PortailSIO

NOUVELLE FORMULE

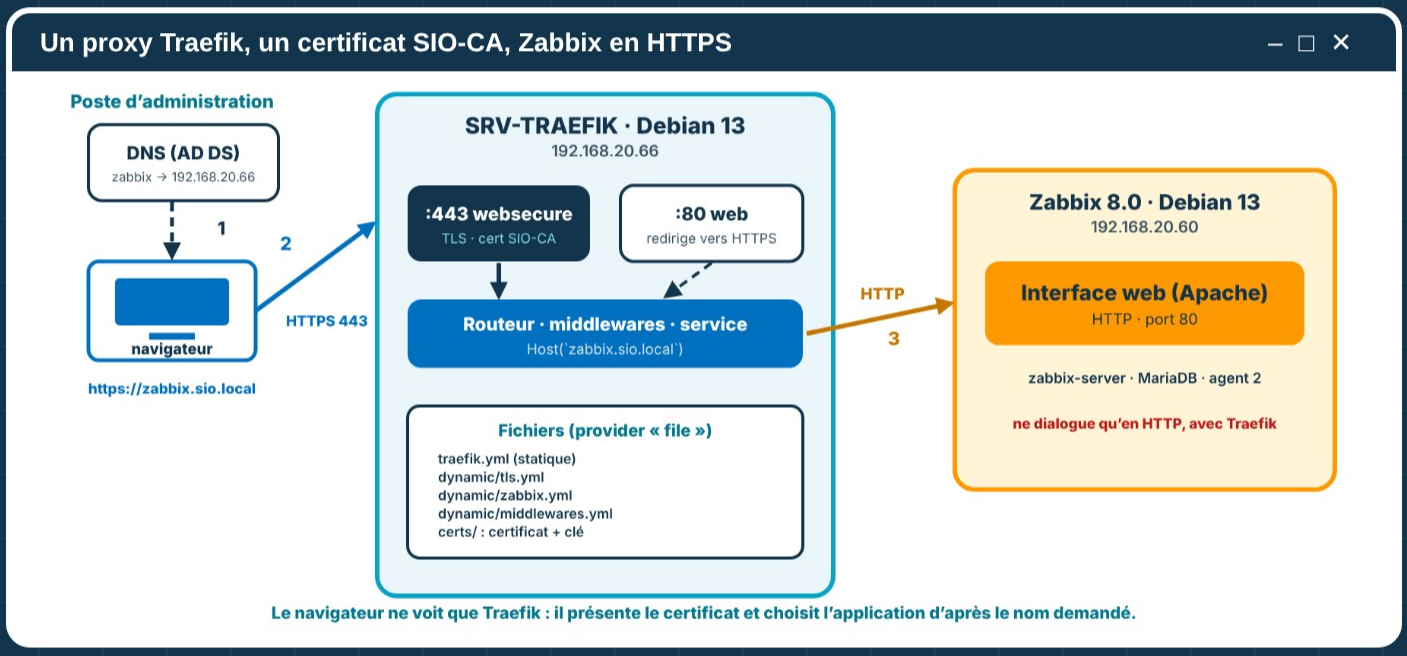
TUTORIEL

DEBIAN 13 · TRAEFIK 3.7 · HTTPS

Installer derrière Traefik

ZABBIX





1 Préparer les deux serveurs et le DNS

2 Installer Zabbix en natif sur Debian 13

3 Installer Traefik

4 Obtenir un certificat auprès de SIO-CA

5 Publier Zabbix avec les fichiers dynamiques de Traefik

6 Terminer l'installation et vérifier la chaîne

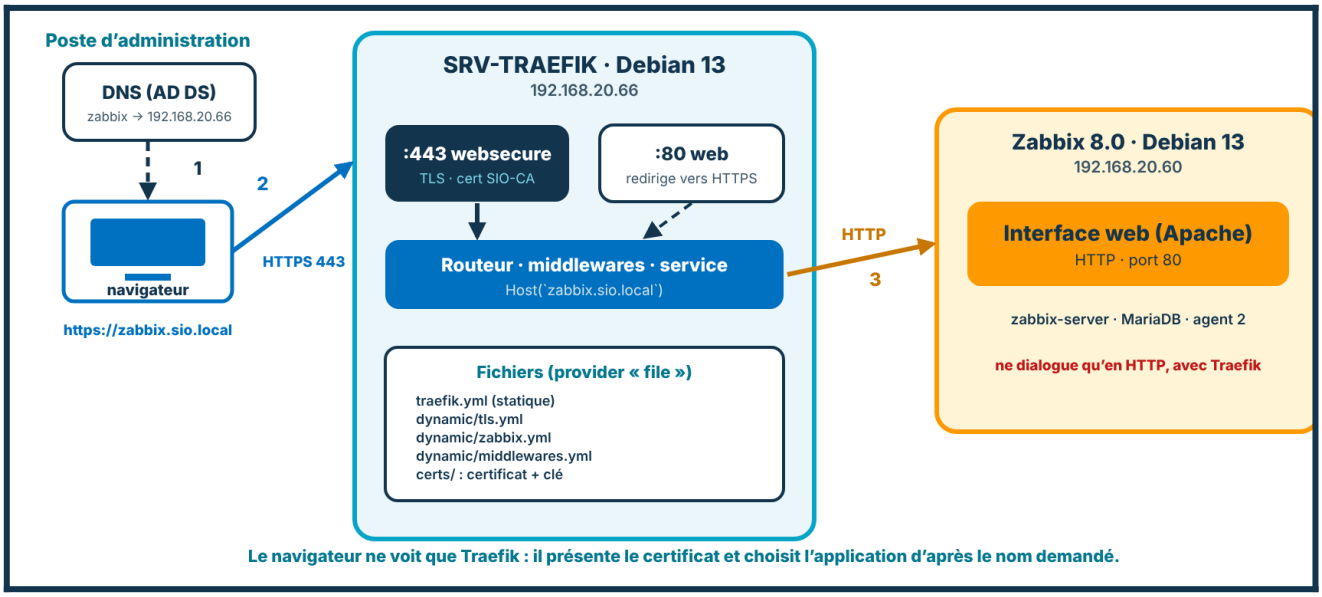
CC BY-NC
Utilisation commerciale interdite
© PortailSIO – 11/2026

Ce tutoriel en ligne
Scannez le QR code. Tous les tutoriels :
portail.sio-ndlp.fr/tutoriels.php

Ce tutoriel a été réalisé avec l'hyperviseur © Proxmox VE. Il peut être réalisé dans un environnement de virtualisation personnel à l'aide des logiciels © VirtualBox ou © VMware Player. Il **suppose que vous disposez de deux machines virtuelles Debian 13** (installation minimale avec serveur SSH) et du domaine **sio.local** construit dans les tutoriels précédents (contrôleurs **SRV-DC1** et **SRV-DC2** sous Windows Server 2025, avec l'autorité de certification **SIO-CA**). Le tutoriel est de niveau **expert** : il demande d'être à l'aise avec Linux, le DNS et les certificats.

Zabbix est une solution de **supervision** libre : elle collecte à intervalle régulier des mesures sur vos serveurs, les compare à des seuils et vous alerte avant que les utilisateurs ne s'aperçoivent de la panne. Son interface web gère les mots de passe de l'administrateur et des utilisateurs : elle doit donc être servie en **HTTPS**.

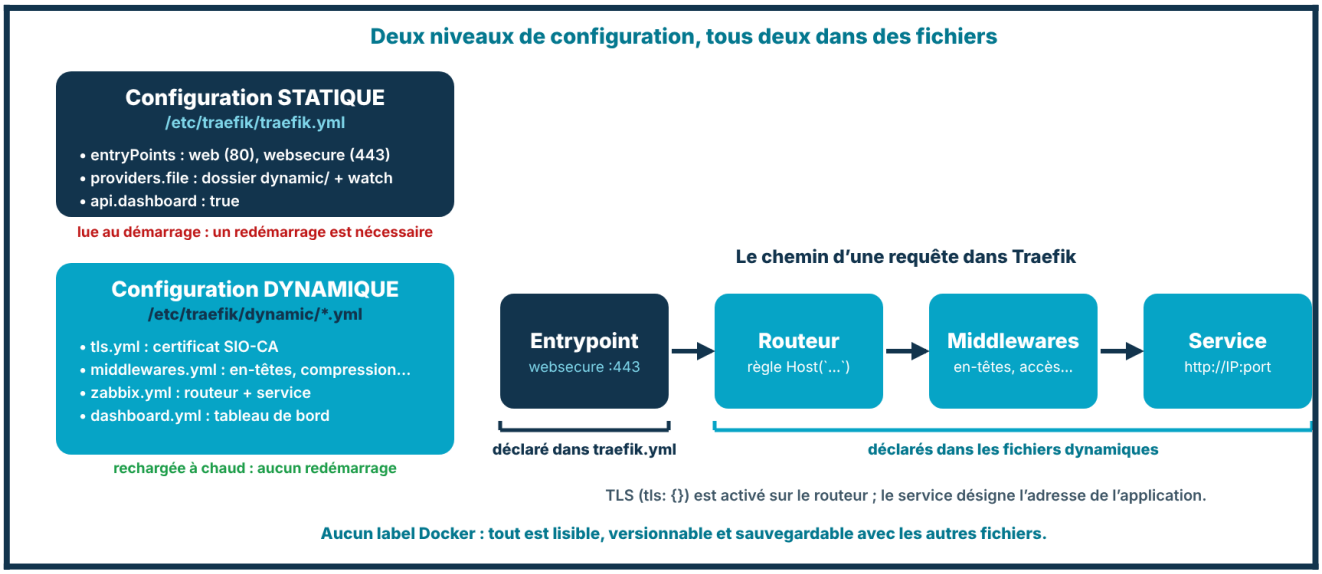
Traefik est un **proxy inverse** (en anglais *reverse proxy*) : un serveur placé **devant** les applications, qui reçoit toutes les connexions des utilisateurs et les transmet à la bonne application. Il apporte trois choses. Il **chiffre** les échanges (c'est lui qui présente le certificat HTTPS), il **aiguille** chaque requête d'après le nom demandé (`zabbix.sio.local`, `glpi.sio.local`…) et il ajoute des **protections** (en-têtes de sécurité, filtrage par adresse). L'application, elle, n'a plus à gérer le HTTPS ni à être exposée aux utilisateurs.



