

**BTS Services informatiques aux organisations - SISR****Session 2022****E4 – Support et mise à disposition de services informatiques****Coefficient 4****DESCRIPTION DE LA REALISATION PROFESSIONNELLE****NOM et prénom du candidat : LEDUC Quentin**

N° candidat : 02145639104

**Contexte de la réalisation professionnelle**

Nous avons déjà une supervision Zabbix chez Scani, mais pour plus de transparence avec nos coopérateurs, nous aimerions avoir ces données en public.

**Intitulé de la réalisation professionnelle**

Installation de Zabbix avec Cachet

**Période de réalisation : DU 19/08/21 AU 23/08/21****Lieu : Joigny****Modalité : Individuelle****Principale(s) activité(s) concernée(s) :**

- Recenser et identifier les ressources numériques
- Vérifier les conditions de la continuité d'un service informatique
- Participer à l'évolution d'un site Web exploitant les données de l'organisation.
- Déployer un service
- Réaliser les tests d'intégration et d'acceptation d'un service
- Mettre en place son environnement d'apprentissage personnel
- Développer son projet professionnel

**Conditions de réalisation**

- **Ressources présentes** Nous avons une supervision Zabbix fonctionnelle
- **Résultats attendus** Avoir ces données en publique
- **Durée de réalisation** 2H

**Modalités d'accès à cette réalisation professionnelle.**Site internet : [www.netwaze.fr](http://www.netwaze.fr)

Aller dans « Réalisations Professionnelles » Mot de passe : Mr.Robot

**Partie 1 – Procédure de mise en œuvre.****PRÉREQUIS MATÉRIEL**

- Un serveur Ubuntu - Debian
- Zabbix
- Cachet

## PRÉREQUIS

Tout d'abord nous mettons à jour notre distribution Linux Ubuntu.

```
$ apt update -y && apt upgrade -y
```

puis on installe les dépôts de Zabbix.

```
$ wget https://repo.zabbix.com/zabbix/5.0/ubuntu-arm64/pool/main/z/zabbix-release/zabbix-release_5.0-1+ubuntu20.04_all.deb
```

On installe le paquet

```
$ dpkg -i zabbix-release_5.0-1+ubuntu20.04_all.deb
```

Puis on met à jour notre distribution

```
$ apt update
```

## INSTALLATION DE ZABBIX

Pour installer Zabbix il suffit de taper la commande : \$ apt install zabbix-server-mysql zabbix-frontend-php zabbix-apache-conf zabbix-agent

## CRÉATION DE LA BASE DE DONNÉES

Zabbix fonctionne avec Mysql / Mariadb et Postgresql. Nous nous allons partir sur une MariadDB.

Installation de MariaDB : \$ apt install mariadb-server

Une fois l'installation fini on configure MariaDB et on créer notre base de données.

```
$ mysql_secure_installation
```

On créer la base de données

```
$ mysql -u root -p
```

```
$ create database zabbix character set utf8 collate utf8_bin;
```

```
$ create user zabbix@localhost identified by 'password';
```

```
$ grant all privileges on zabbix.* to zabbix@localhost;
```

```
$ quit;
```

On apporte le schéma de la base de données (l'infrastructure) de Zabbix dans la base de données MariaDB.

```
$ zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz | mysql -uzabbix -p zabbix
```

```
root@ServeurAdministration:/# zcat /usr/share/doc/zabbix-server-mysql*/create.sql.gz | mysql -uzabbix -p zabbix
Enter password:
root@ServeurAdministration:/#
```

## CONFIGURATION DE ZABBIX

Maintenant nous devons configurer Zabbix dans ces fichiers de configuration.

On ouvre le fichier `/etc/zabbix/zabbix_server.conf` avec nano ou vi et on modifie la ligne 124 `DBPassword=` pour y mettre notre mot de passe de notre base de données.

On ouvre aussi le fichier `/etc/zabbix/apache.conf` pour y configurer notre fuseau horaire. On décommente la ligne 20 et a coté de `Europe/Paris`.

