

Pour qu'une application **Next.js** servie par **Apache** sur Linux passe en **HTTPS** sans afficher « site non sécurisé », il faut :

1. **Installer un certificat SSL valide** (Let's Encrypt, autorité reconnue)
2. **Configurer Apache** pour servir le site en HTTPS
3. (Si besoin) **configurer un reverse proxy** vers l'app Next.js

Voici les étapes complètes et fiables :

 1. Installer un certificat SSL valide (Let's Encrypt / Certbot)

► Installation de Certbot (Debian/Ubuntu)

```
```bash
sudo apt update
sudo apt install certbot python3-certbot-apache
```
```


► Génération du certificat SSL

```
```bash
sudo certbot --apache -d monsite.com -d www.monsite.com
```
```

Certbot :

- * génère un certificat TLS signé par Let's Encrypt
- * configure automatiquement Apache pour HTTPS
- * met en place le renouvellement automatique

👉 Après cela, **le message "site non sécurisé" disparaît**, car le certificat est reconnu par les navigateurs.

 2. Configuration Apache pour Next.js (reverse proxy)

Next.js ne tourne pas nativement dans Apache :
on lance Next.js sur un port (ex : 3000), puis Apache fait proxy vers ce port.

Exemple de config Apache

Fichier : `/etc/apache2/sites-available/monsite.conf``

```
```apache
<VirtualHost *:80>
 ServerName monsite.com
 ServerAlias www.monsite.com

 RewriteEngine On
 RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI}
[END,NE,R=permanent]
</VirtualHost>
```

```
<VirtualHost *:443>
 ServerName monsite.com
 ServerAlias www.monsite.com

 SSLEngine On
 SSLCertificateFile /etc/letsencrypt/live/monsite.com/fullchain.pem
 SSLCertificateKeyFile /etc/letsencrypt/live/monsite.com/privkey.pem

 ProxyPreserveHost On
 ProxyRequests Off
 ProxyPass / http://127.0.0.1:3000/
 ProxyPassReverse / http://127.0.0.1:3000/
</VirtualHost>
```
```

Activer les modules Apache nécessaires

```
```bash
sudo a2enmod ssl proxy proxy_http rewrite
sudo systemctl restart apache2
```
```

 3. Lancer l'app Next.js en production

Dans votre dossier Next.js :

```
```bash
npm run build
npm start
```
```

Par défaut Next.js écoute sur `http://localhost:3000`.

 4. Vérification

Dans votre navigateur :

- * l'icône cadenas doit apparaître
- * aucune alerte « site non sécurisé »
- * certificat émis par Let's Encrypt

Vérification avec OpenSSL :

```
```bash
openssl s_client -connect monsite.com:443 -servername monsite.com
```
```

 Résultat

Votre site Next.js est maintenant accessible en HTTPS avec un certificat valide, sans l'avertissement « site non sécurisé ».

```
=====
=====
DEUXIEME PROMPT
=====
=====
```

Voici un **exemple complet et fonctionnel** de configuration Apache + HTTPS pour une app **Next.js** sur **Red Hat** ***sans Certbot*** (donc avec vos propres fichiers SSL : certificat, clé privée et éventuellement chaîne intermédiaire).

1. Prérequis

* Apache installé :

```
```bash
sudo dnf install httpd mod_ssl
```
```

* Votre app Next.js tourne en mode production :

```
```bash
npm run build
npm start
```
```

Par défaut : ***port 3000***

* Vos fichiers SSL sont prêts et copiés dans :

```
```
/etc/pki/tls/certs/your_domain.crt
/etc/pki/tls/private/your_domain.key
/etc/pki/tls/certs/your_domain_chain.crt (optionnel mais recommandé)
```
```

2. Configuration Apache (VirtualHost HTTPS)

Créez un fichier de configuration, par exemple :

```
```
/etc/httpd/conf.d/nextjs-ssl.conf
```
```

Et placez-y ***ce contenu complet*** :

```
```apache
<VirtualHost *:443>
 ServerName example.com
 ServerAlias www.example.com
```

```

--- SSL ---
SSLEngine on
SSLCertificateFile /etc/pki/tls/certs/your_domain.crt
SSLCertificateKeyFile /etc/pki/tls/private/your_domain.key

Facultatif mais fortement recommandé (chaîne intermédiaire)
SSLCertificateChainFile /etc/pki/tls/certs/your_domain_chain.crt

Sécurisation minimale recommandée
SSLProtocol all -SSLv2 -SSLv3
SSLCipherSuite HIGH:!aNULL:!MD5

--- Reverse proxy vers Next.js ---
ProxyPreserveHost On
ProxyPass / http://127.0.0.1:3000/
ProxyPassReverse / http://127.0.0.1:3000/

ErrorLog /var/log/httpd/nextjs_ssl_error.log
CustomLog /var/log/httpd/nextjs_ssl_access.log combined
</VirtualHost>

Redirection HTTP → HTTPS
<VirtualHost *:80>
 ServerName example.com
 ServerAlias www.example.com
 Redirect "/" "https://example.com/"
</VirtualHost>
```

```

3. Activer le proxy Apache

```

```bash
sudo sed -i 's/^#\ (LoadModule proxy_module\)/\1/'
/etc/httpd/conf.modules.d/00-proxy.conf
sudo sed -i 's/^#\ (LoadModule proxy_http_module\)/\1/'
/etc/httpd/conf.modules.d/00-proxy.conf
```

```

Ou installer explicitement :

```

```bash
sudo dnf install mod_proxy mod_proxy_http
```

```

4. Vérifier la configuration Apache

```

```bash
sudo apachectl configtest
```


```

Vous devez obtenir :

```

**Syntax OK**


```

 5. Redémarrer Apache

```
```bash
sudo systemctl restart httpd
sudo systemctl enable httpd
```
```

 Résultat

Votre site Next.js est maintenant servi en ****HTTPS via Apache**** en reverse proxy, sans Certbot, avec vos propres fichiers SSL.

 Besoin d'adapter le fichier pour plusieurs domaines, un wildcard, ou un service Node géré par systemd ?

Je peux vous générer les fichiers complet prêts à l'emploi.