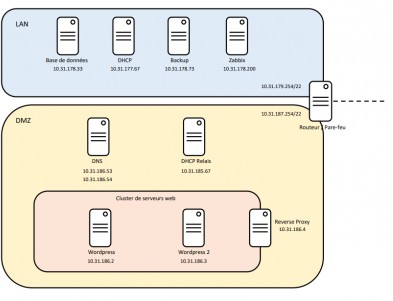
[Accueil](https://sisr2.beaupeyrat.com/doku.php?id=sisr2-asie%3Astart)

[Situation Professionnelle 1 Lucie](https://sisr2.beaupeyrat.com/doku.php?id=sisr2-asie%3Asp1_lucie) [Situation Professionnelle 2 Lucie](#_bookmark0)

# I) Introduction et Contexte

La société GSB souhaite sécuriser le nouveau site internet de la ﬁliale japonaise. Pour ce faire, un second serveur web ainsi qu'un reverse proxy seront créés aﬁn de mettre en place le load balancing. La solution IPS Fail2ban sera mise en place aﬁn de protéger les serveurs web. Enﬁn, la solution de supervision Zabbix sera mise en place aﬁn de monitorer le cluster de serveurs web.

Voici le schéma du réseau comprenant chaque machine utilisée pour cette situation professionnelle :



# HAProxy

1. **Création du reverse proxy et du second serveur web**

Nous clonons dans un premier temps la VM template et lui attribuons une nouvelle adresse IP (10.31.186.4 (haproxy)) et un nouveau nom d'hôte dans le DHCP :

dhclient -r && dhclient -v

# Change le nom d'hôte à l'aide d'une commande hostnamectl set-hostname srv-web1-1 # Ou srv-web2-1

# Change le nom d'hôte directement depuis le fichier de configuration nano /etc/hosts

Nous modiﬁons dans le ﬁchier /etc/hosts le nom d'hôte de la machine et nous nous déconnectons pour actualiser l'aﬃchage du nom.

Nous clonons ensuite notre serveur web pour créer notre second serveur web identique qui aura pour IP 10.31.186.3 et pour nom d'hôte wordpress-sp2.

1. **Conﬁguration du reverse proxy**

Nous installons les paquets nécessaires :

apt update && apt upgrade apt install haproxy

Nous démarrons le service haproxy :

systemctl start haproxy systemctl enable haproxy

Nous modiﬁons le ﬁchier de conﬁguration. Nous ferons en sorte que le reverse proxy soit conﬁguré en mode transparent, c'est-à-dire qu'il laissera passer les requêtes HTTP aﬁn que le serveur web fasse lui-même la redirection HTTPS. Nous gèrerons également les cookies de sorte que lorsqu'une session est ouverte, nous ne soyons pas changé de serveur web à l'envoie d'une autre requête. Nous conﬁgurerons également le load balancing de sorte que le serveur ayant reçu le moins de requêtes reçoive la requête.

frontend http-in

bind \*:80 mode http

#http-request redirect scheme https code 301 unless { ssl\_fc } default\_backend http-servers

frontend https-in

bind \*:443 mode tcp

default\_backend https-servers

backend http-servers

mode http

cookie SERVERID insert indirect nocache

# Redirige les requêtes sur le serveur ayant le moins reçu de

requêtes

balance leastconn

# L'argument check permet de vérifier l'état d'un serveur et de le sortir du load balancing en cas de problème

server wordpress-sp1 10.31.186.2:80 check server wordpress-sp2 10.31.186.3:80 check

backend https-servers

mode tcp

cookie SERVERID insert indirect nocache balance leastconn

server wordpress-sp1 10.31.186.2:443 check server wordpress-sp2 10.31.186.3:443 check

Nous redémarrons le service :

systemctl restart haproxy

1. **Modiﬁcation DNS**

Pour permettre à notre reverse proxy d'effectuer la balance des charges, nous devons rediriger les requêtes du DNS vers ce dernier. Nous changeons donc dans nos DNS le ﬁchier db.japon.gsb.org :

$TTL

604800;

@ IN SOA ns2-1-pub.gsb.org. root.gsb.org. ( 2023101701;

43200;

3600;

3600000;

172800 );

; Adresse de la zone (le ping japon.gsb.org pinguera cette adresse) @ IN A 10.31.186.4;

; DNS Server (on définit les DNS d'autorité pour cette zone) @ IN NS ns2-1-pub.gsb.org.;

@ IN NS ns2-2-pub.gsb.org.;

; Association adresse IP des DNS avec leurs noms (si on ping ces noms on pinguera ces adresses)

ns2-1-pub IN A 10.31.186.53;

ns2-2-pub IN A 10.31.186.54;

; Association adresse IP du cluster de serveurs avec www (le ping [www.japon.gsb.org](http://www.japon.gsb.org/) pinguera ces adresses)

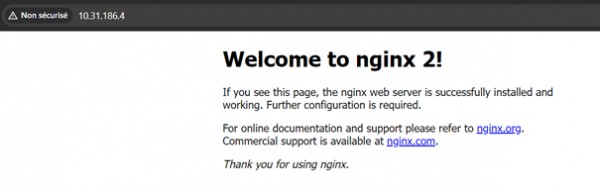
www IN A 10.31.186.4;

nous redémarrons notre service :

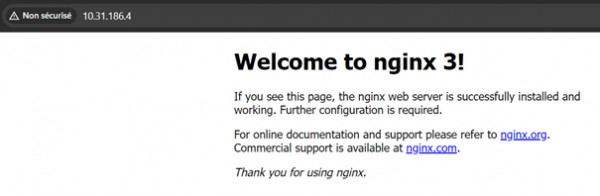
systemctl restart bind9

Pour vériﬁer que le load balancing soit bien opérationnel, nous modiﬁons les pages d'accueil d'Nginx. Nous nous rendons ensuite sur notre navigateur pour rentrer l'adresse IP de notre reverse proxy :

web-1 (10.31.186.2) :



web-2 (10.31.186.3) :



# Fail2ban

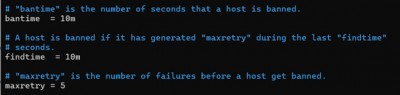
Aﬁn de sécuriser notre nouveau cluster de serveurs web, nous mettrons en place la solution IPS Fail2Ban, permettant de bannir des adresses IP après un certain nombre de tentatives de connexion échouées. Nous commençons par installer le paquet nécessaire :

apt install fail2ban

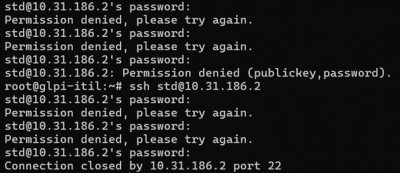
Nous dupliquons son ﬁchier de conﬁguration pour créer une copie sur laquelle on effectuera nos changements. En effet, le ﬁchier de conﬁguration par défaut risque d'être écrasé lors de mises à jour.

cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local

Nous modiﬁons notre nouveau ﬁchier de conﬁguration pour l'adapter à notre besoin :

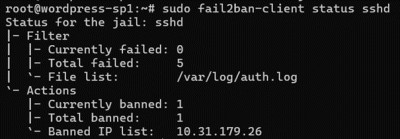


Testons d'échouer une connexion avec un client :



Nous aﬃchons la liste des IP bannies :

fail2ban-client status sshd



Nous pouvons débannir notre IP à l'aide de la commande suivante et vériﬁer la liste des IP bannies :

fail2ban-client set sshd unbanip <IP> fail2ban-client status sshd



# Zabbix

1. **Installation du serveur**

Nous clonons dans un premier temps la VM template et lui attribuons une nouvelle adresse IP (10.31.178.200) et un nouveau nom d'hôte dans le DHCP :

dhclient -r && dhclient -v

# Change le nom d'hôte à l'aide d'une commande hostnamectl set-hostname zabbix-sp2

# Change le nom d'hôte directement depuis le fichier de configuration nano /etc/hosts

Nous modiﬁons dans le ﬁchier /etc/hosts le nom d'hôte de la machine et nous nous déconnectons pour actualiser l'aﬃchage du nom.

