

Compte rendu : stage Sopra steria

Mardi 09 Mai :

* Phase de recherche, reseignement sur le monitoring, qu’est que c’est zabbix ? que peut-on faire avec ? pour qui ? avantages ? inconvénients ? concurrences ?
* Chercher la distribution sur laquelle le déployer

Mercredi 10 Mai :

* Installation de la distrib (orientation debian) debut de codage sur un serveur test
* Choix d’une première version de zabbix
* Prise de connaissance du milieu de l’entreprise et des difficultés que l’on y rencontre (cas navimut pb avec le client qui n’arrive pas à intérroger la bdd oracle de l’équipe

Jeudi 11 Mai :

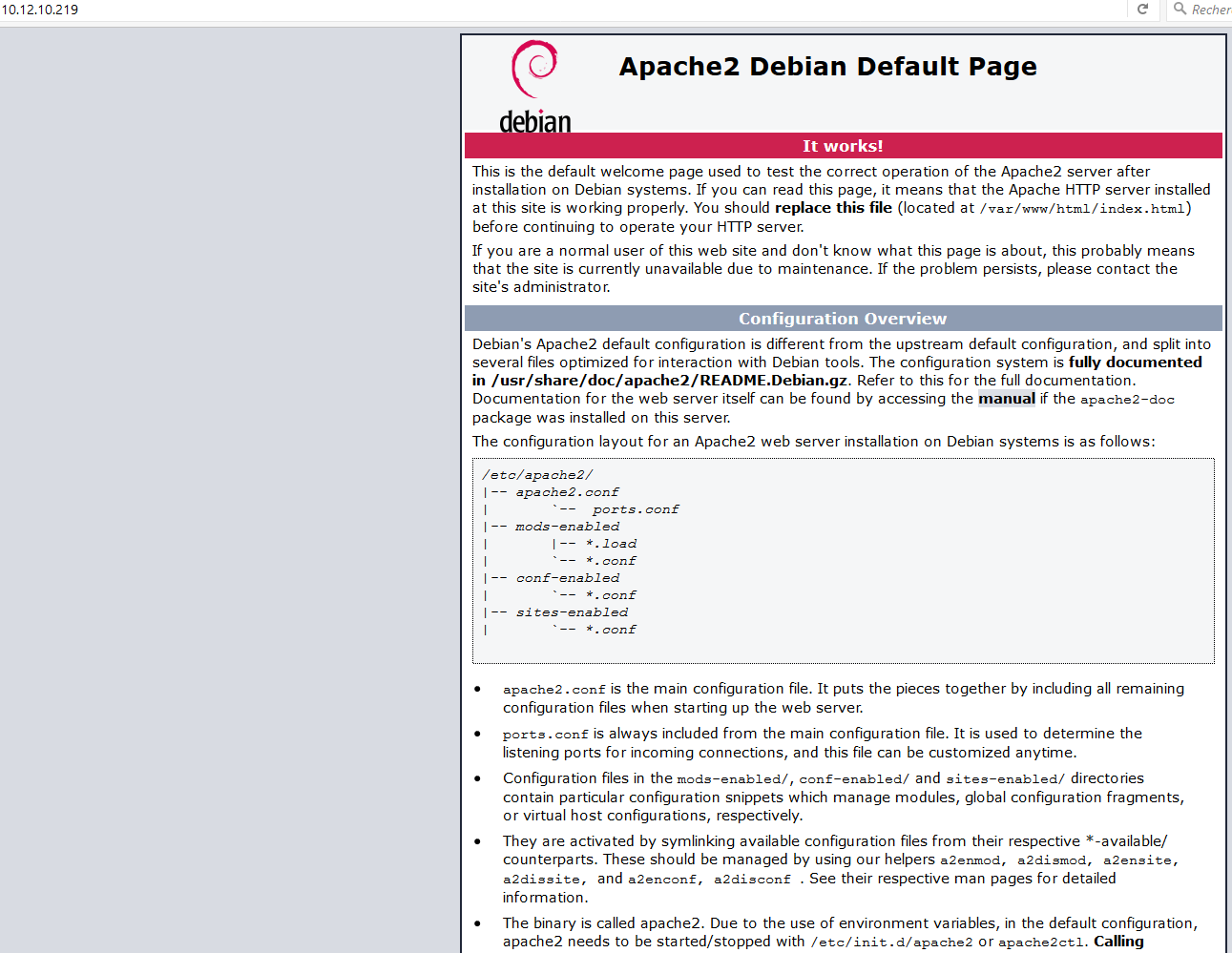
* Soucis avec le serveur de rebond, transfert du domaine de travail sous Oracle VB
* Realisation de la VM sous debian Jessie
* Deuxième selection d’une version de zabbix, cet fois çi plus récente (par la demande du client)

