn_helo_hostname,
        reject_unknown_helo_hostname,
        check_helo_access hash:/etc/postfix/helo_access
```

Créer le fichier suivant:

```
sudo nano /etc/postfix/helo_access
```

Et ajoutez les lignes suivantes à adapter avec votre nom de domaine :

```
samhobbs.co.uk          REJECT          Get lost - you're lying about who you are
mail.samhobbs.co.uk     REJECT          Get lost - you're lying about who you are
```

Ensuitez mapez ce fichier à Postfix:

```
sudo postmap /etc/postfix/helo_access
sudo service postfix restart
```

2. Installation et configuration de Dovecot:

Dovecot est un serveur IMAP et POP3 pour les systèmes d'exploitation Unix et dérivés, conçu avec comme premier but la sécurité. Dovecot est distribué en double licence MIT et GPL version 2 via Wikipedia

Dans le fichier **/etc/dovecot/dovecot.conf** , modifiez la ligne listen pour listen = *

Maintenant il faut ouvrir le fichier **/etc/dovecot/conf.d/10-mail.conf** pour dire à Dovecot ou se trouve le dossier Mail de chaque utilisateur et changez la ligne **mail_location** par **mail_location = maildir:~Maildir**

Dire à Postfix d'utilier Dovecot pour l'authentification SASL.
Ouvrez le fichier **/etc/postfix/main.cf** et ajoutez les lignes suivantes:

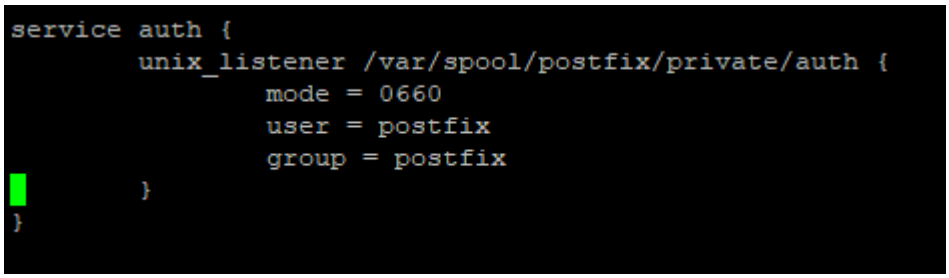
```
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth
smtpd_sasl_auth_enable = yes
```

Ensuite, il faut indiquer à Dovecot d'écouter les demandes d'authentification SASL de Postfix.

Ouvrez le fichier **/etc/dovecot/conf.d/10-master.conf** et commentez le block comment par **service auth** en plaçant un # au début de chaque ligne.

Remplacez ce block avec celui-ci:

```
service auth {
        unix_listener /var/spool/postfix/private/auth {
                mode = 0660
                user = postfix
                group = postfix
        }
}
```



Activez l'authentification en claire en modifiant/ajoutant les lignes suivantes au fichier **/etc/dovecot/conf.d/10-auth.conf** :

```
disable_plaintext_auth = no
auth_mechanisms = plain login
```

On utilisera une connexion SSL/TLS pour se connecter à sa boite mail.

Configuration pour n'authoriser qu'une authentification cryptée:

Dire à Postfix d'écouter sur le port 465:
Dans le fichier **/etc/postfix/master.cf** et décommentez la ligne:

```
smtps     inet  n       -       -       -       -       smtpd
```

Redémarrez Postfix ensuite.

Maintenant Postfix écoute sur le port 465 MAIS autorise les connections non cryptées, pour interdire les connections non cryptées il faut décommentez certaines lignes :

```
smtps     inet  n       -       y       -       -       smtpd
  -o syslog_name=postfix/smtps
  -o smtpd_tls_wrappermode=yes
```

Redémarrez Postfix.

Ajoutez la ligne suivante au fichier **/etc/postfix/main.cf** :

```
smtpd_tls_auth_only = yes
```

Redémarrez Postfix.

Maintenant, modifiez ces lignes pour Postifx n'accepte pas de messages provenant d'utilisateurs non authentifiés:
Dans le fichier /etc/postfix/master.cf:

