Déploiement et configuration de SearXNG

Documentation rédigée par : GADONNAUD Ewen **Formation :** BTS SIO 1ère année - Option SISR **Établissement :** Lycée Paul-Louis Courier, Tours

Date : Mars 2026

1. Présentation et préparation du répertoire

SearXNG est un métamoteur de recherche respectueux de la vie privée. Le déploiement s'effectue via des conteneurs Docker sur le serveur. Nous déploierons alors SearXNG à l'aide d'une image docker dédiée, sur notre infrastructure X200 + VPS, à l'adresse search.ewengadonnaud.xyz Il convient donc de la création d'un record A sur notre registrar au préalable, pointant vers l'adresse du VPS. Création des répertoires nécessaires à l'hébergement de la configuration :

```
sudo mkdir -p /opt/searxng/searxng # Chemin docker couramment utilisé
cd /opt/searxng
```

2. Configuration des paramètres de SearXNG

Création du fichier de configuration principal `settings.yml` permettant de définir le port d'écoute et les paramètres de sécurité :

```yaml
use_default_settings: true
server:
  port: 8080
  bind_address: "0.0.0.0"
  secret_key: "INSERER_CLE_SECRETE_ICI"
  base_url: "https://search.ewengadonnaud.xyz/"
  image_proxy: true
```

Création du fichier `docker-compose.yml`. L'ajout de serveurs DNS publics de manière explicite est nécessaire afin de pallier les erreurs de résolution de noms (temporary failure in name resolution) lors de l'initialisation des moteurs de recherche par le conteneur :

```yaml
version: '3'
services:
  searxng:
    image: searxng/searxng:latest
```

```
    container_name: searxng
    ports:
      - "8080:8080"
    volumes:
      - ./searxng:/etc/searxng
    restart: unless-stopped
    dns:
      - 1.1.1.1
      - 8.8.8.8
```

Lancement de l'infrastructure via Docker :

```
sudo docker compose up -d
```

3. Configuration du Reverse Proxy Nginx

Création du bloc serveur Nginx sur le VPS pour relayer le trafic via le tunnel WireGuard vers le port 8087 de la machine cible :

```
server {
    server_name search.ewengadonnaud.xyz;
    location / {
        proxy_pass http://X.X.X.X: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;
    }
}
```

4. Sécurisation SSL/TLS avec Certbot

Afin de garantir la confidentialité des requêtes (HTTPS), nous allons générer un certificat SSL gratuit via Let's Encrypt en utilisant l'outil Certbot. Installation de Certbot et de son plugin pour Nginx sur le VPS :

```
sudo apt update
sudo apt install python3-certbot-nginx
```

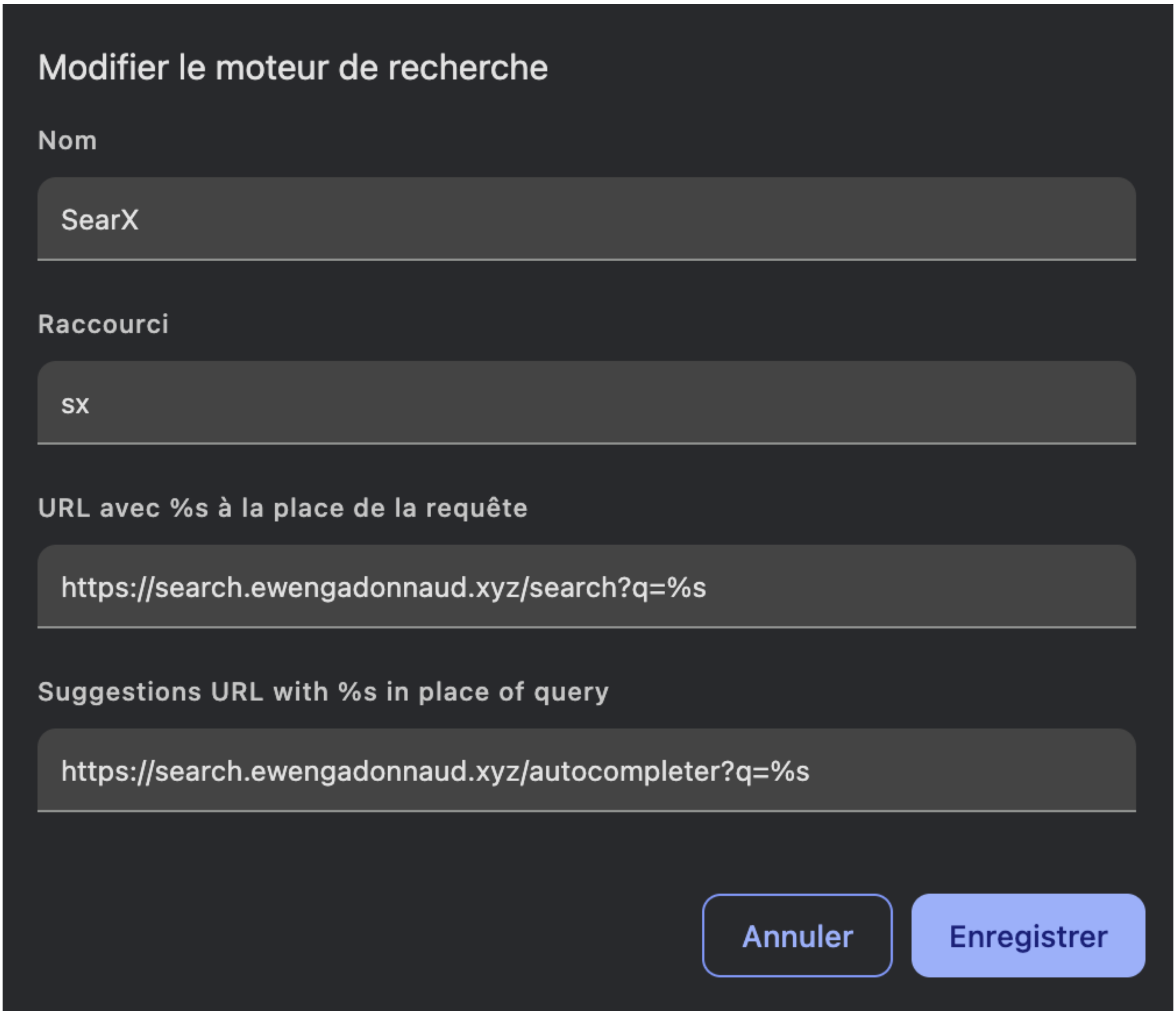
Génération du certificat et configuration automatique du Reverse Proxy Nginx :

```
sudo certbot --nginx -d search.ewengadonnaud.xyz
```

