

Labo 3 :Charlène, Mélina, Alexandre

Haute disponibilité avec Heartbeat

SISR 3

The logo for Heartbeat, featuring the word "HEARTBEAT" in a stylized, outlined font. The letters 'H', 'A', 'R', 'T', 'B', and 'E' are black, while the letters 'A', 'E', and 'A' are red. The 'A's and 'E's are designed to look like stylized hearts or pulses.

2014/2015

Sommaire

I - Introduction.....	3
II - Installation.....	3
III - Configuration.....	4
1- /etc/heartbeat/ha.cf	4
2 - /etc/heartbeat/authkeys.....	6
3 - /etc/heartbeat/haresources	6
IV - Démarrage du programme	7
1- Démarrage des différents services.....	7
2- Vérification de la configuration.....	7
3- Vérification du lancement des services.....	8
4- Simulation d'une panne	9
V- Pour aller plus loin... ..	9

I - Introduction

Heartbeat est un logiciel de surveillance de la disponibilité des programmes, pour les systèmes d'exploitation Linux, FreeBSD, OpenBSD, Solaris et MacOS X. Il est distribué sous licence GPL.

Heartbeat écoute les battements de cœur – des signaux émis par les services d'une grappe de serveurs lorsqu'ils sont opérationnels. Lorsque qu'un serveur devient défaillant, Heartbeat le détecte (puisqu'il n'entend plus ses battements de cœurs) et bascule les services surveillés sur un autre serveur. Pour que cela soit transparent pour les utilisateurs, Heartbeat met en place une IP virtuelle unique qui est balancée entre les deux serveurs.

