

HA-CLUSTER

Principes

HA-Cluster (high-availability cluster) est un gestionnaire de *grappes* sous Linux qui permet d'assurer les messages de synchronisation entre machines participant au **cluster** grâce à l'outil **heartbeat** et d'offrir des fonctions de partage de ressources grâce à un gestionnaire dédié (comme par exemple l'outil **Pacemaker**, non traité ici).

Nous traiterons ici uniquement du paramétrage de l'outil **heartbeat** qui permet du **clustering** en **mode actif/passif**.

Le **cluster** Heartbeat assure une tolérance de panne en assurant la bascule automatique vers un autre serveur disposant des mêmes services et données

Principe

- Deux (ou plusieurs) serveurs participent ensemble au **cluster**. Ils possèdent les mêmes services et données (par exemple une application Web)
- Un serveur est **actif** à un instant donné (c'est à dire au contact des utilisateurs), l'autre est en fonction mais pas accessible aux utilisateurs (il est **passif**).
- Le **cluster** possède une **IP virtuelle** qui est connue des utilisateurs : elle est présente sur le serveur **actif**
- les serveurs s'échangent un signal permanent (par exemple un ping) comme un **battement de coeur / heartbeat** ** pour signaler leur activité * Si le signal est perdu, le serveur **passif active l'**ip virtuelle** du **cluster** sur sa carte réseau (il devient donc **actif**)

Installation du service

L'outil **heartbeat** fait partie des paquetages supportés par *Debian* et *Ubuntu*. Après une mise à jour (*update/upgrade*) de la machine, on procèdera à l'installation comme suit :

```
apt install heartbeat
```

Configuration du service

Procédure

La configuration doit être identique sur tous les membres du cluster. On pourra configurer une machine et recopier les fichiers sur l'autre.

La démarche est la suivante :

1. Installer le service -
2. Enregistrer les associations [nom des nœuds / IP] dans le fichier **//
/etc/hosts//**
3. se placer dans le dossier de configuration **//cd /etc/heartbeat/ //**
4. Créer et configurer le fichier **ha.cf**
5. Créer et configurer le fichier **haresources**
6. Créer et configurer le fichier **authkeys** et en restreindre l'accès
7. Répliquer la configuration sur les autres membres (en pensant à adapter le fichier **hosts** sur chaque machine)
8. Relancer le service **heartbeat**

Les fichiers

La configuration de **heartbeat** passe par le paramétrage de 3 fichiers (présents dans le dossier ****
/etc/heartbeat/** ou **/etc/ha.d/****). Ces fichiers peuvent être à créer.

- **ha.cf** : il définit la liste des équipements (on parle de **nœuds** ou **node**) concernés par le cluster et le mode de gestion de la reprise sur incident
- **haresources** : il indique **dans l'ordre** :
 - la machine étant *maître* dans le cluster
 - l'adresse IP virtuelle active sur le nœud actif avec son masque
 - le nom de la carte réseau virtuelle
 - le ou les services activés dans le cluster.
- **authkeys** : il définit les éléments de sécurité permettant aux membres du cluster d'entrer en communication

Interventions dans les fichiers

Fichier **/etc/hosts**

Le nom des nœuds à enregistrer dans la clause **node** et dans le fichier **hosts** est celui de la machine que l'on peut consulter en tapant la commande sur chaque nœud :

```
hostname
```

Les nœuds participant au cluster doivent tous être renseignés dans le fichier **/etc/hosts** avec l'adresse IP de la carte réseau sur laquelle ils seront contactés.

```
nano /etc/hosts
```

On ne supprimera pas les informations existantes dans ce fichier. La syntaxe ressemblera à cela

```
....
# à ajouter après les lignes existantes
<ip maître> <noeudMaître>
<ip secondaire> <noeudSecondaire>
```

Exemple

```
*****Penser à adapter les valeurs exemple *****
172.28.10.110 B3-WEB110
172.28.10.111 B3-WEB111
```

Pour la suite, on se placera dans le dossier de Heartbeat où tous les fichiers doivent être créés.

```
cd /etc/heartbeat
```

Syntaxe du fichier ha.cf

Aide Paramétrage:

<https://blog.foulquier.info/tutoriels/systeme/mise-en-place-dun-cluster-heartbeat-apache>

```
# mode de gestion des journaux d'activité
logfacility      local0
# temps entre deux interrogations secondaire/maître
keepalive       2
# temps au bout duquel le maître ou le secondaire est considéré hors jeu
deadtime        10
# mode de synchronisation maître/secondaire (ici en broadcast sur la carte
eth0)
bcast           eth0
# liste des noeuds participant au cluster (l'ordre dans lequel on les place
définit la hiérarchie)
node            <noeudMaître> <noeudSecondaire>
# mode de gestion du retour à la normale
auto_failback   off
# commandes pour gérer les actions en cas de défaillance
respawn         hacluster /usr/libexec/heartbeat/ipfail
apiauth         ipfail gid=haclient uid=hacluster
```

Exemple pour la ligne **node** :

```
node B3-WEB110 B3-WEB111
```

Syntaxe du fichier haresources

```
<noeudMaître>
IPaddr::<ip_virtuelle_cluster>/<masque>/<carte_reseau>:<numVirtuel>
```

```
<service(s)_à_inclure>
```

Le fichier **haresources** définit les éléments suivants :

- **<noeudMaitre>** : c'est le nom de la machine qui joue le rôle principal. Le nom doit être celui de la machine et être enregistré dans le fichier **/etc/hosts**
- **<ip_virtuelle_cluster>/<masque>** : il s'agit d'une adresse IP (et de son masque) active sur l'un des *noeuds* (le maître s'il est opérationnel) par lequel les postes consultent le ou les services du cluster
- **<carte_reseau>** : nom de la carte réseau qui possèdera l'adresse virtuelle (par exemple *eth0*)
- **<numVirtuel>** : numéro de l'interface virtuelle qui possèdera l'IP virtuelle du *cluster* qui apparaîtra sous la forme *eth0:1* par exemple
- **<service(s)_a_inclure>** : **heartbeat** peut gérer de multiples services ; on les notera à la suite, séparés par des espaces

Exemple (valeurs à adapter) :

```
B3-WEB110 IPaddr::172.30.0.201/24/eth0:0 apache2 mysql
```

Syntaxe du fichier authkeys

Le fichier **authkeys** contient les informations de sécurité permettant aux *noeuds* de se synchroniser. Il doit donc être protégé contre un accès à des utilisateurs autres que *root*.

Contenu du fichier

```
auth <num_ligne>  
<num_ligne> <algorithme> <passphrase>
```

L'<algorithme> pourra être **md5** (mal sécurisé) ou **sha1**.

Exemple (changer le mot de passe) :

```
auth 2  
2 sha1 textCQr1T3
```

Paramétrage des droits sur le fichier authkeys

On modifiera les droits d'accès pour limiter la lecture du contenu au compte *root*.

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

From:

<https://wiki.sio.bts/> - **WIKI SIO : DEPUIS 2017**

Permanent link:

<https://wiki.sio.bts/doku.php?id=ha>

Last update: **2025/06/01 12:23**