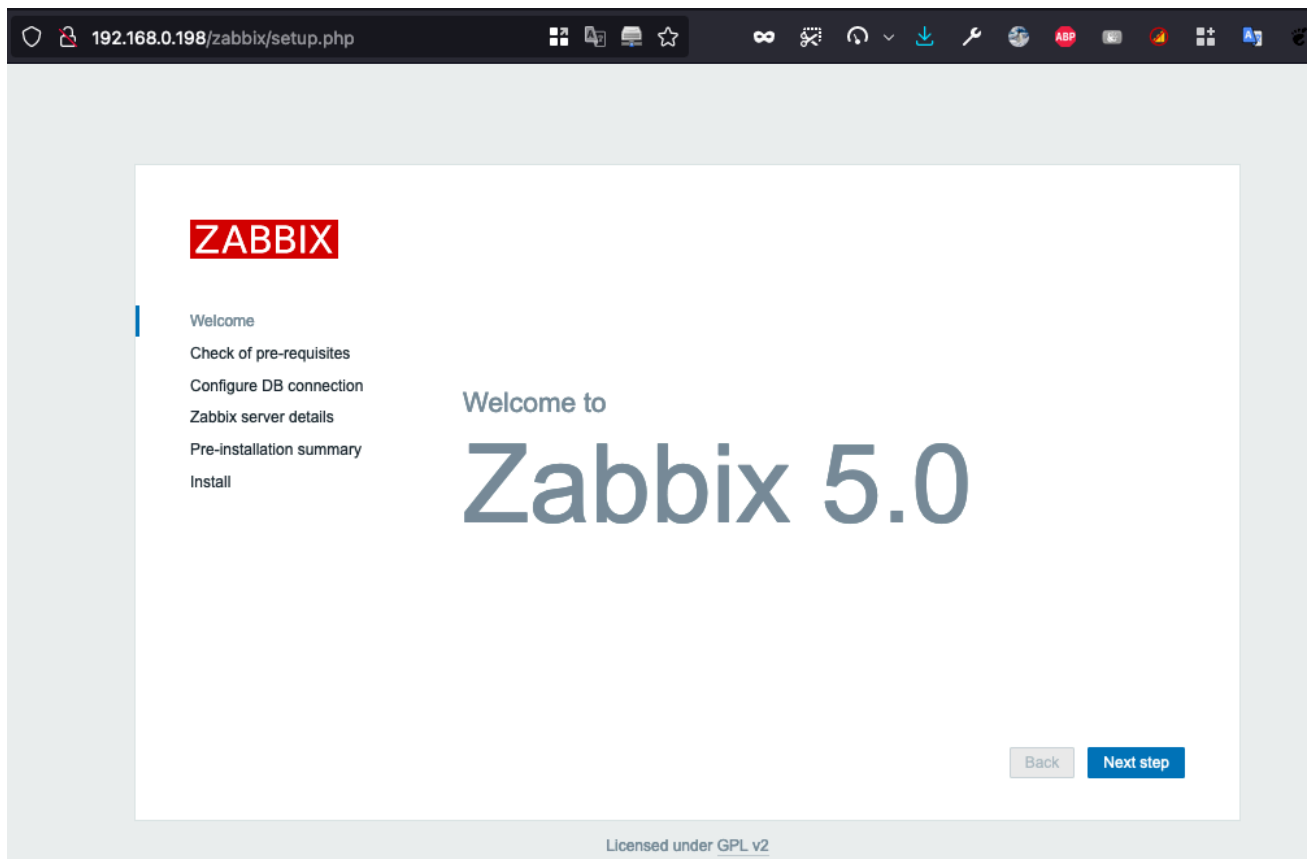
On redémarre zabbix et on l'active au démarrage.

```
$ systemctl restart zabbix-server zabbix-agent apache2
```

```
$ systemctl enable zabbix-server zabbix-agent apache2
```

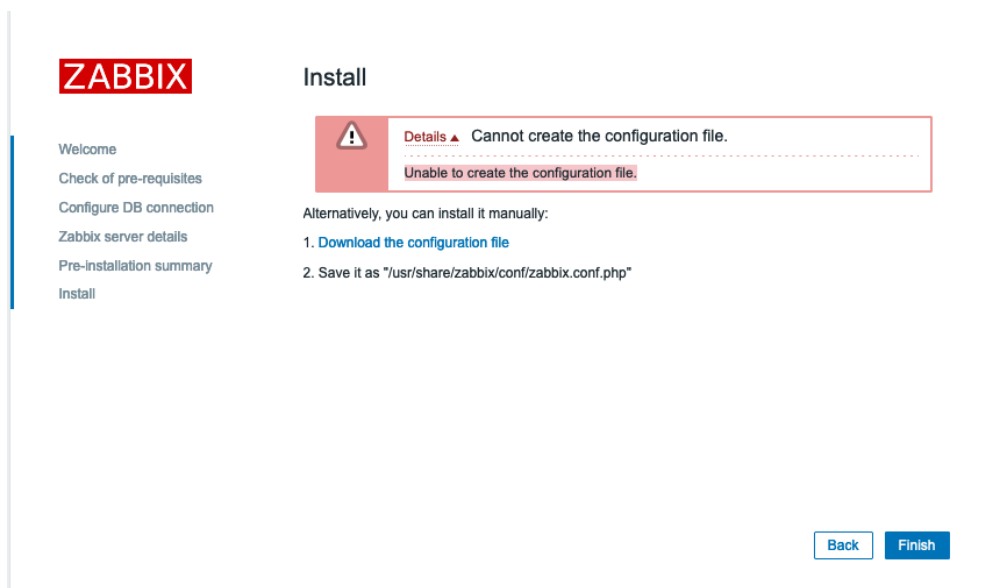
```
root@ServeurAdministration:/# systemctl restart zabbix-server zabbix-agent apache2
root@ServeurAdministration:/# systemctl enable zabbix-server zabbix-agent apache2
Synchronizing state of zabbix-server.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable zabbix-server
Synchronizing state of zabbix-agent.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable zabbix-agent
Synchronizing state of apache2.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable apache2
Created symlink /etc/systemd/system/multi-user.target.wants/zabbix-server.service → /lib/systemd/system/zabbix-server.service.
root@ServeurAdministration:/# systemctl status zabbix-server zabbix-agent apache2
● zabbix-server.service - Zabbix Server
   Loaded: loaded (/lib/systemd/system/zabbix-server.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2021-08-23 11:02:59 UTC; 1min 1s ago
 Main PID: 6853 (zabbix_server)
    Tasks: 38 (limit: 4435)
   CGroup: /system.slice/zabbix-server.service
           └─6853 /usr/sbin/zabbix_server -c /etc/zabbix/zabbix_server.conf
             └─6856 /usr/sbin/zabbix_server: configuration syncer [synced configuration in 0.092776 sec, idle 5.0>
               └─6858 /usr/sbin/zabbix_server: alert manager #1 [sent 0, failed 0 alerts, idle 5.006645 sec duri>
                 └─6859 /usr/sbin/zabbix_server: alerter #1 started
                   └─6860 /usr/sbin/zabbix_server: alerter #2 started
                     └─6861 /usr/sbin/zabbix_server: alerter #3 started
                       └─6862 /usr/sbin/zabbix_server: preprocessing manager #1 [queued 1, processed 12 values, idle 5.0>
                         └─6863 /usr/sbin/zabbix_server: preprocessing worker #1 started
                           └─6864 /usr/sbin/zabbix_server: preprocessing worker #2 started
                             └─6865 /usr/sbin/zabbix_server: preprocessing worker #3 started
                               └─6866 /usr/sbin/zabbix_server: lld manager #1 [processed 0 LLD rules, idle 5.024066sec during 5.0>
                                 └─6867 /usr/sbin/zabbix_server: lld worker #1 started
                                   └─6868 /usr/sbin/zabbix_server: lld worker #2 started
                                     └─6869 /usr/sbin/zabbix_server: housekeeper [startup idle for 30 minutes]
                                       └─6870 /usr/sbin/zabbix_server: timer #1 [updated 0 hosts, suppressed 0 events in 0.001123 sec, i>
                                         └─6871 /usr/sbin/zabbix_server: http poller #1 [got 0 values in 0.001550 sec, idle 5 sec]
                                           └─6872 /usr/sbin/zabbix_server: discoverer #1 [processed 0 rules in 0.001071 sec, idle 60 sec]
                                             └─6874 /usr/sbin/zabbix_server: history syncer #1 [processed 0 values, 0 triggers in 0.000154 sec>
                                               └─6878 /usr/sbin/zabbix_server: history syncer #2 [processed 0 values, 0 triggers in 0.000172 sec>
                                                 └─6879 /usr/sbin/zabbix_server: history syncer #3 [processed 0 values, 0 triggers in 0.000160 sec>
                                                   └─6882 /usr/sbin/zabbix_server: history syncer #4 [processed 0 values, 0 triggers in 0.000120 sec>
                                                     └─6883 /usr/sbin/zabbix_server: escalator #1 [processed 0 escalations in 0.002689 sec, idle 3 sec]
root@ServeurAdministration:/#
```

