
Projet 6: Haute disponibilité d'un serveur Web d'application et réplication de base de données .

Veynand Saint Fiacre Lucille

30/11/2020

01/12/2020



Table des matières

Introduction	2
Contexte de travail et gestion des configurations	2
Gestion du travail en équipe et gestion du projet	2
Activités compétences du référentiel du BTS SIO	2
(reseaucerta.org)	2
La documentation.	2
Le Maquettage	2
Incident problème et assistance	2
Formation, autoformation et veille technologique.	2
Test et vérification	2
Le projet en détail.	2



1 Contexte de travail et gestion des configurations

Voir fichier de contexte

2 Gestion du travail en équipe et gestion du projet

On a travaillé le:

30/11/2020

01/12/2020

07/12/2020

08/12/2020

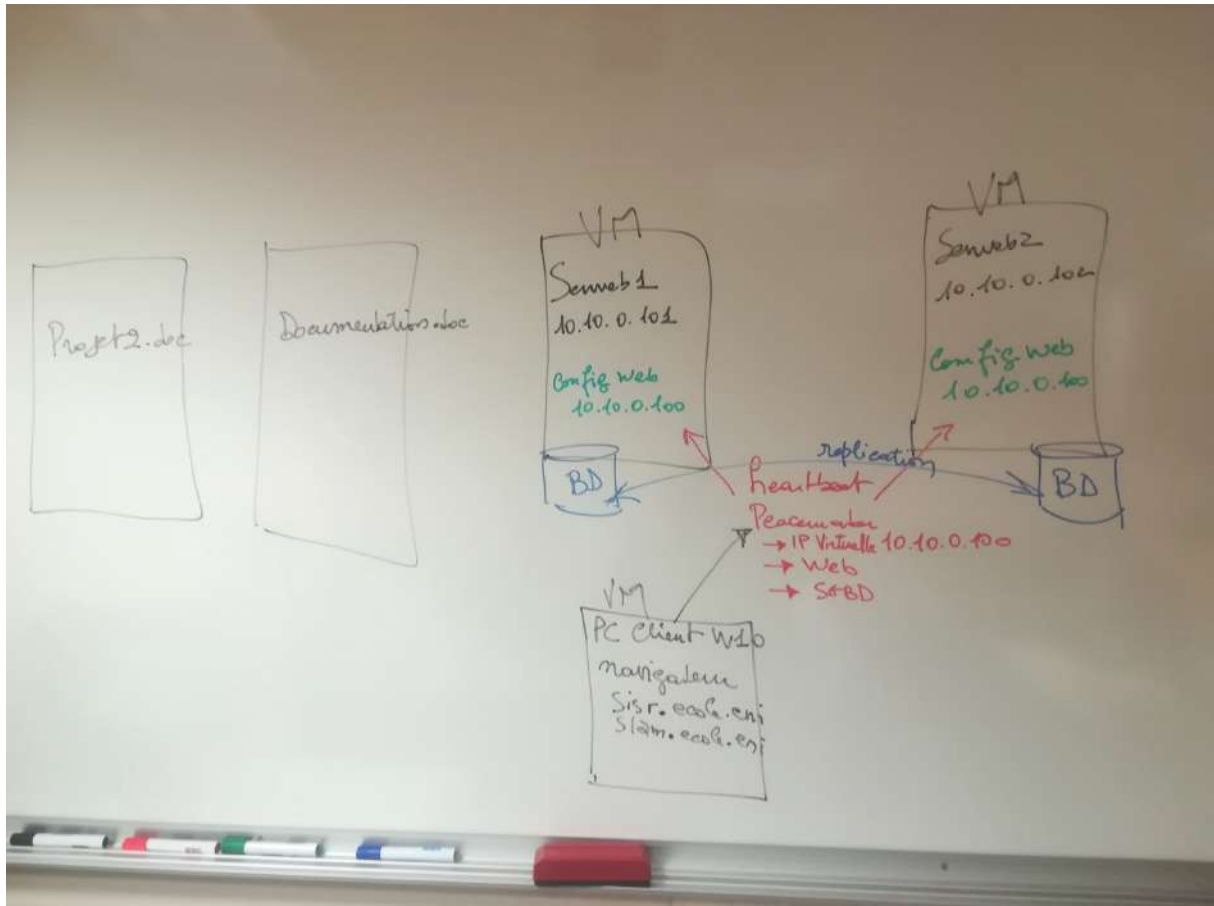
3 Activités compétences du référentiel du BTS SIO (reseaucerta.org)

<p>A1.1.1 Analyse du cahier des charges d'un service à produire</p>	<ul style="list-style-type: none"> • C1.1.1.1 Recenser et caractériser les contextes d'utilisation, les processus et les acteurs sur lesquels le service à produire aura un impact • C1.1.1.2 Identifier les fonctionnalités attendues du service à produire
<p>A1.1.2 Étude de l'impact de l'intégration d'un service sur le système informatique</p>	<ul style="list-style-type: none"> • C1.1.2.1 Analyser les interactions entre services • C1.1.2.2 Recenser les composants de l'architecture technique sur lesquels le service à produire aura un impact
<p>A1.1.3 Étude des exigences liées à la qualité attendue d'un service</p>	<ul style="list-style-type: none"> • C1.1.3.1 Recenser et caractériser les exigences liées à la qualité attendue du service à produire • C1.1.3.2 Recenser et caractériser les exigences de sécurité pour le service à produire
<p>A1.2.1 Élaboration et présentation d'un dossier de choix de solution technique</p> <p>A1.3.4 Déploiement d'un service</p>	<ul style="list-style-type: none"> • C1.3.4.1 Mettre au point une procédure d'installation de la solution • C1.3.4.2 Automatiser



	<p>l'installation de la solution</p> <p>C1.3.4.3 Mettre en exploitation le service</p>
A2.2.3 Réponse à une interruption de service	<ul style="list-style-type: none"> • C2.2.3.1 Appliquer la procédure de continuité du service en mode dégradé - C2.2.3.2 Appliquer la procédure de reprise du service
A3.1.3 Prise en compte du niveau de sécurité nécessaire à une infrastructure	<ul style="list-style-type: none"> • C3.1.3.1 Caractériser des solutions de sécurité et en évaluer le coût • C3.1.3.2 Proposer une solution de sécurité compatible avec les contraintes techniques, financières, juridiques et organisationnelles
A3.2.1 Installation et configuration d'éléments d'infrastructure	<ul style="list-style-type: none"> • C3.2.1.1 Installer et configurer un élément d'interconnexion, un service, un serveur, un équipement terminal utilisateur • C3.2.1.2 Installer et configurer un élément d'infrastructure permettant d'assurer la continuité de service, un système de régulation des éléments d'infrastructure, un outil de métrologie, un dispositif d'alerte <p>C3.2.1.3 Installer et configurer des éléments de sécurité permettant d'assurer la protection du système informatique</p>

4 Le Maquettage



5 Le projet en détail.

1. Lister les techniques et processus communément utilisés pour améliorer la disponibilité d'un service.

- Redondance des serveurs avec leurs virtualisations:
- Équilibre de charges
- Sécurité matérielle avec MAJ et ne pas les laisser les paramètres par défaut
- Sauvegarde des données
- Mettre en place un PRA (Plan de Reprise d'Activité). Soit que faire s'il y'a une panne
- Mettre en place un PCA (Plan de Continuité d'Activité). Soit la définition des priorités

2. Expliquer le terme high availability ?

C'est la haute disponibilité. Elle regroupe les techniques afin de garantir un pourcentage de disponibilité élevé de service.