Le trajet d'une requête

- **1 – Le DNS** : le nom `zabbix.sio.local` est déclaré dans l'annuaire AD DS et pointe vers l'adresse de **Traefik**, pas vers celle de Zabbix
- **2 – HTTPS** : le navigateur se connecte à Traefik sur le port **443**. Traefik présente un certificat délivré par **SIO-CA**, auquel les postes du domaine font déjà confiance. Le port 80 existe aussi, mais il ne fait que renvoyer vers HTTPS
- **3 – HTTP** : Traefik transmet la requête à Zabbix, en clair, sur le réseau des serveurs. Seul Traefik peut joindre Apache : nous fermerons l'accès direct à l'étape 6

Traefik se configure avec **deux niveaux de fichiers**. La configuration **statique** (ports d'écoute, emplacement des autres fichiers) n'est lue qu'au démarrage. La configuration **dynamique** (certificats, routes, middlewares) est répartie en un **fichier par sujet** et rechargée à chaud. **Dans ce tutoriel, toute la configuration passe par ces fichiers : aucun label Docker n'est utilisé**, et Docker n'est d'ailleurs pas installé :



Dans le cadre de ce tutoriel, nous utilisons les machines suivantes (VLAN SERVEURS, réseau 192.168.20.0 /24) :

Machine	Rôle	Système	Adresse IP
SRV-TRAEFIK	Proxy inverse Traefik (nom `srv-traefik`) : seul point d'entrée des utilisateurs	Debian 13	192.168.20.66
SRV-ZBX	Serveur Zabbix : base MariaDB, Apache, serveur et agent (nom `srv-zbx`)	Debian 13	192.168.20.60
SRV-DC1	Contrôleur de domaine, DNS, autorité de certification SIO-CA	Windows Server 2025	192.168.20.250
SRV-DC2	Contrôleur de domaine, DNS	Windows Server 2025	192.168.20.251
Poste admin	Navigateur web (DNS = les deux contrôleurs, poste membre du domaine, réseau 192.168.10.0 /24)	Windows 11	DHCP

Prévoyez **2 vCPU, 4 Go de mémoire et 30 Go de disque** pour SRV-ZBX, et **1 vCPU, 1 Go de mémoire et 10 Go de disque** pour SRV-TRAEFIK. Ces valeurs, comme les adresses et les mots de passe d'exemple, sont à adapter à votre environnement. Les commandes Debian sont exécutées avec un compte autorisé à utiliser `sudo` (ou directement en `root`, sans le préfixe `sudo`) ; les commandes Windows, dans **PowerShell** ouvert **en tant qu'administrateur** sur **SRV-DC1**.

ÉTAPE 1 SUR 6

01 Préparer les deux serveurs et le DNS

Les utilisateurs ne connaîtront que **SRV-TRAEFIK** : les deux noms `zabbix.sio.local` et `traefik.sio.local` désigneront son adresse. Chaque serveur reçoit un nom, une adresse fixe, le bon fuseau horaire et les DNS du domaine.

A – Préparer SRV-ZBX

1. Mettez le système à jour : `sudo apt update && sudo apt full-upgrade -y`
2. Donnez son nom au serveur et déclarez-le dans le fichier `hosts` (nom complet et nom court) :

```
TERMINAL (SRV-ZBX)
sudo hostnamectl set-hostname srv-zbx
echo "192.168.20.60  srv-zbx.sio.local  srv-zbx" | sudo tee -a /etc/hosts
```

3. Réglez le fuseau horaire et générez les langues anglaise et française, nécessaires à l'interface web de Zabbix :

```
TERMINAL (SRV-ZBX)
sudo timedatectl set-timezone Europe/Paris
sudo apt install -y locales
sudo sed -i -e 's/^# *\(en_US.UTF-8 UTF-8\)/\1/' -e 's/^# *\(fr_FR.UTF-8 UTF-8\)/\1/'
/etc/locale.gen
sudo locale-gen
```

4. Donnez une **adresse IP fixe** (adaptez le nom de la carte, ici `ens18`, avec `ip -br address`) : éditez `/etc/network/interfaces`, indiquez les DNS du domaine dans `/etc/resolv.conf`, puis redémarrez le réseau avec `sudo systemctl restart networking` :

```
/etc/network/interfaces
auto ens18
iface ens18 inet static
    address 192.168.20.60/24
    gateway 192.168.20.254
```

```
/etc/resolv.conf
search sio.local
nameserver 192.168.20.250
nameserver 192.168.20.251
```

5. Vérifiez le résultat : nom, fuseau, heure synchronisée, langues et adresse IP :

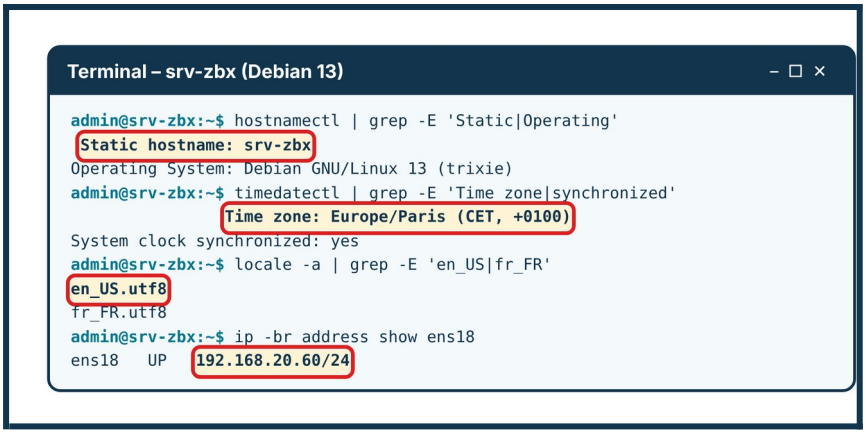


Figure 1 – Contrôle de la préparation de SRV-ZBX

À SAVOIR

Si « System clock synchronized » indique « no », installez un service de temps (`sudo apt install -y chrony`). L'heure doit être juste : Zabbix horodate chaque mesure et les certificats ont des dates de validité.

B – Déclarer les noms dans le DNS

Sur **SRV-DC1**, dans PowerShell (**administrateur**), créez les deux enregistrements. Ils pointent tous deux vers l'adresse de **Traefik** (192.168.20.66) ; l'enregistrement est répliqué sur SRV-DC2 avec la zone :

POWERSHELL (SRV-DC1)

```
Add-DnsServerResourceRecordA -ZoneName sio.local -Name zabbix `
    -IPv4Address 192.168.20.66
Add-DnsServerResourceRecordA -ZoneName sio.local -Name traefik `
    -IPv4Address 192.168.20.66
Resolve-DnsName zabbix.sio.local, traefik.sio.local -Type A | Select Name, Type, TTL,
IPAddress
```

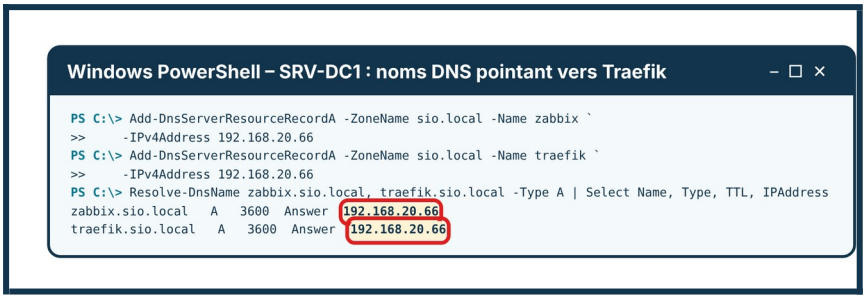


Figure 2 – PowerShell : noms DNS pointant vers Traefik

À SAVOIR

Le nom `zabbix` désigne le **service** rendu aux utilisateurs, pas la machine qui l'héberge : si Zabbix change de serveur un jour, seule la configuration de Traefik change, pas l'adresse que connaissent les utilisateurs.

C – Préparer SRV-TRAEFIK

1. Sur **SRV-TRAEFIK**, mettez le système à jour (`sudo apt update && sudo apt full-upgrade -y`), nommez le serveur, réglez le fuseau et installez `curl` et `tree`, utiles pour les contrôles :

TERMINAL (SRV-TRAEFIK)

```
sudo hostnamectl set-hostname srv-traefik
echo "192.168.20.66  srv-traefik.sio.local  srv-traefik" | sudo tee -a /etc/hosts
sudo timedatectl set-timezone Europe/Paris
sudo apt install -y curl tree wget ca-certificates
```

2. Donnez-lui l'adresse **192.168.20.66** et les mêmes DNS que SRV-ZBX, dans `/etc/network/interfaces` et `/etc/resolv.conf` (même méthode qu'au point A), puis redémarrez le réseau. Contrôlez le résultat, en particulier la résolution des deux noms :

/etc/network/interfaces

```
auto ens18
iface ens18 inet static
    address 192.168.20.66/24
    gateway 192.168.20.254
```



Figure 3 – Contrôle de la préparation de SRV-TRAEFIK

À SAVOIR

`getent hosts` interroge le DNS du domaine comme le ferait une application. Si les noms ne sont pas résolus, vérifiez `/etc/resolv.conf` et que les enregistrements existent sur SRV-DC1 (`Get-DnsServerResourceRecord -ZoneName sio.local -Name zabbix`).

