tutoriel

# SSL pour Nginx : mettre en place un certificat SSL Let's Encrypt avec Certbot

Les protocoles Web **TLS** (et son prédécesseur **SSL**) englobent le trafic dans un contenant protégé et chiffré pour :

- échanger en toute sécurité sans que les messages soient interceptés par un tiers.
- permettre aux utilisateurs de vérifier l'identité des sites auxquels ils se connectent.

L'autorité de certification **Let's Encrypt** fournit gratuitement des certificats TLS/SSL, permettant le HTTPS chiffré sur les serveurs Web. Le logiciel **Certbot** automatise et facilite les étapes de l'installation d'un certificat. le processus d'obtention et d'installation d'un certificat est entièrement automatisé sur les serveurs Apache et Nginx.

Normalement, **Certbot** est exécuté par un administrateur sur un serveur web.

Cette page décrit l'installation et l'exécution de Certbot sur un serveur

Ce tutoriel utilise Certbot pour obtenir un certificat SSL gratuit pour Nginx et le configurer pour qu'il se renouvelle automatiquement.

Ce didacticiel utilise un fichier d'hôte virtuel Nginx distinct au lieu du fichier par défaut. Nous recommandons de créer de nouveaux fichiers d'hôte virtuel Nginx pour chaque domaine, pour conserver les fichiers par défaut comme configuration de secours.

## Pré-requis

- **avoir enregistré officiellement le nom de domaine** avec lequel vous souhaitez utiliser le certificat. Ce nom de domaine doit être configuré et pointer vers votre site (adresse IP)

Dans ce tutoriel, ce sera le domaine **mondomaine.fr**

- **Python**
- **Nginx** installé avec un bloc *server* pour votre domaine.

Ce tutoriel utilisera **/etc/nginx/sites-available/mondomaine.fr** comme exemple.

# Première étape : Installation de Certbot

Sous SSH, installez Certbot sur le serveur web :

```
...@...:~ $ sudo apt update
...@...:~ $ sudo apt install certbot python-certbot-nginx
```

Si vous êtes sous **Apache**, installez **python-certbot-apache** au lieu de python3-certbot-nginx

# Autres étapes

Nous allons récupérer depuis Let’s Encrypt un certificat SSL pour notre serveur.

## Obtenir et installer vos certificats

1. **Configuration de Nginx** :
    1. **Certbot** recherche dans votre config de Nginx une directive **server_name** correspondant au domaine pour lequel vous demandez un certificat.
        - Si vous avez bien configuré l'hôte virtuel lors l'installation de Nginx, le fichier **/etc/nginx/sites-available/mondomaine.fr** doit contenir un bloc *server* pour votre domaine avec la directive **server_name** définie de manière appropriée :

          /etc/nginx/sites-available/mondomaine.fr

          ```
          server {
          <...>
              server_name mondomaine.fr www.mondomaine.fr;
          <...>
          }
          ```

          Si c'est le cas, quittez votre éditeur et passez à l'étape suivante ; sinon, mettez le fichier à jour pour qu'il corresponde.

    2. Vérifiez la syntaxe de vos modifications de configuration avec :

       ```
       ...@...:~ $ sudo nginx -t
       nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
       nginx: configuration file /etc/nginx/nginx.conf test is successful
       ```

       S'il y a des erreurs, rouvrez le fichier de l'hôte et recherchez les fautes de frappe.

3. Si cela fonctionne sans erreur, rechargez Nginx pour charger la nouvelle configuration :

```
...@...:~ $ sudo nginx -s reload
```

4. Certbot peut maintenant trouver le bon bloc server et le mettre à jour.

2. **Autoriser HTTPS sur le pare-feu** ; si le pare-feu **ufw** est activé, ajustez les paramètres pour autoriser le trafic HTTPS :
    1. Affichez le réglage actuel :

```
...@...:~ $ sudo ufw status
Status: active

To                         Action      From
--                         ------      ----
80                         ALLOW       Anywhere
22/tcp                     ALLOW       Anywhere
80 (v6)                    ALLOW       Anywhere (v6)
22/tcp (v6)                ALLOW       Anywhere (v6)
```

    2. Si le port 443 n'est pas ouvert, activez-le par :

```
...@...:~$ sudo ufw allow ssh
```

3. **Obtention d'un certificat SSL**
    - Certbot fournit des plugins pour obtenir des certificats SSL. Le plugin –nginx reconfigure Nginx et recharge la configuration autant que nécessaire. Pour utiliser ce plugin, tapez :

```
...@...:~$ sudo certbot --nginx -d mondomaine.fr -d
www.mondomaine.fr
...
Enter email address (used for urgent renewal and security
notices) (Enter 'c' to
cancel): votre@email.fr
...
Please read the Terms of Service at
...
(A)gree/(C)ancel: A
<...>
Would you be willing to share your email address with ...
...
(Y)es/(N)o: N
...
Please choose whether or not to redirect HTTP traffic to
HTTPS, removing HTTP access.
...
1: No redirect - Make no further changes to the webserver
```

```
configuration.
2: Redirect - Make all requests redirect to secure HTTPS
access. Choose this for
new sites, or if you're confident your site works on HTTPS.
You can undo this
change by editing your web server's configuration.
...
Select the appropriate number [1-2] then [enter] (press 'c' to
cancel): 2
<...>
```

Cette commande lance certbot avec le plugin **–nginx**, en utilisant **-d** pour spécifier les noms de domaine pour lesquels nous voulons un certificat valide.

