

# ***Proyecto integrado ASIR***

## ***Cluster Web en alta disponibilidad con LVS (Linux Virtual Server) con La extensión Keepalived***



***Sergio Cortijo Gutiérrez***

***sergio.cortijo.gutierrez@gmail.com***



## INDICE DEL PROYECTO

1.	Conceptos básicos .....	5 -
2.	Objetivos del proyecto. ....	7 -
3.	¿Qué sistemas operativos vamos a usar? ¿Qué virtualizador vamos a usar? .....	8 -
4.	Configuración de las máquinas. ....	9 -
5.	Instalación y configuración de los programas.....	15 -
5.1	CONFIGURACION DEL SERVIDOR Y CLIENTE DHCP .....	15 -
5.1.1	SERVIDOR DHCP .....	15 -
5.1.2	CLIENTE DHCP.....	18 -
5.2	DRBD (Distributed Replicated Block Device).....	18 -
5.2.1	EXPLICACION .....	18 -
5.2.2	INSTALACION.....	19 -
5.2.3	CONFIGURACION.....	19 -
5.3	OCFS2 (Oracle Cluster File System). ....	22 -
5.3.1	EXPLICACION .....	22 -
5.3.2	INSTALACION.....	22 -
5.3.3	CONFIGURACION.....	23 -
5.4	BALANCEO DE LA CARGA CON LVS.....	25 -
5.4.1	EXPLICACION .....	25 -
5.4.2	ALGORITMOS DEL BALANCEADOR DE CARGA .....	26 -
5.4.3	INSTALACION.....	30 -
5.4.4	CONFIGURACION.....	30 -
5.5	KEEPALIVED .....	32 -
5.5.1	EXPLICACION .....	32 -
5.5.2	INSTALACION.....	32 -
5.5.3	CONFIGURACION.....	32 -
5.6	Apache.....	33 -
5.6.1	EXPLICACION .....	33 -
5.6.2	INSTALACION.....	33 -
5.6.3	CONFIGURACION.....	34 -
5.7	PHP 7.2 .....	36 -
5.7.1	EXPLICACION .....	36 -
5.7.2	INSTALACION.....	36 -
5.7.3	CREACION DE UN PROYECTO .....	36 -
5.8	PostgreSQL .....	37 -

5.8.1	EXPLICACION .....	- 37 -
5.8.2	INSTALACION.....	- 37 -
5.8.3	CONFIGURACION.....	- 37 -
5.8.4	CREACIÓN DE USUARIO Y BASE DE DATOS .....	- 38 -
5.9	Drupal .....	- 39 -
5.9.1	EXPLICACION .....	- 39 -
5.9.2	INSTALACION.....	- 40 -
6.	Conclusión .....	- 42 -
7.	Referencias.....	- 43 -

## 1. Conceptos básicos

### ¿Qué son los clusters?

Un cluster está formado por dos o más sistemas informáticos que funciona como si fuese uno. Estos ordenadores pueden estar unidos mediante una red informática. Los clusters se utilizan para mejorar el rendimiento y la disponibilidad de los servicios ofrecidos. Un cluster puede tener una o varias de las siguientes características:

- Alto rendimiento.
- Alta disponibilidad.
- Escalabilidad.
- Balanceo o equilibrio de carga.

También debemos tener en cuenta que estos sistemas en función de su uso pueden ser:

- Activo/Pasivo: Esto es cuando uno de los servidores está activo mientras el resto está inactivo esperando que el servidor activo deje de dar servicio para asumirlo alguno de ellos.
- Activo/Activo: Cuando todos los servidores del cluster están dando servicios.

### ¿Qué es cluster de alta disponibilidad?

Es un conjunto de dos o más máquinas que se caracterizan por mantener una serie de servicios compartidos y por estar constantemente monitorizándose entre sí.

Si se produce un fallo de hardware en alguna de las máquinas del cluster, el software de alta disponibilidad es capaz de arrancar automáticamente los servicios en cualquiera de las otras máquinas del cluster (*failover*). Y cuando la máquina que ha fallado se recupera, los servicios son nuevamente migrados a la máquina original (*failback*). Esta capacidad de recuperación automática de servicios nos garantiza la alta disponibilidad de los servicios ofrecidos por el cluster, minimizando así la percepción del fallo por parte de los usuarios.

### ¿Qué es cluster de alto rendimiento?

Es un conjunto de ordenadores que está diseñado para dar altas prestaciones en cuanto a capacidad de cálculo. Los motivos para utilizar un cluster de alto rendimiento son:

- el tamaño del problema a resolver
- el precio de la máquina necesaria para resolverlo.

### ¿Qué es la escalabilidad?

Es la propiedad deseable de un sistema, una red o un proceso, que indica su habilidad para extender sus operaciones sin perder calidad, o bien manejar el crecimiento continuo de trabajo de manera fluida, o bien para estar preparado para hacerse más grande sin perder calidad en los servicios ofrecidos.

### ¿Qué es LVS (Linux Virtual Server)?

Es un conjunto de componentes de software integrados para balancear la carga IP a lo largo de un grupo de servidores reales. LVS se ejecuta en un par de computadores configurados similarmente: uno de ellos es un *enrutador LVS activo* y el otro es un *enrutador LVS de respaldo*. El enrutador LVS activo tiene dos roles:

- Balancear la carga entre los servidores reales.
- Revisar la integridad de los servicios en cada servidor real.

### ¿Qué es el keepalived?

Se refiere generalmente a las conexiones de comunicación en una red que no están terminadas pero que se mantienen hasta que el **cliente** o servidor interrumpe la conexión. La característica clave de mantener las keep alive es el envío de un mensaje sin contenido entre un servidor y un cliente. Con este mensaje, uno de los usuarios de la red (cliente o servidor) puede controlar si la conexión se mantendrá y evitar que se cancele. Si la conexión todavía está disponible, se puede utilizar para el intercambio de datos.

### ¿Qué es el balanceo de cargas?

Es un concepto usado en informática que se refiere a la técnica para compartir el trabajo a realizar entre varios procesos, ordenadores, discos u otros recursos. Está íntimamente ligado a los sistemas de multiprocesamiento, o que hacen uso de más de una unidad de procesamiento para realizar labores útiles.

## 2. Objetivos del proyecto.

El objetivo de este proyecto es instalar un clúster de alta disponibilidad activo/pasivo con keepalived. Estas máquinas lo que harán es balancear la carga con LVS(Linux Virtual Server). El balanceador que esté activo en ese momento repartirá las peticiones sobre los diferentes servidores web que estén activos y se comunicarán con el balanceador por una red interna.

Para la sincronización de datos los servidores utilizaran DRBD con OCFS2. Mediante estas herramientas se replicarán los datos de la web y de las bases de datos.

Como servidor web vamos a utilizar Apache con PHP 7.2 que se conectarán a las bases de datos (PostgreSQL) que son internas.

### 3. ¿Qué sistemas operativos vamos a usar? ¿Qué virtualizador vamos a usar?

En este proyecto vamos a usar una Debian que va a ser un servidor DHCP y cuatro máquinas de Ubuntu 18.04 que van a ser clientes.

La elección de Debian (<https://www.debian.org/distrib/>) se debe a su comodidad para configurar servicios a nivel de administrador porque ofrece herramientas gráficas además del modo comando. En este proyecto he instalado la versión Debian GNU/Linux 10 (*Buster*).

Los clientes son máquinas Ubuntu 18.04 LTS (*Bionic Beaver*). Las he elegido por su versatilidad y porque son muy populares (<https://ubuntu.com/download>). Una razón fundamental es que Ubuntu 18.04 LTS trae soporte para DRBD en el kernel.

El virtualizador será VirtualBox (<https://www.virtualbox.org/wiki/Downloads>). Uso la versión 6.1.26 r145957 (Qt5.6.2) que se ejecuta sobre Windows 10.