Nous pouvons maintenant ouvrir notre navigateur internet et taper dans notre url notre adresse ip ainsi que zabbix.



Page d'initialisation de Zabbix

Nous configurons donc notre Zabbix depuis l'interface Web. Rien de très compliqué. Puis nous arrivons sur cette erreur.



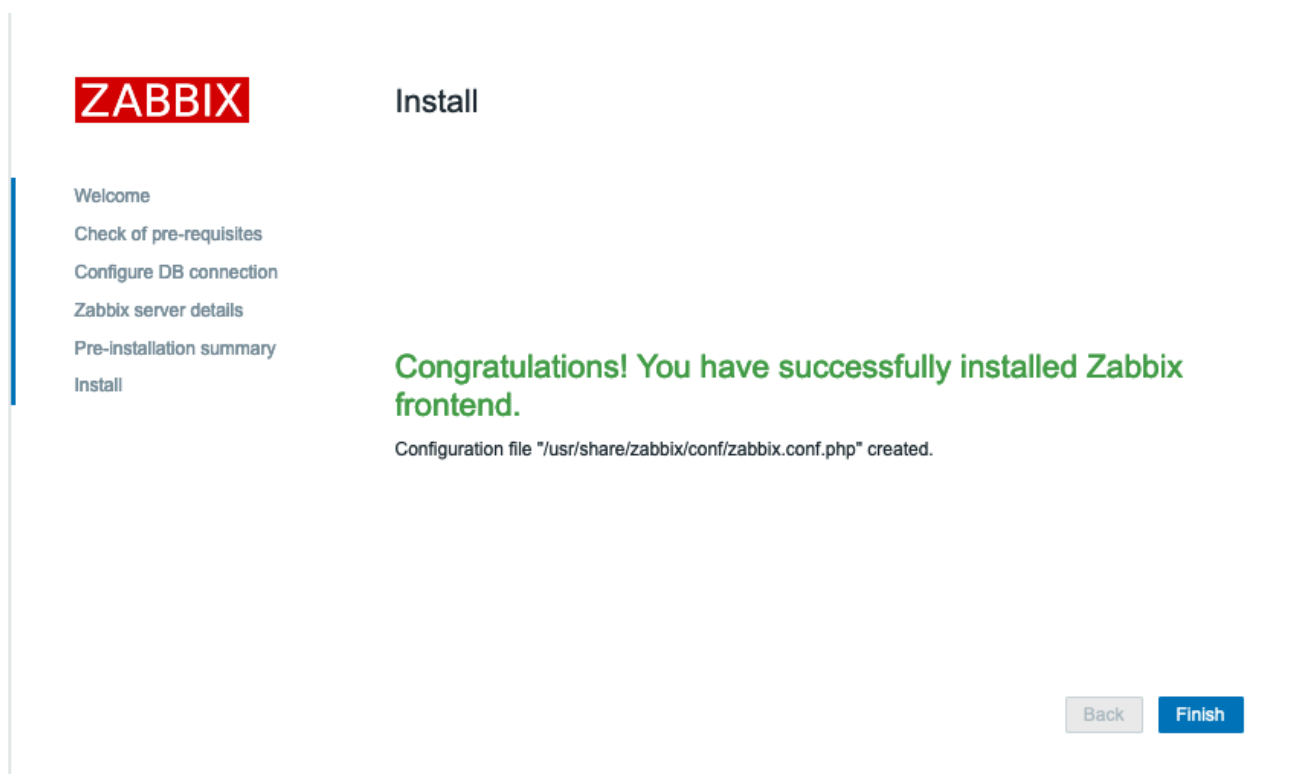
On telecharge notre fichier de configuration puis nous le plaçons dans le répertoire /home/ubuntu avec SCP.

```
kiu@MacBook-Pro-de-kiu Downloads % scp zabbix.conf.php ubuntu@192.168.0.198:/home/ubuntu
ubuntu@192.168.0.198's password:
zabbix.conf.php                                100% 1504   338.5KB/s   00:00
kiu@MacBook-Pro-de-kiu Downloads % █
```

