

**BTS SIO 1B**

## **Projet HAProxy**

**Haute disponibilité et équilibrage de charge avec  
HeartBeat sous Debian Bullseye**



# Sommaire

## I . Contexte & Introduction

## II . Topologie

## III . Prérequis

## IV . Installation & Configuration

- a) HAProxy
- b) HeartBeat
- c) ownCloud

## V . Validation

# I . Contexte & Introduction

## Contexte :

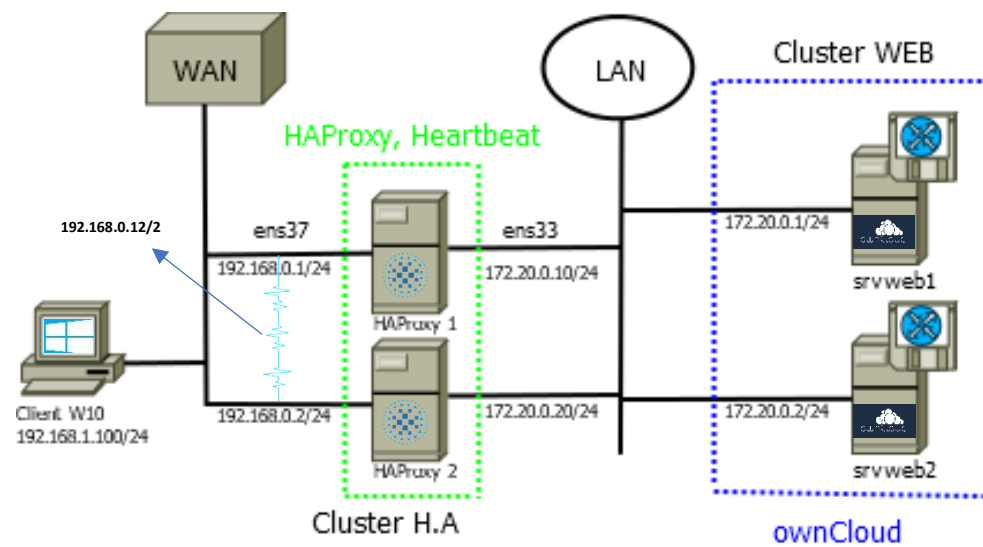
L'installation du cluster web avec l'utilisation du service ownCloud (logiciel offrant une plateforme de stockage et partage de fichiers) a été effectuée. Il est maintenant nécessaire d'installer de nouveaux services pour permettre d'assurer une haute disponibilité ce qui signifie concrètement d'assurer un service pour une entreprise en évitant au maximum les indisponibilités. Nous utiliserons HAProxy ainsi que le logiciel Heartbeat pour réaliser cette mission.

## Introductions :

HAProxy est un logiciel gratuit et open source qui fournit un équilibreur de charge haute disponibilité et un proxy pour les applications TCP et http. L'équilibrage de charge (load balancing en anglais), est une technique utilisée en informatique pour distribuer un travail entre plusieurs processeurs, ordinateurs, disques ou dans notre cas des serveurs. Cette équilibrage des charges s'applique en particulier au domaine des connexions réseau, où elle permet d'assurer la haute disponibilité d'applications et de sites web.

Heartbeat est un logiciel de surveillance de la disponibilité des programmes. Ce logiciel permet d'avoir une écoute sur les battements de cœur des signaux émis par les services d'une grappe de serveurs lorsqu'ils sont opérationnels. Lorsqu'une machine tombe, il exécute des scripts d'initialisations (plus d'attente du battement de cœur) ou est à nouveau disponible. Il permet également de changer une adresse IP entre les machines à l'aide de mécanisme ARP (Address Resolution Protocol).

## II . Topologie



### III . Prérequis

Pour les installations et configurations nous aurons besoin de 4 machines Debian 11, et une machine cliente Windows 10 :

- ❖ Pour le cluster web, une machine srvweb1 et une machine srvweb2
- ❖ Pour le cluster HAProxy, une machine HAProxy 1 et une machine HAProxy 2
- ❖ Pour la validation, une machine cliente Windows 10

### IV . Installation & Configuration

#### a) HAProxy

```
root@Master:~#  
root@Master:~#  
root@Master:~# apt install haproxy  
Lecture des listes de paquets... Fait  
Construction de l'arbre des dépendances... Fait  
Lecture des informations d'état... Fait  
Les paquets supplémentaires suivants seront installés :  
  liblua5.3-0  
Paquets suggérés :  
  vim-haproxy haproxy-doc  
Les NOUVEAUX paquets suivants seront installés :  
  haproxy liblua5.3-0  
0 mis à jour, 2 nouvellement installés, 0 à enlever et 34 non mis à jour.  
Il est nécessaire de prendre 1 899 ko/2 019 ko dans les archives.  
Après cette opération, 4 316 ko d'espace disque supplémentaires seront utilisés.  
Souhaitez-vous continuer ? [O/n] o  
Réception de :1 http://deb.debian.org/debian bullseye/main amd64 haproxy amd64 2.2.9-2+deb11u3 [1 89  
9 kB]  
1 899 ko réceptionnés en 0s (7 968 ko/s)  
Sélection du paquet liblua5.3-0:amd64 précédemment désélectionné.  
(Lecture de la base de données... 24381 fichiers et répertoires déjà installés.)  
Préparation du dépaquetage de .../liblua5.3-0_5.3.3-1.1+b1_amd64.deb ...  
Dépaquetage de liblua5.3-0:amd64 (5.3.3-1.1+b1) ...  
Sélection du paquet haproxy précédemment désélectionné.  
Préparation du dépaquetage de .../haproxy_2.2.9-2+deb11u3_amd64.deb ...  
Dépaquetage de haproxy (2.2.9-2+deb11u3) ...  
Paramétrage de liblua5.3-0:amd64 (5.3.3-1.1+b1) ...  
Paramétrage de haproxy (2.2.9-2+deb11u3) ...  
Created symlink /etc/systemd/system/multi-user.target.wants/haproxy.service → /lib/systemd/system/ha  
proxy.service.  
Traitement des actions différées (« triggers ») pour rsyslog (8.2102.0-2) ...  
Traitement des actions différées (« triggers ») pour libc-bin (2.31-13+deb11u2) ...  
root@Master:~#
```

