Balanceo de carga entre servidores Debian GNU/Linux

Manual original, creado por David Sanchez Cantero.
Mas conocido en la red por Forat.

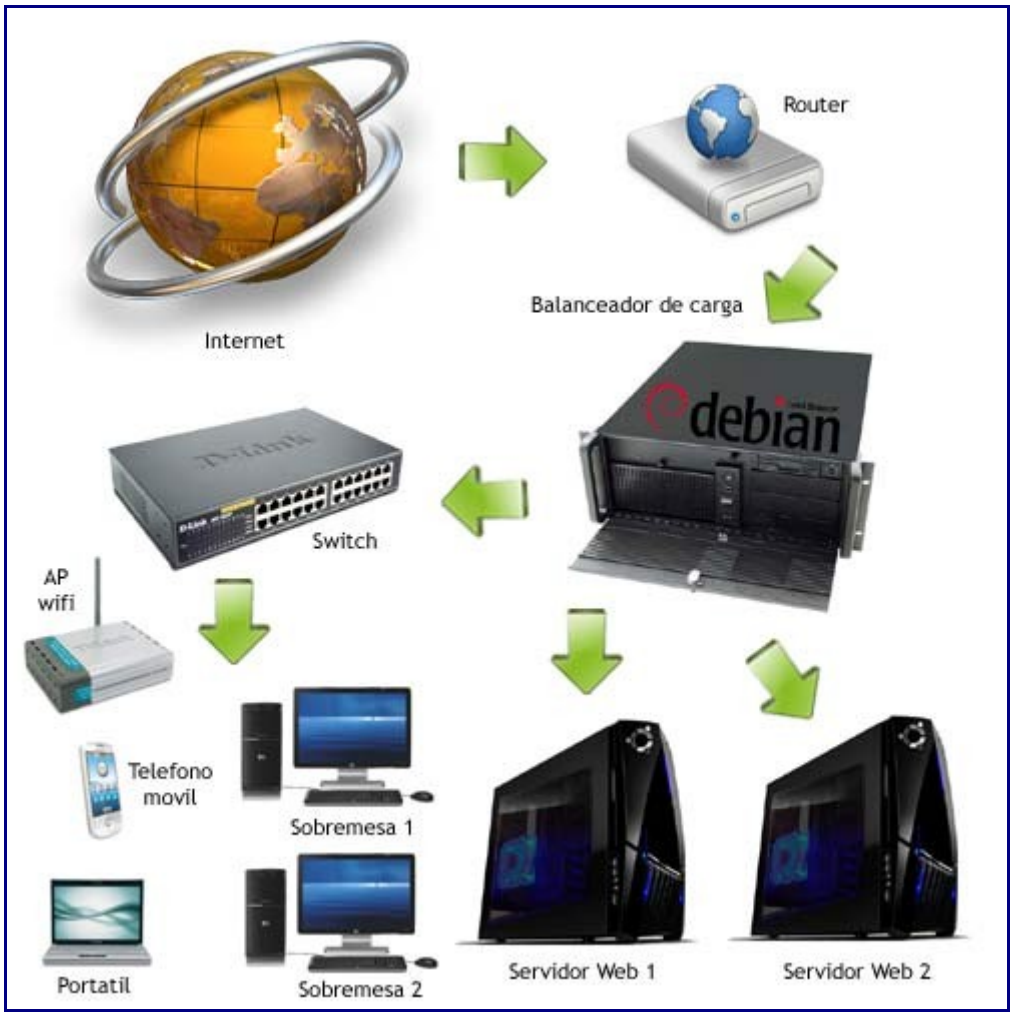
Sitio web oficial donde se fabrican este y otros proyectos
http://www.forat.info/

Este y otros proyectos listos para descargar en formato PDF
http://project.forat.info

Recordemos que este balanceador de carga es capaz de repartir la carga entre varios servidores basándose en el método **Round Robin** basado en **Software Libre PEN Load Balancer**. Este mismo trata de dar paso a cada uno de los servidores correlativamente. **PEN** es capaz de colgar de el y repartir la carga entre 15 servidores como máximo.

Comenzamos con el temario ...

No necesariamente tenemos que usar estas tarjetas de red para balancear la carga ya que podemos configurar a **PEN Load Balancer** para que envié carga a dos servidores directamente conectados y la tercera tarjeta conectarla a un **Switch** y equipar de acceso a internet a toda tu red como podemos ver en la siguiente imagen …

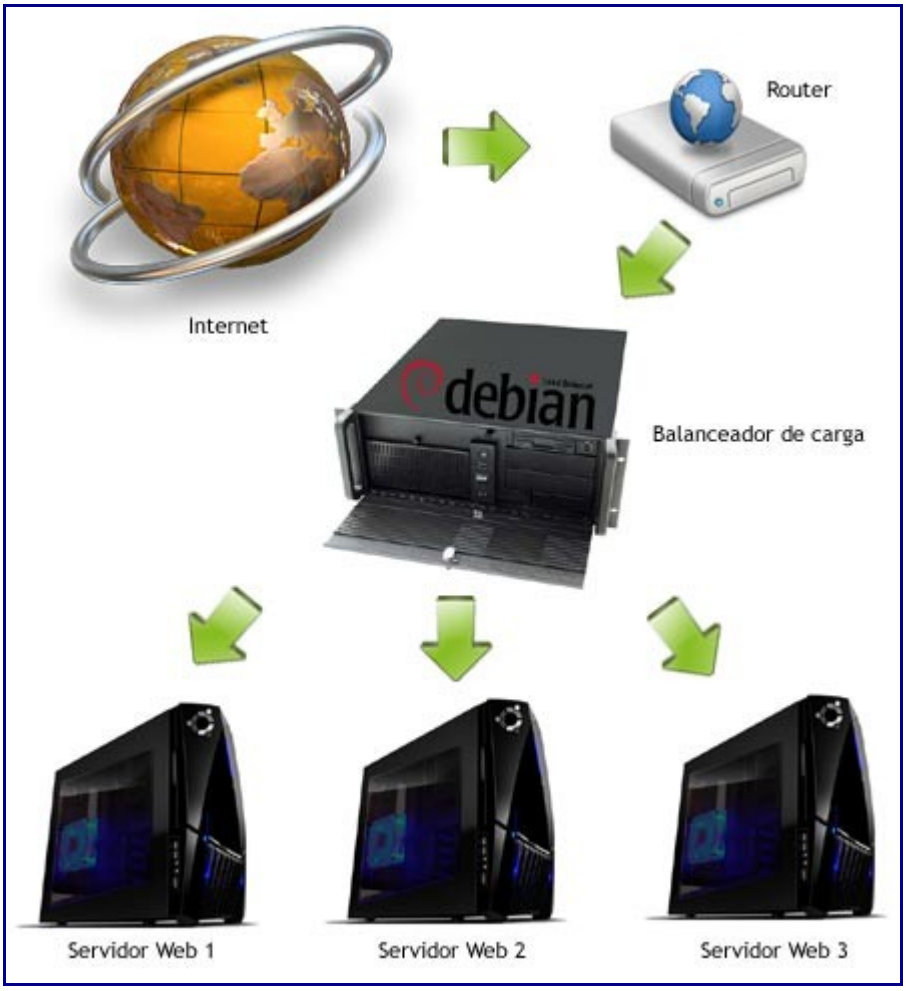


Una vez mas he tratado en esta mini saga de tutoriales que forman este ultimo proyecto que se entienda el como y para que se montan este tipo de maquinas usando solo los programas que necesitamos, gratuitamente y con el hardware justo sin gastar recursos de Hardware innecesarios.

