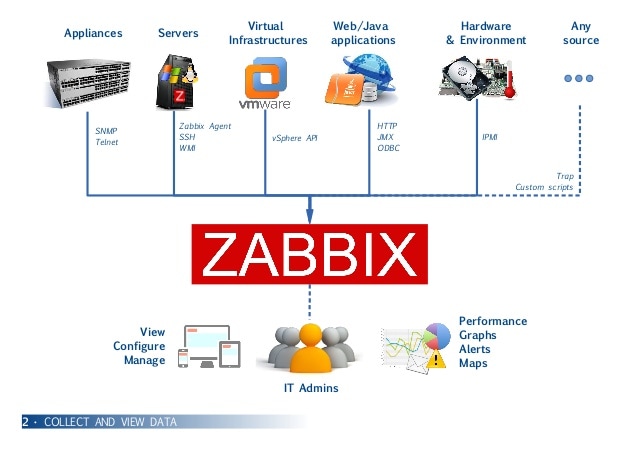
LIBER TECHNOLOGIES

Projet « ANDF – SI Foncier »

*Document d’exploitation Supervision ANDF*

|  |  |
| --- | --- |
| **VOTRE INTERLOCUTEUR COMMERCIAL :**  Abdelkrim EL AKRI  Tél : +33 6 40 15 60 14  abdelkrim.elakri@trustingenierie.com | |
| **VOTRE INTERLOCUTEUR TECHNIQUE :**  Rémi BELLO  Tél : +212 6 54 24 18 24  remi.bello@trustingenierie.com | |
| **Statut :** | DRAFT |
| **Diffusion :** | Restreinte |
| **Version :** | V1-10 |
| **Référence :** | LT\_BENINTF\_MONITORING |

# Stack technique

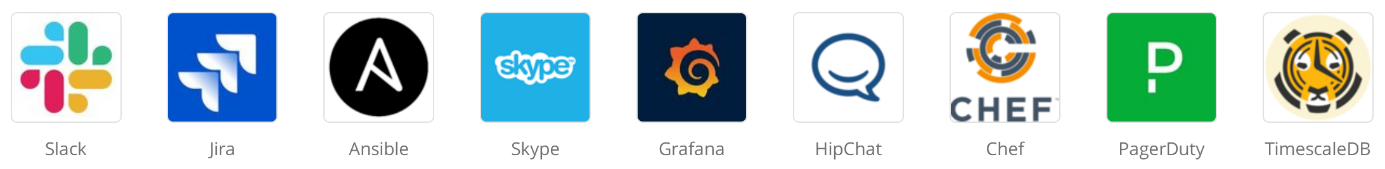
ZABBIX est un logiciel libre (sous licence publique générale GNU version 2) permettant de surveiller l'état de divers services réseau, serveurs et autres matériels réseau et produisant des graphiques dynamiques de consommation des ressources.

Permet de surveiller l’ensemble du réseau, des serveurs et de leurs services :



**Points forts** :

* Solution éprouvée et mature
* Gratuit et open-source
* Packagé avec docker-compose (simplicité de configuration, d’installation et d’exploitation)
* Supporte la supervision en mode pull (le traffic de supervision va de l’ANDF vers le MEF, l’autre sens est impossible)
* Simple à sauvegarder/restaurer (tout contenu dans 1 dossier)
* Intégrable avec de nombreux outils de notification ou de déploiement continu



## Comprendre Zabbix

L’application Zabbix est très bien documentée en line. Les liens suivants peuvent être intéressants :

<https://blog.zabbix.com/zabbix-agent-active-vs-passive/9207/> 🡪 Comprendre les modes actif/passif de l’agent

<https://www.zabbix.com/documentation/1.8/fr/manual/installation/components> 🡪 Liste des composants logiciels et explications sur leurs rôles dans le système

<https://www.zabbix.com/documentation/1.8/fr/manual/config> 🡪 Section consacrée au paramétrage dans la documentation