3. Quelle sont les conséquences d'un arrêt de serveur et/ou interruption de service au sein d'une organisation ?

Des pertes de données, soit une perte du chiffre d'affaire

4. Quelle est la configuration à mettre en place pour assurer la haute disponibilité de service ?

Mettre en place des clusters (groupes de serveurs) avec des serveurs primaires (actifs) et d'autres secondaires (passifs)

5. Que signifient PRA et PCA ?

PRA (Plan de Reprise d'Activité). Soit que faire s'il y'a une panne

PCA (Plan de Continuité d'Activité). Soit la définition des priorités



6. Comment la redondance de serveur peut assurer la haute disponibilité de service. ?

Dans un système composé de serveurs actifs et passifs, Les passifs surveillent les actifs, si un actif ne donne plus de signe de vie, ces derniers prennent la relève.

Une fois le serveur actif réparé, le serveur passif retourne à un état de surveillance

7. Préparer votre serveur web (machine physique ou virtuelle Servweb1)

8. Expliquer le fonctionnement de HEARTBEAT ?

C'est un logiciel détectant la défaillance d'un serveur à l'aide de clusters. Deux serveurs identiques dont un serveur maître actif et un serveur esclave passif. Entre ces derniers s'échange des signaux UDP permettant de vérifier l'état de vie du serveur maître. Si le serveur maître actif n'échange plus de signaux le serveur esclave passif reprend la relève.

9. Quel le rôle du logiciel PACEMAKER ?

Logiciel de gestion de ressources et services dans un cluster (FTP, SGBP, MySQL, Apache...).

Il permet de créer, démarrer, arrêter et superviser ces ressources.

Il gère notamment les adresses virtuelles

10. La haute disponibilité sous-entend que plusieurs machines seront utilisées pour répondre à un même service. Seulement, chaque machine a normalement une adresse IP différente sur le réseau, ce qui est problématique puisque le client ne connaît qu'une seule adresse IP et/ou le nom d'hôte pleinement qualifié (qui n'est associé qu'à une seule adresse IP). Comment alors faire en sorte que toutes les machines en haute disponibilité répondent à la même adresse ?

On attribue une adresse IP virtuelle créée par le logiciel Pacemaker à un hôte. Cette adresse correspond à celle du serveur web. Toutes les requêtes sont dirigées vers l'adresse IP virtuelle, qui est attribuée à un cluster et non pas à une seule machine physique

11. Pourquoi c'est l'adresse IP virtuelle qui doit « apparaître » dans les fichiers de configuration des services (par exemple, dans la configuration des serveurs Web virtuels et celle du serveur DNS) ?

Le serveur web sera défini avec une adresse IP ou un nom FQDN que l'utilisateur connaît.

Soit s'il l'on cherche monbureau numerique.fr il nous redirige sur l'adresse IP 80.247.238.197

12. La plupart des services (comme le service web) inclus dans un système de haute disponibilité utilise des données stockées dans des bases de données qu'il est donc nécessaire de répliquer également comment peut-on assurer cette réplication de données ?

L'utilisation d'un logiciel synchronisant une partition entre deux machines (sorte de RAID en réseau).

Un système maître/esclave dans la réplication MySQL (le maître écrit et les esclaves recopient).

13. Vérifiez que les applications Web sistr et slam sont bien accessibles sur le serveur maître (modifier les fichiers du dns et d'apache avec les nouvelles adresses IP 10.10.R.100 R est le numéro de votre rangée ou groupe de travail).

Nom serveur maître : serweb1

```
root@serweb1:/home/sistr#
```

Adresse Ip 10.10.R.100/24

- ifconfig

```
eth0      Link encap:Ethernet  HWaddr 00:15:5d:1f:44:31
          inet addr:10.10.1.100  Bcast:10.10.1.255  Masque:255.255.255.0
```

- Fichier /etc/hosts



```
root@serweb1:/home/sisr# cat /etc/hosts
#127.0.0.1, localhost permet de vérifier bon fonctionnement de la carte
127.0.0.1    localhost
#Ce fichier permet la correspondance entre une adresse IPV4 et un nom
#utile si absence de DNS
#127.0.0.1    serweb1.ecole.eni serweb1 sisr slam

# The following lines are desirable for IPv6 capable hosts
::1    localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

- Répertoire /etc/bind

```
root@serweb1:/home/sisr# ls /etc/bind
bind.keys  db.empty  ecole.eni_inv  named.conf.options
db.0       db.local  named.conf     rndc.key
db.127     db.root   named.conf.default-zones  zones.rfc1918
db.255     ecole.eni named.conf.local
```

- Fichier /etc/bind/named.conf.local

```
root@serweb1:/home/sisr# cat /etc/bind/named.conf.local
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
zone "ecole.eni" {
    type master;
    file "/etc/bind/ecole.eni";
};

zone "1.10.10.in-addr.arpa" {
    type master;
    file "/etc/bind/ecole.eni_inv";
};
```

- Fichier /etc/bind/ecole.eni



```
root@serweb1:/home/sisr# cat /etc/bind/ecole.eni
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      localhost. root.localhost. (
                        2          ; Serial
                        604800     ; Refresh
                        86400      ; Retry
                        2419200    ; Expire
                        604800 )   ; Negative Cache TTL
;
@         IN      NS       localhost.
sisr      IN      A        10.10.1.100
slam      IN      A        10.10.1.100
toto      IN      A        10.10.1.101
```

- Fichier /etc/bind/ecole.eni_inv

```
root@serweb1:/home/sisr# cat /etc/bind/ecole.eni_inv
;
; BIND reverse data file for local loopback interface
;
$TTL      604800
@         IN      SOA      localhost. root.localhost. (
                        1          ; Serial
                        604800     ; Refresh
                        86400      ; Retry
                        2419200    ; Expire
                        604800 )   ; Negative Cache TTL
;
@         IN      NS       localhost.
100       IN      PTR      sisr
100       IN      PTR      slam
101       IN      PTR      toto
```

- Test DNS



```
root@serweb1:/home/sisr# nslookup 10.10.1.100
Server:          127.0.0.1
Address:         127.0.0.1#53

100.1.10.10.in-addr.arpa      name = sisr.1.10.10.in-addr.arpa.
100.1.10.10.in-addr.arpa      name = slam.1.10.10.in-addr.arpa.

root@serweb1:/home/sisr# nslookup slam.ecole.eni
Server:          127.0.0.1
Address:         127.0.0.1#53

Name:   slam.ecole.eni
Address: 10.10.1.100

root@serweb1:/home/sisr# nslookup sisr.ecole.eni
Server:          127.0.0.1
Address:         127.0.0.1#53

Name:   sisr.ecole.eni
Address: 10.10.1.100
```

- Répertoire `/etc/apache2`

```
root@serweb1:/home/sisr# ls /etc/apache2/
apache2.conf  envvars          magic            mods-enabled    sites-available
conf.d        httputiliteurs  mods-available  ports.conf      sites-enabled
```

- Fichier `/etc/apache2/apache2.conf`



```
root@serweb1:/home/sisr# cat /etc/apache2/apache2.conf | egrep -v "(^#.*)"

LockFile ${APACHE_LOCK_DIR}/accept.lock

PidFile ${APACHE_PID_FILE}

Timeout 300

KeepAlive On

MaxKeepAliveRequests 100

KeepAliveTimeout 5

<IfModule mpm_prefork_module>
    StartServers      5
    MinSpareServers   5
    MaxSpareServers   10
    MaxClients        150
    MaxRequestsPerChild  0
</IfModule>