Introducción

No suelo montar las cosas por que si, si no por que realmente las necesito y esta vez vamos a fabricar una maquina que hace tiempo quería montarme. En esta ocasión me encuentro con que necesito **montar un balanceo de carga entre varios servidores**. Cuando la *carga de un servidor* (*en este caso Web*) es muy alta puede hacer que la maquina se sature y no sea capaz de cumplir su función hasta que se vayan cerrando peticiones y con ellas liberando memoria … Hay muchísimas configuraciones que podemos aplicarle a nuestro **servidor** para que **consuma lo mínimo de memoria** y que a su vez libere lo mas rápido posible lo consumido. Cuando ya hemos aplicado a nuestro **servidor** todo lo que sabemos tenemos la opción de montar otros servidores y compartir esa **carga** que tiene nuestro servidor principal.

Veamos un esquema de lo que vamos a montar …



Este proyecto puede montarse directamente tan solo con dos **servidores** prescindiendo del **bridge** o también llamado (**puente de red**) pero yo prefiero montar un ordenador principal que cuelgue del **Router** y este distribuya la carga entre los servidores. A este mismo ordenador mas adelante le iré incluyendo mas cosas y por esto mismo quiero montarlo en un ordenador aparte.

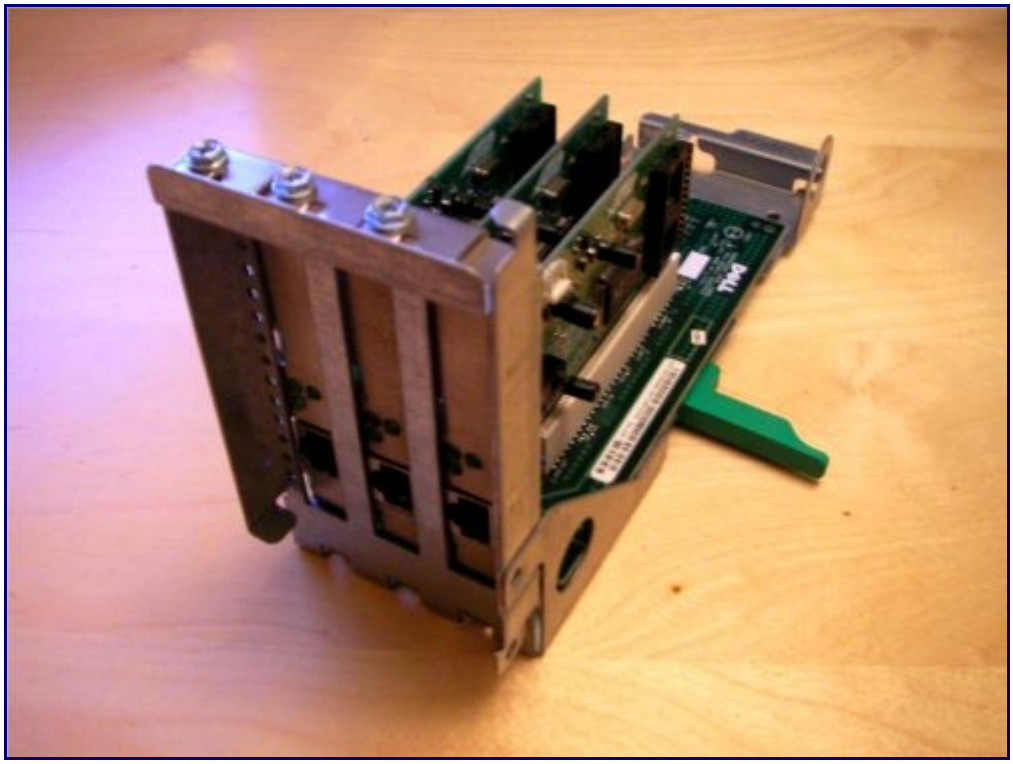
Podríamos usar un **Switch** para conectar todos los ordenadores pero como tengo tarjetas de red para parar un camión voy a agregarle 3 mas además de la que tiene la placa base …

En una conectaremos nuestro router y las otras tres serán ocupadas por los tres servidores que repartirán su carga entre si. La forma que vamos a usar para repartir la carga a nuestros servidores no es otra que la de (una tu y una yo) también llamado método (**round-robin**) respondiendo a peticiones de clientes una vez cada uno.

Esto quiere decir que los tres **servidores** tienen que tener lo mismo exactamente por que si no cada usuario vera cosas diferentes conforme vaya cargando la web y esto puede ser un desastre.

Tengo pensado agregarle otras opciones para que este ordenador cumpla con mas funciones que la de **montar un puente de red** y **distribuir la carga** de los servidores que tengo pero ahora mismo necesito esto mismo y ya que me lo voy a montar aprovecharemos la ocasión para publicarlo y si alguien necesita montarse un **balanceo de carga** rápido y gratis aquí tendrá la solución.

Hardware



Esta maquina tan solo es una practica de algo que puede ser fabricado a gran escala. Este **balanceador de carga** se ocupará de **distribuir el trafico** entrante desde el **Router** hacia los tres **servidores web** que colgarán de el. Si hablamos sobre el **Hardware** necesario para montar esta maquina la respuesta dependerá de el uso que le demos a lo que sirvamos hacia Internet. Si hay algo en lo que necesitamos fijarnos a la hora de montar un **balanceador de carga** es en las **tarjetas de red**, claro que todo el **hardware** que tenga nuestra maquina cuanto mejor sea mas fluido será su trabajo y menos problemas tendremos en el futuro.

En este caso estoy tirando del **Hardware** que tengo y hace tiempo que hay un viejo amigo aparcado en el olvido al que hoy de nuevo vamos a darle uso …

Se trata de un **DELL OptiPlex GX110** de aquellos que te avisan en el arranque que tienes la tapa abierta, bonito, bonito !!

En su interior no hay mas que un **procesador Intel Pentium III a 733 Mhz, 512 Mb de ram y 20 Gb de disco duro**. Este ordenador esta mas que sobrado para **montar un balanceador de carga** y que aguante un trafico aceptable sin que llegue a colapsarse. Si hay algo que me gusta de montarme proyectos en casa es porque si lo monto y veo mas adelante que el proyecto crece siempre puedo ampliar el hardware costando lo justo o tirando de maquinas que tenga por aquí pero mas potentes.

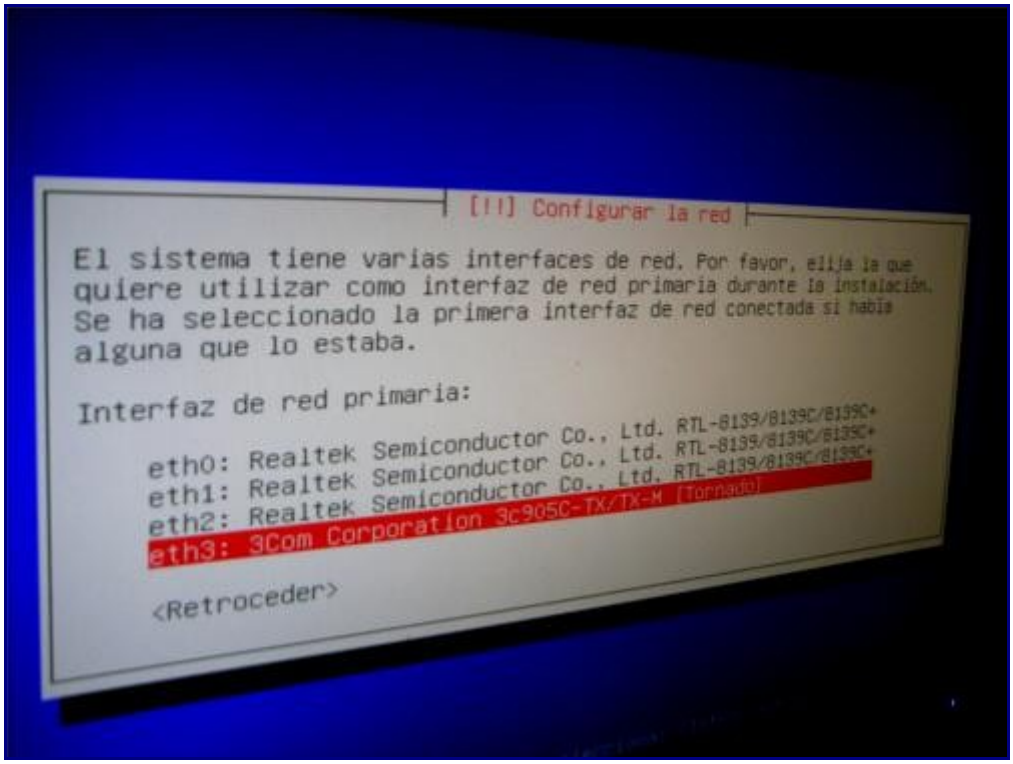
No puedo decir cuales son los requisitos sobre el **hardware** necesario para montar este **balanceador de carga** así que con el mio ya podemos tener un ejemplo de algo aceptable pero no profesional.

