TUTOS-INFO
L'informatique par l'exemple

TUTORIEL

REVERSE PROXY

Installer Nginx Proxy Manager

NGINX
PROXY MANAGER

SOMMAIRE

1 – NGINX PROXY MANAGER, C'EST QUOI ?

2 – INSTALLATION DE DOCKER ET PORTAINER CE SUR LA MACHINE « REBOND »

3 – INSTALLATION DE NGINX PROXY MANAGER EN FRONTEND

4 – PREMIER ACCES A L'INTERFACE WEB DE NGINX PROXY MANAGER

5 – CONFIGURATION DU ROUTAGE SUR LA MACHINE DE « REBOND »

6 – GENERER DES CERTIFICATS LET'S ENCRYPT DEPUIS NGINX PROXY MANAGER

7 – CONFIGURATION DU BACKEND

DIFFICULTE

1 – NGINX PROXY MANAGER, C'EST QUOI ?

Nginx proxy manager (NPM) est un **reverse proxy** qui fonctionne sur **Docker**. NPM est basé sur un serveur Nginx et fournit aux utilisateurs une interface web intuitive permettant une gestion simplifiée de la configuration du frontend et du backend.

L'outil est facile à configurer et ne nécessite pas de connaissances approfondies sur Nginx. Il est capable de générer des certificats SSL Let's Encrypt. NPM est un outil open-source adapté aux environnements de moyenne envergure et aux environnements « home lab ».

Pour réaliser ce tutoriel, **nous disposons d'une machine physique Linux** (Debian par exemple) **avec 2 interfaces réseau**. Elle servira de « **rebond** », sur laquelle nous allons installer Docker. Notre environnement de test est le suivant :

- Une machine virtuelle Debian 12 (2 Go de RAM et 30 Go de disque)
- **2 cartes réseau** sur la machine Debian : l'une servira d'interface « **WAN** » et l'autre d'interface « **LAN** »
- Un accès à Internet

Attention, ce tutoriel présente les commandes saisies avec l'utilisateur « root » afin de ne pas surcharger les lignes de commandes (environnement de test). Ajoutez « sudo » si vous utilisez un utilisateur du système.

2 – INSTALLATION DE DOCKER ET PORTAINER SUR LA MACHINE DEBIAN 12

1ère étape : configuration des interfaces réseau

- Connectez-vous à votre machine Debian avec un utilisateur disposant de droits suffisants (ou en « root »)
- Configurer les cartes réseau de manière statique comme ceci par exemple :

```
# Carte 1 - WAN
auto enp2s0
iface enp2s0 inet static
address 192.168.3.250/24
gateway 192.168.3.254
dns-nameservers 9.9.9.9 1.1.1.1

# Carte 2 - lan
auto enp3s0
iface enp3s0 inet static
address 192.168.168.254
```

2ème étape : installation de Docker et Docker compose

- Redémarrez les interfaces réseau avec la commande : **systemctl restart networking**
- Mettez à jour les dépôts de votre machine Debian 12 : **apt update && apt upgrade -y**
- Installez « **curl** » sur la machine Debian 12 : **apt install curl -y**
- Installez Docker sur la machine Debian 12 : **curl -fsSL https://get.docker.com | sh**

Remarque : ici nous installons Docker avec le script officiel fourni par l'éditeur (environnement de laboratoire). Si vous préférez installer Docker sans utiliser le script, veuillez vous référer à la documentation officielle ici : Debian | Docker Docs

3ème étape : installation de PortainerCE (non obligatoire)

PortainerCE permet de gérer, avec une interface graphique, votre environnement Docker. Il n'est pas obligatoire d'installer PortainerCE ici. Avant de lancer l'installation de PortainerCE, **vous devez ouvrir le port « 9443 » sur votre routeur** et le faire pointer vers votre machine Debian.