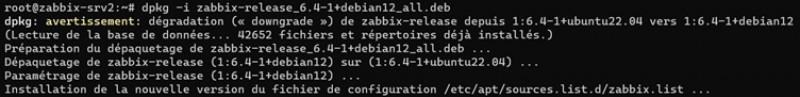
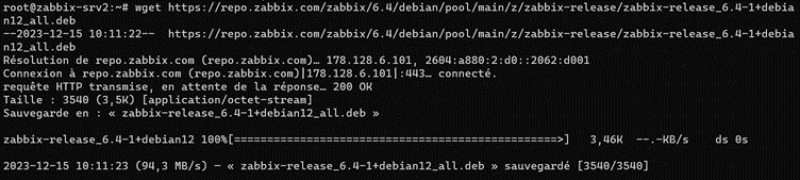
# Conﬁguration du serveur

Dans un premier temps, nous devons installer le dépôt oﬃciel de Zabbix. En effet, ce dernier n'est pas par défaut sur les machines Debian et les paquets téléchargeables sont des paquets obsolètes. Nous entrons donc les commandes ci-dessous :

wget

https://repo.zabbix.com/zabbix/6.4/debian/pool/main/z/zabbix-release/zabbix- release\_6.4-1+debian12\_all.deb

dpkg -i zabbix-release\_6.4-1+debian12\_all.deb



Nous installons ensuite les paquets nécessaires à l'utilisation de Zabbix. Notons que Zabbix a besoin d'un serveur web :

apt update && apt upgrade

apt install zabbix-server-mysql zabbix-frontend-php zabbix-apache-conf zabbix-sql-scripts zabbix-agent apache2 php php-mysql php-mysqlnd php-ldap php-bcmath php-mbstring php-gd php-pdo php-xml libapache2-mod-php

L'outil de supervision Zabbix a besoin de fonctionner avec une base de données. Nous utiliserons la base de données présente sur le réseau LAN. Il n'y a donc pas besoin d'ouvrir de ﬂux dans notre pare- feu étant donné que les machines sont sur le même réseau et peuvent communiquer entre elles.

Nous créons donc sur notre serveur de bases de données notre nouvelle base de données :

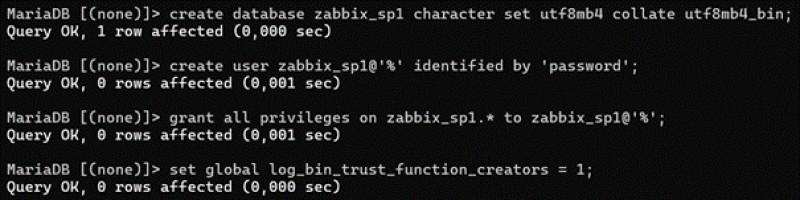
mysql -u root -p

create database zabbix\_sp1 character set utf8mb4 collate utf8mb4\_bin; create user zabbix\_sp1@'%' identified by 'password';

grant all privileges on zabbix\_sp1.\* to zabbix\_sp1@'%';

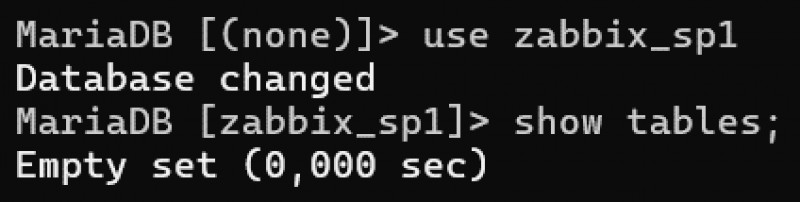
# Cette commande nous permettra d'obtenir les droits pour remplir la base de données à distance via la commande zcat

set global log\_bin\_trust\_function\_creators = 1;



Nous vériﬁons que notre base de données soit créée. Pour l'instant, aucune table n'est créée. Notre base de données est vide :

use zabbix\_sp1 show tables;

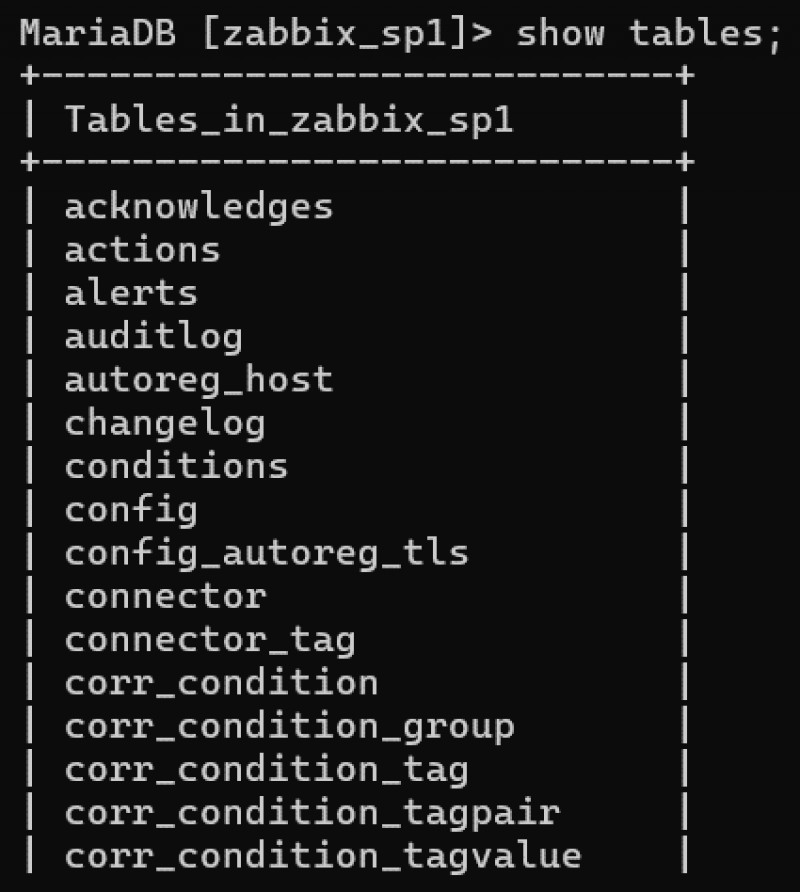


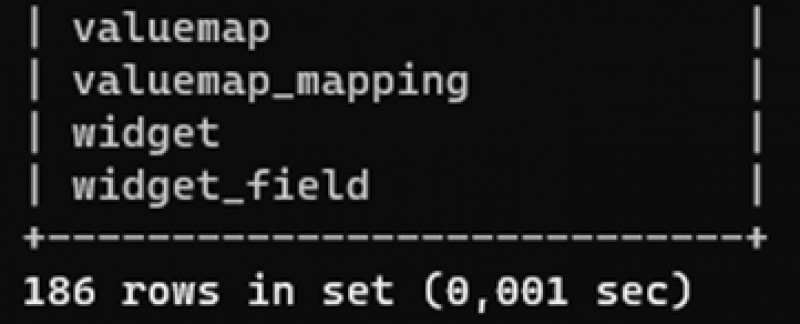
Nous retournons sur le serveur Zabbix et entrons la commande zcat. Cette commande permet de remplir la base de données que nous avons créé précédemment

zcat /usr/share/zabbix-sql-scripts/mysql/server.sql.gz | mysql --default- character-set=utf8mb4 -h 10.31.178.33 -uzabbix\_sp1 -ppassword zabbix\_sp1

Nous retournons sur notre serveur de bases de données pour vériﬁer que notre nouvelle base soit bien complétée :

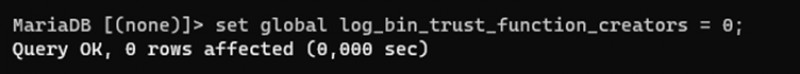
use zabbix\_sp1 show tables





Nous devons maintenant désactiver la variable globale MariaDB permettant d'obtenir des privilèges spéciﬁques :

set global log\_bin\_trust\_function\_creators = 0;

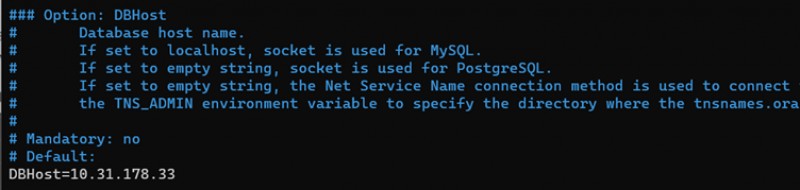


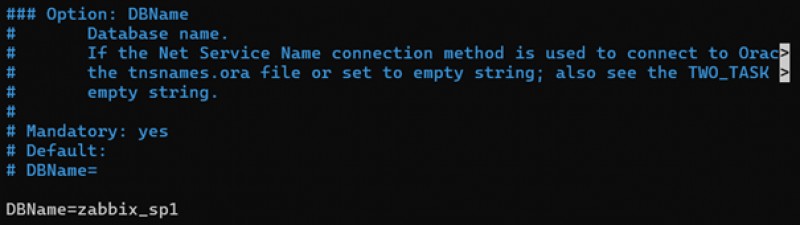
Nous modiﬁons le ﬁchier de conﬁguration zabbix\_server.conf :