<IfModule mpm_worker_module>
    StartServers      2
    MinSpareThreads   25
    MaxSpareThreads   75
    ThreadLimit        64
    ThreadsPerChild   25
    MaxClients        150
```



```

    MaxRequestsPerChild    0
</IfModule>

<IfModule mpm_event_module>
    StartServers           2
    MinSpareThreads        25
    MaxSpareThreads        75
    ThreadLimit            64
    ThreadsPerChild        25
    MaxClients             150
    MaxRequestsPerChild    0
</IfModule>

User ${APACHE_RUN_USER}
Group ${APACHE_RUN_GROUP}

AccessFileName .htaccess

<Files ~ "^.ht">
    Order allow,deny
    Deny from all
    Satisfy all
</Files>

DefaultType None

HostnameLookups Off

ErrorLog ${APACHE_LOG_DIR}/error.log

LogLevel warn

Include mods-enabled/*.load
Include mods-enabled/*.conf

Include ports.conf

LogFormat "%v:%p %h %l %u %t \"%r\" %>s %0 \"%{Referer}i\" \"%{User-Agent}i\"" vhost_combined
LogFormat "%h %l %u %t \"%r\" %>s %0 \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %0" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

Include conf.d/

Include sites-enabled/

ServerName localhost

```

- Fichier /etc/apache2/sites-available/default



```
root@serwebl:/home/sisr# cat /etc/apache2/sites-available/default | egrep -v "(^#.*)"
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    NameVirtualHost 10.10.1.100
    DocumentRoot /var/www
    DirectoryIndex accueil.html index.html
    <Directory />
        Options FollowSymLinks
        AllowOverride None
    </Directory>
    <Directory /var/www/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
    </Directory>

    ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
    <Directory "/usr/lib/cgi-bin">
        AllowOverride None
        Options +ExecCGI -MultiViews +SymLinksIfOwnerMatch
        Order allow,deny
        Allow from all
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log

    # Possible values include: debug, info, notice, warn, error, crit,
    # alert, emerg.
    LogLevel warn
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

- **Fichier /etc/apache2/sites-available/sisr**

```
root@serwebl:/home/sisr# cat /etc/apache2/sites-available/sisr | egrep -v "(^#.*)"
<VirtualHost 10.10.1.100:80>
    DocumentRoot /var/www/sisr
    DirectoryIndex sisr.php
    ServerName sisr.ecole.eni
    <Directory /var/www/sisr>
        AllowOverride AuthConfig
    </Directory>
</VirtualHost>
```

- **Fichier /etc/apache2/sites-available/slam**

```
root@serwebl:/home/sisr# cat /etc/apache2/sites-available/slam | egrep -v "(^#.*)"
<VirtualHost 10.10.1.100:80>
    DocumentRoot /var/www/slam
    DirectoryIndex slam.php
    ServerName slam.ecole.eni
</VirtualHost>
```

- **Fichier applications WEB dans /var/www**

```
root@serwebl:/home/sisr# chmod 777 /var/www/
root@serwebl:/home/sisr# ls /var/www/
index.html  sisr  slam
```

- **Vérification SGBD Mysql et des bases de données**



```
mysql> use sio
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> use sistr
ERROR 1049 (42000): Unknown database 'sistr'
mysql> select * from sistr
    -> ;
+-----+-----+-----+
| id | pseudo      | message                |
+-----+-----+-----+
| 1  | Hamid       | Est mÃ©chant           |
| 2  | ClÃ©ment    | est trÃ©s gentil       |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>
```

14. Définir un cluster et un nœud pour Heartbeat ?

Un cluster est un regroupement d'ordinateurs reliés travaillant en collaboration. Il est constitué d'au moins 2 machines (physique ou machine virtuelle). Les machines d'un cluster sont nommées nœuds ou nodes en anglais.

15. En quoi consiste la gestion d'un service en Cluster (un service étant une ressource au sens de Pacemaker) ?

Cela permet de faire la continuité de services. La gestion consiste à détecter les défaillances d'un poste. Dans un cluster, lorsqu'une machine considérée comme « primaire » n'est plus considérée comme accessible, une deuxième machine est démarrée avec les services et cela afin.

16. Comment peut-on assurer le fonctionnement de plusieurs ressources et services, dépendants les uns des autres, sur le même serveur ?

On peut utiliser Heartbeat sur un serveur physique divisé en plusieurs serveurs virtuels.

17. Installez Heartbeat, configurez les deux fichiers principaux et accordez les droits nécessaires.

```
root@serweb1:/home/sistr# apt-get install heartbeat
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
heartbeat est déjà la plus récente version disponible.
0 mis à jour, 0 nouvellement installés, 0 à enlever et 334 non mis à jour.
root@serweb1:/home/sistr#
```

Une fois les paquets installés ou mis à jour, il faut créer les fichiers de configuration qui se trouvent dans `/etc/ha.d/` et dans `/var/lib/heartbeat`.
On crée le fichier de configuration de Pacemaker.



```
root@serweb1:/home/sisr# cat /etc/ha.d/ha.cf
#gestion des logs
logfacility local7 # fichier de log au niveau debug
logfile /var/log/ha-log
debugfile /var/log/ha-debug
use_logd no

udpport 694 #ports pour battement du coeur
keepalive 2 # Delai entre 2 battements
deadtime 20 # temps necessaire en sec avant de declaré mort

initdead 50 # valeur utilisée pour le démarrage (2x deadtime min)

auto_failback off
# gestion du retour automatique d'une ou plusieurs ressources
# le noeud primaire sera géré par Pacemaker et non pas par heartbeat

# liens des noeuds participants au cluster
node serweb1
node serweb2

# liens réseau utilisé pour les battement
ucast eth0 10.10.1.101
ucast eth0 10.10.1.102

crm yes
# format de configuration des ressources qui exploite un fichier de conf xml
# et le gestion du Cluster est réal par Pacemaker

# Si "crm no" : heartbeat exploite un fichier de conf /etc/ha.d/haressources
# dand une version de comptabilité mais plus limités
On créer le fichier /etc/ha.d/authkeys
Il détermine le niveau de sécurité des échanges entre les différents nœuds du cluster. Lorsque ces
informations passent par le réseau, il vaut mieux encrypter les échanges avec md5 en sha
root@serweb1:/home/sisr# cat /etc/ha.d/authkeys
#Sécurisation des échanges
auth 1
1 sha1 PassPhrase
_
Par mesure de sécurité, les droits sur ce fichier doivent être réduits à la lecture et écriture
seulement à l'utilisateur root (si ce n'est pas fait, Heartbeat ne démarre pas)
root@serweb1:/home/sisr# chmod 600 /etc/ha.d/authkeys
```

Le fichier `/var/lib/heartbeat/crm/cib.xml` est généré au démarrage de Heartbeat à partir des deux fichiers vus précédemment et est complété au fur et à mesure des ajouts des ressources. Le fichier ne doit JAMAIS être modifié directement mais avec des commandes spécifiques.

18. Assurez-vous que les noms d'hôtes présents dans les fichiers de configuration puissent être convertis en adresse IP via `/etc/hosts` et/ou `DNS`.