L'installation de PortainerCE s'effectue de la manière suivante (saisissez les commandes ci-dessous ; attention, la 2ème commande est à saisir sur une seule ligne) :

```
docker volume create portainer_data
```

```
docker run -d -p 8000:8000 -p 9443:9443 --name portainer --restart=always -v /var/run/docker.sock:/var/run/docker.sock -v portainer_data:/data portainer/portainer-ce:latest
```

Une fois le processus terminé, vous pouvez vous connecter à l'interface de PortainerCE depuis un navigateur web. Lors de la première connexion, il vous sera demandé de créer un compte administrateur avec un mot de passe.

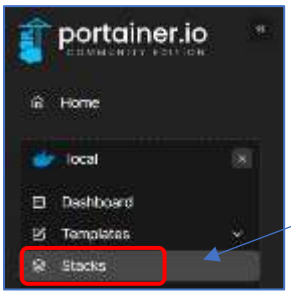
Adresse de connexion à l'interface PortainerCE via votre navigateur : **https://votre_ip_ou_domaine:9443**

3 – CONFIGURATION DU FRONTEND - CREATION DE LA STACK « NGINX PROXY MANAGER »

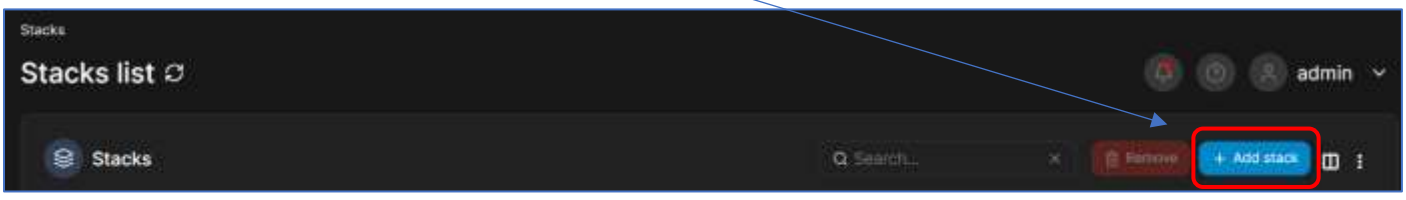
- On commence par la configuration du « frontend » (1ère interface réseau de notre machine Debian)
- Créez, sur la machine Debian, un réseau Docker dédié à NPM avec la commande : **docker network create npm**
- Créez la stack « NPM ». Pour cela vous pouvez le faire de 2 façons :
    - Méthode 1 : création de la stack depuis PortainerCE (avec le Web editor)
    - Méthode 2 : création en ligne de commande sur la machine Debian

Méthode 1 - Création de la stack « NPM » depuis PortainerCE

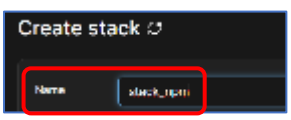
- Connectez-vous à PortainerCE
- Depuis votre environnement « **Local** », cliquez, dans le volet de gauche, sur « **Stacks** » :



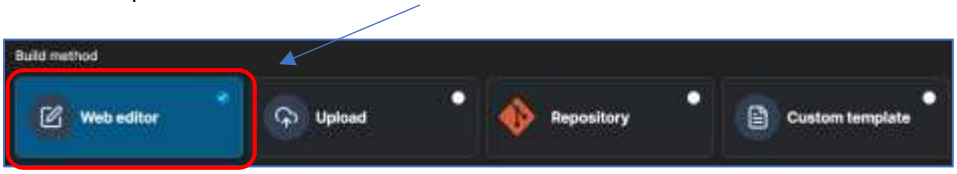
- Dans le volet de droite, cliquez le bouton bleu « **Add stack** » :