Maintenant nous le mettons dans notre dossier Zabbix /usr/share/zabbix/conf/ en root sur notre machine.

```
root@ServeurAdministration:/home/ubuntu# ls
zabbix-release_5.0-1+ubuntu20.04_all.deb  zabbix.conf.php
root@ServeurAdministration:/home/ubuntu# mv zabbix.conf.php /usr/share/zabbix/conf/
root@ServeurAdministration:/home/ubuntu# ls
zabbix-release_5.0-1+ubuntu20.04_all.deb
root@ServeurAdministration:/home/ubuntu# █
```

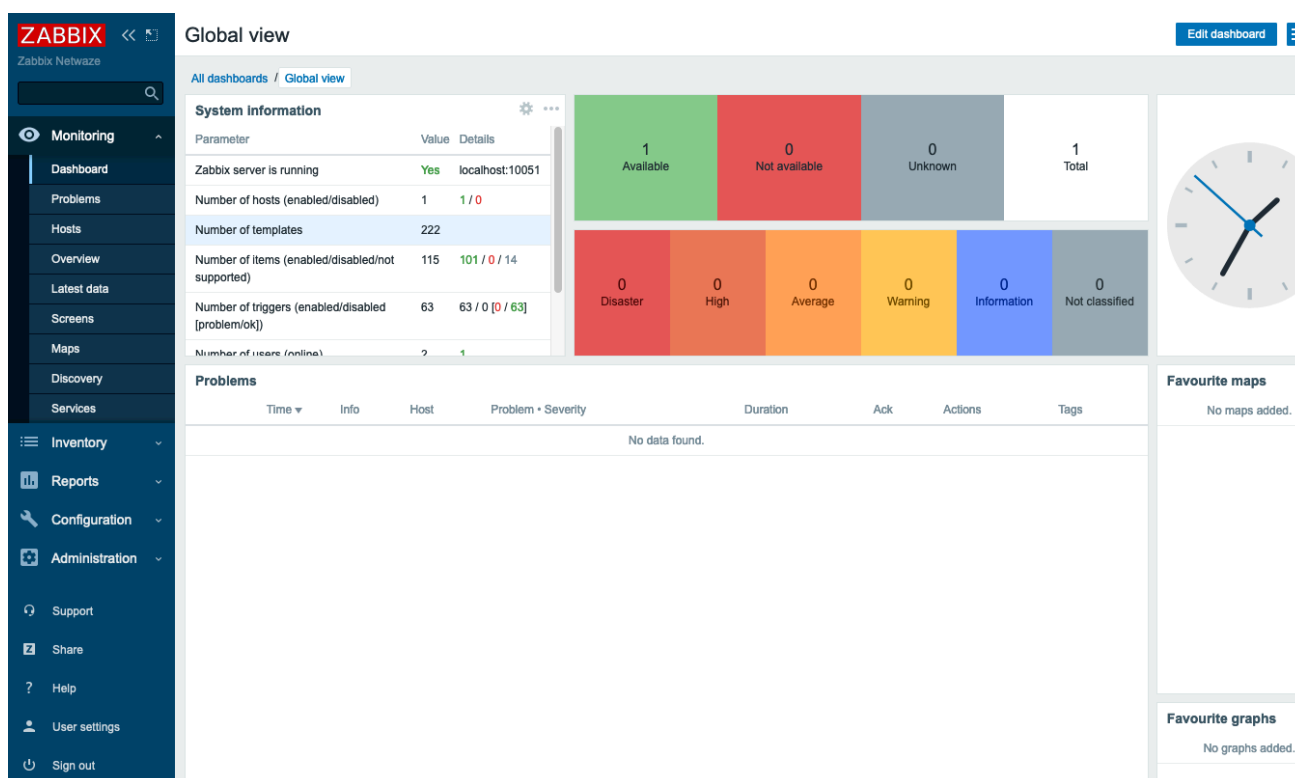
On retourne sur notre page Web et on clique sur le bouton finish :



The screenshot shows the Zabbix installation web interface. On the left, there is a navigation menu with the following items: Welcome, Check of pre-requisites, Configure DB connection, Zabbix server details, Pre-installation summary, and Install. The 'Install' item is highlighted. The main content area displays the Zabbix logo, the title 'Install', and a green message: 'Congratulations! You have successfully installed Zabbix frontend.' Below this message, it states 'Configuration file "/usr/share/zabbix/conf/zabbix.conf.php" created.' At the bottom right, there are two buttons: 'Back' and 'Finish'.