```
smtps     inet  n       -       y       -       -       smtpd
  -o syslog_name=postfix/smtps
  -o smtpd_tls_wrappermode=yes
#  -o smtpd_sasl_auth_enable=yes
#  -o smtpd_reject_unlisted_recipient=no
#  -o smtpd_client_restrictions=$mua_client_restrictions
#  -o smtpd_helo_restrictions=$mua_helo_restrictions
#  -o smtpd_sender_restrictions=$mua_sender_restrictions
   -o smtpd_recipient_restrictions=permit_sasl_authenticated,reject
```

Pour tester, voici un exemple: remplacez les logins par les vôtres:

```
telnet localhost 143
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
* OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR LOGIN-REFERRALS ID ENABLE IDLE STARTTLS AUTH=PLAIN
AUTH=LOGIN] Dovecot ready.
a login "testmail" "test1234"
a OK [CAPABILITY IMAP4rev1 LITERAL+ SASL-IR LOGIN-REFERRALS ID ENABLE IDLE SORT SORT=DISPLAY
THREAD=REFERENCES THREAD=REFS MULTIAPPEND UNSELECT CHILDREN NAMESPACE UIDPLUS LIST-EXTENDED I18NLEVEL=1
CONDSTORE QRESYNC ESEARCH ESORT SEARCHRES WITHIN CONTEXT=SEARCH LIST-STATUS SPECIAL-USE] Logged in
b select inbox
* FLAGS (\Answered \Flagged \Deleted \Seen \Draft)
* OK [PERMANENTFLAGS (\Answered \Flagged \Deleted \Seen \Draft \*)] Flags permitted.
* 1 EXISTS
* 0 RECENT
* OK [UNSEEN 1] First unseen.
* OK [UIDVALIDITY 1385217480] UIDs valid
* OK [UIDNEXT 2] Predicted next UID
* OK [NOMODSEQ] No permanent modsequences
b OK [READ-WRITE] Select completed.
c logout
* BYE Logging out
c OK Logout completed.
Connection closed by foreign host.
```

Ajouter le support TLS:

IMAP marche, mais maintenant il faut faire marcher IMAPS !

Ouvrez le fichier **/etc/dovecot/conf.d/10-master.conf** et décommentez le bloque **service imap-login**