Si me tuviera que **montar un balanceador de carga** desde cero lo primero en lo que mostraría interés seria en las **tarjetas de red** ya que cuanto mas rápidas sean mas rápido pasará el trafico por ellas y mas ancho de banda aguantara.

En este proyecto montaré *3 Realtek* con *velocidad de 10/100 Mbps* y aprovecharé la que incluye la placa base. Esta integrada es una *3Com* y también es de *10/100 Mbps* la cual irá conectada hacia el Router.

Sistema Operativo

El **como instalar Linux Debian** es mas o menos como su anterior versión así que no veo muy necesario montar otro manual para instalar esta distribución. Tan solo tenemos que tener en cuenta durante su instalación lo siguiente …



Como podemos ver aquí están todas las **tarjetas de red** que tenemos instaladas en nuestra maquina.

En mi caso tres **Realtek** y una **3Com** integrada en la **placa base**. Tenemos que seleccionar la que tengamos conectada a nuestro *Router* ya que durante la instalación necesitamos paquetes que descargaremos de Internet …

En la selección de programas listos para instalar tan solo vamos a seleccionar (**Sistema estándar**) para que instale lo mínimo y a partir de hay instalaremos solo lo que necesitemos.

Esta es una de las cosas buenas que tiene **Linux Debian** (*podemos adaptar nuestro sistema a lo que necesitemos de una manera fácil y eficaz*) …

Una vez finalizada la instalación del sistema extraemos el CD y reiniciamos el ordenador. A continuación nos identificamos como súper usuario del sistema (**root**) y voila !! Ya tenemos el sistema operativo base instalado y tan solo consume 27,5 Mb de memoria, excelente !

Actualización …

Una de las primeras cosas que debemos hacer una vez instalado el sistema es actualizar los repositorios y actualizar todos los paquetes del sistema. Esto es algo a tener en cuenta ya que con esto podemos tener las ultimas versiones de todo lo que tenemos instalado, así evitaremos fallos de seguridad y disfrutaremos de las nuevas mejoras.

Para comenzar actualizaremos los repositorios tecleando lo siguiente …

~ codigo ~

```
apt-get update
```

Ya ahora que tenemos los repositorios actualizados, actualizaremos todos los programas si es que hay alguno que actualizar con …

~ codigo ~

```
apt-get upgrade
```

Estos dos pasos es recomendable hacerlos al menos una vez a la semana teniendo siempre en cuenta que cuando el (**upgrade**) actúe actualizará todas las versiones de lo que encuentre desactualizado así que si tenemos programas configurados no esta de mas revisarlo a ver si funciona con normalidad y si no es así buscar primero en los archivos que acabamos de actualizar.

Puente de red o Bridge y acceso remoto SSH

Este ordenador que estoy montando tiene instaladas cuatro **tarjetas de red** las cuales se usarán para conectar el **Router** y las otras para conectar directamente nuestros **servidores**. Si tenemos un ordenador conectado las 24h a la red podemos ahorrarnos el tener un switch conectado a la red eléctrica las 24 h. Por norma los **switch** o **hub de red** consumen al menos 12 voltios y no es mucho pero si tenemos unas cuantas tarjetas de red y nuestro ordenador encendido podemos conectarnos directamente a el para salir a internet y prescindir de un aparato mas consumiendo energía. Este es uno de los posibles usos que podemos darle a este puente de red que vamos a montar en este articulo aunque si estamos montando un **balanceo de carga** también lo vamos a necesitar …

El **puente de red** o también llamado **Bridge** que vamos a montar con cuatro **tarjetas de red** no es que vaya a tener cuatro IPs diferentes si no una que identifique nuestra maquina de cara al **Router** así podremos asignarle un puerto especifico (en este caso el 80) y redirigir el trafico hacia ella, posteriormente hacia nuestros servidores quedando las otras tres tarjetas de manera transparente.Si seguimos al pie de la letra la **instalación de Linux Debian** tan solo instalamos lo esencial para sobrecargar la maquina lo menos posible con software que no necesitemos. Ahora vamos a instalarle el software llamado (**bridge-utils**) el cual nos facilita herramientas para poder **montar el Bridge**.

Lo instalamos con …

~ codigo ~

```
apt-get install bridge-utils
```

Una vez instalado este software y contando con que *no configuramos las tarjetas de red* anteriormente vamos a crear una configuración especifica que montará (**br0**) como una tarjeta de red y hará que las cuatro tarjetas actúen como una sola, con una sola dirección IP y de manera transparente. Bonito, bonito !!

Para poder enviar todas las peticiones de un **puerto** especifico desde el **Router** hacia un ordenador en cuestión necesitamos que este tenga una IP fija y no dinámica. Si en vez de cuatro tarjetas tenéis tres o dos no es problema ya que tan solo tendréis que agregar vuestras interfaces en la linea (*bridge_ports eth0 eth1 eth2 eth3*). Sabidos estos cuatro términos vamos a editar el archivo (**interfaces**) y después lo configuraremos.

Antes de hacer esto no esta de mas que tengamos una copia de seguridad del archivo existente así siempre podremos recurrir a la versión existente del archivo en un momento dado. Hacemos la copia de seguridad con …

~ codigo ~

```
cp /etc/netwotk/interfaces
/etc/network/interfaces0K
```

Una vez hecha la copia de seguridad vamos a editarlo con …

~ codigo ~

```
vi /etc/network/interfaces
```

Borramos todo su contenido y agregamos lo siguiente …

~ codigo ~

```
auto lo
iface lo inet loopback

auto br0
iface br0 inet static
address 192.168.1.200
network 192.168.1.0
netmask 255.255.255.0
broadcast 192.168.1.255
gateway 192.168.1.254
bridge_ports eth0 eth1 eth2 eth3
bridge_maxwait 0
```

Las ips que incluye la configuración son las de mi red, tenéis que sustituir estas por la de vuestra red y cambiar los interfaces eth0 eth1 eth2 eth3 … por los vuestros dependiendo de las tarjetas de red que useis.