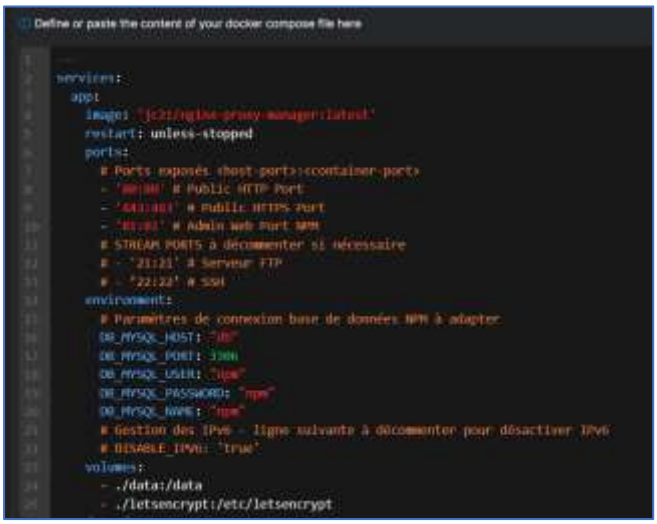
- Commencez par nommer votre stack (rubrique « **Name** ») :



- Cliquez le bouton bleu « **Web editor** » :



- Coller le texte suivant dans l'éditeur Portainer :

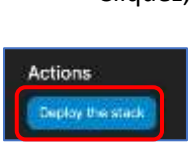



```
---
services:
  app:
    image: 'jc21/nginx-proxy-manager:latest'
    restart: unless-stopped
    ports:
      # Ports exposés <host-port>:<container-port>
      - '80:80' # Public HTTP Port
      - '443:443' # Public HTTPS Port
      - '81:81' # Admin Web Port NPM
      # STREAM PORTS à décommenter si nécessaire
      # - '21:21' # Serveur FTP
      # - '22:22' # SSH
    environment:
      # Paramètres de connexion base de données NPM à adapter
      DB_MYSQL_HOST: "db"
      DB_MYSQL_PORT: 3306
      DB_MYSQL_USER: "npm"
      DB_MYSQL_PASSWORD: "npm"
      DB_MYSQL_NAME: "npm"
      # Gestion des IPv6 - ligne suivante à décommenter pour désactiver IPv6
      # DISABLE_IPV6: 'true'
    volumes:
      - ./data:/data
      - ./letsencrypt:/etc/letsencrypt
    depends_on:
      - db

  db:
    image: 'jc21/mariadb-aria:latest'
    restart: unless-stopped
    environment:
      MYSQL_ROOT_PASSWORD: 'npm'
      MYSQL_DATABASE: 'npm'
      MYSQL_USER: 'npm'
      MYSQL_PASSWORD: 'npm'
      MARIADB_AUTO_UPGRADE: '1'
    volumes:
      - ./mysql:/var/lib/mysql

networks:
  default:
    external: true
    name: npm
```

- Cliquez, dans le bas de la fenêtre, le bouton « **Deploy the stack** » et patientez pendant le déploiement :



Depuis l'interface de PortainerCE, vous pouvez vérifier que 2 nouveaux conteneurs sont apparus et actifs :

NPM est maintenant installé.

Méthode 2 - Création du fichier « docker-compose.yml »)

Vous pouvez créer votre stack en ligne de commande en suivant les manipulations suivantes :

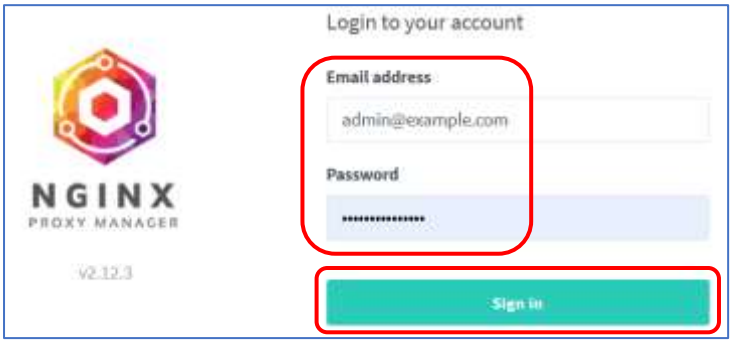
- Depuis votre machine Debian, créez un dossier (emplacement de votre choix) : **mkdir npm**
- Ouvrez l'éditeur nano avec : **nano docker-compose.yml** (le fichier doit porter ce nom !)
- Copiez toutes les lignes de la page précédente dans le fichier, quittez et enregistrez le fichier
- Lancez la création des conteneurs, depuis le dossier où se trouve le fichier « docker-compose.yml » avec la commande : **docker compose up -d**

Depuis l'interface de PortainerCE, vous pouvez vérifier que 2 nouveaux conteneurs sont apparus et actifs :

NPM est maintenant installé.

