Suivi des modifications

Version	Référence	Auteur	Date	Commentaires
A	2SIO_20210940_AG	Antonin Grepilloux	07/10/2021	Création

OBJET :
Document complet de l'installation

COMMENTAIRES :

Document Technique

Ce document contient l'installation complète avec chaque phase détaillée et avec les temps afin de pouvoir calculer le coût d'installation de la suite logicielle.

CONFIGURATION REQUISE

MINIMALE :
Système d'exploitation : Linux Debian 10
Processeur : 1 core
Mémoire vive : 2 GB de mémoire
Espace disque : 8 GB d'espace disque disponible
Adaptateur réseau : NAT

RECOMMANDE :
Système d'exploitation : Linux Debain 10
Processeur : 2 cores
Mémoire vive : 4 GB de mémoire
Espace disque : 16 GB d'espace disque disponible
Adaptateur réseau : NAT

Prérequis :

La machine doit posséder mariadb, heartbeat pour pouvoir installer tous les logiciels. De plus, avant toutes opérations sur la machine il faut exécuter les commandes après avoir fait :

```
root@debian$: su -
```

Tableau :

	Service	Temps d'installation	Couts	Remarques
	MariaDB	→ 2 minutes	//	Aucuns problèmes
	HeartBeat	→ 5 minutes	//	Aucuns problèmes
	Réplication	→ 10 minutes	//	Aucuns problèmes
TOTAL	3 services	~17 minutes	0 euro	0 problèmes connus

Configuration des machines :

Pour changer le nom d'hôte de votre machine en lui-même, il suffit d'utiliser la commande hostnamectl :

```
hostnamectl set-hostname nom-de-votre-machine
```

Remplacez bien évidemment « nom-de-votre-machine » par bdd1 ou bdd2.

Puis, on va changer le nom dans le fichier hosts.

Modification du fichier hosts
Éditez, avec nano par exemple, le fichier /etc/hosts.

```
nano /etc/hosts
```

Nano s'ouvrira avec le contenu du fichier.
A la deuxième ligne, vous devriez avoir une adresse IP indiquée, en dessous de celle de localhost, c'est à dire de 127.0.0.1. En face de cette IP, vous aurez l'ancien nom, à changer par celui que vous avez indiqué ci-dessus. Faites attention de bien utiliser exactement la même orthographe.

Redémarrez vos machines avec la commande suivant :

Ensuite nous allons définir les adresses IP des machines :

```
reboot
```

Pour changer l'adresse IP de la machine, modifiez le fichier interfaces :

```
nano /etc/network/interfaces
```

Dans ce fichier remplacez les lignes :

```
auto eth0
iface eth0 inet dhcp
```

Par les suivantes :

```
auto eth0
iface eth0 inet static
address 192.168.2.101
netmask 255.255.255.0
```

Vous venez alors de passer du mode DHCP (attribution automatique d'adresse IP) au mode statique (définition d'une adresse IP fixe).

Pour que Linux prenne en compte vos changements, vous devez fermer votre fichier et redémarrer le service de connexion réseau avec la ligne suivante :

```
sudo /etc/init.d/networking restart
```

/!\ Attention cette étape est à faire aussi sur la seconde machine.

Solution pour la présentation :

Pour faciliter l'utilisation de la machine et de ses logiciels pour le portage vers l'ESXi, nous avons mis en place un format simple d'utilisation qui est un format de fichier s'appelant OVF (format de fichier qui prend en charge l'échange de dispositifs virtuels entre les produits et les plateformes). Ce format peut être utilisé sur un poste ou un ordinateur portable avec le logiciel Vmware Workstation ou Vmware Player. Il suffit d'importer le fichier Ovf depuis le logiciel pour pouvoir utiliser la machine.

Support externe :

Si le client ne possède pas d'équipement pour supporter des systèmes de virtualisation, nous avons une solution. Il existe deux fabricants qui propose leur service sur la location de serveur de virtualisation. Il y a IBM et DELL. Ces deux fabricants proposent des services suffisants à notre solution donc nous avons analysé les offres les plus adapter à notre solution et l'offre que nous recommandons est le PowerEdge R550 du fabricant DELL.