```
service imap-login {
  inet_listener imap {
    port = 143
  }
  inet_listener imaps {
    port = 993
    ssl = yes
  }
}
```

Activez le SSL dans la configuration Dovecot:
Dans le fichier **/etc/dovecot/conf.d/10-ssl.conf** trouvez la ligne **ssl et** mettez **ssl = yes**

Dans le même fichier, trouvez la ligne ssl_protocols et desactivez certaines version de SSL qui présentes des failles importantes:

ssl_protocols = !SSLv3

Maintenant, il faut créer les certificats pour la connexion TLS.
On se servira de certificats auto-signés.

Déplacez vous dans le dossier suivant **cd /usr/share/dovecot** et éxécutez le script présent dans ce dossier **./mkcert.sh**

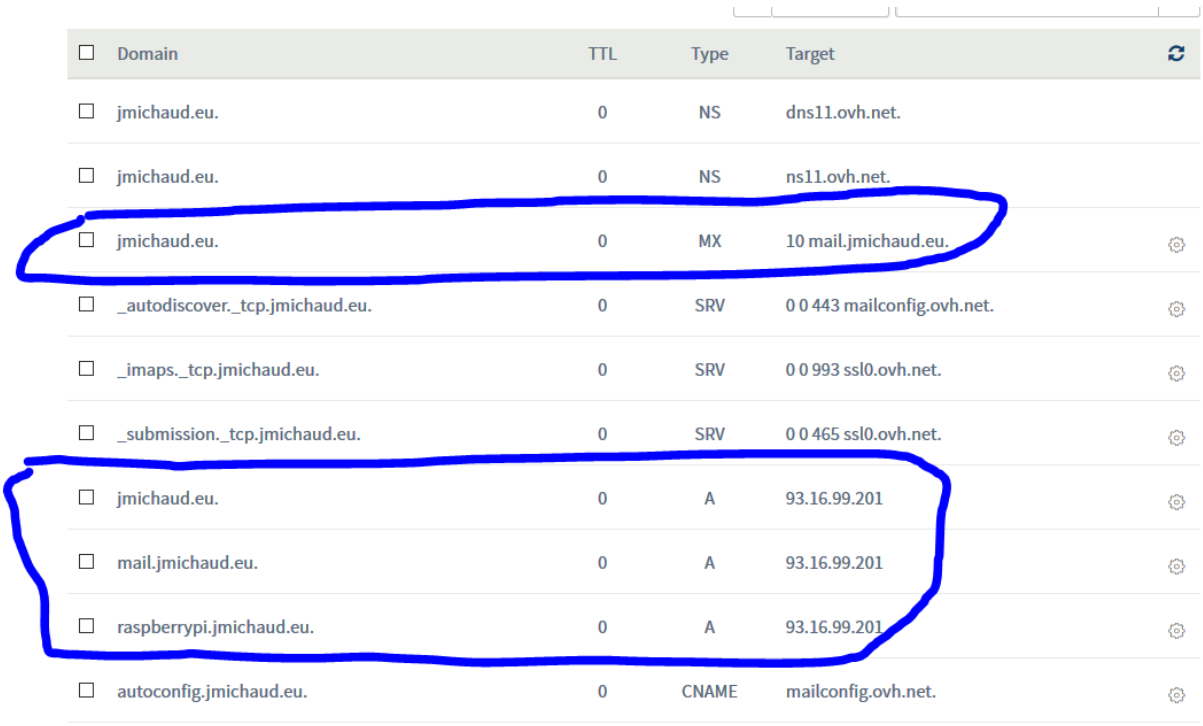
Maintenant, il faut dire à Dovecot ou se trouvent ces certificats, ouvrez le fichier **/etc/dovecot/conf.d/10-ssl.conf** et décommentez les lignes suivantes:

```
#ssl_cert = </etc/dovecot/dovecot.pem
#ssl_key = </etc/dovecot/private/dovecot.pem
```

Rechargez Dovecot sudo service dovecot reload

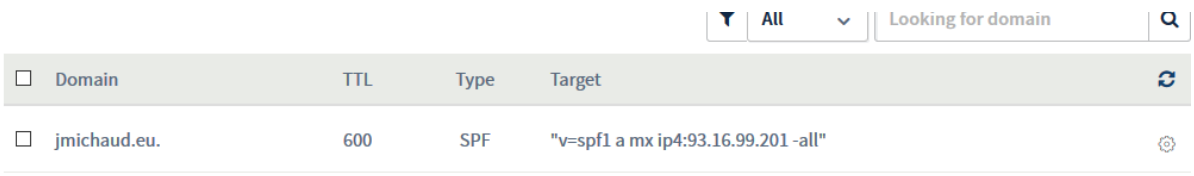
Configuration des entrées DNS:

Sur mon panneau de configuration OVH, j'ai ajouté les entrées suivantes:



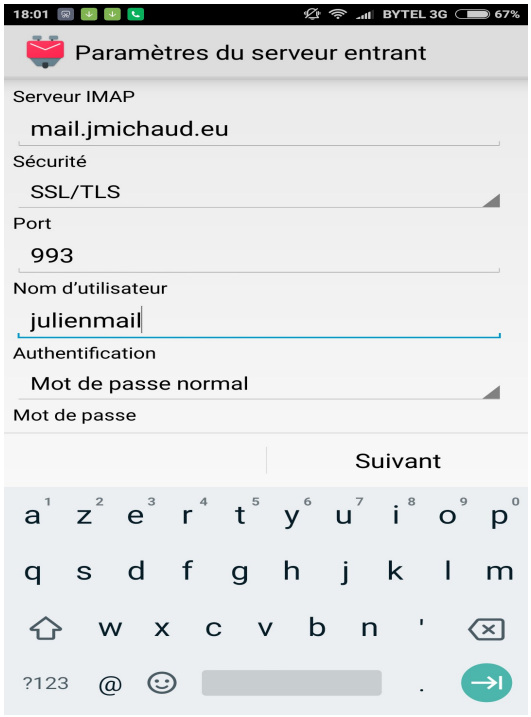
Domain	TTL	Type	Target
jmichaud.eu.	0	NS	dns11.ovh.net.
jmichaud.eu.	0	NS	ns11.ovh.net.
jmichaud.eu.	0	MX	10 mail.jmichaud.eu.
_autodiscover._tcp.jmichaud.eu.	0	SRV	0 0 443 mailconfig.ovh.net.
_imaps._tcp.jmichaud.eu.	0	SRV	0 0 993 ssl0.ovh.net.
_submission._tcp.jmichaud.eu.	0	SRV	0 0 465 ssl0.ovh.net.
jmichaud.eu.	0	A	93.16.99.201
mail.jmichaud.eu.	0	A	93.16.99.201
raspberrypi.jmichaud.eu.	0	A	93.16.99.201
autoconfig.jmichaud.eu.	0	CNAME	mailconfig.ovh.net.

