LE BRAS
Maxime
Réseaux & Systèmes

Le 01/10/2026



Projet Supervision et sécurisation de services web

Sommaire

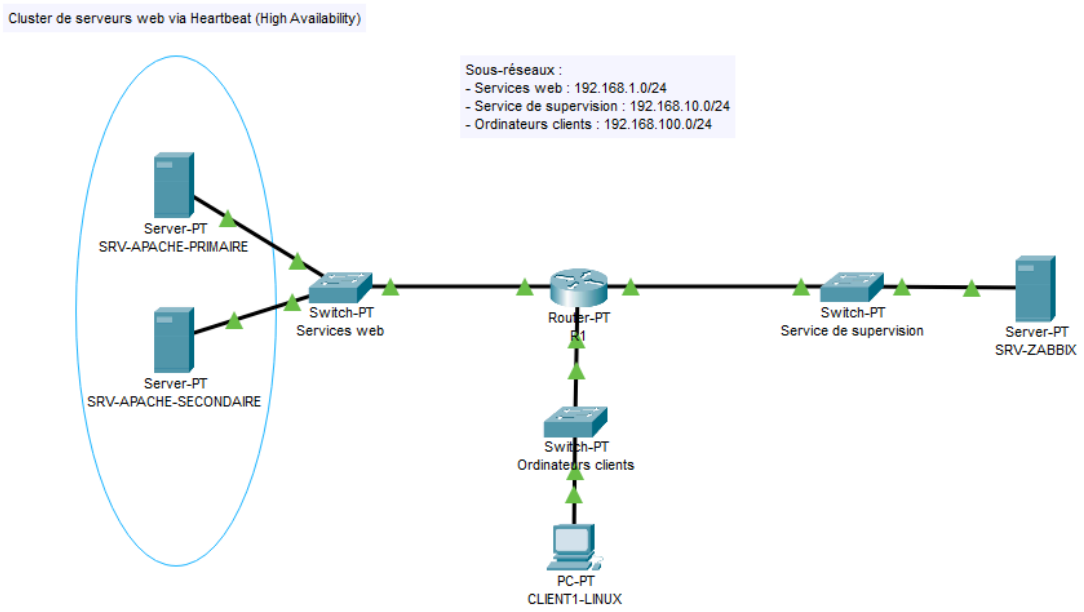
1. Contexte du projet **3**
2. Organisation de l'infrastructure réseau **3**
2.1. Schéma réseau 3
2.2. Plan d'adressage des machines (ordinateurs, serveurs, routeurs) 3
3. Sécurisation des services web **4**
3.1. Installation et configuration des services web 4
3.2. Mise en place de la haute disponibilité 5
3.3. Tests de fonctionnement 7
4. Supervision des services web **9**
4.1. Mise en place du serveur de supervision 9
4.2. Mise en place de l'agent de supervision 12
4.3. Tests de fonctionnement 13
5. Bibliographie **14**
6. Conclusion **14**

1. Contexte du projet

J'ai fait ce projet, car la supervision et la haute disponibilité sont complémentaires dans le cadre de la sécurité d'une infrastructure réseau.
De plus, la maîtrise de ces concepts informatiques est fondamentale dans le domaine de l'IT.
Enfin, cela m'a permis de combiner mes acquis avec de nouvelles notions.

2. Organisation de l'infrastructure réseau

2.1. Schéma réseau



2.2. Plan d'adressage des machines (ordinateurs, serveurs, routeurs)

Web :

- SRV-APACHE-PRIMAIRE : 192.168.1.1/24
- SRV-APACHE-SECONDAIRE : 192.168.1.2/24
- IP partagée (Heartbeat - HA) : 192.168.1.3/24

Supervision :

- SRV-ZABBIX : 192.168.10.1/24

Clients :