✔ **POINT DE CONTRÔLE**

- SRV-ZBX répond au nom **srv-zbx** et à l'adresse **192.168.20.60**, SRV-TRAEFIK au nom **srv-traefik** et à **192.168.20.66**
- `zabbix.sio.local` et `traefik.sio.local` sont résolus en **192.168.20.66** depuis SRV-TRAEFIK
- Le fuseau horaire est **Europe/Paris** et l'horloge est **synchronisée** sur les deux serveurs

02

ÉTAPE 2 SUR 6

Installer Zabbix en natif sur Debian 13

Cette étape installe Zabbix 8.0 LTS sur **SRV-ZBX**, comme dans le tutoriel « Installer Zabbix et superviser ses hôtes » ; seul l'essentiel est repris ici. À la fin, l'interface web répond en HTTP sur SRV-ZBX mais n'est pas encore configurée : l'assistant d'installation sera lancé à l'étape 6, **à travers Traefik**.

A – Installer MariaDB et créer la base

1. Installez MariaDB (version 11.8, prise en charge par Zabbix 8.0) : `sudo apt install -y mariadb-server`
2. Sécurisez l'installation avec `sudo mariadb-secure-installation` : laissez le mot de passe root vide (Entrée), répondez **n** aux deux premières questions (authentification par socket déjà active, pas de nouveau mot de passe root) et **Y** à toutes les suivantes (comptes anonymes, connexion distante de root, base de test, rechargement)
3. Ouvrez la console avec `sudo mariadb`, créez la base **zabbix** (jeu de caractères imposé par Zabbix), l'utilisateur **zabbix** et ses droits :

MARIADB

```
CREATE DATABASE zabbix CHARACTER SET utf8mb4 COLLATE utf8mb4_bin;
CREATE USER 'zabbix'@'localhost' IDENTIFIED BY 'Zbx_SIO_2026';
GRANT ALL PRIVILEGES ON zabbix.* TO 'zabbix'@'localhost';
SET GLOBAL log_bin_trust_function_creators = 1;
QUIT;
```

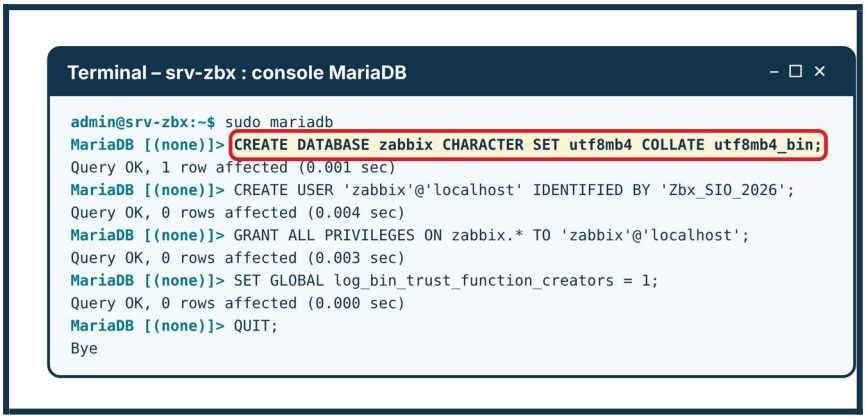


Figure 4 – Console MariaDB : création de la base et de l'utilisateur

À SAVOIR

Le classement utf8mb4_bin est imposé par Zabbix. L'option `log_bin_trust_function_creators` autorise la création des fonctions stockées pendant l'import ; elle est remise à 0 juste après.

REMARQUE IMPORTANTE

`Zbx_SIO_2026` n'est qu'un exemple : **choisissez un mot de passe solide** pour l'utilisateur **zabbix**. Vous le saisirez à l'import du schéma, dans `zabbix_server.conf` et dans l'assistant web.

B – Ajouter le dépôt officiel et installer les paquets

1 Ajoutez le dépôt de Zabbix 8.0 pour Debian 13, puis installez le serveur, l'interface web, la configuration Apache, les scripts SQL et l'agent 2 :

TERMINAL (SRV-ZBX)

```
sudo apt install -y wget
wget https://repo.zabbix.com/zabbix/8.0/release/debian/pool/main/z/\
zabbix-release/zabbix-release_latest_8.0+debian13_all.deb
sudo dpkg -i zabbix-release_latest_8.0+debian13_all.deb
sudo apt update
sudo apt install -y zabbix-server-mysql zabbix-frontend-php zabbix-apache-conf \
    zabbix-sql-scripts zabbix-agent2
```

C – Importer le schéma de la base

1 Importez le schéma (saisissez le mot de passe de l'utilisateur **zabbix** ; l'import dure une à deux minutes et n'affiche rien s'il réussit), remettez l'option temporaire à 0 puis vérifiez que les tables existent :

TERMINAL (SRV-ZBX)

```
zcat /usr/share/zabbix/sql-scripts/mysql/server.sql.gz | \
    mariadb --default-character-set=utf8mb4 -uzabbix -p zabbix
sudo mariadb -e "SET GLOBAL log_bin_trust_function_creators = 0;"
sudo mariadb zabbix -e "SHOW TABLES LIKE 'host%';"
```

La dernière commande doit lister plusieurs tables, dont **hosts**. Si elle ne renvoie rien, l'import a échoué (voir le dépannage en fin de tutoriel).

D – Indiquer le mot de passe au serveur Zabbix

1 Éditez `sudo nano /etc/zabbix/zabbix_server.conf`, retrouvez la ligne `# DBPassword=`, supprimez le `#` et saisissez le mot de passe de l'utilisateur **zabbix** :

/etc/zabbix/zabbix_server.conf

```
DBHost=localhost
DBName=zabbix
DBUser=zabbix
DBPassword=Zbx_SIO_2026
```

E – Démarrer les services

1 Activez la configuration Apache de Zabbix, démarrez les services et activez-les au démarrage du système :

TERMINAL (SRV-ZBX)

```
sudo a2enconf zabbix
sudo systemctl restart zabbix-server zabbix-agent2 apache2
sudo systemctl enable zabbix-server zabbix-agent2 apache2
```

2 Vérifiez que les quatre services sont actifs et que les ports sont ouverts :

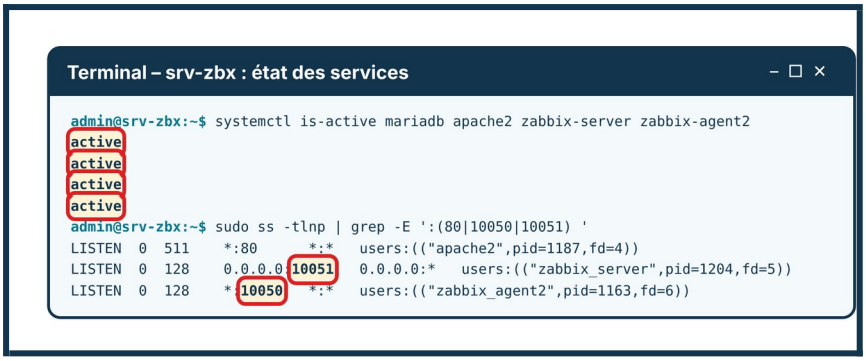


Figure 5 – Services actifs et ports en écoute sur SRV-ZBX

À SAVOIR

Apache écoute sur le port 80 : c'est lui que Traefik va joindre. Si « zabbix-server » est « failed », consultez `sudo tail -n 30 /var/log/zabbix/zabbix_server.log`.

✔ POINT DE CONTRÔLE

- Le dépôt est celui de la **8.0** et les cinq paquets sont installés
- La table **hosts** existe dans la base **zabbix** et `DBPassword` est renseigné
- **mariadb**, **apache2**, **zabbix-server** et **zabbix-agent2** sont « active »
- `curl -sI http://192.168.20.60/zabbix/` répond depuis SRV-TRAEFIK (le port 80 de SRV-ZBX est joignable)

03 ÉTAPE 3 SUR 6 Installer Traefik

Traefik s'installe sur **SRV-TRAEFIK** : tout ce qui suit, jusqu'à l'étape 5 incluse, est exécuté sur cette machine (sauf indication contraire). L'installation se fait **en natif**, sans conteneur : un exécutable, un compte système et un service.

A – Télécharger et installer l'exécutable

Traefik est distribué sous la forme d'**un seul exécutable**, sans dépendance. On le télécharge depuis la page des versions du projet (GitHub), on vérifie son **empreinte** (somme de contrôle) puis on l'installe dans `/usr/local/bin`. Sur **SRV-TRAEFIK** :

TERMINAL

```
cd /tmp
V=3.7.16
URL=https://github.com/traefik/traefik/releases/download/v$V
wget -q $URL/traefik_v${V}_linux_amd64.tar.gz $URL/traefik_v${V}_checksums.txt
sha256sum -c --ignore-missing traefik_v${V}_checksums.txt
tar -xzf traefik_v${V}_linux_amd64.tar.gz traefik
sudo install -m 0755 traefik /usr/local/bin/traefik
traefik version | head -1
```

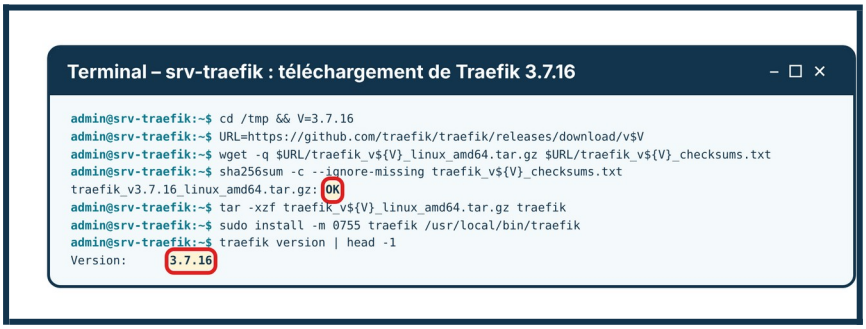


Figure 6 – Téléchargement, vérification et installation de Traefik 3.7.16

À SAVOIR

La ligne « OK » prouve que le fichier téléchargé n'a pas été altéré. Si la commande répond « No such file » ou « FAILED », vérifiez le numéro de version dans la page des versions du projet : les noms de fichiers en dépendent.