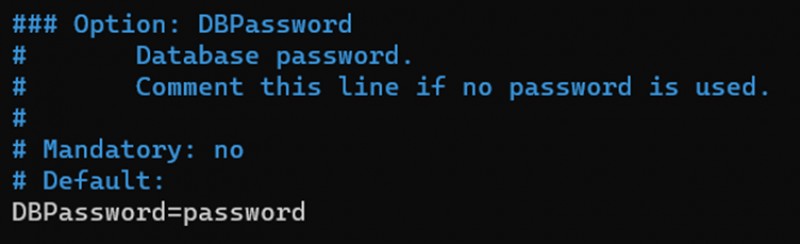
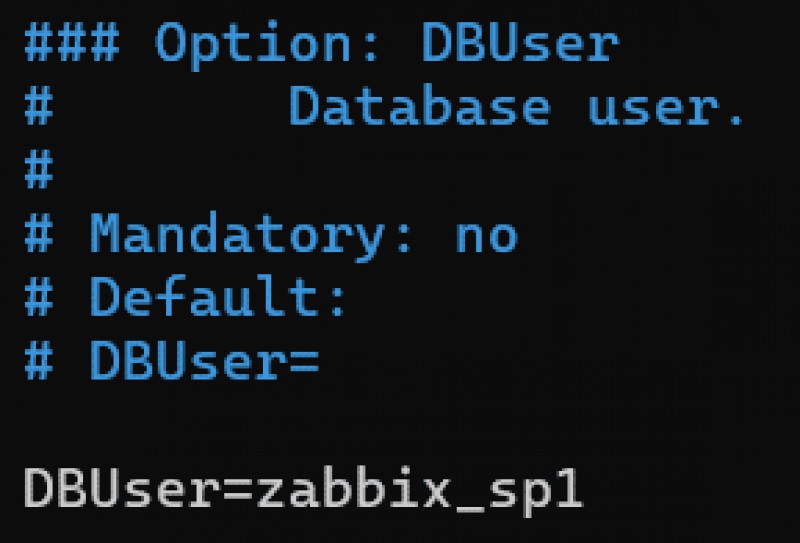
nano /etc/zabbix/zabbix\_server.conf

Nous devons vériﬁer que les éléments suivants soient correctement remplis et les modiﬁer si nécessaire :

- DBHost=10.31.178.33 - DBName=zabbix\_sp1 - DBUser=zabbix\_sp1 - DBPassword=password





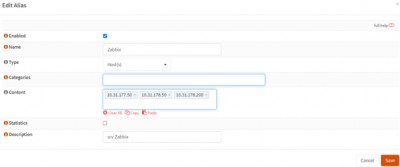


Nous pouvons redémarrer le service zabbix-server :

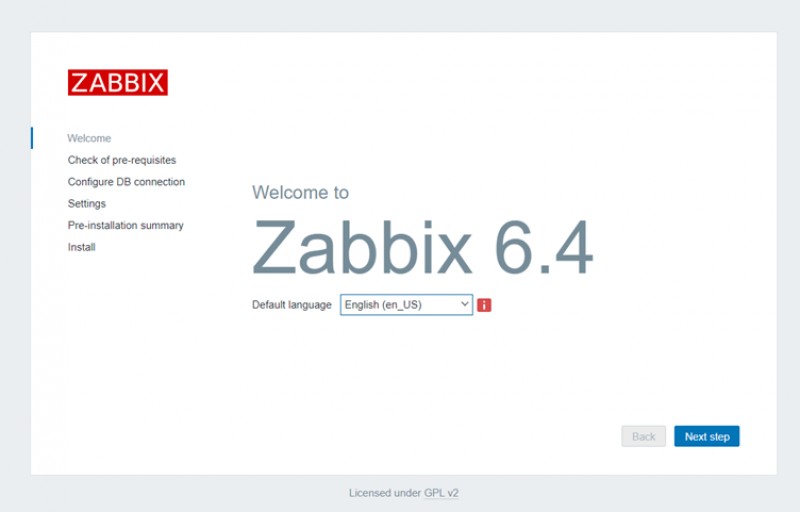
systemctl restart zabbix-server

# Installation via l'interface web

Pour accéder à l'interface graphique, nous devons ajouter notre serveur à l'alias Zabbix dans notre pare-feu, puis le redémarrer aﬁn de prendre en compte le changement dans la règle.

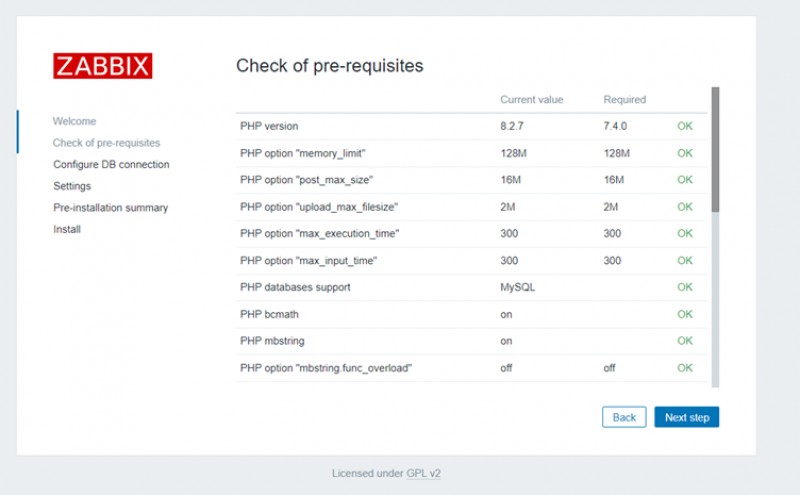


Nous nous rendons sur l'interface graphique d'installation via le lien <http://10.31.178.200/zabbix>: Nous laissons l'interface en anglais pour l'installation.



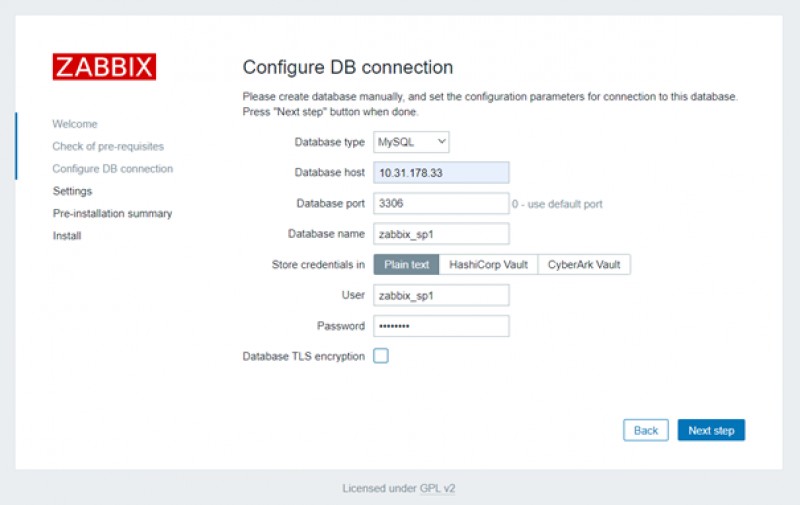
Nous regardons si tous les paquets nécessaires sont installés. Dans le cas contraire, nous les

installons avant de démarrer l'installation de Zabbix.

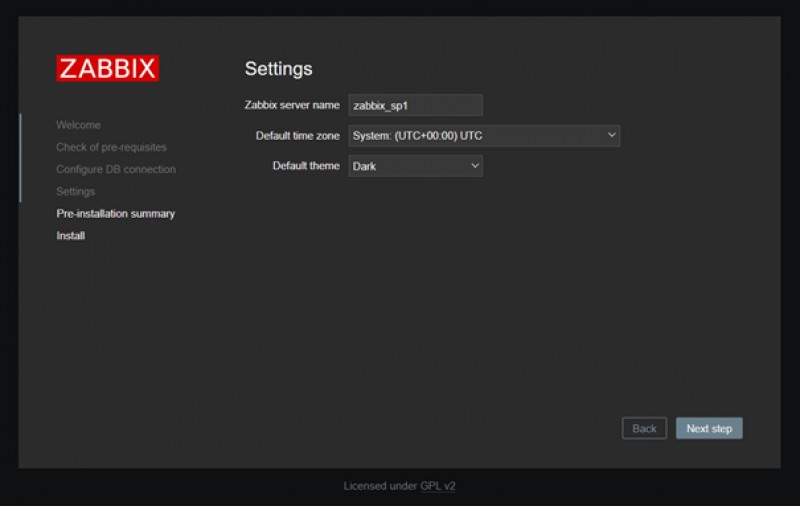


En cas de problème de langue, suivre les étapes de conﬁguration de langue trouvables après la partie sur l'installation de Zabbix.

