

Granja web creada mediante Raspberys Pi 2

Introducción

En este trabajo se va a tratar la creación y configuración de una granja web mediante tres Raspberry Pi 2. Dos de ellas se usarán como servidores instalándole LAMP y replicando los datos de forma automática. La tercera se usará como balanceador de carga aprovechando un router que lo usaremos como switch para enrutar el tráfico.

¿Por qué es una buena opción usar Raspberrys Pi para montar una granja web?

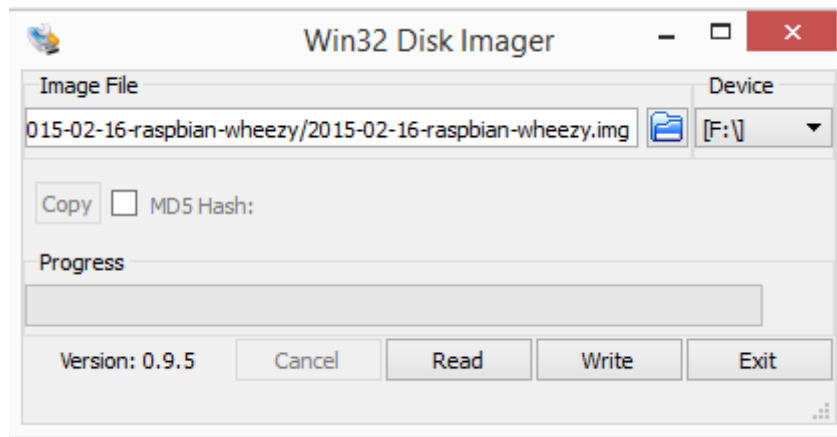
La respuesta es bastante simple, por su consumo. Estamos en una época donde uno de los principales costes a los que debe hacer frente una empresa es el premio de la energía. España es uno de los países europeos en los que hay un coste por €/kWh de toda Europa. Esto si lo unimos al sector de la computación donde para afrontar miles de visitas diarias se requieren una considerable cantidad de servidores podemos decir que una granja web o cluster con varias Raspberrys Pi puede ser una alternativa para una pequeña o mediana empresa. Cada Raspberry puede ser alimentada con un cargador de un teléfono móvil y requiere tan solo un voltaje de 5V.

Pese a su ínfimo consumo, estos pequeños dispositivos poseen un rendimiento bastante decente. Montan en su interior un procesador con arquitectura ARM de 4 núcleos donde cada número funciona a 900Mhz. Posee 1Gb de memoria RAM y en cuanto a su gráfica es capaz de trabajar con contenido audiovisual en resolución 1080p.

Configuración inicial

Lo primero que debemos realizar es bajar el sistema operativo Raspbian para nuestras Raspberrys. Ese SO es una modificación de Debian la cual esta optimizada para nuestros dispositivos. Se puede obtener desde su página oficial (<http://www.raspbian.org>) o desde la página de la organización de Raspberry.

El siguiente paso es instalar Raspbian en las Raspberrys. Es algo extremadamente sencillo. Desde Windows descargamos el programa Win32DiskImager y seleccionamos la imagen del SO y la letra del dispositivo donde está insertada la tarjeta SD. La siguiente imagen es un claro ejemplo de cómo se realiza la selección de dichos parámetros.



Tan solo nos queda darle a “Write” para que el SO esté debidamente instalado en nuestra tarjeta SD.

Para trabajar con comodidad crearemos un usuario root el cual no está activado por defecto. Raspbian viene un con usuario definido llamado pi. Su contraseña es raspberry. Accedemos con esos datos mediante ssh.

Como en cualquier versión de Debian o derivado el usuario root se activa mediante:

```
Sudo passwd
```

Le indicamos la contraseña y ya tendremos activado el administrador.

Configurar IP Estática en Raspberry

Para siempre disponer de la misma IP en cada dispositivo configuraremos dicha IP de forma manual y no la dejaremos a expensas de DHCP.

Editamos el fichero `/etc/network/interfaces`. La máscara de red y la puerta de enlace estarán correctamente si nuestro switch está por defecto en la dirección 192.168.1.1. Si no es nuestro caso la deberemos editar por la IP que hayamos establecido.

