Balanceador de Carga

(Haproxy + keepalived)

Página Web

(Wordpress + MySQL)

Vicente Fernández Andújar
Cristóbal Olivencia Carrión

1. Introducción

Este trabajo tiene como finalidad crear más de un servidor con más de un balanceador de carga y con réplica de base de datos. Ésto tiene muchas aplicaciones, por ejemplo tener más de un servidor web con los datos replicados de forma que si uno cae, el otro sigue trabajando, lo mismo ocurre con el balanceador, si uno no funciona el otro reparte los recursos lo que implica alta disponibilidad. No solo se enfoca en evitar pérdida de información, con la replicación de la base de datos también se consigue tener sincronizados los nodos de los servidores y lo que se realiza en uno se actualiza en el resto.

2. Componentes y configuración

2.1 Material

- Dos ordenadores como servidores web.
- Dos Raspberry Pi como balanceadores de carga.
- Un router para simular una red local.
- Un ordenador como cliente de los servidores.

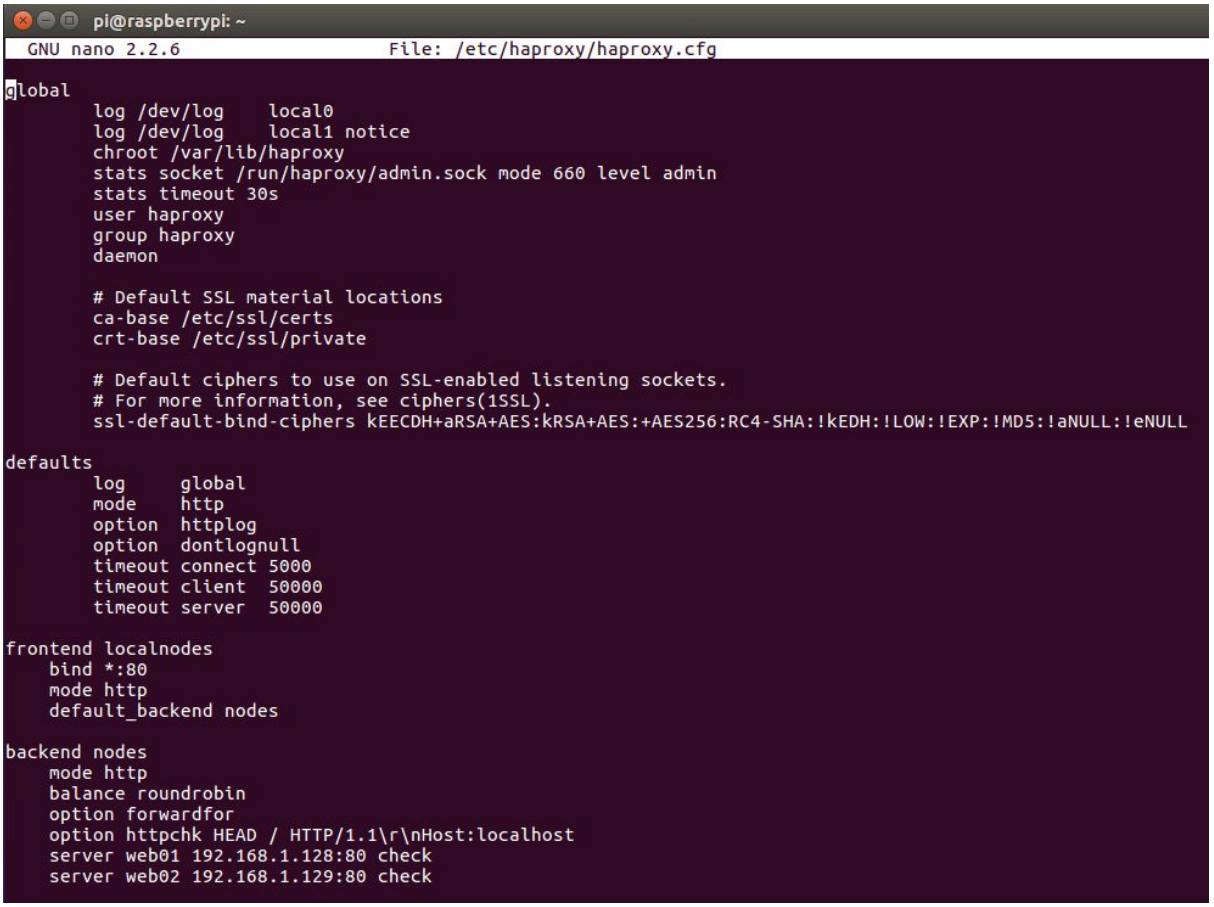
2.2 Balanceador de carga

A. <u>HAProxy</u>