B – Créer l'utilisateur et les dossiers

1 Traefik ne doit pas tourner en `root` : on crée un compte système dédié, sans mot de passe ni interpréteur de commandes, et les trois dossiers de configuration (statique, dynamique, certificats) :

TERMINAL

```
sudo useradd --system --no-create-home --shell /usr/sbin/nologin traefik
sudo install -d -o traefik -g traefik /etc/traefik /etc/traefik/dynamic
sudo install -d -o traefik -g traefik /etc/traefik/certs /var/log/traefik
```

C – Écrire la configuration statique

La configuration de Traefik se compose de **deux parties**. La **statique** décrit les bases (ports d'écoute, d'où viennent les routes) et n'est lue qu'au démarrage. La **dynamique** décrit les routes, les certificats et les middlewares ; elle est lue dans des **fichiers** et rechargée **à chaud**. Créez `sudo nano /etc/traefik/traefik.yml` :

/etc/traefik/traefik.yml

```
entryPoints:
  web:
    address: ":80"
    http:
      redirections:
        entryPoint:
          to: websecure
          scheme: https
          permanent: true
  websecure:
    address: ":443"

providers:
  file:
    directory: /etc/traefik/dynamic
    watch: true

api:
  dashboard: true

log:
  level: INFO

accessLog:
  filePath: /var/log/traefik/access.log
```

Bloc	Rôle
`entryPoints`	Les ports d'écoute : **web** (80) redirige tout vers **websecure** (443, HTTPS)
`providers.file`	La source des routes : les fichiers du dossier `dynamic`, surveillés (`watch`) pour un rechargement à chaud
`api.dashboard`	Active le tableau de bord ; il sera publié par un routeur protégé (étape 5), pas sur un port ouvert
`accessLog`	Journal des requêtes, utile pour diagnostiquer une route

REMARQUE IMPORTANTE

Traefik offre aussi d'autres sources de configuration (notamment les **labels Docker**). Ici, **toute la configuration passe par des fichiers** : un fichier par application, lisible, comparable et sauvegardable comme n'importe quel autre fichier du serveur. Aucun accès au moteur Docker n'est donc nécessaire.

D – Créer le service systemd et le démarrer

1 Créez `sudo nano /etc/systemd/system/traefik.service` :

/etc/systemd/system/traefik.service

```
[Unit]
Description=Traefik (proxy inverse)
After=network-online.target
Wants=network-online.target

[Service]
User=traefik
Group=traefik
ExecStart=/usr/local/bin/traefik --configFile=/etc/traefik/traefik.yml
AmbientCapabilities=CAP_NET_BIND_SERVICE
NoNewPrivileges=true
Restart=on-failure
LimitNOFILE=65536

[Install]
WantedBy=multi-user.target
```

La ligne `AmbientCapabilities` autorise le compte `traefik` à ouvrir les ports **80 et 443** (ports réservés) sans être `root`. Activez et démarrez le service, puis contrôlez-le :

TERMINAL

```
sudo systemctl daemon-reload
sudo systemctl enable --now traefik
systemctl is-active traefik
sudo ss -tlnp | grep traefik
```



Figure 7 – Traefik démarré : arborescence et ports en écoute

À SAVOIR

À ce stade, le dossier `dynamic` est vide : Traefik répond « 404 page not found » à toute requête, ce qui est normal. En cas d'échec, `sudo journalctl -u traefik -n 20` indique la ligne du fichier YAML en cause (indentation, guillemets).

✔ POINT DE CONTRÔLE

- `sha256sum` affiche **OK** et `traefik version` indique **3.7.16**
- Le service **traefik** est « active » et écoute sur les ports **80** et **443**
- Les dossiers `/etc/traefik/dynamic` et `/etc/traefik/certs` existent et appartiennent à `traefik`

04 ÉTAPE 4 SUR 6 Obtenir un certificat auprès de SIO-CA

Les navigateurs n'acceptent un site HTTPS que si son certificat est délivré par une autorité **qu'ils connaissent**. Dans le domaine **sio.local**, c'est l'autorité interne **SIO-CA** (construite dans le tutoriel « Activer LDAPS sur AD DS ») : les postes Windows du domaine lui font déjà confiance. Nous allons lui demander un certificat pour **zabbix.sio.local et traefik.sio.local**. La clé privée est créée sur **SRV-TRAEFIK** et ne le quitte jamais : seule la demande (CSR) voyage.

A – Créer la clé privée et la demande de certificat

1 Créez le fichier de description `sudo nano /tmp/zabbix.cnf` : le certificat doit porter **tous les noms** utilisés (champ « Subject Alternative Name »), les navigateurs ignorant le nom principal seul :

/tmp/zabbix.cnf

```
[req]
distinguished_name = dn
req_extensions = ext
prompt = no

[dn]
CN = zabbix.sio.local

[ext]
subjectAltName = DNS:zabbix.sio.local, DNS:traefik.sio.local
```

2 Générez la clé privée (RSA 2048) et la demande de certificat, puis contrôlez le contenu de la demande :

TERMINAL

```
sudo openssl req -new -newkey rsa:2048 -nodes -config /tmp/zabbix.cnf \
    -keyout /etc/traefik/certs/zabbix.key -out /tmp/zabbix.csr
openssl req -in /tmp/zabbix.csr -noout -subject -verify
openssl req -in /tmp/zabbix.csr -noout -text | grep -A1 Alternative
```

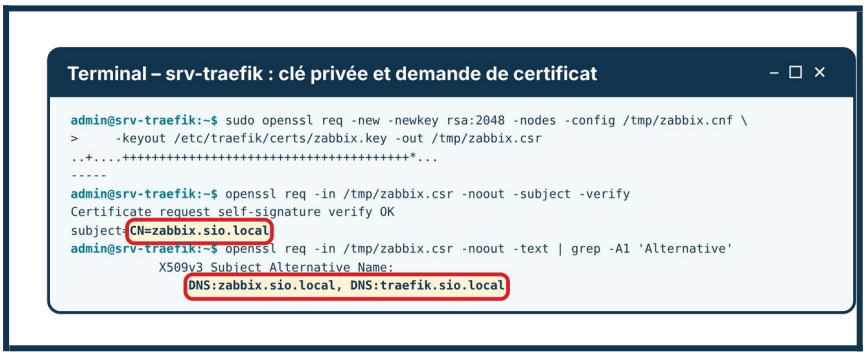


Figure 8 – Clé privée et demande de certificat avec ses deux noms

À SAVOIR

Le fichier `.key` est la clé privée : elle reste sur SRV-TRAEFIK, avec des droits restreints (voir plus bas). Le fichier `.csr` ne contient que la clé **publique** et les noms : il peut circuler librement.

B – Faire signer la demande par SIO-CA

Sur **SRV-DC1** (qui héberge l'autorité de certification), dans PowerShell ouvert **en tant qu'administrateur**, récupérez la demande, soumettez-la avec le modèle « **Serveur Web** » (nom interne `WebServer`) puis renvoyez le certificat obtenu vers SRV-TRAEFIK :

POWERSHELL

```
scp admin@192.168.20.66:/tmp/zabbix.csr C:\Temp\
certreq -submit -config "SRV-DC1.sio.local\SIO-CA" `
    -attrib "CertificateTemplate:WebServer" `
    C:\Temp\zabbix.csr C:\Temp\zabbix.cer
scp C:\Temp\zabbix.cer admin@192.168.20.66:/tmp/
```



Figure 9 – Signature de la demande par l'autorité SIO-CA

À SAVOIR

Si `certreq` répond « Refusé » ou « En attente », le modèle « Serveur Web » n'est pas activé sur l'autorité (console certsrv > Modèles de certificats > Nouveau > Modèle de certificat à délivrer) ou le compte n'a pas le droit « Inscrire » dessus.

C – Installer le certificat sur SRV-TRAEFIK

1. Convertissez le certificat reçu au format PEM, donnez la clé au compte `traefik` seul, puis vérifiez que le certificat correspond bien à la clé :

TERMINAL

```
sudo openssl x509 -in /tmp/zabbix.cer -out /etc/traefik/certs/zabbix.crt
sudo chown -R traefik:traefik /etc/traefik/certs
sudo chmod 600 /etc/traefik/certs/zabbix.key
sudo openssl x509 -in /etc/traefik/certs/zabbix.crt -noout -subject -issuer -enddate
sudo openssl x509 -in /etc/traefik/certs/zabbix.crt -noout -ext subjectAltName
cmp <(sudo openssl x509 -in /etc/traefik/certs/zabbix.crt -noout -pubkey) \
    <(sudo openssl pkey -in /etc/traefik/certs/zabbix.key -pubout) \
    && echo "Clé et certificat correspondent"
```

À SAVOIR

Si OpenSSL répond « Could not read certificate » (ou « unable to load »), le fichier reçu est au format binaire : ajoutez l'option `-inform der` à la première commande.



Figure 10 – Certificat installé : émetteur, noms et correspondance avec la clé

À SAVOIR

Le certificat vaut deux ans environ (durée du modèle). Notez sa date de fin : **un certificat expiré rend le site inaccessible** (alerte du navigateur). Pour un renouvellement, répétez ces trois points puis remplacez les deux fichiers : Traefik les relit à chaud.