Le PowerEdge R550 dotée de processeurs Intel Xeon Scalable de 3e génération, offre une flexibilité supérieure aux entreprises qui donnent un excellent rapport qualité/prix.
Le powerEdge R550 est conçu pour être peu encombrant.
Il intègre beaucoup de fonctionnalités intéressantes pour le client comme :

- BMC Truesight
- Microsoft System Center
- Red Hat Ansible Modules
- VMware vCenter et vRealize Operations Manager
- IBM Tivoli Netcool/OMNIbus
- IBM Tivoli Network Manager IP Edition
- Micro Focus Operations Manager
- Nagios Core
- Nagios XI
- Firmware signé de manière chiffrée
- Secure Boot
- Secure Erase
- Silicon Root of Trust
- System Lockdown (nécessite iDRAC9 Enterprise ou Datacenter)
- TPM 1.2/2.0 FIPS, CC-TCG certifié, TPM 2.0 Chine NationZ

De plus il possède une sécurité suffisante pour le client.

MariaDB :

MariaDB s'installe simplement avec la commande:

```
apt update && upgrade
apt install mariadb-server -y
```

HeartBeat :

HeartBeat s'installe simplement avec la commande:

```
apt-get install heartbeat -y
```

Une fois heartbeat installer nous allons devoir créer trois fichiers dans le dossier « /etc/ha.d/ »

– ha.cf : Pour la configuration générale de HeartBeat
– haresources : Pour la configuration des ressources
– authkeys : Pour la clef partagé entre les serveurs du cluster

Voici le contenu du fichier ha.cf

```
mcast eth0 239.0.0.10 694 1 0

warntime 4
deadtime 5
initdead 15
keepalive 2

#Re-balance les services sur le node primaire quand il revient en ligne
auto_failback off

#Serveurs du cluster
node bdd1
node bdd2
```

Pensez donc à rajouter les deux lignes suivantes sur le fichier /etc/hosts sur vos deux nodes:

```
##nodes
192.168.2.101    bdd1
192.168.2.102    bdd2
```

Nous allons maintenant créer le contenu du fichier haresources

```
bdd1 IPaddr::192.168.2.100/24/eth0
```

Et pour finir nous allons créer le fichier authkeys. Ce fichier contient une clef partagée entre les deux serveurs. Cela peut être un mot de passe, ou un simple mot. Voici le mien:

```
auth 3
3 md5 my-auth-key
```

(Remplacer « my-auth-key » par un mot de passe de votre choix)

Ce fichier la doit avoir les permissions 600. Donc sur les deux serveurs tapez:

```
chmod 600 /etc/ha.d/authkeys
```

Toutes ces instructions sont à refaire sur la seconde machine.

Et voilà tout est bon maintenant.

Sur bdd1 démarrez Heartbeat avec la commande suivante:

```
/etc/init.d/heartbeat start
```

Vous pouvez aussi vérifier avec la commande ifconfig, vous verrez qu'une nouvelle interface eth0:0 a été créer avec l'adresse IP configuré dans le fichier haresources.

Maintenant sur bdd2 démarrez aussi heartbeat avec la commande:

```
/etc/init.d/heartbeat start
```

Et voilà Heartbeat est opérationnel.

Replication MySQL :

Il faut créer un utilisateur pour la réplication sur le maître. Le nommer "replication" :

```
# mariadb –u root -p
mysql> create user replication@'localhost' identified by 'replication' ;
```

Configuration du serveur MySQL Maître :

Ouvrez une console sur le serveur mysql maître et tapez les commandes suivantes :

```
# mysql –u root -p
mysql> GRANT REPLICATION SLAVE on *.* to 'replication'@'%' identified by
'replication';
```

Modifiez le fichier de configuration /etc/mysql/my.cnf du serveur mySQL maître dans la section [mysqld] :

```
[mysqld]
bind-address  = 0.0.0.0
server-id  = 1
log-bin
```

Redémarrez le serveur mySQL maître:

```
# /etc/init.d/mysql restart
```

Vérification qu'il y ait bien un log binaire :

```
# mariadb –u root -p
mysql> SHOW MASTER STATUS;
```

Configuration du serveur MySQL Esclave :

Configurer le serveur mySQL esclave pour qu'il écoute sur le réseau :
Modifiez le fichier de configuration /etc/mysql/my.cnf du serveur mySQL esclave dans la section [mysqld] :

```
[mysqld]
bind-address  = 0.0.0.0
server-id  = 10
```

Effectuer une sauvegarde sur le maître par la commande suivante :

```
# mysqldump  --master-data --single-transaction --all-databases > dump.sql
#cp dump.sql /tmp/dump.sql
```