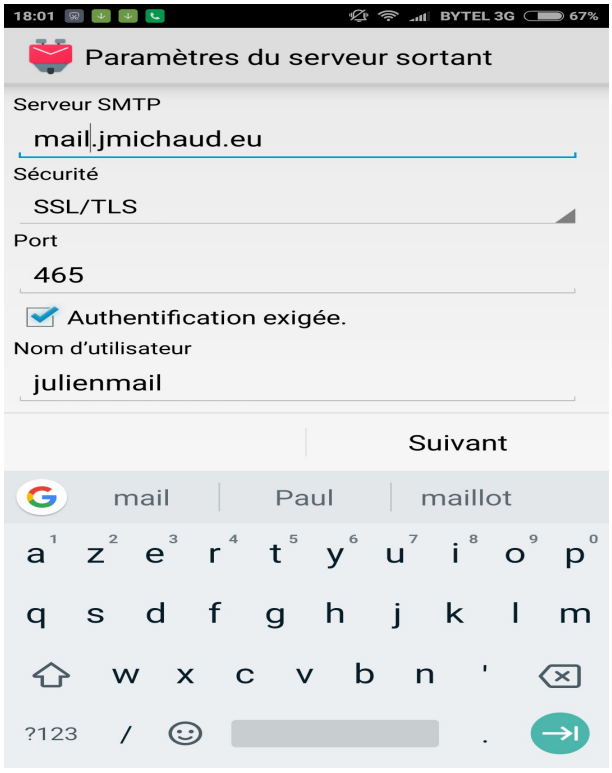
Configurez une entrée SPF:



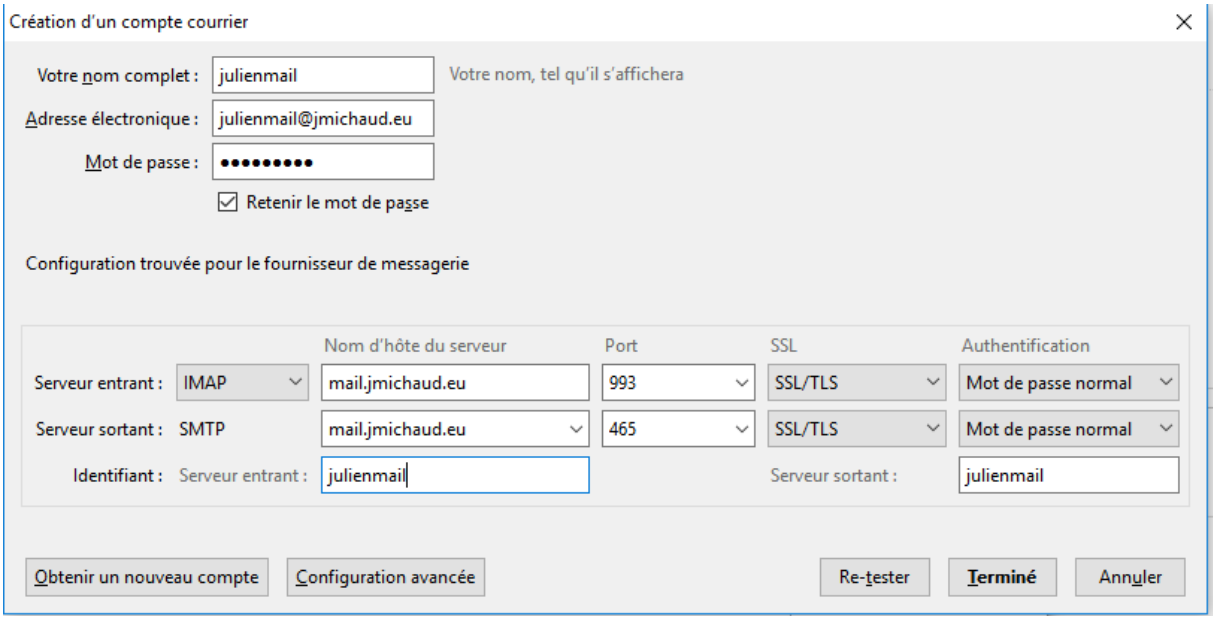
Domain	TTL	Type	Target
jmichaud.eu.	600	SPF	"v=spf1 a mx ip4:93.16.99.201 -all"

Avec un client mail Android comme K9 Mail j'ai rentrer les informations suivantes pour acceder à ma boite mail:


18:01
BYTEL 3G
67%
Paramètres du serveur entrant
Serveur IMAP
mail.jmichaud.eu
Sécurité
SSL/TLS
Port
993
Nom d'utilisateur
julienmail
Authentification
Mot de passe normal
Mot de passe
Suivant


18:01
BYTEL 3G
67%
Paramètres du serveur sortant
Serveur SMTP
mail.jmichaud.eu
Sécurité
SSL/TLS
Port
465
Authentification exigée.
Nom d'utilisateur
julienmail
Suivant
mail
Paul
maillot

Sur un client lourd comme Thunderbird:



3. Configuration d'un Webmail:

Installez apache et php:

```
sudo apt install apache2 php7.0 libapache2-mod-php7.0
```

Installez ensuite les extensions php requis pour Rainloop:

```
ssudo apt install php7.0-curl php7.0-xml
```

Maintenant, créez un dossier rainloop dans votre home et placez vous dedans:

Executez cette commande pour installer la derniere version de Rainloop:

```
curl -s http://repository.rainloop.net/installer.php | php
```

Ensuite faites un **cd ..** et mv rainloop **/var/www/html/**

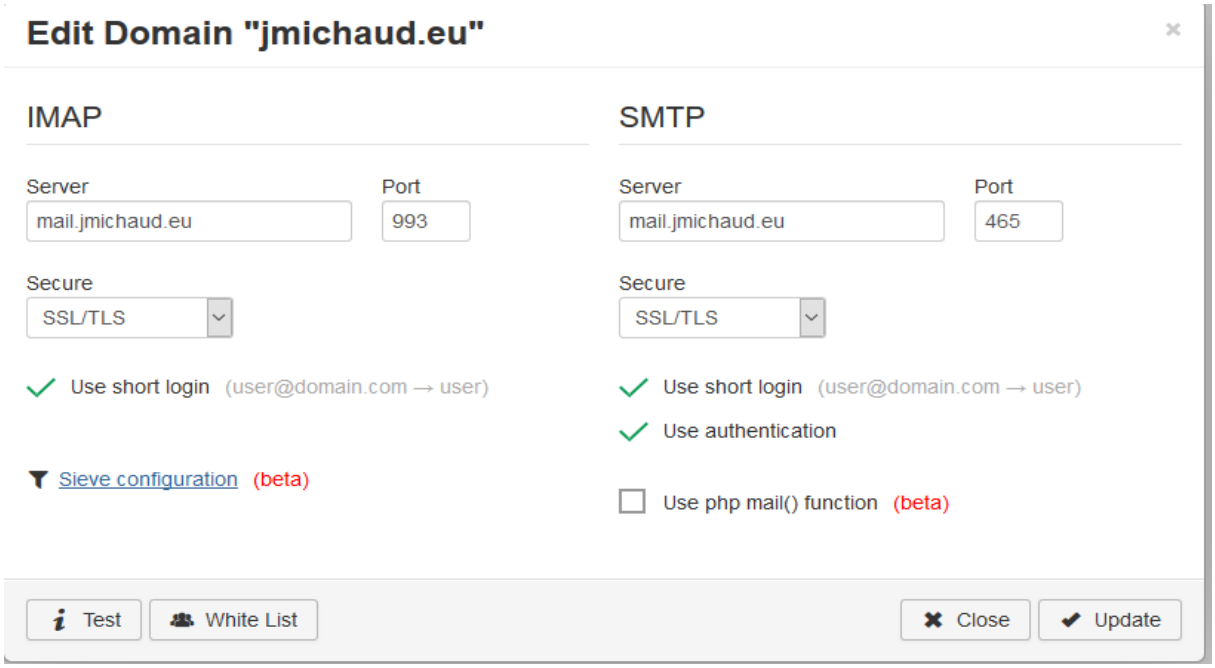
Définiseez l'utilisateur www-data comme proprietaire du dossier:

```
sudo chown www-data:www-data /var/www/html/rainloop/ -R
```

Ensuite, rendez-vous sur l'ip de votre rasp suivi de /rainloop/?admin
xx.xx.xx.xx/rainloop/?admin

Les logins par défauts sont admin/12345

Une fois connecté, allez dans Domains et ajouter le votre:



Une fois fait, rendez-vous dans /etc/apache2/sites-available pour créer le VirtualHost de Rainloop:

```
# Apache2 vhost configuration sample for Roundcube
# https://linode.com/content/email/clients/installing-roundcube-on-ubuntu-14-04/

<VirtualHost *:80>
  # Virtual host configuration + information (replicate changes to *:443 below)
  ServerAdmin julienmail@jmichaud.eu
  ServerName webmail.jmichaud.eu
  DocumentRoot /var/www/html/rainloop
  # ErrorLog /var/log/apache2/webmail.jmichaud.eu/error.log
  # CustomLog /var/log/apache2/webmail.jmichaud.eu/access.log combined

  # Permanently redirect all HTTP requests to HTTPS
  RewriteEngine on
  RewriteCond %{SERVER_PORT} !^443$
  RewriteRule ^/(.*) https://%{HTTP_HOST}/$1 [NC,R=301,L]
</VirtualHost>

<IfModule mod_ssl.c>
<VirtualHost *:443>
  # Virtual host configuration + information (replicate changes to *:80 above)
  ServerAdmin julienmail@hjmichaud.eu
  ServerName webmail.jmichaud.eu
  DocumentRoot /var/www/html/rainloop
   ErrorLog /var/log/apache2/webmail.jmichaud.eu/error.log
   CustomLog /var/log/apache2/webmail.jmichaud.eu/access.log combined

  # SSL certificate + engine configuration
  SSLEngine on
  SSLCertificateFile /etc/letsencrypt/live/webmail.jmichaud.eu/fullchain.pem
  SSLCertificateKeyFile /etc/letsencrypt/live/webmail.jmichaud.eu/privkey.pem