Una vez ajustado todo a nuestra medida guardamos y salimos del editor vi con las teclas (ESC) y seguidamente (:wq!). Ahora y para reiniciar la red y que coja las nuevas configuraciones tendremos que reiniciar el demonio con …

~ codigo ~

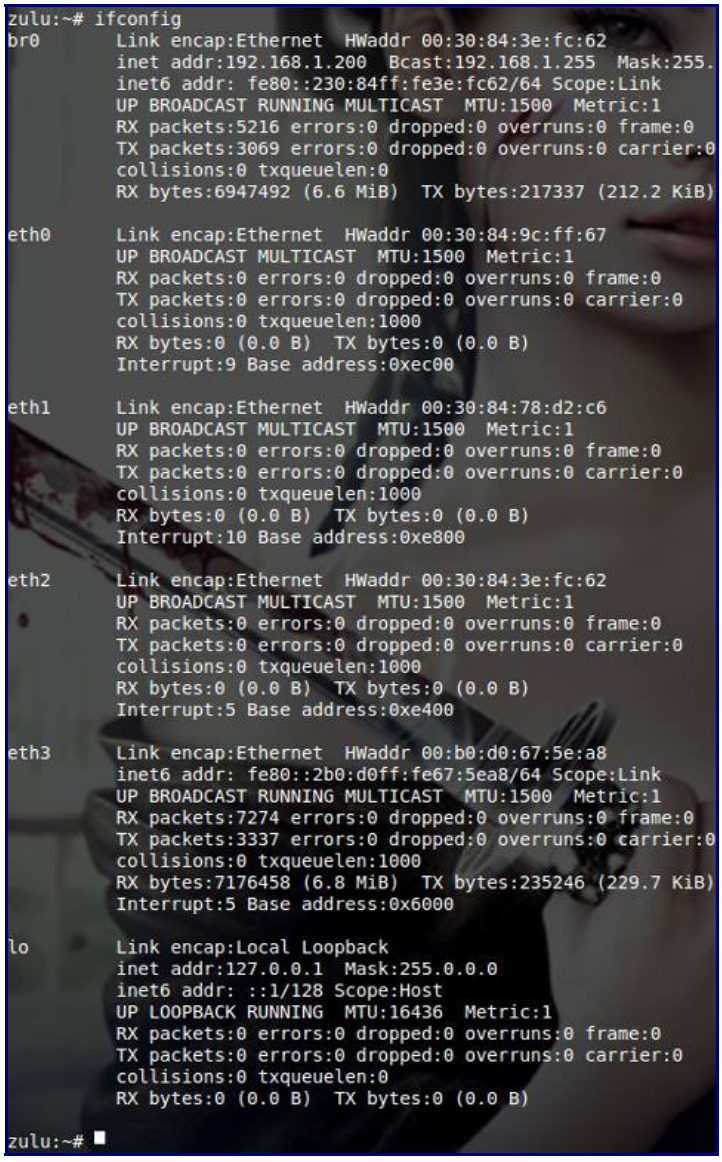
```
/etc/init.d/networking restart
```

Ahora para comprobar que todo fue bien y que se creo el interfaz (br0) con la ip 192.168.1.200 y las demás tarjetas se han quedado sin IP teclearemos el siguiente comando …

~ codigo ~

```
ifconfig
```

Y veremos algo como esto …



Ahora si conectamos cualquier ordenador a una de las tarjetas de red que quedan libres …

Recordemos que tendremos que configurar la tarjeta de red del ordenador cliente con la ip dentro de nuestro rango, en mi caso 192.168.1.x y como gateway la ip de nuestro futuro balanceador de carga.

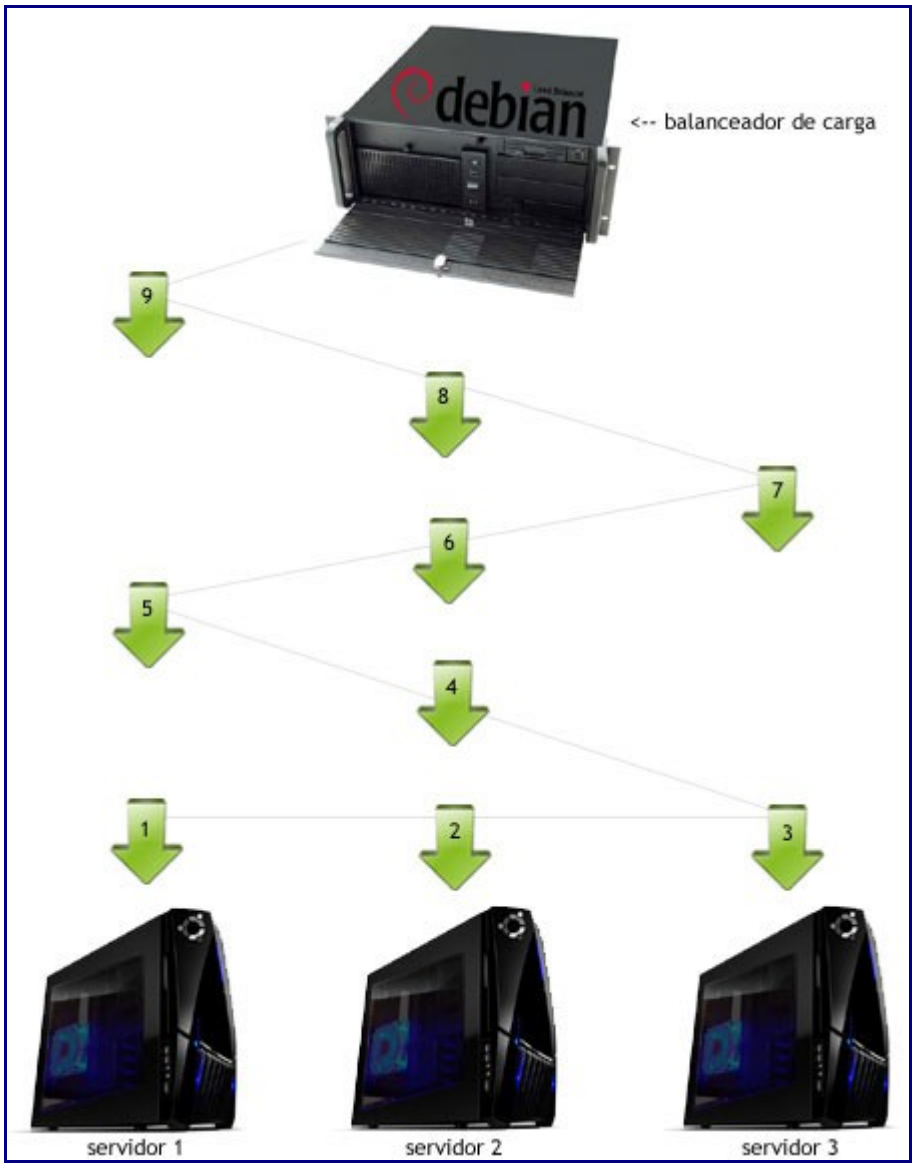
Ahora ya podemos navegar a través de el usándolo de *puente*, *ladrón*, *hub*, *switch* o como lo queramos llamar. Para terminar y como no podríamos incluirle el software para conectarnos remotamente desde una terminal vía **SSH**. Vamos a instalar el paquete …

~ codigo ~

```
apt-get install ssh
```

Si aun no conocéis las virtudes del acceso remoto por ssh os recomiendo echarle un ojo al tutorial (**Configuración de Red y manejo remoto vía OpenSSH con SSH y SFTP**) donde podremos aprender como explotarlo y así prescindir de un monitor mas en nuestra sala de ordenadores.