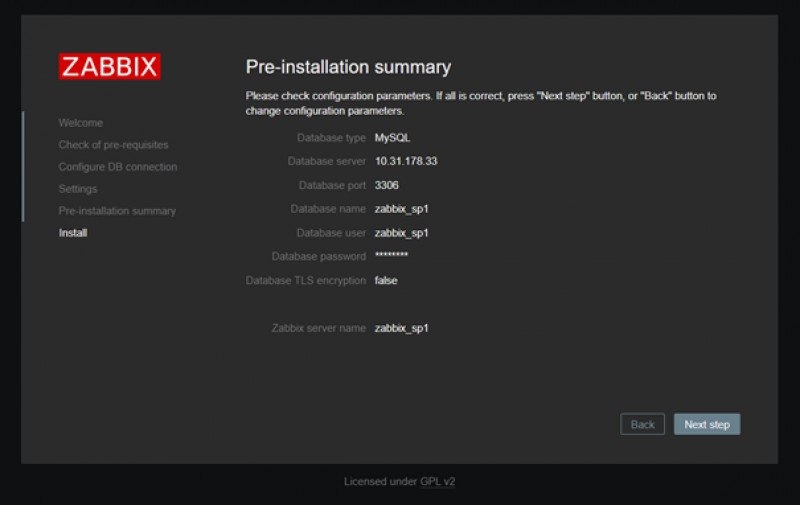
Nous entrons les informations pour connecter Zabbix à notre base de données.



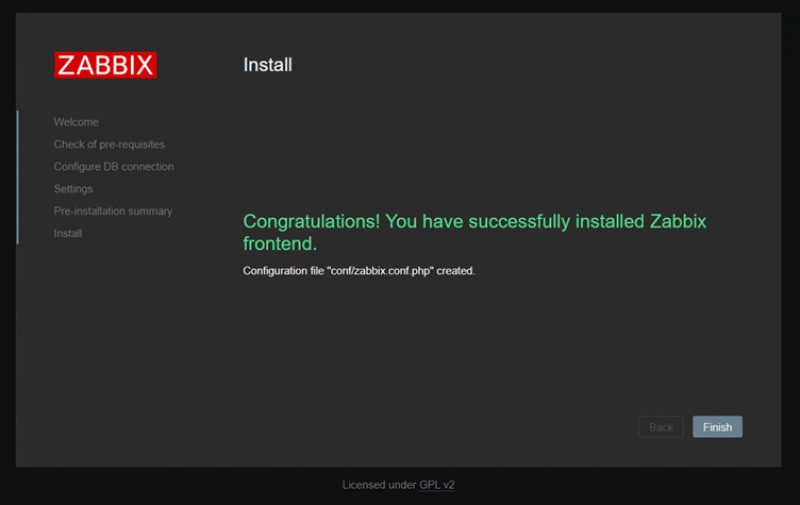
Nous choisissons les paramètres de Zabbix (en activant le thème sombre).



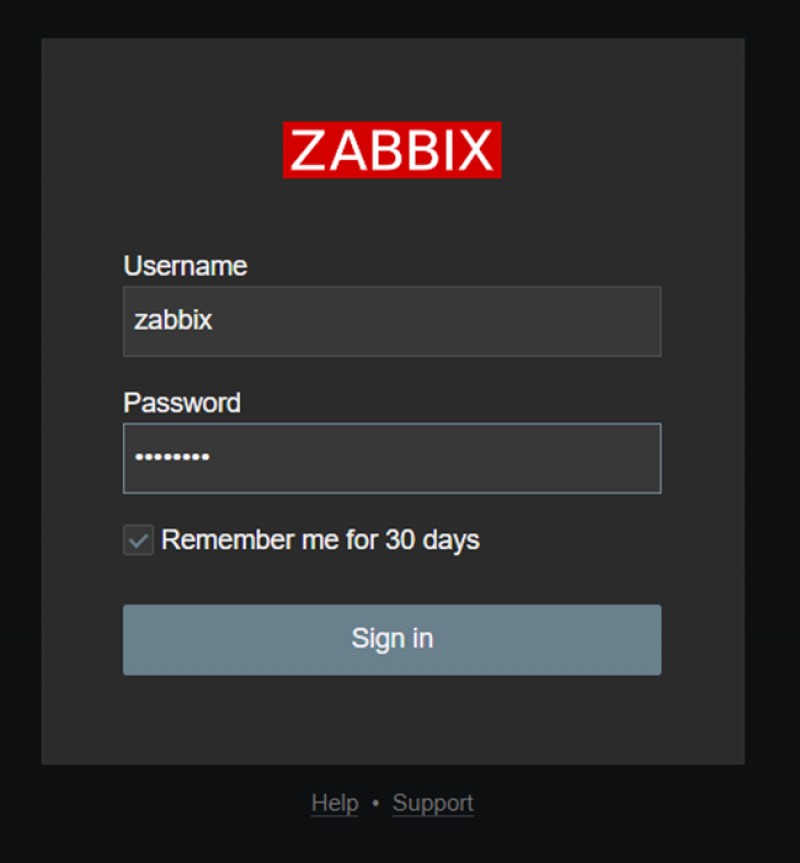
Nous vériﬁons que les éléments entrés soient corrects.



Nous pouvons terminer l'installation.



Nous pouvons maintenant nous connecter avec les identiﬁants par défaut : Admin/zabbix :

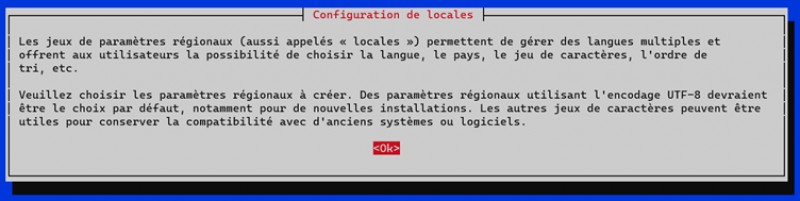


Nous pouvons voir sur notre tableau de bord que beaucoup d'erreurs empêchent le bon fonctionnement des graphes.

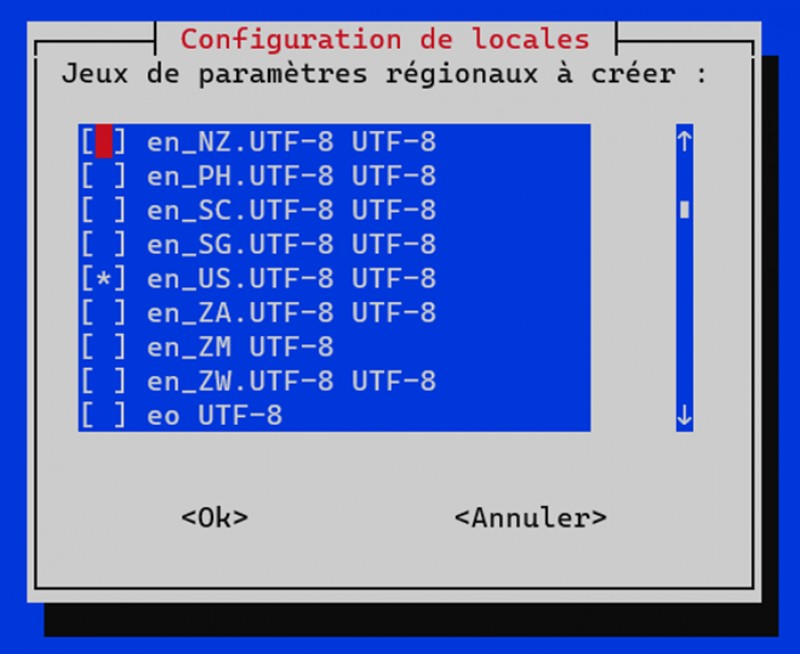


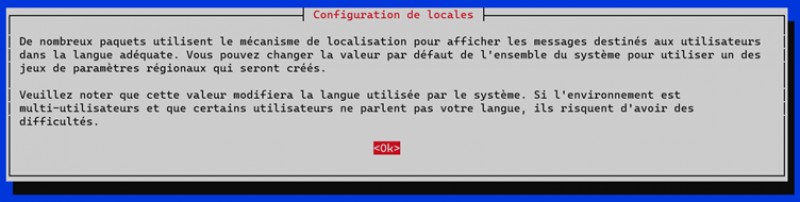
Pour ce faire, nous devons installer un nouveau paquet :

