BTS SIO 2018-2019
PPE

Monitoring d’un parc informatique avec Zabbix

NORMAND Aubin
CHARLET Maxence
KEITA Yamoutou

Table des matières

I. Contexte

A. Description de l’entreprise

L’industrie pharmaceutique est un secteur très lucratif dans lequel le mouvement de fusion acquisition est très fort. Les regroupements de laboratoires ces dernières années ont donné naissance à des entités gigantesques au sein desquelles le travail est longtemps resté organisé selon les anciennes structures. Des déboires divers récents autour de médicaments ou molécules ayant entraîné des complications médicales ont fait s'élever des voix contre une partie de l'activité des laboratoires : la visite médicale, réputée être le lieu d' arrangements entre l'industrie et les praticiens, et tout du moins un terrain d'influence opaque.

Le laboratoire Galaxy Swiss Bourdin (GSB) est issu de la fusion entre le géant américain Galaxy (spécialisé dans le secteur des maladies virales dont le SIDA et les hépatites) et le conglomérat européen Swiss Bourdin (travaillant sur des médicaments plus conventionnels), lui même déjà union de trois petits laboratoires . En 2009, les deux géants pharmaceutiques ont uni leurs forces pour créer un leader de ce secteur industriel. L'entité Galaxy Swiss Bourdin Europe a établi son siège administratif à Paris. Le siège social de la multinationale est situé à Philadelphie, Pennsylvanie, aux Etats-Unis.

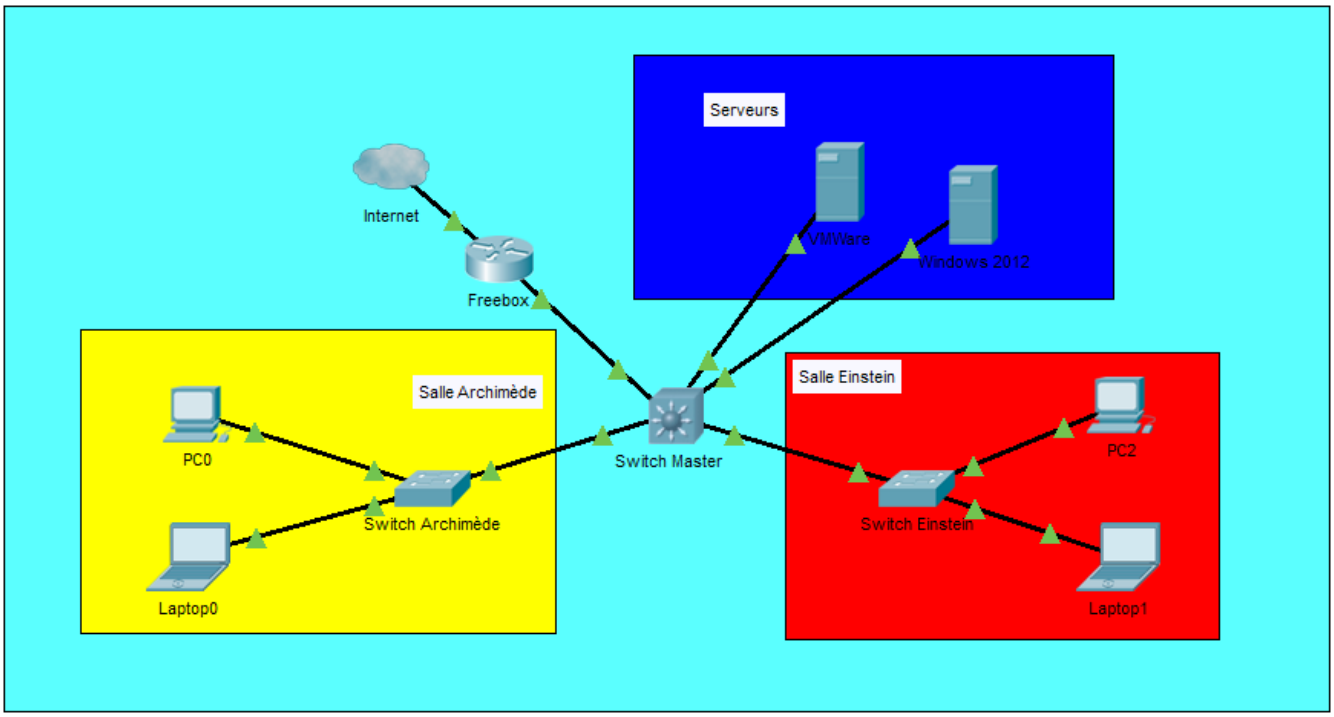
B. Existant

Nous souhaitons superviser le parc de l’école ENSUP qui est composé de :

- 1 serveur Windows 2012-R2 : il a pour rôles Active Directory, DHCP et DNS
- 1 serveur VMWare où des machines virtuelles hébergent d’autre serveurs
- 1 switch master, 2 switchs slaves : le premier s’occupe de la gestion de VLANs, les deux seconds sont les hubs des salles Archimède et Einstein

Une supervision correctement déployée permettra d’anticiper les éventuelles pannes pouvant avoir lieu sur le parc, facilitant ainsi son dépannage et sa maintenance. Elle pourrait également permettre une optimisation plus fine du réseau.

C. Topologie logique



II. Recherche

A. Lexique

- Supervision/Monitoring : surveillance et mesure de performance/disponibilité/intégrité, de façon à pouvoir faire des graphiques ou encore être alerté lors de tout comportement anormal
- Templates : cela correspond à une « trame de fond », par exemple, un template HTML représente l'architecture type que devrait avoir une page web
- LAMP : acronyme désignant un ensemble de logiciels libres permettant de construire des serveurs de site web (Linux, Apache, MySql, PHP/Perl/Python)
- Dépôt (Repository) : Stockage centralisé où les fichiers sont localisés en vue de leur distribution sur le réseau ou encore directement accessible aux utilisateurs (c'est l'endroit où l'on télécharge les paquets Linux)
- Open-source : méthode d'ingénierie logicielle qui consiste à développer un logiciel, ou des composants logiciels, et de laisser en libre accès le code source produit , le plus souvent gratuitement
- Push : l'information est descendante (serveur → agent)
- Pull : l'information est montante (agent → serveur)

B. Choix de la solution logicielle

Afin d'avoir une solution la plus flexible et la moins coûteuse possible, on s'est orienté naturellement vers l'open-source. Pour le système d'exploitation, Linux sera notre choix de prédilection.
Quant aux solutions logicielles permettant de superviser des machines sous Windows comme GNU/Unix, plusieurs sont également gratuites et open-sources. Trois ont retenu notre attention :

	Avantages	Défauts
Nagios (Push)	• Très vieux produit (créé en 1999) il existe de nombreux plugins • La supervision à distance supporte SSH ou SSL • La remontée d'alerte est entièrement paramétrable	• Difficile à installer et configurer (difficile d'ajouter un hôte) • Interface avec trop d'options • Pas de représentation graphique • Les mises à jour de configuration doivent se faire coté supervisé et superviseur
Prometheus (Pull)	• On peut créer ses propres agents • Facile d'installation, pas de configuration à faire • Facile à adapter à la taille de l'organisation • Très flexible d'utilisation, puisque seul le fichier de configuration compte • Technologie récente, donc en vogue	• L'interface graphique fournie n'est pas explicite • On ne peut pas faire de recherche de données • Prometheus fonctionne avec son propre langage • Il faut renseigner tous les agents en statique dans un fichier de configuration
Zabbix (Pull/Push)	• Interface graphique claire et modifiable • Possibilité de créer ses propres graphiques, avec des données en particulier • Mise à jour de la configuration uniquement du côté superviseur, via l'interface web	• Interface avec énormément d'options, la notion de templates n'est pas facile à appréhender au début • La communication des données est en clair par défaut, l'emploi d'un certificat (ou de clé privée) reste cependant possible

	• Les agents sont simples d'installation et légers • Les sondes et tests disponibles sont extrêmement variés : on peut superviser un terminal, serveur en passant par les applications web • Créé par des administrateurs systèmes pour des administrateur systèmes (templates et règles)	

Après avoir peser le pour et le contre, nous avons retenu Zabbix pour sa flexibilité et sa facilité d'installation.