Balanceo de carga con PEN Load Balancer



Llegados a este punto donde ya tenemos el **Hardware** listo, **Linux Debian** instalado y el **puente de red o Bridge** funcionando vamos a instalar el software que montará el **balanceo de carga**. Hay varios tipos de software o soportes que están dispuestos a ofrecernos diferentes **balanceos de carga**. Yo no necesito un balanceo de carga extra sofisticado por lo que busco algo sencillo que cumpla con la función que yo necesito, que no es otra que la de compartir el trafico entre varios servidores y conseguir así una carga menor en cada uno de ellos …

En el esquema anterior podemos ver el funcionamiento de las peticiones que llegarán poco a poco hacia nuestros servidores. La técnica que aplicaremos a este **balanceo de carga** no es otra que la llamada **Round Robin**, esta técnica es muy valida para liberar carga de cualquier tipo de servidor tenga el sistema operativo que tenga ya que podemos repartir las peticiones entre varias maquinas unicamente hacia un puerto, rotando así conseguimos que a cada servidor le de tiempo a relajarse hasta que vuelva a tocarle además reduciremos las peticiones individualmente. Una vez le tocará responder a un servidor y así sucesivamente entre el grupo de servidores que tengamos.

PEN Load Balancer es el software que usaremos para montar este **balanceo de carga** practico y compatible con casi todos lo que necesitemos balancear hablando de **servidores web** por ejemplo. No es importante el sistema operativo que tenga instalado cada servidor ya que tan solo iremos repartiendo peticiones entre ellos hacia un puerto en cuestión.

Apuntes …

Cuando un usuario carga una web de uno de nuestros servidores no pasará la siguiente petición al siguiente servidor que esté en espera hasta que la misma se cierre. Por ejemplo si tenemos instalado el **servidor web apache** podemos configurar cuanto tiempo que queremos que esté quede en espera hasta que la misma sea cerrada. Si en todos los servidores que cuelguen de nuestro balanceador de carga configuramos que el TimeOut sea de 30 segundos se efectuara el cambio de servidor cada 30 segundos desde la ultima petición.

” Esto es algo que podemos dejarlo por defecto o configurarlo a nuestro gusto aunque he pensado que no estaría mal dar esta clase de apunte. “

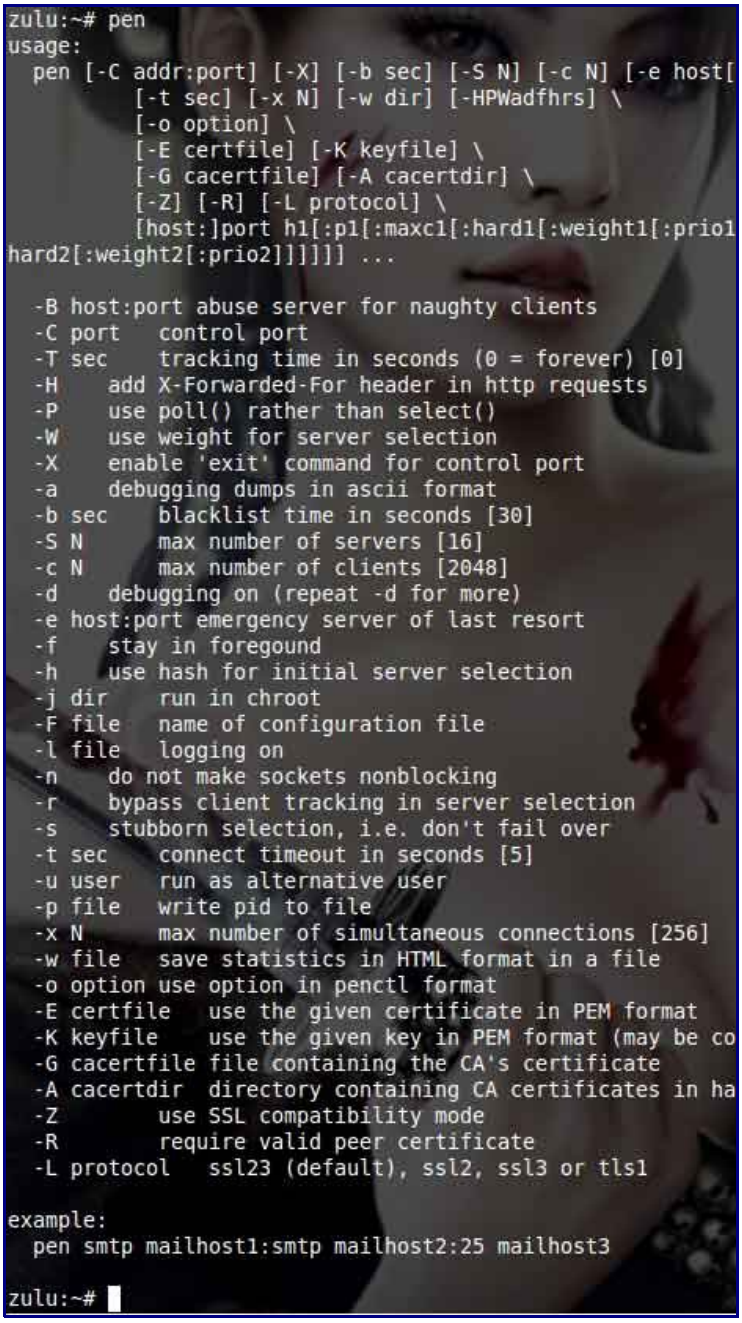
Instalación …

Bien después de tener el **puente de red o bridge** funcionando vamos a continuar por **instalar PEN** en nuestra maquina con …

~ codigo ~

```
apt-get install pen
```

PEN es tan simple que bajo una linea de comandos podremos ejecutarlo completamente configurado. Si introducimos (*pen*) en nuestra terminal veremos las opciones simplificadas de las que dispone …



Si queremos obtener una información mas extendida simplemente podemos teclear (*man pen*), recordando que podemos salir de la ayuda con la tecla (**q**).

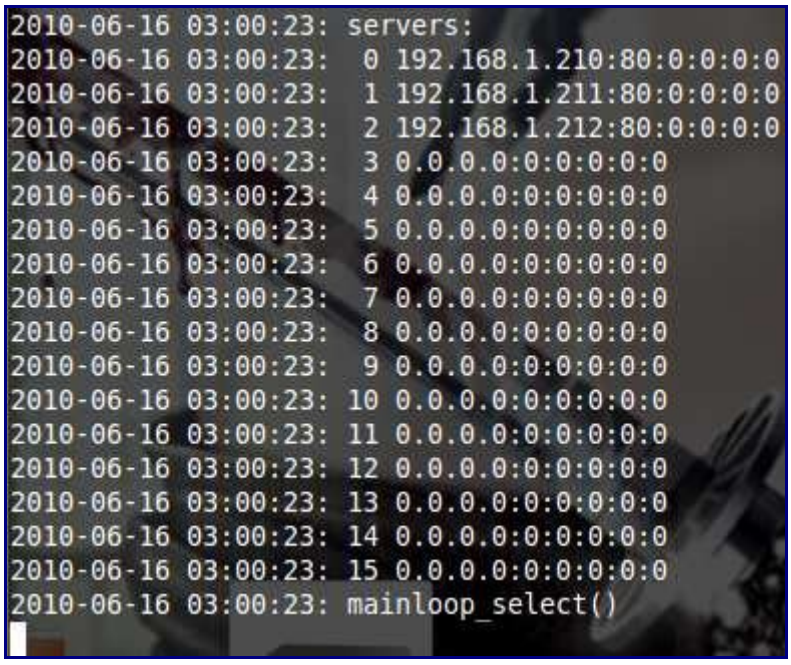
Ejecución …

Una vez instalado vamos lanzaremos a **PEN** con una simple linea de comandos como esta …

~ codigo ~

```
pen -r -a -f -d 192.168.1.200:80
192.168.1.210:80 192.168.1.211:80
192.168.1.212:80
```

Al final de la carga se queda en espera y muestra una tabla como esta …



En la tabla se listan los tres servidores donde balancearemos la carga y también vemos que se pueden incluir hasta 15 (*Ya me gustaría*).