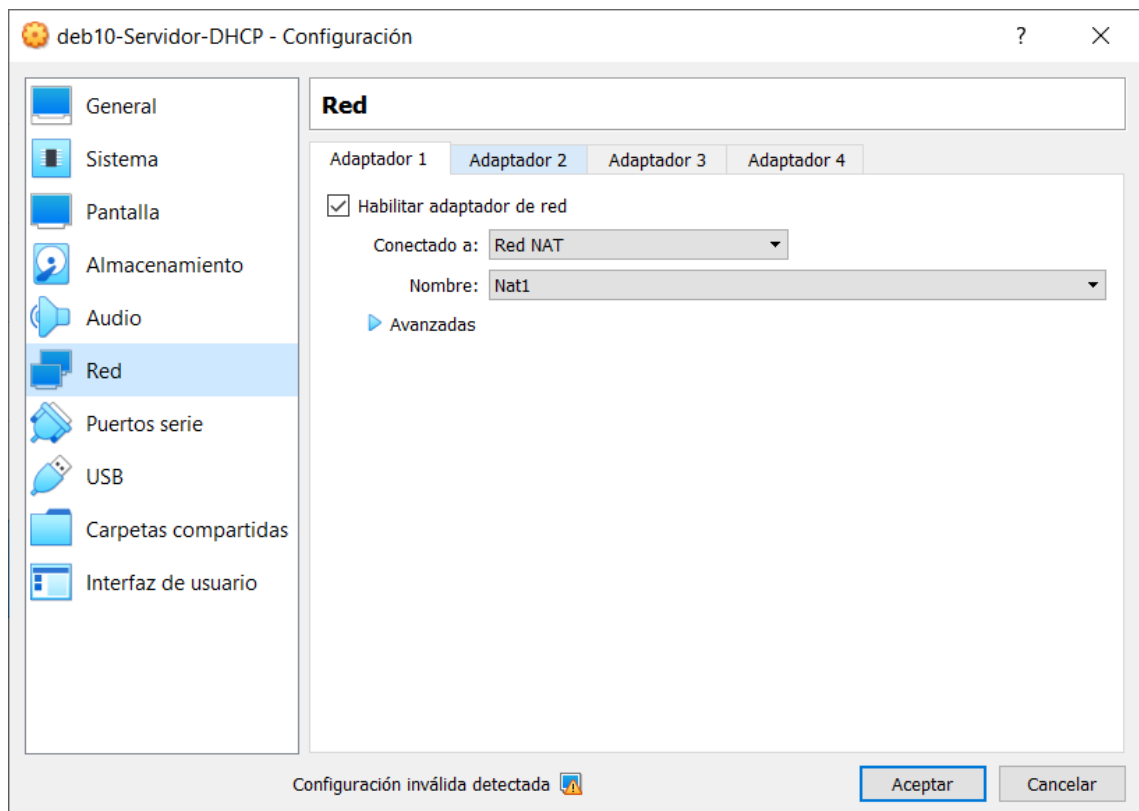


## 4. Configuración de las máquinas.

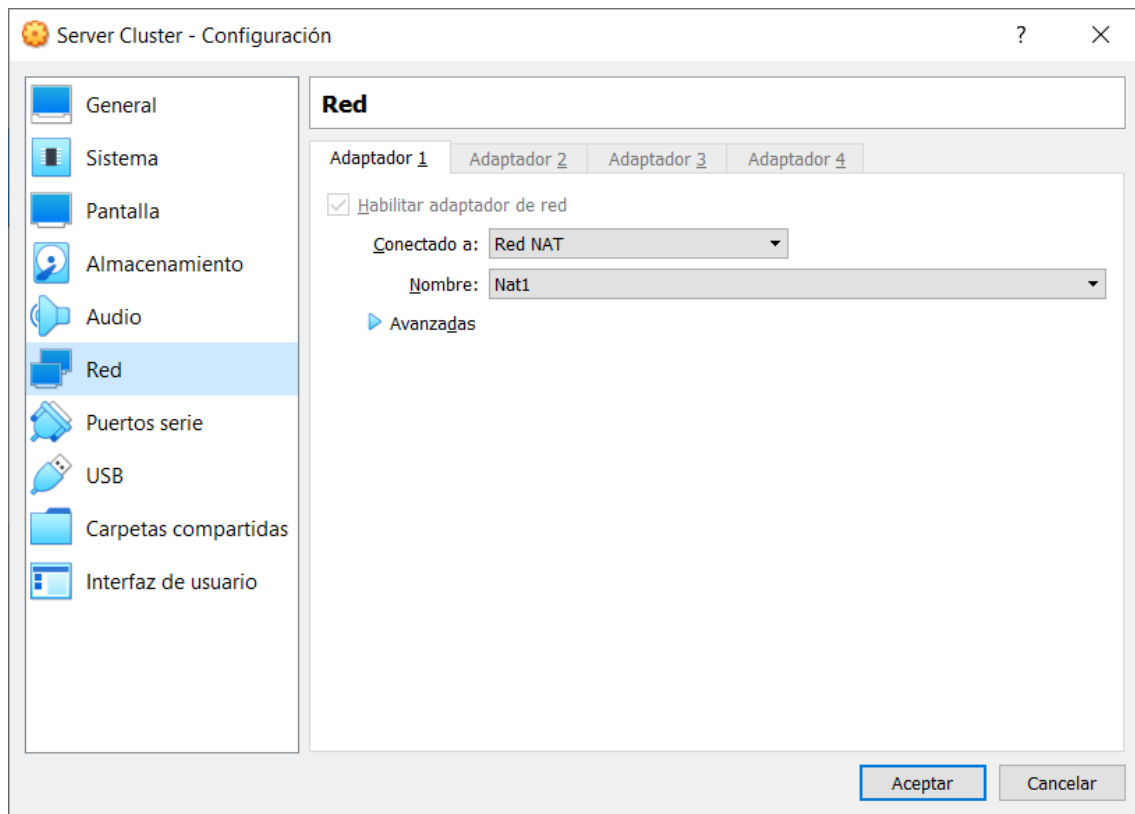
Primero preparamos cinco máquinas virtuales: una (Debian) se utiliza como servidor DHCP y las otras (Ubuntu) como clientes. De las clientes, dos se usan como estructuras maestro-esclavo de keepalived DR (*Server cluster* y *Empresa1*) y dos se usan como replicas o apoyo RS (*Empresa2* y *Empresa3*).

Keepalived está listo para estar en modo. Las máquinas están conectadas modo NAT cuya red es la 10.0.2.0/24 y va a tener un servidor DHCP.

