

?Ajout d'un sous-domaine sur le reverse proxy Nginx génération Let's Encrypt

Cette procédure décrit l'ajout d'un nouveau sous-domaine sur le reverse proxy Nginx, avec :

- création de la configuration Nginx dans `/etc/nginx/conf.d/` ;
- redirection HTTP vers HTTPS ;
- certificat Let's Encrypt ;
- authentification HTTP Basic via `.htpasswd` ;
- accès sans authentification depuis le LAN ;
- proxy vers un serveur interne ;
- prise en charge des WebSockets ;
- tests DNS, Nginx, TLS et backend.

1. Architecture

Le principe est le suivant :



Dans cette infrastructure, les configurations Nginx sont stockées directement dans :

```
/etc/nginx/conf.d/
```

Un fichier est créé par sous-domaine.

Exemple :

```
/etc/nginx/conf.d/browser.numericare.fr.conf
```

2. Informations nécessaires avant de commencer

Déterminer les éléments suivants :

Élément	Exemple
Sous-domaine	<code>browser.numericare.fr</code>
Serveur backend	<code>10.151.151.229</code>
Port backend	<code>3001</code>
Protocole backend	<code>HTTPS</code>
Fichier Nginx	<code>/etc/nginx/conf.d/browser.numericare.fr.conf</code>
Authentification	<code>/etc/nginx/.htpasswd</code>

Avant de commencer, confirmer également que le serveur backend est accessible depuis le reverse proxy.

3. Vérifier le DNS

Créer ou modifier l'enregistrement DNS du sous-domaine.

Le sous-domaine doit pointer vers l'adresse IP publique qui arrive sur le reverse proxy.

Exemple :

```
browser.numericare.fr → IP publique du reverse proxy
```

Ne pas faire pointer directement le DNS vers l'adresse privée du backend :

```
❌ browser.numericare.fr → 10.151.151.229
```

Depuis le reverse proxy, vérifier la résolution :

```
dig +short browser.numericare.fr
```

ou :

```
nslookup browser.numericare.fr
```

Le résultat doit correspondre à l'adresse IP publique du reverse proxy.

4. Vérifier le backend

Depuis le reverse proxy, tester la connectivité vers le serveur interne.

Pour un backend HTTPS :

```
curl -vk https://10.151.151.229:3001/
```

Pour un backend HTTP :

```
curl -v http://10.151.151.229:3001/
```

Tester également le port :

```
nc -vz 10.151.151.229 3001
```

Le résultat attendu est similaire à :

```
Connection to 10.151.151.229 3001 port [tcp/*] succeeded!
```

Si le backend n'est pas accessible, ne pas poursuivre la configuration du reverse proxy.

5. Vérifier le répertoire Let's Encrypt

Le répertoire utilisé pour les challenges ACME doit exister :

```
sudo ls -ld /var/www/certbot
```

S'il n'existe pas :

```
sudo mkdir -p /var/www/certbot
sudo chown -R www-data:www-data /var/www/certbot
```

6. Créer le fichier Nginx

Créer le fichier correspondant au sous-domaine :

```
sudo nano /etc/nginx/conf.d/SOUS-DOMAINE.numericare.fr.conf
```

Exemple pour `browser.numericare.fr` :

```
sudo nano /etc/nginx/conf.d/browser.numericare.fr.conf
```

7. Configuration HTTP / HTTPS

Utiliser la structure suivante :

```
map $http_upgrade $connection_upgrade {
    default upgrade;
    ''      close;
}

server {
    listen 80;
    server_name SOUS-DOMAINE.numericare.fr;

    # Let's Encrypt ACME HTTP-01
    location ^~ /.well-known/acme-challenge/ {
        root /var/www/certbot;
        try_files $uri =404;
    }

    # Tout le reste → HTTPS
    location / {
        return 301 https://$host$request_uri;
    }
}
```

```

server {
    listen 443 ssl proxy_protocol;
    server_name SOUS-DOMAINE.numericare.fr;

    # =====
    # SSL
    # =====
    ssl_certificate /etc/letsencrypt/live/SOUS-DOMAINE.numericare.fr/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/SOUS-DOMAINE.numericare.fr/privkey.pem;
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;
    include /etc/letsencrypt/options-ssl-nginx.conf;
    keepalive_timeout 70;

    # =====
    # REAL IP
    # =====
    real_ip_header proxy_protocol;

    set_real_ip_from 10.151.151.230;
    set_real_ip_from 10.151.151.228;
    set_real_ip_from 10.151.151.102;

    # =====
    # AUTHENTICATION
    # =====
    satisfy any;

    # Pas de mot de passe depuis le LAN
    allow 10.151.151.0/24;

    # Authentification obligatoire depuis l'extérieur
    deny all;

    auth_basic "Restricted Access";
    auth_basic_user_file /etc/nginx/.htpasswd;

    # =====
    # PROXY
    # =====
    location / {

```

```
proxy_pass https://10.151.151.xxx:PORT;

proxy_buffering off;

proxy_set_header Host $http_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;

# WebSockets
proxy_set_header Upgrade $http_upgrade;
proxy_set_header Connection $connection_upgrade;
proxy_http_version 1.1;
}
}
```

Important : si plusieurs fichiers Nginx utilisent le bloc :

```
map $http_upgrade $connection_upgrade {
    default upgrade;
    ''      close;
}
```

le `map` ne doit être déclaré **qu'une seule fois** dans toute la configuration Nginx.