III. Installation

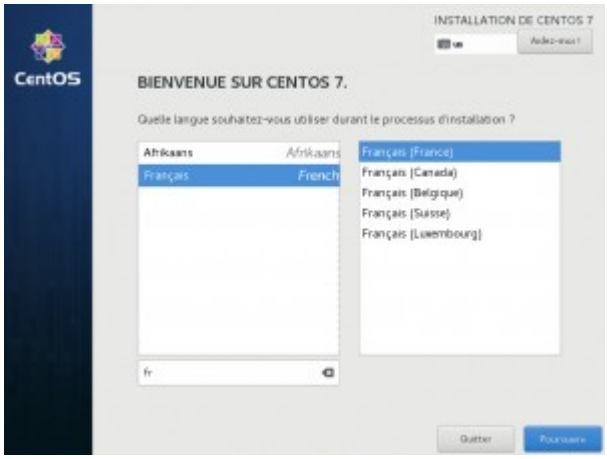
A. Installation LAMP

Le système d'exploitation retenu sera CentOS 7, pour sa stabilité, sa documentation et ses similitudes avec RedHat, distribution orientée entreprise

1. CentOS 7

Une fois l'iso de CentOS 7 téléchargée, on peut lancer l'installation (ici, il s'agit d'une machine virtuelle Virtualbox) :

- Choix de la langue : Pour commencer, il est nécessaire de choisir la langue (ici Français, clavier français azerty)



- <u>Configuration du système :</u> Il faut ensuite renseigner plusieurs paramètres, à savoir le partitionnement du disque dur, le nom du système et l'interface réseau et les utilisateurs et mots de passe associés

- <u>Partitionnement du disque dur :</u> la configuration par défaut est suffisante, donc pas besoin d'y toucher

- <u>Nom du système et interface réseau :</u> Il est nécessaire d'activer l'interface réseau en haut à droite (sous peine de ne pas être connecté à quelque réseau que ce soit!), puis de modifier le nom de l'hôte (par défaut localhost.localdomain)

- Utilisateurs et mot de passe associés : Après avoir validé les paramètres précédents, il nous est présenté l'écran ci-dessous pendant le processus d'installation, il faut donc créer un mot de passe pour le compte *root*, qui a tous les droits, mais également un compte basique pour des raisons de sécurité

Une fois l'installation finie, il ne nous reste plus qu'à redémarrer. Suite à ce redémarrage, on peut rentrer les commandes suivantes :

```
yum update -y
yum install -y open-vm-tools
```

2. Apache, PostgreSQL et PHP

On commence par installer les paquets et dépôts nécessaires pour télécharger/compiler :

```
yum install -y yum-utils wget epel-release gcc
```

Ensuite, on installe les paquets nécessaires pour les paquets requis par back-end de Zabbix, déjà disponibles sur les repos CentOS :

```
yum install -y postgresql pgsql pgsql-devel php-gd php-gd-devel php-bcmath php
bcmath-devel php-ctype php-ctype-devel php-xml php-xml-devel libxml libxml-devel
php-xmlreader php-xmlreader-devel php-xmlwrite php-xmlwrite-devel php-session
openldap openldap-devel OpenIPMI OpenIPMI-devel php-net-socket php-net-socket-devel
php-mbstring php-xmlwrite-devel php-gettext php-gettext-devel php-ldap php-ldap-
devel php-pgsql libpcre3 libevent libevent-devel zlib zlib-devel libssh2 libssh2-
devel fping fping-devel net-snmp net-snmp-devel java java-devel openssl postgresql-
server openssl-devel gnutls gnutls-devel postgresql-contrib postgresql-devel
libxml2-devel libcurl-devel unixODBC unixODBC-devel httpd
```

Une fois ces paquets installés, on peut télécharger et installer les paquets non disponibles sur les dépôts de CentOS :

```
wget https://repo.zabbix.com/non-supported/rhel/7/x86_64/iksemel-1.4-
2.el7.centos.x86_64.rpm
wget https://repo.zabbix.com/non-supported/rhel/7/x86_64/iksemel-devel-1.4-
2.el7.centos.x86_64.rpm
```

```
wget https://repo.zabbix.com/non-supported/rhel/7/x86_64/iksemel-utils-1.4-2.el7.centos.x86_64.rpm
rpm -Uvh iks*
```

Une fois cela fait, on peut récuperer le code source de Zabbix. On commence par créer un dossier zabbix à la racine, puis on télécharge l'archive contenant le code avant de la décompresser :

```
cd /
mkdir zabbix
cd /zabbix
wget https://sourceforge.net/projects/zabbix/files/ZABBIX%20Latest%20Stable/4.0.0/zabbix-4.0.0.tar.gz
tar -zxvf zabbix-4.0.0.tar.gz
```

On crée un groupe et un utilisateur « zabbix » qui seront utilisés pour Zabbix et la base de données :

```
groupadd zabbix
useradd -g zabbix zabbix
```

On initialise notre base de données, on crée un utilisateur ainsi qu'une database pour les données que Zabbix stockera :

```
cd /zabbix/zabbix-4.0.0/database/postgresql
postgresql-setup initdb
systemctl enable postgresql
systemctl start postgresql
sudo -u postgres createuser --pwprompt zabbix (choisir un mot de passe !)
sudo -u postgres createdb -O zabbix -E Unicode -T template0 zabbix
```

On peut maintenant créer les tables de la façon suivante à partir des templates de Zabbix :

```
cat schema.sql | sudo -u zabbix psql zabbix
cat images.sql | sudo -u zabbix psql zabbix
cat data.sql | sudo -u zabbix psql zabbix
```

On peut maintenant compiler, avec toutes les options que l'on aura retenues :

```
cd /zabbix/zabbix-4.0.0
./configure --enable-server --enable-agent --with-postgresql --with-net-snmp --with-ssh2 --enable-ipv6 --with-libcurl --with-libxml2 --with-jabber --enable-java --with-ldap --with-openssl --with-openipmi –with-unixodbc
make install
```

On peut ensuite copier notre front-end dans le bon répertoire :

```
mkdir /var/www/html/zabbix
cd /zabbix/zabbix-4.0.0/frontends/php
cp -a . /var/www/html/zabbix
```

Il ne nous manque plus qu'à modifier les fichiers suivants:

- /etc/php.ini :

```
php_value max_execution_time 300
php_value memory_limit 128M
php_value post_max_size 16M
```

```
php_value upload_max_filesize 2M
php_value max_input_time 300
php_value max_input_vars 10000
php_value always_populate_raw_post_data -1
php_value date.timezone Europe/Paris
```

- /zabbix/zabbix-4.0.0/conf/zabbix_server.conf :

```
DBHost=localhost
DBName=[Nom de la table]
DBUser=zabbix
DBPassword=[Mot de passe de l’utilisateur zabbix]
```

- /etc/selinux/config :

Après avoir désactiver SELinux à chaud pour éviter tout problème de droit au niveau de notre serveur web (avec la commande `setenforce 0`), on modifie le fichier de configuration afin que SELinux ne soit pas réactivé suite à un redémarrage :

```
SELINUX=disabled
```

- /var/lib/pgsql/data/pg_hba.conf :

Il faut changer les **METHOD** en **md5** ou **trust**.