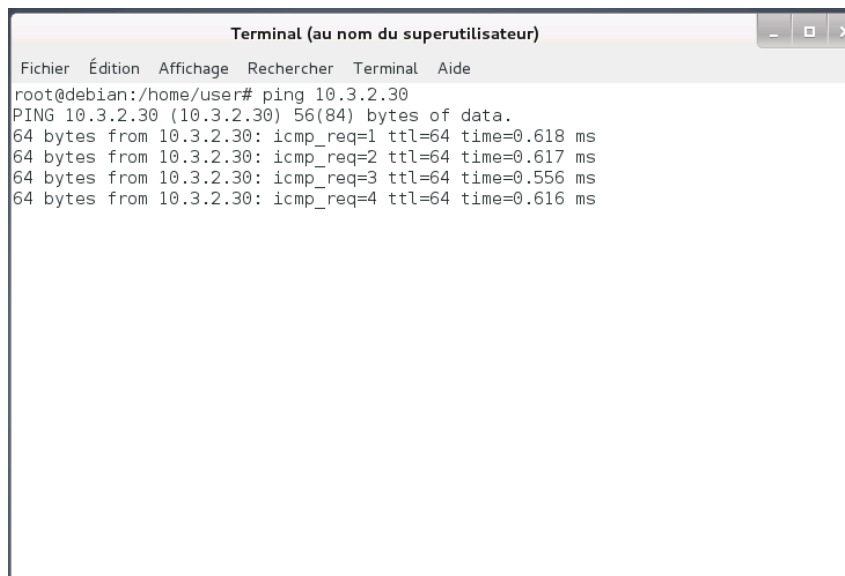
II - Installation

Pour installer ce type de service, nous avons installés deux machines virtuelles sous Debian. Une des deux machines sera le serveur actif et la seconde le serveur passif (backup- failover). Pour chaque machine, nous lançons donc la commande "**apt-get update**" pour rechercher les potentiels mises à jour. Nous procédons ensuite à l'installation du logiciel Heartbeat en lançant la commande "**apt-get install Heartbeat**".



```
Terminal (au nom du superutilisateur)
Fichier Édition Affichage Rechercher Terminal Aide
root@debian:/home/user# apt-get install heartbeat
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
 cluster-glue gawk libcfg4 libcib1 libconfdb4 libcoroipcc4 libcoroipcs4
 libcorosync4 libcp4 libcrmcluster1 libcrmcommon2 libcurl3 libesntp6 libeivs4
 libheartbeat2 liblogsys4 liblrm2 libnet1 libnspr4-0d libopenhpi2
 libopenipmi0 libpe-rules2 libpe-status3 libpengine3 libpils2 libpload4
 libplumb2 libplumbgp2 libquorum4 libsam4 libsigsegv2 libstonith1
 libstonithd1 libtotem-pg4 libtransitioner1 libvotequorum4 libxml2-utils
 openhpid pacemaker resource-agents
Paquets suggérés :
 gawk-doc
Les NOUVEAUX paquets suivants seront installés :
 cluster-glue gawk heartbeat libcfg4 libcib1 libconfdb4 libcoroipcc4
 libcoroipcs4 libcorosync4 libcp4 libcrmcluster1 libcrmcommon2 libcurl3
 libesntp6 libeivs4 libheartbeat2 liblogsys4 liblrm2 libnet1 libnspr4-0d
 libopenhpi2 libopenipmi0 libpe-rules2 libpe-status3 libpengine3 libpils2
 libpload4 libplumb2 libplumbgp2 libquorum4 libsam4 libsigsegv2 libstonith1
 libstonithd1 libtotem-pg4 libtransitioner1 libvotequorum4 libxml2-utils
 openhpid pacemaker resource-agents
0 mis à jour, 41 nouvellement installés, 0 à enlever et 12 non mis à jour.
Il est nécessaire de prendre 7 648 ko dans les archives.
```

Nous testons ensuite la connectivité entre les deux serveurs grâce à une requête Ping, les deux serveurs sont donc dans le même sous réseau. Il est également nécessaire d'installer le paquet "**DRDB**" qui permet la réplication entre deux serveurs.

A terminal window titled "Terminal (au nom du superutilisateur)" with a menu bar containing "Fichier", "Édition", "Affichage", "Rechercher", "Terminal", and "Aide". The terminal shows the command "ping 10.3.2.30" and its output: "PING 10.3.2.30 (10.3.2.30) 56(84) bytes of data.", "64 bytes from 10.3.2.30: icmp_req=1 ttl=64 time=0.618 ms", "64 bytes from 10.3.2.30: icmp_req=2 ttl=64 time=0.617 ms", "64 bytes from 10.3.2.30: icmp_req=3 ttl=64 time=0.556 ms", and "64 bytes from 10.3.2.30: icmp_req=4 ttl=64 time=0.616 ms".

```
Terminal (au nom du superutilisateur)
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
root@debian:/home/user# ping 10.3.2.30
PING 10.3.2.30 (10.3.2.30) 56(84) bytes of data.
64 bytes from 10.3.2.30: icmp_req=1 ttl=64 time=0.618 ms
64 bytes from 10.3.2.30: icmp_req=2 ttl=64 time=0.617 ms
64 bytes from 10.3.2.30: icmp_req=3 ttl=64 time=0.556 ms
64 bytes from 10.3.2.30: icmp_req=4 ttl=64 time=0.616 ms
```

III - Configuration

Pour faire fonctionner Heartbeat, il faut créer 3 fichiers de configuration, dans le dossier **/etc/heartbeat**. Avant toute création de fichiers, il faut modifier le nom des deux machines en allant dans le fichier **"/etc/hostname"**.

1- **/etc/heartbeat/ha.cf**

Ce fichier détermine la liste des machines à utiliser et la manière de dialoguer entre elles. Le fichier n'existant pas nous le créons à l'aide de la commande **"nano /etc/heartbeat/ha.cf"**. La fenêtre du fichier en question s'ouvre, nous déposons alors les informations suivantes dans le fichier :

```
*ha.cf [Lecture seule] (/etc/heartbeat) - gedit
Fichier  Édition  Affichage  Rechercher  Outils  Documents  Aide
Ouvrir  Enregistrer  Annuler
*ha.cf x
slogfile /var/log/heartbeat.log
logfacility local0
node debian1
node debian2
keepalive 1
deadtime 10
bcast eth0
udpport 694
ping 10.3.2.1
auto_failback on
initdead 120
Texte brut  Largeur des tabulations: 8  Lig 1, Col 2  INS
```

Pour plus de détails nous expliquons à quoi sert chaque ligne du fichier créer.