Las 4 ips apuntando hacia el puerto 80 son las del balanceador de carga, servidor 1, servidor 2, servidor 3 y así podríamos continuar sucesivamente.

Bien, ahora ya tenemos nuestro **balanceador de carga** preparado para ir repartiendo peticiones hacia el puerto 80 entre las IPs de nuestros servidores.

Pruebas …

Para probar si su funcionamiento es correcto tan solo tenéis que colocar un contenido web diferente en cada raíz de vuestros servidores web y seguidamente abrir un navegador e introducir la IP de nuestro balanceador de carga. Si vamos recargando el navegador veremos que va cambiando el contenido de un servidor a otro conforme las peticiones de cada servidor web caduquen.

Si observamos el comando que lanzamos anteriormente incluye la opción **-d** que no es otra que la de hacer **Debug**, al acceder a uno de los servidores se reflejará en la pantalla donde lanzamos **PEN** como se conecta y como se desconecta caducando la conexión dando paso al siguiente servidor.

Ahora tenemos que incluir el mismo contenido web en los tres servidores para que no se note que el que nos visite en realidad está visitando la misma web pero mostrada por varios servidores. *Imaginaos cuanta carga pueden aguantar aquellas maquinas que tenemos en el desván si las conectamos en red hacia el balanceador de carga, PEN acepta hasta 15 servidores, esto puede ser de locos*.

Para terminar y para no tener que lanzar la linea con el comando **PEN** y su configuración cada vez que arranquemos la maquina vamos a **incluirlo en el arranque** y esta vez no incluiremos el modo **Debug** y se ejecutará en segundo plano.

Para ejecutar **PEN** al inicio del sistema vamos a incluirlo junto a su configuración dentro del archivo **rc.local** situado en el directorio /**etc**, en el podemos ejecutar programas rápidamente sin tener que crear un **script de arranque** …

~ codigo ~

```
vi /etc/rc.local
```

Una vez dentro incluimos nuestra linea de configuración sin la opción **-d** …

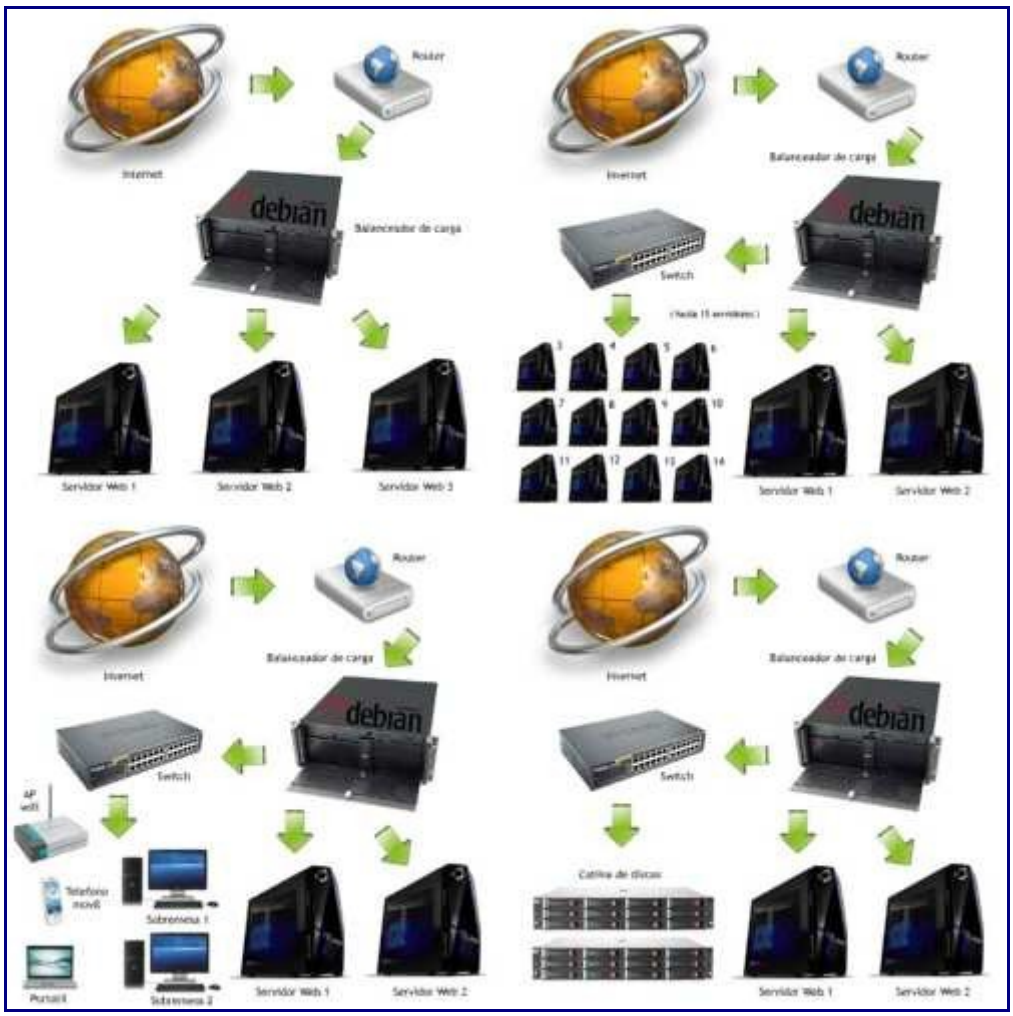
~ codigo ~

```
pen -r -a -f 192.168.1.200:80
192.168.1.210:80 192.168.1.211:80
192.168.1.212:80
```

Ahora guardamos y salimos del editor pulsando la tecla (**ESC**) y seguidamente (**:wq!**).

Una vez hecho ya podemos apagar la maquina y usarla cuando la necesitemos como balanceador lista para funcionar. 🙂