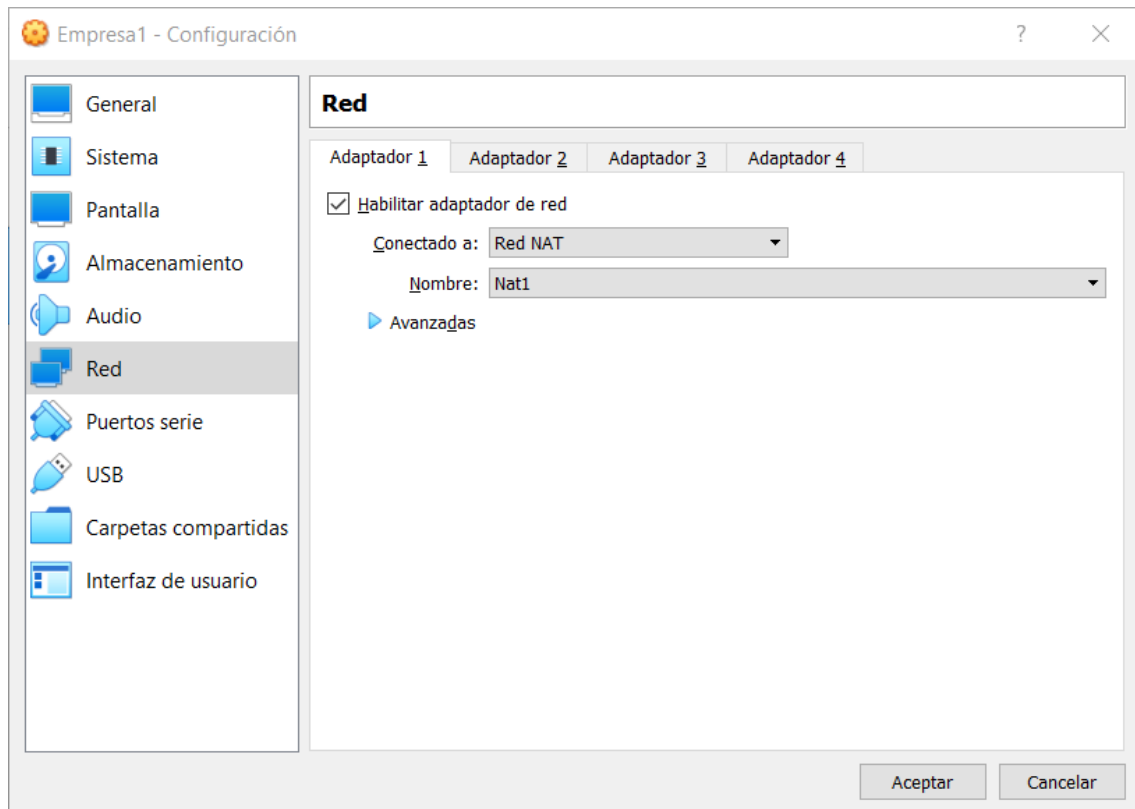
### 1. Server DHCP: 10.0.2.2/24



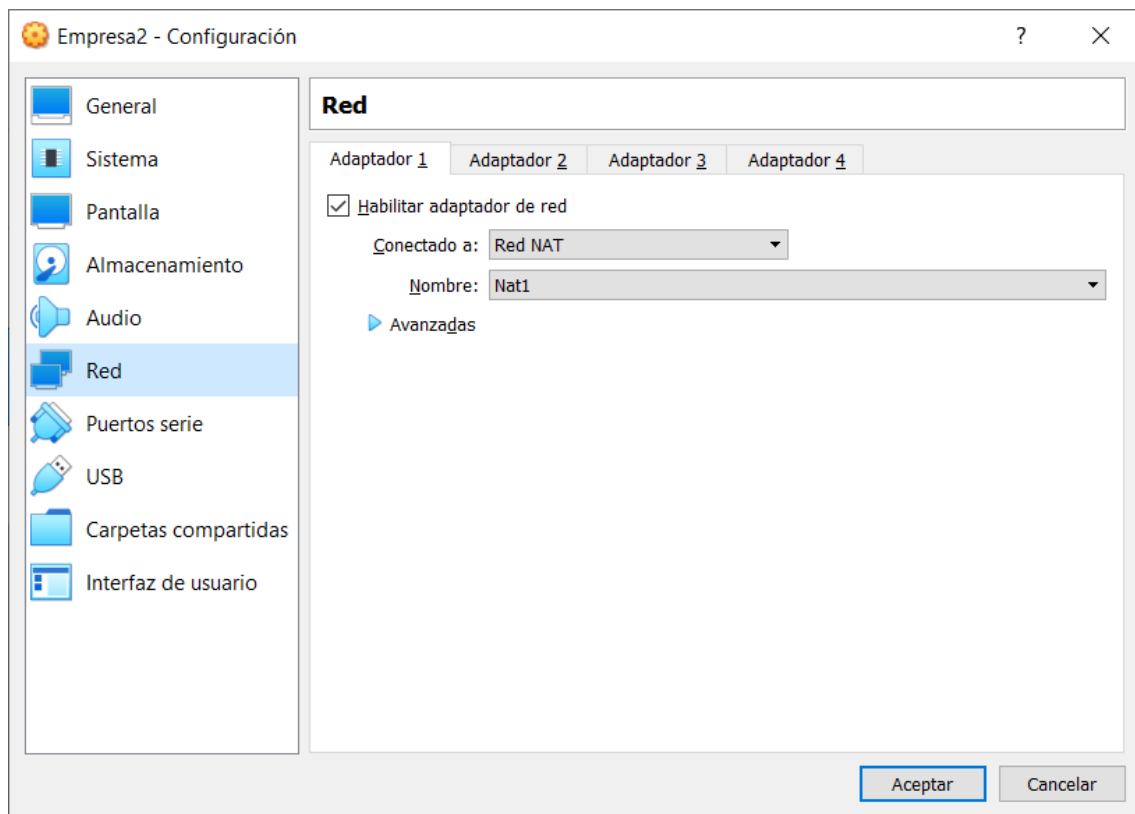
## 2. Server Cluster: 10.0.2.3/24



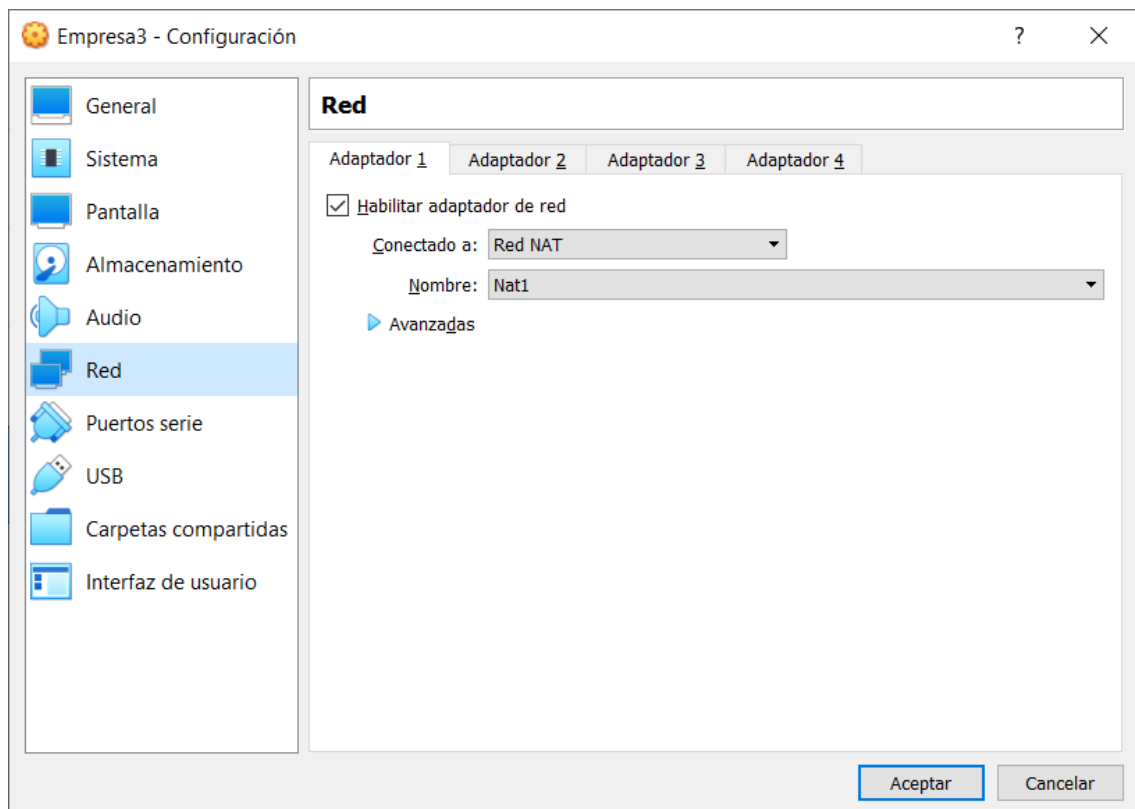
## 3. Empresa1: 10.0.2.4/24



## 4. Empresa2 : 10.0.2.5/24



## 5. Empresa3: 10.0.2.6/24



**Nos aseguramos de que la hora de las cuatro máquinas esté sincronizada y que firewall y SELinux (seguridad del kernel de linux) estén cerrados.**

En la máquina Debian vamos a configurar un servidor DHCP que reparta a cada máquina las IPs en un rango establecido.

La configuración de la maquina debian es la siguiente:

- Ponemos la máquina con un IP estática, le he puesto 10.0.2.2 en el fichero **interfaces** que se encuentra en **/etc/network** y lo tenemos que SIEMPRE como administrador (root). El comando es el siguiente:

nano /etc/network/interfaces

La estructura es la siguiente:

auto enp0s3

iface enp0s3 inet static

address 10.0.2.2

netmask 255.255.255.0

gateway 10.0.2.1

```
root@debDHCP:~# cat /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 enp0s3
#iface enp0s3 inet dhcp

auto enp0s3
iface enp0s3 inet static
    address 10.0.2.2
    netmask 255.255.255.0
    gateway 10.0.2.1
```

- Cuando queramos saber el gateway tenemos que poner la orden `ip route`

```
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:b2:5b:f1 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.2/24 brd 10.0.2.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:feb2:5bf1/64 scope link
        valid_lft forever preferred_lft forever
```

```
root@debDHCP:~# ip route
default via 10.0.2.1 dev enp0s3 onlink
10.0.2.0/24 dev enp0s3 proto kernel scope link src 10.0.2.2
root@debDHCP:~#
```

- Si queremos “salir” hacia internet tenemos que modificar el fichero **resolv.conf** que está en /etc. El comando es el siguiente: `nano /etc/resolv.conf`

Ponemos lo siguiente:

```
nameserver 8.8.8.8
```

```
nameserver 8.8.4.4
```

```
root@debDHCP:~# cat /etc/resolv.conf
nameserver 8.8.8.8
nameserver 8.8.4.4
```

- Reiniciamos el servicio de red con `service networking restart`

- Para comprobar que lo hemos configurado correctamente ponemos:  
ip addr  
ping 8.8.8.8

```
root@debDHCP:~# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data:
64 bytes from 8.8.8.8: icmp_seq=1 ttl=113 time=16.0 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=113 time=21.7 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=113 time=18.1 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=113 time=17.1 ms
64 bytes from 8.8.8.8: icmp_seq=5 ttl=113 time=19.3 ms
64 bytes from 8.8.8.8: icmp_seq=6 ttl=113 time=17.3 ms
^C
--- 8.8.8.8 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 13ms
rtt min/avg/max/mdev = 16.017/18.260/21.662/1.823 ms
root@debDHCP:~#
```

## 5. Instalación y configuración de los programas.

### 5.1 CONFIGURACION DEL SERVIDOR Y CLIENTE DHCP

#### 5.1.1 SERVIDOR DHCP

En este punto vamos a estar configurando las máquinas para poner en funcionamiento el cluster.

En la primera máquina, que es un debian, vamos a configurar el servidor DHCP y en las demás vamos a instalar el cliente DHCP. El programa que voy a usar es **isc-dhcp-server** y para los clientes es **isc-dhcp-client**.

La configuración de la máquina debian (**isc-dhcp-server**)

- Instalamos el paquete **isc-dhcp-server** con esta orden `apt install isc-dhcp-server`
- El servicio *DHCP* sólo debe estar disponible para la red interna. Por eso, debe aceptar conexiones por la interfaz interna (*eth0*, en este caso). Esto puede indicarse en el archivo de configuración `/etc/default/isc-dhcp-server`:

```
# Defaults for dhcp initscript

# sourced by /etc/init.d/dhcp

# installed at /etc/default/isc-dhcp-server by the maintainer
scripts

# This is a POSIX shell fragment

# On what interfaces should the DHCP server (dhcpcd) serve DHCP
requests?

#         Separate multiple interfaces with spaces, e.g. "eth0
eth1".

INTERFACES="enp0s3"
```

- La parte principal de la configuración está contenida en el archivo */etc/dhcp/dhcpd.conf*.

```
# Sample configuration file for ISC dhcpd for Debian

# The ddns-updates-style parameter controls whether or not the
server will

# attempt to do a DNS update when a lease is confirmed. We default
to the

# behavior of the version 2 packages ('none', since DHCP v2 didn't
# have support for DDNS.)

ddns-update-style none;

# option definitions common to all supported networks...

option domain-name "example.org";

option domain-name-servers ns1.example.org, ns2.example.org;


default-lease-time 600;

max-lease-time 7200;


# If this DHCP server is the official DHCP server for the local
# network, the authoritative directive should be uncommented.

authoritative;

# Use this to send dhcp log messages to a different log file (you
also

# have to hack syslog.conf to complete the redirection).

log-facility local7;
```



```
## SubNet proyectoCluster

subnet 10.0.2.0 netmask 255.255.255.0 {

    range 10.0.2.3 10.0.2.6;

    option routers 10.0.2.1;

    option broadcast-address 10.0.2.255;

}

[...]
```

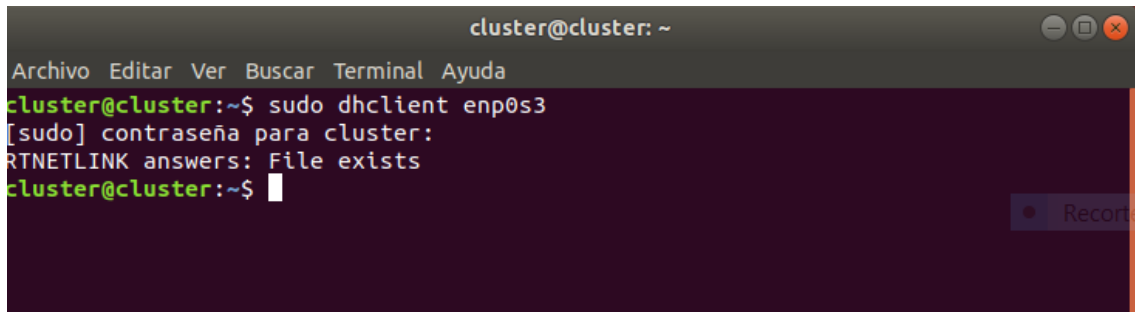
- Reiniciar el servicio *dhcp*: `/etc/init.d/isc-dhcp-server restart` o `systemctl restart isc-dhcp-server`
- Para comprobar que el servicio se ha levantado correctamente: `systemctl status isc-dhcp-server`

```
root@debDHCP:~# systemctl restart isc-dhcp-server.service
root@debDHCP:~# systemctl status isc-dhcp-server.service
• isc-dhcp-server.service - LSB: DHCP server
   Loaded: loaded (/etc/init.d/isc-dhcp-server; generated)
   Active: active (running) since Mon 2021-11-29 09:35:58 CET; 16s ago
     Docs: man:systemd-sysv-generator(8)
  Process: 944 ExecStart=/etc/init.d/isc-dhcp-server start (code=exited, status=0/SUCCESS)
    Tasks: 1 (limit: 1149)
   Memory: 4.5M
    CGroup: /system.slice/isc-dhcp-server.service
            └─956 /usr/sbin/dhcpd -4 -q -cf /etc/dhcp/dhcpd.conf enp0s3

nov 29 09:35:56 debDHCP systemd[1]: isc-dhcp-server.service: Succeeded.
nov 29 09:35:56 debDHCP systemd[1]: Stopped LSB: DHCP server.
nov 29 09:35:56 debDHCP systemd[1]: Starting LSB: DHCP server...
nov 29 09:35:56 debDHCP isc-dhcp-server[944]: Launching IPv4 server only.
nov 29 09:35:56 debDHCP dhcpd[956]: Wrote 4 leases to leases file.
nov 29 09:35:56 debDHCP dhcpd[956]: Server starting service.
nov 29 09:35:58 debDHCP isc-dhcp-server[944]: Starting ISC DHCPv4 server: dhcpd.
nov 29 09:35:58 debDHCP systemd[1]: Started LSB: DHCP server.
root@debDHCP:~# _
```

### 5.1.2 CLIENTE DHCP

Para recibir la IP del servidor DHCP primero tenemos que descárganos el cliente **isc-dhcp-client** y lanzar la orden `sudo dhclient enp0s3`. Por último reiniciamos la red.



```
cluster@cluster: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
cluster@cluster:~$ sudo dhclient enp0s3  
[sudo] contraseña para cluster:  
RTNETLINK answers: File exists  
cluster@cluster:~$
```

Como vemos en la captura pone este mensaje porque ya lo hemos ejecutado previamente. Lo mismo pone en la demás máquinas.