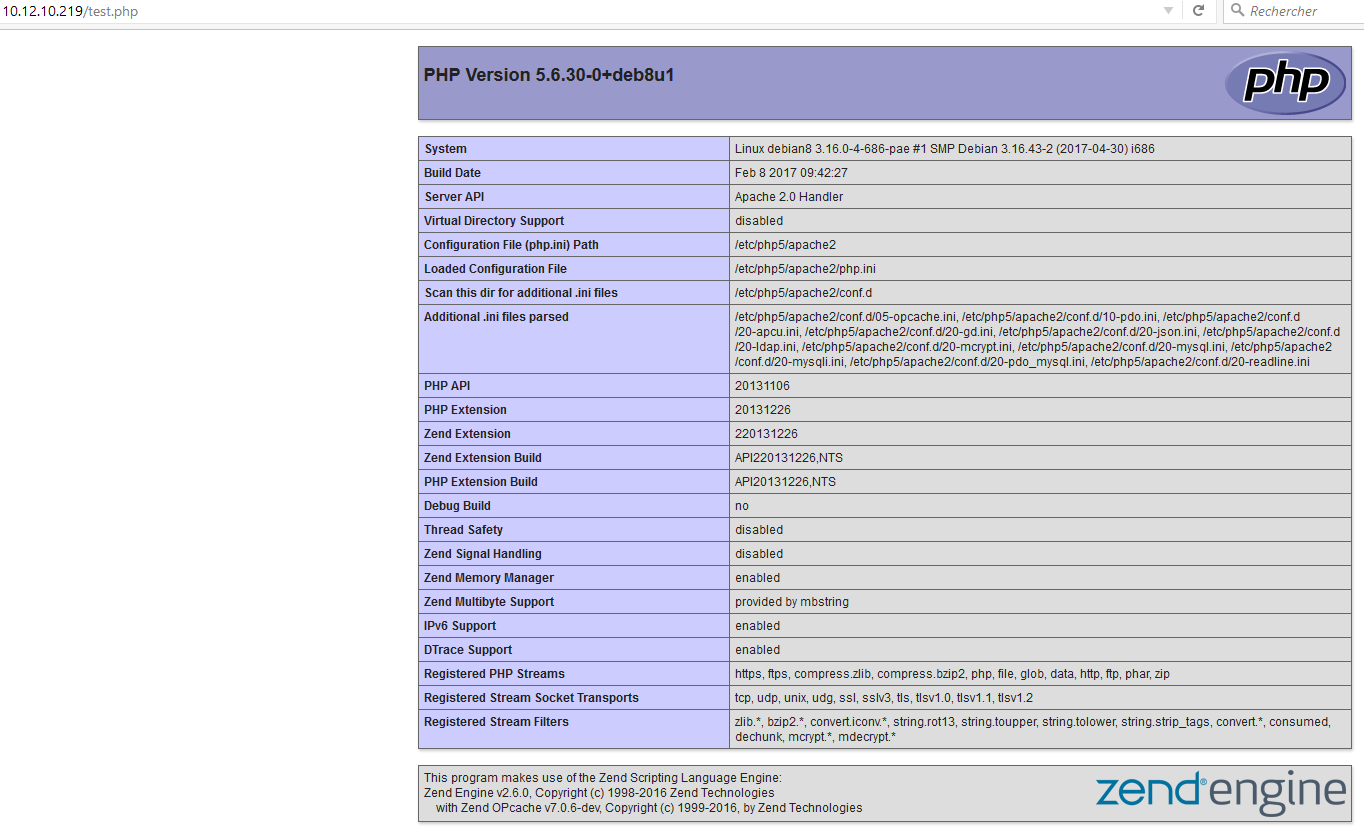
Vendredi 12 Mai :