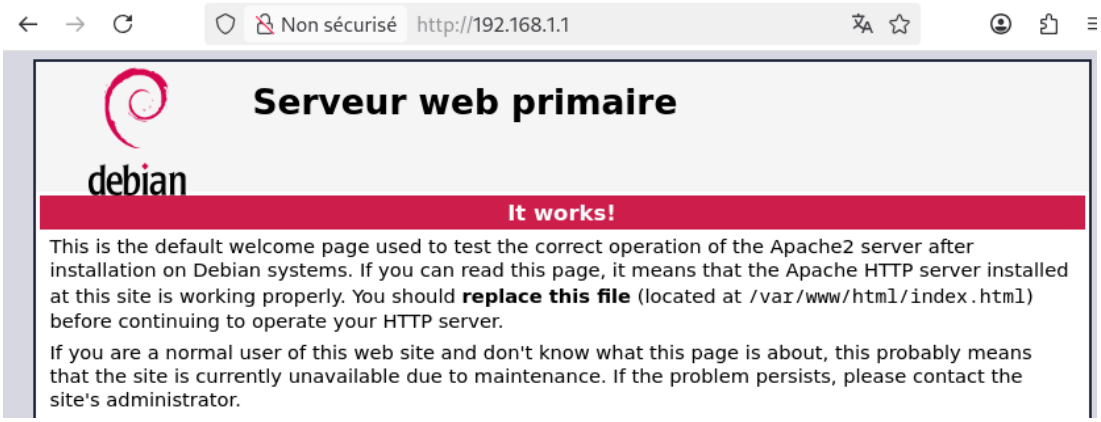
- CLIENT1-LINUX : 192.168.100.1/24

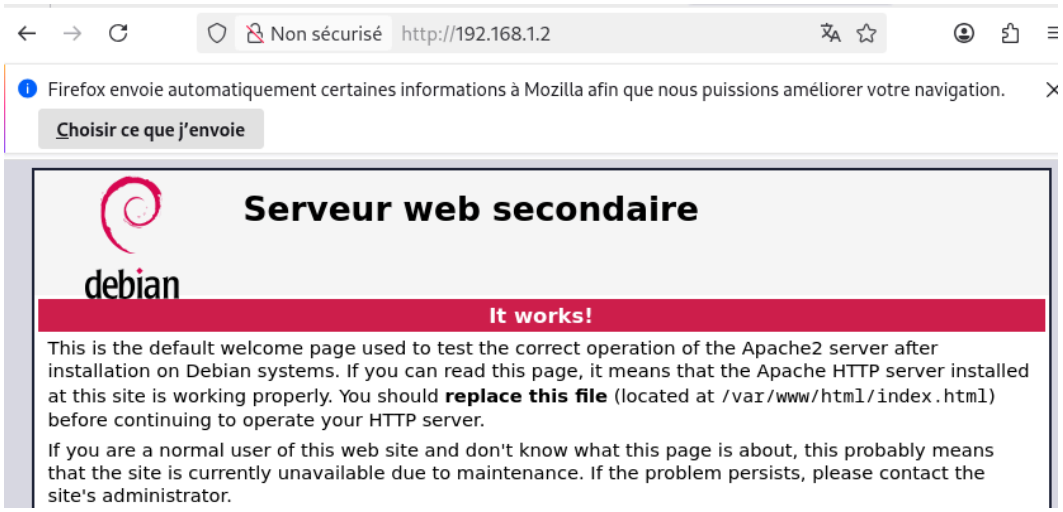
R1 :

- ens33 : 192.168.100.254/24
- ens37 : 192.168.1.254/24
- ens38 : 192.168.10.254/24

3. Sécurisation des services web

3.1. Installation et configuration des services web





Cela représente les deux serveurs web Apache qu'on utilisera pour configurer le cluster via heartbeat téléchargé via le gestionnaire de packages apt.

3.2. Mise en place de la haute disponibilité



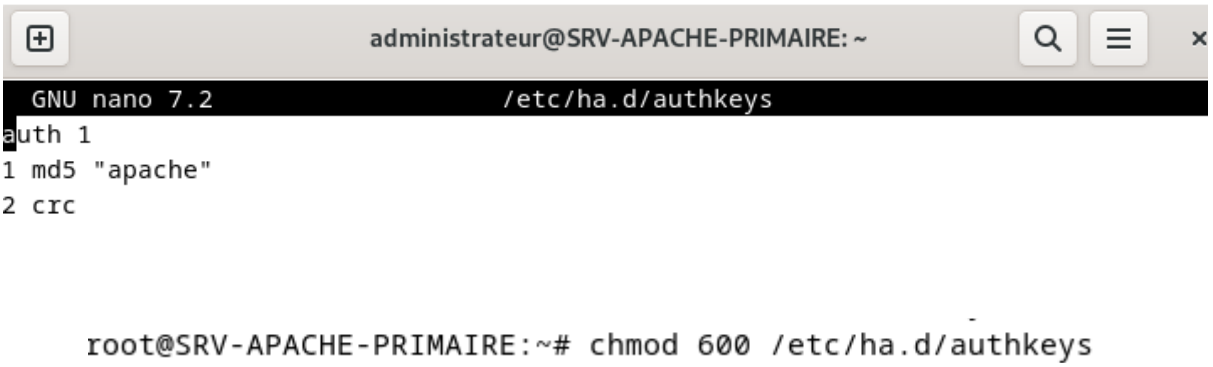


La configuration des noms d'hôtes est essentielle, car le système Heartbeat s'appuie sur ceux-ci pour identifier ses serveurs (ou nœuds) dans le cluster.