4 – ACCES A L'INTERFACE WEB DE GESTION DE NGINX PROXY MANAGER

- Ouvrez un navigateur et saisissez l'IP de votre serveur Debian (ou votre nom de domaine) suivie du port « 81 »
- Par exemple, pour une adresse locale, on aura : http://192.168.10.12:81
- La page d'authentification s'affiche ; saisissez « **admin@example.com** » et « **changeme** » et cliquez « **Sign in** » :



Lors de la 1ère connexion, saisissez le mail par défaut « admin@example.com » et le mot de passe par défaut « changeme » ; l'étape suivante vous demandera de changer ces identifiants par mesure de sécurité.

5 – CONFIGURATION DU ROUTAGE SUR LA MACHINE DEBIAN 12

Dans ce tutoriel, nous souhaitons accéder à 3 serveurs Proxmox qui sont connectés sur la 2ème interface réseau de notre machine Debian (le « backend »). Nous devons, pour commencer, activer le routage entre les 2 interfaces réseau de la machine Debian.

Pour rappel, la 1ère interface réseau de notre machine Debian nommée « enp2s0 » correspond au « frontend » et la seconde interface nommée « enp3s0 » correspond au « backend ».

1ère étape : activation du routage

- Activez le routage en éditant le fichier « **sysctl.conf** » : **nano /etc/sysctl.conf**
- **Décommentez la ligne « net.ipv4.ip_forward = 1 »**
- Quittez et sauvegardez les modifications
- Relancez les interfaces avec la commande : **systemctl restart networking**

2ème étape : activation et configuration des règles iptables pour le NAT

Vous devez configurer les règles iptables pour permettre le NAT entre les deux interfaces réseau. Pour cela, utilisez les commandes suivantes :

- **iptables -t nat -A POSTROUTING -o enp2s0 -j MASQUERADE**
- **iptables -A FORWARD -i enp3s0 -o enp2s0 -j ACCEPT**
- **iptables -A FORWARD -i enp2s0 -o enp3s0 -m state --state RELATED,ESTABLISHED -j ACCEPT**

La première commande permet de masquer les adresses IP des serveurs Proxmox lorsqu'ils sortent sur Internet via l'interface enp2s0. Les deux autres commandes permettent le transfert des paquets entre les interfaces enp3s0 et enp2s0.

3ème étape : rendre les règles iptables persistantes

Pour que les règles iptables soient appliquées après un redémarrage de la machine, vous devez les sauvegarder. Sur Debian, vous pouvez utiliser le paquet iptables-persistent :

- **apt-get install iptables-persistent**
- **netfilter-persistent save**
- Relancez les interfaces avec la commande : **systemctl restart networking**

Assurez-vous que chaque serveur Proxmox utilise l'adresse IP de la 2ème interface réseau de la machine Debian 12 (192.168.168.254 dans notre cas) comme passerelle par défaut. Vous pouvez configurer cela dans le fichier de configuration réseau de chaque serveur (ou via l'interface de gestion de Proxmox).

Pour vérifier que les règles fonctionnent correctement, vous pouvez effectuer des tests de connectivité depuis les serveurs Proxmox en tentant de pinger le DNS 8.8.8.8 par exemple.