- 1) On commence par installer HAProxy à l'aide de la commande « apt install haproxy »

```

GNU nano 5.4 /etc/network/interfaces *
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug ens33
iface ens33 inet static
address 172.20.0.10/24

allow-hotplug ens37
iface ens37 inet static
address 192.168.0.1/24

```

- 2) On configure les cartes réseaux de la machine (ens33 et ens37) à l'aide de la commande « nano /etc/network/interfaces »

```

root@Master:~# ifdown ens33
root@Master:~# ifdown ens37
root@Master:~# ifup ens33
root@Master:~# ifup ens37
root@Master:~# _

```

- 3) On redémarre les cartes réseaux en les éteignant à l'aide de la commande « ifdown », puis on les rallume à l'aide de la commande « ifup ».

```

root@Master:~#
root@Master:~#
root@Master:~# ifconfig -a
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.20.0.10 netmask 255.255.255.0 broadcast 172.20.0.255
    inet6 fe80::20c:29ff:fe90:e437 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:90:e4:37 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 21 bytes 1662 (1.6 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

ens37: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.0.1 netmask 255.255.255.0 broadcast 192.168.0.255
    inet6 fe80::20c:29ff:fe90:e441 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:90:e4:41 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 28 bytes 2248 (2.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Boucle locale)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

root@Master:~#

```

- 4) On vérifie la configuration des adresses IP a bien été effectué à l'aide de la commande « ifconfig -a »

```

root@Master:~#
root@Master:~#
root@Master:~# nano /etc/haproxy/haproxy.cfg

```

- 5) On va maintenant configurer le service HAProxy en ouvrant le fichier de configuration à l'aide de la commande « nano /etc/haproxy/haproxy.cfg »

```

# Configuration du balancement
listen clusterWebHA
bind 192.168.0.1:80

# Mode écoute
mode http

#Mode du balancement (roundrobin (50%-50%))
balance roundrobin

# Option
option httpclose
option forwardfor

# Liste des serveurs impliqués par le balancement
server srvWeb1 172.20.0.1:80 check
server srvWeb2 172.20.0.2:80 check

# Pour les statistiques
stats enable
stats hide-version
stats refresh 30s
stats show-node
stats auth admin:password
stats uri /statistique

```

- 6) On ajoute les lignes présentes sur le screen, puis on enregistre le fichier.

```

root@Master:~#
root@Master:~#
root@Master:~# service haproxy restart
root@Master:~# service haproxy status
• haproxy.service - HAProxy Load Balancer
  Loaded: loaded (/lib/systemd/system/haproxy.service; enabled; vendor preset: enabled)
  Active: active (running) since Thu 2022-03-31 14:56:35 CEST; 7s ago
    Docs: man:haproxy(1)
           file:/usr/share/doc/haproxy/configuration.txt.gz
  Process: 653 ExecStartPre=/usr/sbin/haproxy -f $CONFIG -c -q $EXTRA_OPTS (code=exited, status=0/0)
 Main PID: 655 (haproxy)
   Tasks: 2 (limit: 2301)
  Memory: 36.1M
    CPU: 70ms
  CGroup: /system.slice/haproxy.service
          └─655 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -S /run/hap
            657 /usr/sbin/haproxy -Ws -f /etc/haproxy/haproxy.cfg -p /run/haproxy.pid -S /run/hap

mars 31 14:56:35 Master systemd[1]: Starting HAProxy Load Balancer...
mars 31 14:56:35 Master haproxy[655]: Proxy clusterWebHA started.
mars 31 14:56:35 Master haproxy[655]: Proxy clusterWebHA started.
mars 31 14:56:35 Master haproxy[655]: [NOTICE] 089/145635 (655) : New worker #1 (657) forked
mars 31 14:56:35 Master systemd[1]: Started HAProxy Load Balancer.
mars 31 14:56:37 Master haproxy[657]: [WARNING] 089/145637 (657) : Server clusterWebHA/srvWeb1 is D
mars 31 14:56:38 Master haproxy[657]: [WARNING] 089/145638 (657) : Server clusterWebHA/srvWeb2 is D
mars 31 14:56:38 Master haproxy[657]: [NOTICE] 089/145638 (657) : haproxy version is 2.2.9-2+deb11u3
mars 31 14:56:38 Master haproxy[657]: [NOTICE] 089/145638 (657) : path to executable is /usr/sbin/h
mars 31 14:56:38 Master haproxy[657]: [ALERT] 089/145638 (657) : proxy 'clusterWebHA' has no server
lines 1-24/24 (END)