<https://www.zabbix.com/documentation/1.8/fr/manual/config/host_templates> 🡪 Les templates

<https://www.zabbix.com/documentation/1.8/fr/manual/config/triggers> 🡪 Les déclencheurs

<https://www.zabbix.com/documentation/1.8/fr/manual/config/user_permissions> 🡪 Les permissions utilisateur

Table des matières

[1. Stack technique 2](#_Toc80912967)

[1.1. Comprendre Zabbix 3](#_Toc80912968)

[3. Installation 4](#_Toc80912969)

[3.1. Déployer le serveur 4](#_Toc80912970)

[3.2. Déployer les agents 4](#_Toc80912971)

[3.2.1. Windows 4](#_Toc80912972)

[3.2.2. Linux 5](#_Toc80912973)

[5. CONFIGURATION 7](#_Toc80912974)

[5.1. Hosts 7](#_Toc80912975)

[5.2. Monitorer un service Web 8](#_Toc80912976)

[5.3. Monitorer l’ouverture/fermeture d’un port 8](#_Toc80912977)

[5.4. Monitorer la connexion à internet d’un hôte 9](#_Toc80912978)

[5.5. Détecter des ouvertures de ports 9](#_Toc80912979)

[5.6. Configurer les notifications 9](#_Toc80912980)

# Installation

## Déployer le serveur

Avant d’installer l’application, aller voir chaque fichier « .env\_\* » et paramétrer si besoin.

Notamment le fichier « .env\_web » permet de configurer le nom de l’application (ZBX\_SERVER\_NAME).

L’application s’initialise une première fois avec :

docker-compose up

Les logs de lancement des différents services s’affichent et permettent de vérifier le bon déroulement. Puis arrêter l’application avec « Ctrl+C ». Ce premier lancement a permis de créer le dossier « zbx\_env » qui contient l’ensemble des données à sauvegarder.

**Configurer SSL côté backend**

Dans le répertoire contenant le dossier **zbx\_env**

sudo cp votre-certificat.crt ./zbx\_env/etc/ssl/nginx/ssl.crt

sudo cp votre-certificat.key ./zbx\_env/etc/ssl/nginx/ssl.key

sudo openssl dhparam -out ./zbx\_env/etc/ssl/nginx/dhparam.pem 4096

# sudo chown zabbix:zabbix ./zbx\_env/etc/ssl/nginx/\*

sudo chmod 744 ./zbx\_env/etc/ssl/nginx/\*

L’application peut maintenant être relancée :

docker-compose up -d

Par défaut, connexion en HTTPS sur le port 3000

Login par défaut : Admin/zabbix

Première chose à faire : modifier ce mot de passe ! (dans User settings)

Configurer également le fuseau horaire : Administration > General > GUI > Default time zone

## Déployer les agents

Installation d’un agent sur les serveurs à monitorer, qui va ouvrir le port 10050 sur la machine.

### Windows

<https://www.zabbix.com/documentation/current/manual/installation/install_from_packages/win_msi>

### Linux

Agent 1 :

<https://bestmonitoringtools.com/zabbix-agent-linux-install-on-ubuntu-centos-rhel-debian-rasbian/>

sudo apt -y install zabbix-agent

sudo nano /etc/zabbix/zabbix\_agentd.conf

Modifier les propriétés ci-dessous :

|  |  |
| --- | --- |
| **Propriété** | **Commentaire** |
| Server | Mettre l’IP ou le DNS du serveur zabbix. L’agent va uniquement accepter les requêtes de ce serveur. |
| ListenPort | Port d’écoute : laisser par défaut à « 10050 » |
| ServerActive | En mode passif (zabbix server se connecte à l’agent) laisser commenté.  Pour activer en mode ‘actif’ (l’agent se connecte au serveur zabbix) décommenter et indiquer l’IP ou le DNS du serveur zabbix. |
| Hostname | Nom d’hôte, doit correspondre à la config zabbix. |
| HostMetadata | En configuration standard : « Linux Docker » |
| TLSConnect | « psk » |
| TLSAccept | « psk » |
| TLSPSKIdentity | Donner un nom descriptif par exemple « ZBX-AGENT-MEF-ETERRE »  Personnaliser pour chaque agent déployé. |
| TLSPSKFile | « /etc/zabbix/zabbix\_agent.psk » |

