

Configuration d'un Reverse Proxy HTTPS avec Nginx

Ce dépôt contient un guide complet pour configurer un reverse proxy HTTPS avec Nginx sur Ubuntu Server. Idéal pour les environnements de développement, de test, ou pour des projets académiques.

Table des matières

- [Introduction](#)
- [Prérequis](#)
- [Installation et configuration](#)
 - [1. Installation d'Nginx](#)
 - [2. Génération d'un certificat SSL avec mkcert](#)
 - [3. Création du contenu web](#)
 - [4. Configuration de Nginx](#)
 - [5. Test et vérification](#)
- [Configuration pour exposer le service](#)
- [Détails techniques et fonctionnement](#)
- [Problèmes courants et solutions](#)

Introduction

Ce tutoriel vous guide à travers la configuration d'un reverse proxy Nginx avec une redirection automatique de HTTP vers HTTPS, en utilisant des certificats SSL localement approuvés générés par mkcert. Cette configuration est parfaite pour des environnements de développement sécurisés ou des projets éducatifs.

Prérequis

- Ubuntu Server (version 18.04 ou plus récente)
- Accès root ou utilisateur avec privilèges sudo
- Connaissance de base des commandes Linux

Installation et configuration

1. Installation d'Nginx

Commencez par mettre à jour la liste des paquets et installer Nginx :

```
sudo apt update
sudo apt install -y nginx
```

Vérifiez que Nginx est correctement installé et démarré :

```
sudo systemctl status nginx
```

2. Génération d'un certificat SSL avec mkcert

mkcert est un outil qui simplifie la création de certificats de développement localement approuvés :

```
# Installation des dépendances
sudo apt install -y libnss3-tools


# Téléchargement et installation de mkcert
wget https://github.com/FiloSottile/mkcert/releases/download/v1.4.3/mkcert-v1.4.3-linux-amd64
chmod +x mkcert-v1.4.3-linux-amd64
sudo mv mkcert-v1.4.3-linux-amd64 /usr/local/bin/mkcert

# Installation de l'autorité de certification locale
mkcert -install

# Création du répertoire pour les certificats
sudo mkdir -p /etc/nginx/ssl

# Génération du certificat pour votre domaine
mkcert -key-file /etc/nginx/ssl/ipssibankso.key -cert-file /etc/nginx/ssl/ipssibankso.crt ipss

# Ajustement des permissions
sudo chmod 644 /etc/nginx/ssl/ipssibankso.crt
sudo chmod 600 /etc/nginx/ssl/ipssibankso.key
```



3. Création du contenu web

Créez un répertoire pour votre site web et une page HTML de démonstration :

```
# Création du répertoire pour le site
sudo mkdir -p /var/www/ipssibankso

# Création de la page HTML
sudo nano /var/www/ipssibankso/index.html
```

Ajoutez ce contenu HTML à votre fichier :

```

<!DOCTYPE html>
<html>
<head>
  <title>ipssibankso.com - Site Sécurisé</title>
  <style>
    body { font-family: Arial, sans-serif; margin: 40px; line-height: 1.6; }
    h1 { color: #0066cc; }
    .secure { color: green; font-weight: bold; }
    .box { border: 1px solid #ddd; padding: 20px; border-radius: 5px; background-color: #f
  </style>
</head>
<body>
  <h1>Bienvenue sur ipssibankso.com</h1>
  <div class="box">
    <h2>Site configuré avec <span class="secure">HTTPS sécurisé</span></h2>
    <p>Caractéristiques de cette configuration :</p>
    <ul>
      <li>Reverse proxy avec Nginx</li>
      <li>Redirection automatique HTTP vers HTTPS</li>
      <li>Certificat SSL localement approuvé</li>
      <li>Configuration complète pour un site web sécurisé</li>
    </ul>
    <p>Ce site est maintenant accessible via HTTPS et affiche le cadenas de sécurité sans
  </div>
</body>
</html>

```

4. Configuration de Nginx

Créez un fichier de configuration pour votre site :

```
sudo nano /etc/nginx/sites-available/ipssibankso.conf
```

Ajoutez cette configuration :

```

# Redirection HTTP vers HTTPS
server {
  listen 80;
  server_name ipssibankso.com www.ipssibankso.com;
  return 301 https://$host$request_uri;
}

# Configuration HTTPS
server {
  listen 443 ssl;
  server_name ipssibankso.com www.ipssibankso.com;

  # Certificats SSL