## 5.2 DRBD (Distributed Replicated Block Device).

### 5.2.1 EXPLICACION

DRBD es un software que permite hacer **réplica de los datos de una partición entre varias máquinas**. Es decir, que, si tengo una partición del mismo tamaño en dos máquinas, con DRBD puedo hacer una réplica del contenido de esta partición de forma automática, para que en el caso de que una máquina falle, tenga todo el contenido de esa partición accesible desde la otra máquina. Es como un **RAID1**, pero entre distintas máquinas.

Ubuntu 18.04 LTS trae soporte para DRBD en el kernel, por lo que no necesita cargar ningún módulo, que en versiones más antiguas si hiciera falta instalar más módulos.

DRBD crea un dispositivo de bloque `drbdX` accesible desde los dos servidores. El servidor primario es el que tiene acceso de lectura y escritura mientras que el secundario solo tiene de lectura. A partir de la versión 0.8, DRBD nos permite tener dos servidores primarios, por lo que los dos pueden leer y escribir.

### 5.2.2 INSTALACION

Nos descargamos el paquete en superusuario o poniendo:

```
# sudo su
```

```
#apt install drbd-utils
```

```
$ sudo apt install drbd-utils
```

Yo he instalado en dos máquinas virtuales:

1. Server Cluster
2. Empresa 1

Con esto se instalarán los siguientes comandos:

- drbdmeta: se utiliza para mostrar, crear y modificar el sistema de metadatos. Se encarga de la sincronización.
- drbdsetup: permite asociar el dispositivo drbd con los ficheros de bloques y sirve para cambiar casi todos los parámetros de configuración.
- drbdadm: esta herramienta es la que realmente se utiliza ya que en función de los parámetros que se le introduzcan, ejecuta un drbdmeta o un drbdsetup.

### 5.2.3 CONFIGURACION

Buscamos el fichero de configuración de DRBD que se encuentra en /etc llamado drbd.conf y si queremos ver el contenido de este fichero hay que poner en la consola: `cat /etc/drbd.conf`. Antes, se incluía la configuración completa en este fichero, pero ahora contiene dos includes:

```
include "drbd.d/drbd.conf";  
  
include "drbd.d/*.res";
```

Esto lo que vine a hacer es agregar todos los ficheros que se llamen `global_common.conf` o que terminen en `.res` y se encuentre en el directorio `/etc/drbd.d`. Con esta separación podemos separar la configuración global de la de los RAID-1 que vamos a crear.

En el fichero `global_common.conf` encontramos las opciones que afectarán a todos los dispositivos que creamos, si en la definición de este dispositivo no pone lo contrario. En este fichero podemos ver dos secciones: `global`, donde se configuran los parámetros globales de DRBD y `common`, donde se configuran los parámetros de configuración comunes.

Los ficheros de configuración `.res` solo contienen la sección `resource`, donde se definen los recursos. Dentro de esta sección podemos poner las mismas opciones en la sección

common del fichero de configuración común para todos los dispositivos. La parte más importante dentro de este fichero es la que define al cliente (on cliente).

Todo lo que esté en ese apartado es exclusivo para ese cliente, mientras lo que esté afuera afectará a todos los clientes. Con este esquema de ficheros, podemos definir los parámetros comunes en la sección common del fichero `global_common.conf` y los parámetros específicos de un recurso en su fichero de configuración. A continuación, voy a describir mi fichero de configuración.

En la sección global ponemos el contador a no. Esto hace que no se conecte a la página DRBD y no genere estadística, para acceder a este fichero hay que poner `cat /etc/drbd.d/global_common.conf`. Sería escrito así:

```
global{
    usage-count no;
}
```

En la parte common he definido todos los parámetros de configuración para que los nodos estén en primario los dos. Estas opciones también se podrían haber puesto en la parte **resource** de un recurso concreto. Lo primero que he configurado en esta parte es el protocolo de sincronización C, el cual es necesario si queremos tener dos nodos primarios. Los otros protocolos son A y B.

### **protocol C;**

Estos protocolos tienen las siguientes características:

- A: Protocolo de replicación asíncrona.
- B: Protocolo de replicación síncrona de memoria(semi-síncrono).
- C: Protocolo de replicación síncrona.

En esta sección se le dice a DRBD que cuando arranque, lo haga en modo primario.

```
startup {
    become-primary-on both;
}
```

En net se definen los parámetros de red y qué hacer en caso de fallo, en este caso se ha configurado para que se permitan dos primarios y que cuando haya un fallo porque nadie tenga el recurso, lo tenga uno solo, o lo tengan los dos, y que no hagan nada para evitar la corrupción de datos, de eso ya se encargará OCFS2.

```
net {
    allow-two-primaries;
```

```
after-sb-0pri discard-zero-changes;  
after-sb-1pri discard-secondary;  
after-sb-2pri disconnect;  
}
```

Por último, ponemos la velocidad de sincronización en la sección syncer.

```
syncer {  
    rate 1000M;  
}
```

Ya en el fichero de configuración del recurso, el cual he llamado drbd.res, solo se encuentra la sección resource y en ella se define el nombre del recurso, el disco físico que va a utilizar, el nombre del dispositivo que nos va a crear DRBD. También le vamos a decir que los metadatos se guarden junto con los datos y también pondremos la dirección ip y el puerto tcp por el que se hará la sincronización.

```
global { usage-count no; }  
common { syncer { rate 100M; } }  
resource drbd {  
    protocol C;  
    startup {  
        wfc-timeout 15;  
        degr-wfc-timeout 60;  
    }  
    net {  
        cram-hmac-alg sha1;  
        shared-secret "secret";  
    }  
    on cluster {  
        device /dev/drbd0;  
        disk /dev/sdb1;  
        address 10.0.2.6:7788;  
        meta-disk internal;
```

```
    }  
    on empresa1 {  
        device /dev/drbd0;  
        disk /dev/sdb1;  
        address 10.0.2.4:7788;  
        meta-disk internal;  
    }  
}
```

### 5.3 OCFS2 (Oracle Cluster File System).

#### 5.3.1 EXPLICACION

OCFS2 es un sistema de fichero en clúster que permite el acceso simultáneo de varios nodos. Está desarrollado por Oracle y tiene licencia GNU. Cada nodo dispone de un sistema de ficheros montados, normalmente escribe en un fichero metadata permitiendo a los otros nodos saber que se encuentra disponible. Limitaciones:

- Permite hasta 32000 directorios.
- El tamaño máximo de un volumen es de 4PB.
- El tamaño máximo de un fichero es de 4PB.
- La longitud del nombre del fichero es de 255 bytes.

#### 5.3.2 INSTALACION

La instalación en Ubuntu 18.04 LTS de este software es sencilla, ya que está disponible en los repositorios:

```
# apt install ocfs2-tools  
$ sudo install ocfs2-tools
```

### 5.3.3 CONFIGURACION

Tras la instalación se nos instalarán dos servicios:

- ocfs2: se encarga de los recursos que están montados en el clúster. Su fichero de configuración se encuentra en **/etc/ocfs2/cluster.conf** y contiene la siguiente información:

```
cluster@cluster:~$ cat /etc/ocfs2/cluster.conf
node:
    ip_port = 7777
    ip_address = 10.0.2.6
    number = 0
    name = empresa1
    cluster = ocfs2

node:
    ip_port = 7777
    ip_address = 10.0.2.4
    number = 1
    name = empresa2
    cluster = ocfs2

cluster:
    node_count = 2
    name = ocfs2
cluster@cluster:~$
```

- o2cb: se encarga de los parámetros técnicos del clúster y su fichero de configuración se encuentra en **/etc/default/o2cb**. En este fichero encontramos lo siguiente:

```
cluster@cluster: ~
Archivo Editar Ver Buscar Terminal Ayuda

cluster:
    node_count = 2
    name = ocfs2
cluster@cluster:~$ cat /etc/default/o2cb
#
# This is a configuration file for automatic startup of the O2CB
# driver. It is generated by running 'dpkg-reconfigure ocfs2-tools'.
# Please use that method to modify this file.
#
# O2CB_ENABLED: 'true' means to load the driver on boot.
O2CB_ENABLED=true
# O2CB_BOOTCLUSTER: If not empty, the name of a cluster to start.
O2CB_BOOTCLUSTER=ocfs2
# O2CB_HEARTBEAT_THRESHOLD: Iterations before a node is considered dead.
O2CB_HEARTBEAT_THRESHOLD=31
# O2CB_IDLE_TIMEOUT_MS: Time in ms before a network connection is considered dead.
O2CB_IDLE_TIMEOUT_MS=30000
# O2CB_KEEPLIVE_DELAY_MS: Max. time in ms before a keepalive packet is sent.
O2CB_KEEPLIVE_DELAY_MS=2000
# O2CB_RECONNECT_DELAY_MS: Min. time in ms between connection attempts.
O2CB_RECONNECT_DELAY_MS=2000
cluster@cluster:~$
```

Este fichero se puede configurar de forma automática con el comando:

```
# dpkg-reconfigure ocfs2-tools
```