HAProxy es un balanceador de carga de software libre que proporciona una **alta disponibilidad** y un servidor proxy para aplicaciones basadas en TCP y HTTP que **distribuyen las solicitudes** a través de varios servidores.

Utilizaremos dos Raspberry Pi para balancear la carga de los servidores web, siendo una Raspberry Pi el **nodo activo** y otra el **nodo pasivo**. Cuando el nodo activo deja de trabajar el nodo pasivo pasa a coger los mandos.

La configuración que hemos utilizado para los balanceadores son los siguientes:



La configuración que hemos utilizado para los balanceadores son los siguientes:

- Indicamos el protocolo que vamos a usar: http.
- El tiempo de espera para conexión.
- El tiempo de espera para los clientes.
- El tiempo de espera para los servidores.
- El puerto que vamos a usar: 80
- La IP de los servidores, usando la orden check, que mira si el servidor está activo antes de acceder a él.

Esta configuración es igual para los dos nodos balanceadores.

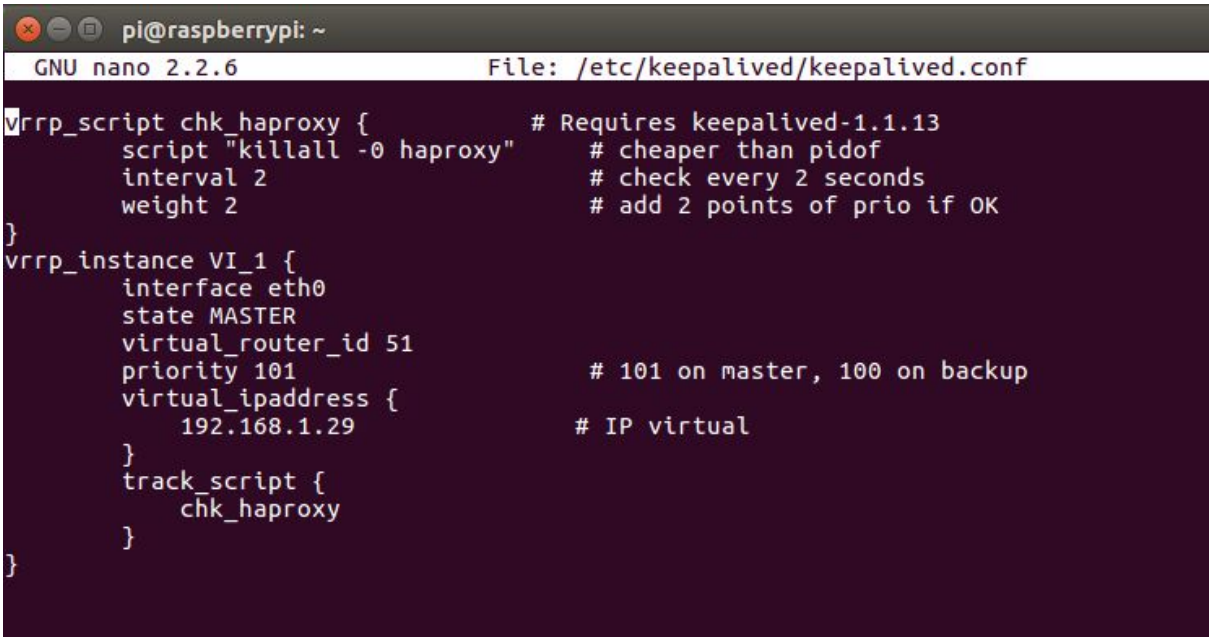
B. Keepalived

Keepalived es un software de enrutamiento, el objetivo principal de este programa es proporcionar un **equilibrio de carga y alta disponibilidad** para los balanceadores de carga.