✔ POINT DE CONTRÔLE

- La demande `zabbix.csr` contient les noms **zabbix.sio.local** et **traefik.sio.local**
- SIO-CA a émis le certificat (« **Émis** ») et il est copié dans `/etc/traefik/certs`
- L'émetteur affiché est **SIO-CA** et la clé privée est lisible uniquement par `traefik`

ÉTAPE 5 SUR 6

05 Publier Zabbix avec les fichiers dynamiques de Traefik

Nous créons maintenant les fichiers du dossier `/etc/traefik/dynamic`. Traefik les **surveille** : il charge chaque fichier dès qu'il est enregistré, sans redémarrage. Un fichier invalide est ignoré (et signalé dans le journal) : le reste de la configuration continue de fonctionner. Les trois premiers fichiers sont **communs** à toutes les applications que Traefik publiera un jour ; le dernier est propre à Zabbix.

A – Déclarer le certificat et les options TLS

1. Créez `sudo nano /etc/traefik/dynamic/tls.yml` : il indique à Traefik le certificat à présenter (il choisit le bon d'après le nom demandé par le navigateur) et impose TLS 1.2 au minimum :

/etc/traefik/dynamic/tls.yml

```
tls:
  certificates:
    - certFile: /etc/traefik/certs/zabbix.crt
      keyFile: /etc/traefik/certs/zabbix.key
  options:
    default:
      minVersion: VersionTLS12
```

B – Créer les middlewares réutilisables

Un **middleware** modifie la requête ou la réponse au passage dans Traefik. On en déclare trois dans un fichier, que tous les routeurs pourront réutiliser. Créez `sudo nano /etc/traefik/dynamic/middlewares.yml` :

/etc/traefik/dynamic/middlewares.yml

```
http:
  middlewares:
    secure-headers:
      headers:
        stsSeconds: 31536000
        contentTypeNosniff: true
        referrerPolicy: same-origin
        customFrameOptionsValue: SAMEORIGIN
    compression:
      compress: {}
    admin-seulement:
      ipAllowList:
        sourceRange:
          - 192.168.20.0/24
          - 192.168.10.0/24
```

Middleware	Effet
`secure-headers`	Ajoute des en-têtes de sécurité : HSTS (le navigateur n'utilisera plus que HTTPS pendant un an), interdiction de deviner le type des fichiers, cadres limités au même site
`compression`	Compresse les réponses (pages plus légères sur le réseau)
`admin-seulement`	Autorise uniquement les adresses listées (`ipAllowList`) : à utiliser pour les interfaces d'administration

REMARQUE IMPORTANTE

Adaptez les réseaux de `sourceRange` : ils doivent contenir l'adresse du **poste d'administration** (ici 192.168.10.0/24, à remplacer par votre réseau), sinon vous ne pourrez plus ouvrir le tableau de bord. **Le HSTS est mémorisé un an par le navigateur** : ne l'activez pas sur un nom de domaine que vous pourriez devoir exposer un jour en HTTP.

C – Publier le tableau de bord de Traefik

1. Créez `sudo nano /etc/traefik/dynamic/dashboard.yml` : un routeur, réservé aux administrateurs, vers le service interne `api@internal` :

/etc/traefik/dynamic/dashboard.yml

```
http:
  routers:
    dashboard:
      rule: "Host(`traefik.sio.local`)"
      entryPoints:
        - websecure
      middlewares:
        - admin-seulement
        - secure-headers
      service: api@internal
      tls: {}
```

Un routeur associe trois éléments : une **règle** (`rule` : ici, le nom de site demandé), des **middlewares** (appliqués dans l'ordre) et un **service** (la destination). `tls: {}` active HTTPS sur ce routeur avec le certificat déclaré à l'étape A.

D – Publier Zabbix

Un fichier par application : celui-ci décrit le **routeur** (quel nom, quels middlewares), le **service** (où se trouve Zabbix) et un middleware propre à Zabbix. Comme l'interface de Zabbix est servie sous `/zabbix` et non à la racine du site, ce middleware renvoie un visiteur qui saisit `https://zabbix.sio.local` vers `https://zabbix.sio.local/zabbix/`. Créez `sudo nano /etc/traefik/dynamic/zabbix.yml` :

/etc/traefik/dynamic/zabbix.yml

```
http:
  routers:
    zabbix:
      rule: "Host(`zabbix.sio.local`)"
      entryPoints:
        - websecure
      middlewares:
        - secure-headers
        - compression
        - zabbix-racine
      service: zabbix
      tls: {}

  middlewares:
    zabbix-racine:
      redirectRegex:
        regex: '^https://zabbix\.sio\.local/?$'
        replacement: "https://zabbix.sio.local/zabbix/"

  services:
    zabbix:
      loadBalancer:
        servers:
          - url: "http://192.168.20.60"
```

Élément	Signification
`rule`	Condition d'aiguillage : la requête est destinée à ce routeur si le nom de site demandé est `zabbix.sio.local`
`entryPoints: websecure`	Le routeur n'écoute que sur le port 443 ; le port 80 ne fait que rediriger (configuration statique)
`middlewares`	Appliqués dans l'ordre : en-têtes de sécurité, compression, puis redirection de la racine. Les deux premiers viennent de `middlewares.yml`, le troisième est défini plus bas dans le même fichier
`service: zabbix`	Renvoie au service du même nom, déclaré en bas du fichier
`redirectRegex`	Compare l'adresse complète demandée à l'expression régulière ; si elle correspond (racine seule), renvoie vers `/zabbix/`. Les autres adresses ne sont pas touchées
`loadBalancer.servers`	L'adresse d'Apache sur SRV-ZBX. Une seule adresse ici ; en ajouter une deuxième répartirait la charge entre deux serveurs
`tls: {}`	Active HTTPS sur ce routeur avec le certificat déclaré dans `tls.yml`

REMARQUE IMPORTANTE

Dans un fichier YAML, une **erreur d'indentation** change le sens du fichier ou l'invalide. L'expression régulière est écrite entre **apostrophes** : dans ce cas, les barres obliques inverses (`\.`) sont conservées telles quelles. Entre guillemets, il faudrait les doubler.

✔ **POINT DE CONTRÔLE**

- Le dossier `/etc/traefik/dynamic` contient **tls.yml**, **middlewares.yml**, **dashboard.yml** et **zabbix.yml**
- Les réseaux de `sourceRange` incluent l'adresse du **poste d'administration**
- Le certificat et la clé sont bien référencés dans `tls.yml` (chemins `/etc/traefik/certs/zabbix.crt` et `.key`)

06 ÉTAPE 6 SUR 6 — Terminer l'installation et vérifier la chaîne

A – Contrôler les fichiers et le démarrage

1 Sur **SRV-TRAEFIK**, affichez l'arborescence de la configuration, les ports en écoute et le nombre d'erreurs du journal des cinq dernières minutes :

TERMINAL (SRV-TRAEFIK)

```
sudo tree /etc/traefik
sudo ss -tlnp | grep traefik | awk '{print $1, $4}'
sudo journalctl -u traefik --since "5 min ago" --no-pager | grep -ci error
```

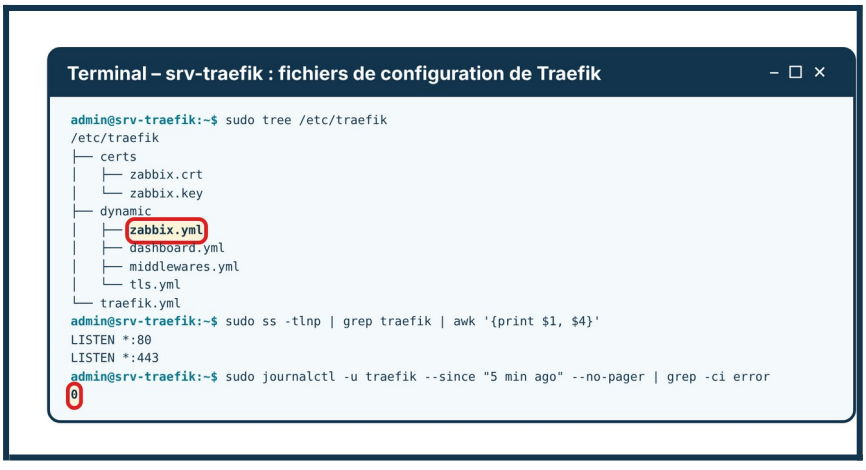


Figure 11 – Fichiers de configuration de Traefik

À SAVOIR

Un compteur différent de `0` indique un fichier mal formé : `sudo journalctl -u traefik -n 20` donne le nom du fichier et la ligne. Traefik n'a pas besoin d'être redémarré : les fichiers dynamiques sont rechargés à chaud.

B – Tester la chaîne en ligne de commande

Pour tester depuis SRV-TRAEFIK lui-même, ce serveur doit faire confiance à **SIO-CA** (les postes Windows du domaine lui font déjà confiance, pas Debian). Sur **SRV-DC1**, exportez le certificat de l'autorité et copiez-le :

POWERSHELL (SRV-DC1)

```
certutil -ca.cert C:\Temp\sio-ca.cer
scp C:\Temp\sio-ca.cer admin@192.168.20.66:/tmp/
```

Sur **SRV-TRAEFIK**, convertissez-le au format PEM, ajoutez-le aux autorités de confiance du système et vérifiez le certificat de Traefik par rapport à ce magasin :

TERMINAL (SRV-TRAEFIK)

```
sudo openssl x509 -inform der -in /tmp/sio-ca.cer \
    -out /usr/local/share/ca-certificates/sio-ca.crt
sudo update-ca-certificates
openssl verify /etc/traefik/certs/zabbix.crt
```



À SAVOIR

« OK » prouve que le certificat de Traefik est signé par SIO-CA. Si OpenSSL répond « Could not read certificate », le fichier exporté est déjà au format PEM : copiez-le simplement dans `/usr/local/share/ca-certificates/sio-ca.crt`.