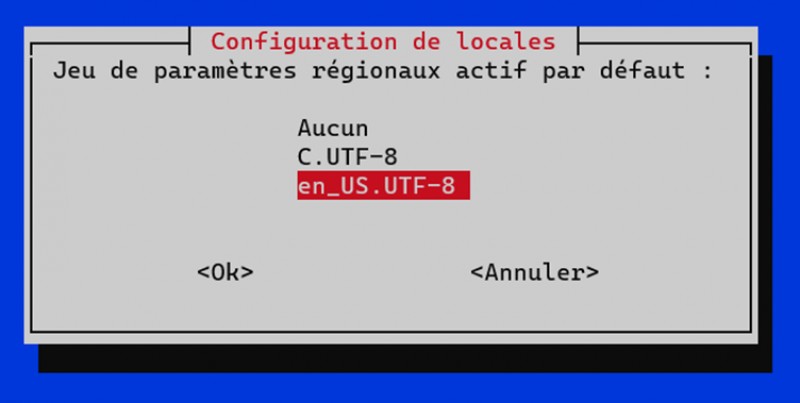
apt install locales



Nous choisissons les paramètres régionaux en\_US.UTF-8. Ce paramètre nous permettra d'enlever toutes les erreurs :

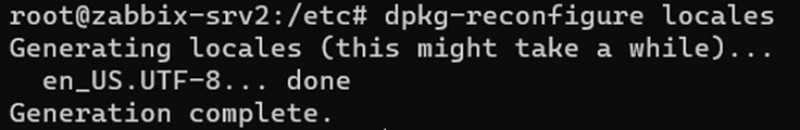






Nous entrons ensuite la commande suivante pour sauvegarder les changements :

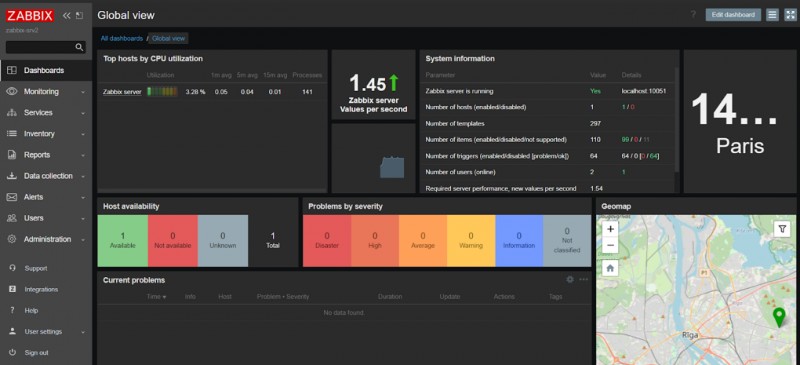
dpkg-reconfigure locales



Nous redémarrons le service zabbix\_server :

systemctl restart zabbix\_server

Nous pouvons nous rendre dès à présent sur notre tableau de bord.



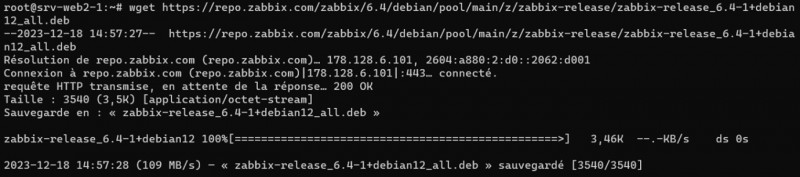
</panel> <panel type=“info” title=“III) Mise en place de la supervision”>

# Installation de l'agent sur la machine à monitorer

Nous pouvons maintenant installer le dépôt oﬃciel de Zabbix :

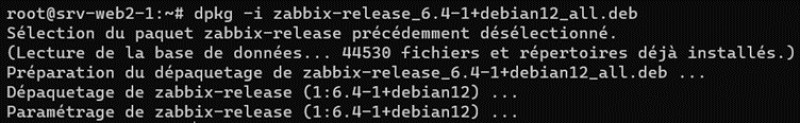
wget

https://repo.zabbix.com/zabbix/6.4/debian/pool/main/z/zabbix-release/zabbix- release\_6.4-1+debian12\_all.deb



Nous dépaquetons le dépôt oﬃciel de Zabbix :

dpkg -i zabbix-release\_6.4-1+debian12\_all.deb



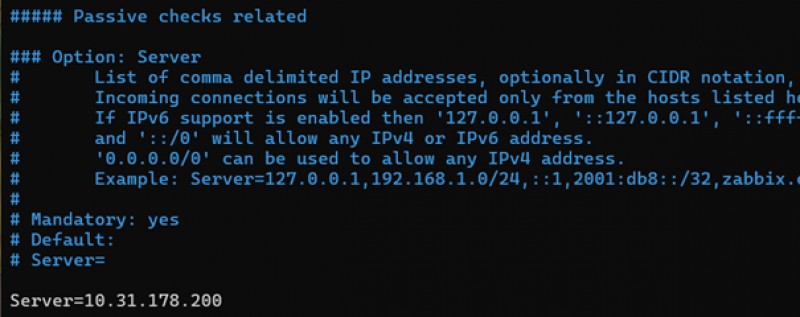
apt update && apt upgrade

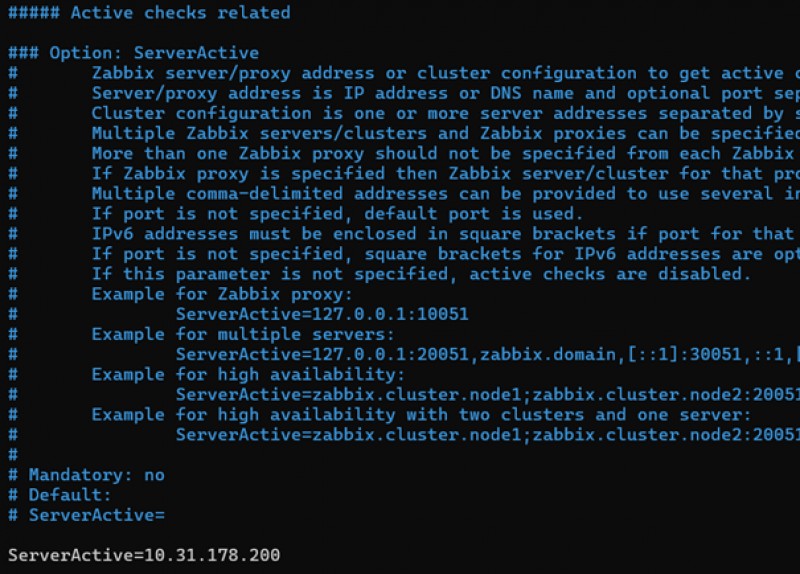
apt install zabbix-agent2 zabbix-agent2-plugin-\*

Nous modiﬁons le ﬁchier de conﬁguration zabbix\_agent2.conf. Nous entrons l'adresse IP de notre serveur Zabbix :