Logfile est un fichier où loguer les évènements relatifs à heartbeat (lancement, arrêt, etc.).

Logfacility indique quelle "facility" le syslog devra utiliser pour loguer les évènements. Les valeurs peuvent changer selon les systèmes, mais local0 reste une valeur sûre.

Keepalive est une intervalle entre 2 battements de cœur.

Deadtime est le temps nécessaire avant de considérer qu'un nœud est mort.

Bcast spécifie l'interface réseau utilisée par Heartbeat pour envoyer les battements de cœur (en UDP).

Node liste des machines utilisées pour la haute disponibilité, séparées par des espaces.

Auto failback est le comportement à adopter si la machine en panne revient sur le réseau. Si le paramètre est à 'Off', elle se met simplement en attente. Avec 'On', elle redevient la machine active, et celle qui fonctionne à l'heure actuelle repasse en passive.

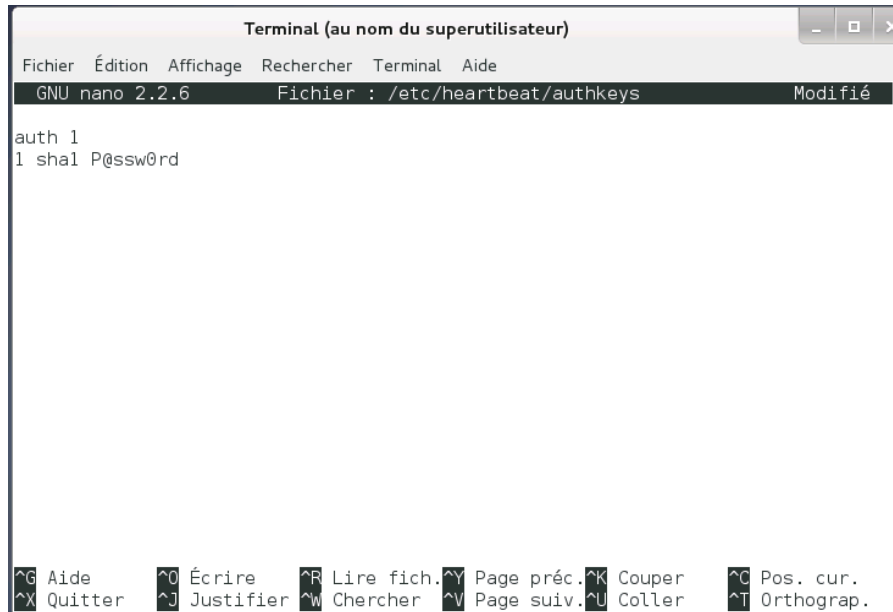
Respawn permet de lancer un programme au démarrage de heartbeat, qui sera tout le temps actif (Le premier paramètre est l'id de l'utilisateur qui lancera le programme. Le second, le programme lui-même).

On instancie ici ipfail, qui est à peu près le seul logiciel intéressant à lancer. Il permet d'accélérer la détection d'erreur, en regardant la disponibilité des liens réseaux en plus d'attendre un certain nombre de battements de cœur. Ceci permet au serveur passif de prendre la main dès qu'une connexion réseau est coupée, au lieu d'attendre la non réception de X battements de cœur.

Apiauth(Utilisé conjointement avec respawn) indique quels groupes et utilisateurs ont le droit de dialoguer avec les programmes lancés par respawn. Ces programmes étant normalement utilisés pour des besoins internes, il semble logique d'interdire les accès extérieurs.