```

- 7) On redémarre le service HAProxy à l'aide de la commande « service haproxy restart », puis on vérifie que le service est bien actif à l'aide de la commande « service haproxy status ». On peut voir que le service indique « Active : active (running) » le service est donc bien fonctionnel.

```
GNU nano 5.4 /etc/network/interfaces
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug ens33
iface ens33 inet static
address 172.20.0.20/24

allow-hotplug ens37
iface ens37 inet static
address 192.168.0.2/24
```

```
GNU nano 5.4 /etc/haproxy/haproxy.cfg
listen clusterWebH4
bind 192.168.0.2:80

# Mode écoute
mode http

# Mode du balancement (roundrobin (50%-50%))
balance roundrobin

# Option
option httpclose
option forwardfor

# Liste des serveurs impliqués par le balancement
server srvWeb1 172.20.0.1:80 check
server srvWeb2 172.20.0.2:80 check

# Pour les statistiques
stats enable
stats hide-version
stats refresh 30s
stats show-node
stats auth admin:password
stats uri /statistique
```

- 8) A partir de cette étape on duplique la machine pour gagner du temps et procéder aux configurations de la seconde machine srvweb2 en modifiant d'abord l'interface réseau comme ci-dessous, puis le fichier de configuration du service haproxy.

## b) HeartBeat

```
root@Master:~# apt install heartbeat wget unzip -y
```

- 1) On va maintenant procéder à l'installation du logiciel HeartBeat à l'aide de la commande « apt install heartbeat wget unzip -y ».

```
Paramétrage de perl (5.32.1-4+deb11u2) ...
Paramétrage de libsnmp40:amd64 (5.9+dfsg-3+b1) ...
Paramétrage de python3-google-auth (1.5.1-3) ...
Paramétrage de python3-boto3 (1.20.0+repack-1) ...
Paramétrage de python3-google-auth-httplib2 (0.0.4-2) ...
Paramétrage de liblrm2 (1.0.12-20) ...
Paramétrage de libtimedate-perl (2.3300-2) ...
Paramétrage de libcrmcluster29:amd64 (2.0.5-2) ...
Paramétrage de liblrm28:amd64 (2.0.5-2) ...
Paramétrage de python3-oauth2client (4.1.2-7) ...
Paramétrage de libpacemaker1:amd64 (2.0.5-2) ...
Paramétrage de python3-s3transfer (0.3.4-1) ...
Paramétrage de libnet-telnet-perl (3.04-1.1) ...
Paramétrage de snmp (5.9+dfsg-3+b1) ...
Paramétrage de cluster-glue (1.0.12-20) ...
addgroup: Le groupe « haclient » existe déjà en tant que groupe système. Fin de la procédure.
L'utilisateur système « hacluster » existe déjà. Fin de la procédure.
Created symlink /etc/systemd/system/multi-user.target.wants/logd.service → /lib/systemd/system/logd.service.
Paramétrage de python3-boto3 (1.19.14-1) ...
Paramétrage de python3-google-api (1.7.11-4) ...
Paramétrage de fence-agents (4.7.1-1) ...
Paramétrage de resource-agents (1:4.7.0-1) ...
resource-agents-deps.target is a disabled or a static unit, not starting it.
Paramétrage de pacemaker-resource-agents (2.0.5-2) ...
Paramétrage de heartbeat (1:3.0.6-11+deb11u1) ...
update-rc.d: warning: start and stop actions are no longer supported; falling back to defaults
Created symlink /etc/systemd/system/multi-user.target.wants/heartbeat.service → /lib/systemd/system/heartbeat.service.
Paramétrage de pacemaker (2.0.5-2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/pacemaker.service → /lib/systemd/system/pacemaker.service.
Paramétrage de pacemaker-cli-utils (2.0.5-2) ...
Traitement des actions différées (« triggers ») pour libc-bin (2.31-13+deb11u2) ...
root@Master:~#
```

- 2) Le logiciel HeartBeat est en train de s'installer.



```
root@Slave:~# apt install heartbeat wget unzip -y
```