Le damos a enter por defecto

Tras tenerlo configurado, reiniciamos los servicios:

```
# /etc/init.d/ocfs2 restart
```

```
# /etc/init.d/o2cb restart
```

Con esta herramienta se instalarán los siguientes comandos:

- mount.ocfs2: Para montar el sistema de ficheros en un directorio.
- mkfs.ocfs2: Crea el sistema de ficheros en un dispositivo. El servicio o2cb debe estar parado. fsck.ocfs2: Chequea el sistema de ficheros.
- mounted.ocfs2: Detecta los volúmenes ocfs2 que haya montados.
- tuneefs.ocfs2: Sirve para cambiar parámetros del sistema de ficheros
- debugfs.ocfs2: Es un depurador del sistema de ficheros y sirve para ver las estructuras del dispositivo.
- ocfs2cdsl: Permite crear enlaces simbólicos a un fichero o directorio

La que nosotros utilizaremos con el dispositivo de bloque será mkfs.ocfs2, para formatearla en ocfs2:

```
# mkfs.ocfs2 /dev/drbd0
```

Para montar la partición una vez formateada:

```
# mount /dev/drbd0 /opt
```

Y como queremos que se monte con la máquina tenemos que añadir la siguiente línea al fichero fstab:

```
/dev/drbd1 /opt ocfs2 rw,_netdev,heartbeat=local 0 0
```

Para que monte la partición correctamente no me valió con eso, así que añadí una línea en el fichero /etc/rc.local, para que cargara de nuevo el fichero fstab. Con esto debemos tener cuidado, ya que todos los programas que necesiten los ficheros de /opt (va a ser donde voy a colocar los ficheros que se tengan que sincronizar, como los de apache) tienen que arrancar después de que se monte la partición. La línea que añadí al fichero fue:

```
mount -a
```



## 5.4 BALANCEO DE LA CARGA CON LVS

### 5.4.1 EXPLICACION

El balanceo de la carga es una técnica que se utiliza para compartir el trabajo entre varios procesos, ordenadores, discos u otros recursos. El balanceo de carga se puede hacer mediante diferentes algoritmos. En nuestro caso vamos a utilizar LVS.

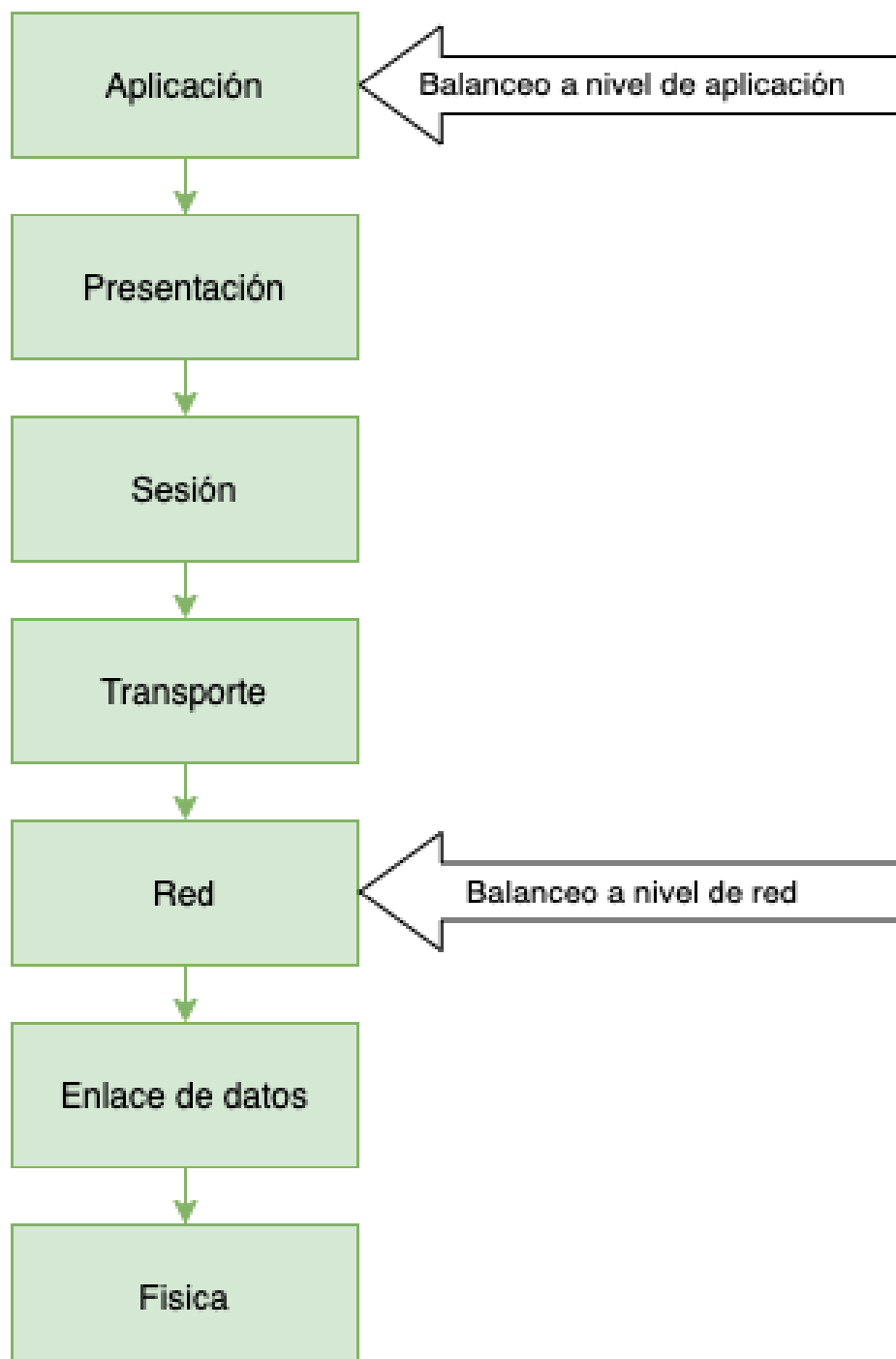
LVS es un software que nos permite balancear la carga entre varios servidores. LVS está integrado dentro del kernel de Linux con lo cual se consigue un mayor rendimiento, además existe gran cantidad de documentación en la red. LVS también es la opción más genérica, con LVS se puede balancear tanto HTTP, FTP, MySQL o cualquier otro protocolo de red.

LVS nos permite diferentes configuraciones de red. Las configuraciones que disponemos son:

- **NAT:** Con esta técnica se modifica el datagrama IP. En esta técnica hay un dispositivo intermedio actuando como puerta de enlace. Este dispositivo recibe un paquete, modifica el datagrama, y lo envía a uno de los servidores que tenga asignado para ese servicio. El servidor procesa esa petición y se la envía a la puerta de enlace de nuevo. La puerta de enlace vuelve a cambiar el datagrama y se lo envía al cliente. La configuración de un servidor virtual mediante NAT es muy transparente, ya que no requiere modificaciones ni configuraciones especiales en los sistemas operativos de los servidores reales, lo cual facilita su implementación. Una de las principales desventajas de esta técnica es que los servidores deben de encontrarse en la misma red que el balanceador, pero esto no es un gran impedimento. La principal desventaja de esta técnica es el cuello de botella que puede causar el tener que pasar todos los paquetes tanto de ida como de vuelta por la puerta de enlace. En este caso el balanceador tiene que cambiar tanto la dirección IP como la MAC de los paquetes de los clientes y de los servidores reales.
- **Direct Routing(Enrutamiento directo):** En este caso el balanceador de carga sigue siendo el que recibe las peticiones pero son los servidores los que envían la respuesta. El balanceador y los servidores deben de estar conectados por una LAN. En esta técnica el principal inconveniente es el arp, ya que todos los servidores reales tienen que saber que la ip virtual es suya, pero si lo ponemos en una interfaz física con arp, todos dirán que la dirección virtual es la suya. Para arreglar esto tenemos dos métodos, unos es desactivar el arp de los servidores, por lo que ya solo el balanceador contestará y otra es asignarle la dirección a la interfaz loopback, por lo que el servidor real sabrá que pertenece a él pero no contestará a las peticiones arp ya que son en otra red. Para que funcione bien, los servidores no deben responder a los mensajes ARP. En este caso el balanceador de carga solo tiene que cambiar la dirección MAC para direccionar las peticiones al servidor real.

- **IP Tunneling(Encapsulación IP):** En el IP Tunneling, los servidores no tienen que estar conectados directamente a una red donde se encuentre el balanceador. En este caso el balanceador recibe el datagrama del cliente con destino el servicio virtual. El datagrama lo reenvía con dirección Ip del cliente como origen y dirección IP del servicio virtual como destino, este datagrama va metido dentro de otro datagrama con dirección Ip del balanceador como origen y dirección Ip del servidor como destino. El paquete puede ser enviados por dispositivos intermedios como routers.