- /var/ww /html/zabbix /zabbix.conf.php (fichier à créer) :

```
<?php // Zabbix GUI configuration file. global $DB; $DB['TYPE'] = 'POSTGRESQL';
$DB['SERVER'] = 'localhost'; $DB['PORT'] = '5432'; $DB['DATABASE'] = 'zabbix';
$DB['USER'] = 'zabbix'; $DB['PASSWORD'] = '1234'; // Schema name. Used for IBM DB2
and PostgreSQL. $DB['SCHEMA'] = ''; $ZBX_SERVER = 'localhost'; $ZBX_SERVER_PORT =
'10051'; $ZBX_SERVER_NAME = ''; $IMAGE_FORMAT_DEFAULT = IMAGE_FORMAT_PNG; >
```

B. Installation Zabbix

Avant de pouvoir finir l’installation de Zabbix au travers de l’interface web, il faut autoriser le trafic au niveau du pare-feu (pour y avoir accès et pour que Zabbix puisse récupérer les données de ses futurs agents). Fiwalld est le pare-feu par défaut de CentOS 7, les commandes sont donc les suivantes :

```
firewall-cmd --permanent –-add-port=5432/tcp
firewall-cmd --permanent --add-port=10050/tcp
firewall-cmd --permanent --add-service=http
firewall-cmd --reload
```

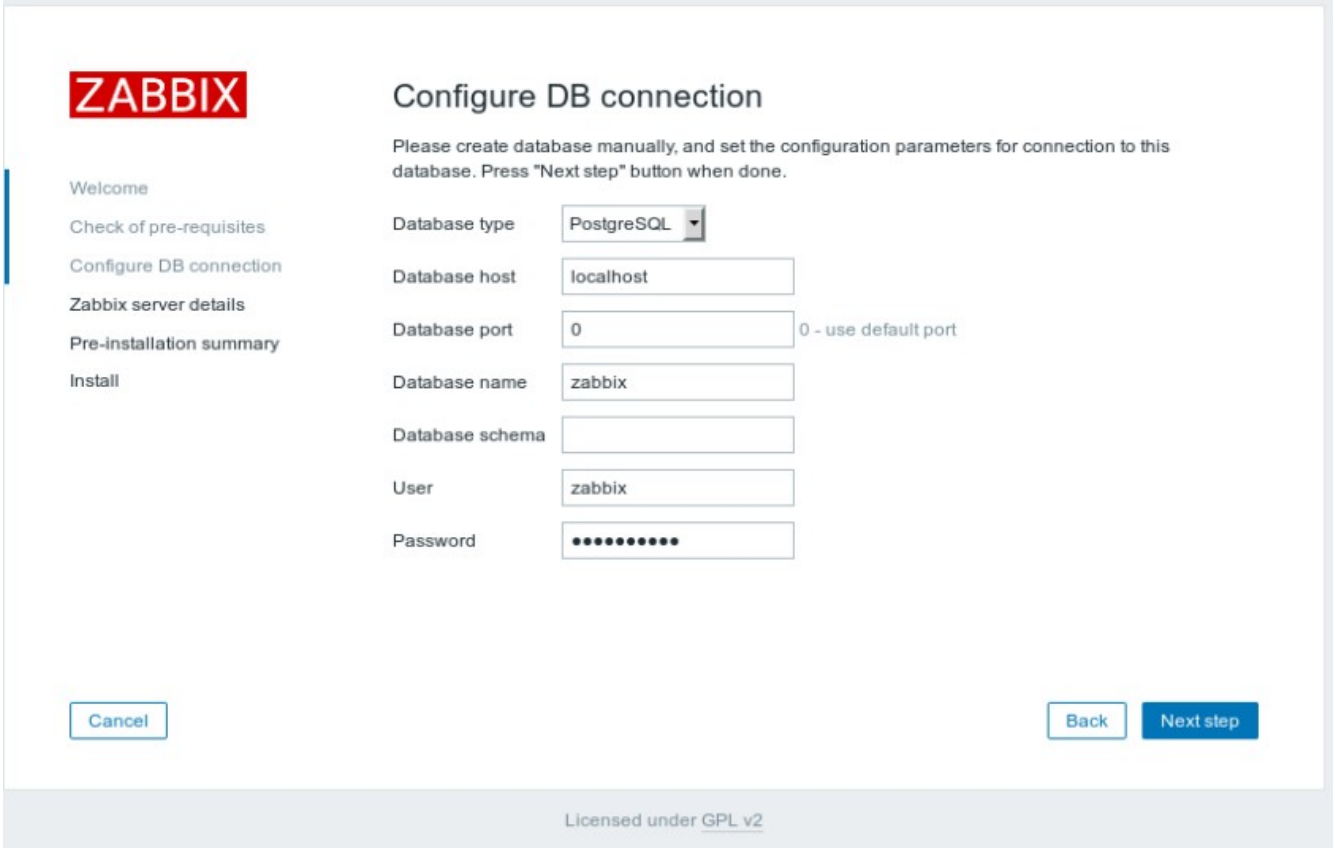
On peut maintenant démarrer les services web, ainsi que les services Zabbix :

```
/usr/local/sbin/zabbix_server
/usr/local/sbin/zabbix_agentd
systemctl restart httpd
```

IV. Configuration

A. Fin d'installation

Une fois le service lancé, on peut se rendre sur l'interface web à l'adresse http://localhost/zabbix. Une fois cela fait, on peut cliquer sur suivant jusqu'à tomber sur l'écran ci-dessous, qui est bien sur à remplir :



On peut alors cliquer sur suivant jusqu'à la fin de l'installation, pour ensuite se connecter au front-end de Zabbix et enfin rentrer dans le vif du sujet (les identifiants par défaut sont **Admin**/**zabbix)**.

B. Lexique

- SNMP (Simple Network Management Protocol): protocole permettant aux administrateurs réseau de gérer les équipements du réseau et de diagnostiquer les problèmes de réseau
- Agent: programme qui observe et reporte le comportement d'un équipement

- Auto-enregistrement (auto-registration): capacité à automatiser l'enregistrement d'un équipement
- Trigger : signifiant *détente* (d'une arme à feu) ou encore *actionner* en anglais, il s'agit d'un évènement entraînant l'exécution d'une action

C.Ajouts d'hôtes

1. Via agent

a. Machine à superviser

On peut télécharger le nécessaire depuis le site de Zabbix, qui propose des agents pré-compilés. Ici, on prendra l'exemple de l'agent pour Windows.

On commence par télécharger l'archive zip disponible sur le site de Zabbix, avant de décompresser son contenu dans un dossier zabbix à la racine du disque. Il suffit alors juste de modifier le fichier de configuration (zabbix_agentd.win.conf) en renseignant les paramètres suivants :

- Server = [IP DU SERVEUR]
- ListenPort = 10050
- Hostname = [NOM DE LA MACHINE]

On peut alors lancer la commande suivante, dans un invite de commande lancé en tant qu'administrateur :

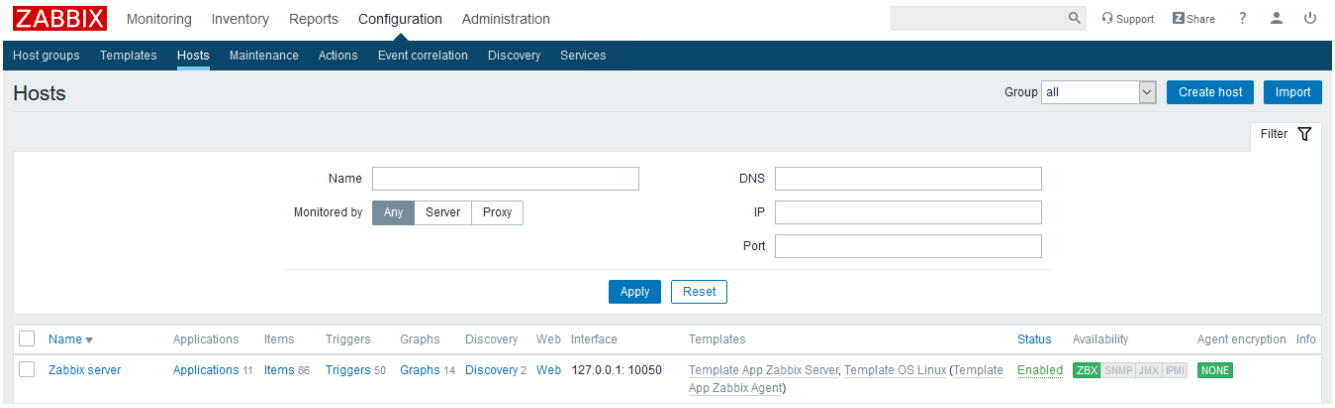
```
C:\> c:\zabbix\zabbix_agentd.exe -c c:\zabbix\zabbix_agentd.win.conf -i
```

