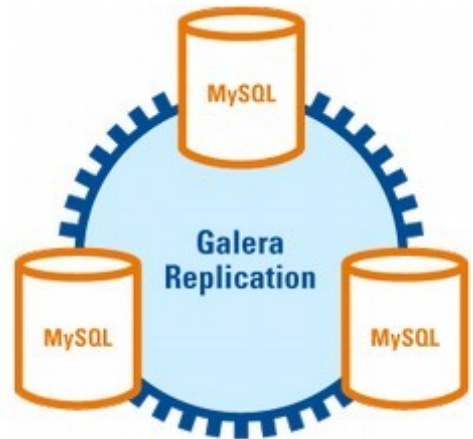


*Servidor Web HA: Galera, Pacemaker, Corosync, Heartbeat, GlusterFS, Apache*



## Índice de contenido

1.1	Introducción a la alta disponibilidad.....	3
1.2	Objetivo del proyecto.....	3
2.1	Galera.....	3
2.1.1	Introducción a Galera.....	3
2.1.2	Instalación de Galera.....	4
2.1.3	Configuración de Galera.....	4
2.2	Apache.....	6
2.2.1	Software Apache.....	6
2.2.1	Instalación y configuración de Apache.....	6
2.3	Pacemaker, Corosync, Heartbeat.....	6
2.3.1	Introducción al software.....	6
2.3.2	Instalación y configuración del clúster.....	7
2.3.3	Configuración recurso VIP.....	10
2.3.4	Añadiendo el recurso de Apache.....	11
2.4	GlusterFS.....	11
2.4.1	Introducción a GlusterFS.....	11
2.4.2	Instalación de GlusterFS.....	12
2.4.2	Configuración de GlusterFS.....	12
2.5	Drupal.....	13
2.5.1	Introducción a Drupal.....	13
2.5.2	Instalación de Drupal.....	13
2.5.2.1	Configuración de la base de datos.....	13
2.5.2.2	Configuración de Drupal.....	14
3.	Errores surgidos.....	14
4.	Bibliografía.....	15

# 1. Introducción al objetivo

## 1.1 Introducción a la alta disponibilidad

Alta disponibilidad (High availability) hace referencia a un protocolo que asegura un cierto grado absoluto de continuidad operacional durante un período de medición dado. Disponibilidad se refiere a la habilidad de la comunidad de usuarios para acceder al sistema, someter nuevos trabajos, actualizar o alterar trabajos existentes o recoger los resultados de trabajos previos.

Si un usuario no puede acceder al sistema se dice que está no disponible. El término tiempo de inactividad (downtime) es usado para definir cuándo el sistema no está disponible.

## 1.2 Objetivo del proyecto

El objetivo de este proyecto es la creación de un servidor web en HA. Para ello utilizaremos el sistema operativo **Debian**, en el que instalaremos un software dedicado a la alta disponibilidad, un clúster de base de datos para que la información de la misma este disponible en todos los nodos, un sistema de almacenamiento que posea la misma configuración en ambos nodos y un servidor web con un CMS implementado que ofrezca un recurso.

# 2. Configuración de los componentes

## 2.1 Galera

### 2.1.1 Introducción a Galera

MariaDB Galera Cluster es un clúster de sincronización multi-maestro de MariaDB. Está disponible solo en Linux , y sólo es compatible con los motores de almacenamiento **InnoDB** y **XtraDB**.

MariaDB Galera Cluster utiliza la biblioteca Galera para implementar de replicación.

La implementación de la API de replicación en MariaDB sucede en el proyecto de MySQL - wsrep de código abierto .

## 2.1.2 Instalación de Galera

Los siguientes pasos deben hacerse en las tres máquinas

Lo primero será declarar las máquinas que formarán el clúster en el fichero `/etc/hosts`:

```
10.0.0.133 nodoha.gonzalonazareno.com nodoha
10.0.0.134 nodoha2.gonzalonazareno.com nodoha2
10.0.0.135 nodoha3.gonzalonazareno.com nodoha3
```

El siguiente paso será añadir el repositorio de MariaDB a nuestra lista de repositorios (importante realizar antes de instalar cualquier programa un **apt-get update**). Para añadir el repositorio, realizamos las siguientes instrucciones:

```
apt-get install software-properties-common

apt-key adv --recv-keys --keyserver keyserver.ubuntu.com
0xcbc082a1bb943db

add-apt-repository 'deb [arch=amd64,i386]
http://tedeco.fi.upm.es/mirror/mariadb/repo/10.0/debian jessie
main'

apt-get update
```

En el paso anterior, hemos añadido la clave de repositorio a nuestro depósito de claves y hemos añadido la línea de MariaDB a nuestro `sources.list`.