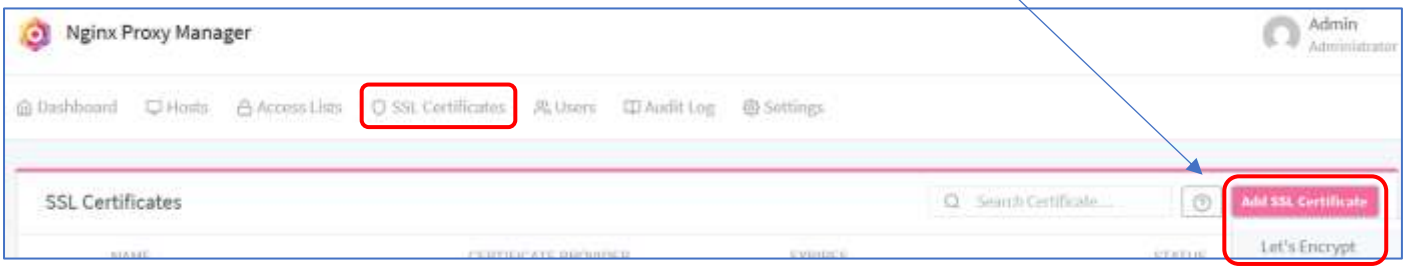
6 – GENERATION D'UN CERTIFICAT LET'S ENCRYPT (si vous possédez un domaine hébergé)

Nginx Proxy Manager est capable de générer automatiquement des certificats Let's Encrypt si vous possédez un nom de domaine hébergé.

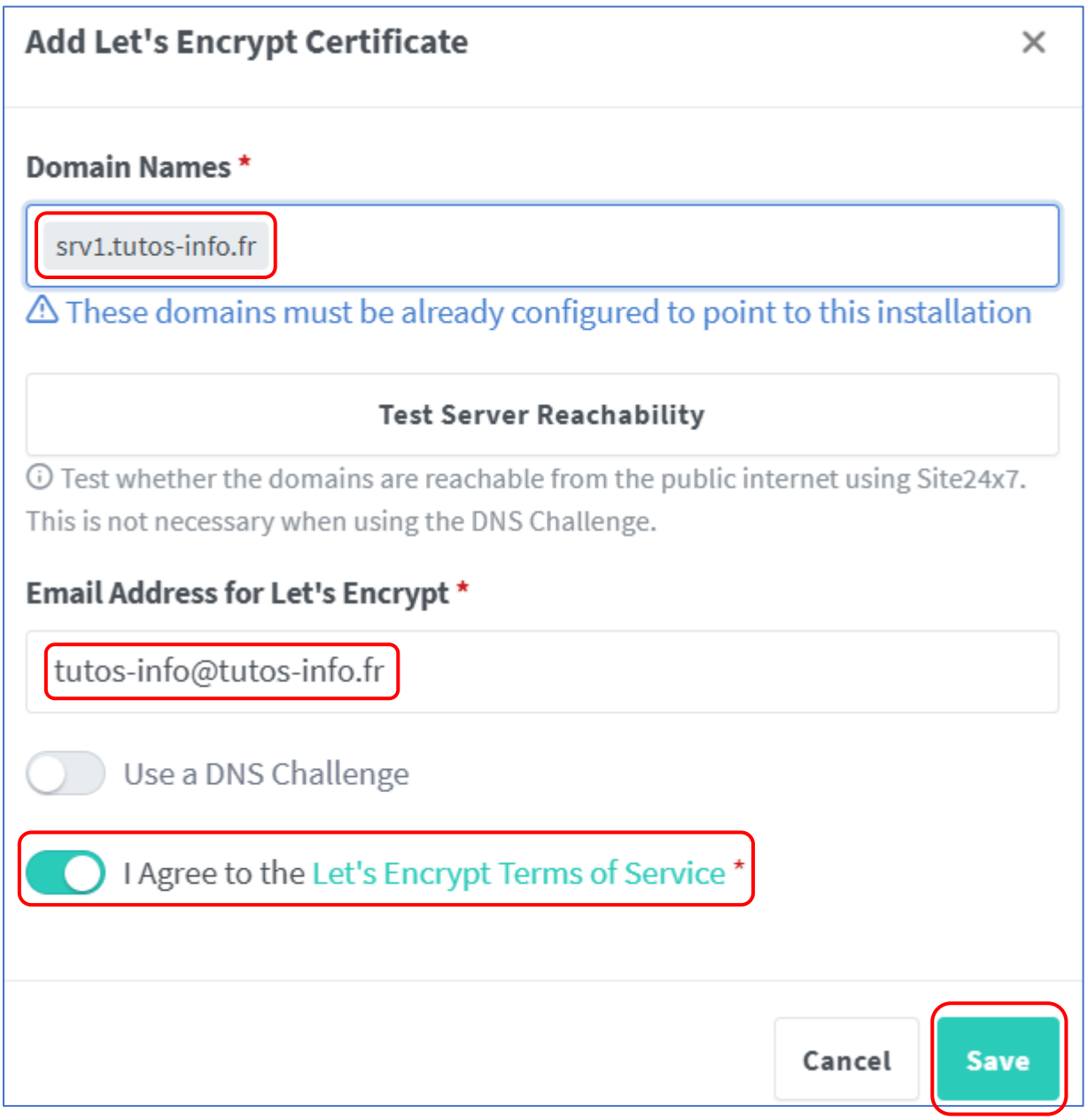
Si vous souhaitez générer un certificat Let's Encrypt, commencez par créer un enregistrement DNS de type « A » dans votre zone DNS (voir sur la console de gestion de votre hébergeur) et faites pointer cette entrée sur l'IP publique de votre serveur Nginx Proxy Manager.