Keepalived permite crear una IP virtual que utilizaremos para acceder a los balanceadores.

Cómo usamos la misma IP para los dos balanceadores si uno se desconecta automáticamente comienza a trabajar el otro balanceador.

En la configuración de keepalived especificamos quien será el **nodo Master** y quien el **nodo Slave**, junto la **IP virtual** para poder acceder directamente a los nodos balanceadores, como si fuera uno.



Configuración del Master

Configuración del Slave

2.3 Servidor web

En los servidores web hemos instalado **LAMP** en dos máquinas, que consta de:

- Una máquina con **LINUX,** hemos utilizado la distribución Ubuntu 16.04 como SO de los servidores web.
- **Apache,** que es un servidor web HTTP de código abierto.
- **MySQL,** es un sistema de gestión de bases de datos relacional, hemos utilizado esta base de datos puesto que es de código abierto y es bastante popular y es la que usa wordpress por defecto.
- **PHP,** es un lenguaje de código abierto adecuado para el desarrollo web y que puede ser incrustado en HTML.

2.4 Página Web

Se ha utilizado **Wordpress**, que es un sistema de gestión de contenidos o CMS utilizado para crear páginas web. Para su instalación tenemos que seguir una serie de pasos:

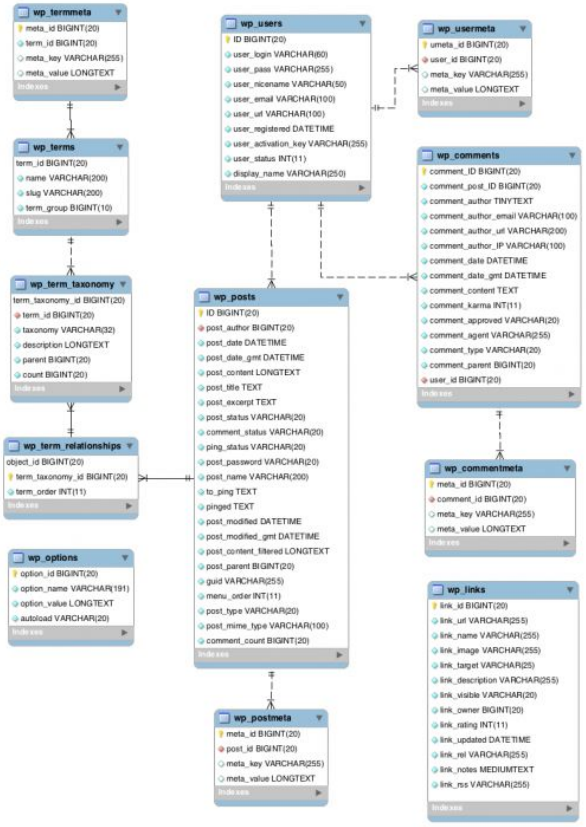
- **Configurar DB:** Creación de la base de datos y del usuario con permisos para operar en ella.
- **Configurar wp-config.php:** Archivo de configuración de Wordpress para indicar los datos relevantes para su correcto funcionamiento donde se indica la DB y usuario que se ha creado anteriormente.
- **Asistente en línea** del usuario admin donde se crea el usuario administrador de Wordpress (no de la DB).
- **Configurar .htaccess:** Para las reglas de reescritura de las direcciones URL.
- **Creación de nuevo sites-available** en apache: Indicar las reglas y la dirección del sitio para poder servirlo.

Una vez instalado en una máquina migrar sólo los archivos de Wordpress a la otra máquina.

2.5 Base de datos

MySQL es un sistema de gestión de bases de datos relacional. Habrá dos bases de datos master y slave que contendrán la misma información.