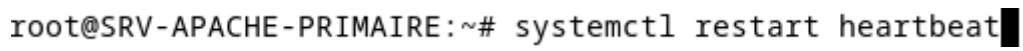


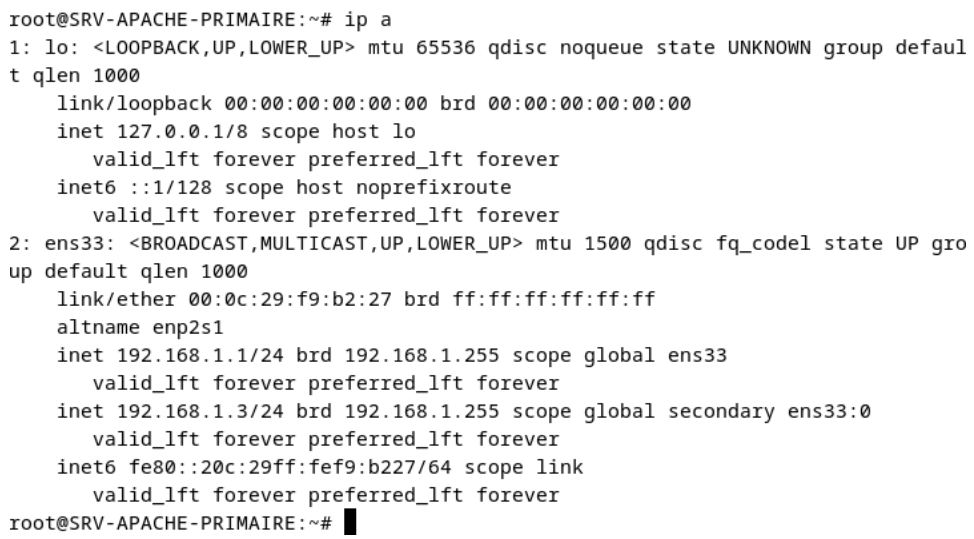


Ceci représente la configuration du cluster avec l'adresse IP partagée et ses paramètres (identification du serveur maître, mise en place des fichiers de logs, automatisation du retour vers son serveur principal dans le cas où on le met en route, etc…).



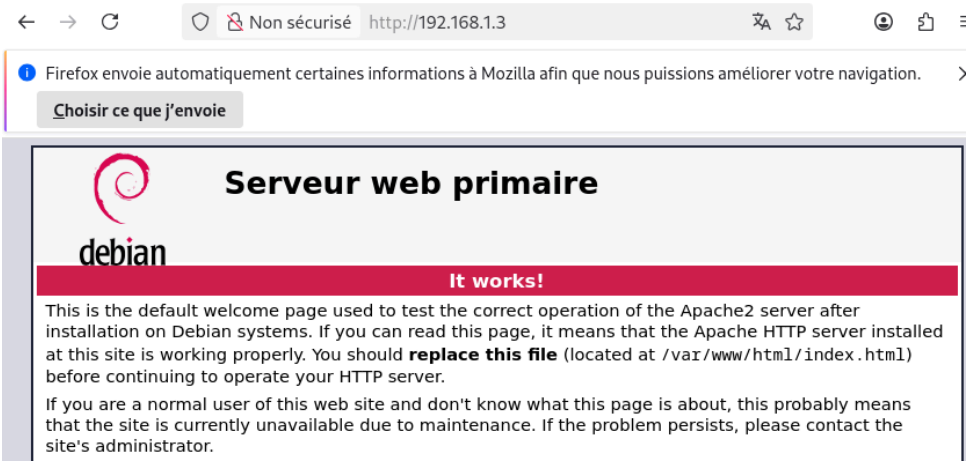
La mise en place du fichier /etc/ha.d/authkeys permet de sécuriser les communications entre les nœuds du cluster via un message secret où on va l'adapter en MD5 (algorithme de hachage cryptographique).