Remarque : il est également possible de générer un certificat Let's Encrypt en utilisant le « Challenge DNS » mais, pour cela, il faut créer une API chez votre hébergeur, qui vous donnera des identifiants spécifiques qui permettront la génération d'un certificat Let's Encrypt (non étudié ici).

- Connectez-vous à l'interface de gestion de Nginx Proxy Manager
- Dans le menu du haut, cliquez sur « **SSL Certificates** »
- Cliquez, en haut à droite, le bouton « **Add SSL Certificate** » et « **Let's Encrypt** » :



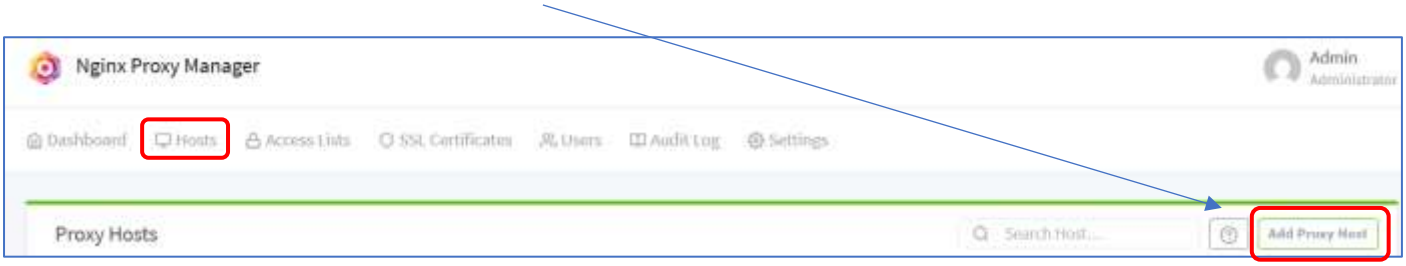
- Complétez la fenêtre et cliquez le bouton « **Save** » pour demander le certificat :



Patientez le temps de l'obtention du certificat Let's Encrypt. Si vous n'obtenez pas le certificat, vérifiez la configuration de votre zone DNS chez votre hébergeur (enregistrement de type « A » et adresse IP publique qui pointe vers le sous-domaine souhaité).