La base de datos de Wordpress tiene la siguiente estructura:



Para configurar la base de datos en réplica, primero en la master tenemos que realizar lo siguiente:

- **Crear usuario administrador** y dar permisos para la DB de Wordpress.
- **Modificar el archivo de configuración** MySQL indicando:
    - *bind-address = IP-maestro*
    - *id-server = 1*
    - *log_bin = /var/log/mysql/mysql-bin.log*
- **Crear usuario esclavo** y dar permisos, mediante la consola de MySQL:
    - *GRANT REPLICATION SLAVE ON *.* TO 'slave_user'@'%' IDENTIFIED BY 'password';*
- Bloquear DB y **volcar datos a archivo SQL** con mysqldump como vimos en clase de prácticas.

En la esclavo:

- **Modificar archivo de configuración** MySQL indicando:
    - *id-server = 2*
    - *log_bin = /var/log/mysql/mysql-bin.log*
    - *relay-log = /var/log/mysql/mysql-relay-bin.log*
    - *binlog_do_db = wordpressdatabase*
- **Importar DB** desde el archivo SQL de la master.
- **Activar replicación** indicando la DB master de la siguiente forma:
    - *CHANGE MASTER TO MASTER_HOST='IP-maestro',MASTER_USER='slave_user',*

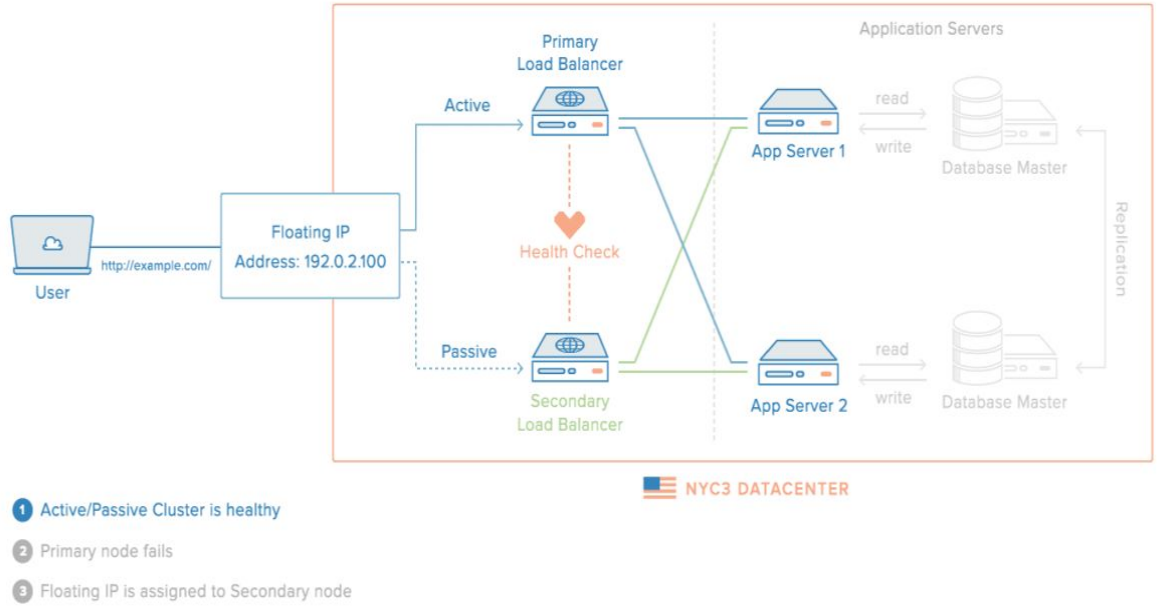
MASTER_PASSWORD='password',
MASTER_LOG_FILE='mysql-bin.000001', MASTER_LOG_POS= 107;

- **Iniciar esclavo** con:
    - *START SLAVE;*

Realizado lo anterior tendremos la base de datos replicada y sincronizada para ambos servidores web.

3. Funcionamiento

En esta imagen podemos ver el funcionamiento básico del trabajo realizado:



Tenemos los dos balanceadores de carga donde tenemos solo uno activo que se encargará de repartir los recursos a los dos servidores web mientras esté funcionando. En el momento que deje de funcionar se pondrá activo el segundo balanceador con la misma función.