#### 5.4.2 ALGORITMOS DEL BALANCEADOR DE CARGA

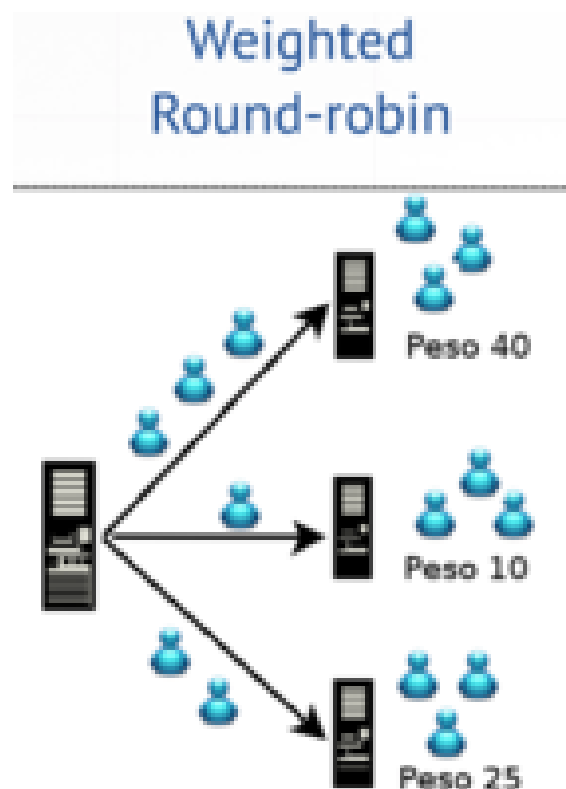


#### 5.4.2.1 ALGORIMOS DE CAPA DE RED

## Round Robin

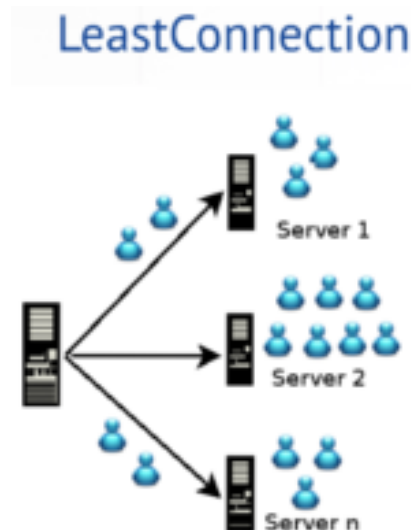
La carga de tráfico se distribuye al primer servidor disponible y luego ese servidor se coloca en la cola. Si los servidores son idénticos y no hay conexiones persistentes, este algoritmo puede resultar eficaz. Hay dos tipos principales de algoritmos por turnos:

- **round-robin con peso** Si los servidores no tienen la misma capacidad, este algoritmo se puede utilizar para distribuir la carga. Se pueden asignar algunos pesos o parámetros de eficiencia a todos los servidores de un grupo y, en base a eso, de manera cíclica similar, se distribuye la carga.
- **round-robin dinámico** Los pesos que se asignan a un servidor para identificar su capacidad también se pueden calcular en tiempo de ejecución. El round robin dinámico ayuda a enviar las solicitudes a un servidor según el peso del tiempo de ejecución.



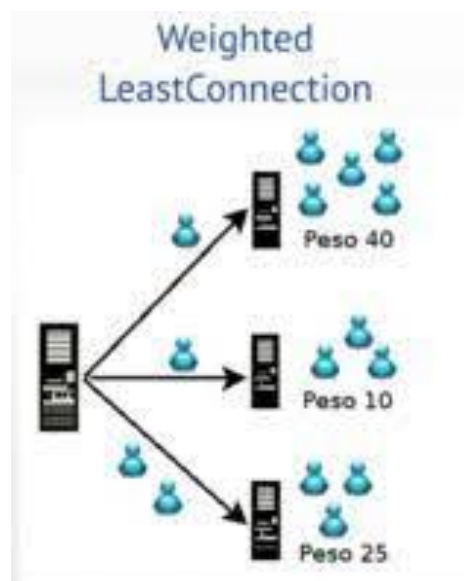
## Algoritmo de menos conexiones (least-connection)

Este algoritmo calcula el número de conexiones activas por servidor durante un tiempo determinado y dirige el tráfico entrante al servidor con la menor cantidad de conexiones. Esto es muy útil en los escenarios donde se requiere una conexión persistente.



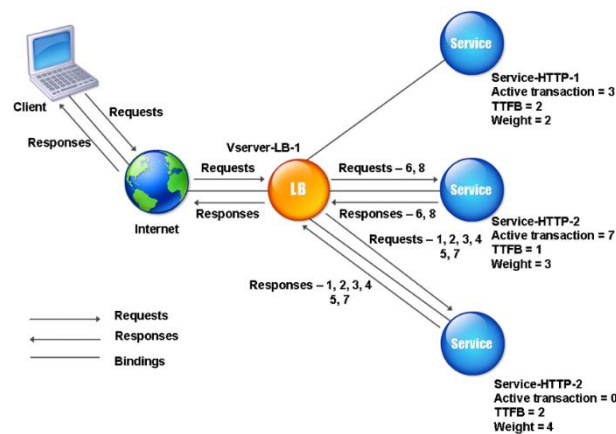
## Algoritmo de menos conexiones con peso (Weighted least-connection)

Esto es similar al algoritmo de conexiones mínimas anterior, pero aparte del número de conexiones activas a un servidor, también tiene en cuenta la capacidad del servidor.



## Algoritmo de menor tiempo de respuesta (least-response-time)

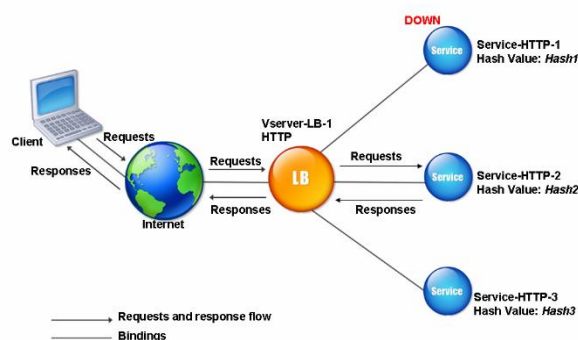
De nuevo, esto es similar al algoritmo de conexiones mínimas, pero también considera el tiempo de respuesta de los servidores. La solicitud se envía al servidor con el menor tiempo de respuesta.



## Algoritmo de Hashing

Los diferentes parámetros de solicitud se utilizan para determinar dónde se enviará la solicitud. Los diferentes tipos de algoritmos basados en esto son:

- **Hash de fuente/destino** Las direcciones IP de origen y destino se combinan para determinar el servidor que atenderá la solicitud. En caso de una conexión interrumpida, la misma solicitud se puede redirigir al mismo servidor al volver a intentarlo.
- **Hash de url** La URL de solicitud se usa para realizar un hash, y este método ayuda a reducir la duplicación de cachés del servidor al evitar almacenar el mismo objeto de solicitud en muchas cachés.



#### 5.4.2.2 ALGORITMO DE CAPA DE APLICACIÓN

En esta capa, el tráfico se puede distribuir según el contenido de la solicitud; por lo tanto, los balanceadores de carga pueden tomar una decisión mucho más informada. La respuesta del servidor también se puede rastrear, ya que ha viajado desde el servidor, y esto ayuda a determinar la carga del servidor de manera mucho más efectiva.

Uno de los algoritmos más importantes utilizados en esta capa es el algoritmo de solicitud menos pendiente. Este algoritmo dirige el tráfico de solicitudes HTTP pendientes al servidor más disponible. Este algoritmo es útil para ajustar el pico repentino en las solicitudes al monitorear la carga del servidor.

#### 5.4.3 INSTALACION

Para la instalación simplemente tenemos que instalar la herramienta administrativa, ya que LVS viene integrado en el núcleo de linux.

```
# apt install ipvsadm
```

#### 5.4.4 CONFIGURACION

Su fichero de configuración es /etc/default/ipvsadm y el fichero de configuración de sus reglas es etc/ipvsadm.rules. En mi caso no me va a hacer falta este fichero ya que keepalived será el encargado de manejarlos.

A parte de por ficheros de configuración se puede configurar con comandos. La primera vez lo configuré por comandos para ver si tenía bien configurado los clientes, ya que resultaba más fácil. Para configurar LVS por comando, lo primero que tenemos que definir es el servicio que vamos a balancear, en mi caso el puerto 80 y el algoritmo, si no ponemos ninguno coge el que trae por defecto(wlc) yo he puesto wrr. El comando es el siguiente:

```
ipvsadm -A -t 10.0.0.4:80 -s wrr
```

Tras esto tendremos que ir añadiendo todos los servidores reales:

```
ipvsadm -a -t 10.0.0.4:80 -r 10.0.2.3:80 -g
```

```
ipvsadm -a -t 10.0.0.4:80 -r 10.0.2.4:80 -g
```

```
empresa2@empresa2:~$ sudo ipvsadm -A -t 10.0.0.4:81 -s wrr
[sudo] contraseña para empresa2:
empresa2@empresa2:~$ ipvsadm -a -t 10.0.0.4:81 -r 10.0.2.4:81 -g
Can't initialize ipvs: No space left on device
Are you sure that IP Virtual Server is built in the kernel or as module?
empresa2@empresa2:~$ sudo ipvsadm -a -t 10.0.0.4:81 -r 10.0.2.4:81 -g
empresa2@empresa2:~$ clear
```

## 5.5 KEEPALIVED

### 5.5.1 EXPLICACION

Keepalived es un demonio que monitoriza el estado de los servicios que se indica en su fichero de configuración. Keepalived implementa un framework basado en tres tipos de chequeo: nivel 3, nivel 4 y nivel 5/7. Este Framework da al demonio la habilidad de chequear el estado de un pool de Linux Virtual Server (LVS). Cuando un servidor del pool está parado, keepalived informa al kernel de Linux para borrar a este servidor de la entrada de la topología de LVS. Además, keepalived implementa una pila VRRPv2 (es un protocolo de redundancia que significa Virtual Router Redundancy Protocol version2) para manejar los nodos del balanceador.

En resumen, controla en el propio balanceador (que es un clúster activo/pasivo) el nodo que es el activo y el pasivo, de modo que si uno cae el otro siga dando el servicio de balanceo. Y, por otro lado, monitoriza a los servidores reales viendo si tienen los servicios levantados o no, para sacarlos de la tabla de candidatos a recibir peticiones del servicio monitorizado.

### 5.5.2 INSTALACION

Viene en los repositorios de Ubuntu para instalarlo:

```
# apt install Keepalived
```

### 5.5.3 CONFIGURACION

Su fichero de configuración es /etc/keepalived/keepalived.conf y contiene tres partes:

**global\_defs;** en esta primera parte tenemos las definiciones globales. En mi caso solo tengo el identificador del director. En este apartado se puede configurar el envío de correos si se tiene un servidor de correos. Mi apartado queda así.

**virtual\_server;** directivas de cada servidor virtual. Aquí se define la ip virtual y el puerto. Se define tantas como ipvirtuales o servicios tengamos. En este apartado encontramos las opciones:

- delay\_loop, intervalo de chequeo en segundos.
- lb\_algo, algoritmo de reparto.
- lb\_kind, método de redirección.
- protocol, protocolo que va a redirigir.
- sorry\_server, donde redirigir las peticiones en caso de que todos los servidores reales estén caídos.



- `real_server`, especifica un servidor real
- `wight`, peso que tiene el servidor real a la hora del balanceo.
- `TCP_CHECK`, en este apartado se especifica que chequee el puerto 80 TCP con un tiempo de espera.