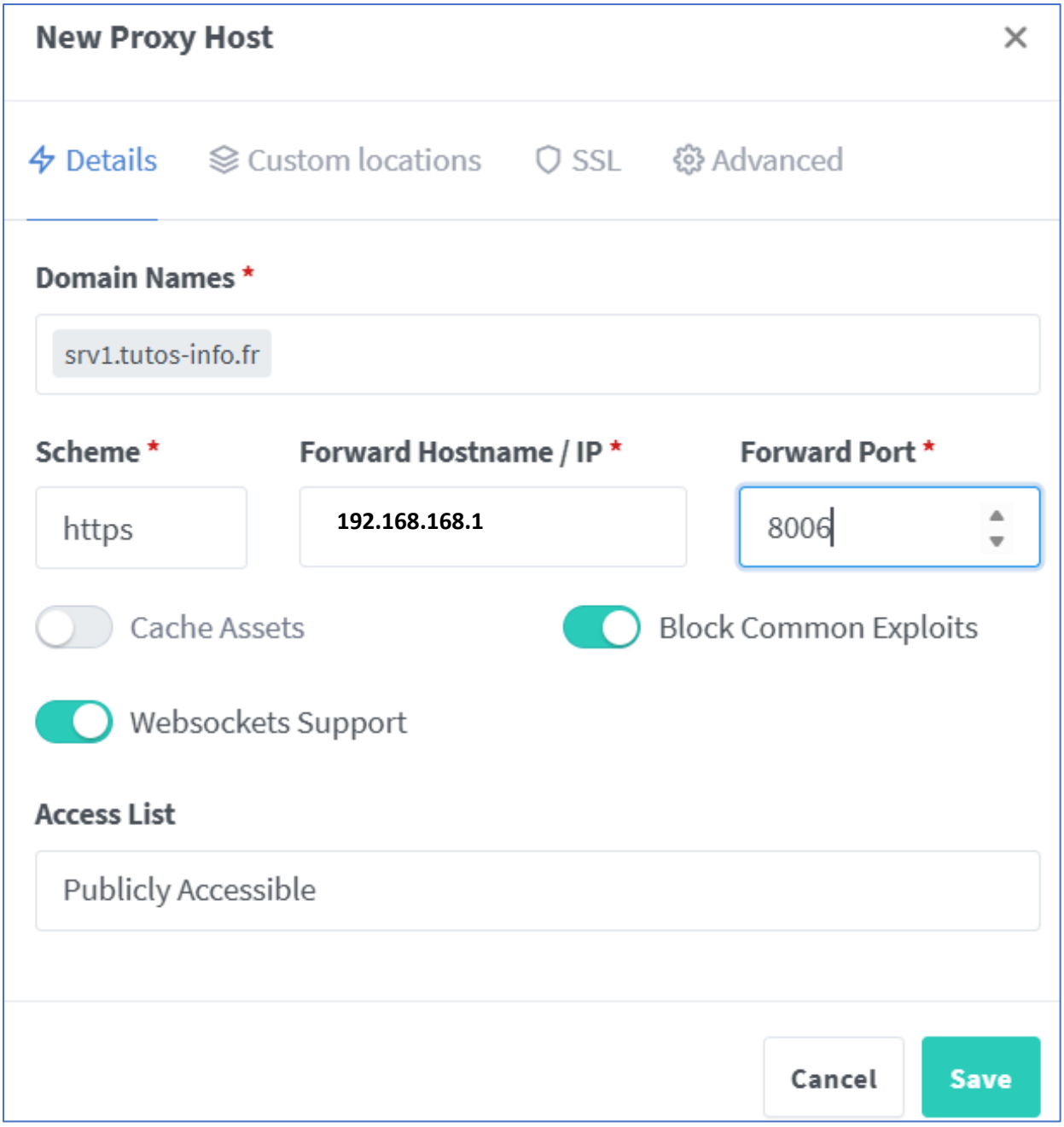
7 – CONFIGURATION DU BACKEND DANS NGINX PROXY MANAGER

- Connectez-vous à l'interface de gestion de Nginx Proxy Manager
- Cliquez le menu « **Hosts** » et « **Add Proxy Host** » en haut à droite :