- 3) On répète la procédure sur la seconde machine en installant le logiciel HeartBeat

```
Paramétrage de perl (5.32.1-4+deb11u2) ...
Paramétrage de libsnmp40:amd64 (5.9+dfsg-3+b1) ...
Paramétrage de python3-google-auth (1.5.1-3) ...
Paramétrage de python3-boto3 (1.20.0+repack-1) ...
Paramétrage de python3-google-auth-httplib2 (0.0.4-2) ...
Paramétrage de liblrm2 (1.0.12-20) ...
Paramétrage de libtimedate-perl (2.3300-2) ...
Paramétrage de libcrmcluster29:amd64 (2.0.5-2) ...
Paramétrage de liblrm28:amd64 (2.0.5-2) ...
Paramétrage de python3-oauth2client (4.1.2-7) ...
Paramétrage de libpacemaker1:amd64 (2.0.5-2) ...
Paramétrage de python3-s3transfer (0.3.4-1) ...
Paramétrage de libnet-telnet-perl (3.04-1.1) ...
Paramétrage de snmp (5.9+dfsg-3+b1) ...
Paramétrage de cluster-glue (1.0.12-20) ...
addgroup: Le groupe « haclient » existe déjà en tant que groupe système. Fin de la procédure.
L'utilisateur système « hacluster » existe déjà. Fin de la procédure.
Created symlink /etc/systemd/system/multi-user.target.wants/logd.service → /lib/systemd/system/logd.service.
Paramétrage de python3-boto3 (1.13.14-1) ...
Paramétrage de python3-googleapi (1.7.11-4) ...
Paramétrage de fence-agents (4.7.1-1) ...
Paramétrage de resource-agents (1:4.7.0-1) ...
resource-agents-deps.target is a disabled or a static unit, not starting it.
Paramétrage de pacemaker-resource-agents (2.0.5-2) ...
Paramétrage de heartbeat (1:3.0.6-11+deb11u1) ...
update-rc.d: warning: start and stop actions are no longer supported; falling back to defaults
Created symlink /etc/systemd/system/multi-user.target.wants/heartbeat.service → /lib/systemd/system/heartbeat.service.
Paramétrage de pacemaker (2.0.5-2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/pacemaker.service → /lib/systemd/system/pacemaker.service.
Paramétrage de pacemaker-cli-utils (2.0.5-2) ...
Traitement des actions différées (« triggers ») pour libc-bin (2.31-13+deb11u2) ...
root@Slave:~# _
```

- 4) Le logiciel HeartBeat est en train de s'installer sur la seconde machine.

```
GNU nano 5.4 /etc/ha.d/ha.cf
logfile /var/log/heartbeat
logfacility local0

keepalive 5

deadtime 30

bcast ens33

node Master Slave

auto_failback on
```

- 5) On va configurer les différents fichiers de configuration (avec le même contenu pour les 2 machines) pour HeartBeat, en commençant par le fichier suivant « nano /etc/ha.d/ha.cf » qui est un fichier vide, en renseignant les lignes indiquées ci-dessus puis on enregistre le fichier.

```
GNU nano 5.4 /etc/ha.d/haresources
Master IPaddr::192.168.0.10/24/ens37 haproxy
```

- 6) On configure le prochain fichier à l'aide de la commande « nano /etc/ha.d/haresources » qui est un fichier vide, en renseignant la ligne indiquée ci-dessus puis on enregistre le fichier.

```
GNU nano 5.4 /etc/ha.d/authkeys
auth 1
1 md5 sio1b
```

- 7) On configure le dernier fichier à l'aide de la commande « nano /etc/ha.d/authkeys » en ajoutant les lignes indiquées ci-dessus puis on enregistre le fichier.

```
GNU nano 5.4 /etc/hosts
127.0.0.1 localhost
172.20.0.1 srvWeb1
172.20.0.2 srvWeb2
# The following lines are desirable for IPv6 capable hosts
::1 localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

- 8) On configure les deux fichiers /etc/hosts présents au niveau des 2 serveurs web à l'aide de la commande « nano/etc/hosts » et en y ajoutant les lignes ci-dessus.

```
[Service]
RuntimeDirectory=heartbeat
RuntimeDirectorymode=755
```

- 9) Au niveau des 2 machines serveurs web, on édite le fichier de service de heartbeat à l'aide de la commande « systemctl edit heartbeat service » et en ajoutant les lignes ci-dessus.

```

root@Slave:~# service heartbeat status
• heartbeat.service - Heartbeat High Availability Cluster Communication and Membership
   Loaded: loaded (/lib/systemd/system/heartbeat.service; enabled; vendor preset: enabled)
   Drop-In: /etc/systemd/system/heartbeat.service.d
            └─override.conf
   Active: active (running) since Thu 2022-04-07 14:15:22 CEST; 8min ago
     Docs: man:heartbeat(8)
           http://www.linux-ha.org/wiki/Documentation
  Main PID: 552 (heartbeat)
    Tasks: 4 (limit: 2301)
   Memory: 10.4M
      CPU: 614ms
   CGroup: /system.slice/heartbeat.service
           └─552 heartbeat: master control process
             └─571 heartbeat: FIFO reader
               └─572 heartbeat: write: bcast ens37
                 └─573 heartbeat: read: bcast ens37

avril 07 14:15:25 Slave heartbeat[552]: [552]: info: Status update for node master: status active
avril 07 14:15:25 Slave heartbeat[552]: [552]: info: Comm_now_up(): updating status to active
avril 07 14:15:25 Slave heartbeat[552]: [552]: info: Local status now set to: 'active'
avril 07 14:15:25 Slave heartbeat[615]: [615]: debug: notify_world: setting SIGCHLD Handler to SIG_
avril 07 14:15:25 Slave harc(default)[623]: info: Running /etc/ha.d/rc.d/status status
avril 07 14:15:35 Slave heartbeat[552]: [552]: info: local resource transition completed.
avril 07 14:15:35 Slave heartbeat[552]: [552]: info: Initial resource acquisition complete (T_RESOU
avril 07 14:15:35 Slave heartbeat[697]: [697]: info: No local resources [/usr/share/heartbeat/Resou
avril 07 14:15:36 Slave heartbeat[552]: [552]: info: remote resource transition completed.
avril 07 14:15:51 Slave heartbeat[552]: [552]: info: Clock jumped backwards. Compensating.
lines 1-27/27 (END)

```

- 10) On vérifie que le logiciel HeartBeat est bien fonctionnel au niveau de la première machine à l'aide de la commande « service heartbeat status ». La commande indique que le service est « Active : active (running) » donc il est bien fonctionnel.