Server=10.31.178.200

ServerActive=10.31.178.200



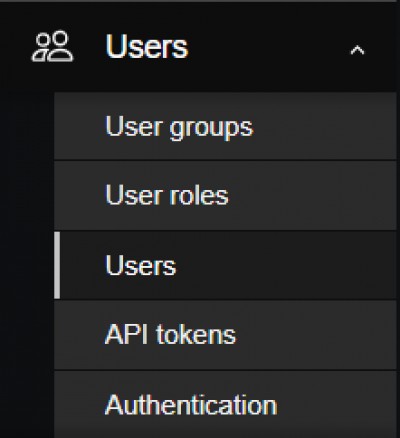


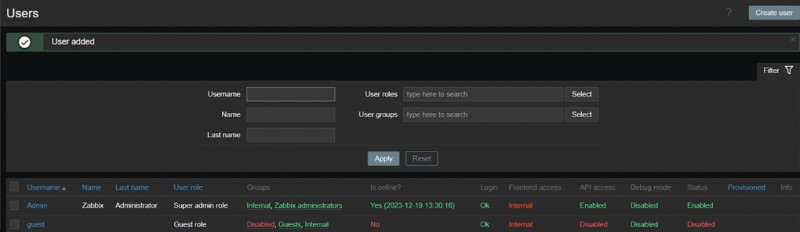
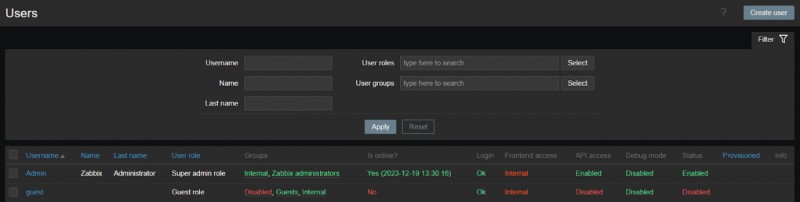
Nous pouvons redémarrer notre service :

systemctl restart zabbix\_agent2

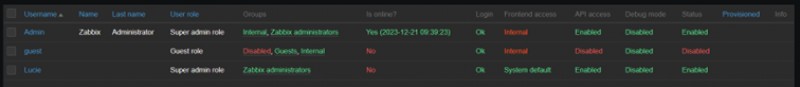
# Conﬁguration de Zabbix

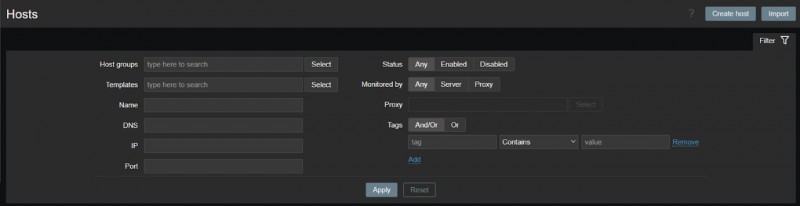
Nous pouvons maintenant créer un nouvel utilisateur. Pour cela, nous allons dans le menu des utilisateurs :



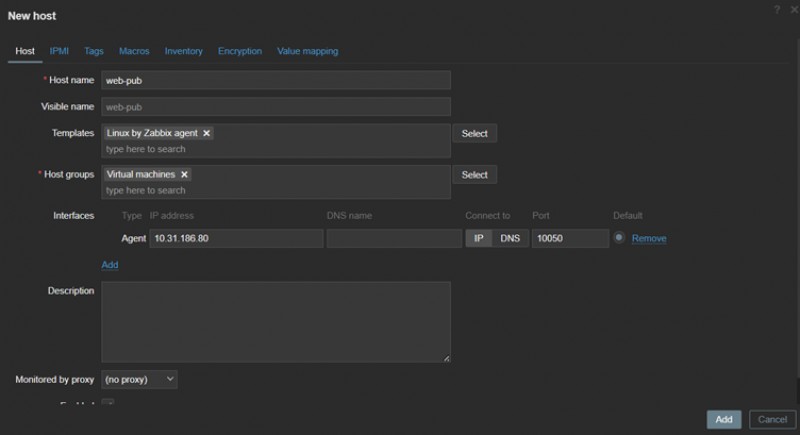


Nous créons un nouvel utilisateur qui aura pour rôle “Super admin role”. Nous cliquons donc sur le bouton Create User.

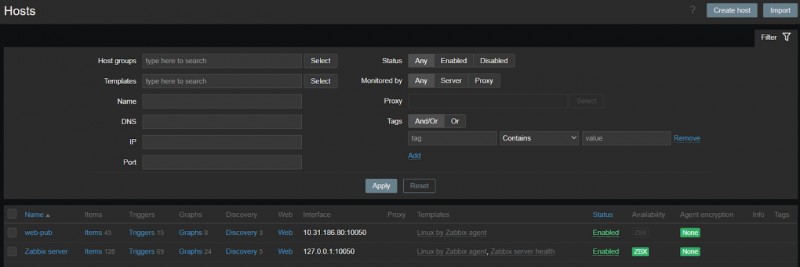




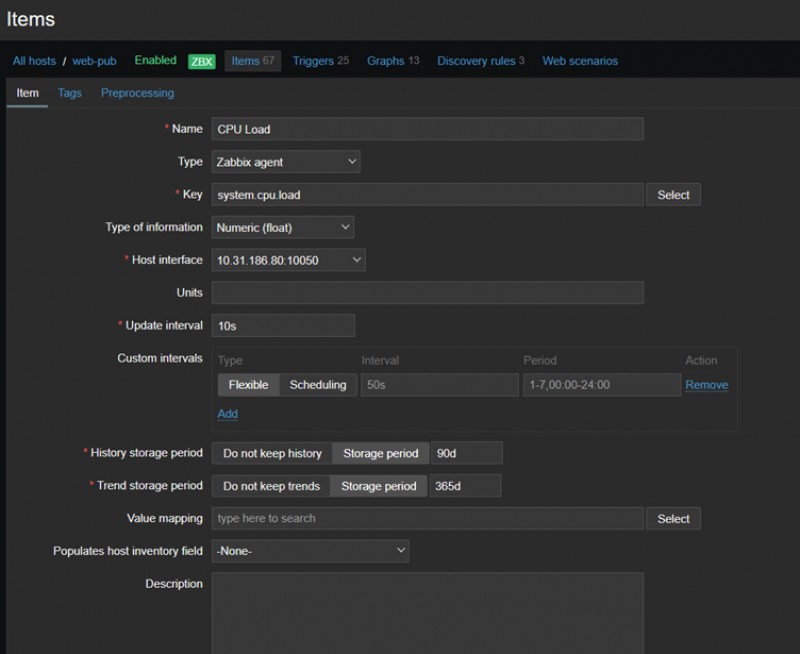
page des hôtes ^



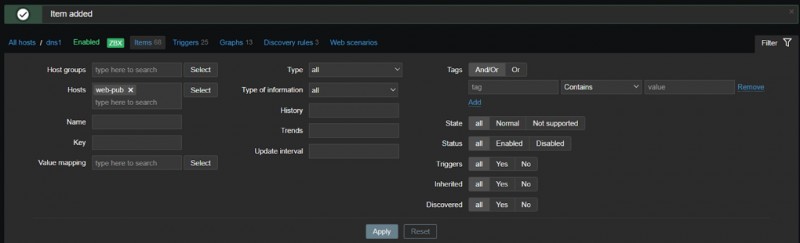
création d'un hôte



aﬃchage des hotes



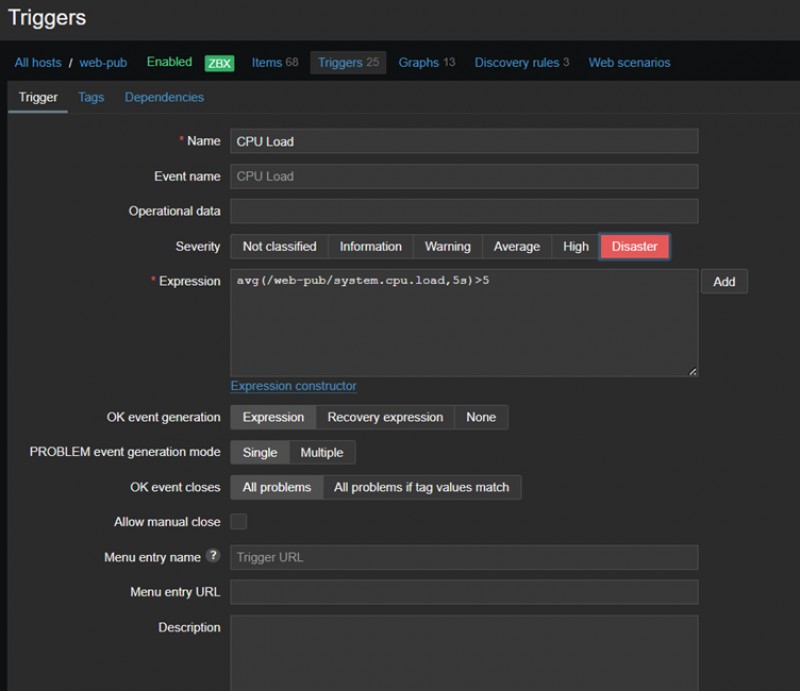
création d'un item pour l'alerting



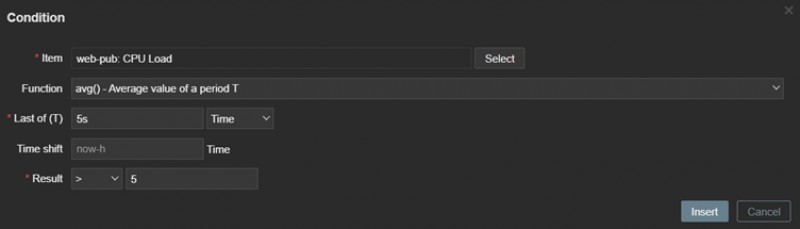
item ajouter



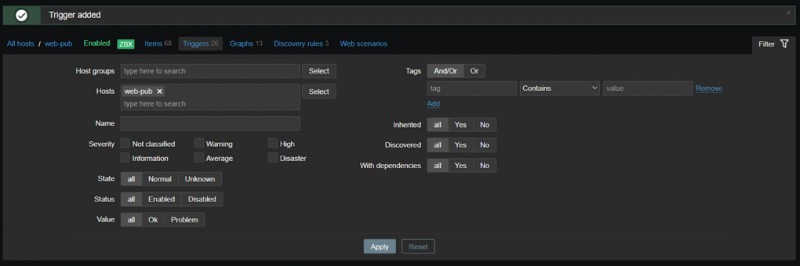
triggers pour les hote (alerte)