Figure 12 – Confiance dans SIO-CA et vérification du certificat

1 Testez la redirection HTTP vers HTTPS, la redirection de la racine vers `/zabbix/` et le certificat présenté :

TERMINAL (SRV-TRAEFIK)

```
curl -sI http://zabbix.sio.local | grep -iE '^(HTTP|location)'
curl -sI https://zabbix.sio.local/ | grep -iE '^(HTTP|location)'
curl -sv https://zabbix.sio.local/ -o /dev/null 2>&1 \
    | grep -E 'subject:|issuer:|verify'
```



À SAVOIR

Le premier test montre le port 80 qui renvoie vers HTTPS ; le deuxième, la racine du site qui renvoie vers `/zabbix/` (middleware `zabbix-racine`) ; le troisième, l'émetteur du certificat. « 404 page not found » signale qu'aucun routeur ne correspond au nom demandé.

Figure 13 – Redirections et certificat vus par curl

C – Ouvrir le tableau de bord de Traefik

Depuis le **poste d'administration** (membre du domaine), ouvrez `https://traefik.sio.local`. Le navigateur affiche le cadenas **sans avertissement** : il fait confiance à SIO-CA. Le tableau de bord liste les routeurs lus dans les fichiers :

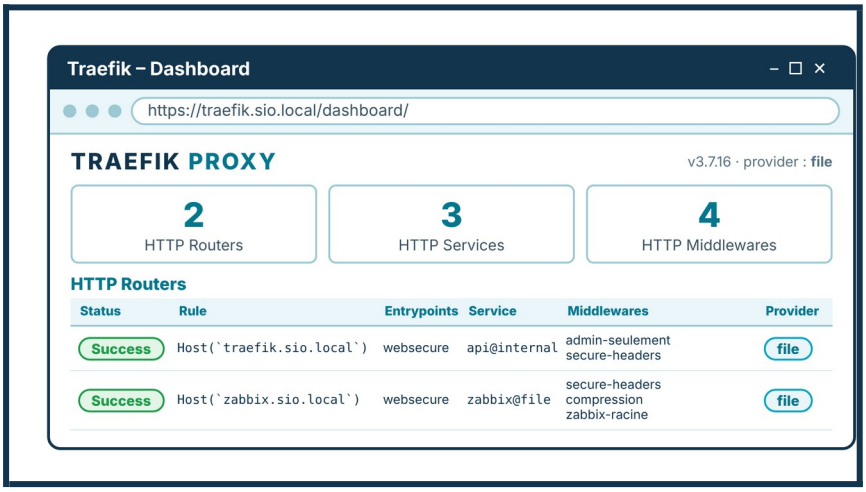


Figure 14 – Tableau de bord de Traefik : routeurs, services et middlewares

À SAVOIR

Un routeur « Success » est actif ; une erreur s'affiche en rouge avec sa cause. Le provider « file » rappelle que la configuration vient des fichiers. Un message « 403 Forbidden » signale que l'adresse du poste n'est pas dans `sourceRange` (middleware `admin-seulement`).

À SAVOIR

Un poste **hors domaine** n'a pas SIO-CA dans ses autorités de confiance : le navigateur affiche un avertissement. Importez le fichier `sio-ca.cer` dans les « Autorités de certification racines de confiance » du poste, ou déployez-le par GPO pour tous les postes du domaine.

D – Terminer l'installation de Zabbix

L'interface web se termine de configurer dans le navigateur, avec un assistant en six pages, **à travers Traefik**. Ouvrez `https://zabbix.sio.local` : la racine renvoie vers `/zabbix/`, qui renvoie elle-même vers l'assistant :

1 Page « **Welcome** » : laissez la langue « **English (en_US)** » puis cliquez « **Next step** » :

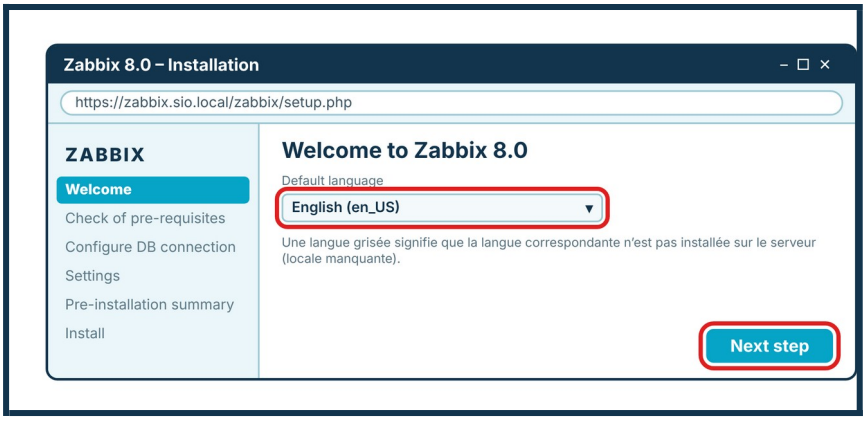


Figure 15 – Assistant d'installation : page de bienvenue

À SAVOIR

L'adresse affichée est bien `https://zabbix.sio.local/zabbix/setup.php` : la page a traversé Traefik. Une langue grisée indique que sa « locale » n'est pas générée sur SRV-ZBX (étape 1).

2 Page « **Check of pre-requisites** » : toutes les lignes doivent afficher « **OK** ». Cliquez « **Next step** » :

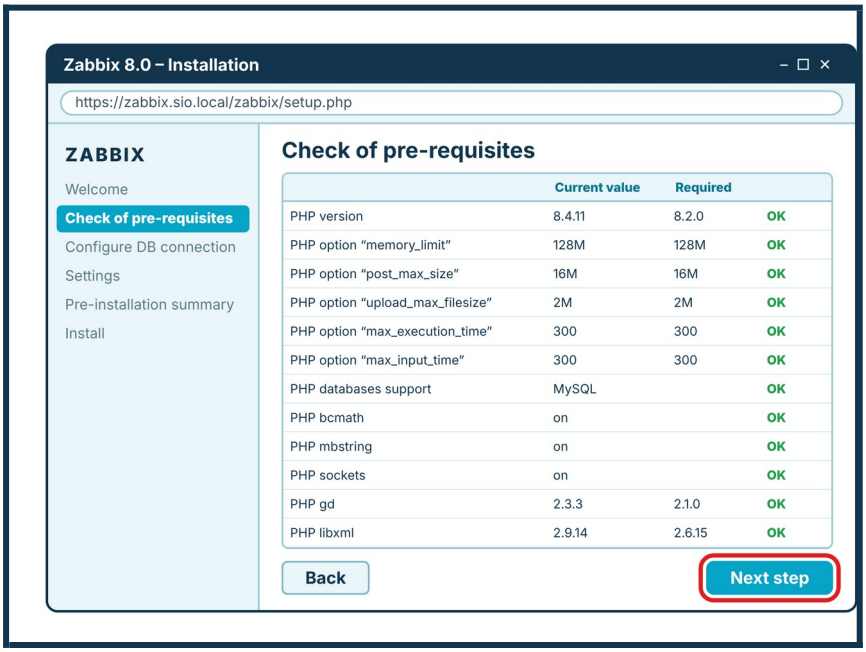


Figure 16 – Assistant d'installation : vérification des prérequis

À SAVOIR

Une ligne rouge signale un module PHP manquant ou une limite trop basse : corrigez-la, relancez Apache (`sudo systemctl restart apache2`) puis cliquez « Back » et « Next step ».

3 Page « **Configure DB connection** » : type « **MySQL** », hôte `localhost`, port `0`, base `zabbix`, utilisateur `zabbix` et son mot de passe :

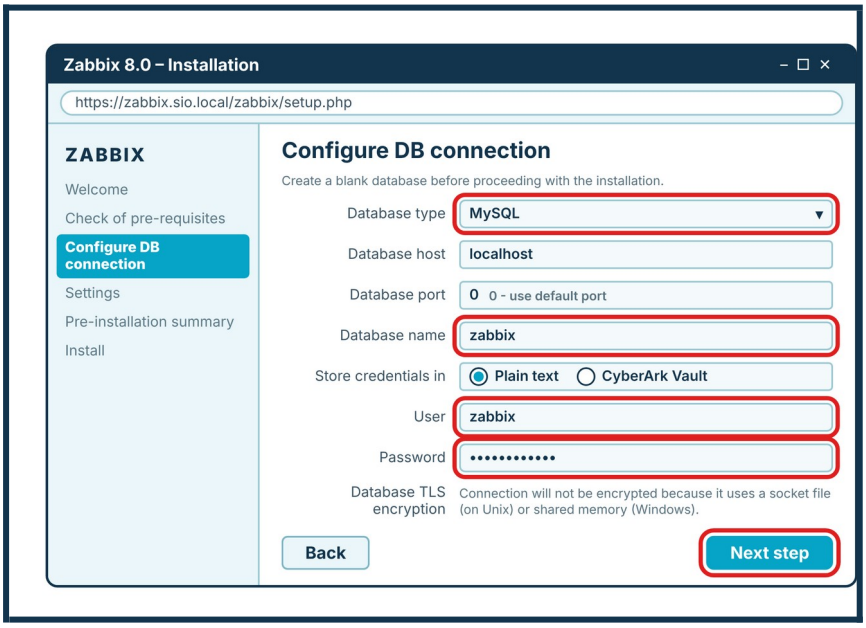


Figure 17 – Assistant d'installation : connexion à la base de données

À SAVOIR

La base étant sur le même serveur que Zabbix (et non sur SRV-TRAEFIK), la connexion passe par un fichier socket. « Cannot connect to the database » vient presque toujours d'un mot de passe erroné ou d'un schéma non importé.