Après avoir mis la même configuration sur les deux serveurs web, on va redémarrer le service heartbeat pour mettre à jour ce qui a été mis en place et on constate que tout est fonctionnel avec l'apparition de l'adresse IP virtuelle sur le nœud principal.

3.3. Tests de fonctionnement

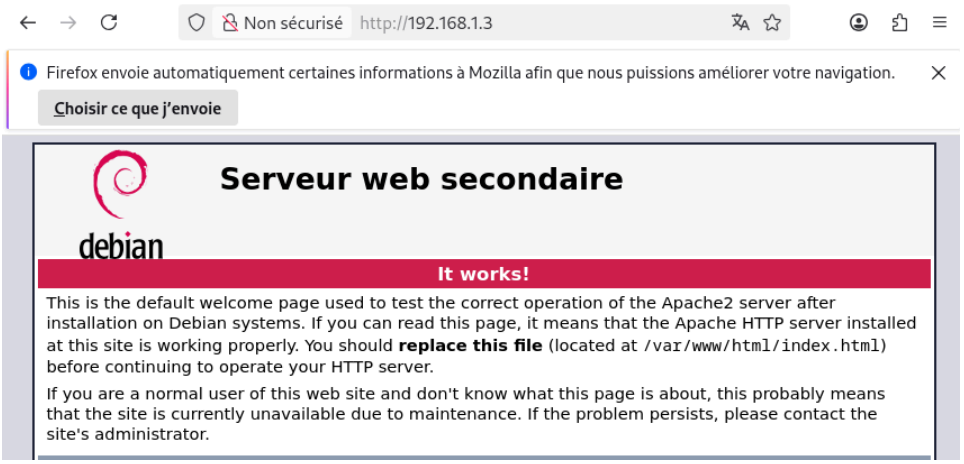



```
root@SRV-APACHE-PRIMAIRE:~# ifdown ens33
```

Le premier test consiste à créer une panne réseau sur le serveur primaire afin de voir si le serveur secondaire prend le relais.

```
root@SRV-APACHE-SECONDAIRE:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gro
up default qlen 1000
    link/ether 00:0c:29:88:f6:94 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.1.2/24 brd 192.168.1.255 scope global ens33
       valid_lft forever preferred_lft forever
    inet 192.168.1.3/24 brd 192.168.1.255 scope global secondary ens33:0
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe88:f694/64 scope link
       valid_lft forever preferred_lft forever
```

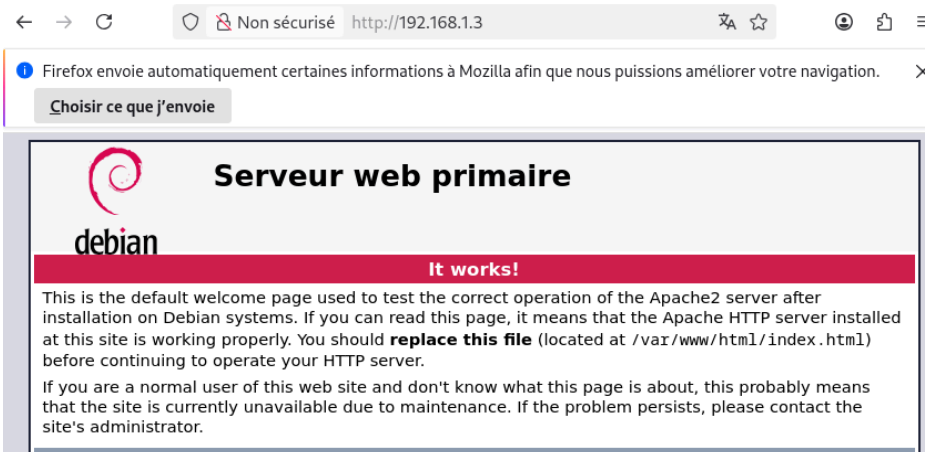
Il est fonctionnel puisqu'on voit que l'adresse IP partagée est présente sur le serveur secondaire ce qui veut dire qu'il remplace le serveur primaire.




```
root@SRV-APACHE-PRIMAIRE:~# ifup ens33
```

Le deuxième test consiste à remettre en route l'interface réseau du serveur primaire afin de voir s'il reprend son rôle de serveur maître au sein du cluster.

```
root@SRV-APACHE-PRIMAIRE:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defau
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP gr
up default qlen 1000
    link/ether 00:0c:29:f9:b2:27 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.1.1/24 brd 192.168.1.255 scope global ens33
       valid_lft forever preferred_lft forever
    inet 192.168.1.3/24 brd 192.168.1.255 scope global secondary ens33:2
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fef9:b227/64 scope link
       valid_lft forever preferred_lft forever
root@SRV-APACHE-PRIMAIRE:~#
```



