Lancer la debian deb10-si5 ==> ip : 10.N°salle.x.160 (gateway 10.N°salle.254.0)
Mettre en DNS : 10.N°salle.79.90 ==> nano /etc/systemd/network/01.network ==>DNS= 10.N°salle.79.90
==> systemctl restart systemd-networkd
Testez ==> nslookup votrenomN°salle.siomb ==> doit renvoyer votre IP (10.N°salle.x.160)
apt-get update

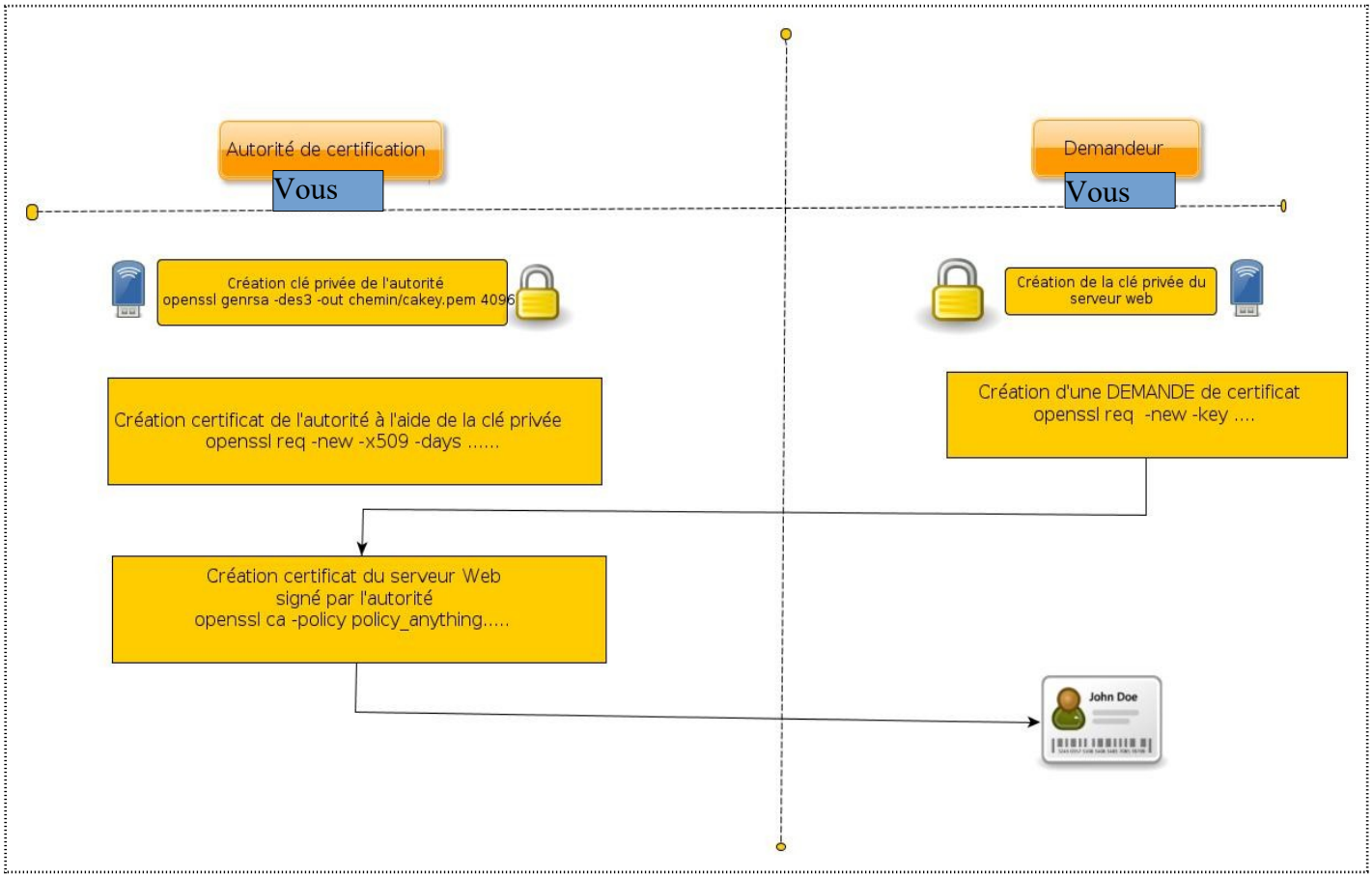
Serveur HTTPS

HTTPS repose sur le chiffrement SSL qui lui-même repose sur le chiffrement asymétrique comme SSH.

Activation du module SSL

- a**2enmod ssl**
- systemctl restart apache2

==> Allez dans /etc/ssl



Autorité *de certification*

Préparation du système :
OpenSSL fournit une application pour créer des certificats client ou serveur. Il s'appuie sur un fichier de configuration /etc/ssl/openssl.cnf .
Toutes les commandes d'openssl vont chercher divers paramètres dans ce fichier (à adapter aux besoins) sauf si la ligne de commande donne d'autres directives.

nano /etc/ssl/openssl.cnf ==> voici la partie qui nous intéresse

```
[ CA_default ]
dir               = ./demoCA              # Where everything is kept
certs             = $dir/certs            # Where the issued certs are kept
crl_dir           = $dir/crl              # Where the issued crl are kept
database          = $dir/index.txt        # database index file
...
new_certs_dir     = $dir/newcerts         # default place for new certs.
certificate       = $dir/cacert.pem       # The CA certificate
serial            = $dir/serial           # The current serial number
...
crl               = $dir/crl.pem          # The current CRL
private_key       = $dir/private/cakey.pem      # The private key