Par mesure de simplification, on peut utiliser `/etc/hosts` :

```
10.10.R.101 serweb1
10.10.R.102 serweb2
```

On peut configurer les fichiers relatifs à `bind9`



```
root@serweb1:/home/sisr# cat /etc/bind/ecole.eni
;
; BIND data file for local loopback interface
;
$TTL      604800
@         IN      SOA      localhost. root.localhost. (
                        2          ; Serial
                        604800     ; Refresh
                        86400      ; Retry
                        2419200    ; Expire
                        604800 )   ; Negative Cache TTL
;
@         IN      NS       localhost.
sisr      IN      A        10.10.1.100
slam      IN      A        10.10.1.100
serweb1   IN      A        10.10.1.101
serweb2   IN      A        10.10.1.102

root@serweb1:/home/sisr# cat /etc/bind/ecole.eni_inv
;
; BIND reverse data file for local loopback interface
;
$TTL      604800
@         IN      SOA      localhost. root.localhost. (
                        1          ; Serial
                        604800     ; Refresh
                        86400      ; Retry
                        2419200    ; Expire
                        604800 )   ; Negative Cache TTL
;
@         IN      NS       localhost.
100       IN      PTR      sisr
100       IN      PTR      slam
101       IN      PTR      serweb1
102       in      PTR      serweb2
```




```
root@serweb1:/home/sisr# service bind9 restart
[ ok ] Stopping domain name service...: bind9.
[ ok ] Starting domain name service...: bind9.
root@serweb1:/home/sisr# nslookup serweb1.ecole.eni
Server:          127.0.0.1
Address:         127.0.0.1#53
```

```
Name:   serweb1.ecole.eni
Address: 10.10.1.101
```

```
root@serweb1:/home/sisr# nslookup serweb2.ecole.eni
Server:          127.0.0.1
Address:         127.0.0.1#53
```

```
Name:   serweb2.ecole.eni
Address: 10.10.1.102
```

```
root@serweb1:/home/sisr# nslookup 10.10.1.101
Server:          127.0.0.1
Address:         127.0.0.1#53
```

```
101.1.10.10.in-addr.arpa      name = serweb1.1.10.10.in-addr.arpa.
```

```
root@serweb1:/home/sisr# nslookup 10.10.1.102
Server:          127.0.0.1
Address:         127.0.0.1#53
```

```
102.1.10.10.in-addr.arpa      name = serweb2.1.10.10.in-addr.arpa.
```

19. Clonez le serveur primaire et procédez à la configuration IP des serveurs maître et esclave.

Nom serveur maître : serweb1 adresse ip 10.10.R.101/24 dns 127.0.0.1

```
root@serweb1:/home/sisr# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:15:5d:1f:44:31
          inet adr:10.10.1.101  Bcast:10.10.1.255  Masque:255.255.255.0
          adr inet6: fe80::215:5dff:felf:4431/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:408 errors:0 dropped:0 overruns:0 frame:0
          TX packets:138 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:50459 (49.2 KiB)  TX bytes:21924 (21.4 KiB)

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:549 errors:0 dropped:0 overruns:0 frame:0
          TX packets:549 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          RX bytes:134852 (131.6 KiB)  TX bytes:134852 (131.6 KiB)
```




Nom serveur esclave : serweb2 adresse Ip 10.10.102/24 dns 127.0.0.1

```
root@serweb1:/home/sisr# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:15:5d:1f:44:32
          inet adr:10.10.1.102  Bcast:10.10.1.255  Masque:255.255.255.0
          adr inet6: fe80::215:5dff:felf:4432/64 Scope:Lien
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:61 errors:0 dropped:0 overruns:0 frame:0
          TX packets:677 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:1000
          RX bytes:5606 (5.4 KiB)  TX bytes:57840 (56.4 KiB)

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:547 errors:0 dropped:0 overruns:0 frame:0
          TX packets:547 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          RX bytes:144759 (141.3 KiB)  TX bytes:144759 (141.3 KiB)
```

20. Démarrez les nœuds et vérifiez que Heartbeat s'est correctement lancé sur les deux nœuds.

On démarre Heartbeat et active le service sur les deux serveurs

```
root@serweb1:/home/sisr# service heartbeat start
Starting High-Availability services: Done.
```

```
root@serweb2:/home/sisr# service heartbeat start
Starting High-Availability services: Done.
```

État du serveur1

```
root@serweb1:/home/sisr# crm status
=====
Last updated: Tue Dec  1 09:16:47 2020
Last change: Tue Dec  1 09:13:51 2020 via crmd on serweb2
Stack: Heartbeat
Current DC: serweb2 (ed0d3cd3-322e-4e97-9829-92a6cc29e093) - partition with quorum
Version: 1.1.7-ee0730e13d124c3d58f00016c3376alde5323cff
2 Nodes configured, unknown expected votes
0 Resources configured.
=====
```

```
Online: [ serweb2 serweb1 ]
```

État du serveur2

```
root@serweb2:/home/sisr# crm status
=====
Last updated: Tue Dec  1 09:16:35 2020
Last change: Tue Dec  1 09:13:51 2020 via crmd on serweb2
Stack: Heartbeat
Current DC: serweb2 (ed0d3cd3-322e-4e97-9829-92a6cc29e093) - partition with quorum
Version: 1.1.7-ee0730e13d124c3d58f00016c3376alde5323cff
2 Nodes configured, unknown expected votes
0 Resources configured.
=====
```

```
Online: [ serweb2 serweb1 ]
```

21. Vérifiez la configuration de Heartbeat sur chaque nœud, désactivez STONITH et le quorum.

Pour voir la liste des nœuds



```
root@serweb1:/home/sisr# cl_status listnodes
serweb2
serweb1
```

Pour vérifier l'état d'un nœud

```
|root@serweb1:/home/sisr# cl_status nodestatus serweb1
active
|root@serweb1:/home/sisr# cl_status nodestatus serweb2
active
```

Désactivation de STONITH

Stonith (Shoot The Other Member/Machine In The Head) permet d'isoler les nœuds morts et cela afin d'éviter que ces derniers corrompent le cluster. Lorsqu'un nœud est corrompu, l'hôte défaillant est redémarré ou éteint.

```
|root@serweb1:/home/sisr# crm configure property stonith-enabled=false
root@serweb2:/home/sisr# crm configure property stonith-enabled=false
```

Désactivation de quorum

Le quorum est le nombre minimum de nœuds en ligne pour être capable de valider une décision.

Dans le cas d'un cluster avec Pacemaker, il faut que plus de la moitié des nœuds soit en ligne. Avec un cluster à deux nœuds, il n'y a plus de quorum dès qu'un nœud est perdu du fait que notre cluster est composé de deux nœuds, car le fonctionnement de Pacemaker est de stopper les processus si la moitié des clusters sont inopérants.

```
root@serweb1:/home/sisr# crm configure property no-quorum-policy="ignore"
|root@serweb2:/home/sisr# crm configure property no-quorum-policy="ignore"
```

22. Vérifiez de nouveau la configuration ainsi que le fichier xml généré.

```
root@serweb1:/home/sisr# crm configure show
node $id="140201aa-3e92-4819-9e93-39619f51e14c" serweb1
node $id="ed0d3cd3-322e-4e97-9829-92a6cc29e093" serweb2
property $id="cib-bootstrap-options" \
    dc-version="1.1.7-ee0730e13d124c3d58f00016c3376a1de5323c ff" \
    cluster-infrastructure="Heartbeat" \
    stonith-enabled="false" \
    no-quorum-policy="ignore"
```