**`rrp_instance`**; éstas son las directivas de los directores. Y sus opciones son:

- `state`, rol que va a tomar (MASTER o BACKUP)
- `interface`, interfaz de red de ese nodo.
- `lvs_sync_daemon_inteface`, interfaz para la sincronización de lvs
- `virtual_router_id 51`, router virtual de la instancia.
- `priority`, prioridad del nodo, el nodo principal debe tener un número mayor.
- `advert_int`, intervalo entre chequeos en segundos.
- `smtp_alert`, para que nos alerte por SMTP, en mi caso no hace nada.
- `Authentication`, en esta sección se configura el tipo de autenticación entre los balanceadores.
- `auth_type`, tipo de contraseña.
- `auth_pass`, contraseña.
- `virtual_ipaddress`, en esta sección se define la dirección ip del cluster

## 5.6 Apache

### 5.6.1 EXPLICACION

Un servidor web es un programa que implementa el protocolo HTTP. Se ejecuta continuamente en un ordenador manteniéndose a la espera de recibir peticiones por parte de un cliente.

Como servidor web se ha escogido apache. Apache es un servidor web HTTP de código abierto que implementa el protocolo HTTP. Es el servidor web más utilizado y es altamente configurable.

### 5.6.2 INSTALACION

Para la instalación, la podemos hacer mediante los repositorios:

```
# apt install apache2
```

En los repositorios también tenemos muchos módulos que podemos ir añadiendo a apache para hacerlo más configurable.

### 5.6.3 CONFIGURACION

Los ficheros de configuración de apache se encuentran en /etc/apache2, pero yo utilizaré solo los que se encuentran en el directorio /etc/apache2/sites-available/. Tras crear el fichero en ese directorio tendremos que crear un enlace a ../sites-enabled, apache incorpora una herramienta que lo hace automáticamente llamada a2ensite . Estos ficheros de configuración son para configurar diferentes sitios virtuales.

Tras instalar LVS, para probar que funcionaba correctamente, simplemente modifiqué el archivo /var/www/index.html y puse el nombre de la máquina. Con eso podía saber a qué máquina me había enviado el balanceador. Tras saber la máquina en la que estaba, le quitaba la red y volvía a cargar la página para confirmar que el balanceador me dirigía al otro servidor real.

Cuando ya estaba seguro de que los balanceadores me funcionaban correctamente, me puse a modificar apache para que estuvieran los archivos sincronizados y cree varios sitios virtuales. Lo primero que hice para sincronizarlo, fue parar el servicio y copiar los directorios en /opt, que es donde se encuentra montado el disco drbd1. Después borre los originales.

```
# /etc/init.d/apache2 stop
# mkdir /opt/etc
# mkdir /opt/var/
# cp -pr /etc/apache2 /opt/etc/
# cp -pr /var/www /opt/var
# rm -r /etc/apache2/*
# rm -r /var/www/*
```

Ahora tenemos que hacer que los ficheros sigan en sus directorios. Para esto decidí utilizar el mount con la opción --bind, para hacerlo permanente añadimos las siguientes líneas a el fichero /etc/fstab. Estas líneas tienen que ir después que la que añadimos para

```
/opt: /opt/etc/apache2 /etc/apache2 none rw,bind 0 0
/opt/var/www /var/www none rw,bind 0 0
```

Por último, montamos las particiones e iniciamos el servicio.

```
# mount -a
# /etc/init.d/apache2 start
```

La parte de borrar y montar las particiones, tendremos que hacerla en los dos servidores, la de la copia, solo en uno. Tras esto pasamos a la configuración de los sitios virtuales. Para ello creamos dos ficheros de configuración. El primero es para saber en qué máquina nos encontramos y el segundo para que nos dirija a drupal.

El segundo fichero de configuración que tenemos que crear en `/etc/apache2/sites-available/` es el de **mysite**. Es muy parecido al anterior, solo especificamos que si entramos con la dirección **mysite** nos dirija al directorio `/var/www/mysite`:

```
<VirtualHost *:80>

    ServerAdmin webmaster@localhost
    ServerName mysite

    DocumentRoot /var/www/mysite

    <Directory /var/www/mysite>
        RewriteEngine on
        RewriteBase /mysite
        RewriteCond %{REQUEST_FILENAME} !-f
        RewriteCond %{REQUEST_FILENAME} !-d
        RewriteRule ^(.*)$ index.php?q=$1 [L,QSA]
    </Directory>
</VirtualHost>
```

Tras

tener los ficheros de configuración tenemos que añadirlos, y eliminar el que trae por defecto. También tenemos que añadir los nuevos directorios.

```
# a2dissite 000-default
```

```
# a2ensite mysite
```

```
# mkdir /var/www/mysite
```

Para que el fichero de configuración **mysite** sea el de por defecto entramos en `/etc/apache2/sites-enabled/` y le ponemos 000 delante, así apache lo leera primero.

```
# cd /etc/apache2/sites-enabled/
```

## 5.7 PHP 7.2

### 5.7.1 EXPLICACION

PHP es un lenguaje de programación interpretado, diseñado originalmente para la creación de páginas web dinámicas y puede ser incrustado en páginas HTML. Es usado normalmente para que el servidor ejecute un código. El cliente le envía la petición, el servidor la procesa y le envía la respuesta

### 5.7.2 INSTALACION

Para la instalación podemos hacerla desde los repositorios. En nuestro caso, como vamos a utilizar drupal tenemos que instalar un php específico. También tendremos que instalar php para que pueda leer de bases de datos postgresql.

```
# apt install php7.2 php7.2-gd php7.2-pgsql
```

### 5.7.3 CREACION DE UN PROYECTO

Tras esto, creamos la página que nos sirve para comprobar que php está funcionando y poder saber en qué servidor nos encontramos. Está es la página:  
cat /var/www/ha/index.php

```
<html>
  <head>
    <titlte>Server</title>
  </head>
  <body>
    <?php phpinfo(); ?>
  </body>
</html>
```

Por último, reiniciamos apache

```
# /etc/init.d/apache2 restart
```

## 5.8 PostgreSQL

### 5.8.1 EXPLICACION

Un servidor de base de datos es un programa que provee servicios de base de datos a otros programas u otras computadoras, como es definido por el modelo cliente-servidor.

En aplicaciones web, lo más utilizado es MySQL, aunque en algunas también podemos utilizar PostgreSQL. En mi caso para salir de lo más común he utilizado PostgreSQL.

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD

### 5.8.2 INSTALACION

La instalación la he realizado desde los repositorios.

```
# apt install postgresql
```

### 5.8.3 CONFIGURACION

Los ficheros de configuración de PostgreSQL, se encuentran en /etc/postgresql y las bases de datos en /var/lib/postgresql. En nuestro caso no tenemos que tocar ningún fichero de configuración.

Cuando tengamos instalado postgresql, tenemos que, al igual que con apache, sincronizarlos. Es lo mismo, pero con diferentes directorios. Primero paramos la base de datos, copiamos los ficheros necesarios, borramos los antiguos, montamos los nuevos e iniciamos el servicio.

```
# /etc/init.d/postgresql stop
# cp -pr /etc/postgresql /opt/etc
# cp -pr /etc/postgresql-common/ /opt/etc
# mkdir /opt/var/lib
# cp -pr /var/lib/postgresql/ /opt/var/lib
# rm -r /etc/postgresql/*
# rm -r /etc/postgresql-common/*
# rm -r /var/lib/postgresql/*
# echo '/opt/etc/postgresql /etc/postgresql none rw,bind 0
0'>>/etc/fstab
# echo '/opt/etc/postgresql-common /etc/postgresql-common none
rw,bind 0 0'>>/etc/fstab
```

```
# echo '/opt/var/lib/postgresql /var/lib/postgresql none rw,bind  
0 0'>>/etc/fstab
```

```
# mount -a
```

```
# /etc/init.d/postgresql start
```

Este proceso, al igual que con apache, tenemos que desactivarlo y añadirlo al rc.local:

```
# apt install chkconfig
```

```
# chkconfig -d apache2
```

Y añadimos la siguiente línea a /etc/rc.local, después del mount -a.

```
# /etc/init.d/postgresql start
```

#### 5.8.4 CREACIÓN DE USUARIO Y BASE DE DATOS

Tras esto vamos a crear un usuario drupal y una base de datos drupal. Para crear el usuario, nos tenemos que loguear como usuario postgres y después entrar en la base de datos. Una vez creado el usuario, nos salimos y volvemos a iniciar postgres, esta vez con el usuario drupal, y creamos la base de datos.

```
# sudo su postgres
```

```
$ psql
```

```
postgres=# CREATE USER drupal with password 'drupal'
```

```
postgres=# createdb;
```

```
\q
```

```
$ psql -U drupal -h localhost -d template1
```

```
template1=> CREATE DATABASE drupal;
```

```
postgres=# CREATE USER drupal with password 'drupal'
postgres=# createdb;
CREATE ROLE
postgres=# \q
postgres@empresa3:~$ ps
ps                ps2pdf13          ps2txt            psicc             pstree
ps2ascii          ps2pdf14          psfaddtable       psidtopgm         pstree.x11
ps2epsi           ps2pdfwr          psfgettable       pslog
ps2pdf            ps2ps             psfstriptime      psql
ps2pdf12          ps2ps2            psfxtable         pstopnm
postgres@empresa3:~$ psql -U drupal -h localhost -d template1
Password for user drupal:
psql (10.19 (Ubuntu 10.19-0ubuntu0.18.04.1))
SSL connection (protocol: TLSv1.3, cipher: TLS_AES_256_GCM_SHA384, bits: 256, co
mpression: off)
Type "help" for help.

template1=> CREATE DATABASE 'drupal';
ERROR:  syntax error at or near "'drupal'"
LINE 1: CREATE DATABASE 'drupal';
                        ^
template1=> CREATE DATABASE drupal;
CREATE DATABASE
template1=>
```

## 5.9 Drupal

### 5.9.1 EXPLICACION

Drupal es un sistema de gestión de contenido modular y muy configurable.

Es un programa de código abierto, con licencia GNU/GPL, escrito en PHP, desarrollado y mantenido por una activa comunidad de usuarios. Destaca por la calidad de su código y de las páginas generadas, el respeto de los estándares de la web, y un énfasis especial en la usabilidad y consistencia de todo el sistema.

