
Rabanov
Rabanov Hotel vendée
Version <1.0>

HOTEL

RABANOV



Mise en place du Serveur Web

| | |
|------------------------------|------------------|
| GSB | Version: <1.0> |
| Mise en place du cluster Web | Date: 26/04/2016 |

Historique des révisions

| Date | Version | Description | Auteur |
|------------|---------|---|------------------|
| 25/04/2016 | <1.0> | Mise en place de la première installation | Brice Harismendy |
| | | | |
| | | | |
| | | | |

| | |
|------------------------------|------------------|
| GSB | Version: <1.0> |
| Mise en place du cluster Web | Date: 26/04/2016 |

Table des matières

1. Introduction

- 1.1 Contexte du projet
- 1.2 Objectifs du document
- 1.3 Portée

2. Éléments de configuration

- 2.1 Installation de HaProxy
- 2.2 Mise en place du load balancing
- 2.3 Réplication des base de données
- 2.4 Synchronisation des fichiers avec rsync

| | |
|------------------------------|------------------|
| GSB | Version: <1.0> |
| Mise en place du cluster Web | Date: 26/04/2016 |

Mise en place du cluster WEB

1. Introduction

L'hotel Rabanov vient d'être construit et un cluster web va y être installer il faut le sécuriser et rédiger une documentation technique permettant de le reconfigurer en cas de grave problème.

1.1 Contexte du projet

Un groupe Hotelier international a lancé sur la cote vendéenne la construction d'un nouvel hotel 4 étoiles de luxe « Le Rabanov »

1.2 Objectifs du document

Ce document a pour objectif de permettre la reconfiguration rapide du cluster web de l'hotel

1.3 Portée

Cette documentation s'adresse a l'équipe technique de l'hotel Rabanov

2. Éléments de configuration

Matériel nécessaire : deux serveur web debian 8,1

2.1 Installation de HaProxy

Après avoir configuré l'adresse ip :

installez HaProxy : apt-get update & apt-get install haproxy

2.2 Mise en place du load balancing

On va mettre un load balancing à 50/50 avec maintien des session (cookie) pour cela renseigner le fichier /etc/haproxy/haproxy.cfg

nano /etc/haproxy/haproxy.cfg

ce qui donne (à ajouter a la fin) :

```
frontend proxypublic
    bind 192.168.28.203:80
    default_backend fermeweb
backend fermeweb
    balance roundrobin
    option httpclose
    option forwardfor
    option httpchk HEAD / HTTP/1.0
    cookie RABANOVHOTEL insert indirect #creation du cookie
```

| | |
|------------------------------|------------------|
| GSB | Version: <1.0> |
| Mise en place du cluster Web | Date: 26/04/2016 |

```
server web1 192.168.28.201:80 cookie w1 weight 3 check#déclaration du serveur web1
server web2 192.168.28.202:80 cookie w1 weight 3 check#déclaration du serveur web2
stats uri /stats #definition de l'url d'accès aux statistiques
stats auth apo: P@ssw0rd #mot de passe et login de l'accès aux stats
stats refresh 30s #fréquence de rafraîchissement des statistiques
appsession RABANOVHOTEL len 64 timeout 24h request-learn #définition de la durée des
sessions
```

2.3 Réplication des bases de données

On crée sur chacun des serveurs un utilisateur nommé slave autorisé à se connecter à l'autre serveur et ayant comme seule privilège "réplication slave"

ensuite on configure my.cnf du 192,168,28,201 comme ceci :

```
[mysqld]
log_error = /var/log/mysql/error.log
server-id      = 1
log_bin        = /var/log/mysql/mysql-bin.log
binlog_do_db   = rabanov
```

et pour le 192,168,28,202 :

```
[mysqld]
log_error = /var/log/mysql/error.log
server-id      = 2
log_bin        = /var/log/mysql/mysql-bin.log
binlog_do_db   = rabanov
```

puis sur chaque serveur j'ai fait les commandes suivantes :

```
touch /var/log/mysql/mysql-bin.log
touch /var/log/mysql/mysql-error.log
chown mysql -Rf /var/log/mysql/
```

puis connecter vous sur mysql :