En address indicamos la IP que queremos asignarle. Obviamente debe ser una IP de las cuales son posibles asignar en la red local y no una cualquiera. Introducimos el valor y guardamos el fichero.

```
GNU nano 2.2.6      File: /etc/network/interfaces

auto lo

iface lo inet loopback
iface eth0 inet static
address 192.168.1.142
netmask 255.255.255.0
gateway 192.168.1.1

allow-hotplug wlan0
iface wlan0 inet manual
wpa-roam /etc/wpa_supplicant/wpa_supplicant.conf
```

Configuración LAMP

Vamos a instalar LAMP para que nuestros dispositivos sean totalmente funcionales como servidores. Llamamos LAMP al pack Linux Apache, MySQL y PHP. Por tanto con esto se posee lo esencial para hacer funcionar un servidor web.

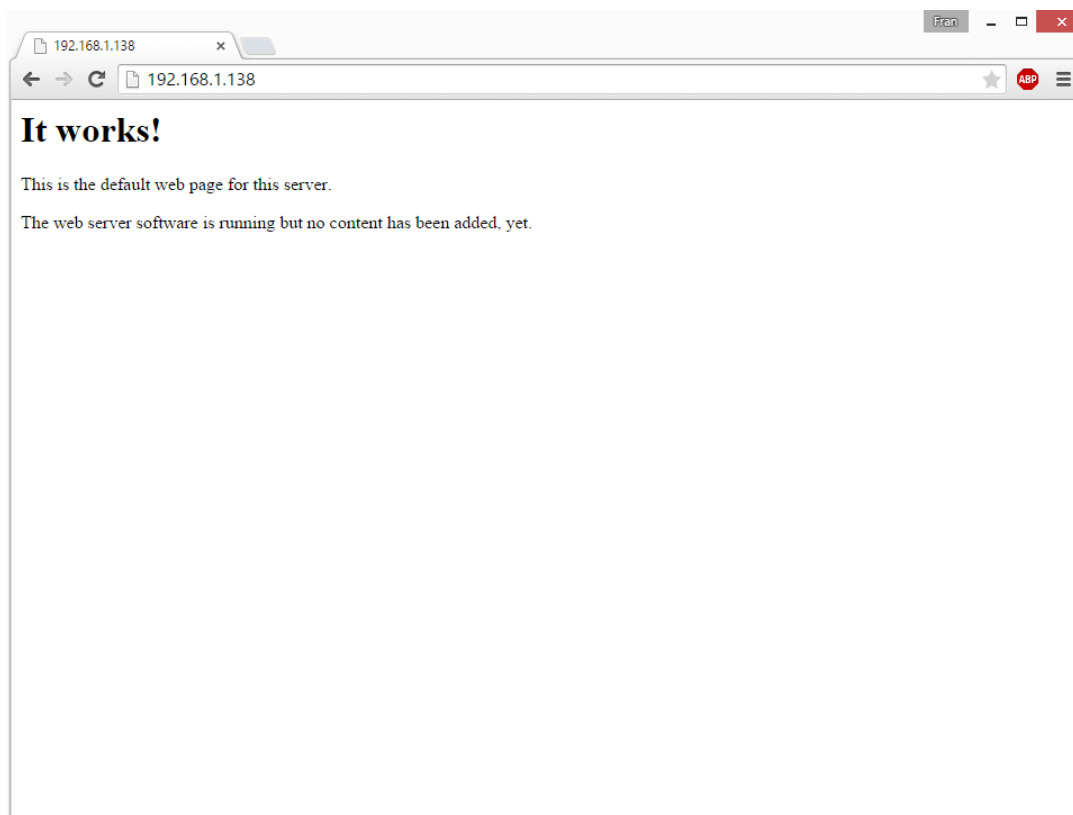
La instalación es bastante sencilla y se realiza exactamente igual que en cualquier otro sistema con Linux.

Actualizamos los repositorios con `sudo apt-get update`.

Realizado esto nos dispondremos a instalar Apache. Este es un servidor web de código abierto el cual implementa el protocolo HTTP/1.1. Destaca en su sencillez, en que es modular y extensible. Lo instalamos mediante:

```
sudo apt-get install apache2
```

Activamos el servicio con `sudo service apache2 start` y accedemos desde otro ordenador a la IP de la Raspberry para comprobar que funciona correctamente. Debe aparecernos una página en blanco donde nos diga que el servidor web está funcionando. Nos mostrará una imagen similar a esta:



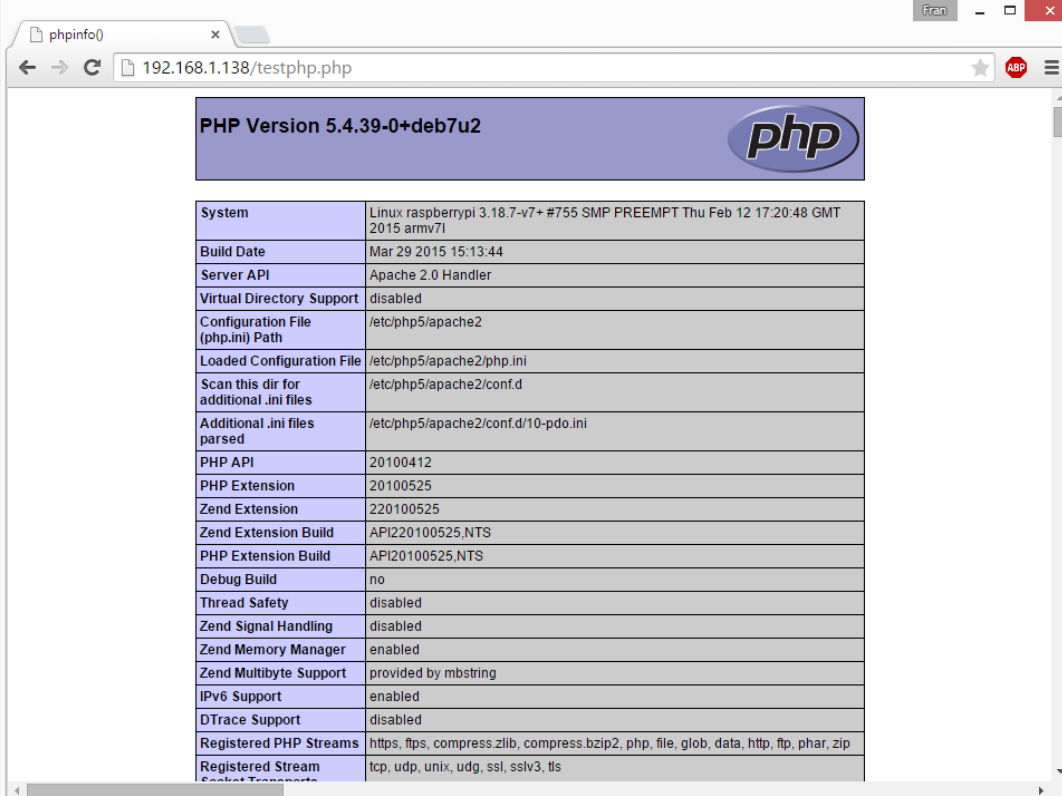
La localización de las páginas webs están ubicadas en la ruta `/var/www`.

El siguiente paso es instalar PHP. Instalaremos php5 y la biblioteca para Apache. Debemos introducir lo siguiente:

```
sudo apt-get install php5 libapache2-mod-php5
```

Para comprobar el correcto funcionamiento de php generamos un fichero php de prueba. Creamos el fichero mediante `sudo nano /var/www/testphp.php` e insertamos en él, el siguiente código php: `<?php phpinfo(); ?>`

Accedemos desde el navegador a <http://ipraspberry/testphp.php> y nos mostrará una pantalla similar a la siguiente:



System	Linux raspberrypi 3.18.7-v7+ #755 SMP PREEMPT Thu Feb 12 17:20:48 GMT 2015 armv7l
Build Date	Mar 29 2015 15:13:44
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php5/apache2
Loaded Configuration File	/etc/php5/apache2/php.ini
Scan this dir for additional .ini files	/etc/php5/apache2/conf.d
Additional .ini files parsed	/etc/php5/apache2/conf.d/10-pdo.ini
PHP API	20100412
PHP Extension	20100525
Zend Extension	220100525
Zend Extension Build	API220100525.NTS
PHP Extension Build	API20100525.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	https, ftps, compress.zlib, compress.bzip2, php, file, glob, data, http, ftp, phar, zip
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, sslv3, tls

Nos muestra la información sobre la versión de php y su configuración en la máquina instalada. Con esto pasamos al siguiente punto, MySQL.