El diseño de Drupal es especialmente idóneo para construir y gestionar comunidades en Internet. No obstante, su flexibilidad y adaptabilidad, así como la gran cantidad de módulos adicionales disponibles, hace que sea adecuado para realizar muchos tipos diferentes de sitios web.

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.

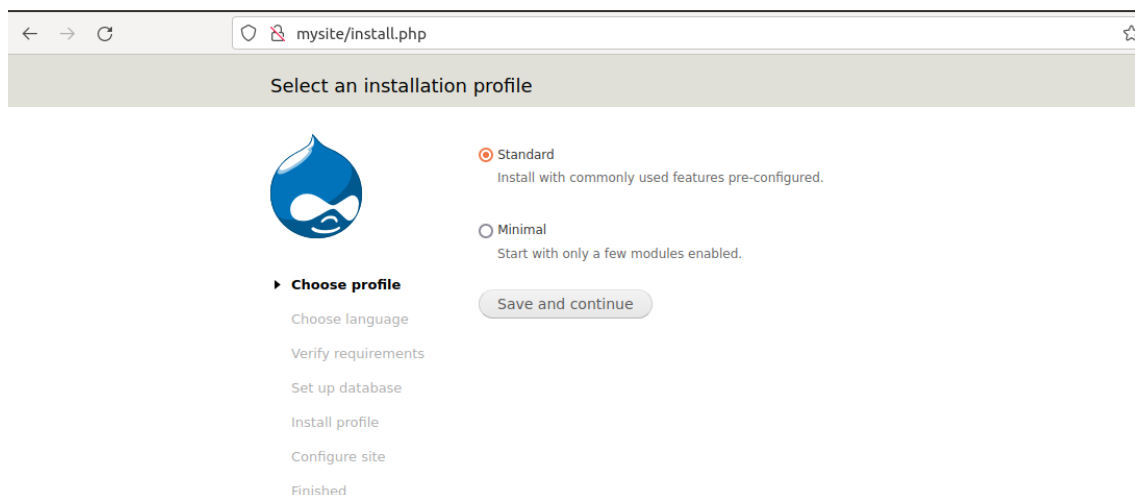
### 5.9.2 INSTALACION

Para la instalación, me he descargado de la página principal, un archivo comprimido que descomprimiremos en /var/www. Se descomprime como drupal-7.14, lo renombraremos a drupal, ya que es como configuramos el apache. También nos descargaremos el idioma en español.

```
# wget "http://ftp.drupal.org/files/projects/drupal-7.14.tar.gz"
# wget "http://ftp.drupal.org/files/translations/7.x/drupal/drupal7.13.es.po"
# cd /var/www
# tar -zxvf /tmp/drupal-7.14.tar.gz
# mv /drupal-7.14 drupal
# cp /tmp/drupal-7.13.es.po /var/www/drupal/profiles/standard/translations/
# chown www-data:www-data /var/www
```

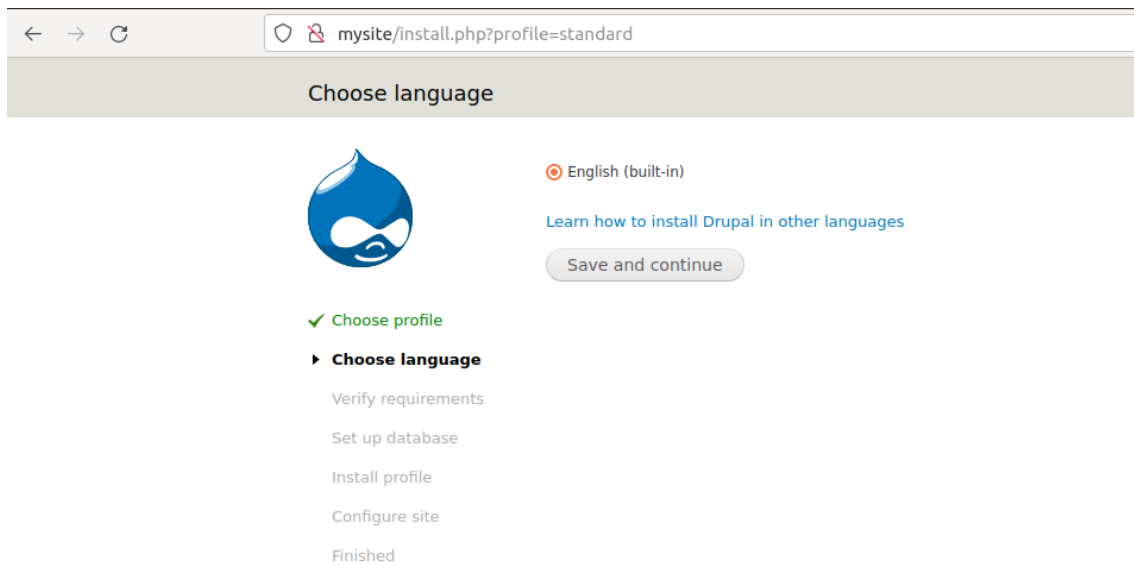
Ahora nos metemos en el navegador, ponemos la dirección de nuestro sitio y seguimos los pasos.

Le damos a instalación estándar





Elegimos el módulo de inglés, que es el único que viene por defecto



Choose language

English (built-in)

[Learn how to install Drupal in other languages](#)

Save and continue

✓ Choose profile

► Choose language

Verify requirements

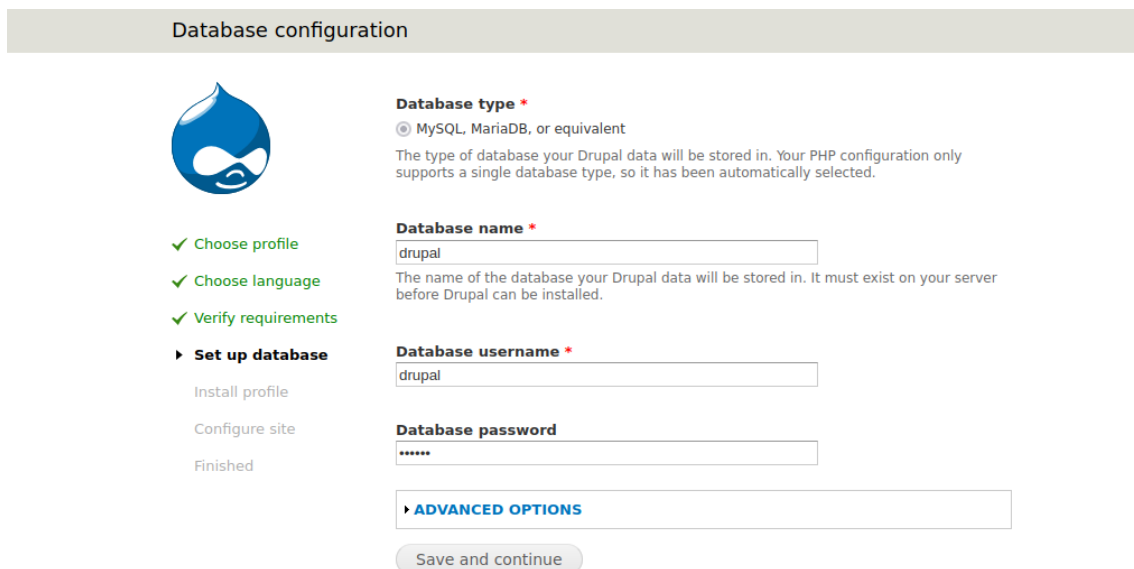
Set up database

Install profile

Configure site

Finished

Rellenamos los datos de nuestra base de datos.



Database configuration

Database type \*

☒ MySQL, MariaDB, or equivalent

The type of database your Drupal data will be stored in. Your PHP configuration only supports a single database type, so it has been automatically selected.

Database name \*

drupal

The name of the database your Drupal data will be stored in. It must exist on your server before Drupal can be installed.

Database username \*

drupal

Database password

\*\*\*\*\*

► ADVANCED OPTIONS

Save and continue

✓ Choose profile

✓ Choose language

✓ Verify requirements

► Set up database

Install profile

Configure site

Finished

## 6. Conclusión

Como conclusión del proyecto he conseguido cumplir los objetivos propuestos:

- He conseguido poner en funcionamiento el cluster con las máquinas virtuales correctamente configuradas.
- He implementado un servidor DHCP que asigna IPs a los nodos del cluster.
- Las cuatro máquinas se comunican correctamente.
- La implementación del balanceo de cargas ha sido correcta (ipvsadm).
- El servidor web Apache se ha comunicado con la base de datos correctamente
- La base datos incorpora un usuario de ejemplo con un entorno web llamado **Drupal**

## 7. Referencias

<http://www.clusterinformatica.blogspot.com/2011/05/cluster-informatica.html>

<https://www.okitup.com/cluster-web-de-alta-disponibilidad/>

<http://www.estrellateyarde.org/virtualizacion/clusters-ha-con-lvs>

[https://es.ryte.com/wiki/Keep\\_Alive](https://es.ryte.com/wiki/Keep_Alive)

<https://programmerclick.com/article/9145768439/>

<https://www.drupal.org/documentation>

<https://www.postgresql.org/docs/current/>

<https://linbit.com/drbd/>

<https://tecadmin.net/setup-ip-failover-on-ubuntu-with-keepalived/>

[https://docs.oracle.com/cd/E37670\\_01/E37355/html/ol\\_ocfs2.html](https://docs.oracle.com/cd/E37670_01/E37355/html/ol_ocfs2.html)

<https://programmerclick.com/article/798474112/>

<https://httpd.apache.org/docs/>

<https://www.php.net/docs.php>

<https://www.ticarte.com/contenido/drupal-8-guia-de-referencia>

<https://www.dataprix.com/es/book/export/html/194368>

<https://www.todopostgresql.com/como-crear-base-de-datos-en-postgresql/>

<https://comoinstalar.me/como-instalar-drupal-en-ubuntu-18-04-lts/>

<https://www.linuxito.com/gnu-linux/nivel-basico/911-crear-una-base-de-datos-y-usuario-en-un-servidor-postgresql>

<https://ricardogeek.com/conceptos-basicos-sobre-balanceo-de-carga/>