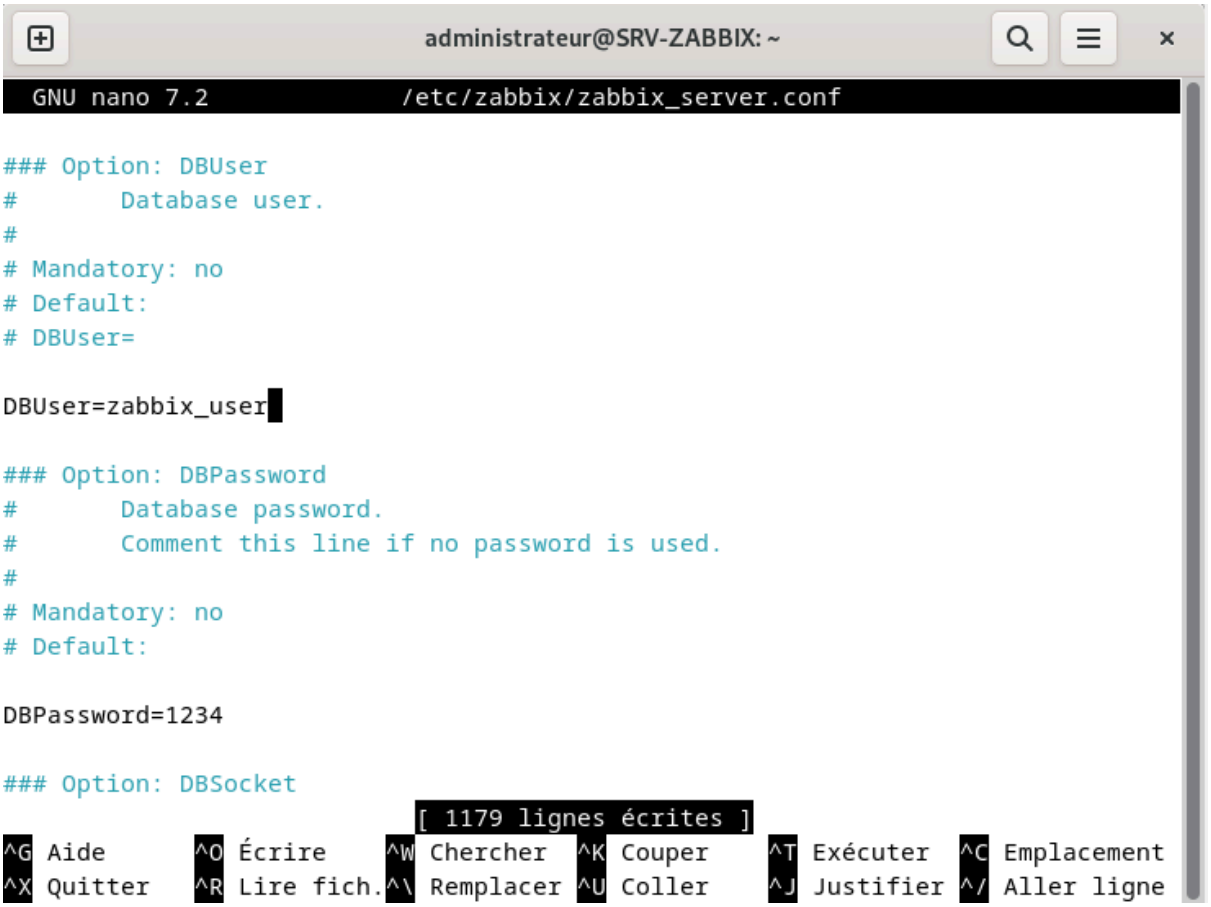
Il est opérationnel puisque le serveur primaire a bien repris l'adresse IP partagée du cluster.