Une fois cela fait, il suffit de s'assurer que le port 10050 soit bien ouvert avant de lancer le service :

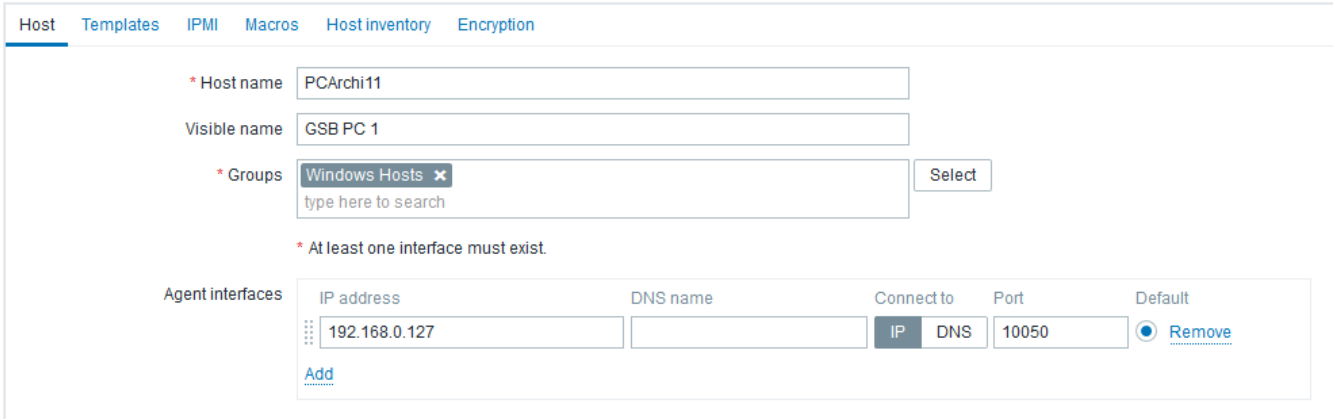
```
netsh advfirewall firewall add rule name="Open Port 10050 in" dir=in action=allow
protocol=TCP localport=10050
netsh advfirewall firewall add rule name="Open Port 10050 out" dir=out action=allow
protocol=TCP localport=10050

C:\> c:\Zabbix\zabbix_agentd.exe --config C:\Zabbix\zabbix_agentd.conf --start
```

b. Serveur

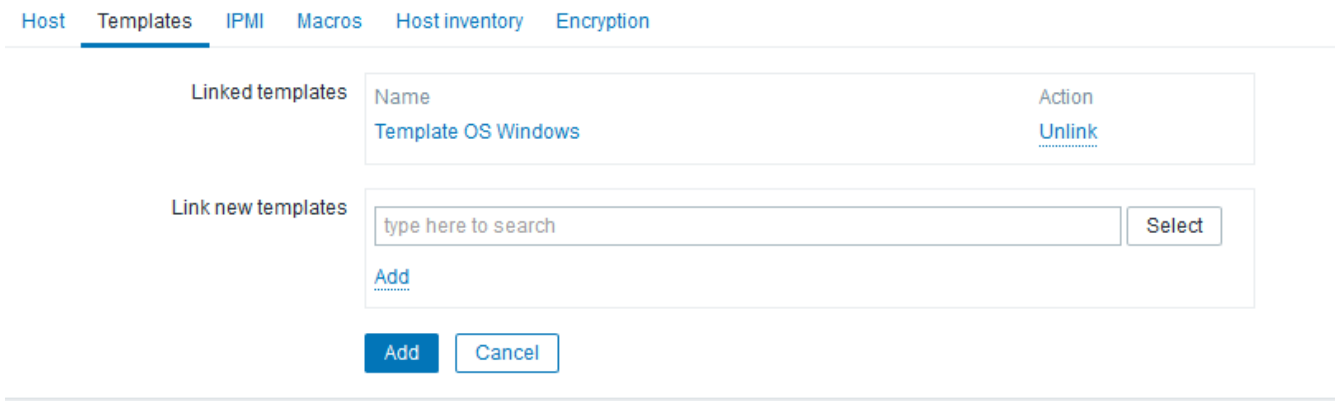
Sur l'interface web, il faut se rendre dans **Configuration → Hosts**, puis, cliquer sur **Create host**



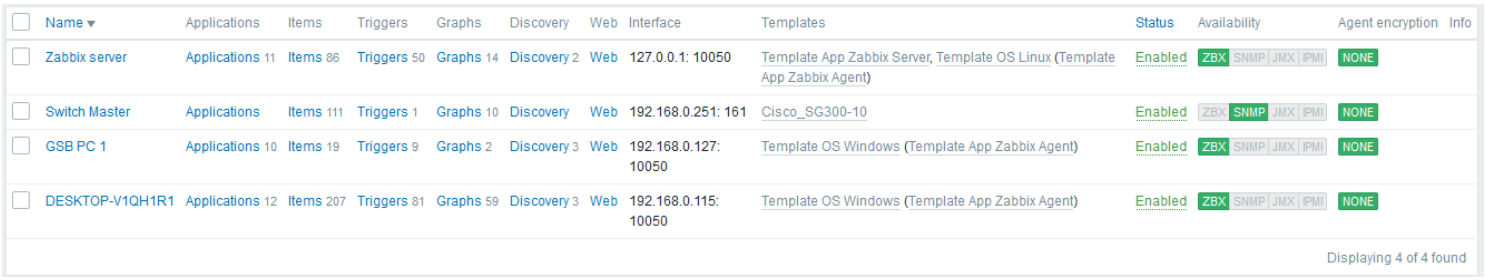
Ensuite, il nous faut indiquer le **Host name** (nom de la machine) les **Groups** et l'adresse IP de la machine :



On se rend ensuite dans l'onglet **Templates**, et on sélectionne le template **OS Windows** avant de cliquer sur **Add** (celui en dessous du champ **Link new templates**) :



On peut alors cliquer sur **Add** (le bouton en bas) et voir notre machine être bien prise en compte par Zabbix (il faut que le petit bouton **ZBX** soit en vert) :