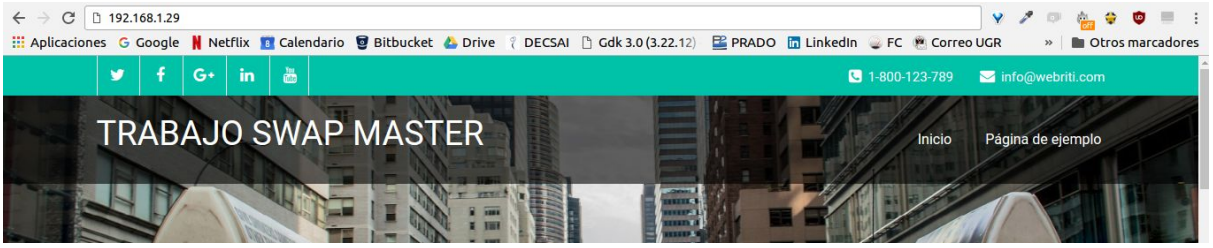
También puede darse el caso de que uno de los servidores web falle, de ser así se podría a funcionar el otro servidor web y al tener la base de datos en réplica maestro-esclavo no habría pérdida de información. Una vez arreglada la máquina que fallaba, ésta volvería a tener la base de datos actualizada y sincronizada.

Como hemos dicho anteriormente los dos servidores web comparten la misma información de la base de datos, la diferencia se encuentra en el archivo PHP de la página principal o index de cada una. En la máquina 1 se ha añadido al título *TRABAJO SWAP* la palabra *MASTER* y a la máquina 2 se ha añadido *SLAVE*. Así podremos diferenciar qué máquina está sirviendo el contenido.

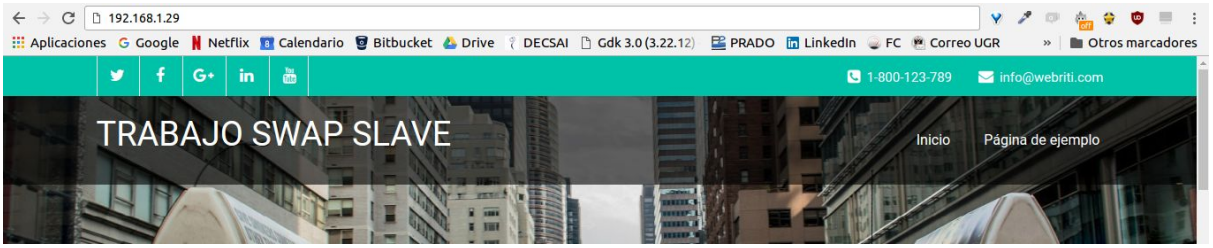
Es importante indicar las diferentes IPs involucradas:

- **IP virtual** (la que se utiliza para conectarnos al balanceador) es 192.168.1.29.
- **IP Máquina 1** es 192.168.1.128.
- **IP Máquina 2** es 192.168.1.129.

Cuando nos conectamos por primera vez a través de la IP virtual obtenemos la web que sirve la máquina 1:

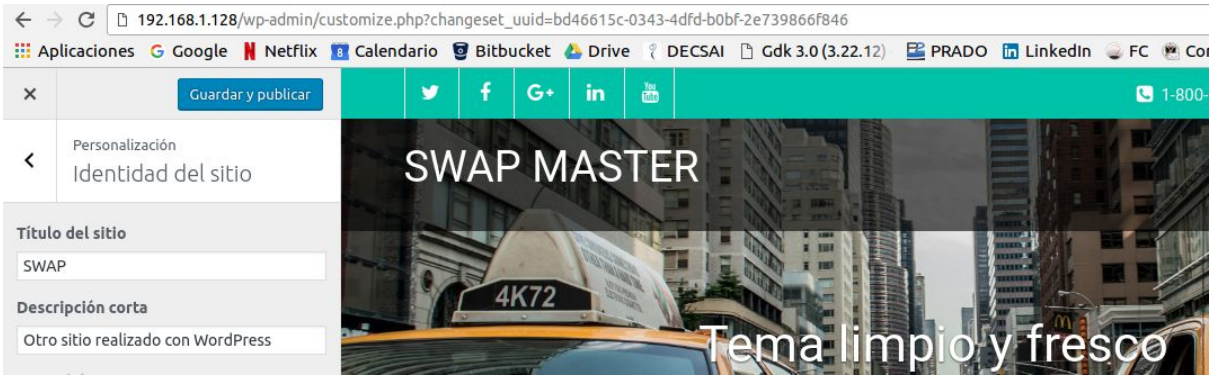


Si volvemos a recargar la página nos mostrará el contenido de la máquina 2:

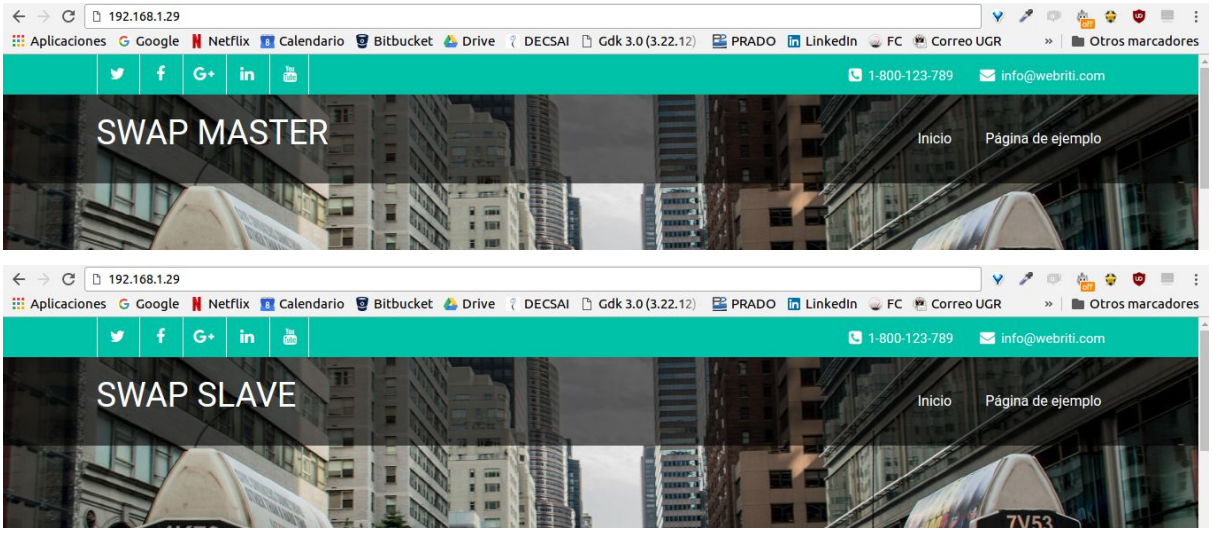


Si apagamos el balanceador activo, se pondrá a funcionar el pasivo sirviendo exactamente el mismo contenido anterior.

Posteriormente para comprobar que la base de datos está en réplica, vamos a realizar una pequeña modificación en ella, por ejemplo vamos a modificar el título desde uno de los servidores web (este cambio se refleja en la base de datos, no en el archivo PHP):



Guardamos los cambios y nos volvemos a conectar al balanceador para comprobar que se ha modificado el título en ambas máquinas:



Así comprobamos que el cambio ha sido efectivo en ambas bases de datos. Ésto funciona si se realiza a través de cualquiera de los servidores web ya que tenemos la BD en réplica.

4. Referencias

[How To Install Linux, Apache, MySQL, PHP (LAMP) stack on Ubuntu 16.04]

[How To Set Up Master Slave Replication in MySQL]

[How To Install Wordpress on Ubuntu 14.04]

[How To Set Up Highly Available HAProxy Servers with Keepalived and Floating IPs on Ubuntu 14.04]

[Wordpress Documentation]