Le fichier de configuration xml généré

```
root@serweb1:/home/sisr# cat /var/lib/heartbeat/crm/cib.xml
<cib epoch="7" num_updates="0" admin_epoch="0" validate-with="pacemaker-1.2" crm_feature_set="3
.0.6" update-origin="serweb1" update-client="cibadmin" cib-last-written="Tue Dec 1 10:39:09 20
20" have-quorum="1" dc-uuid="ed0d3cd3-322e-4e97-9829-92a6cc29e093">
  <configuration>
    <crm_config>
      <cluster_property_set id="cib-bootstrap-options">
        <nvpair id="cib-bootstrap-options-dc-version" name="dc-version" value="1.1.7-ee0730e13d
124c3d58f00016c3376alde5323cff"/>
        <nvpair id="cib-bootstrap-options-cluster-infrastructure" name="cluster-infrastructure"
value="Heartbeat"/>
        <nvpair id="cib-bootstrap-options-stonith-enabled" name="stonith-enabled" value="false"
/>
        <nvpair id="cib-bootstrap-options-no-quorum-policy" name="no-quorum-policy" value="igno
re"/>
      </cluster_property_set>
    </crm_config>
    <nodes>
      <node id="ed0d3cd3-322e-4e97-9829-92a6cc29e093" type="normal" uname="serweb2"/>
      <node id="140201aa-3e92-4819-9e93-39619f51e14c" type="normal" uname="serweb1"/>
    </nodes>
    <resources/>
    <constraints/>
  </configuration>
</cib>root@serweb1:/home/sisr#
```

23. Créez et vérifiez la ressource permettant le « failover IP ». En option, vous créez un moniteur pour cette ressource avec un test de la ressource de 10 secondes, un timeout de démarrage de 30 secondes et un timeout d'arrêt de 40 secondes.

Pour s'assurer de la haute disponibilité, il faut configurer un paramètre avec une valeur "primitive" dans le CiB(Cluster Information Base).

Cette configuration peut être

- De type OCF développé pour Pacemaker et sont, pour la plupart dans le répertoire /usr/lib/ocf/resource.d/heartbeat/ou /ocf heartbeat et crm ra mist ocf pacemaker.
- De type LSB: script de démarrage sur Linux présent dans /etc/init.d.

La failover d'IP est le basculement de l'adresse IP virtuelle du serveur maître vers le serveur esclave si le premier est défaillant.

```
root@serweb1:/home/sisr# crm configure primitive IPFailover ocf:heartbeat:IPAddr
2 params ip="10.10.1.100" _cidr_netmask="255.255.255.0" nic="eth0" iflabel="VIP"
```

24. Sur quel nœud la ressource a-t-elle été lancée ? Justifiez et vérifiez.

```
root@serweb1:/home/sisr# crm status
=====
Last updated: Mon Dec 7 11:02:10 2020
Last change: Mon Dec 7 10:53:53 2020 via cibadmin on serweb1
Stack: Heartbeat
Current DC: serweb2 (83c3647d-2375-4a93-980b-3769d88aec84) - partition with quorum
Version: 1.1.7-ee0730e13d124c3d58f00016c3376alde5323cff
2 Nodes configured, unknown expected votes
1 Resources configured.
=====

Online: [ serweb2 serweb1 ]

IPFailover (ocf::heartbeat:IPAddr2): Started serweb1
root@serweb1:/home/sisr#
```

25. Migrez éventuellement la ressource (si elle n'y était pas déjà) sur le nœud primaire ; vérifiez que la ressource passe bien sur ce nœud et que la préférence a été donnée à ce nœud.



RAS

26. Testez l'accès au service Web à partir de l'adresse IP virtuelle.

Sur serweb1

Application SISR exemple d'application php-mysql : vous entrez un pseudo et un message qui seront enregistrés dans la base de données sio

Pseudo :

Message :

Clément envoie le message: est très gentil

Hamid envoie le message: Est méchant

applitest SLAM

slam.ecole.eni

Application SLAM exemple d'application php-mysql : vous entrez un pseudo et un message qui seront enregistrés dans la base de données sio

Pseudo :

Message :

```
root@serweb1:/home/sisr# nslookup sisr.ecole.eni
Server:          127.0.0.1
Address:         127.0.0.1#53

Name:   sisr.ecole.eni
Address: 10.10.1.100

root@serweb1:/home/sisr# nslookup slam.ecole.eni
Server:          127.0.0.1
Address:         127.0.0.1#53

Name:   slam.ecole.eni
Address: 10.10.1.100
```

Sur serweb2

applitest SISR

sisr.ecole.eni

Application SISR exemple d'application php-mysql : vous entrez un pseudo et un message qui seront enregistrés dans la base de données sio

Pseudo :

Message :

Clément envoie le message: est très gentil

Hamid envoie le message: Est méchant



applitest SLAM x +

slam.ecole.eni

Rechercher

Application SLAM exemple d'application php-mysql : vous entrez un pseudo et un message qui seront enregistrés dans la base de données sio

Pseudo :

Message :

Envoyer

```
root@serweb2:/home/sisr# nslookup slam.ecole.eni
Server:      127.0.0.1
Address:     127.0.0.1#53
```

```
Name:  slam.ecole.eni
Address: 10.10.1.100
```

```
root@serweb2:/home/sisr# nslookup sisr.ecole.eni
Server:      127.0.0.1
Address:     127.0.0.1#53
```

```
Name:  sisr.ecole.eni
Address: 10.10.1.100
```

27. Quel autre service est obligatoirement impacté par le changement d'adresse IP ?
L'autre service impacté est le service DNS qui fait correspondre le nom DNS à une adresse IP.

28. Testez le failover IP des deux manières présentées en annexe 2. Y a-t-il une différence ?

1ère méthode:

On arrête un nœud avec `crm node standby` ici le serweb1

```
|root@serweb1:/home/sisr# crm node standby
```

On vérifie l'état du cluster sur serweb1 et serweb2

```
root@serweb1:/home/sisr# crm status
```

```
=====
```

```
Last updated: Mon Dec  7 13:05:24 2020
```

```
Last change: Mon Dec  7 13:03:57 2020 via crm_attribute on serweb1
```

```
Stack: Heartbeat
```

```
Current DC: serweb1 (9f66c5df-00bf-4284-aea2-031ca8b8308d) - partition with quorum
```

```
Version: 1.1.7-ee0730e13d124c3d58f00016c3376alde5323cff
```

```
2 Nodes configured, unknown expected votes
```

```
1 Resources configured.
```

```
=====
```

```
Node serweb1 (9f66c5df-00bf-4284-aea2-031ca8b8308d): standby
```

```
Online: [ serweb2 ]
```

```
IPFailover      (ocf::heartbeat:IPaddr2):      Started serweb2
```




```
root@serweb2:/home/sisr# crm status
```

```
=====
```

```
Last updated: Mon Dec  7 13:05:55 2020
```

```
Last change: Mon Dec  7 13:03:57 2020 via crm_attribute on serweb1
```

```
Stack: Heartbeat
```

```
Current DC: serweb1 (9f66c5df-00bf-4284-aea2-031ca8b8308d) - partition with quorum
```

```
Version: 1.1.7-ee0730e13d124c3d58f00016c3376alde5323cff
```

```
2 Nodes configured, unknown expected votes
```

```
1 Resources configured.
```

```
=====
```

```
Node serweb1 (9f66c5df-00bf-4284-aea2-031ca8b8308d): standby
```

```
Online: [ serweb2 ]
```

```
IPFailover (ocf::heartbeat:IPaddr2): Started serweb2
```