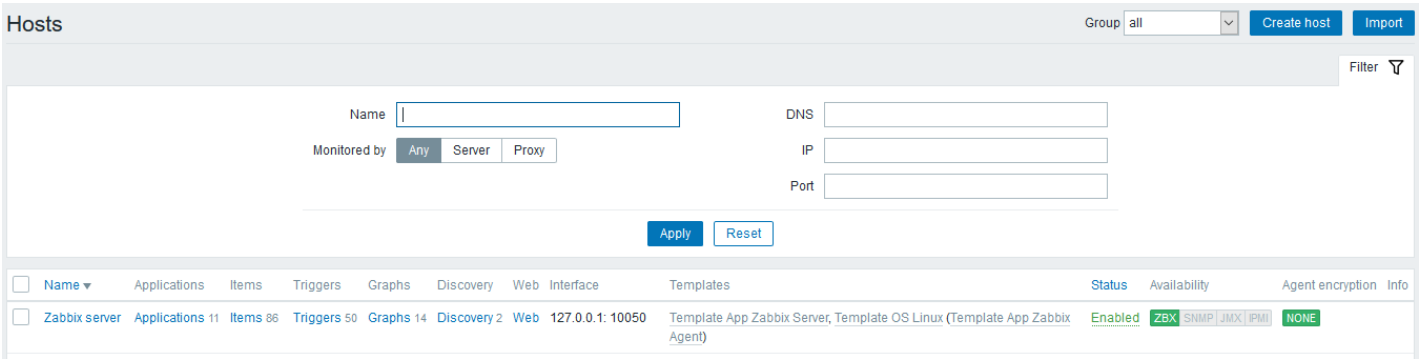
2. Via SNMP

a. Machine à superviser

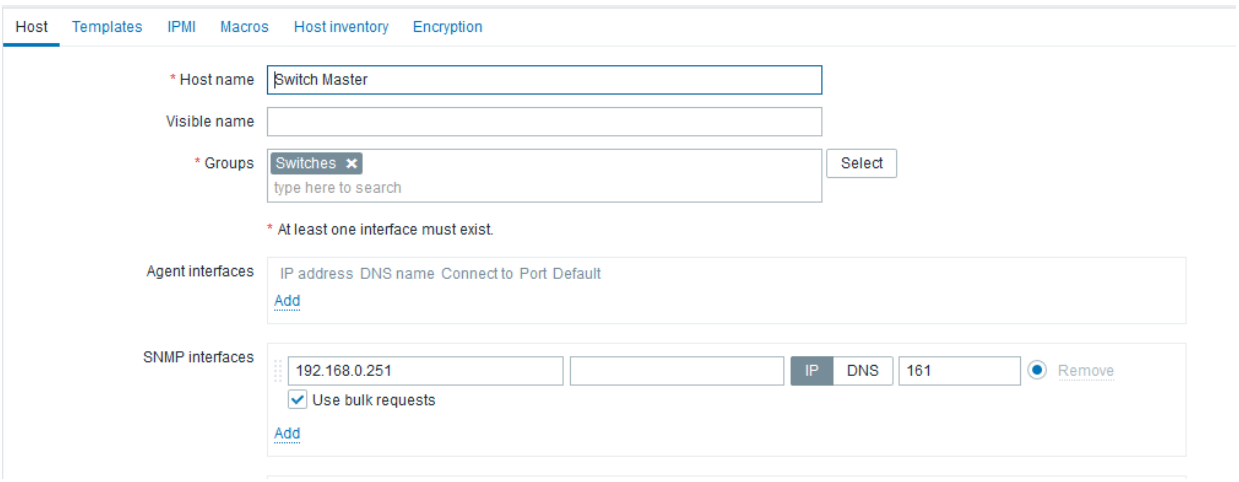
Il faut activer le protocole SNMP sur la machine à superviser, car ce protocole n'est pas activé par défaut. Il faut donc suivre la documentation de la machine en question.

b. Serveur

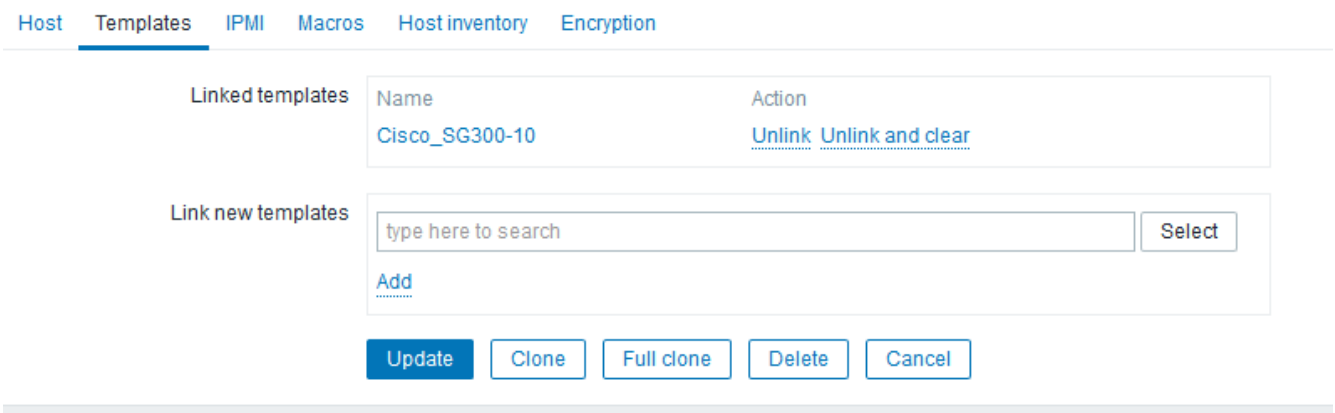
Sur l'interface web, il faut se rendre dans **Configuration → Hosts**, puis, cliquer sur **Create host :**



Ensuite, il nous faut indiquer le **Host name** (nom de la machine) les **Groups** et l'adresse IP de la machine, mais cette fois ci, dans **SNMP Interfaces** :



On se rend ensuite dans l'onglet **Templates**, et on sélectionne le template correspondant à notre équipement (ici Cisco SG300-10) avant de cliquer sur **Add** (celui en dessous du champ **Link new templates**) :

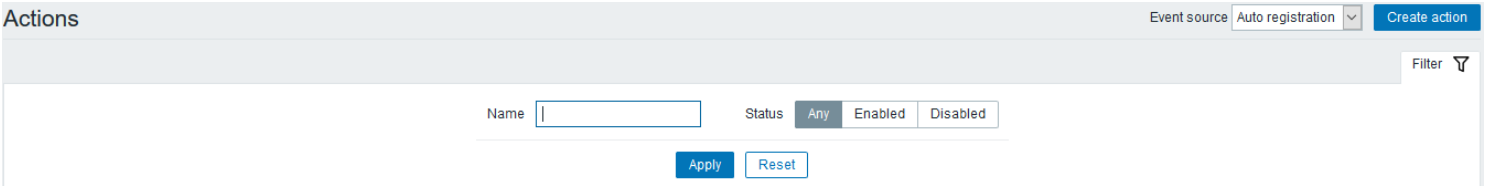


On peut alors cliquer sur **Add** (le bouton en bas) et voir notre machine être bien prise en compte par Zabbix (il faut que le petit bouton **SNMP** soit en vert) :

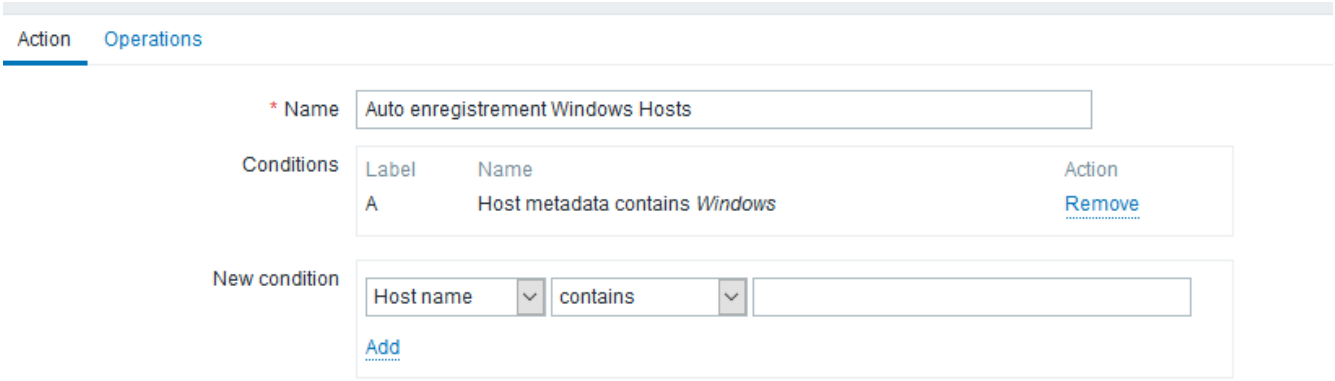
3. Création de règles d'auto-enregistrement

L'un des majeurs soucis avec les méthodes d'ajouts vues ci-dessus et que l'on doit ajouter les machines une à une, ce qui prendrait un temps bien trop long. Zabbix dispose de règles qui permettent de réagir avec une action prédéfinie à un événement. Ainsi, si une certaine condition est remplie, alors l'action se lance. On peut donc faire en sorte que si une machine contacte notre serveur, ce dernier réagisse en conséquence. Voyons l'exemple d'une règle pour ajouter des machines sous Windows.