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

et entrez les commandes suivantes :

```
(sur 192,168,28,201)
CHANGE MASTER TO
MASTER_HOST='192.168.28.202',
MASTER_PORT=3306,
MASTER_USER='slave',
MASTER_PASSWORD='P@ssw0rd',
MASTER_LOG_FILE='mysql-bin.000009',
MASTER_LOG_POS=1101;
```

| | |
|------------------------------|------------------|
| GSB | Version: <1.0> |
| Mise en place du cluster Web | Date: 26/04/2016 |

```
(et sur 192,168,28,202)
CHANGE MASTER TO
MASTER_HOST='192.168.28.201',
MASTER_PORT=3306,
MASTER_USER='slave',
MASTER_PASSWORD='P@ssw0rd',
MASTER_LOG_FILE='mysql-bin.000008',
MASTER_LOG_POS=1513;
```

2.4 Synchronisation des fichiers avec rsync

Tout d'abord installez rsync (apt-get install rsync)

On commence par tester de récupérer sur un des serveurs du cluster un fichier présent sur un autre serveur :

fait sur 192,168,28,201 :

```
rsync -avzu bharismendy@192.168.28.201:/var/www/ /home/bharismendy/
```

création de l'utilisateur de transfert sur le master :

```
root@debian# adduser replication
```

```
root@debian# usermod -g developpement replication
```

```
root@debian# adduser replication www-data
```

Pour envoyer les fichier sans mot de passe on va utiliser des clés public/privé

sur le serveur de développement on exécute les commande suivante dans le fichier root

```
root@debian: mkdir /home/replication/.ssh
```

```
root@debian:/home/bharismendy# ssh-keygen -t dsa
```

Generating public/private dsa key pair.

Enter file in which to save the key (/root/.ssh/id_dsa): /home/replication/.ssh

Enter passphrase (empty for no passphrase):

Enter same passphrase again:

On envoie la clé sur le serveur de sauvegarde :

Pour pouvoir envoyer nos fichier sans mot de passe on envoie la clé publique sur l'autre serveur : dans l'ordre :

sur l'autre serveur (28,202) :

```
root@debian: mkdir /home/replication/.ssh
```

sur le master (28,201) :

```
scp /home/replication/.ssh/id_dsa.pub
```

```
root@192.168.28.202:/home/replication/.ssh/authorized_keys
```

sur le serveur esclave :

| | |
|------------------------------|------------------|
| GSB | Version: <1.0> |
| Mise en place du cluster Web | Date: 26/04/2016 |

```
root@debian: chmod 0600 /home/replication/.ssh/authorized_keys
```

on peut tenter de copier en ligne de commande :

```
rsync -avz -e "ssh -i /home/replication/.ssh/id_dsa" root@192.168.28.201:/var/www/
/home/bharismendy/
```

automatisation de la commande avec une chron :

nous partons de cette commande a executer sur le slave :

```
rsync -avz -e "ssh -i /home/replication/id_dsa" root@192.168.28.201:/var/www/public_html/
/var/www/public_html/
```

elle copie les fichiers une fois par heure car le site web est encore legé : (sur le slave)

root@debian: crontab -e → '-e' pour 'edit' ça permet d'éditer le fichier des tâches automatisé
on rajoute l'entrée suivante à la suite des commentaires :

```
0 20 * * * rsync -avz --delete -e "ssh -i /home/replication/id_dsa"
root@192.168.28.201:/var/www/public_html/ /var/www/public_html/
```

explication :

0 : a la minute 0

1^{er}* : chaque heure

2^{ème}* : chaque du mois

3^{ème}* : chaque mois

4^{ème}* : chaque jour de la semaine

il faut maintenant relancer cron :

```
root@debian: /etc/init.d/cron restart
```

3. Conclusion

Le cluster web est maintenant configuré, Il permet une tolérance aux sites web ainsi qu'une répartition de charge sur notre site web. Le site web peut donc maintenant accueillir des centaines de visite et les répartissant entre les serveurs du cluster. De plus un cookie a été mis en place permettant d'éviter que l'utilisateur change de serveur a chaque nouvelle page chargée.