4 Page « **Settings** » : nom du serveur `Zabbix SIO`, fuseau « **(UTC+01:00) Europe/Paris** », thème « **Blue** ». Cliquez « **Next step** » :

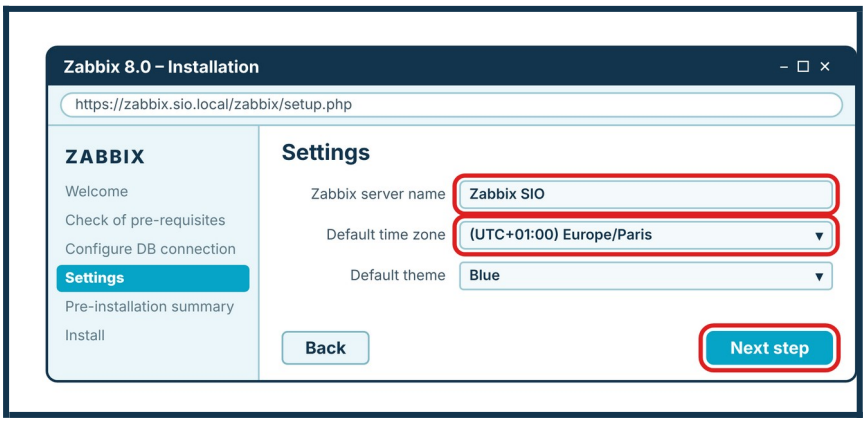


Figure 18 – Assistant d'installation : paramètres généraux

À SAVOIR

Le fuseau choisi ici est celui de l'affichage par défaut : chaque utilisateur pourra définir le sien dans son profil.

5 Page « **Pre-installation summary** » : contrôlez les paramètres puis cliquez « **Next step** » ; la page « **Install** » confirme l'installation, cliquez « **Finish** » :

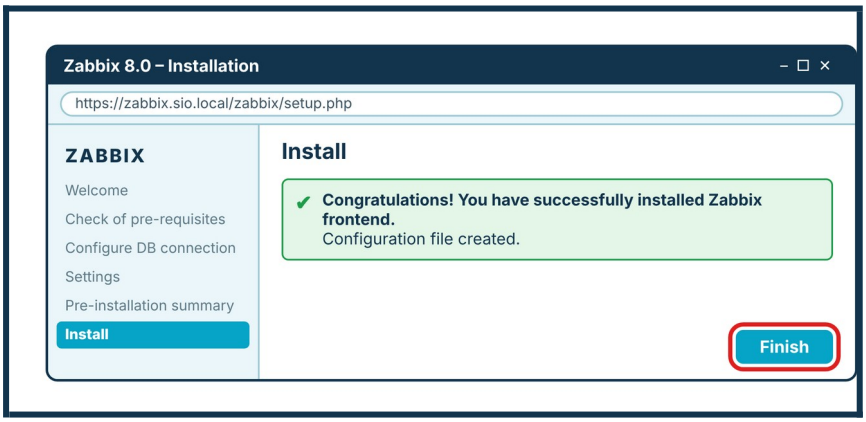


Figure 19 – Assistant d'installation : installation terminée

À SAVOIR

L'assistant écrit le fichier de configuration de l'interface web : il ne se relance plus ensuite.

6 Sur la page de connexion, saisissez l'utilisateur `Admin` et le mot de passe `zabbix` (valeurs par défaut), puis cliquez « **Sign in** » :



Figure 20 – Page de connexion de Zabbix

À SAVOIR

Attention à la casse : le compte s'appelle « Admin » avec un A majuscule. Le cadenas du navigateur doit être fermé : la page est bien servie en HTTPS.

Le tableau de bord « **Global view** » s'affiche. Dans « **System information** », la ligne « **Zabbix server is running** » doit indiquer « **Yes** » :

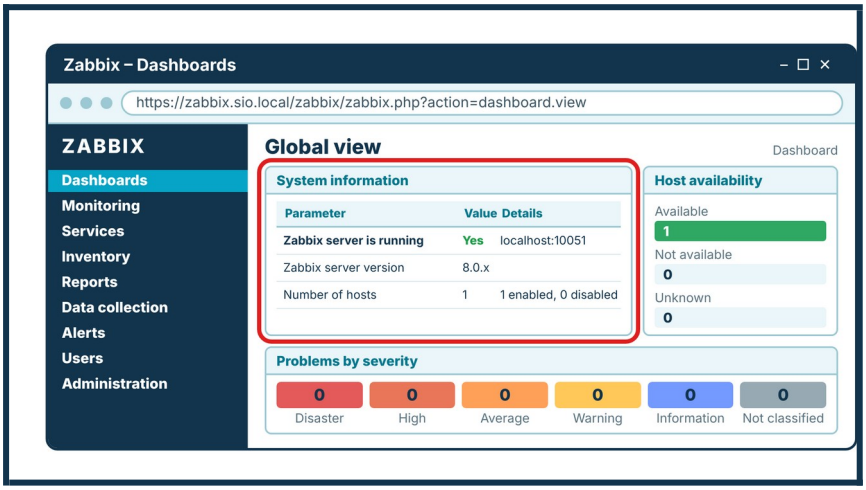


Figure 21 – Tableau de bord Global view

À SAVOIR

Si la ligne indique « No », le serveur Zabbix est arrêté ou ne se connecte pas à la base : consultez `/var/log/zabbix/zabbix_server.log` sur SRV-ZBX.

REMARQUE IMPORTANTE

Changez immédiatement le mot de passe du compte Admin : « zabbix » est le mot de passe par défaut, connu de tous. Cliquez le menu « **Users** » puis « **Users** », ouvrez le compte « **Admin** », cliquez « **Change password** », saisissez le mot de passe actuel puis deux fois le nouveau, et cliquez « **Update** ».

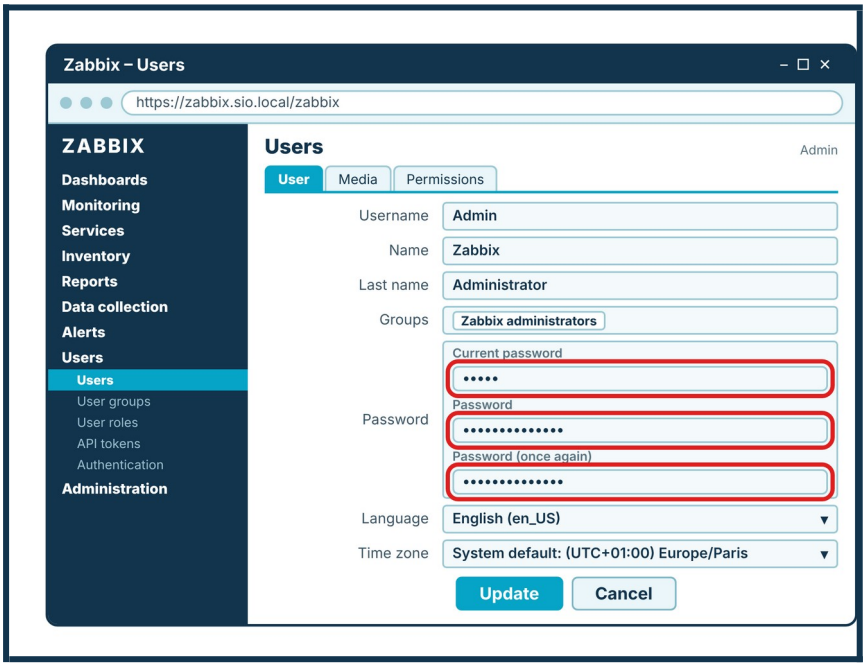


Figure 22 – Users > Users > Admin : changement du mot de passe

À SAVOIR

Dans la même fiche, la liste « Language » permet de passer l'interface en français (« French (fr_FR) ») grâce à la langue générée à l'étape 1. Les noms de menus de ce tutoriel sont ceux de l'interface en anglais.

E – Voir l'adresse réelle des utilisateurs dans Apache

Pour Apache, toutes les connexions viennent désormais de **Traefik** : le journal d'accès (et le journal d'audit de Zabbix, qui reprend l'adresse vue par le serveur web) affichent donc 192.168.20.66 au lieu de l'adresse de l'utilisateur. Traefik transmet l'adresse d'origine dans l'en-tête `X-Forwarded-For` ; le module **remoteip** d'Apache peut la lire, à condition de **ne faire confiance qu'à Traefik**. Le même fichier indique à PHP que la connexion d'origine était en HTTPS (en-tête `X-Forwarded-Proto`). Sur **SRV-ZBX**, créez `sudo nano /etc/apache2/conf-available/proxy-traefik.conf` :

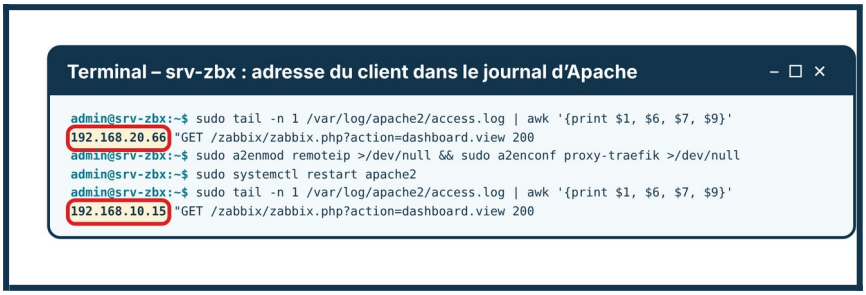
/etc/apache2/conf-available/proxy-traefik.conf

```
RemoteIPHeader X-Forwarded-For
RemoteIPInternalProxy 192.168.20.66
SetEnvIf X-Forwarded-Proto "^https$" HTTPS=on
```

1. Activez le module et la configuration, redémarrez Apache, puis rechargez une page de Zabbix dans le navigateur et lisez la dernière ligne du journal (colonnes : adresse, requête, code) :

TERMINAL (SRV-ZBX)

```
sudo tail -n 1 /var/log/apache2/access.log | awk '{print $1, $6, $7, $9}'
sudo a2enmod remoteip > /dev/null && sudo a2enconf proxy-traefik > /dev/null
sudo systemctl restart apache2
sudo tail -n 1 /var/log/apache2/access.log | awk '{print $1, $6, $7, $9}'
```



À SAVOIR

Avant : l'adresse de Traefik. Après : l'adresse du poste d'administration. La directive `RemoteIPInternalProxy` limite la confiance à Traefik ; sans elle, n'importe quel client pourrait falsifier l'en-tête `X-Forwarded-For`.