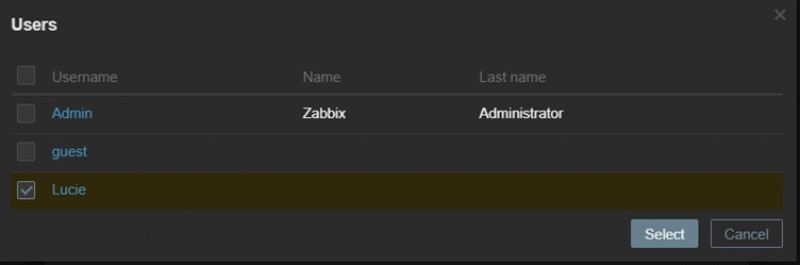
création d'un triggers pour un hote



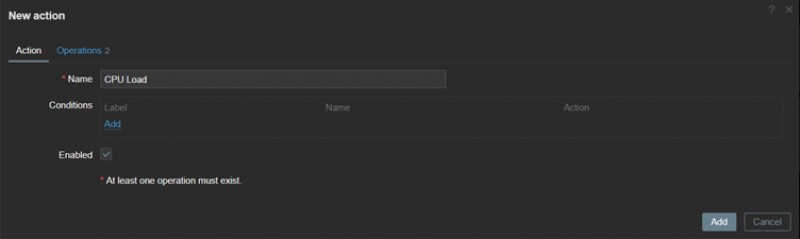
condition du trigger



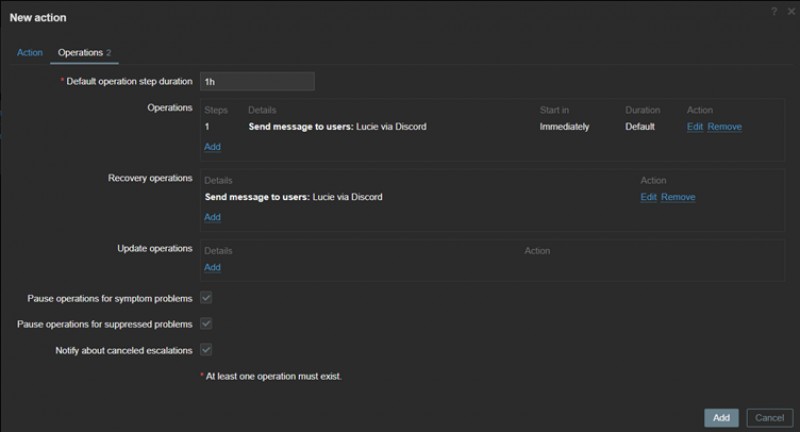
triggers ajouté



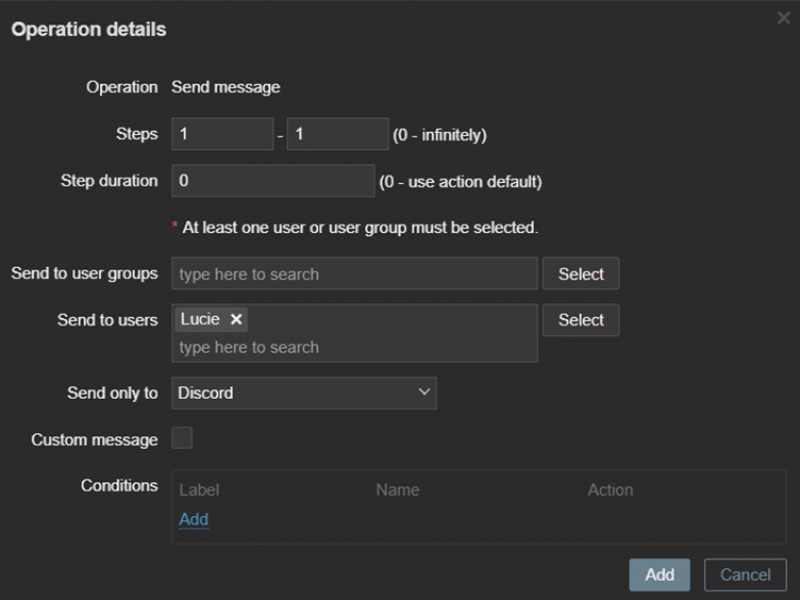
user



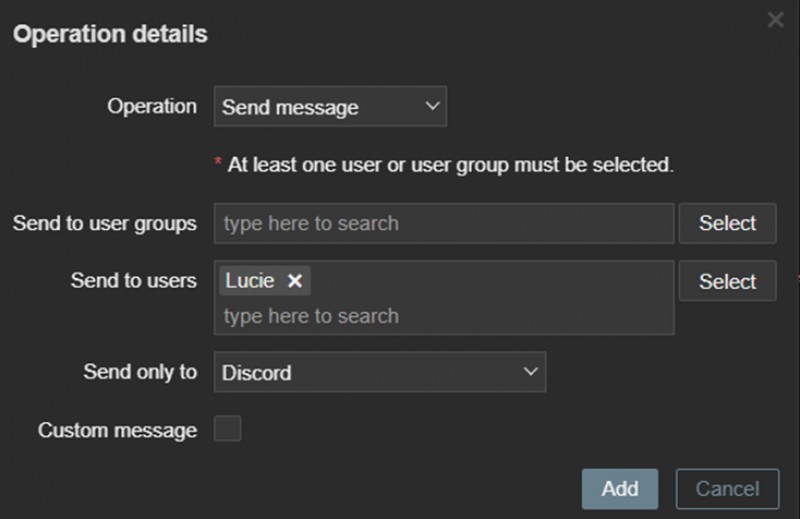
trigger action



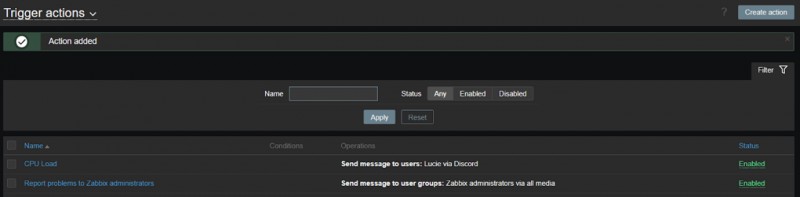
trigger action operation



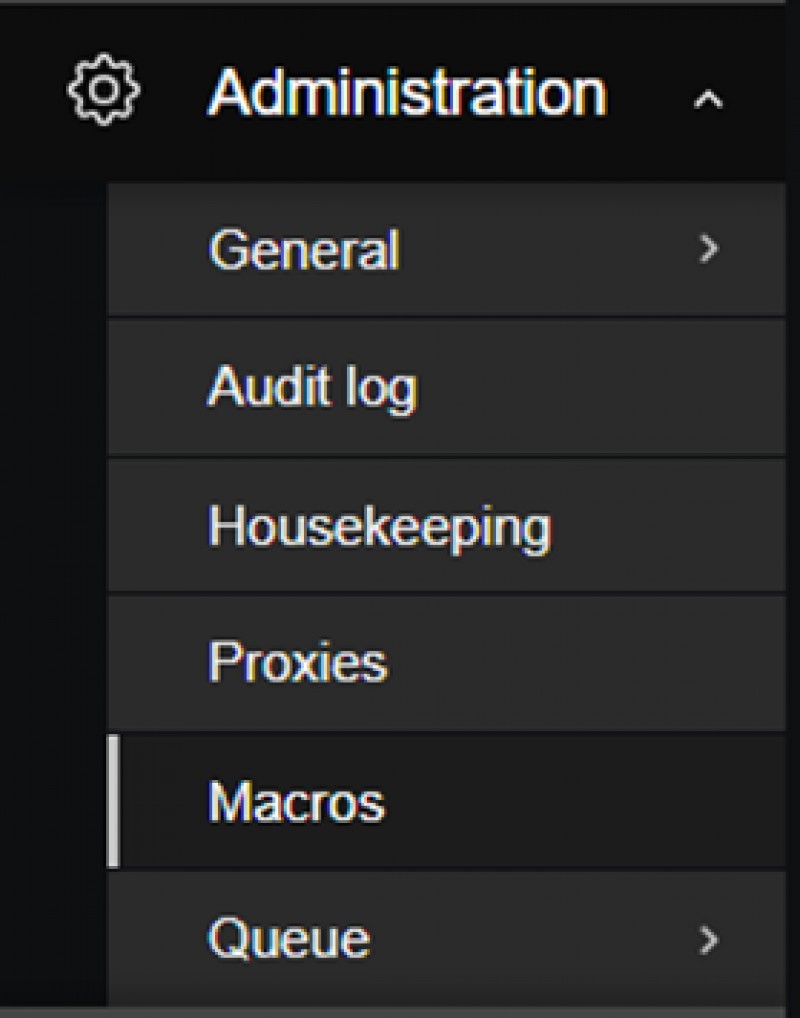
detail operation trigger



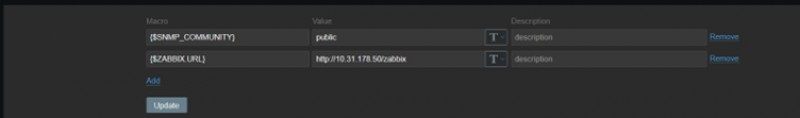
detail operation trigger suite



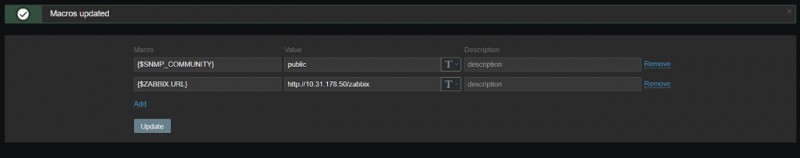
action ajouté



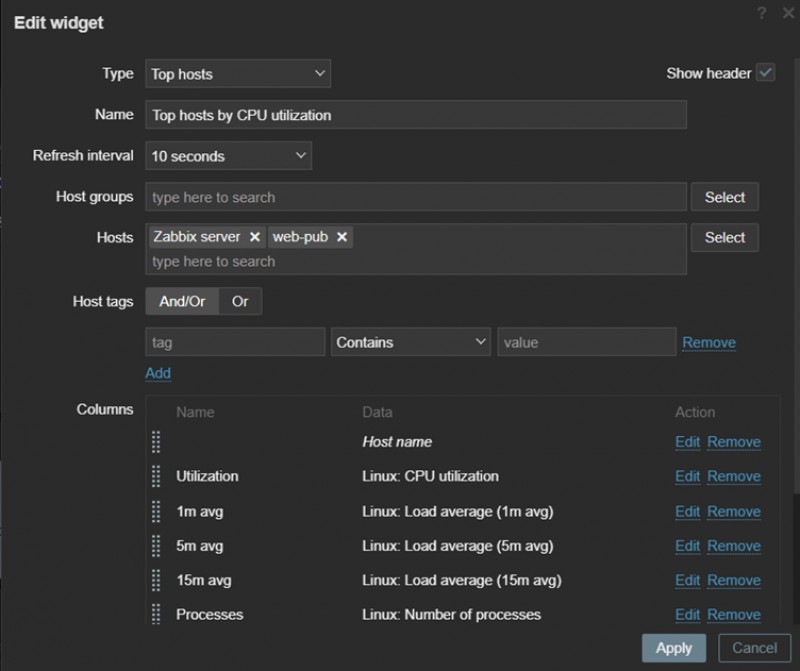
sidebar pour les macro



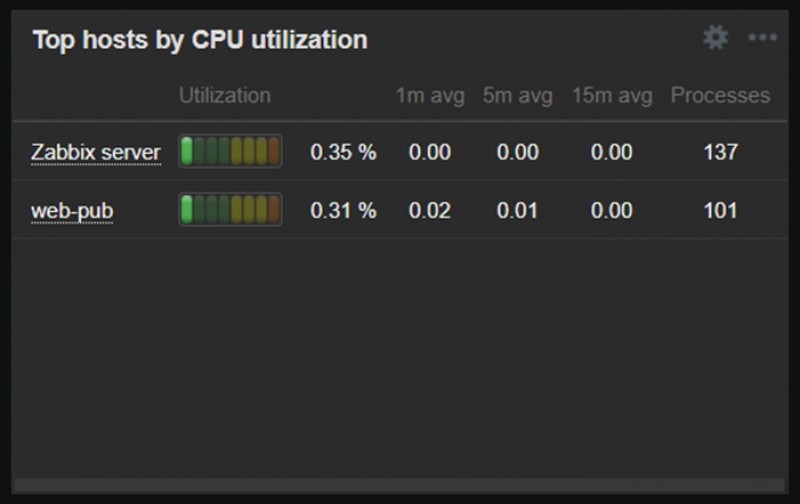