Instalamos MariaDB:

```
apt-get install mariadb-server
```

Instalamos Galera (esto nos borrará `mariadb-server`, necesitado para obtener archivos de configuración):

```
apt-get install -y rsync galera mariadb-galera-server
```

## 2.1.3 Configuración de Galera

Antes de hacer alguna modificación, debemos de asegurarnos de que el servicio `mysql` esta detenido en las tres máquinas. Realizamos un `systemctl status mysql` para comprobarlo:

Creamos el fichero independiente `/etc/mysql/conf.d/galera.cnf` con el siguiente contenido:

(Ejemplo para nodoha)

```
[mysqld]
#mysql settings
binlog_format=ROW
default-storage-engine=innodb
innodb_autoinc_lock_mode=2
query_cache_size=0
query_cache_type=0
bind-address=0.0.0.0
#galera settings
wsrep_provider=/usr/lib/galera/libgalera_smm.so
wsrep_cluster_name="warner"
wsrep_cluster_address="gcomm://10.0.0.133,10.0.0.134,10.0.0.135"
wsrep_node_address=10.0.0.133
wsrep_sst_method=rsync
```

Importante modificar `wsrep_node_address` en cada nodo.

Arrancamos mysql en el primer nodo:

```
/etc/init.d/mysql start --wsrep-new-cluster
```

Realizamos la siguiente consulta en mysql para obtener datos del cluster:

```
SELECT VARIABLE_VALUE as "cluster size"
FROM INFORMATION_SCHEMA.GLOBAL_STATUS
WHERE VARIABLE_NAME="wsrep_cluster_size";
```

```
+-----+
| cluster size |
+-----+
| 1           |
+-----+
```

Arrancamos mysql en los otros nodos y volvemos a ejecutar la consulta anterior:

```
+-----+
| cluster size |
+-----+
| 3           |
+-----+
```

## 2.2 Apache

### 2.2.1 Software Apache

El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.12.

Su nombre se debe a que alguien quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo, y la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de EEUU, y en esos momentos la preocupación de su grupo era que llegasen las empresas y "civilizasen" el paisaje que habían creado los primeros ingenieros de internet. Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. En inglés, a patchy server (un servidor "parcheado") suena igual que Apache Server.

El servidor Apache es desarrollado y mantenido por una comunidad de usuarios bajo la supervisión de la Apache Software Foundation dentro del proyecto HTTP Server (httpd).

### 2.2.1 Instalación y configuración de Apache

Vamos a instalar Apache con el componente de PHP para la posterior instalación de nuestro CMS.

Para ello, ejecutamos el siguiente comando en todos los nodos:

```
apt-get install apache2 php5 php5-gd php5-mysql
```

Por último, activamos y arrancamos apache en las tres máquinas:

```
systemctl start apache2
```

## 2.3 Pacemaker, Corosync, Heartbeat

### 2.3.1 Introducción al software

Para conseguir la Alta Disponibilidad de nuestros servicios, se detallará como llevar a cabo la configuración de tres herramientas:

- Pacemaker: Verifica el estado de los recursos (o servicios) que le han sido asignados. En caso algún servicio falle, Pacemaker puede reiniciarlo.
- Corosync: Debe configurarse para que nuestros servidores se pueden comunicar como un clúster.
- Heartbeat: Encargado de revisar que cada nodo se halle funcionando. En caso un nodo falle migrará los recursos a otro nodo que también se halle ejecutando el servicio heartbeat.

Mientras heartbeat se encarga que revisar el estado de cada nodo, Pacemaker es el responsable de verificar el estado de los servicios que deseemos sean HA dentro de los nodos.

### 2.3.2 Instalación y configuración del clúster

El primer paso será añadir a nuestros repositorios el repositorio **backports** de Debian a nuestro **sources.list**:

```
deb http://ftp.debian.org/debian jessie-backports main
```

Actualizamos los repositorios (**apt-get update**) e instalamos pacemaker (**corosync** se instala en las dependencias)

```
apt-get -t jessie-backports install pacemaker
```

A continuación, debemos de comprobar que los servicios **corosync** y **pacemaker** no están activos. Ejecutamos **systemctl stop pacemaker** y **systemctl stop corosync**.

El siguiente paso será crear una clave para la sincronización con corosync. Para ello, debemos de ejecutar el comando **corosync-keygen**.