```

Remplacez donc **./demoCA par /etc/ssl**
$dir/cacert.pem par $dir/private/cacert.pem

- cd /etc/ssl

L'arborescence à créer est donc la suivante sauf si le répertoire existe déjà :

- certs accueille les nouveaux certificats à envoyer aux demandeurs ==> en principe il existe déjà
- crl accueille les certificats révoqués
- newcerts accueille une copie des certificats que l'organisation garde
- private accueillera les clés privées ==> en principe il existe déjà

==> mkdir certs crl newcerts private

```
drwxrwxr-x 2 root root     20480 23 août   2011 certs
drwxr-xr-x 2 root root      4096 12 sept. 12:35 crl
-rw-r--r-- 1 root root         0 12 sept. 12:36 index.txt
drwxr-xr-x 2 root root      4096 12 sept. 12:36 newcerts
-rwxrwxr-x 1 root root      9374 27 sept. 16:22 openssl.cnf
drwxrwxr-x 2 root ssl-cert  4096 27 sept. 19:55 private
-rw-r--r-- 1 root root         2 12 sept. 12:36 serial
root@glpi-m2l:/etc/ssl#
```

Les fichiers suivants seront à créer (voir plus bas) :

- private/cakey.pem est la clé privée de l'autorité de certification
- index.txt va contenir la liste des certificats que l'on générera
- serial contient le numéro de série du prochain certificat créé : il est donc nécessaire de l'initialiser à «01 »

touch index.txt
nano serial et mettre 01 dedans

Création : clé privée et certificat de l'autorité de certification

openssl genrsa -des3 -out private/cakey.pem 4096
bloch
==>Crée un fichier cakey.pem contenant la clé privée de 4096 bits protégée par une « pass phrase » (option -des3) Cette « pass phrase » (mot de passe) sera demandée à chaque utilisation de la clé.