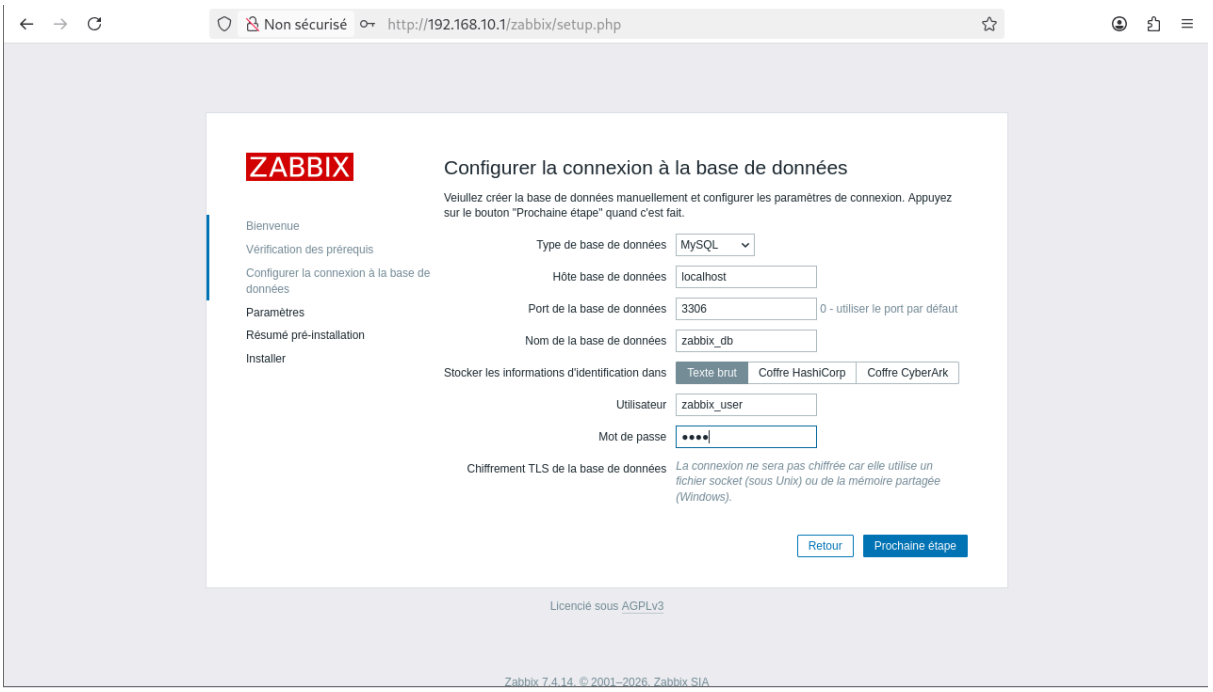
4. Supervision des services web

4.1. Mise en place du serveur de supervision

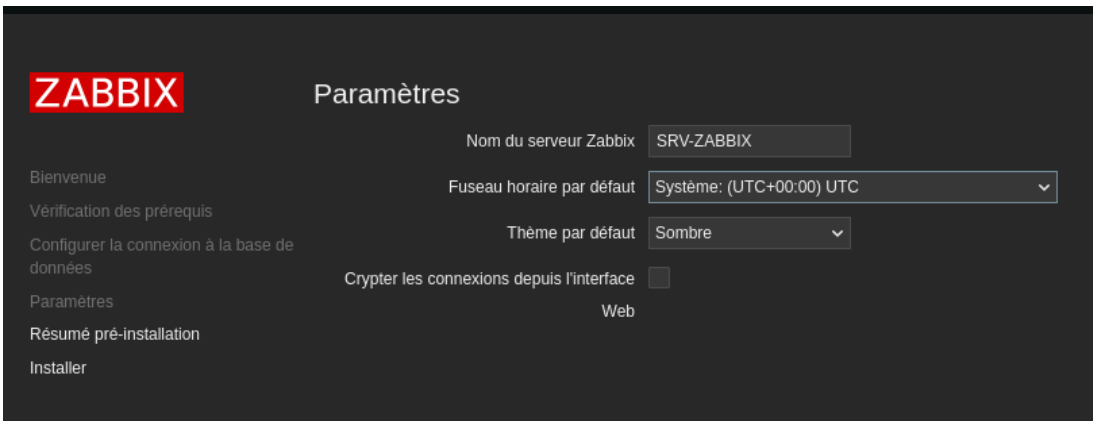
```
root@SRV-ZABBIX:~# apt update && apt full-upgrade -y
root@SRV-ZABBIX:~# wget https://repo.zabbix.com/zabbix/7.4/release/debian/pool/main/z/zabbix-release/zabbix-release_latest_7.4+debian12_all.deb
root@SRV-ZABBIX:~# dpkg -i zabbix-release_latest_7.4+debian12_all.deb
root@SRV-ZABBIX:~# apt install mariadb-server zabbix-server-mysql zabbix-frontend-php zabbix-apache-conf zabbix-sql-scripts zabbix-agent
root@SRV-ZABBIX:~# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 5
Server version: 10.11.18-MariaDB-0+deb12u1 Debian 12

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE zabbix_db;
Query OK, 1 row affected (0,000 sec)

MariaDB [(none)]> CREATE USER zabbix_user@localhost IDENTIFIED BY "1234";
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON zabbix_db.* TO zabbix_user@localhost;
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> quit;
root@SRV-ZABBIX:~# dpkg -L zabbix-sql-scripts | grep mysql
/usr/share/zabbix/sql-scripts/mysql
/usr/share/zabbix/sql-scripts/mysql/option-patches
/usr/share/zabbix/sql-scripts/mysql/option-patches/history_upgrade_prepare.sql
/usr/share/zabbix/sql-scripts/mysql/proxy.sql
/usr/share/zabbix/sql-scripts/mysql/server.sql.gz
root@SRV-ZABBIX:~# zcat /usr/share/zabbix/sql-scripts/mysql/server.sql.gz | mysql -u zabbix_user -p zabbix_db
Enter password:
root@SRV-ZABBIX:~#
```