On pense à réactiver le nœud serweb1 avec crm node online

```
root@serweb1:/home/sisr# crm node online
```

2ème méthode:

On arrête le service Heartbeat sur l'un des deux serveurs

```
root@serweb2:/home/sisr# service heartbeat stop
```

```
Stopping High-Availability services: Done.
```

```
root@serweb2:/home/sisr# crm status
```

```
Connection to cluster failed: connection failed
```

```
root@serweb1:/home/sisr# crm status
```

```
=====
```

```
Last updated: Mon Dec  7 13:12:48 2020
```

```
Last change: Mon Dec  7 13:06:45 2020 via crm_attribute on serweb1
```

```
Stack: Heartbeat
```

```
Current DC: serweb1 (9f66c5df-00bf-4284-aea2-031ca8b8308d) - partition with quorum
```

```
Version: 1.1.7-ee0730e13d124c3d58f00016c3376alde5323cff
```

```
2 Nodes configured, unknown expected votes
```

```
1 Resources configured.
```

```
=====
```

```
Online: [ serweb1 ]
```

```
OFFLINE: [ serweb2 ]
```

```
IPFailover (ocf::heartbeat:IPaddr2): Started serweb1
```

On redémarre le service

```
root@serweb2:/home/sisr# service heartbeat start
```

```
Starting High-Availability services: Done.
```

29. Sur les deux nœuds, arrêtez le service HTTP et supprimez ce service du démarrage.

On arrête le service

```
root@serweb1:/home/sisr# service apache2 stop
```

```
[ ok ] Stopping web server: apache2.
```

```
root@serweb1:/home/sisr# █
```

```
root@serweb2:/home/sisr# service apache2 stop
```

```
[ ok ] Stopping web server: apache2 ... waitir
```

```
root@serweb2:/home/sisr# █
```

On désactive les services



```
root@serweb1:/home/sisr# update-rc.d apache2 disable
update-rc.d: using dependency based boot sequencing
insserv: warning: current start runlevel(s) (empty) of script `apache2' overrides LSB defaults (2 3 4 5).
insserv: warning: current stop runlevel(s) (0 1 2 3 4 5 6) of script `apache2' overrides LSB defaults (0 1 6).

root@serweb2:/home/sisr# update-rc.d apache2 disable
update-rc.d: using dependency based boot sequencing
insserv: warning: current start runlevel(s) (empty) of script `apache2' overrides LSB defaults (2 3 4 5).
insserv: warning: current stop runlevel(s) (0 1 2 3 4 5 6) of script `apache2' overrides LSB defaults (0 1 6).
```

30. Créez la ressource « serviceWeb » et vérifiez sur quel nœud elle démarre.

On crée une ressource nommée «serviceWeb» avec comme paramètre du fichier de configuration d'Apache2.

```
root@serweb1:/home/sisr# crm configure primitive serviceWeb ocf:heartbeat:apache params configfile="/etc/apache2/apache2.conf"
" op monitor interval="60s" op start timeout="40s" op stop timeout="40s"
```

31. Vérifiez que le service HTTP soit bien démarré sur le serveur spécifié et non activé sur l'autre.

```
root@serweb1:/home/sisr# crm resource move IPFailover serweb1
root@serweb1:/home/sisr# crm resource move serviceWeb serweb1
root@serweb1:/home/sisr# crm status
=====
Last updated: Mon Dec  7 14:28:55 2020
Last change: Mon Dec  7 14:28:40 2020 via crm_resource on serweb1
Stack: Heartbeat
Current DC: serweb2 (57a4a1fb-7c9d-48c5-9779-b01b212c5589) - partition with quorum
Version: 1.1.7-ee0730e13d124c3d58f00016c3376alde5323cff
2 Nodes configured, unknown expected votes
2 Resources configured.
=====

Online: [ serweb2 serweb1 ]

IPFailover      (ocf::heartbeat:IPAddr2):      Started serweb1
serviceWeb      (ocf::heartbeat:apache):      Started serweb1
```

32. Groupez les ressources de manière à ce qu'elles soient actives sur le même nœud.

```
root@serweb1:/home/sisr# crm configure group groupweb IPFailover serviceWeb meta migration-threshold="5"
INFO: resource references in location:cli-prefer-IPFailover updated
INFO: resource references in location:cli-prefer-serviceWeb updated
WARNING: cli-prefer-IPFailover: referenced node serweb1 does not exist
WARNING: cli-prefer-serviceWeb: referenced node serweb1 does not exist
root@serweb1:/home/sisr# crm status
=====
Last updated: Mon Dec  7 14:33:07 2020
Last change: Mon Dec  7 14:32:24 2020 via cibadmin on serweb1
Stack: Heartbeat
Current DC: serweb1 (9f66c5df-00bf-4284-aea2-031ca8b8308d) - partition with quorum
Version: 1.1.7-ee0730e13d124c3d58f00016c3376alde5323cff
2 Nodes configured, unknown expected votes
2 Resources configured.
=====

Online: [ serweb2 serweb1 ]

Resource Group: groupweb
  IPFailover (ocf::heartbeat:IPAddr2):      Started serweb1
  serviceWeb (ocf::heartbeat:apache):      Started serweb1
```

33. Quels sont les tests que vous effectuez pour vérifier la solution mise en place ?

On vérifie l'état du service web sur les deux nœuds avec la commande *netstat -tlnl | grep apache:*
Sur serweb1



```
root@serweb1:/home/sisr# netstat -tlnl | grep apache
tcp6      0      0 :::80          :::*           LISTEN     11448/apache2
```

Sur serweb2

```
root@serweb2:/home/sisr# netstat -tlnl | grep apache
root@serweb2:/home/sisr#
```

Serweb2 ne renvoie rien puisqu'il est actuellement un serveur secondaire

Pour tester la redondance on va désactiver les ressources IPFailover et serviceWeb de serweb1

```
root@serweb1:/home/sisr# crm node standby
root@serweb2:/home/sisr# crm status
=====
Last updated: Mon Dec  7 14:44:48 2020
Last change: Mon Dec  7 14:44:37 2020 via crm_attribute on serweb1
Stack: Heartbeat
Current DC: serweb1 (9f66c5df-00bf-4284-aea2-031ca8b8308d) - partition with quorum
Version: 1.1.7-ee0730e13d124c3d58f00016c3376alde5323cff
2 Nodes configured, unknown expected votes
2 Resources configured.
=====

Node serweb1 (9f66c5df-00bf-4284-aea2-031ca8b8308d): standby
Online: [ serweb2 ]
```

```
Resource Group: groupweb
  IPFailover (ocf::heartbeat:IPAddr2):      Started serweb2
  serviceWeb (ocf::heartbeat:apache):      Started serweb2
```

La commande *netstat -tlnl | grep apache* renvoie alors quelque chose sur serweb2 car il a repris la relève.

```
root@serweb2:/home/sisr# netstat -tlnl | grep apache
tcp6      0      0 :::80          :::*           LISTEN     9754/apache2
```

34. Écrivez la procédure détaillée que vous allez mettre en œuvre pour configurer la réplication de la base de données.