Génération du certificat de l'autorité de certification

Il est possible de créer en une seule commande, à partir de cette clé, un certificat x509 auto-signé :

openssl req -new -x509 -days 365 -key /etc/ssl/private/cakey.pem -out /etc/ssl/private/cacert.pem

req -new : demande d'un nouveau certificat
-x509 : le certificat demandé est auto-signé
-days 365: la durée de validité du certificat sera du nombre de jours spécifié

Le mot de passe demandé est celui de la « pass phrase » saisie précédemment puisque vous utilisez la clé privée que vous avez protégée.
Il faut ensuite donner des renseignements sur l'autorité dFre certification :
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:herault
Locality Name (eg, city) : beziers
Organization Name (eg, company) [Internet Widgits Pty Ltd]: BTS
Organizational Unit Name (eg, section) []: SIO
Common Name (eg, YOUR name) []:CAXX.siomb ==> XX=N° de votre poste
Email Address []:
Ce certificat va permettre de vérifier l'authenticité et la validité du certificat du serveur Web.
Seule le "Common Name" doit être unique

Côté demandeur (le serveur web)

Génération du certificat du serveur WEB

Cette génération s'effectue en trois étapes.

- ### *Première étape : création de la clé privée du serveur*

  On génère la clé privée de la même manière que précédemment mais sans la protéger par une « pass phrase » (sans l'option -des3) car sinon cette dernière sera demandée à chaque lancement d'Apache2.

  openssl genrsa -out private/serveurwebkey.pem 4096

  A partir de la clé privée générée, il s'agit de créer un fichier de demande de certificat.

- ### *Deuxième étape : création de la demande de certificat*

  Pour générer le CSR (Certificate Signing Request) c'est à dire le formulaire de demande de certificat à partir de la clé privée préalablement créée, la commande est la suivante :

openssl req -new -key /etc/ssl/private/serveurwebkey.pem -out /etc/ssl/demande.pem

Le système va demander de saisir des champs ;

- Country Name (2 letter code) [AU]: FR
- State or Province Name (full name) [Some-State]: Europe
- Locality Name (eg, city) []: Paris
- Organization Name (eg, company) [Internet Widgits Pty Ltd]: SIO
- Organizational Unit Name (eg, section) []: Laboratoire France
- **Common Name (eg, YOUR name) [] nom DNS : www.votrenomN°salle.siomb**
- Email Address []:

Ce n'est pas la peine de saisir d'autres "extra attributes"... (challenge,compagny name..)

Common Name (CN) doit en principe correspondre au nom pleinement qualifié (nom DNS : celui que vous allez saisir dans l'URL de votre serveur qui correspond aussi à la directive **ServerName** dans le fichier de configuration d'Apache).
==> vérifiez la création du certificat dans private/

Troisième étape : production du certificat signé par l'autorité de certification à partir de la demande précédente

Cette signature et la production du certificat se réalisent avec la commande suivante :
vous êtes dans /etc/ssl

```
openssl ca -policy policy_anything -out /etc/ssl/private/certificat.pem
-infiles /etc/ssl/demande.pem
```

ca : c'est l'autorité de certification qui agit.
-policy policy_anything : c'est la rubrique [policy_anything] du fichier de configuration openssl.cnf qui sera utilisée. Dans notre cas seul le « Common Name » sera testé.
-out chemin/nom_du_certificat.pem : le certificat généré et à envoyer
-infiles chemin/nom_fichier_dem.pem : le fichier de demande de certificat utilisé pour générer le certificat.
Le certificat du serveur (que l'on doit normalement envoyer à celui qui nous l'a demandé) a bien été créé et il y a aussi une copie du certificat « 01.pem » qui a été automatiquement créé dans newcerts. L'autorité de certification le garde car elle a besoin de cet original en cas de révocation...