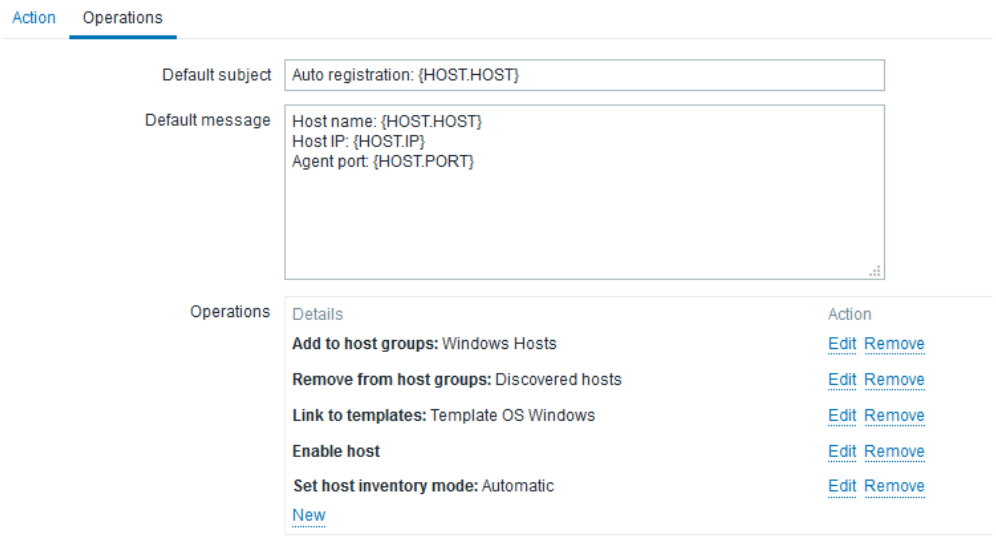
On commence par se rendre dans **Configuration → Action**, puis on clique sur le bouton **Create action** :



On établit la condition *Host name contains Windows* (littéralement, le nom d'hôte contient Windows) avant de cliquer sur **Add** :



On se rend ensuite dans l'onglet **Operations** avant de définir la réaction que devra adopter le serveur :



On peut cliquer sur le bouton **Add**. Une fois ces règles définies, plus besoin d'ajouter manuellement les machines côtés serveurs. L'utilisation d'un script (fichier bat) et d'un Active Directory permettrait le déploiement de l'agent sur la totalité du parc, ce qui réduit grandement le nombre d'opérations nécessaires.

C. Événements et alertes

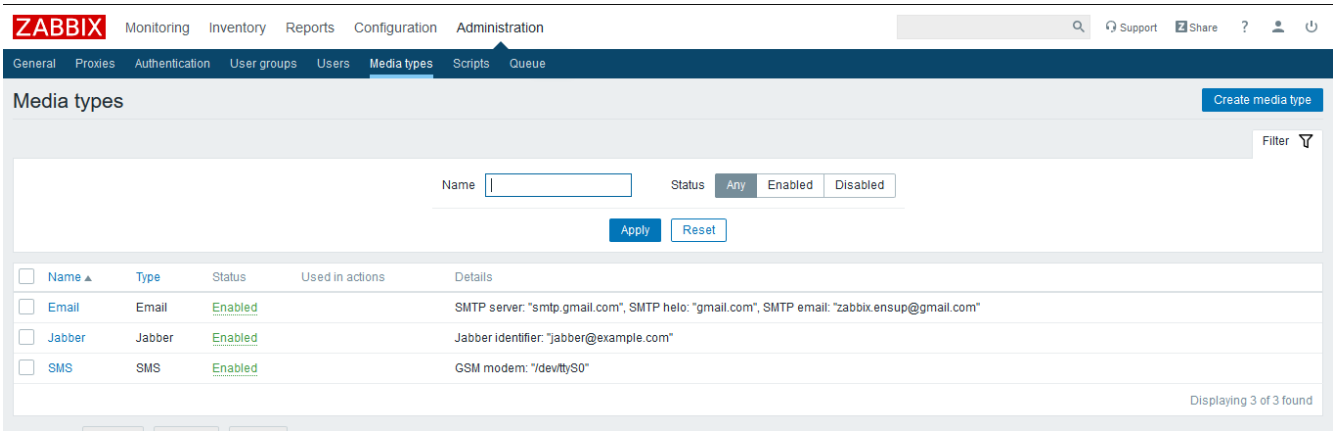
L’une des forces de tout outil de supervision est sa capacité à paramétrer un système d’alerte de façon à pouvoir réagir le plus rapidement possible (voir anticiper) tout comportement anormal du système d’information.

Ce qui fait que Zabbix est un excellent outil de monitoring, c’est son système d’information qui est réglable à souhait : chaque donnée qui est récupérée peut servir d’élément déclencheur (de trigger) à une action ou alerte. Les trois moyens d’être informé sont par Email, Jabber et SMS. Ici, on ne verra que le premier, mais les deux autres sont également possible et non-exclusifs.

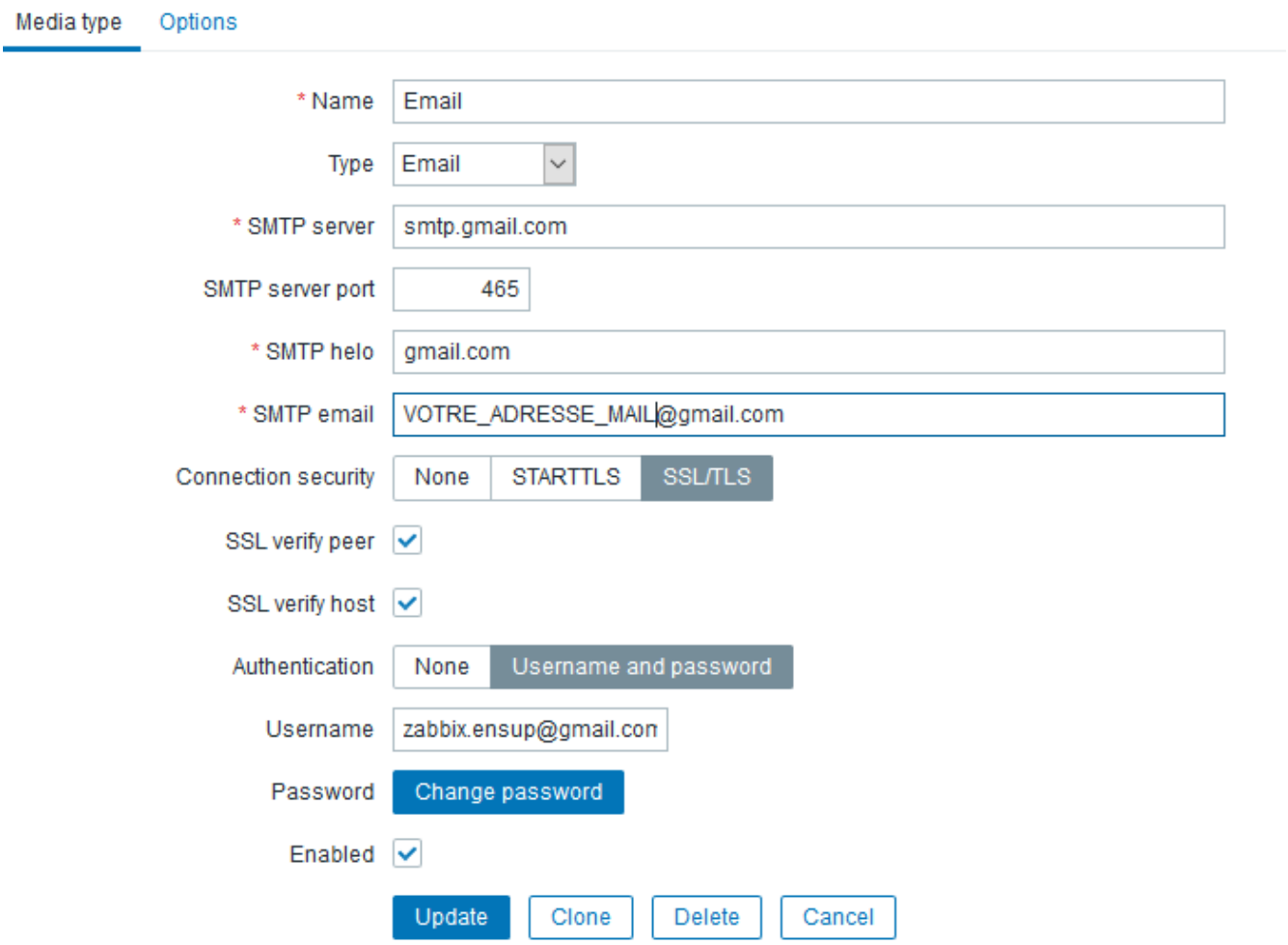
1. Paramétrage des adresses d’envoi et de réception