Il faut maintenant générer une clé PSK pour sécuriser les transactions avec le serveur. La librairie OpenSSL est utilisée ici pour générer un chiffre aléatoire.

sudo openssl rand -hex 32

Puis enregistrer la clé dans le fichier :

sudo nano /etc/zabbix/zabbix\_agent.psk

sudo chown zabbix:zabbix /etc/zabbix/zabbix\_agent.psk

sudo chmod 700 /etc/zabbix/zabbix\_agent.psk

Activer le service à chaque démarrage :

sudo systemctl enable zabbix-agent

Et démarrer l’agent :

sudo systemctl restart zabbix-agent

### Agent 2

Agent 2 : apporte le monitoring docker :

<https://medium.com/geekculture/how-to-install-zabbix-agent2-on-linux-c603023207d2>

Exemple pour UBUNTU :

<https://www.zabbix.com/download?zabbix=5.2&os_distribution=ubuntu&os_version=20.04_focal&db=postgresql&ws=nginx>

wget https://repo.zabbix.com/zabbix/5.4/ubuntu/pool/main/z/zabbix-release/zabbix-release\_5.4-1+ubuntu20.04\_all.deb

dpkg -i zabbix-release\_5.4-1+ubuntu20.04\_all.deb

apt update

apt install zabbix-agent2

nano /etc/zabbix/zabbix\_agent2.conf

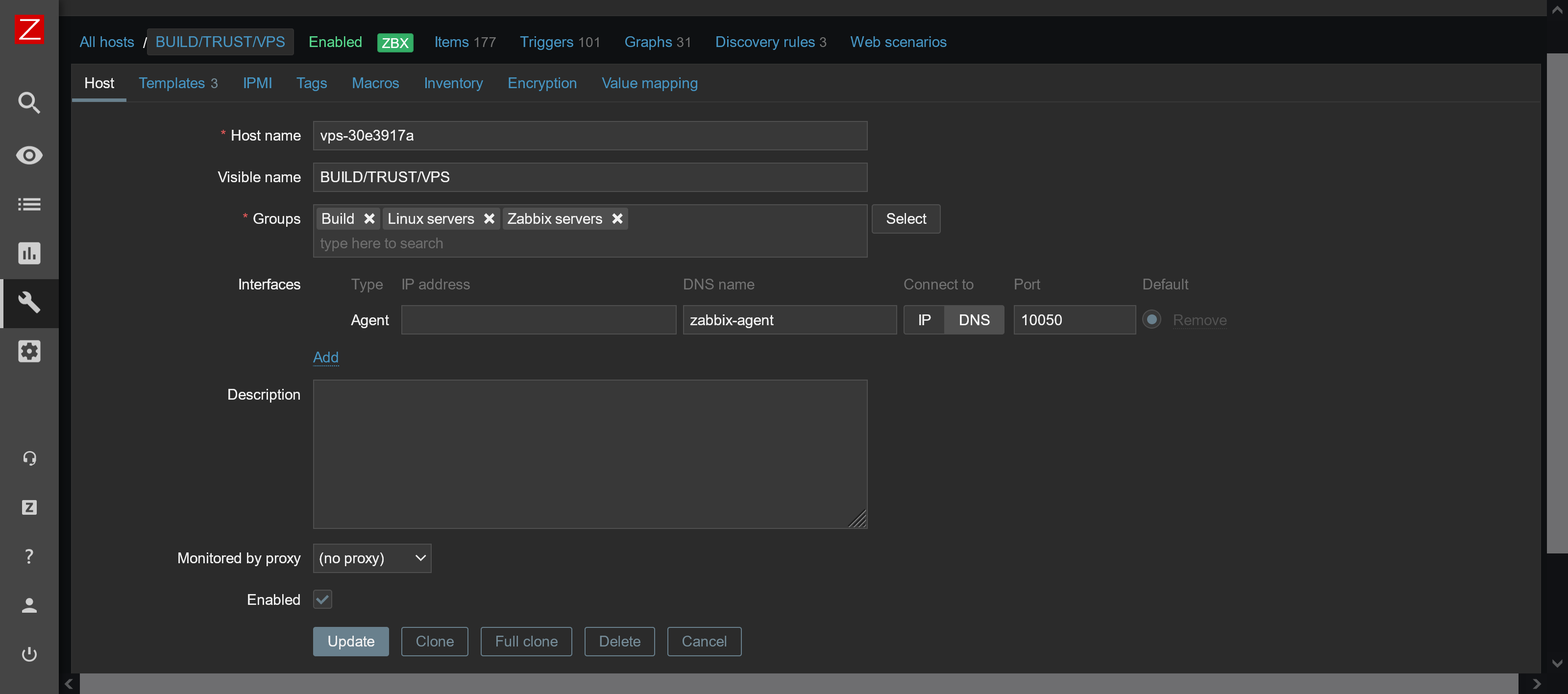
# CONFIGURATION

## Hosts

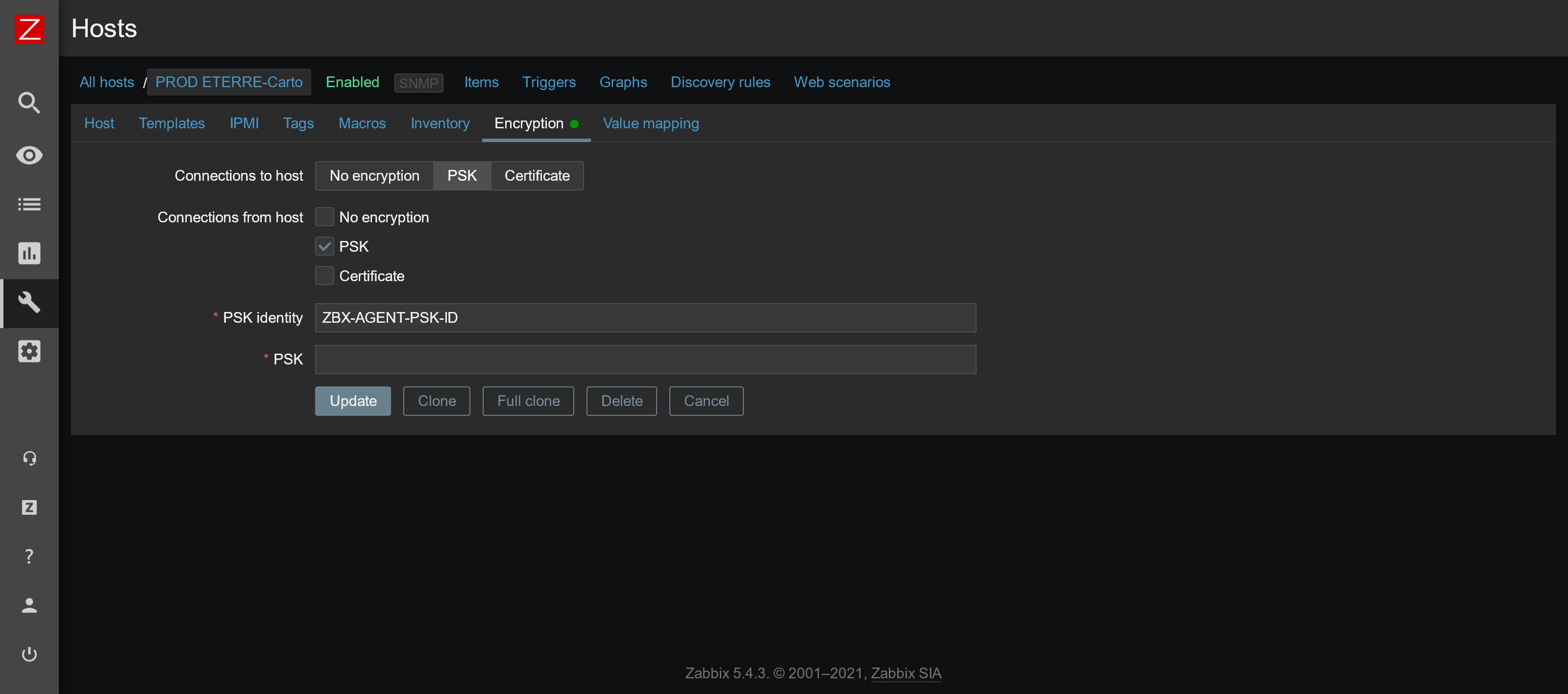
Créer un hôte. Le champ hostname doit correspondre avec la config de l’agent zabbix.

