# certbot pour Let's Encrypt

**Certbot** est un binaire qui permet de mettre en œuvre un certificat SSL pour un domaine d'un site Internet.

Voici les actions a effectuer sur **Linux Debian 10**.

Depuis 2020, **certbot** est installé depuis **snapcraft**.

## 1. Installer snap

Si vous n'avez pas encore installé **snap**, il faut executer les commandes suivantes dans un **terminal**. Les commandes doivent avoir une élevation des droits.

```
sudo apt update
sudo apt install snapd
sudo snap install core
```

Si **snap** est déjà installé, assurez-vous d'être à jour :

```
sudo snap refresh core
```

## 2. Installer certbot

Exécutez la commande suivante oour installer **cerbot** sur votre machine. Ceci est a exéctuer dans un terminal.

```
sudo snap install --classic certbot
sudo ln -s /snap/bin/certbot /usr/bin/certbot
```

## 3. Générer un certificat - gestion manuelle

Si vous suivez ce guide, je vous conseille une gestion manuelle de vos certificats.

Il y a deux variables :

| | |
|---|---|
| **dirWeb** | Dossier des données du site. Pour habitude, c'est quelque chose du genre `/var/www/html/` |
| **siteName** | Nom du site Internet. Par exemple `abonnel.fr` |

```
read -p "Quel est votre nom de domaine ? " siteName
read -p "Dossier Web du site ? " dirWeb
sudo certbot certonly --webroot -w $dirWeb -d $siteName --register-unsafely-without-email
```

L'avantage de ce script :

- pas d'arrêt d'Apache 2
- pas de mail à saisir
- autonomie sur la configuration Apache 2

Quelques **chemins** à retenir :

| | |
|---|---|
| **fichier de configuration** | /etc/letsencrypt/renewal/$siteName.conf |
| **dossier archive** | /etc/letsencrypt/archive/$siteName |
| **fichier cert** | /etc/letsencrypt/live/$siteName/cert.pem |
| **fichier privkey** | /etc/letsencrypt/live/$siteName/privkey.pem |
| **fichier chain** | /etc/letsencrypt/live/$siteName/chain.pem |
| **fichier fullchain** | /etc/letsencrypt/live/$siteName/fullchain.pem |

Pour enregistrer un domaine principal avec et sans les **www**, il faut utiliser le script suivant :

```
echo "Quel nom de domaine (avec www) ?"
read siteNameWww
echo "Quel nom de domaine (sans les www) ?"
read siteName
sudo certbot certonly --webroot -w /var/www/html/ -d $siteName -d $siteNameWww --register-unsafely-without-email
```

# 4. Générer un certificat - gestion automatique

Si vous avez effectué le paragraphe précédent generer-un-certificat-gestion-manuelle, il est inutile de suivre les recommandations de ce chapitre.

Exécutez le commande suivant pour générer un certificat et édité votre configuration Apache de manière automatique. Le protocole **https** s'active de manière automatique, en une seule étape.

```
sudo certbot --apache
```

# 5. Renouveler les certificats automatiquement

Éditer la tâche des tâches Linux du compte `root`, *crontab* :

```
sudo crontab -e
```

La tâche doit exécutée le programme `certbot` avec l'option de renouvellement, `renew`. L'option `post-hook` permet d'indiquer la commande à exécuter après le traitement de `certbot`. Dans notre cas, on demande à `systemctl` de recharger la configuration `Apache 2`.

```
0 23 1-7 */2 4 python -c 'import random; import time;
time.sleep(random.random() * 3600)' && /usr/bin/certbot renew --post-hook
"systemctl reload apache2" >> /var/log/letsencrypt/renew.log
```

Explications :

```
Tous les deux mois ( 0 23 1-7 */2 4 )
à 23 heures ( 0 23 1-7 */2 4 ),
le premier jeudi ( 0 0 1-7 */2 4 ),
lancement d'un script Python, qui retarde 1 heure au maximum
(random.random() * 3600),
l’exécution de la mise à jour de certbot.
```

Vous pouvez trouver d'autres informations sur la page https://crontab.guru/#0_23_1-7_*/2_4

# 6. Afficher les dates du certificats

Pour afficher les dates de génération et d'expiration d'un certificat en local sur un serveur :

```
sudo ls /etc/letsencrypt/live/
read -p "Quel est votre nom de domaine ?" siteName
sudo openssl x509 -noout -dates -in
/etc/letsencrypt/live/$siteName/fullchain.pem
```

—

Pour afficher les dates de génération et d'expiration d'un certificat d'un site distant :

```
openssl s_client -connect www.w3.org:443| openssl x509 -noout -dates
```

Exemple de résultat :

```
(81) cedric24c@dskcdc001:~ $ openssl s_client -connect www.w3.org:443| openssl x509 -noout -dates
depth=2 C = US, ST = New Jersey, L = Jersey City, O = The USERTRUST Network, CN = USERTrust RSA Certification Authori
ty
verify return:1
depth=1 C = FR, ST = Paris, L = Paris, O = Gandi, CN = Gandi Standard SSL CA 2
verify return:1
depth=0 OU = Domain Control Validated, OU = Gandi Standard Wildcard SSL, CN = *.w3.org
verify return:1
notBefore=May 23 00:00:00 2019 GMT
notAfter=Jun  1 23:59:59 2021 GMT
```

# 7. Réinitialiser la configuration Let's Encrypt

letsencrypt_erase

```
#!/bin/bash
# réinitialiser let's encrypt pour un domaine precis
echo Affichage des noms possibles :
sudo ls /etc/letsencrypt/live
echo .
echo "Quel nom de domaine ?"
read siteName
sudo rm -fr /etc/letsencrypt/live/$siteName/
sudo rm /etc/letsencrypt/renewal/$siteName.conf
sudo rm -fr /etc/letsencrypt/archive/$siteName/
```