* Apport des paquets nécéssaires à l’install de zabbix (zabbix-frontend-php, zabbix-server-mysql

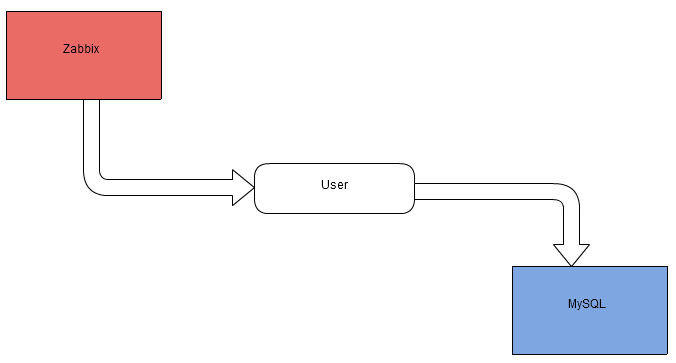
Wget (vers les repo.zabbix.com, apache2)

test d’apache et du php





* Install et config, création de la base de donnée zabbix, requète sql afin de donner les privilèges à un utilisateur requète sql pour identifier la base l’utilisateur et son mdp
* Le frontend, définition du fuseau horaire pour le php, edition du fichier conf et renseignement des id bdd ultilisateur mdp host



* Premier lancement de zabbix

Lundi 15 Mai :

* Première approche avec l’application
* Monitoring de la base de données de zabbix
* Rencontre de soucis avec les requètes apt-get et wget, impossible de dl l’agent zabbix

Mardi 16 Mai :

* « pareil »

Mercredi 17 Mai :

* « pareil »

Jeudi 18 Mai :

* « oh miracle ça fonctionne à nouveau, restriction sur le réseau soit disant dû au piratage informatique »
* Réalisation de la seconde VM

Vendredi 19 Mai :

* Ajout de nouveau diagramme sous zabbix

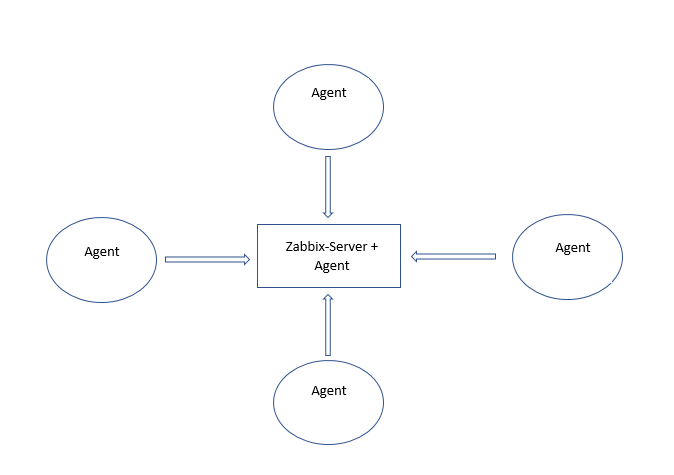
Lundi 22 Mai :

* Supervision mysql
* CHANGEMENT DE METHODE (une VM avec le serveur zabbix l’autre avec l’agent, recherche sur la méthode de travail de l’agent passive and active checks)