==> validez par "y" deux fois
Si erreur index.txt.attr ==> créer un fichier /etc/ssl/index.txt.attr et mettre dedans :
unique_subject = yes

Configuration d'Apache côté serveur (demandeur) pour https://

Créez un répertoire ==> mkdir /home/repssl
Editez le fichier /etc/apache2/sites-available/site.conf

<VirtualHost ***:443>**
SSLEngine on
SSLCertificateFile /etc/ssl/private/certificat.pem
SSLCertificateKeyFile /etc/ssl/private/serveurwebkey.pem
ServerAdmin webmaster@localhost
ServerName www.votrenomN°salle.siomb
DocumentRoot /home/repssl/

```
<Directory /home/repssl>
require all granted
</Directory>
```

</VirtualHost>

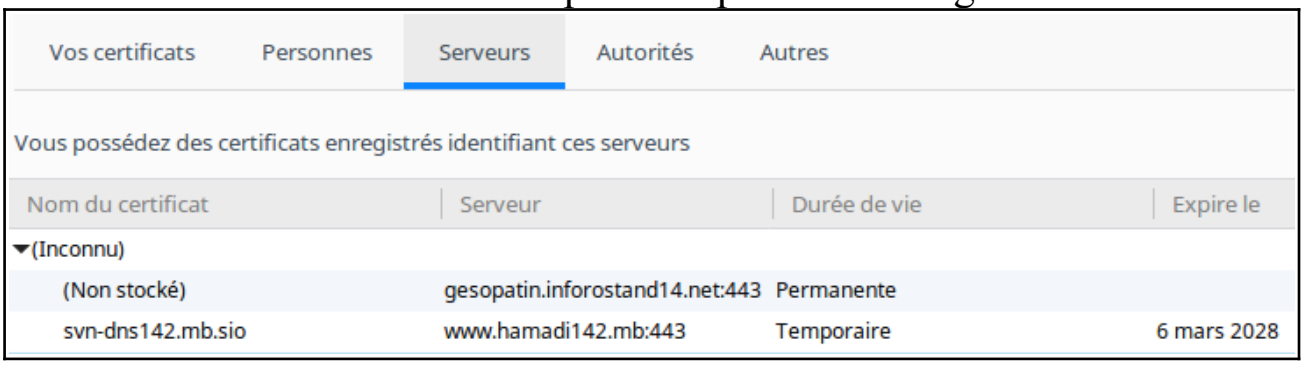
==> Activer notre virtualhost ==> a2ensite site
==> Activer le site ssl par défaut ==> a2ensite default-ssl.conf

Relancez apache ➔service apache2 restart
Créer une page info.php contenant <?php phpinfo() ;?> dans /home/repssl