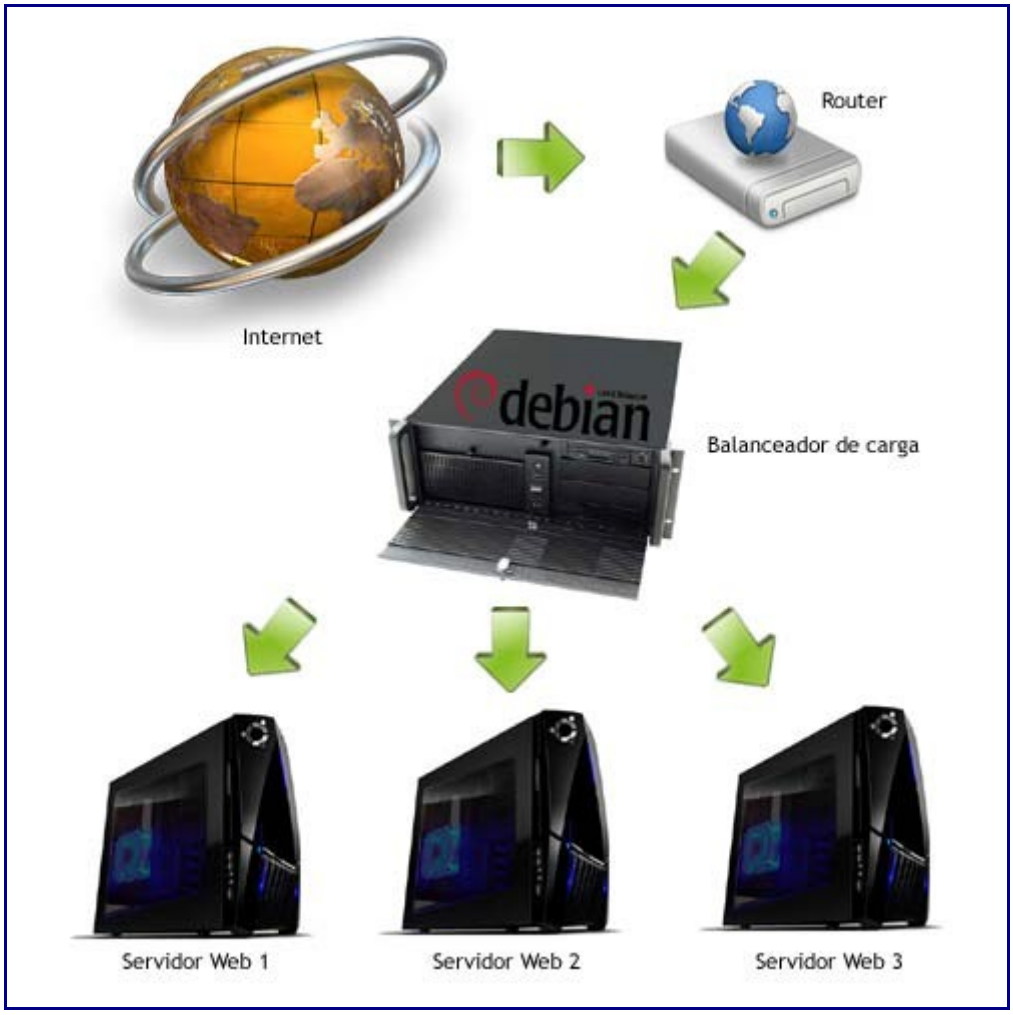
Posibles usos



Una vez tenemos todo instalado ya tenemos un **balanceador de carga**, lo que hace es exactamente esto (**Balancear la carga**) y así poder repartir entre varios servidores el peso del **trafico**, **transferencias** y con ello reducir en cada una de las maquinas que tengamos conectadas el consumo de **memoria**, **microprocesador**, **discos** y demás. Este tipo de trabajo que hace nuestro **balanceador de carga** también podemos encontrarlo en el mercado en forma de **Hardware** los cuales tienen un **Linux embedido** que cumple esta tarea y otras cuantas que hacen que se encarezca en el precio final de la maquina.

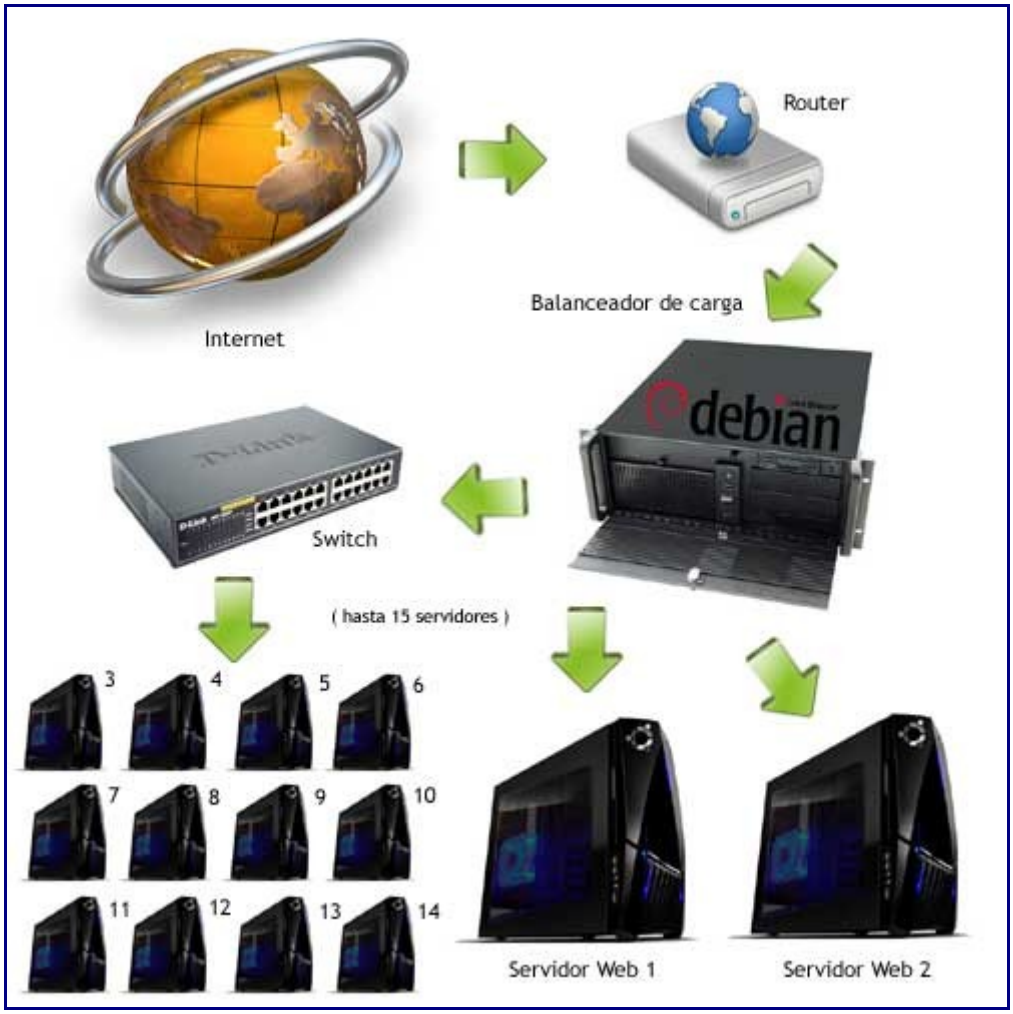
Nosotros con **Linux Debian**, algunas configuraciones y el software libre **PEN Load Balancer** hemos conseguido que una maquina que no cueste mucho dinero cumpla tal función al menos para empezar, ahora veamos que podemos hacer con ella y que usos podemos darle …