Ces commandes permettent d'avoir une configuration fonctionnelle avec la mise en place de MySQL (création d'un utilisateur et de la base de données, importation d'un script SQL lié au service Zabbix, etc…).

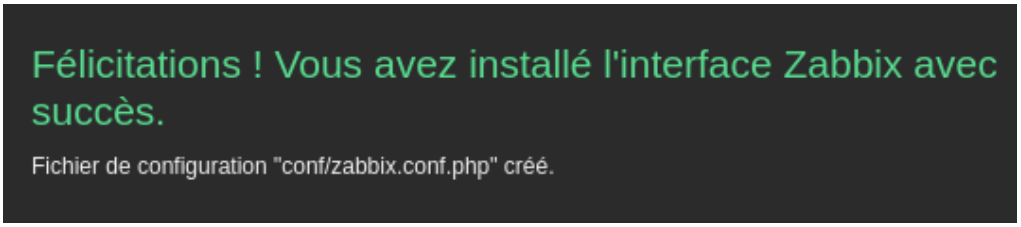




Ceci correspond à la configuration de la liaison entre la base de données et le service Zabbix.







L'horodatage est fondamental en supervision, car une dérive de temps d'une heure sur un événement survenu à 10h sur un serveur quelconque, rendrait l'analyse des journaux système inefficace.

4.2. Mise en place de l'agent de supervision

```
root@SRV-APACHE-PRIMAIRE:~# wget https://repo.zabbix.com/zabbix/7.4/release/debi
an/pool/main/z/zabbix-release/zabbix-release_latest_7.4+debian12_all.deb
```

```
root@SRV-APACHE-PRIMAIRE:~# dpkg -i zabbix-release_latest_7.4+debian12_all.deb
```

```
root@SRV-APACHE-PRIMAIRE:~# apt update && apt install zabbix-agent
```

Ceci correspond à l'installation du client Zabbix qui va être intégré aux deux services Apache.

```
 GNU nano 7.2                 /etc/zabbix/zabbix_agentd.conf
# Default:
# Server=

Server=192.168.10.1
```

```
ServerActive=192.168.10.1

### Option: Hostname
#       List of comma delimited unique
#       Required for active checks and
#       Value is acquired from Hostnam
#
# Mandatory: no
# Default:
# Hostname=

Hostname=SRV-ZABBIX
```

On effectue la configuration du client en faisant une liaison réseau entre lui-même et le serveur afin qu'il soit bien référencé dans celui-ci.

```
root@SRV-APACHE-PRIMAIRE:~# systemctl enable zabbix-agent
Synchronizing state of zabbix-agent.service with SysV service script with /lib/s
ystemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable zabbix-agent
root@SRV-APACHE-PRIMAIRE:~# systemctl start zabbix-agent
root@SRV-APACHE-PRIMAIRE:~# systemctl status zabbix-agent.service
● zabbix-agent.service - Zabbix Agent
     Loaded: loaded (/lib/systemd/system/zabbix-agent.service; enabled; preset:>
     Active: active (running) since Tue 2026-09-22 17:34:07 CEST; 5min ago
   Main PID: 7740 (zabbix_agentd)
      Tasks: 13 (limit: 2234)
     Memory: 9.7M
        CPU: 649ms
     CGroup: /system.slice/zabbix-agent.service
             ├─7740 /usr/sbin/zabbix_agentd -c /etc/zabbix/zabbix_agentd.conf
```

Cette commande permet de lancer zabbix-agent au démarrage de la machine.