a. Envoi

Rendons-nous dans **Administration → Media types :**

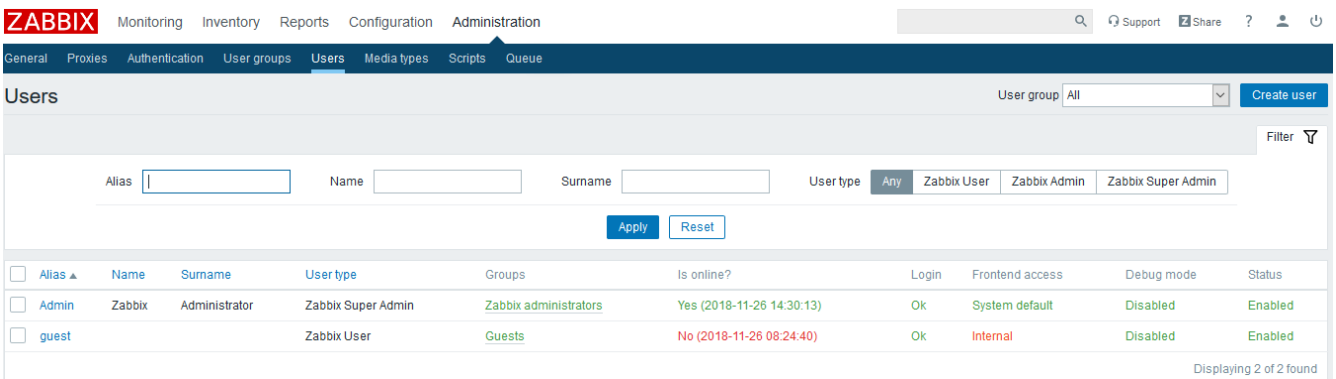


On clique alors sur **Email** afin de pouvoir paramétrer notre mail d’envoi ; nous avons pris les réglages pour un compte Gmail, mais un serveur de messagerie autre (voire interne) fonctionnerait tout aussi bien:

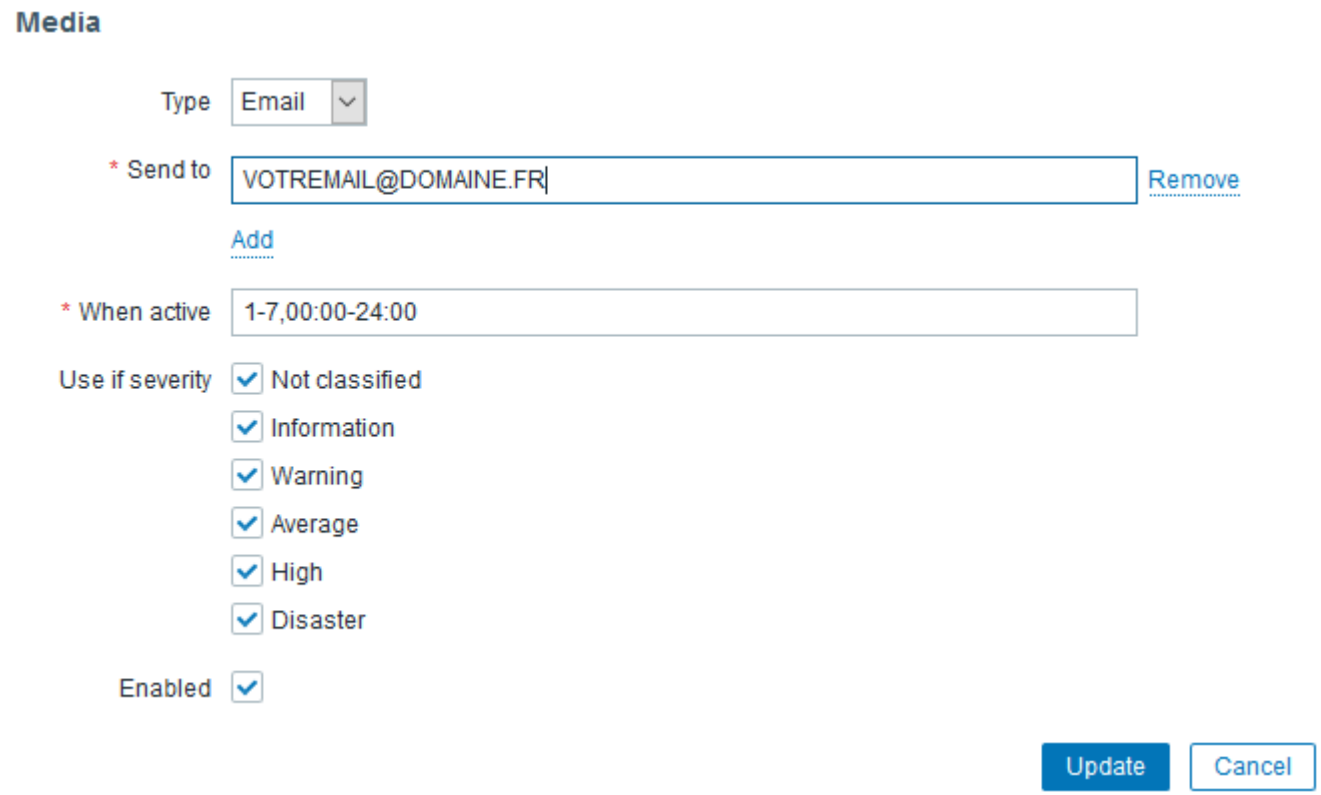


b. Réception

Cette fois-ci, on se rend dans **Administration → Users** :



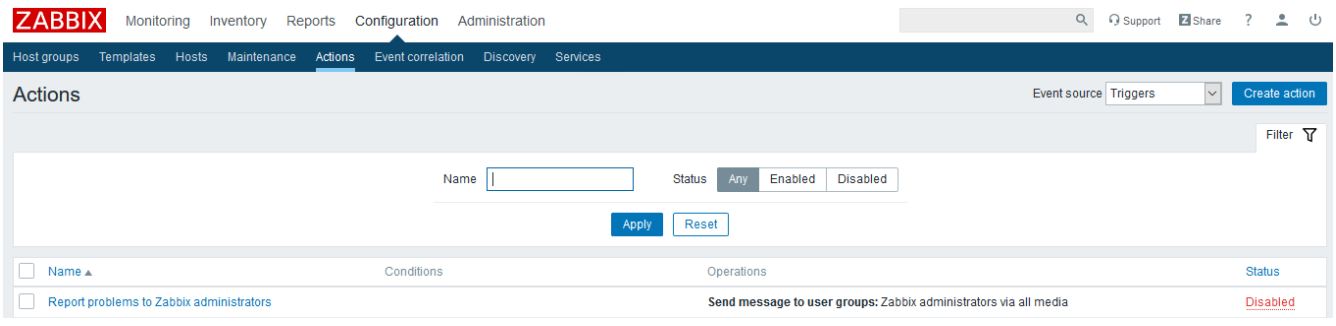
On peut alors cliquer sur l'utilisateur à contacter (dans notre cas, ce sera l'utilisateur Admin) :