Esta clave deberá ser copiada a los otros nodos en la ruta **/etc/corosync/authkey**. Podemos copiar la clave con scp:

```
scp /etc/corosync/authkey root@nodoha2:/etc/corosync/authkey
```

Antes de arrancar los siguientes servicios, debemos modificar/añadir las siguientes líneas del fichero **/etc/corosync/corosync.conf**.

```
totem {
    version: 2
    cluster_name: hacloud
    token: 3000
    token_retransmits_before_loss_const: 10
    clear_node_high_bit: yes
    crypto_cipher: none
    crypto_hash: none
    interface {
        ringnumber: 0
        bindnetaddr: 10.0.0.0
        mcastport: 5405
        ttl: 1
    }
}

logging {
    fileline: off
    to_stderr: no
    to_logfile: yes
    logfile: /var/log/corosync/corosync.log
    syslog_facility: daemon
    debug: off
    timestamp: on
    logger_subsys {
        subsys: QUORUM
        debug: off
    }
}
```



```

quorum {
    provider: corosync_votequorum
    expected_votes: 2
}

nodelist {
    node {
        ring0_addr: nodoha
    }
    node {
        ring0_addr: nodoha2
    }
    node {
        ring0_addr: nodoha3
    }
}

```

Como hicimos anteriormente con apache, debemos activar e iniciar los servicios pacemaker y corosync en todos los nodos:

```

systemctl enable pacemaker
systemctl start pacemaker

systemctl enable corosync
systemctl start corosync

```

Una vez arrancado, comprobamos que los nodos se comunican con **crm\_mon**:

```

Last updated: Wed Jun 15 16:30:18 2016           Last change: Tue
Jun 14 15:07:13 20
16 by root via cibadmin on nodoha
Stack: corosync
Current DC: nodoha (version 1.1.14-70404b0) - partition with
quorum

```

```
3 nodes and 3 resources configured
```

```
Online: [ nodoha nodoha2 nodoha3 ]
```

Deshabilitaremos la característica de **Stonith** en nuestro clúster con el siguiente comando (solo ejecutamos en un nodo):

(Para ejecutar comandos como **crm configure**, debemos de instalar el paquete **crmsh** adicionalmente)

```
crm configure property stonith-enabled=false
```

### 2.3.3 Configuración recurso VIP

Para crear el recurso perteneciente a IP virtual, debemos de ejecutar el siguiente comando:

(La ip seleccionada es orientativa)

```
crm configure primitive HA_VIP ocf:heartbeat:IPaddr2 ip=10.0.0.250
cidr_netmask=24 op monitor interval=20s
```

Lo siguiente será asociar la ip creada a una ip flotante de nuestro proyecto de OpenStack. La ip flotante elegida es la **172.22.206.1**

Para ello, debemos acceder a Openstack a través de la línea de comandos. Una vez dentro, debemos de realizar el siguiente conjunto de secuencias:

```
neutron net-list (Identificamos la red)
neutron port-create b9713d19-cc5f-40db-8534-5b9f49630318 --fixed-
ip ip_address=10.0.0.250 (Creamos un puerto para la ip virtual)
neutron floatingip-associate --fixed-ip-address (vip) 10.0.0.250
(id ip flotante) 055916b4-35ce-41d8-a6eb-4e30fac1723e (id ip
virtual) 11ea5e68-a219-479a-9587-e9da782688b6 (asociamos la VIP
con la IP flotante)
```

### 2.3.4 Añadiendo el recurso de Apache

Para añadir el recurso de Apache, debemos de ejecutar el siguiente comando:

```
crm_configure primitive WebServer ocf:heartbeat:apache params  
configfile="/etc/apache2/apache2.conf" op monitor timeout="20s"
```

Una vez creado, debemos de comprobar que esté iniciado con el comando **crm\_mon**. Si no es así, debemos de detener el servicio apache (**systemctl stop apache2**) de la máquina donde se este ejecutando este recurso.

Arrancamos el recurso de Apache con el comando **crm\_resource start WebServer**.

Por último, queremos asegurarnos de que ambos recursos se ejecutan en el mismo host, por lo que vamos a definir una restricción de colocación con una puntuación de INFINITY.

```
crm configure assist weak-bond WebServer HA_VIP
```

El orden de los recursos en la definición de la restricción es importante. Se especifica que el recurso de Apache ( WebServer ) debe ejecutarse en el mismo anfitrión en que la VIP ( HA\_VIP ) está activa. Esto también significa que WebServer no está autorizado a ejecutar en cualquier lugar si HA\_VIP no está activo.

## 2.4 GlusterFS

### 2.4.1 Introducción a GlusterFS

GlusterFS es un multiescalable sistema de archivos para NAS que permite agregar varios servidores de archivos sobre Ethernet en un gran entorno de archivos de red en paralelo. Se pueden encontrar siendo utilizado en una gran variedad de entornos y aplicaciones como computación en nube, ciencias biomédicas y almacenamiento de archivos. GlusterFS está licenciado bajo la licencia GNU v3.