MySQL es un sistema de gestión de bases de datos, esencial para las webs dinámicas actuales que requieren almacenar información personalizada de cada visitante. La instalación tanto del cliente como el servidor MySQL se hace de la siguiente forma:

```
sudo apt-get install mysql-server mysql-client php5-mysql
```

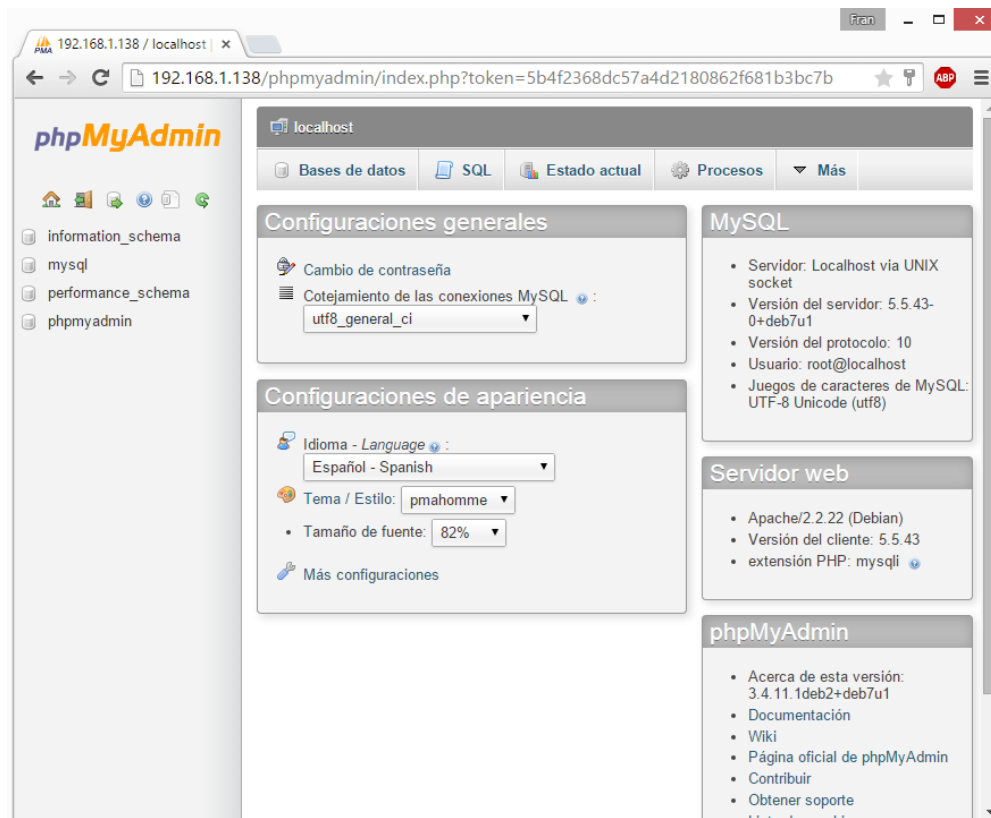
El acceso a MySQL se realiza mediante `mysql -u root -p` donde nos pedirá la contraseña que hayamos introducido en la instalación.

Ya tenemos instalado LAMP aunque instalaremos dos aplicaciones adicionales que nos pueden ser de mucha utilidad, PhpMyAdmin y VSFTP.

PhpMyAdmin es una herramienta que nos permitirá administrar las bases de datos de una forma fácil y sencilla mediante una interfaz web. El comando a introducir para su instalación es el siguiente:

```
apt-get install phpmyadmin
```

Para acceder a PhpMyAdmin introducimos <http://ipservidor/phpmyadmin> e introducimos el usuario y contraseña. Hecho esto debe aparecernos una interfaz similar a esta:



VSFTP también es una aplicación muy importante. Nos permitirá subir los ficheros webs a nuestro servidor de forma remota si necesidad de ssh. Lo instalamos con:

```
sudo apt-get install vsftpd
```

Accederemos al fichero `sudo nano /etc/vsftpd.conf` y desactivaremos el acceso anónimo y descomentaremos el acceso local y la escritura.

Reiniciamos el servicio y ya podremos subir ficheros por FTP a nuestras raspberrys.

Todos estos pasos hay que realizarlos en las 2 raspberrys que pretendemos usar como servidores web.