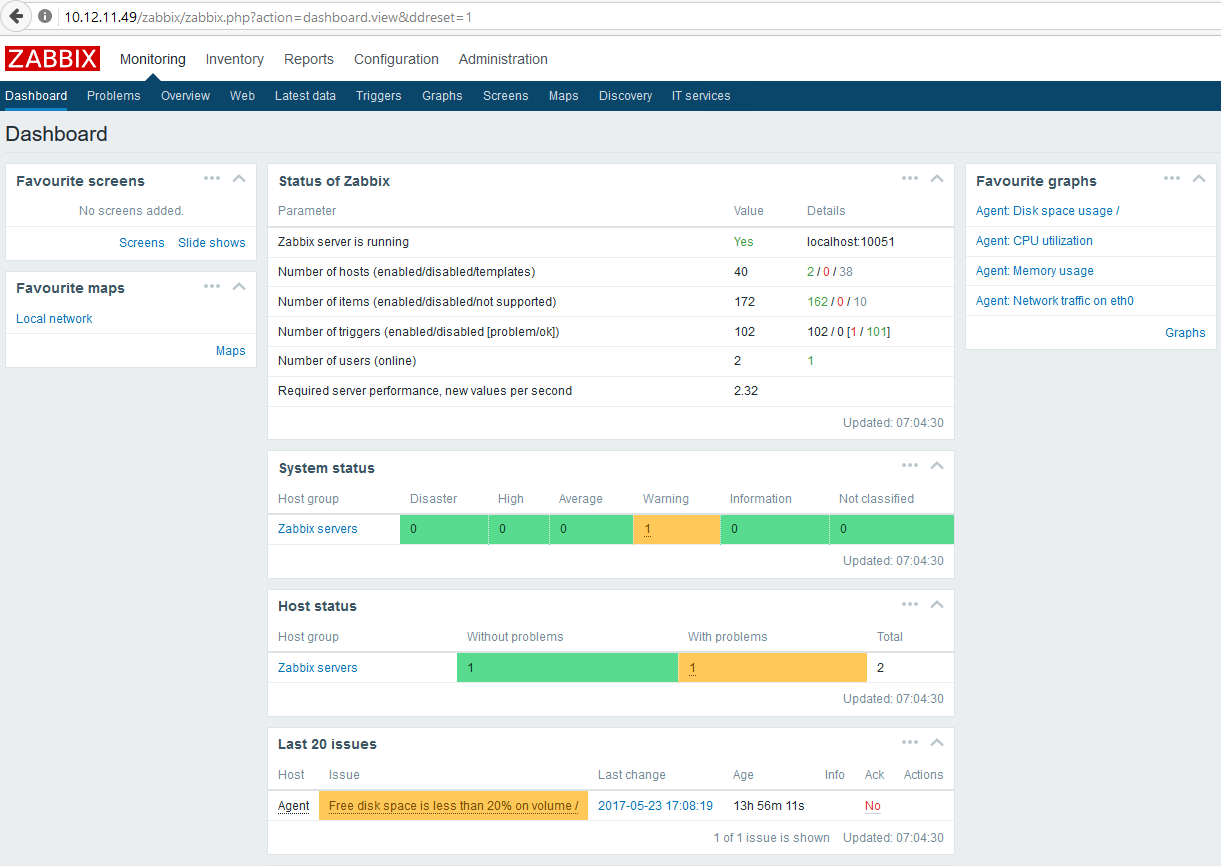
Mardi 23 Mai :

* Config des VM changement du type de réseau accès par pont réalisation de test de communication entre les VM, les deux se ping mutuellement. Config de l’agent
* Il est possible qu’un serveur zabbix est besoin d’un agent interne pour pourvoir se monitorer
* Du coup, sur le la partie contenant le serveur zabbix, ajouter un agent afin que le serveur puisse se monitorer lui-même.

Des agents doivent graviter autour du serv afin de récupérer les différentes données du serv.