```

avril 07 14:15:20 Master haproxy[677]: [WARNING] 096/141520 (677) : Server clusterWebHA/srvWeb1 is
avril 07 14:15:20 Master haproxy[677]: [NOTICE] 096/141520 (677) : haproxy version is 2.2.9-2+deb11
avril 07 14:15:20 Master haproxy[677]: [NOTICE] 096/141520 (677) : path to executable is /usr/sbin/p
avril 07 14:15:20 Master haproxy[677]: [ALERT] 096/141520 (677) : sendmsg()/writev() failed in logg
avril 07 14:15:20 Master haproxy[677]: [WARNING] 096/141520 (677) : Server clusterWebHA/srvWeb2 is
avril 07 14:15:20 Master haproxy[677]: [ALERT] 096/141520 (677) : proxy 'clusterWebHA' has no serve
lines 1-24/24 (END)

root@Master:~# service heartbeat status
• heartbeat.service - Heartbeat High Availability Cluster Communication and Membership
   Loaded: loaded (/lib/systemd/system/heartbeat.service; enabled; vendor preset: enabled)
   Drop-In: /etc/systemd/system/heartbeat.service.d
            └─override.conf
   Active: active (running) since Thu 2022-04-07 14:15:14 CEST; 8min ago
     Docs: man:heartbeat(8)
           http://www.linux-ha.org/wiki/Documentation
  Main PID: 559 (heartbeat)
    Tasks: 4 (limit: 2301)
   Memory: 18.2M
      CPU: 949ms
   CGroup: /system.slice/heartbeat.service
           └─559 heartbeat: master control process
             └─592 heartbeat: FIFO reader
               └─593 heartbeat: write: bcast ens37
                 └─594 heartbeat: read: bcast ens37

avril 07 14:15:34 Master heartbeat[559]: [559]: debug: StartNextRemoteRscReq(): child count 1
avril 07 14:15:34 Master heartbeat[803]: [803]: debug: notify_world: setting SIGCHLD Handler to SIG_
avril 07 14:15:34 Master harc(default)[809]: info: Running /etc/ha.d/rc.d/ip-request-resp ip-reque
avril 07 14:15:34 Master ip-request-resp(default)[819]: received ip-request-resp IPaddr::192.168.0.
avril 07 14:15:34 Master ResourceManager(default)[835]: info: Acquiring resource group: master IPad
avril 07 14:15:34 Master /usr/lib/ocf/resource.d/heartbeat/IPaddr(IPaddr_192.168.0.10)[895]: INFO:
avril 07 14:15:34 Master ResourceManager(default)[914]: info: Running /etc/ha.d/resource.d/IPaddr
avril 07 14:15:34 Master IPaddr(IPaddr_192.168.0.10)[971]: INFO: Using calculated netmask for 192.1
avril 07 14:15:34 Master IPaddr(IPaddr_192.168.0.10)[1009]: INFO: eval ifconfig ens37:0 192.168.0.1
avril 07 14:15:34 Master /usr/lib/ocf/resource.d/heartbeat/IPaddr(IPaddr_192.168.0.10)[1041]: INFO
lines 1-27/27 (END)

```

- 11) On vérifie que le logiciel HeartBeat est bien fonctionnel au niveau de la seconde machine à l'aide de la commande « service heartbeat status ». La commande indique que le service est « Active : active (running) » donc il est bien fonctionnel.

### c) ownCloud

```
root@SrvWeb2:~# ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.20.0.2 netmask 255.255.255.0 broadcast 172.20.0.255
    inet6 fe80::20c:29ff:fed5:6e09 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:d5:6e:09 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 8 bytes 656 (656.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Boucle locale)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

- 1) On configure 2 cartes réseaux sur les 2 machines à l'aide de la commande « nano /etc/network/interfaces » et en renseignant pour la première carte (ens33) une adresse obtenue statiquement que nous configurons nous-mêmes à savoir 172.20.0.1/24 pour la première machine et 172.20.0.2/24 pour la seconde machine

```
root@Server:~# apt update && apt upgrade -y
Réception de :1 http://deb.debian.org/debian bullseye InRelease [116 kB]
Réception de :2 http://security.debian.org/debian-security bullseye-security InRelease [44,1 kB]
Réception de :3 http://deb.debian.org/debian bullseye-updates InRelease [39,4 kB]
Réception de :4 http://security.debian.org/debian-security bullseye-security/main Sources [110 kB]
Réception de :5 http://deb.debian.org/debian bullseye/main Sources [8 627 kB]
Réception de :6 http://security.debian.org/debian-security bullseye-security/main amd64 Packages [12
3 kB]
Réception de :7 http://security.debian.org/debian-security bullseye-security/main Translation-en [78
,7 kB]
Réception de :8 http://deb.debian.org/debian bullseye-updates/main Sources.diff/Index [6 117 B]
Réception de :9 http://deb.debian.org/debian bullseye-updates/main amd64 Packages.diff/Index [6 117
B]
Réception de :10 http://deb.debian.org/debian bullseye-updates/main Sources T-2022-03-26-2012.00-F-2
022-03-26-2012.00.pdiff [184 B]
Réception de :10 http://deb.debian.org/debian bullseye-updates/main Sources T-2022-03-26-2012.00-F-2
022-03-26-2012.00.pdiff [184 B]
Réception de :11 http://deb.debian.org/debian bullseye-updates/main amd64 Packages T-2022-03-26-2012
.00-F-2022-03-26-2012.00.pdiff [168 B]
Réception de :11 http://deb.debian.org/debian bullseye-updates/main amd64 Packages T-2022-03-26-2012
.00-F-2022-03-26-2012.00.pdiff [168 B]
Réception de :12 http://deb.debian.org/debian bullseye/main amd64 Packages [8 182 kB]
Réception de :13 http://deb.debian.org/debian bullseye/main Translation-en [6 241 kB]
23,6 Mo réceptionnés en 6s (3 628 ko/s)
Lecture des listes de paquets... 99%
```