ajout des macro pour message discord



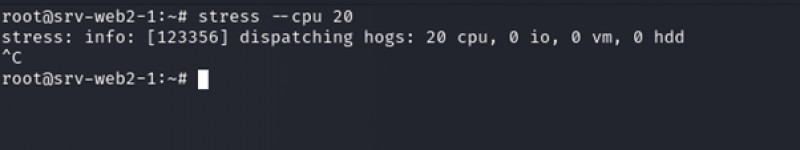
ajout des macro pour message discord



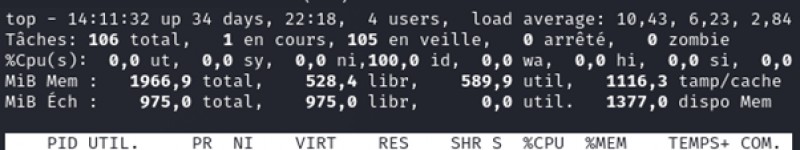
conﬁguration dashboard



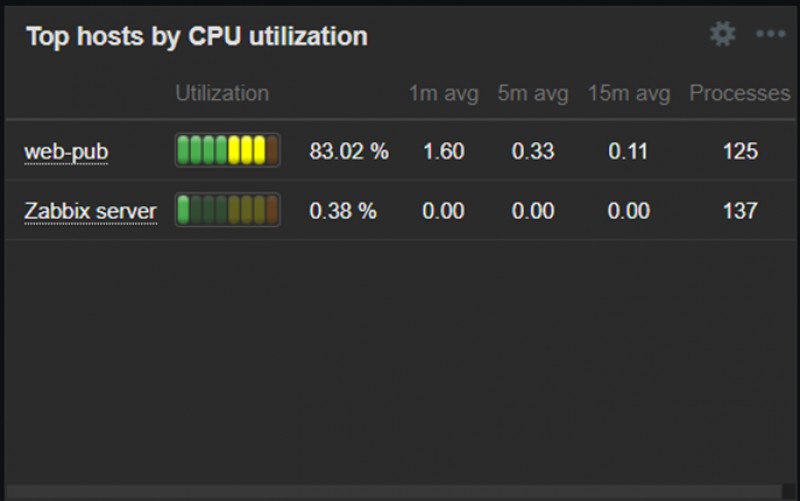
dashboard avant test



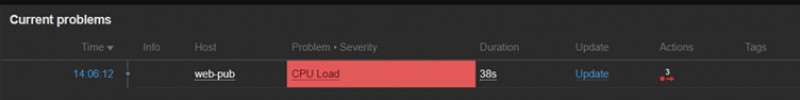
test stress cpu



stress cpu



dashboard pendant le test



alerte dans zabbix



dashboard après le test



From:

<https://sisr2.beaupeyrat.com/>- **Documentations SIO2 option SISR**

Permanent link:

[**https://sisr2.beaupeyrat.com/doku.php?id=sisr2-asie:sp2\_lucie**](https://sisr2.beaupeyrat.com/doku.php?id=sisr2-asie%3Asp2_lucie)

Last update: **2024/05/30 11:50**