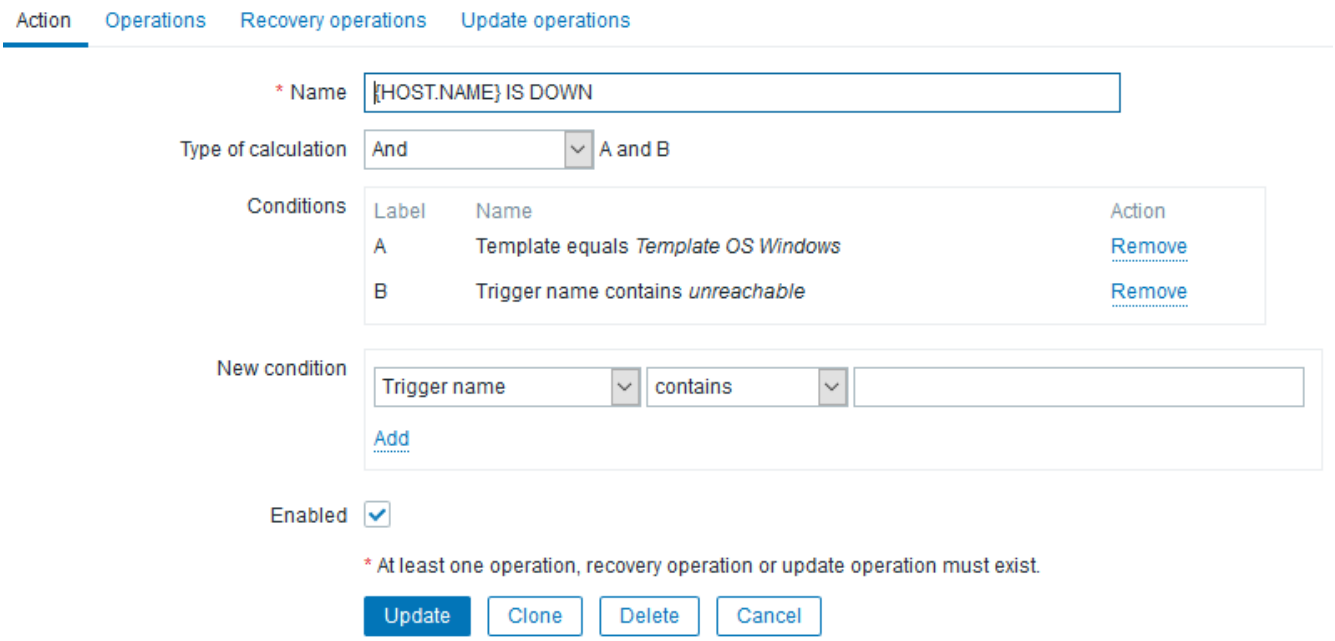
N .B : On peut choisir d'alerter un utilisateur en fonction du niveau de sévérité du trigger que l'on aura choisi, de sorte à pouvoir établir des niveaux de support.

2. Réglage d'action

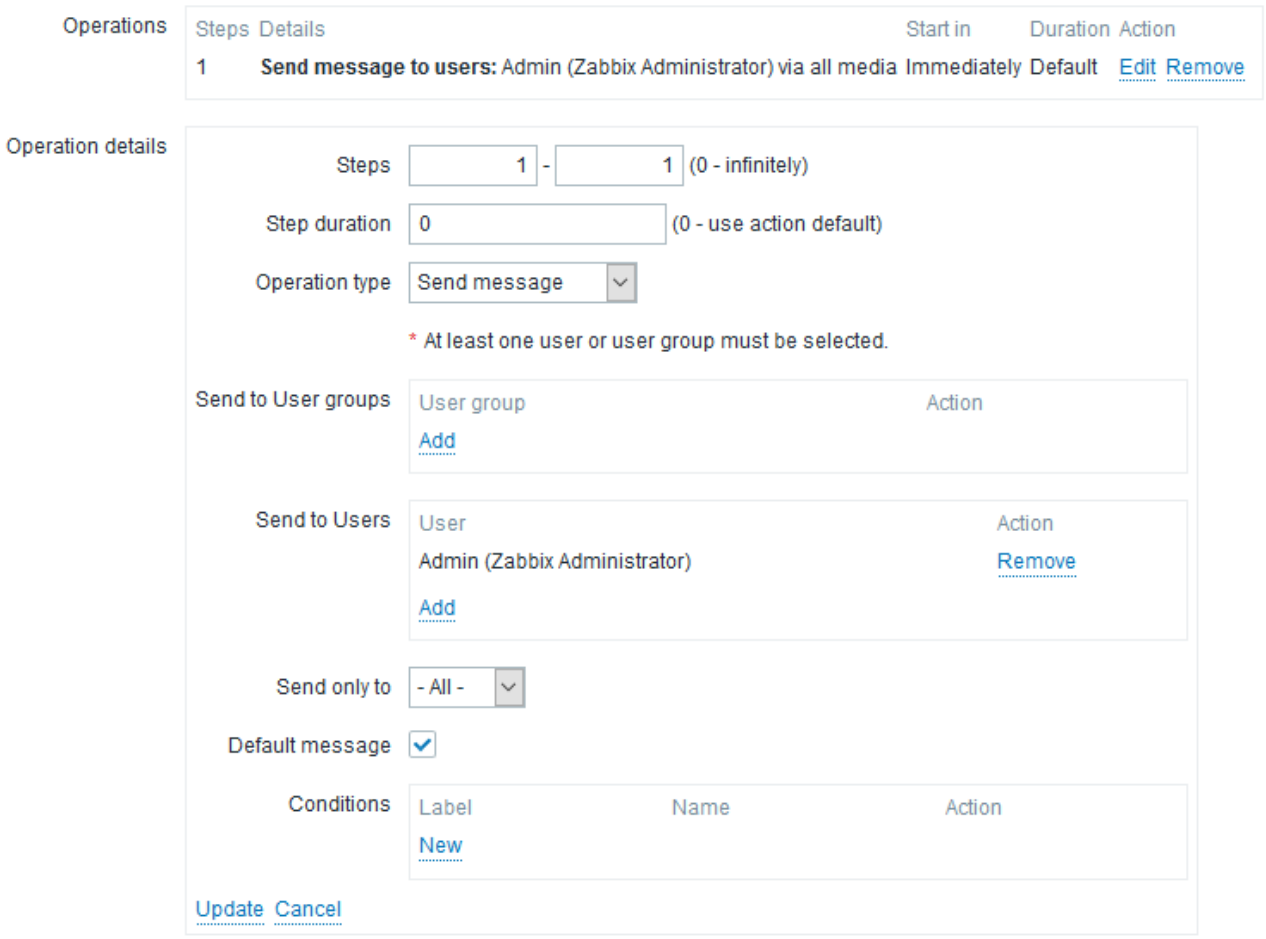
Pour configurer une action, il faut se rendre dans **Configuration → Actions** :



On clique ensuite sur **Create action** :



On peut ensuite ajouter des conditions (avec les opérateurs logiques ET /OU). Ici, nos conditions sont « Le template sélectionné est *Template OS Windows* et le trigger contient le mot *inatteignable* ». Ainsi, lorsque qu'une machine étant sous le template *OS Windows* et étant inatteignable par ICMP, l'action pourra se déclencher. Encore faut-il la définir… Pour cela, il faut se rendre dans l'onglet **Operations** :



On n’a juste à indiquer l’utilisateur à contacter si le trigger est activé.

V. Conclusion

A. Faisabilité du projet

Grâce à une très grande communauté, ainsi qu’une extrême flexibilité et une excellente documentation, Zabbix, bien que pouvant être déstabilisant de par ses notions (templates, triggers, actions…) mais reste très simple d’utilisation, adapté aux petits comme au grand systèmes d’informations, puisque l’on peut ajouter des hôtes manuellement ou automatiser cette tâche grâce à des outils tels qu’Active Directory et les règles. De plus, les alertes et règles sont entièrement paramétrables et les templates le sont également mais peuvent surtout être téléchargés, la communauté n’hésitant pas à partager.

On retiendra également le fait que l’utilisation de certificat ou l’établissement d’un lien avec un LDAP est possible.

B. Retour d’expérience

Le fait d’utiliser Zabbix nous a permit de découvrir le système d’exploitation CentOS 7, qui est d’une grande stabilité, de découvrir la supervision ainsi que ses possibilités (prévention/dépannage/sécurité) ainsi que l’intérêt d’automatiser au maximum ce qui peut l’être (gain de temps, plus d’erreur humaine à l’exception de la configuration). Le fait de l’avoir compiler depuis le code source, d’avoir du modifier pas mal de réglages (pare-feu, PostgreSQL …) nous a permit de voir et revoir pas mal de notions connexes à notre sujet.