Bingo ça marche :

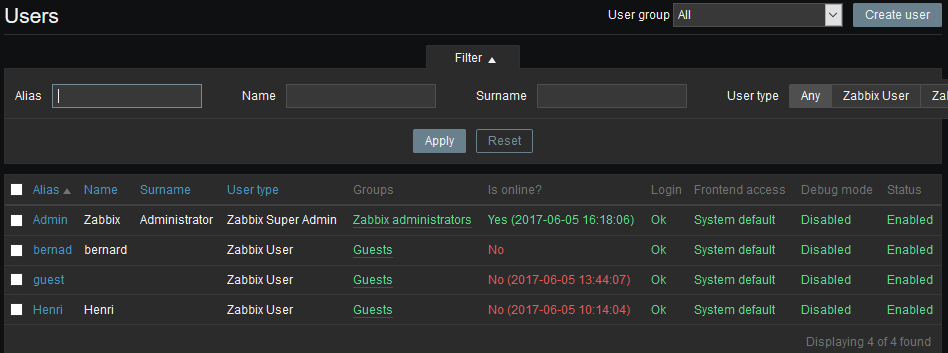


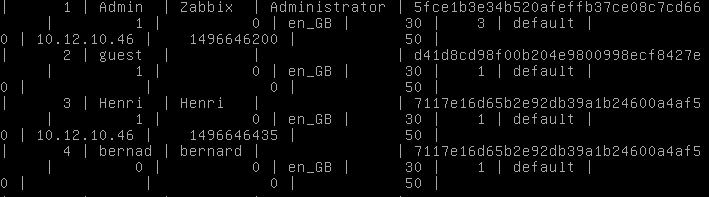
Mercredi 24 Mai :

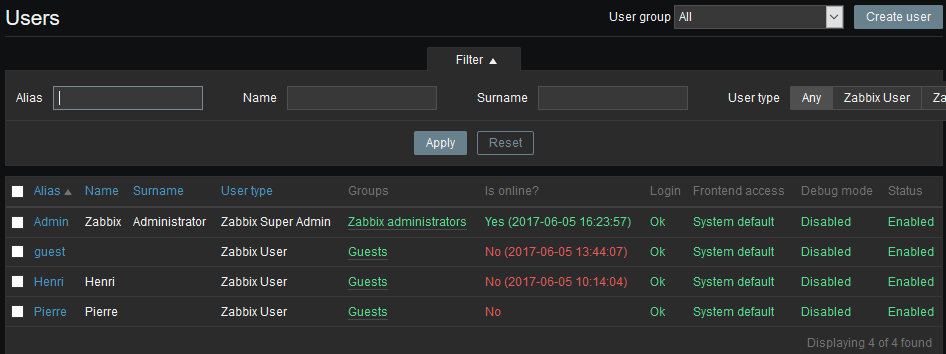
Au Mercredi 31 Mai, réalisation de la doc, ajout de contenu à la mission, réalisation d’un serveur lamp, installation et config d’un phpmyadmin pour phase de test avec l’agent.

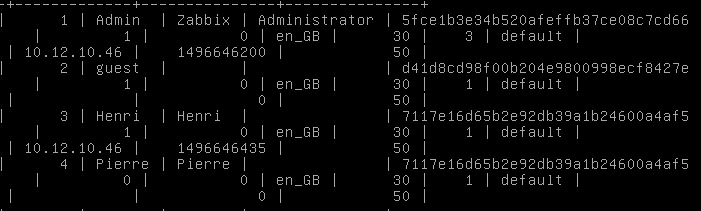
Jeudi , vendredi 01 02 juin :

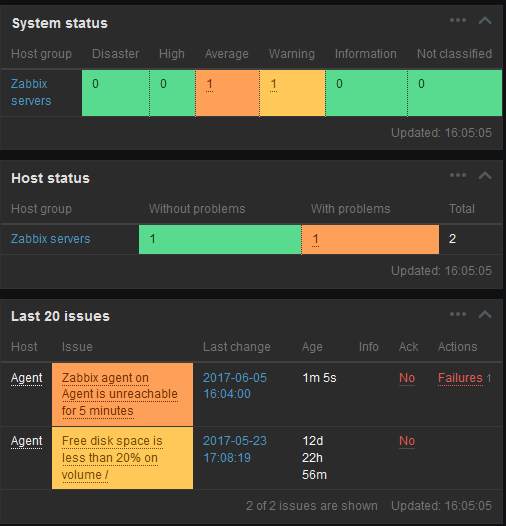
Le user réalisé au tout début n’est autre qu’un user servant de passerelle à l’application zabbix afin d’accéder au mysql, ainsi depuis zabbix il est possible de modifier des tables dans la base de donnée