- 2) On commence par récupérer les mises à jour et les installer sur la machine à l'aide de la commande « apt update && apt upgrade -y ».

```
root@Server:~# apt install apache2 php mariadb-server mp3info curl libcurl4-openssl-dev bzip2 php-gd
php-mysql php-common libcurl4 php-curl zip php-zip php-dom php-intl php-mbstring wget php-zip_
```

- 3) On installe les paquets utiles pour le fonctionnement d'ownCloud.

```
root@Server:~# mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 30
Server version: 10.5.15-MariaDB-0+deb11u1 Debian 11

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)]>
```

```
MariaDB [(none)]> create database ownCloudDB;
```

```
MariaDB [(none)]> grant all privileges on *.* to oc_admin@'localhost' identified by 'password';
Query OK, 0 rows affected (0.001 sec)
```

```
MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.001 sec)
```

- 4) Une fois les paquets installés, on se connecte au système de gestion de base de données pour créer une base de données et un utilisateur administrateur de la BDD d'ownCloud à l'aide des commandes ci-dessus.

```
root@Server:~# wget https://download.owncloud.org/community/owncloud-10.9.1.tar.bz2
--2022-03-31 15:16:10-- https://download.owncloud.org/community/owncloud-10.9.1.tar.bz2
Résolution de download.owncloud.org (download.owncloud.org)... 167.233.14.167, 2a01:4f8:1c1d:3d1::1
Connexion à download.owncloud.org (download.owncloud.org)|167.233.14.167|:443... connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : 64725210 (62M) [application/x-bzip2]
Sauvegarde en : « owncloud-10.9.1.tar.bz2 »

owncloud-10.9.1.tar.bz2 100%[=====] 61,73M 13,0MB/s ds 4,8s
2022-03-31 15:16:15 (12,9 MB/s) - « owncloud-10.9.1.tar.bz2 » sauvegardé [64725210/64725210]
```

- 5) On télécharge ownCloud depuis le site officiel à l'aide de la commande suivante : « wget https://download.owncloud.org/community/owncloud-10.9.1tar.bz2 »

```
root@Server:~# tar xjvf owncloud-10.9.1.tar.bz2_
```

```
owncloud/lib/private/Repair/RepairMismatchFileCachePath.php
owncloud/lib/private/Repair/Preview.php
owncloud/lib/private/Repair/DisableExtraThemes.php
owncloud/lib/private/Repair/RemoveGetETagEntries.php
owncloud/lib/private/Repair/OldGroupMembershipShares.php
owncloud/lib/private/Repair/DropOldTables.php
owncloud/lib/private/Repair/SharePropagation.php
owncloud/lib/private/Repair/RemoveRootShares.php
owncloud/lib/private/Repair/DropOldJobs.php
owncloud/lib/private/Repair/Apps.php
owncloud/lib/private/Repair/RepairInvalidShares.php
owncloud/lib/private/Repair/SqliteAutoIncrement.php
owncloud/lib/private/Repair/RepairUnmergedShares.php
owncloud/lib/private/Repair/Collation.php
owncloud/lib/private/Repair/RepairSubShares.php
owncloud/lib/private/Repair/RepairMimeType.php
owncloud/lib/private/Repair/RepairOrphanedSubshare.php
owncloud/lib/private/Repair/InnoDB.php
owncloud/lib/private/Repair/MoveAvatarIntoSubFolder.php
owncloud/lib/private/Repair/MoveAvatarOutsideHome.php
owncloud/lib/private/Repair/SearchLuceneTables.php
owncloud/lib/private/AppConfig.php
owncloud/resources/
owncloud/resources/config/
owncloud/resources/config/mimetypealiases.dist.json
owncloud/resources/config/ca-bundle.crt
owncloud/resources/config/readme.md
owncloud/resources/config/mimetypesmapping.dist.json
owncloud/resources/codesigning/
owncloud/resources/codesigning/intermediate.crl.pem
owncloud/resources/codesigning/root.crt
owncloud/resources/codesigning/core.crt
owncloud/COPYING
owncloud/index.php
owncloud/CHANGELOG.md
owncloud/status.php
```

- 6) On décompresse le fichier téléchargé à l'aide de la commande « tar xjvf owncloud-10.9.1.tar.bz2 »

```
root@Server:~# mv owncloud /var/www/html
```