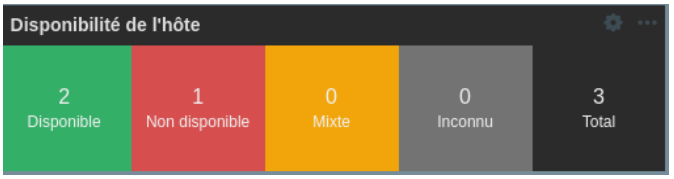
```
root@SRV-APACHE-PRIMAIRE:~# curl "http://127.0.0.1/server-status?auto"
127.0.0.1
ServerVersion: Apache/2.4.68 (Debian)
ServerMPM: event
Server Built: 2026-06-12T05:08:45
CurrentTime: Tuesday, 22-Sep-2026 17:41:26 CEST
RestartTime: Tuesday, 22-Sep-2026 14:47:32 CEST
ParentServerConfigGeneration: 1
ParentServerMPMGeneration: 0
ServerUptimeSeconds: 10433
ServerUptime: 2 hours 53 minutes 53 seconds
Load1: 0.32
Load5: 0.17
Load15: 0.07
Total Accesses: 15
Total kBytes: 56
Total Duration: 55
```

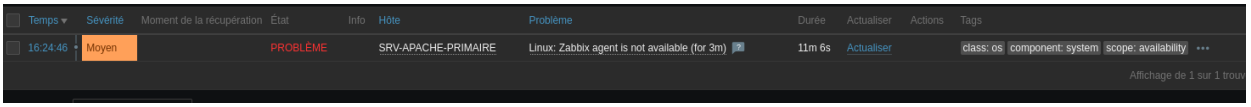
A la suite de cette commande, elle permet d'analyser les informations systèmes du serveur avec l'horloge système, le temps de fonctionnement, la version du service Apache avec la distribution Linux utilisée, etc…

4.3. Tests de fonctionnement

```
root@SRV-APACHE-PRIMAIRE:~# ifdown ens33
root@SRV-APACHE-PRIMAIRE:~# ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST> mtu 1500 qdisc fq_codel state DOWN group default
 qlen 1000
    link/ether 00:0c:29:f9:b2:27 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
root@SRV-APACHE-PRIMAIRE:~#
```

Le premier test consiste à voir si le serveur Zabbix détecte la panne réseau sur le service Apache primaire.



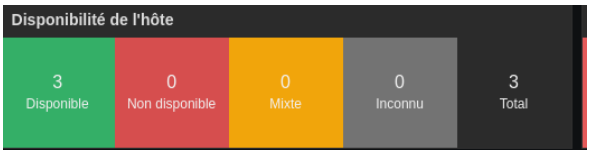


On constate que le test est positif puisqu'on voit qu'il ne détecte plus l'agent Zabbix du serveur Apache primaire.

```
root@SRV-APACHE-PRIMAIRE:~# ifup ens33
```

Le deuxième test est composé de la remise en marche de l'interface réseau du serveur web principal et de la vérification de sa disponibilité sur le serveur Zabbix.

```
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP grc
up default qlen 1000
    link/ether 00:0c:29:f9:b2:27 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.1.1/24 brd 192.168.1.255 scope global ens33
       valid_lft forever preferred_lft forever
    inet 192.168.1.3/24 brd 192.168.1.255 scope global secondary ens33:6
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fef9:b227/64 scope link
       valid_lft forever preferred_lft forever
```



Le résultat de ce test est satisfaisant puisque le serveur web principal est remonté sur le service Zabbix.

5. Bibliographie

- https://maxlbs.ovh/tp_bts-sio/tp_heartbeat.pdf (TP3 Bloc 3 Cybersécurité - BTS SIO 2ème année - Haute disponibilité et répartition de charges des serveurs)
- https://maxlbs.ovh/procedures/personnelles/procedure_installation_zabbix_debian12.pdf (Procédure qui comporte l'installation du serveur et de l'agent Zabbix)

6. Conclusion

Pour conclure, ce projet personnel m'a permis de réviser la mise en place d'un système de haute disponibilité et de découvrir la supervision avec Zabbix, de l'installation du serveur au déploiement d'un agent. À l'avenir, j'aimerais apprendre à mettre en place un VPN site-à-site.