Configurer également une interface :

* De type agent quand un agent peut être déployé
* SNMP sinon
* Renseigner l’IP du serveur et l’utiliser de préférence pour se connecter
* Le DNS peut être utilisé notamment si l’hôte est exposé sur internet



Configurer le chiffrement : mode PSK pour toutes les connexions + remplir identity et PSK.



|  |  |
| --- | --- |
|  | En mode **passif**, on peut vérifier les connexions reçues du serveur par l’agent :  /var/log/zabbix-agent/zabbix\_agentd.log  En mode **actif**, on peut vérifier les connexions provenant des agents en consultant les logs côté serveur :  docker logs zabbix-docker\_zabbix-server\_1 -f |

## Monitorer un service Web

Dans : Configuration > Hosts > (Choisir un hôte) > Web scenarios

Créer un scénario, indiquer l’URL et le code attendu en retour.

Il est possible d’associer chaque scénario à un déclencheur.

Dans : Configuration > Hosts > (Choisir un hôte) > Triggers

Créer un trigger en utilisant l’expression suivante :

* Item : le web scenario (dans la liste des items)
* Fonction : change()
* Valeur : 1

Ce trigger peut ensuite être associé à un service, pour signaler qu’une application est indisponible.

Dans : Configuration > Services > (Choisir un service)

Le trigger se configure dans l’écran de configuration du service.

## Monitorer l’ouverture/fermeture d’un port

Créer un item qui correspond au test sur l’ouverture ou non du port.

Dans : Configuration > Hosts > (Choisir un hôte) > Items

Créer un item avec les paramètres suivants (ici pour tester le port 443) :

* Name : Test TCP Port 443
* Key : net.tcp.service[http,,443]
* Type of information : Numeric (Unsigned)

Créer ensuite un déclencheur (trigger) qui correspond au changement d’état oui/non de l’item.

Dans : Configuration > Hosts > (Choisir un hôte) > Triggers

Créer un trigger en utilisant l’expression suivante :

* Item : celui créé précédemment
* Fonction : change()
* Valeur : 1

Ce trigger peut ensuite être associé à un service, pour signaler qu’une application est indisponible.

Dans : Configuration > Services > (Choisir un service)

Le trigger se configure dans l’écran de configuration du service.

## Monitorer la connexion à internet d’un hôte

Créer un Web Scenario (voir §3.3) qui pointe vers Google ou autre site internet à réponse rapide.

Une solution plus optimale est de tester la connexion avec le protocole DNS moins coûteux, voir : <https://www.zabbix.com/forum/zabbix-help/43679-what-is-a-good-way-to-set-a-trigger-that-checks-for-internet-connectivity>

## Détecter des ouvertures de ports

Piste : <https://share.zabbix.com/cat-app/misc/template-listening-port-discovery>

*Fin du document*