==> tester la résolution DNS ==> nslookup www.votrenomN°salle.siomb

Testez depuis un client : https://www.votrenomN°salle.siomb/info.php

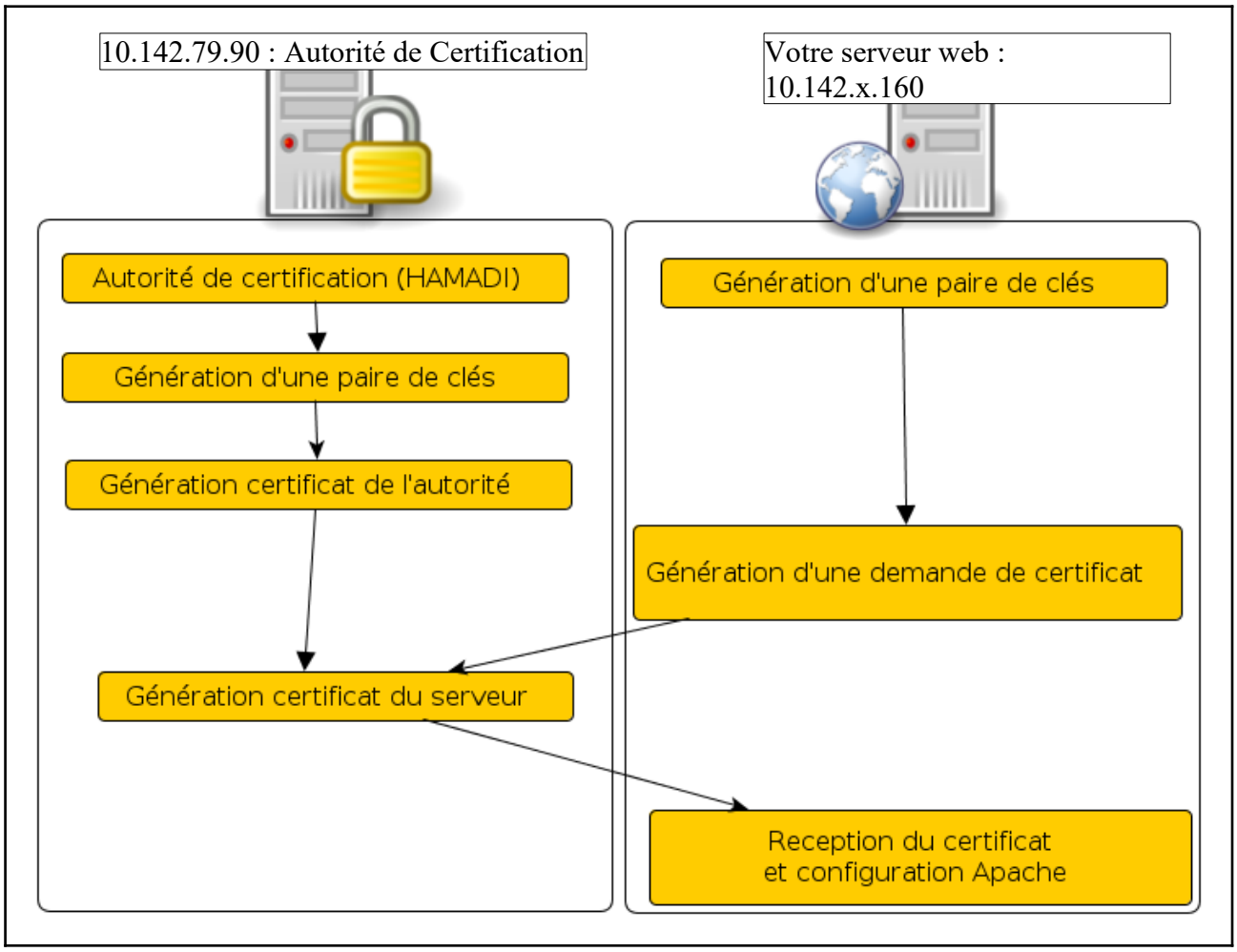
==> Ajouter une exception... dans le navigateur
Vérifiez le contenu du certificat présenté par votre navigateur.



Résumé des commandes

	AC	Serveur
Création clé privée	openssl genrsa -des3 -out private/cakey.pem 4096	openssl genrsa -out private/serveurwebkey.pem 4096
Certificat de l'autorité de certification	openssl req -new -x509 -days 365 -key private/cakey.pem -out cacert.pem	
Demande d'un certificat		openssl req -new -key private/serveurwebkey.pem -out demande.pem ==>envoi du fichier "demande.pem" à l'autorité de certification
Production du certificat pour le serveur demandeur	openssl ca -policy policy_anything -out private/certificat.pem -infiles **demande.pem** envoi du fichier "certificat.pem" au demandeur	
Configuration Apache		**SSLEngine on** **SSLCertificateFile /etc/ssl/private/certificat.pem** **SSLCertificateKeyFile /etc/ssl/private/serveurwebkey.pem**

2nd partie ==> Séparer le serveur Web de l'autorité de certification



L'autorité de certification est : 10.N°salle.79.90.

Faire une demande : openssl req -new -key private/serveurwebkey.pem -out **demandeX.pem (X=N°poste sur 1 chiffre si <10)**

Vous devez transmettre vos demandes de certificats via scp avec l'utilisateur sio mot de passe bloch.
==> scp **demandeX.pem** sio@10.N°salle.79.90:/home/sio

L'autorité génère le certificat **reponseX.pem et le copier dans /var/www/html**

Vous Récupérez ce certificat via wget (copie via http)==>
cd /etc/ssl/private

wget http://10.N°salle.79.90/reponseX.pem

Créez un répertoire /home/achamadi

Créez un virtualhost **"/etc/apache2/sites-available/achamadi.conf"** qui pointe sur /home/achamadi avec une page index.html contenant "Certifcat Hamadi poste XX" , accessible en https en utilisant ce certificat.