2 - /etc/heartbeat/authkeys

Nous créons comme précédemment le fichier "**authkeys**" et nous inscrivons les informations comme ci dessous :



```
Terminal (au nom du superutilisateur)
Fichier  Édition  Affichage  Recherche  Terminal  Aide
GNU nano 2.2.6  Fichier : /etc/heartbeat/authkeys  Modifié
auth 1
1 sha1 P@ssw0rd
^G Aide      ^O Écrire    ^R Lire fich.^Y Page préc.^K Couper    ^C Pos. cur.
^X Quitter   ^J Justifier ^W Chercher  ^V Page suiv.^U Coller    ^T Orthograp.
```

On peut souligner également qu'il existe trois modes d'authentification :

- *crc* (réseaux sécurisés, comme un câble croisé par exemple)
- *md5* (bonne alternative de sécurité)
- *sha1* (méthode la plus sûre)

Il faut également penser à protéger ce fichier pour qu'il ne soit plus visible par n'importe qui (si vous ne le faites pas, le programme ne se lancera pas) grâce à la commande suivante "**chmod 600 /etc/heartbeat/authkeys**".

```
root@debian:/home/user# nano /etc/heartbeat/ha.cf
root@debian:/home/user# nano /etc/heartbeat/authkeys
root@debian:/home/user# chmod 600 /etc/heartbeat/authkeys
root@debian:/home/user# █
```

3 - /etc/heartbeat/haresources

Comme pour les deux fichiers suivants, nous créons le fichier "**haresources**" et nous inscrivons les informations comme ci dessous en indiquant le nom du serveur maitre et l'adresse IP virtuelle du cluster :

```
haresources (/etc/heartbeat) - gedit
Fichier  Édition  Affichage  Rechercher  Outils  Documents  Aide
Ouvrir  Enregistrer  Annuler
haresources x
debian1 IPaddr::10.3.2.50/8/eth0:0 apache2
Texte brut  Largeur des tabulations: 8  Lig 1, Col 1  INS
```

IV - Démarrage du programme

1- Démarrage des différents services

Pour pouvoir lancer le service Heartbeat, le service apache2 doit être lancer. Pour cela nous lançons la commande "**service apache2 start**". Une fois le service apache lancer, nous lançons également le service Heartbeat avec la commande "**service Heartbeat start**".

2- Vérification de la configuration

Nous vérifions que l'activation de l'adresse IP virtuelle s'est faite correctement. Pour cela, nous lançons la commande "**ifconfig**" sur le serveur maître. Nous constatons qu'une nouvelle interface est apparue avec ":0" à la fin, c'est à dire que notre interface virtuelle s'appelle "eth0:0" comme ci dessous.

```

Terminal (au nom du superutilisateur)
Fichier Édition Affichage Rechercher Terminal Aide
CRITICAL: Non-idle resources can affect data integrity!
info: If you don't know what this means, then get help!
info: Read the docs and/or the source to /usr/share/heartbeat/ResourceManager for more details
.
ResourceManager[5678]: CRITICAL: Non-idle resources will affect resource takeback!
ResourceManager[5678]: CRITICAL: Non-idle resources may affect data integrity!
Done.

root@debian1:/home/user# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:15:5d:dc:21:0a
          inet  adr:10.3.2.30  Bcast:10.255.255.255  Masque:255.0.0.0
          adr inet6: fe80::215:5dff:fedc:210a/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:23978 errors:0 dropped:0 overruns:0 frame:0
          TX packets:4017 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:3414261 (3.2 MiB)  TX bytes:623711 (609.0 KiB)

eth0:0    Link encap:Ethernet  HWaddr 00:15:5d:dc:21:0a
          inet  adr:10.3.2.50  Bcast:10.255.255.255  Masque:255.0.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

lo        Link encap:Boucle locale
          inet  adr:127.0.0.1  Masque:255.0.0.0
          adr inet6: ::1/128 Scope:Hôte
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:2030 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2030 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          RX bytes:357162 (348.7 KiB)  TX bytes:357162 (348.7 KiB)

```

3- Vérification du lancement des services

Nous vérifions ensuite que les services se sont lancés correctement grâce à la commande "**ps aws**". Nous constatons bien que le service apache2 ainsi que le service Heartbeat fonctionne bien.