- 7) On déplace le dossier résultat 'owncloud' sous le serveur web à l'aide de la commande « mv owncloud /var/www/html »

```
root@Server:~# chown -R www-data:www-data /var/www/html/owncloud
```

- 8) On change le propriétaire ainsi que le groupe propriétaire du dossier owncloud en www-data qui est l'utilisateur correspondant à apache2 à l'aide de la commande « chown -R www-data :www-data /var/www/html/owncloud »

```

root@Server:~# ls -l /var/www/html
root@Server:~# ls -l /var/www/html
total 4440
-rw-r--r-- 1 root root 4534581 3 févr. 17:15 master.zip
drwxr-xr-x 12 www-data www-data 4096 12 janv. 15:27 owncloud
drwxr-xr-x 3 root root 4096 31 mars 14:17 thegrill-master

```

- 9) On vérifie que le propriétaire et que le groupe propriétaire est bien www-data à l'aide de la commande « `ls -l /var/www/html` »

```

root@Server:~# service apache2 restart

```

```

root@SrvWeb1:~# service apache2 status
• apache2.service - The Apache HTTP Server
  Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor preset: enabled)
  Active: active (running) since Thu 2022-04-07 17:41:03 CEST; 2min 46s ago
    Docs: https://httpd.apache.org/docs/2.4/
  Process: 459 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SUCCESS)
 Main PID: 564 (apache2)
   Tasks: 6 (limit: 2301)
  Memory: 25.3M
    CPU: 383ms
  CGroup: /system.slice/apache2.service
          └─564 /usr/sbin/apache2 -k start
            └─576 /usr/sbin/apache2 -k start
              └─577 /usr/sbin/apache2 -k start
                └─578 /usr/sbin/apache2 -k start
                  └─579 /usr/sbin/apache2 -k start
                    └─585 /usr/sbin/apache2 -k start

avril 07 17:41:00 SrvWeb1 systemd[1]: Starting The Apache HTTP Server...
avril 07 17:41:03 SrvWeb1 apachectl[481]: AH00557: apache2: apr_socketaddr_info_get() failed for SrvWeb1
avril 07 17:41:03 SrvWeb1 apachectl[481]: AH00558: apache2: Could not reliably determine the server's fully qualified domain name, please add the 'ServerName' directive with the fully qualified domain name to the configuration.
avril 07 17:41:03 SrvWeb1 systemd[1]: Started The Apache HTTP Server.

```

- 10) On redémarre le service apache2 à l'aide de la commande « `service apache2 restart` ». Le service affiche 'Active : active (running)' donc apache2 est bien fonctionnel.

```

root@Server:~# nano /etc/apache2/sites-enabled/000-default.conf

```

- 11) On va configurer le fichier `/etc/apache2/sites-enabled/000-default.conf` à l'aide de la commande « `nano` ».

```

GNU nano 5.4 /etc/apache2/sites-enabled/000-default.conf *
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html/owncloud_

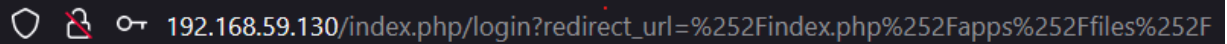
```

- 12) On modifie le DocumentRoot d'apache2 en « `/var/www/html/owncloud` ».



```
root@Server:~# nano /var/www/html/owncloud/config/config.php_
```

- 13) On va éditer le fichier suivant à l'aide de la commande  
« nano /var/www/html/owncloud/config/config.php ».



- 14) On effectue une tentative de connexion à l'appli owncloud, ou la machine physique en utilisant le navigateur et l'adresse IP de l'interface en NAT (obtenue dynamiquement)



```
GNU nano 5.4 /var/www/html/owncloud/config/config.php
<?php
$CONFIG = array (
    'instanceid' => 'ocumb8kmmbg',
    'passwordsalt' => 'ZQTnPu4IRr0qsRcdB49JCzBs9p6uXk',
    'secret' => '1zDu7m6EfB8PDL5K2stXaFRgM65ispoJH0UnygaABoWJyxJV',
    'trusted_domains' =>
        array (
            0 => '192.168.59.130',
        ),
    'datadirectory' => '/var/www/html/owncloud/data',
    'overwrite.cli.url' => 'http://192.168.59.130',
    'dbtype' => 'mysql',
    'version' => '10.9.1.2',
    'dbname' => 'ownCloudDB',
    'dbhost' => 'localhost',
    'dbtableprefix' => 'oc_',
    'mysql.utf8mb4' => true,
    'dbuser' => 'oc_admin',
    'dbpassword' => 'pwUqH/ykiFuJ13e0LK8LRIZYw18aXu',
);
```

- 15) On édite le fichier php à l'aide de la commande  
« nano /var/www/html/owncloud/config/config.php » pour changer l'utilisateur 'oc\_adminX'  
présent au niveau de l'avant dernière ligne en 'oc\_admin'.



```
root@Server:~# nano -c /etc/apache2/sites-available/000-default.conf_
GNU nano 5.4 /etc/apache2/sites-enabled/000-default.conf *
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html/owncloud_
```

- 16) On édite le fichier suivant « nano -c /etc/apache2/sites-available/000-default.conf » pour  
modifier la ligne 12 DocumentRoot et ajouter '/owncloud après /var/www/html '.



```
root@Server:~# service apache2 restart
```