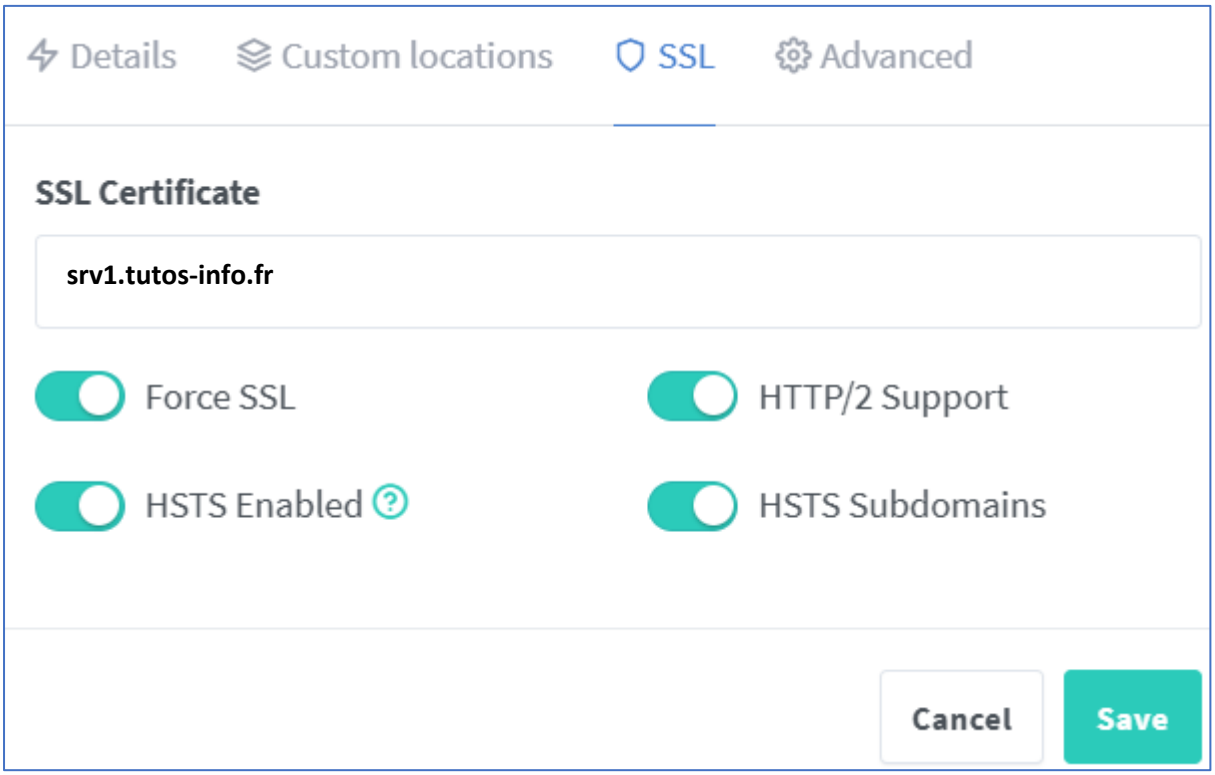


- Dans l'onglet « Détails », ajoutez le proxy host (1er serveur Proxmox) ainsi puis cliquez l'onglet « **SSL** » :

1 – ONGLET DETAILS



- Sélectionnez le certificat Let's Encrypt préalablement généré et cliquez le bouton « **Save** » :



Le serveur apparaît dans la liste des « Proxy Hosts ». Si vous cliquez sur l'hôte, vous devriez pouvoir accéder à l'interface web de votre 1er serveur Proxmox :

Répétez les opérations 6 et 7 pour ajouter les autres serveurs si nécessaire. Il est possible de créer des « Access List » pour des réglages personnalisées (autorisations et interdictions d'accès).

Vous trouverez plus d'informations sur ces options ici : Guide | Nginx Proxy Manager