```

```

ssl_certificate /etc/nginx/ssl/ipssibankso.crt;
ssl_certificate_key /etc/nginx/ssl/ipssibankso.key;

# Paramètres SSL recommandés
ssl_protocols TLSv1.2 TLSv1.3;
ssl_prefer_server_ciphers on;

# Racine du site web
root /var/www/ipssibankso;
index index.html;

location / {
    try_files $uri $uri/ =404;

    # Pour un reverse proxy vers une application backend, décommentez ces lignes
    # proxy_pass http://localhost:8080;
    # proxy_set_header Host $host;
    # proxy_set_header X-Real-IP $remote_addr;
    # proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    # proxy_set_header X-Forwarded-Proto $scheme;
}
}

```

Activez votre configuration et redémarrez Nginx :

```

# Créer un lien symbolique
sudo ln -s /etc/nginx/sites-available/ipssibankso.conf /etc/nginx/sites-enabled/

# Vérifier la configuration
sudo nginx -t

# Si tout est OK, redémarrer Nginx
sudo systemctl restart nginx

```

5. Test et vérification

Pour tester localement, ajoutez votre domaine dans le fichier hosts :

```
sudo nano /etc/hosts
```

Ajoutez cette ligne :

```
127.0.0.1 ipssibankso.com www.ipssibankso.com
```

Accédez à votre site via un navigateur web :

1. Ouvrez un navigateur sur votre machine virtuelle

2. Accédez à <http://ipssibankso.com>
3. Vous devriez être automatiquement redirigé vers <https://ipssibankso.com>
4. Comme vous avez correctement installé mkcert, vous ne devriez pas voir d'avertissement de sécurité
5. Vérifiez la présence du cadenas dans la barre d'adresse

Configuration pour exposer le service

Pour rendre votre site accessible depuis Internet, vous pouvez utiliser différentes méthodes :

Option 1 : Utiliser ngrok (pour des démonstrations temporaires)

```
# Installation de ngrok
wget https://bin.equinox.io/c/bNyj1mQVY4c/ngrok-v3-stable-linux-amd64.tgz
tar xvzf ngrok-v3-stable-linux-amd64.tgz
sudo mv ngrok /usr/local/bin/

# Lancement du tunnel
ngrok http 80
```

Option 2 : Utiliser un domaine réel avec Let's Encrypt

Si vous avez un domaine réel et que votre serveur est accessible depuis Internet :

```
sudo apt install certbot python3-certbot-nginx
sudo certbot --nginx -d votre-domaine.com -d www.votre-domaine.com
```

Détails techniques et fonctionnement

Qu'est-ce qu'un reverse proxy ?

Un reverse proxy est un serveur intermédiaire qui reçoit les requêtes des clients et les transmet aux serveurs appropriés. Voici ses principaux avantages :

- **Sécurité** : Masque l'infrastructure backend
- **SSL Termination** : Gère le chiffrement/déchiffrement
- **Load Balancing** : Peut distribuer les requêtes entre plusieurs serveurs
- **Mise en cache** : Améliore les performances

Fonctionnement de la redirection HTTP vers HTTPS

La configuration définie dans ce tutoriel utilise la directive `return 301` pour rediriger automatiquement tout trafic HTTP vers HTTPS, garantissant que toutes les communications sont chiffrées.

Avantages de mkcert pour le développement

- Installation d'une autorité de certification locale
- Certificats reconnus par le navigateur sans avertissements
- Support pour plusieurs domaines et sous-domaines
- Parfait pour les environnements de développement

Problèmes courants et solutions

Nginx ne démarre pas

Erreur : nginx: [emerg] could not build server_names_hash

Solution : Augmenter la taille des buckets de hash dans nginx.conf :

```
http {  
    server_names_hash_bucket_size 128;  
    # autres configurations...  
}
```

Problèmes avec les certificats SSL

Erreur : Avertissements de sécurité dans le navigateur

Solution : Vérifier que mkcert est correctement installé et que l'autorité de certification est installée dans le navigateur avec `mkcert -install`

Erreurs 404 après configuration

Problème : Le contenu n'est pas trouvé

Solution : Vérifier les chemins dans la configuration Nginx et s'assurer que les répertoires ont les bonnes permissions :

```
sudo chown -R www-data:www-data /var/www/ipssibankso
```