*Fin de la configuration de notre Zabbix*

Nous pouvons maintenant nous connecter à notre espace Zabbix. L'utilisateur par défaut est Admin et le mot de passe zabbix.



Maintenant nous allons lui créer un host virtuel pour pouvoir le prendre à distance.

Pour ce faire nous allons créer un nouveau fichier dans /etc/apache2/sites-available/ que je vais appeler scani.conf puis on lui rentre cette configuration par défaut.

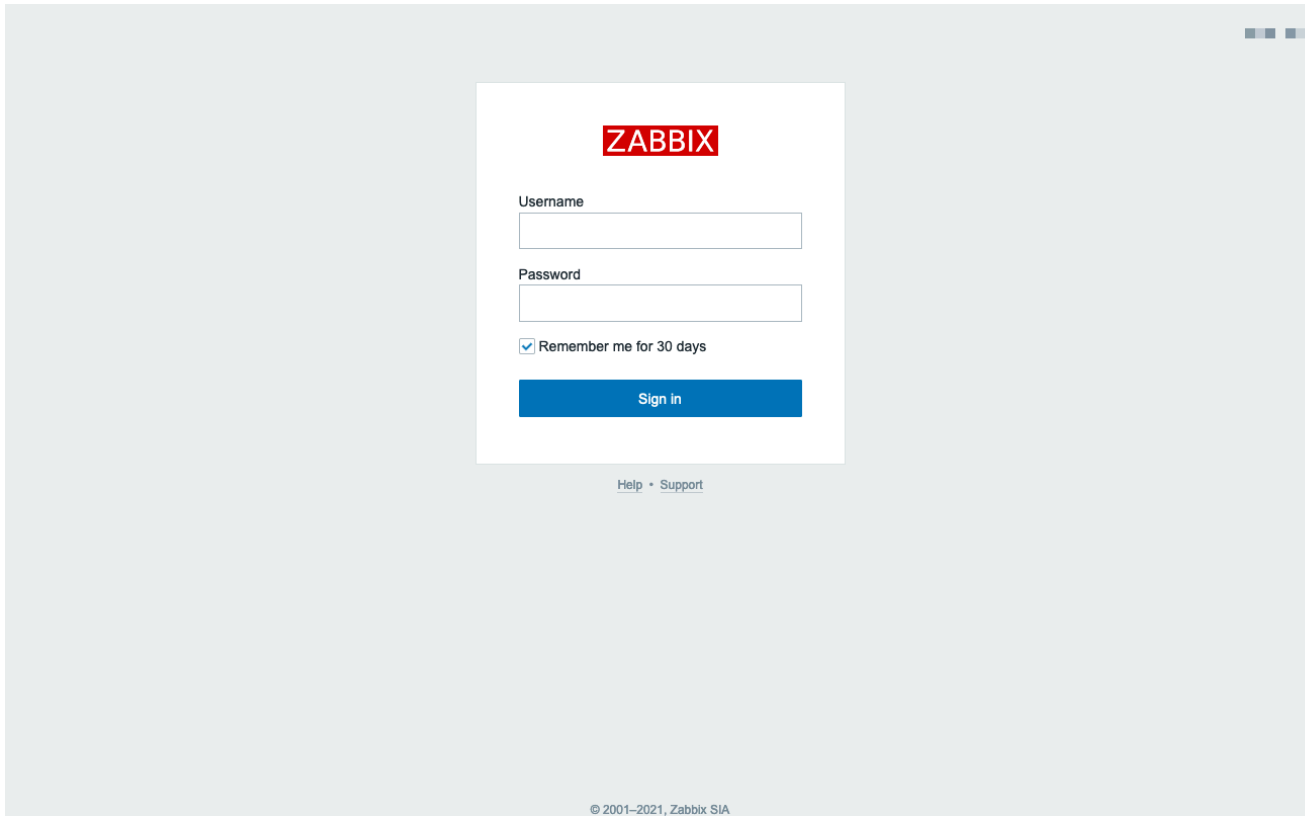
```
<VirtualHost *:80>
  ServerAdmin zabbix@scani.fr
  ServerName zabbix.scani.fr
  ServerAlias www.zabbix.scani.fr
  DocumentRoot /usr/share/zabbix
  ErrorLog ${APACHE_LOG_DIR}/zabbix.error.log
  CustomLog ${APACHE_LOG_DIR}/zabbix.access.log combined
</VirtualHost>
```

On active notre hôte virtuel.

```
$ sudo a2ensite zabbix.netwaze.fr.conf
```

On redémarre apache

```
$ systemctl restart apache2
```



*Notre Zabbix avec notre hôte virtuel*

## INSTALLATION DE CACHET

Pour installer cachet on commence par créer une base de données.

```
$ mysql -u root -p
$ CREATE DATABASE cachet;
$ CREATE USER 'cachet'@'localhost' IDENTIFIED BY 'password';
$ GRANT ALL ON cachet.* TO 'cachet'@'localhost' IDENTIFIED BY 'password' WITH GRANT
OPTION;
$ FLUSH PRIVILEGES;
$ EXIT;
```