### 2.4.2 Instalación de GlusterFS

Para instalar GlusterFS, debemos de obtener el repositorio del mismo. Para ello, debemos de ejecutar las siguientes intrucciones en los tres nodos:

```
wget -nd -nc -r -A.deb  
http://download.gluster.org/pub/gluster/glusterfs/LATEST/Debian/jessie/
```

Realizamos un **apt-get update** e instalamos gluster:

```
apt-get install glusterfs-server
```

Activamos y arrancamos gluster:

```
systemctl enable glusterfs-server  
systemctl start glusterfs-server
```

## 2.4.2 Configuración de GlusterFS

El primer paso será comprobar la disponibilidad de los otros dos nodos. Para ello, ejecutamos lo siguiente en la máquina 1:

```
gluster peer probe nodoha2  
gluster peer probe nodoha3
```

Debemos de ejecutar **gluster peer probe nodoha** en cualquiera de las otras dos máquinas.

Creamos la carpeta **/gluster** en el directorio raíz de las tres máquinas. Con esto, podemos crear nuestro volumen de la siguiente manera:

```
gluster volume create fs_ha replica 3 nodoha:/gluster  
nodoha2:/gluster nodoha3:/gluster force
```

Una vez creado correctamente, lo iniciamos con:

```
gluster volume start fs_ha
```

Por último, montamos el volumen en cada máquina para la replicación con el siguiente comando:

```
mount -t glusterfs localhost:/fs_ha /var/www/html/drupal
```

Y editamos el fichero `/etc/rc.local` añadiendo la línea anterior:

```
mount -t glusterfs localhost:/fs_ha /var/www/html/drupal
exit 0
```

## 2.5 Drupal

### 2.5.1 Introducción a Drupal

Drupal es un marco de gestión de contenidos o CMS libre, modular, que permite publicar artículos, imágenes, archivos y que también ofrece la posibilidad de otros servicios añadidos como foros, encuestas, votaciones, blogs y administración de usuarios y permisos.

Drupal es un sistema dinámico: en lugar de almacenar sus contenidos en archivos estáticos en el sistema de ficheros del servidor de forma fija, el contenido textual de las páginas y otras configuraciones son almacenados en una base de datos y se editan utilizando un entorno Web.

### 2.5.2 Instalación de Drupal

#### 2.5.2.1 Configuración de la base de datos

Lo primero que haremos será crear la base de datos y el usuario que utilizaremos para almacenar el contenido de Drupal. Para crear la base de datos:

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

Creamos el usuario:

```
MariaDB [(none)]> create user 'drupal'@localhost identified by
'xxxxxxxxxx';
```

Establecemos todos los privilegios de la base de datos al usuario recién creado:

```
MariaDB [(none)]> grant all privileges on drupal.* to
'drupal'@localhost identified by 'drupal';
```

#### 2.5.2.2 Configuración de Drupal

Nos descargamos el software desde la página oficial en la carpeta `/var/www/html`:

```
wget https://ftp.drupal.org/files/projects/drupal-7.43.zip
```

Descomprimos el archivo descargado y movemos el contenido al directorio principal:

```
unzip drupal-7.43.zip
mv drupal-7.43.zip/* .
rm -rf drup*
```

Y asignamos los permisos adecuados a los ficheros:

```
chown -R www-data:www-data *
```

El ultimo paso es la instalación del CMS, en la que se nos pedirá la información de la base de datos y el usuario administrador que crearemos en el CMS de Drupal.

### 3. Errores surgidos

Durante la implantación de todos los componentes he tenido los diferentes errores:

- El servicio de MySQL no respondía, lo que me llevó a cambiar incluso de SO. Mirando los logs observé que hacia referencia a los recursos de la RAM. Comprobé que no había RAM suficiente y redimensioné las instancias.

Una vez realizado esto, todo volvió a funcionar correctamente.

- El recurso de Apache en pacemaker no arrancaba. Solucionado con la parada del servicio **apache2** en la máquina donde se ejecutaba el recurso de apache. Una vez detenido, el servicio es gestionado por apache.
- Muchos quebraderos de cabeza con la replicación de GlusterFS. Realizaba toda la configuración correctamente pero no me replicaba. Esto se debía a que no montaba el volumen con el cliente de GlusterFS.

### 4. Bibliografía

<https://mariadb.org/installing-mariadb-galera-cluster-on-debian-ubuntu/>

<https://wiki.debian.org/Debian-HA/ClustersFromScratch>

## **Servidor Web HA**

[https://www.suse.com/documentation/sle\\_ha/book\\_sleha/data/sec\\_ha\\_config\\_crm\\_resources.html](https://www.suse.com/documentation/sle_ha/book_sleha/data/sec_ha_config_crm_resources.html)

[http://www.gluster.org/community/documentation/index.php/Getting\\_started\\_install](http://www.gluster.org/community/documentation/index.php/Getting_started_install)

[http://www.gluster.org/community/documentation/index.php/Getting\\_started\\_configure](http://www.gluster.org/community/documentation/index.php/Getting_started_configure)