Certbot va automatiquement détecter le bloc serveur Nginx configuré à l'étape précédente, le modifier pour y inclure les certificats de sécurité, et forcer la redirection du trafic de HTTP vers HTTPS.

5. Intégration au navigateur web

Pour configurer SearXNG comme moteur de recherche par défaut au sein du navigateur client : * **Nom :** SearXNG * **Raccourci :** sx * **URL avec %s à la place de la requête :** `https://search.ewengadonnaud.xyz/search?q=%s`



6. Activation des favicons (optionnel)

Par défaut, SearXNG n'affiche pas les favicons des sites dans les résultats de recherche. Cette fonctionnalité est désactivée car elle génère une charge serveur supplémentaire. Son activation nécessite trois modifications.

L'activation des favicons augmente significativement la charge en requêtes client/serveur. À ne configurer que si les ressources du serveur le permettent.

Étape 1 — Ajout du resolver dans `settings.yml`

Ajouter une section `search` dédiée dans le fichier `settings.yml` (attention : ne pas placer cette clé dans la section `server`) :

```yaml
search:
  favicon_resolver: "duckduckgo"
```

Les resolvers disponibles sont : `duckduckgo`, `allesedv`, `google`, `yandex`.

Étape 2 — Création du fichier de cache `favicons.toml`

Créer le fichier `/opt/searxng/searxng/favicons.toml`. Ce fichier configure le cache SQLite qui stocke les icônes localement afin de limiter les requêtes externes répétées :

```toml
[favicons]
cfg_schema = 1

[favicons.cache]
db_url = "/var/cache/searxng/faviconcache.db"
LIMIT_TOTAL_BYTES = 2147483648
```

Étape 3 — Déclaration du volume de cache dans `docker-compose.yml`

Le cache doit être persistant entre les redémarrages du conteneur. Modifier le `docker-compose.yml` pour y ajouter un volume nommé dédié :

```yaml
version: '3'
services:
  searxng:
    image: searxng/searxng:latest
    container_name: searxng
    ports:
      - "8080:8080"
    volumes:
      - ./searxng:/etc/searxng
      - searxng-cache:/var/cache/searxng
    restart: unless-stopped
    dns:
      - 1.1.1.1
      - 8.8.8.8

volumes:
  searxng-cache:
```

Redémarrer le conteneur pour appliquer l'ensemble des modifications :

```
docker compose restart
```

From:
https://wiki.ewengadonnaud.xyz/ - **Base de savoir réseaux/cyber/devops**

Permanent link:
https://wiki.ewengadonnaud.xyz/doku.php?id=services:searxng&rev=1773145886

Last update: **2026/03/10 13:31**