Nous téléchargeons cachet directement depuis le dépôt GitHub du projet mais avant nous installons git, curl et composer.

```
$ apt install curl git
$ curl -sS https://getcomposer.org/installer | sudo php -- --install-dir=/usr/local/bin --
filename=composer
$ cd /var/www/
```

```
$ sudo git clone -b 2.4 --single-branch https://github.com/cachethq/Cachet.git cachet
```

A l'heure d'écriture, la version stable est la 2.4.

```
root@ServeurAdministration:/var/www# sudo git clone -b 2.4 --single-branch https://github.com/cachethq/Cachet.g
it cachet
Cloning into 'cachet'...
remote: Enumerating objects: 50994, done.
remote: Counting objects: 100% (1/1), done.
remote: Total 50994 (delta 0), reused 0 (delta 0), pack-reused 50993
Receiving objects: 100% (50994/50994), 34.70 MiB | 4.81 MiB/s, done.
Resolving deltas: 100% (34895/34895), done.
root@ServeurAdministration:/var/www#
```

Une fois télécharger nous avons un nouveau dossier Cachet.

Nous pouvons passer à la configuration du fichier de configuration.

```
$ sudo cp /var/www/cachet/.env.example /var/www/cachet/.env
$ nano /var/www/cachet/.env
```

Puis nous entrons nos configurations de base de données et nous pouvons sauvegarder et quitter.

```
GNU nano 4.8 /var/www/cachet/.env Modified
APP_ENV=production
APP_DEBUG=false
APP_URL=http://localhost
APP_TIMEZONE=UTC
APP_KEY=
DEBUGBAR_ENABLED=false

DB_DRIVER=mysql
DB_HOST=localhost
DB_UNIX_SOCKET=null
DB_DATABASE=cachet
DB_USERNAME=cachet
DB_PASSWORD=
DB_PORT=null
DB_PREFIX=null

CACHE_DRIVER=file
```



Maintenant nous devons installer tous les paquets avec composer pour que cachet fonctionne correctement. `$ sudo composer install --no-dev -o`

```
root@ServeurAdministration:/var/www# cd cachet/
root@ServeurAdministration:/var/www/cachet# composer install --no-dev -o
Do not run Composer as root/super user! See https://getcomposer.org/root for details
Continue as root/super user [yes]? yes
Installing dependencies from lock file
Verifying lock file contents can be installed on current platform.
Package operations: 96 installs, 0 updates, 0 removals
  - Failed to download kylekatarnls/update-helper from dist: The zip extension and unzip/7z commands are both missing, skipping.
Your command-line PHP is using multiple ini files. Run `php --ini` to show them.
  - Now trying to download from source
  - Syncing kylekatarnls/update-helper (1.2.1) into cache
  - Failed to download symfony/polyfill-ctype from dist: The zip extension and unzip/7z commands are both missing, skipping.
Your command-line PHP is using multiple ini files. Run `php --ini` to show them.
  - Now trying to download from source
```

Puis nous installons la clé de l'application et cachet

```
$ php artisan key:generate
```

```
$ php artisan cachet:install
```

Pour l'installation nous allons faire au plus simple car sinon on va se perdre dans les options. Quand cachet va poser ces questions (les 2 premières) appuyer sur entrer. L'installation se fera toute seule.

```
Migrating: 2017_07_18_214718_CreateIncidentComponents
Migrated: 2017_07_18_214718_CreateIncidentComponents
Migrating: 2017_09_14_180434_AlterIncidentsAddUserId
Migrated: 2017_09_14_180434_AlterIncidentsAddUserId
Migrating: 2018_04_02_163328_CreateTaggablesTable
Migrated: 2018_04_02_163328_CreateTaggablesTable
Migrating: 2018_04_02_163658_MigrateComponentTagTable
Migrated: 2018_04_02_163658_MigrateComponentTagTable
Migrating: 2018_06_14_201440_AlterSchedulesSoftDeletes
Migrated: 2018_06_14_201440_AlterSchedulesSoftDeletes
Migrating: 2018_06_17_182507_AlterIncidentsAddNotifications
Migrated: 2018_06_17_182507_AlterIncidentsAddNotifications
Migrating: 2019_12_12_131400_AlterJobsDropReserved
Migrated: 2019_12_12_131400_AlterJobsDropReserved
Database seeding completed successfully.
Clearing cache...
Application cache cleared!
Cache cleared!
The [public/storage] directory has been linked.
Cachet is installed ⚡
root@ServeurAdministration:/var/www/cachet#
```

Et nous donnons les droits à apache d'aller dans le dossier de Cachet.

```
$ sudo chown -R www-data:www-data /var/www/cachet/
```

```
$ sudo chmod -R 755 /var/www/cachet/
```

On lui créer à présent un virtuel host dans apache que l'on va appeler status.scani.fr

```
$ nano /etc/apache2/sites-available/status.scani.fr.conf
```

et on lui rentre la configuration suivante :

```
<VirtualHost *:80>
```

```
ServerAdmin admin@netwaze.fr
DocumentRoot /var/www/cachet/public
ServerName status.scani.fr
ServerAlias www.status.scani.fr
```

```
<Directory /var/www/cachet/public>
    Options FollowSymlinks
    AllowOverride All
    Require all granted
</Directory>
```

```
ErrorLog ${APACHE_LOG_DIR}/cachet.error.log
CustomLog ${APACHE_LOG_DIR}/cachet.access.log combined
```

```
<Directory /var/www/cachet/public>
    RewriteEngine on
    RewriteBase /
    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteRule ^(.*) index.php [PT,L]
```

```
</Directory>
```

```
</VirtualHost>
```