Método 1



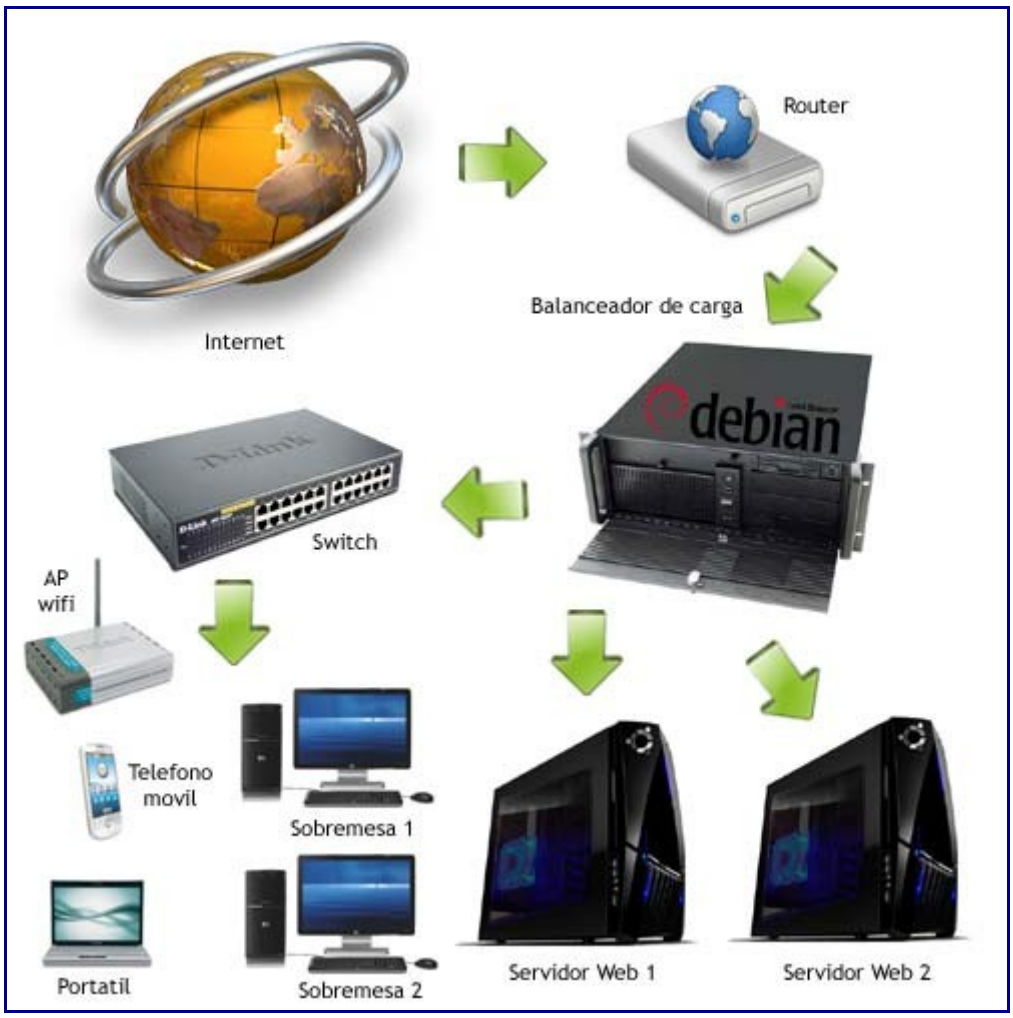
El método 1 es en el que nos hemos basado a la hora de montar este proyecto. Con el podemos colgar del balanceador a tres servidores que repartirán entre ellos la carga que les ocasione el acceso web. Esto es algo sencillo y puede que en un hogar no sea del todo practico pero para montar un alojamiento correcto si.

Método 2



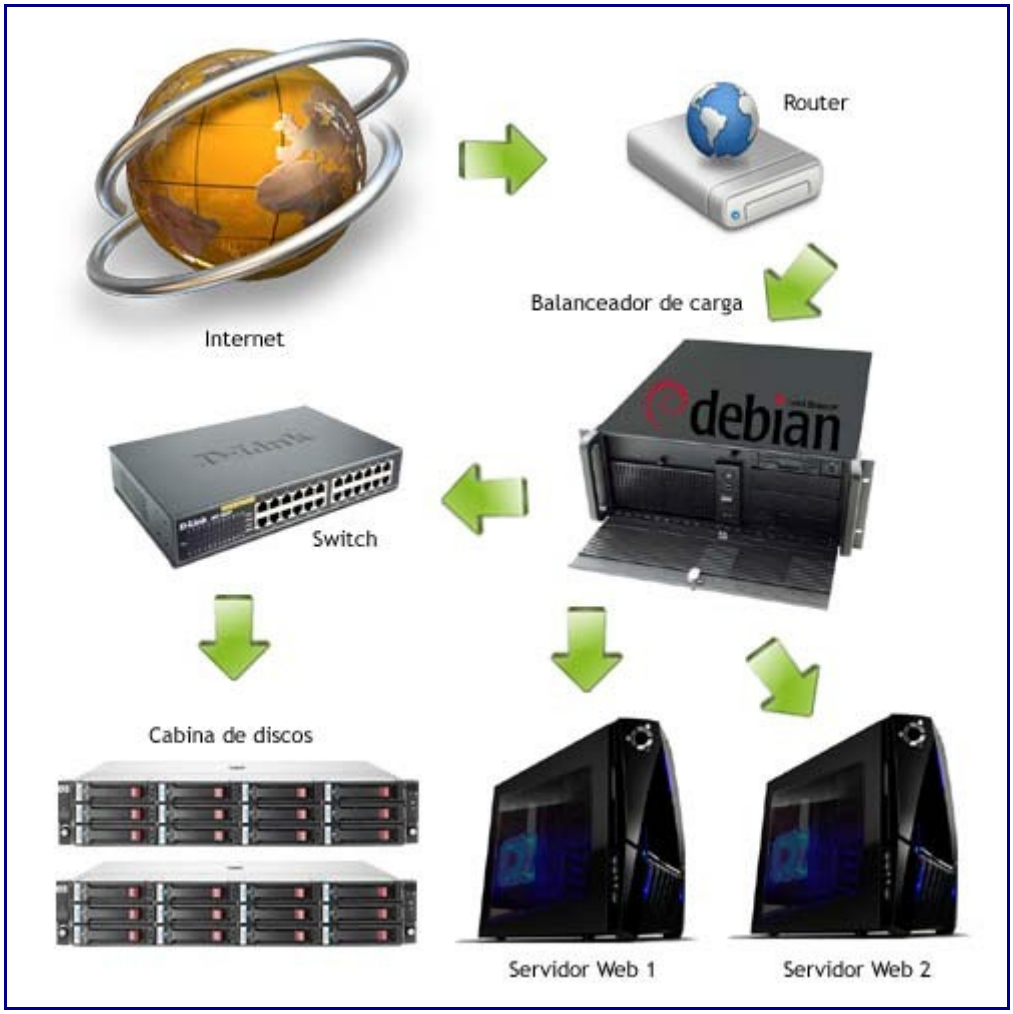
El método 2 es con el que mas repartiríamos la carga y como mas relajadas irían ya que si comienza el **balanceo de carga** en el servidor 1 y reparte las peticiones hasta el 14, cuando vuelvan a llegar al 1 la maquina ya le dio tiempo de sobras a recuperarse de memoria y procesador. Recordemos que se pueden incluir hasta 15 servidores colgando de nuestro balanceador de carga según **PEN Load Balancer** y si solo tenemos 3 tarjetas de red como si tenemos 1 o 2 siempre podemos conectarle un buen Switch y ampliar las conexiones de red.

Método 3



El método 3 puede ser el que mas se adapte a cualquiera que quiera tener un balanceador en casa. Como disponemos de tres tarjetas de red podemos dejar el **balanceador de carga** y dos **servidores** conectados y por otro lado conectar un **Switch** del que cuelguen uno o los ordenadores de escritorio que tengamos, nuestro **punto de acceso wifi** y el de nuestro **teléfono móvil** y uno o los ordenadores portátiles que tengamos. Recordemos que antes de **instalar PEN Load Balancer** montamos un **puente de red** o **Brige** que nos conecta a la red de forma transparente con lo que podemos usar este aparato para que cumpla su cometido y por otro lado puede darnos acceso a la red cuando lo necesitemos.

Método 4



El método 4 es el mas pro de todos si hablamos de alta disponibilidad y rendimiento. El balanceador reparte la carga hacia dos servidores y los dos acceden a la cabina de discos que está montada por red. Esta comparte todo lo referente al entorno web entre los dos servidores por lo que nos evitamos así tener que replicar o mas bien montar un cluster entre los dos servidores.

Para que fuera del todo correcto yo incluiría en la configuración dos tarjetas de red en cada uno de los servidores y dos swicht, uno entre el balanceador de carga y otro entre los dos servidores y la cabina de discos. Esto aseguraría aun mas la disponibilidad de los datos y repartiría bien la carga entre dos modestos servidores.

Si usamos una cabina de discos o simplemente un ordenador que cumpla con esa función recordemos que la velocidad de la red es esencial ya que una vez entremos en producción necesitamos tener un trafico lo mas fluido posible para tener una buena velocidad de acceso a los datos que queremos servir.

Posibles usos puede tener lo que podamos imaginar tan solo necesitamos **Hardware**, **Software Libre** y un poco de imaginación, ahora os toca a vosotros, hasta la próxima

---