Figure 23 – Adresse du client dans le journal d'Apache

F – Fermer l'accès direct à Zabbix

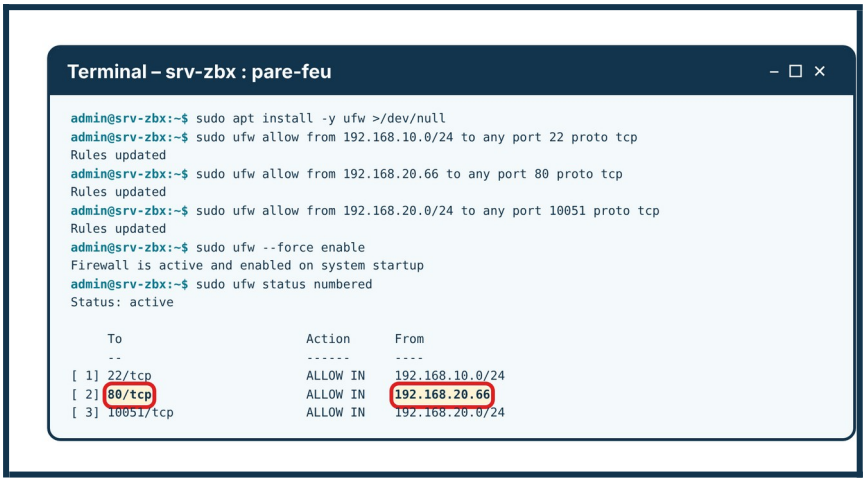
Tant qu'Apache répond à tout le monde, un utilisateur peut contourner Traefik (et donc le HTTPS) en saisissant `http://192.168.20.60/zabbix`. On installe un pare-feu sur SRV-ZBX qui n'autorise que l'indispensable : le **SSH** depuis le réseau d'administration, le **port 80 depuis Traefik seulement** et le port **10051** pour les agents Zabbix actifs.

TERMINAL (SRV-ZBX)

```
sudo apt install -y ufw
sudo ufw allow from 192.168.10.0/24 to any port 22 proto tcp
sudo ufw allow from 192.168.20.66 to any port 80 proto tcp
sudo ufw allow from 192.168.20.0/24 to any port 10051 proto tcp
sudo ufw --force enable
sudo ufw status numbered
```

REMARQUE IMPORTANTE

Autorisez le SSH avant d'activer le pare-feu : une règle d'accès SSH erronée vous enferme dehors (adaptez le réseau d'administration 192.168.10.0/24 au vôtre). En cas de blocage, reprenez la main depuis la **console Proxmox** et saisissez `sudo ufw disable`. Gardez cette console ouverte pendant l'opération. Le trafic local (agent de SRV-ZBX vers le serveur Zabbix) n'est pas concerné.



À SAVOIR

Le pare-feu refuse tout ce qui n'est pas listé (politique par défaut d'ufw : entrant refusé, sortant autorisé). Les agents Zabbix passifs (Zabbix vers l'hôte) ne sont pas concernés : la connexion part du serveur.

Figure 24 – Pare-feu de SRV-ZBX : seul Traefik joint le port 80

Contrôlez depuis le **poste d'administration** : `http://192.168.20.60/zabbix` ne répond plus (le navigateur attend puis abandonne), alors que `https://zabbix.sio.local` fonctionne toujours.

À SAVOIR

Zabbix reste exposé aux **agents** sur le port 10051 : ce trafic ne passe pas par Traefik, qui ne publie ici que des sites web. Pour chiffrer aussi les échanges entre agents et serveur, Zabbix propose un chiffrement par certificat ou par clé pré-partagée (hors du cadre de ce tutoriel).

✔ **POINT DE CONTRÔLE**

- `openssl verify` répond **OK** et `curl` affiche **308** puis **302** vers `/zabbix/`
- Le tableau de bord de Traefik affiche les routeurs **dashboard** et **zabbix** en « Success »
- L'assistant s'est déroulé en **HTTPS** à l'adresse `https://zabbix.sio.local` et le mot de passe **Admin** a été changé
- Le journal d'Apache affiche l'adresse du **poste d'administration**, pas celle de Traefik
- `http://192.168.20.60/zabbix` est **inaccessible** depuis le poste d'administration

Notre Zabbix est **publié en HTTPS** : les utilisateurs ne connaissent que `https://zabbix.sio.local`, Traefik présente le certificat de l'autorité interne et filtre l'accès, et Apache n'est plus joignable que par Traefik. Pour y ajouter des hôtes à superviser (Linux, Windows, auto-enregistrement), reportez-vous au tutoriel « Installer Zabbix et superviser ses hôtes » : seule l'adresse de l'interface change.

Dépannage : les symptômes les plus fréquents

Symptôme	Cause probable	Solution
`sha256sum` : FAILED, ou `wget` : « 404 Not Found »	Numéro de version ou nom de fichier incorrect	Vérifier la version dans la page des versions du projet Traefik et la variable `V`
Le service traefik ne démarre pas	Erreur dans `traefik.yml`, ou droits insuffisants sur `/etc/traefik` ou `/var/log/traefik`	`sudo journalctl -u traefik -n 20` ; contrôler l'indentation ; `sudo chown -R traefik:traefik /etc/traefik /var/log/traefik`
`certreq` : « Refusé » ou « En attente »	Le modèle « Serveur Web » n'est pas publié sur SIO-CA ou le compte n'a pas le droit « Inscrire »	Console `certsrv` > Modèles de certificats > Nouveau > Modèle à délivrer ; vérifier les droits du modèle
Avertissement de certificat dans le navigateur	Poste hors domaine (SIO-CA inconnue), nom absent des noms alternatifs ou certificat expiré	Importer SIO-CA ; vérifier `openssl x509 -noout -ext subjectAltName` ; contrôler `-enddate`
« 404 page not found » (réponse de Traefik)	Aucun routeur ne correspond : nom absent du DNS ou de la règle `Host`, fichier invalide ou non chargé	`getent hosts zabbix.sio.local` ; relire `zabbix.yml` ; consulter le journal et le tableau de bord
« 502 Bad Gateway »	Traefik ne joint pas Apache : service arrêté, mauvaise adresse, pare-feu	Depuis SRV-TRAEFIK : `curl -sI http://192.168.20.60/zabbix/` ; vérifier `systemctl status apache2` et `sudo ufw status`
« 403 Forbidden » sur `traefik.sio.local`	L'adresse du poste n'est pas dans `sourceRange`	Ajouter le réseau du poste dans `middlewares.yml` (rechargement automatique)
Zabbix s'affiche sans mise en forme ou renvoie en boucle	Le service pointe sur une adresse en `https://` alors qu'Apache parle en HTTP, ou une redirection supplémentaire interfère	L'adresse du service est `http://192.168.20.60` ; ne pas rediriger `/zabbix/` vers lui-même
Le journal d'Apache et d'audit de Zabbix affichent l'adresse de Traefik	Module remoteip non activé ou adresse de `RemoteIPInternalProxy` incorrecte	`apache2ctl -M | grep remoteip` ; relire `proxy-traefik.conf` ; `sudo systemctl restart apache2`
Accès SSH perdu après activation du pare-feu	Règle du port 22 absente ou réseau d'administration incorrect	Console Proxmox : `sudo ufw disable`, corriger la règle, réactiver
Assistant Zabbix : « Cannot connect to the database »	Mauvais mot de passe, base vide ou MariaDB arrêtée	`mariadb -uzabbix -p zabbix -e "SHOW TABLES;"` ; `systemctl status mariadb`
« Zabbix server is not running » en haut de l'interface	Service arrêté ou `DBPassword` incorrect	`systemctl status zabbix-server` et `/var/log/zabbix/zabbix_server.log`

Pense-bête : commandes et emplacements

Commande / emplacement	Utilité
`sudo systemctl status traefik`	État de Traefik (`restart` après une modification de `traefik.yml`)
`sudo journalctl -u traefik -f`	Journal de Traefik en direct (erreurs de fichiers, de certificat)
`/var/log/traefik/access.log`	Journal des requêtes : routeur et service utilisés pour chacune
`/etc/traefik/traefik.yml`	Configuration statique : ports, source des fichiers dynamiques (lue au démarrage)
`/etc/traefik/dynamic/*.yml`	Configuration dynamique : rechargée à chaud, un fichier par sujet
`/etc/traefik/certs/`	Certificat et clé privée (clé lisible uniquement par `traefik`)
`openssl x509 -in /etc/traefik/certs/zabbix.crt -noout -enddate`	Date d'expiration du certificat : à surveiller, renouvellement en suivant l'étape 4
`sudo ufw status numbered`	Règles du pare-feu de SRV-ZBX
`systemctl status zabbix-server` / `zabbix-agent2` / `apache2`	État des services de SRV-ZBX (journal : `/var/log/zabbix/`)
`https://traefik.sio.local`	Tableau de bord de Traefik : routeurs, services, middlewares, erreurs