On active ensuite Cachet et on redémarre Apache

```
$ a2ensite services.netwaze.fr.conf
```


```
$ sudo a2enmod rewrite
```

```
$ sudo systemctl restart apache2
```

Nous pouvons maintenant ouvrir notre navigateur internet et aller sur notre site

Cachet

---



# Cachet

Configuration de l'environnement    Configuration de la page de statut    Compte administrateur    Terminer la configuration

● ○ ○ ○

**Gestionnaire de cache**

**Queue Driver**

**Gestionnaire de session**

**Mail Driver**

**Mail Host**

**Mail From Address**

**Mail Username**

**Mail Password**

Nous configurons notre gestionnaire de cache, queue Driver et gestionnaire de session sur Database puis nous configurons l'envoi de mail (je vais mettre log) puis nous continuons la configuration basic de Cachet.



Configuration de l'environnement    Configuration de la page de statut    Compte administrateur    Terminer la configuration



**Cachet a été configuré avec succès !**

[Aller au tableau de bord](#)

## FAIRE FONCTIONNER ZABBIX ET CACHET ENSEMBLE

Pour faire fonctionner Zabbix et Cachet ensemble on va utiliser un script Python créer par Artem Alexandrov alias qk4l sur Github.

Nous allons dans notre dossier Cachet et allons télécharger Git Clone le dossier de Zabbix-Cachet.

```
$ /var/www/cachet
```

```
$ git clone https://github.com/qk4l/zabbix-cachet.git
```

```
root@ServeurAdministration:/# cd /var/www/cachet
root@ServeurAdministration:/var/www/cachet# git clone https://github.com/qk4l/zabbix-cachet.git
Cloning into 'zabbix-cachet'...
remote: Enumerating objects: 215, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 215 (delta 3), reused 5 (delta 2), pack-reused 205
Receiving objects: 100% (215/215), 63.03 KiB | 2.63 MiB/s, done.
Resolving deltas: 100% (119/119), done.
root@ServeurAdministration:/var/www/cachet#
```

On va dans le dossier télécharger et on modifie le config-example.yml en config.yml

```
$ mv config.example.yml config.yml
```

et on installe les dépendances et logiciels nécessaires pour le script. (Au cas ou installer python3-pip)

```
$ pip install -r requirements.txt
```

```
root@ServeurAdministration:/var/www/cachet/zabbix-cachet# pip install -r requirements.txt
Collecting PyYAML==5.1
  Downloading PyYAML-5.1.tar.gz (274 kB)
    |#####| 274 kB 3.7 MB/s
Collecting pyzabbix==0.7.4
  Downloading pyzabbix-0.7.4.tar.gz (3.0 kB)
Collecting requests==2.20.0
  Downloading requests-2.20.0-py2.py3-none-any.whl (60 kB)
    |#####| 60 kB 328 kB/s
Collecting pytz==2019.3
  Downloading pytz-2019.3-py2.py3-none-any.whl (509 kB)
    |#####| 509 kB 4.0 MB/s
Collecting urllib3<1.25,>=1.21.1
```

Avant de lancer le script zabbix-cachet.py il faut que l'on aille configurer notre Zabbix pour que le script cherche bien les bonnes informations.

Par conséquent on peut mettre notre ssh de coté.

## CONFIGURATION DES SERVICES ZABBIX

Dans notre Zabbix partie configuration et Services il faut configurer un nouveau service. Pour que le script fonctionnent bien il faut créer un service Cachet avec l'ID 5. Pour ce faire nous allons créer 4 faux ID et les supprimer ensuite.

root	Add child	
1	Add child Delete	Problem, if at least one child has a problem
2	Add child Delete	Problem, if at least one child has a problem
3	Add child Delete	Problem, if at least one child has a problem
4	Add child Delete	Problem, if at least one child has a problem
Cachet	Add child Delete	Problem, if at least one child has a problem

Maintenant nous rajoutons un hôte (notre serveur proxy) il faut d'abord le créer, pour ce faire il suffit d'aller dans Configuration puis host et d'installer sur notre machine l'agent Zabbix et d'en faire la configuration. Pour des raisons de sécurité je ne vais pas divulguer ma configuration.

Installation de l'agent Zabbix : `dnf install zabbix40-agent.aarch64` (j'ai fait un report à la communauté redhat que l'agent n'était pas à jour mais ceci devrait fonctionner.)

Donc nous ajoutons un nouvel enfant (add child) qui sera Serveur Web et nous en rajoutons un autre pour être « Serveur Nginx »

Nous pouvons configurer plusieurs template et configurer plus de services mais ce n'est pas le but ici, je ferai un article sur les templates et ce que je surveille sur mon site internet.