Vérifier avant d'en ajouter un nouveau :

```
sudo grep -R "map \$http_upgrade" /etc/nginx/
```

Si le `map` existe déjà, ne pas le recopier dans le nouveau fichier.

8. Ne pas recharger Nginx immédiatement

Si le certificat n'existe pas encore, la configuration HTTPS référence des fichiers inexistants :

```
/etc/letsencrypt/live/SOUS-DOMAINE.numericare.fr/fullchain.pem
/etc/letsencrypt/live/SOUS-DOMAINE.numericare.fr/privkey.pem
```

Il est donc préférable de générer le certificat avant le premier reload de cette nouvelle configuration.

9. Générer le certificat Let's Encrypt

Utiliser Certbot avec le webroot :

```
sudo certbot certonly \  
  --webroot \  
  -w /var/www/certbot \  
  -d SOUS-DOMAINE.numericare.fr
```

Exemple :

```
sudo certbot certonly \  
  --webroot \  
  -w /var/www/certbot \  
  -d browser.numericare.fr
```

Certbot doit terminer avec un message indiquant que le certificat a été obtenu.

10. Vérifier le certificat

```
sudo ls -l /etc/letsencrypt/live/SOUS-DOMAINE.numericare.fr/
```

Les fichiers suivants doivent notamment exister :

```
cert.pem  
chain.pem  
fullchain.pem  
privkey.pem
```

11. Tester la configuration Nginx

Avant tout reload :

```
sudo nginx -t
```

Le résultat attendu :

```
syntax is ok
test is successful
```

En cas d'erreur, **ne pas recharger Nginx**.

12. Recharger Nginx

Si le test est correct :

```
sudo systemctl reload nginx
```

Vérifier que Nginx fonctionne toujours :

```
sudo systemctl status nginx --no-pager
```

Le service doit être :

```
Active: active (running)
```

13. Tester le HTTPS

Depuis le reverse proxy :

```
curl -vk https://SOUS-DOMAINE.numericare.fr/
```

Exemple :

```
curl -vk https://browser.numericare.fr/
```

La connexion TLS doit être établie et le certificat doit correspondre au sous-domaine.

14. Vérifier le certificat avec OpenSSL

```
openssl s_client \
  -connect SOUS-DOMAINE.numericare.fr:443 \
  -servername SOUS-DOMAINE.numericare.fr
```

Vérifier que le certificat présenté correspond au sous-domaine.

15. Tester l'accès depuis le LAN

Depuis une machine située dans le réseau `10.151.151.0/24`, ouvrir :

```
https://SOUS-DOMAINE.numericare.fr
```

Grâce à :

```
satisfy any;
allow 10.151.151.0/24;
deny all;

auth_basic "Restricted Access";
auth_basic_user_file /etc/nginx/.htpasswd;
```

l'accès depuis le LAN doit être autorisé **sans demande de mot de passe**.

16. Tester l'accès depuis Internet

Utiliser un réseau extérieur au LAN, par exemple une connexion mobile 4G/5G.

Ouvrir :

```
https://SOUS-DOMAINE.numericare.fr
```

Le navigateur doit demander les identifiants HTTP Basic.

Les identifiants sont ceux présents dans :

```
/etc/nginx/.htpasswd
```

17. Tester les WebSockets

Si l'application utilise les WebSockets, vérifier son fonctionnement directement dans l'application.

La configuration suivante doit être présente :

```
proxy_set_header Upgrade $http_upgrade;
proxy_set_header Connection $connection_upgrade;
proxy_http_version 1.1;
```

Elle permet notamment aux applications interactives utilisant WebSocket de fonctionner correctement derrière Nginx.

18. Vérifier les logs

En cas de problème :

```
sudo tail -f /var/log/nginx/access.log
```

Dans un autre terminal :

```
sudo tail -f /var/log/nginx/error.log
```

Puis refaire une connexion au sous-domaine.

19. Diagnostic rapide

Symptôme	Vérification
DNS incorrect	<code>dig +short SOUS-DOMAIN.numericare.fr</code>
Backend inaccessible	<code>nc -vz IP PORT</code>
Backend HTTPS	<code>curl -vk https://IP:PORT/</code>
Erreur Nginx	<code>sudo nginx -t</code>
Erreur TLS	<code>openssl s_client ...</code>
Erreur proxy	<code>sudo tail -f /var/log/nginx/error.log</code>
Erreur HTTP Basic	Vérifier <code>/etc/nginx/.htpasswd</code>
WebSocket	Vérifier <code>Upgrade</code> et <code>Connection</code>

20. Vérification finale

Exécuter :

```
echo "=== DNS ==="
dig +short SOUS-DOMAINE.numericare.fr

echo "=== NGINX ==="
sudo nginx -t

echo "=== CERTIFICAT ==="
sudo ls -l /etc/letsencrypt/live/SOUS-DOMAINE.numericare.fr/

echo "=== SERVICE NGINX ==="
sudo systemctl is-active nginx

echo "=== BACKEND ==="
nc -vz IP_BACKEND PORT_BACKEND

echo "=== HTTPS ==="
curl -Ik https://SOUS-DOMAINE.numericare.fr/
```

Le nouveau sous-domaine est considéré comme opérationnel lorsque :

- le DNS pointe vers l'IP publique du reverse proxy ;
- le backend est accessible depuis Nginx ;
- le certificat Let's Encrypt est valide ;
- `nginx -t` est OK ;
- Nginx est actif ;
- le HTTPS fonctionn

Revision #5

Created 2026-06-26 09:43:12 UTC by Nico là

Updated 2026-09-11 07:24:39 UTC by Nico là