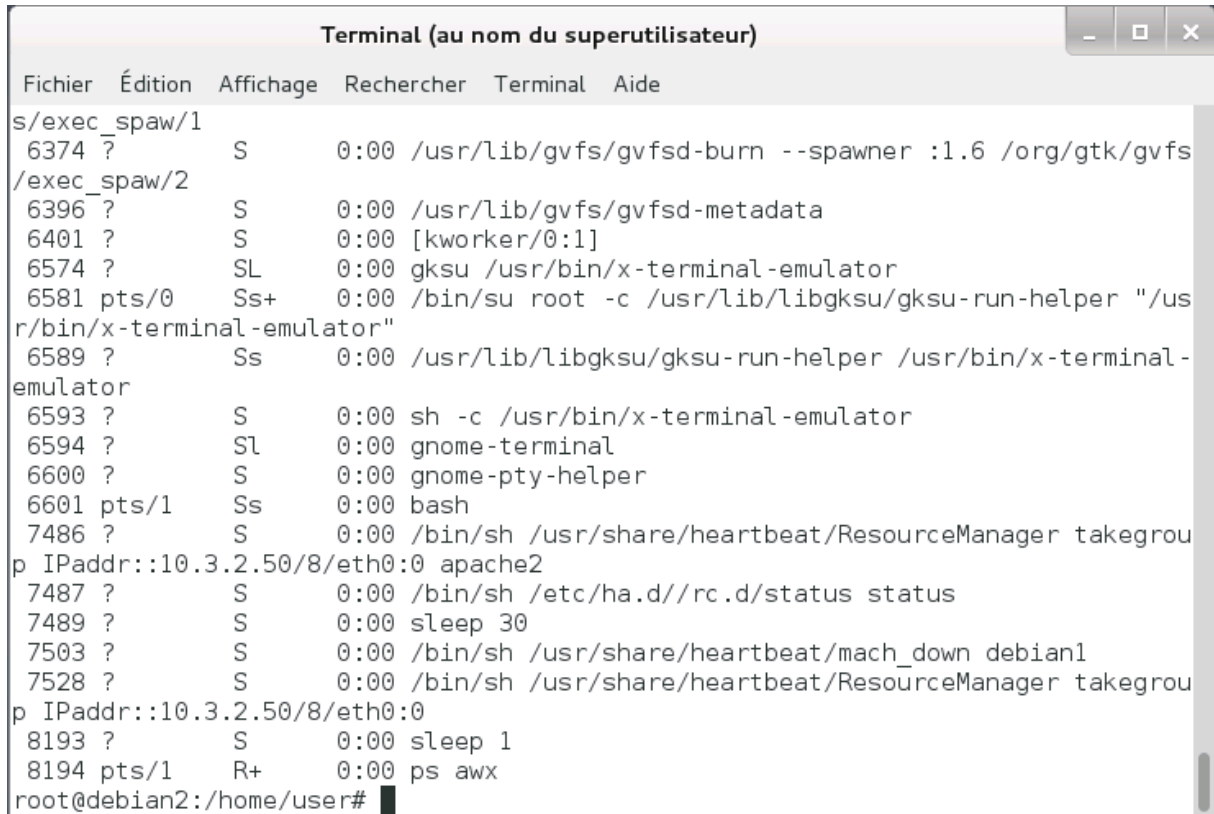
```

Terminal (au nom du superutilisateur)
Fichier Édition Affichage Rechercher Terminal Aide
3634 ?      S      0:00 /usr/lib/x86_64-linux-gnu/gconf/gconfd-2
3636 ?      S      0:00 gnome-pty-helper
3637 pts/2    Ss     0:00 bash
3749 ?      SLs    0:02 heartbeat: master control process
3754 ?      SL     0:00 heartbeat: FIFO reader
3755 ?      SL     0:00 heartbeat: write: bcast eth0
3756 ?      SL     0:00 heartbeat: read: bcast eth0
3757 ?      SL     0:00 heartbeat: write: ping 10.3.2.1
3758 ?      SL     0:00 heartbeat: read: ping 10.3.2.1
3778 ?      Ss     0:00 /usr/sbin/exim4 -bd -q30m
3921 ?      Ss     0:00 /usr/sbin/winbindd
3951 ?      S      0:00 /usr/sbin/winbindd
3958 ?      S      0:00 /usr/sbin/xrdp
3961 ?      S      0:00 /usr/sbin/xrdp-sesman
3987 ?      Ss     0:00 /usr/sbin/dovecot -c /etc/dovecot/dovecot.conf
4011 ?      Ss     0:00 /usr/sbin/minissdpsd -i 0.0.0.0
4019 tty1    Ss+    0:00 /sbin/getty 38400 tty1
4020 tty2    Ss+    0:00 /sbin/getty 38400 tty2
4021 tty3    Ss+    0:00 /sbin/getty 38400 tty3
4022 tty4    Ss+    0:00 /sbin/getty 38400 tty4
4023 tty5    Ss+    0:00 /sbin/getty 38400 tty5
4024 tty6    Ss+    0:00 /sbin/getty 38400 tty6
4025 ?      S      0:00 dovecot/anvil
4026 ?      S      0:00 dovecot/log
4028 ?      S      0:00 dovecot/config
5380 ?      Ss     0:00 /usr/sbin/apache2 -k start
5382 ?      S      0:00 /usr/sbin/apache2 -k start
5385 ?      SL     0:00 /usr/sbin/apache2 -k start
5386 ?      SL     0:00 /usr/sbin/apache2 -k start
5830 ?      S      0:00 [kworker/0:0]
5831 pts/2    R+     0:00 ps aws