- Installation des services de base de données sur les serveurs.
- Autoriser la réplication de bases de données entre plusieurs serveurs: existence d'un utilisateur MySQL ayant des droits suffisants sur le serveur maître.
- `mysql> GRANT REPLICATION SLAVE ON *.* TO replicateur@%' IDENTIFIED BY 'mdprePLICATEUR';`
- Mise en place des bases de données à répliquer qui doivent être identiques sur tous les serveurs avant de commencer la réplication. Il faut qu'aucune nouvelle donnée ne soit enregistrée pendant le laps de temps nécessaire à la configuration: bloquer l'écriture sur la ou les base(s) que l'on souhaite répliquer (sur le serveur web: **FLUSH TABLES WITH READ LOCK**).
- Configuration de la réplication via le fichier `/etc/mysql/my.cnf` sur chaque serveur (maître et esclave): la réplication de bases de données implique une configuration différente sur chaque serveur, en fonction de son rôle.
- Déclaration du serveur maître au serveur esclave
Débloquer l'écriture sur le serveur maître (UNLOCK TABLES.)

35. Mettez en œuvre la réplication selon cette procédure. (configuration1 maître-esclave)

- Installation de MySQL

- serweb1

`apt-get install mysql-server mysql-client`

démarrage du service avec la commande `service mysql start`.

- serweb2

`apt-get install mysql-server mysql-client`



démarrage du service avec la commande `service mysql start`.

- Sur le serveur maître (serweb1)

- Création de l'utilisateur réplicateur ayant les droits suffisants

```
root@serweb1:/home/sisr# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 51
Server version: 5.5.46-0+deb7u1 (Debian)

Copyright (c) 2000, 2015, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> GRANT REPLICATION SLAVE ON *.* TO replicateur@%' IDENTIFIED BY 'mdpreplicateur'
-> ;
Query OK, 0 rows affected (0.00 sec)
```

- On vide les caches des privilèges et des tables, tout en les verrouillant en lecture seule, à l'aide de la commande FLUSH :

```
mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> FLUSH TABLES WITH READ LOCK;
Query OK, 0 rows affected (0.25 sec)
```

- On affiche le status du serveur maître et on prend note de la position et du nom de fichier

```
mysql> show master status
-> ;
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mysql-bin.000001 | 107      | sistr,siam    |                   |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

- On modifie le fichier `/etc/mysql/my.cnf` de cette sorte:

```
# localhost which is more compatible
+1 #bind-address          = 127.0.0.1
server-id                = 1
log-error=/var/log/mysql.err
log_bin                  = /var/log/mysql/mysql-bin.log
expire_logs_days         = 10
max_binlog_size          = 100M
binlog_do_db             = sio
```

- Sur le serveur esclave (serweb2)

- On modifie le fichier `/etc/mysql/my.cnf` de cette sorte:

```
# localhost which is more compatible
+1 #bind-address          = 127.0.0.1
```



```
server-id          = 99
log-error          = /var/log/mysql.err
log_bin           = /var/log/mysql/mysql-bin.log
expire_logs_days  = 10
max_binlog_size   = 100M
master-retry-count = 20
binlog_do_db      = sio
log-slave-updates
```

- on déclare le serveur maître

```
|mysql> change master to master_host='10.10.1.101', master_user='replicateur', master_p
|assword='mdpreplicateur', master_log_file='mysql-bin.000001', master_log_pos=107;
```

- on démarre l'esclave et on vérifie son état:

```
|mysql> start slave;
```



```
mysql> show slave status \G
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
      Master_Host: 10.10.1.101
      Master_User: replicateur
      Master_Port: 3306
      Connect_Retry: 60
      Master_Log_File: mysql-bin.000001
      Read_Master_Log_Pos: 107
      Relay_Log_File: mysqld-relay-bin.000002
      Relay_Log_Pos: 253
      Relay_Master_Log_File: mysql-bin.000001
      Slave_IO_Running: Yes
      Slave_SQL_Running: Yes
      Replicate_Do_DB:
      Replicate_Ignore_DB:
      Replicate_Do_Table:
      Replicate_Ignore_Table:
      Replicate_Wild_Do_Table:
      Replicate_Wild_Ignore_Table:
      Last_Errno: 0
      Last_Error:
      Skip_Counter: 0
      Exec_Master_Log_Pos: 107
      Relay_Log_Space: 410
      Until_Condition: None
      Until_Log_File:
      Until_Log_Pos: 0
      Master_SSL_Allowed: No
      Master_SSL_CA_File:
      Master_SSL_CA_Path:
      Master_SSL_Cert:
      Master_SSL_Cipher:
      Master_SSL_Key:
      Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
      Last_IO_Errno: 0
      Last_IO_Error:
      Last_SQL_Errno: 0
      Last_SQL_Error:
      Replicate_Ignore_Server_Ids:
      Master_Server_Id: 1
1 row in set (0.00 sec)
```

- **Après cela on débloquent les tables de serweb1**

```
mysql> unlock tables;
```

- **On redémarre ensuite le service mysql sur les deux serveurs**

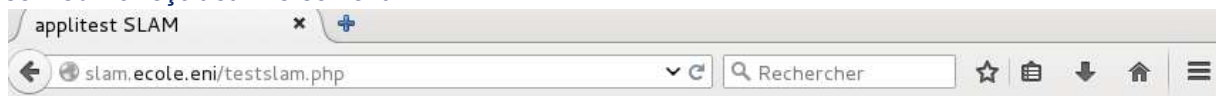


```
root@serweb1:/home/sisr# service mysql restart
[ ok ] Stopping MySQL database server: mysqld.
[ ok ] Starting MySQL database server: mysqld ..
[info] Checking for tables which need an upgrade, are corrupt or were
not closed cleanly..

root@serweb2:/home/sisr# service mysql restart
[ ok ] Stopping MySQL database server: mysqld.
[ ok ] Starting MySQL database server: mysqld ..
[info] Checking for tables which need an upgrade, are corrupt or were
not closed cleanly..
```

36. Proposez et réalisez des tests permettant de vérifier l'opérationnalité de la solution1.

À l'aide du site web slam.ecole.eni on se rend compte que si on crée une requête sur serweb1, serweb2 la reçoit et inversement



Application SLAM exemple d'application php-mysql : vous entrez un pseudo et un message qui seront enregistrés dans la base de données sio

Pseudo :

Message :

Lekbouri envoie le message: Domaine de diffusion de serweb1

Lekbouri envoie le message: Domaine de diffusion de serweb2

37. Transformez l'architecture maître-esclave en architecture multi maître afin de ne pas perdre de données lorsque le serveur maître sera de nouveau opérationnel après une défaillance. Vous rédigerez au préalable une procédure.

- On crée l'utilisateur réplicateur ayant des droits suffisants sur serweb2
- On s'assure que les bases de données soient identiques entre les deux serveurs
- On modifie les fichiers de configuration /etc/mysql/my.cnf dans serweb1 et serweb2 et cela afin que les deux soit esclave et maître à la fois
- On redémarre le service MySQL pour que les modifications soient prises en compte
- On bloque l'écriture sur les bases du serveur maître serweb2.
- On récupère le log binaire et la position du jeu de commandes dans serweb2
- Ainsi on déclare sur serweb1 le nouveau maître de serweb2
- On teste la configuration des deux serveurs
- On débloque les deux tables sur les deux serweb

- On crée l'utilisateur réplicateur ayant des droits suffisants sur serweb2

```
mysql> GRANT REPLICATION SLAVE ON *.* TO replicateur@%' IDENTIFIED BY 'mdpreplicateur';
Query OK, 0 rows affected (0.00 sec)
```

- On s'assure que les bases de données soient identiques entre les deux serveurs
- On modifie les fichiers de configuration /etc/mysql/my.cnf dans serweb1 et serweb2 et cela afin que les deux soit esclave et maître à la fois

Serweb1

```
# bind-address = 127.0.0.1
```



```
server-id          = 1
log-error          = /var/log/mysql.err
log_bin            = /var/log/mysql/mysql-bin.log
expire_logs_days   = 10
max_binlog_size     = 100M
master-retry-count = 20

|binlog_do_db      = sio|
replicate_do_db    = sio_
log-slave-updates
```

Serweb2

```
# bind-address      = 127.0.0.1
server-id          = 2
log-error          = /var/log/mysql.err
log_bin            = /var/log/mysql/mysql-bin.log
expire_logs_days   = 10
max_binlog_size     = 100M
master-retry-count = 20
|binlog_do_db      = sio|
replicate_do_db    = sio_
log-slave-updates
```

- On redémarre le service MySQL pour que les modifications soient prises en compte
- On bloque l'écriture sur les bases du serveur maître serweb2.