- La première fois que vous exécutez certbot, vous serez invité à saisir une adresse e-mail et à accepter les conditions d'utilisation.
- Puis certbot vérifie que vous êtes le propriétaire du domaine pour lequel vous demandez un certificat.
- Si cela réussit, certbot vous demande comment configurer vos paramètres HTTPS :
  - **1** : Pas de redirection - Ne faites plus de modification à la configuration du serveur Web.
  - **2** : Redirection - Rediriger toutes les demandes vers un accès sécurisé HTTPS.

5. Sélectionnez votre choix puis appuyez sur ENTREE.
6. La configuration est mise à jour et Nginx se recharge pour récupérer les nouveaux paramètres.
7. certbot termine avec un message vous indiquant que le processus a réussi et où sont stockés vos certificats :

```
...@...:~ $
IMPORTANT NOTES:
 - Congratulations! Your certificate and chain have been saved
at
   /etc/letsencrypt/live/example.com/fullchain.pem. Your cert
will
   expire on 2017-10-23. To obtain a new or tweaked version of
this
   certificate in the future, simply run certbot again with
the
   "certonly" option. To non-interactively renew *all* of your
   certificates, run "certbot renew"
 - Your account credentials have been saved in your Certbot
   configuration directory at /etc/letsencrypt. You should
make a
   secure backup of this folder now. This configuration
directory will
   also contain certificates and private keys obtained by
Certbot so
   making regular backups of this folder is ideal.
 - If you like Certbot, please consider supporting our work
```

```
by:

   Donating to ISRG / Let's Encrypt:
https://letsencrypt.org/donate
   Donating to EFF:
https://eff.org/donate-le
```

8. Vos certificats sont maintenant téléchargés, installés et configurés.
9. Rechargez votre site Web en utilisant https:// et examinez l'icône de cadenas de votre navigateur qui doit être verte.
10. Si vous testez votre serveur à l'aide du test de https://www.ssllabs.com/ssltest/analyze.html (renseignez votre domaine et testez), il obtiendra une note A.

4. **Vérification du renouvellement automatique de Certbot** :
   - Les certificats de Let’s Encrypt ne sont valables que quatre-vingt-dix jours.
   - certbot renouvelle automatiquement le certificat en ajoutant à /etc/cron.d un script qui s'exécute deux fois par jour et renouvellera automatiquement tout certificat dans les trente jours suivant l'expiration.
   - Pour tester le processus de renouvellement :

     ```
     ...@...:~$ sudo certbot renew --dry-run
     ```

     Si vous ne voyez aucune erreur, vous êtes prêt.

   - Si nécessaire, Certbot renouvellera vos certificats et rechargera Nginx pour récupérer les modifications.
   - Si le processus de renouvellement automatisé échoue, Let’s Encrypt enverra un message à l'e-mail que vous avez spécifié, vous avertissant que votre certificat est sur le point d'expirer.

### Obtenir seulement un certificat

```
...@...:~ $ sudo certbot certonly --nginx
```

## Conclusion

Dans ce didacticiel, nous avons installé le client certbot Let’s Encrypt, téléchargé des certificats SSL pour notre domaine, configuré Nginx pour utiliser ces certificats et configuré le renouvellement automatique des certificats.

## Problèmes connus

## Voir aussi

- **(fr)** https://letsencrypt.org/fr
- **(fr)** Tuto : certificat SSL gratuit avec Let’s Encrypt
- **(en)** Get Certbot
- **(en)** certbot instructions
- **(en)** https://www.digitalocean.com/community/tutorials/how-to-set-up-let-s-encrypt-with-nginx-server-blocks-on-ubuntu-16-04
- **(en)** https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-ubuntu-14-04
- **(en)** https://kloscomputing.co.uk/wordpress/2018/03/30/raspberry-pi-and-ssl-certificate-using-lets-encrypt/
- **(en)** https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-debian-10
- **(en)** https://www.techcoil.com/blog/installing-certbot-on-raspbian-buster-for-obtaining-lets-encrypts-browser-trusted-certificates-for-your-raspberry-pi-server-applications/
- **(en)** https://pimylifeup.com/raspberry-pi-ssl-lets-encrypt/

---

*Basé sur « How To Set Up Let's Encrypt with Nginx Server Blocks on Ubuntu 16.04 » par Hazel Virdó.*