17) On redémarre le service apache2.

```
root@Server:~# nano /var/www/html/owncloud/config/config.php -c_
```

```
GNU nano 5.4 /var/www/html/owncloud/config/config.php
<?php
$CONFIG = array (
    'instanceid' => 'ocumb8kmambg',
    'passwordsalt' => '74JSJbfAfS0zSqWZoKAYxmm60SJzVp',
    'secret' => 'pqfnvzGOSnuJtYMQriUSfF7GZcRUPZnXtNrmmAL+B7QZLS1g',
    'trusted_domains' =>
        array (
            0 => '172.20.0.1',
        ),
    'datadirectory' => '/var/www/html/owncloud/data',
    'logtimezone' => 'UTC',
    'overwriteurl' => 'http://172.20.0.1:8080/owncloud/'
);
```

```
GNU nano 5.4 /var/www/html/owncloud/config/config.php
<?php
$CONFIG = array (
    'instanceid' => 'ocumb8kmmbg',
    'passwordsalt' => '74JSJbfAfS0zSqWZoKAYxmm60SJzVp',
    'secret' => 'pqfnvzG0SnuJtYMqriUSff7GZcRUPZnXtNrmmAL+B7QZLS1g',
    'trusted_domains' =>
        array (
            0 => '172.20.0.2',
        )
    )
}
```

18) On ouvre le fichier de configuration php à l'aide de la commande « nano /var/www/html/owncloud/config/config.php -c » (le '-c' indique les lignes présentes à l'intérieur du fichier lors de l'ouverture). Puis au niveau de la ligne 8 on ajoute notre adresse IP et on enregistre le fichier (en distinguant bien l'adresse de la machine 1 de la machine 2).

## V . Validation

```
root@Slave:~#  
root@Slave:~#  
root@Slave:~# ping 172.20.0.1  
PING 172.20.0.1 (172.20.0.1) 56(84) bytes of data.  
64 bytes from 172.20.0.1: icmp_seq=1 ttl=64 time=0.479 ms  
64 bytes from 172.20.0.1: icmp_seq=2 ttl=64 time=0.556 ms  
64 bytes from 172.20.0.1: icmp_seq=3 ttl=64 time=0.584 ms  
64 bytes from 172.20.0.1: icmp_seq=4 ttl=64 time=0.534 ms  
  
--- 172.20.0.1 ping statistics ---  
4 packets transmitted, 4 received, 0% packet loss, time 3054ms  
rtt min/avg/max/mdev = 0.479/0.538/0.584/0.038 ms  
root@Slave:~# _
```

- 1) La machine HAProxy 2 (slave) est capable de communiquer avec la machine srvweb1 .

```
root@Slave:~# ping 172.20.0.2  
PING 172.20.0.2 (172.20.0.2) 56(84) bytes of data.  
64 bytes from 172.20.0.2: icmp_seq=1 ttl=64 time=0.628 ms  
64 bytes from 172.20.0.2: icmp_seq=2 ttl=64 time=0.553 ms  
64 bytes from 172.20.0.2: icmp_seq=3 ttl=64 time=0.562 ms  
^C  
--- 172.20.0.2 ping statistics ---  
3 packets transmitted, 3 received, 0% packet loss, time 2036ms  
rtt min/avg/max/mdev = 0.553/0.581/0.628/0.033 ms  
root@Slave:~#
```

- 2) La machine HAProxy 2 (slave) est capable de communiquer avec la machine srvweb2.

```

root@SrvWeb1:~#
root@SrvWeb1:~#
root@SrvWeb1:~# ping 172.20.0.2
PING 172.20.0.2 (172.20.0.2) 56(84) bytes of data.
64 bytes from 172.20.0.2: icmp_seq=1 ttl=64 time=0.938 ms
64 bytes from 172.20.0.2: icmp_seq=2 ttl=64 time=0.497 ms
64 bytes from 172.20.0.2: icmp_seq=3 ttl=64 time=0.518 ms
^X64 bytes from 172.20.0.2: icmp_seq=4 ttl=64 time=0.516 ms
64 bytes from 172.20.0.2: icmp_seq=5 ttl=64 time=0.507 ms
^C
--- 172.20.0.2 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4066ms
rtt min/avg/max/mdev = 0.497/0.595/0.938/0.171 ms
root@SrvWeb1:~# ping 172.20.0.2

```

- 3) La machine srvweb1 est capable de communiquer avec la machine srvweb2

```

root@SrvWeb2:~#
root@SrvWeb2:~#
root@SrvWeb2:~# ping 170.20.0.1
ping: connect: Le réseau n'est pas accessible
root@SrvWeb2:~# ping 172.20.0.1
PING 172.20.0.1 (172.20.0.1) 56(84) bytes of data.
64 bytes from 172.20.0.1: icmp_seq=1 ttl=64 time=0.723 ms
64 bytes from 172.20.0.1: icmp_seq=2 ttl=64 time=0.546 ms
64 bytes from 172.20.0.1: icmp_seq=3 ttl=64 time=8.19 ms
^C
--- 172.20.0.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2049ms
rtt min/avg/max/mdev = 0.546/3.152/8.187/3.561 ms
root@SrvWeb2:~#

```

- 4) La machine srvweb2 est capable de communiquer avec la machine srvweb1.