  # Roundcube directory permissions + restrictions
   <Directory />
    Options +Indexes +FollowSymLinks +ExecCGI
    AllowOverride All
    Order deny,allow
    Allow from all
    Require all granted
</Directory>
</VirtualHost>
</IfModule>
```

Ce Virtualhost redirigera les requête http en https.
Faites attention à votre **ServerName,** il servira pour joindre votre site depuis Internet

Maintenant, on peut signer son site avec Let's Encrypt.

Téléchargez le client certbot:

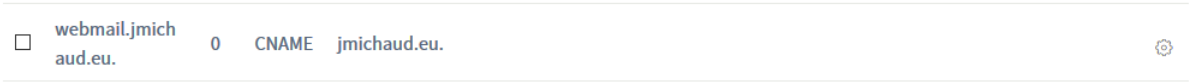
```
sudo apt install software-properties-common

sudo add-apt-repository ppa:certbot/certbot

sudo apt update

sudo apt install certbot python-certbot-apache
```

Ensuite, executez cette commande en remplacant ce qui est en rouge par vos infos, mon ServerName se nomme webmail.jmichaud.eu donc le “mail.example.com” sera remplacé par ca.

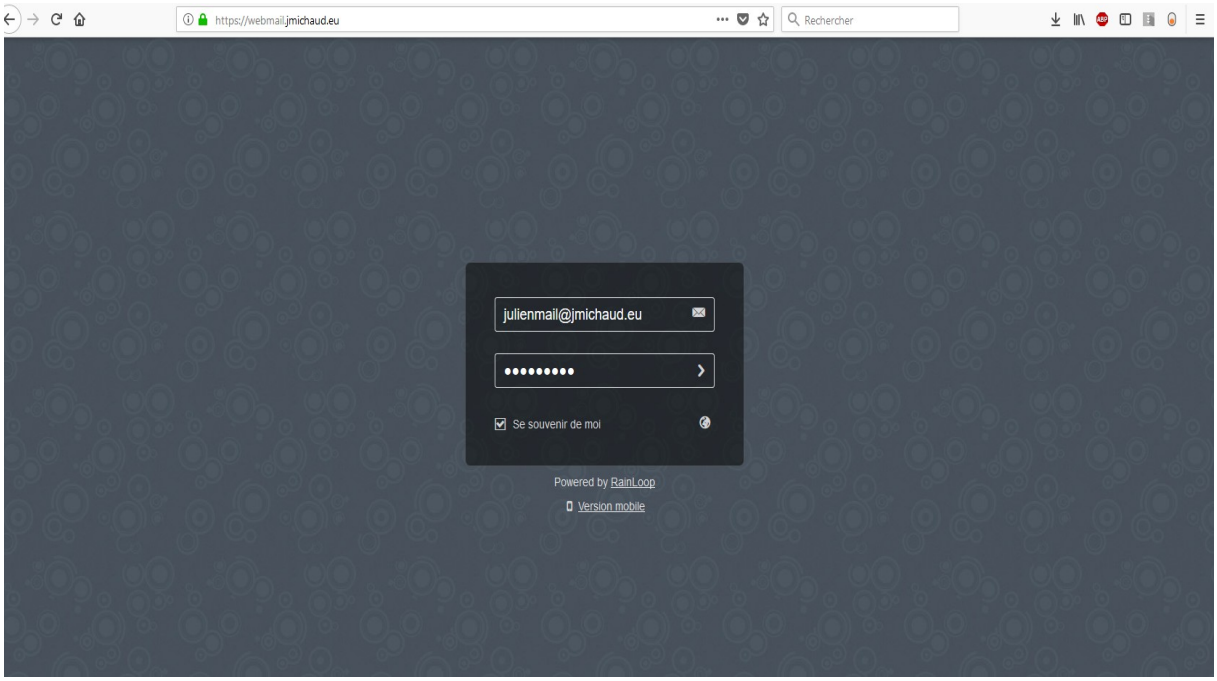
```
sudo certbot --apache --agree-tos --email your-email-address -d mail.example.com
```

Et voila !

Ecrire cette entrée DNS pour acceder à votre webmail depuis Internet:



Une fois fait, vous pouvez acceder à votre webmail depuis internet:


https://webmail.jmichaud.eu
Rechercher
julienmail@jmichaud.eu
Se souvenir de moi
Powered by RainLoop
Version mobile