Aide ==> reprendre le modèle de :

"/etc/apache2/sites-available/site.conf"

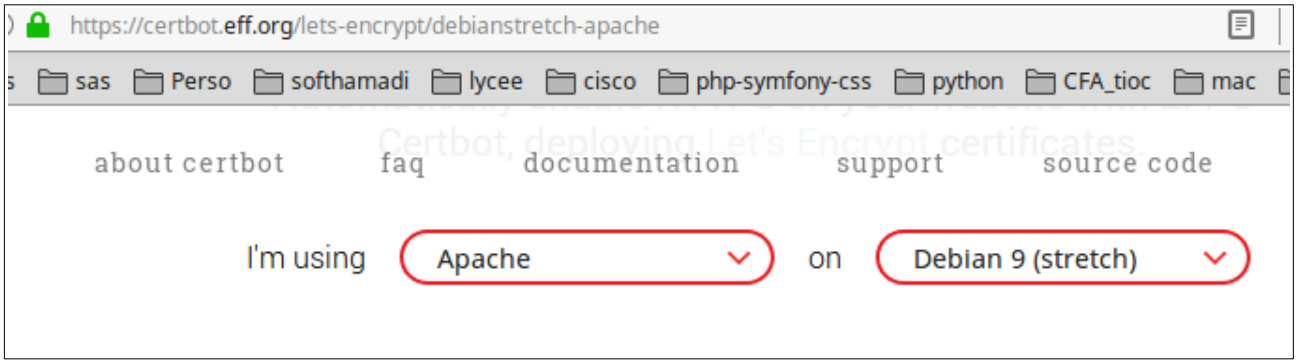
Activer le site ==> a2ensite **achamadi.conf**

Test ⇒ https://webmail.votrenomN°salle.siomb

Certificat libre et gratuit

```
IL faut une adresse IP Publique et un nom de domaine publique.

Pour téléchrger un certificat gratuit ==> il faut un client qui
gère le protocole ACME comme le client ACME Certbot.
https://certbot.eff.org
```




```
Installation ==>
apt-get update
apt-get install certbot python-certbot-apache
certbot --apache certonly
```

Configuration

Certificat et clé publique par défaut enregistrés dans
/etc/letsencrypt/live/votrenomN°salle.siomb/**fullchain.pem**

La clé privée dans :
/etc/letsencrypt/live/votrenomN°salle.siomb/**privkey.pem**

-Répertoire à sauvegarder régulièrement /etc/letsencrypt.
nano sitehttps.conf

<VirtualHost ***:443>**
SSLEngine on
SSLCertificateFile /etc/letsencrypt/live /votrenomN°salle.siomb/fullchain.pem
SSLCertificateKeyFile /etc/letsencrypt/live/votrenomN°salle.siomb/privkey.pem
ServerAdmin webmaster@localhost
ServerName www.votrenomN°salle.siomb
DocumentRoot /home/repssl/

```
<Directory /home/repssl>
require all granted
</Directory>
```

</VirtualHost>

Redirection du port 80 sur le 443

```
<VirtualHost *:80>
        ServerName      www.g07.joutes.pw
        ServerAlias     g07.joutes.pw
```

```
        Redirect permanent / https://www.g07.joutes.pw/
        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined
        </VirtualHost>
```

Ne pas oublier d'activer le module ssl ==> a2enmod ssl

Renouvellement automatique ==> cron

Le certificat est valable 90 jours

```
$ certbot renew --dry-run
```

==> faire une tâche cron qui se lance tous les 90 jours