Restaurer la sauvegarde sur l'esclave :
Il ne reste plus qu'à restaurer ces données sur le serveur mySQL esclave (et tous les autres esclaves éventuels) :
-Transférer avec la commande scp le fichier dump.sql sur le serveur esclave :
Il faut avoir installé ssh sur les 2 serveurs :

```
# apt-get install ssh
# scp dump.sql identifiant-machine-esclave@adresseip:/tmp/ (sur le serveur maitre)
# mv /tmp/dump.sql /root/ (sur le serveur esclave)
#cd /root/
```

-restaurer les bases de données avec la commande suivante. Attention, cela va écraser toutes les bases de données présentes :

```
# mysql < dump.sql
```

Finaliser la configuration de l'esclave :
Il reste à positionner les variables MASTER_HOST, MASTER_USER, MASTER_PASSWORD :
Sur le serveur mySQL esclave, ouvrez une session console et tapez les commandes suivantes :

```
# mysql –u root -p
mysql> CHANGE MASTER TO MASTER_HOST='192.168.2.101',
MASTER_USER='replication', MASTER_PASSWORD='replication';
```

Redémarrez le serveur mySQL esclave :

```
# /etc/init.d/mysql restart
```

Si nécessaire lancez le serveur mySQL esclave :

```
# mysql –u root -p
mysql> START SLAVE;
```

Pour vérifier que la réplication s'effectue correctement, lancez la commande suivante sur l'esclave :

```
# mysql –u root -p
mysql> SHOW SLAVE STATUS;
```

Il y a 1 information à vérifier :
À présent, les modifications effectuées sur le serveur mySQL maître se retrouveront répliquées sur le serveur mySQL esclave.

Configurer l'accès au partage :

Pour finalisé l'installation complète nous devons pouvoir envoyé une backup directement sur le serveur de partage.

Pour cela nous allons bloquer tous les ports de la machines et seulement ouvrir le port 22 (SSH) pour administrer la machine et le port 445 (SMB).

Cela permet de sécurisé la machine et permettre uniquement des communications en SSH et SMB.

Nous allons installer ufw, un logiciel qui permet d'ouvrir ou bloquer les ports.

```
apt install smbclient -y
apt install ufw -y
ufw enable
ufw default deny incoming
ufw allow 445
```

Nous avons bloquer tous les ports et nous avons uniquement autoriser le port 445 pour smbclient.

Backup automatique :

Installation à faire uniquement sur le serveur Maitre, donc BDD1.

Nous allons mettre en place un script permettant d'exporter les bdd automatiquement avec les commandes suivante :

```
# cd /etc/cron.daily/
#mkdir /etc/backup
# touch /etc/cron.daily/backup.sh
# chmod 755 /etc/cron.daily/backup.sh
# nano /etc/cron.daily/backup.sh
```

```
#!/bin/sh
cd /etc/backup && rm *.gz
now="$(date +'%d_%m_%Y_%H_%M_%S')"
filename="db_backup_$now".gz
backupfolder="/etc/backup"
fullpathbackupfile="$backupfolder/$filename"
logfile="$backupfolder/"backup_log_"$(date +'%Y_%m')".txt
echo "mysqldump started at $(date +'%d-%m-%Y %H:%M:%S')" >> "$logfile"
mysqldump --user=root --password=root --default-character-set=utf8 –all-databses |
gzip > "$fullpathbackupfile"
echo "mysqldump finished at $(date +'%d-%m-%Y %H:%M:%S')" >> "$logfile"
chown root "$fullpathbackupfile"
chown root "$logfile"
echo "file permission changed" >> "$logfile"
find "$backupfolder" -name db_backup_* -mtime +8 -exec rm {} \;
echo "old files deleted" >> "$logfile"
echo "operation finished at $(date +'%d-%m-%Y %H:%M:%S')" >> "$logfile"
echo "**********" >> "$logfile"
smbclient //backup10/backup10 @zerty2QWERTY! -U administrateur -
p@zerty2QWERTY! -c 'prompt OFF; recurse ON; cd //backup10/backup10; lcd
/etc/backup/; mput *'

exit 0
```

Changer la ligne mysqldump avec vos identifiant mysql.
Changer la ligne smbclient en modifiant les identifiant par les votre et //backup10/backup10 par votre dossier partagé.

Ensuite nous configurons l'exécution automatique du script chaque jour à 12 heures.

```
chmod +x backup.sh
crontab -e
0 12 * * * usr/bin/sh /etc/cron.daily/backup.sh
```