## CONFIGURATION DU SCRIPT

Nous pouvons retourner sur notre ssh et configurer la config de notre script.

```
$ nano config.yml
```

On lui indique nos informations et on peut sauvegarder et quitter. (API ce trouve dans notre profil de cachet)

```
zabbix:
  user: [REDACTED]
  pass: [REDACTED]
  server: http://zabbix.[REDACTED].fr
  https-verify: false

cachet:
  token: [REDACTED]
  server: https://services.[REDACTED].fr
  https-verify: false

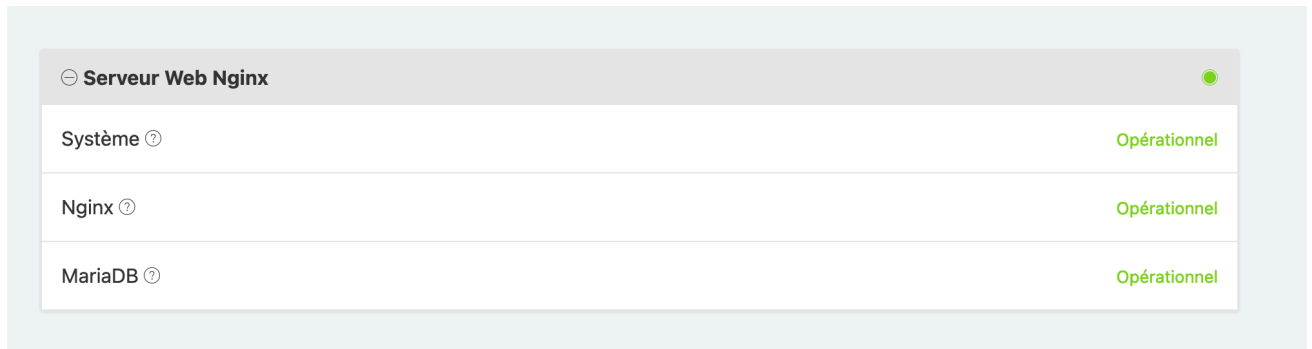
settings:
  # IT Service which will be a root for Cachet Components
  # Leave it empty if you want to use /
  root_service: 'Cachet'

  # How often check Zabbix for new incidents
  update_inc_interval: 30 # in seconds
  # How often check Zabbix for new IT Services
  update_comp_interval: 120 # in seconds
```

Nous pouvons maintenant lancer notre script avec la commande : `$ python3 zabbix-cachet.py`

```
root@ServeurAdministration:/var/www/cachet/zabbix-cachet# python3 zabbix-cachet.py
2021-08-23 14:36:35 UTC INFO: (MainThread) Zabbix Cachet v.1.3.7 started
2021-08-23 14:36:35 UTC INFO: (MainThread) JSON-RPC Server Endpoint: http://zabbix.netwaze.fr/api_jsonrpc.php
2021-08-23 14:36:36 UTC INFO: (MainThread) Zabbix ver: 5.0.14. Cachet ver: 2.4.0-dev
2021-08-23 14:36:36 UTC INFO: (MainThread) Component Zabbix was created in group id 0.
2021-08-23 14:36:36 UTC INFO: (MainThread) Successfully synced Cachet components with Zabbix Services
2021-08-23 14:36:36 UTC INFO: (MainThread) Restart triggers_watcher worker
2021-08-23 14:36:36 UTC INFO: (Trigger Watcher) start trigger watcher
```

Notre script fonctionne et est lancé et commence à se mettre en écoute des problèmes.



⊖ Serveur Web Nginx	
Système ?	Opérationnel
Nginx ?	Opérationnel
MariaDB ?	Opérationnel

On voit que notre Serveur Nginx répond bien avec notre script et Cachet. (Entre temps j'ai rajouter des fonctionnalités comme Nginx et MariaDB) Système équivaut à l'agent zabbix.

## Partie 2 – Validation.

### TEST

On va maintenant causer une panne sur notre serveur NGINX, on va simplement arrêter Nginx, Mariadb et l'agent zabbix.

```
$ systemctl stop nginx
```

```
$ systemctl stop mariadb
```

```
$ systemctl stop zabbix-agent
```

```
[[root@WebServer-Rocky ~]# systemctl stop nginx ]
[[root@WebServer-Rocky ~]# systemctl stop mariadb ]
[[root@WebServer-Rocky ~]# systemctl stop zabbix-agent ]
[root@WebServer-Rocky ~]# █
```

Time	Info	Host	Problem • Severity	Duration	Ack
11:10:20		ServeurWebNginx	Zabbix agent is not available (for 3m)	47s	No
11:06:29		ServeurWebNginx	MySQL: Service is down	4m 38s	No
11:06:13		ServeurWebNginx	Nginx: Process is not running	4m 54s	No
11:00					

Certains systèmes rencontrent une panne majeure

Serveur Web Nginx	
Système	Panne majeure
Nginx	Panne partielle
MariaDB	Panne majeure

### Incidents antérieurs

samedi 4th septembre 2021

Système Serveur Web Nginx | Zabbix agent is not available (for 3m)

Serveur Web Nginx | Système check failed - Sep 04, 11:10

Zabbix agent is not available (for 3m)

Nous constatons que Zabbix remonte les informations correspondantes à nos pannes et cache et nous les récupère pour les afficher dans la page web avec le problème correspondant.

Une fois que les services sont redémarrer tout repasse au vert et les problèmes sont corriger.

Time	Severity	Recovery time	Status	Info	Host	Problem
11:17:27	Information	11:26:27	RESOLVED		ServeurWebNginx	MySQL: Service has been restarted (uptime < 10m) ?
⊖ Serveur Web Nginx						Dernière mise à jour Samedi 4th septembre 2021 11:17:27
Système ?						Opérationnel
Nginx ?						Opérationnel
MariaDB ?						Opérationnel

### Partie 3 – Veille technologique.

Après des recherches nous n'avons rien trouver d'autre qui pouvait ce connecter avec Zabbix donc Cachet est la seule solution que nous pouvons utiliser.