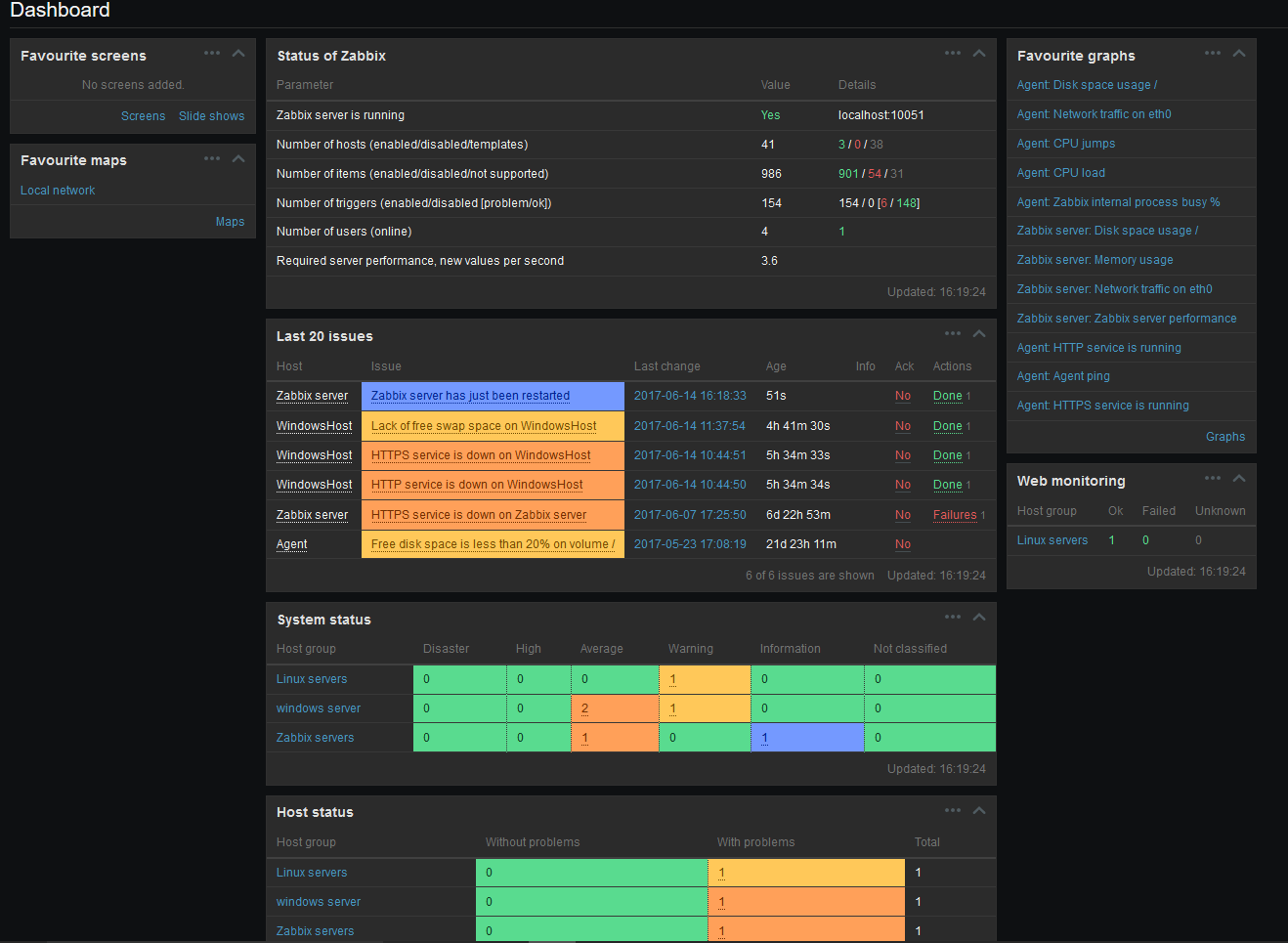








Semaine 12 juin au 16 juin



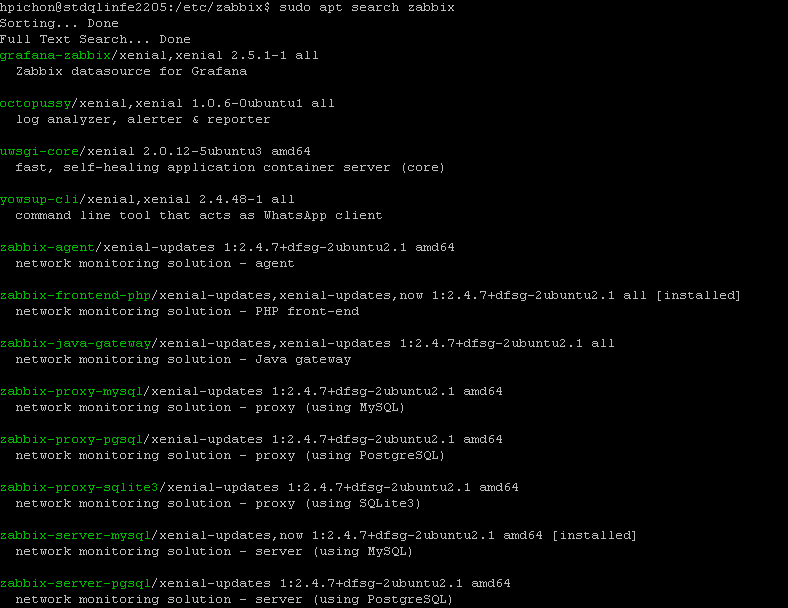
Ajout de contenu, web monitoring,

Supervision d’un windows

Réalisation sur Serveur :

Dû au restriction il nous est impossible d’accéder au repo afin d’obtenir la version de zabbix qui nous interesse

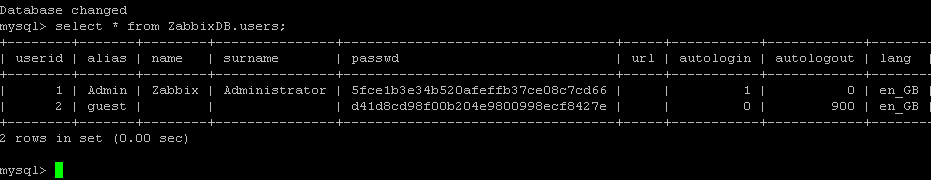
Apt search zabbix



Nous utiliserons donc la version 2.4.7

Install du zabbix server et du frontend et des différents modules nécéssaires

Attribution d’un password à root, création d’une base et d’un user



Apport des fichiers .sql images datas schemas à la base

Config du fichier du zabbix-server

Nous y indiquerons :

dbname

dbuser

dbpassword

config du php.ini

max execution time

max input time

memory limit :

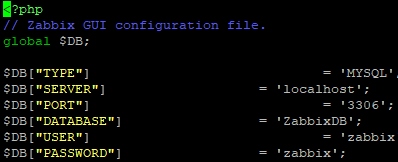
upload max filesize : 16M

Créer le fichier zabbix.php.conf

Sudo nano /etc/zabbix/Zabbix.php.conf

Via il est possible que installeur souhaite avoir les droits d’accès à ce fichier afin d’y écrire cd dont il a besoin

Attribuer les lui.





Fonctionnement des templates : ces templates possèdent des applications qui eux-mêmes possèdent des items possèdent des keys, qui sont définis dans le fichier de conf userparameter\_ « l’application ».conf

Exemple : userparameter\_mysql.conf

Grace à ça nous recevons bien plus que de simple information de présence d’un service avec net.tcp.service.

nous remontons jusqu’à la couche applicatif du model osi afin d’avoir des informations de la matrice elle même, là où net.tcp.service s’arrète au réseau, nous permettant tout de même de savoir si celui-ci est présent