```
mysql> FLUSH TABLES WITH READ LOCK;
Query OK, 0 rows affected (0.15 sec)
```

- On récupère le log binaire et la position du jeu de commandes dans serweb2

```
mysql> show master status
-> ;
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mysql-bin.000002 |      262 | sio,sio_     |                   |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

- Ainsi on déclare sur serweb1 le nouveau maître de serweb2

```
mysql> change master to master_host='10.10.1.102', master_user='replicateur', master_password='mdprelicateur', master_log_file='mysql-bin.000002', master_log_pos=262;
Query OK, 0 rows affected (0.59 sec)
```

- On teste la configuration des deux serveurs, on utilise la commande `show slave status \G`; et on s'assure que les deux serveurs renvoient "Yes" à `Slave_IO_Running`; et `Slave_SQL_Running`;

Serweb1

```
Slave_IO_Running: Yes
Slave_SQL_Running: Yes
```

Serweb2



Slave_IO_Running: Yes
Slave_SQL_Running: Yes

- On débloquent les deux tables sur les deux serveurs

Serveur1

```
mysql> unlock tables;  
Query OK, 0 rows affected (0.00 sec)
```

Serveur2

```
mysql> unlock tables;  
Query OK, 0 rows affected (0.00 sec)
```

38. Proposez et réalisez des tests permettant de vérifier l'opérationnalité de la solution.

On tente de rajouter des informations dans la base de données à partir de serveur1 pendant que le service mysql de serveur2 est désactivé. On redémarre après serveur2 et on vérifie si les changements fonctionnent

```
root@serveur2:/home/sisr# service mysql stop  
[ ok ] Stopping MySQL database server: mysqld.
```

À partir de serveur1

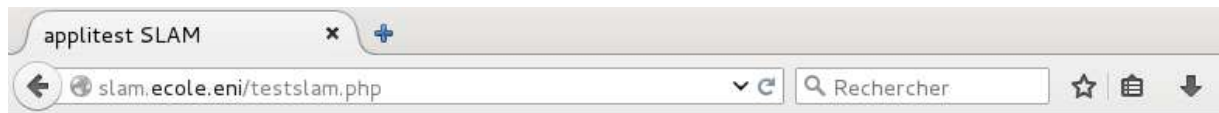


Application SLAM exemple d'application php-mysql : vous entrez un pseudo et un message qui seront enregistrés dans la base de données

Pseudo :

Message :

Lekbouri envoie le message: Domaine de diffusion de serveur1 sans que serveur2 soit actif
Sur serveur2 après le redémarrage du service



Application SLAM exemple d'application php-mysql : vous entrez un pseudo et un message, seront enregistrés dans la base de données

Pseudo :

Message :

Lekbouri envoie le message: Domaine de diffusion de serveur1 sans que serveur2 soit actif

39. Intégrez cette solution au Cluster.

- Sur chaque serveur :

```
root@serveur1:/home/sisr# insserv -r -v mysql  
insserv: remove service /etc/init.d/./rc0.d/K02mysql  
insserv: remove service /etc/init.d/./rc1.d/K02mysql  
insserv: remove service /etc/init.d/./rc2.d/S02mysql  
insserv: remove service /etc/init.d/./rc3.d/S02mysql  
insserv: remove service /etc/init.d/./rc4.d/S02mysql  
insserv: remove service /etc/init.d/./rc5.d/S02mysql  
insserv: remove service /etc/init.d/./rc6.d/K02mysql  
insserv: creating .depend.boot  
insserv: creating .depend.start  
insserv: creating .depend.stop
```

```
root@serveur2:/home/sisr# insserv -r -v mysql  
insserv: remove service /etc/init.d/./rc0.d/K02mysql  
insserv: remove service /etc/init.d/./rc1.d/K02mysql  
insserv: remove service /etc/init.d/./rc2.d/S02mysql  
insserv: remove service /etc/init.d/./rc3.d/S02mysql  
insserv: remove service /etc/init.d/./rc4.d/S02mysql  
insserv: remove service /etc/init.d/./rc5.d/S02mysql  
insserv: remove service /etc/init.d/./rc6.d/K02mysql  
insserv: creating .depend.boot  
insserv: creating .depend.start  
insserv: creating .depend.stop
```

- Sur un des deux nœuds :



```
root@serweb1:/home/sisr# crm configure primitive serviceMySQL ocf:heartbeat:mysql params
socket="/var/run/mysqld/mysqld.sock"
```

```
WARNING: serviceMySQL: default timeout 20s for start is smaller than the advised 120
```

```
WARNING: serviceMySQL: default timeout 20s for stop is smaller than the advised 120
```

- Cloner avec la commande

```
root@serweb1:/home/sisr# crm configure clone cserviceMySQL serviceMySQL
```

40. Proposez et réalisez des tests permettant de vérifier l'opérationnalité de la solution complète en complétant votre tableau, faire les tests à partir d'un client différent des 2 serveurs.

Actions à effectuer	Résultats attendus	Résultats obtenus	Statut
On stoppe le nœud serweb1.	Les ressources (mise à part la ressource service MySQL) migrent sur le nœud serweb2.	Node serweb1 (5f65c5ef-00ef-4284-aea2-091ca8b8388d): standby OnLine: [serweb2] Resource Group: groupweb IPFailover (ocf:heartbeat:IPaddr2): Started serweb2 serviceWeb (ocf:heartbeat:apache): Started serweb2 Clone Set: cserviceMySQL (serviceMySQL) Started: [serweb2] Stopped: [serweb1]	OK
On ajoute une entrée dans la base de donnée	Une entrée dans la table SLAM est faite	mysql> select * from slam; +-----+-----+-----+ id pseudo message +-----+-----+-----+ 12 Lekbouri Les domaines de diffusions +-----+-----+-----+ 1 row in set (0.00 sec)	OK
On redémarrage le nœud serweb1	Les ressources migrent sur ce nœud et les modifications apportées sur la base de données sont visibles.	OnLine: [serweb2 serweb1] Resource Group: groupweb IPFailover (ocf:heartbeat:IPaddr2): Started serweb1 serviceWeb (ocf:heartbeat:apache): Started serweb1 Clone Set: cserviceMySQL (serviceMySQL) Started: [serweb1 serweb2] mysql> select * from slam; +-----+-----+-----+ id pseudo message +-----+-----+-----+ 12 Lekbouri Les domaines de diffusions +-----+-----+-----+ 1 row in set (0.00 sec)	OK

41. Adapter le projet à votre contexte.