```


4- Simulation d'une panne

Pour vérifier le nœud entre le serveur maître et le serveur esclave, nous simulons une panne sur le serveur maître. Nous effectuons ensuite la commande "**ps aws**" sur le serveur esclave pour vérifier qu'il prends bien le relais. Le serveur esclave constate bien que le serveur maître n'est plus actif "**mach_down debian1**" et démarre donc les services lancés sur le serveur maître.



```
Terminal (au nom du superutilisateur)
Fichier  Édition  Affichage  Rechercher  Terminal  Aide
s/exec_spaw/1
6374 ?      S        0:00 /usr/lib/gvfs/gvfsd-burn --spawner :1.6 /org/gtk/gvfs
/exec_spaw/2
6396 ?      S        0:00 /usr/lib/gvfs/gvfsd-metadata
6401 ?      S        0:00 [kworker/0:1]
6574 ?      SL       0:00 gksu /usr/bin/x-terminal-emulator
6581 pts/0    Ss+     0:00 /bin/su root -c /usr/lib/libgksu/gksu-run-helper "/usr/bin/x-terminal-emulator"
6589 ?      Ss      0:00 /usr/lib/libgksu/gksu-run-helper /usr/bin/x-terminal-emulator
6593 ?      S        0:00 sh -c /usr/bin/x-terminal-emulator
6594 ?      SL       0:00 gnome-terminal
6600 ?      S        0:00 gnome-pty-helper
6601 pts/1    Ss      0:00 bash
7486 ?      S        0:00 /bin/sh /usr/share/heartbeat/ResourceManager takegroup IPaddr::10.3.2.50/8/eth0:0 apache2
7487 ?      S        0:00 /bin/sh /etc/ha.d//rc.d/status status
7489 ?      S        0:00 sleep 30
7503 ?      S        0:00 /bin/sh /usr/share/heartbeat/mach_down debian1
7528 ?      S        0:00 /bin/sh /usr/share/heartbeat/ResourceManager takegroup IPaddr::10.3.2.50/8/eth0:0
8193 ?      S        0:00 sleep 1
8194 pts/1    R+     0:00 ps aws
root@debian2:/home/user#
```

V- Pour aller plus loin...

Il existe d'autres logiciels pour assurer un service de continuité comme safekit pour Windows et Linux et SCVMM pour Windows. Heartbeat reste le logiciel le plus utilisé pour ce genre de service.