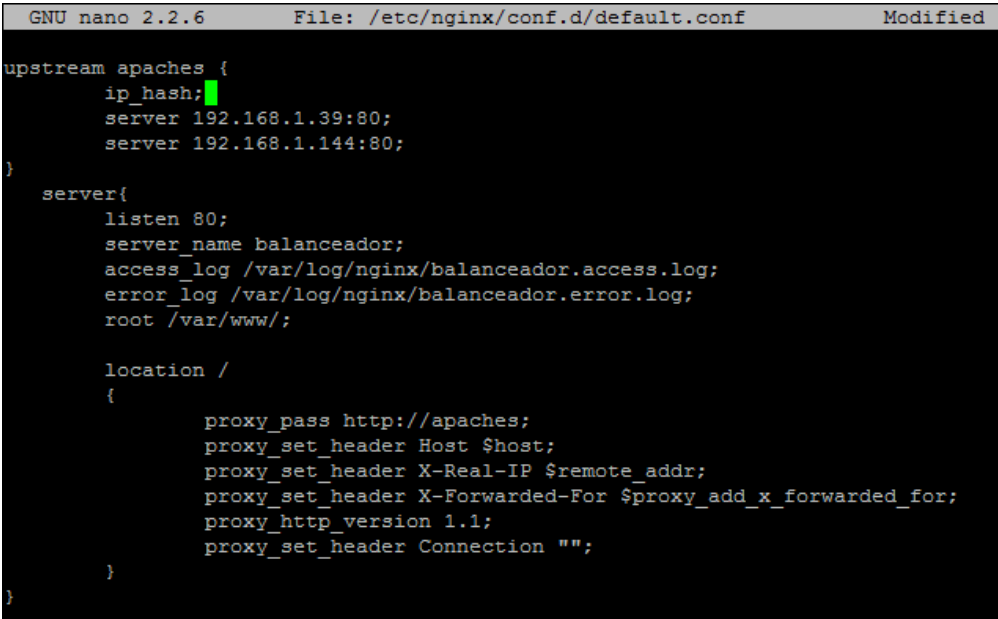
Configuración del balanceador de carga

En la tercera raspberry instalaremos el balanceador de carga. Usaremos Nginx. La instalación es bastante sencilla. Instalamos el paquete Nginx mediante la orden:

```
apt-get install nginx
```

Realizado esto vamos al fichero de configuración ubicado en `/etc/nginx/conf.d/default.conf`. En upstream apaches introducimos las IPs de las raspberrys con Apache y su puerto. Como queremos una configuración round-robin no le indicamos ningún peso a ninguna de las máquinas.

Otro factor a tener en cuenta es el tema de las sesiones. Pretendemos instalar un CMS (foro, blog...) lo que requiere autenticación. Cuando estemos rotando entre los dos servidores nuestra sesión no se guardará y no podremos por tanto iniciar sesión. Esto lo podemos solucionar de dos formas, una conectando ambos servidores a una única base de datos. Esto lo descartaremos ya que nuestra intención es que no tenga dependencia ninguna máquina de otra. La otra opción es con `ip_hash`. Esta opción de nginx asocia las IPs de los visitantes con una sola máquina destino lo que hace que podamos iniciar la sesión sin problemas en esa máquina. Ciertamente nosotros no estaremos rotando el tráfico entre las distintas raspberrys pero como suponemos que habrá diversos usuarios conectados, cada usuario estará asociado a una de las dos raspberrys.



```
GNU nano 2.2.6      File: /etc/nginx/conf.d/default.conf      Modified
upstream apaches {
    ip_hash;
    server 192.168.1.39:80;
    server 192.168.1.144:80;
}

server{
    listen 80;
    server_name balanceador;
    access_log /var/log/nginx/balanceador.access.log;
    error_log /var/log/nginx/balanceador.error.log;
    root /var/www/;

    location /
    {
        proxy_pass http://apaches;
        proxy_set_header Host $host;
        proxy_set_header X-Real-IP $remote_addr;
        proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
        proxy_http_version 1.1;
        proxy_set_header Connection "";
    }
}
```

Configuración switch

Como switch entre nuestras 3 raspberrys se ha usado un router de los que proporcionan los proveedores de ADSL. Dicho router es un ZTE h108n.

Accedemos a la administración del router (cuya IP es 192.161.1.1). El usuario y la contraseña en este router y en la mayoría de los casos es: admin como usuario y admin como contraseña.

Debemos buscar en el router NAT o mapeo de puertos para permitir la entrada y salida de peticiones del balanceador en el puerto 80.

Añadimos una nueva regla e indicamos los siguientes parámetros:

- Puerto de inicio y fin externo: 80.
- Protocolo: TCP.
- Host interno: La IP local del balanceador.
- Puerto Interno: 80.
- Nombre de la regla: algo que lo identifique como por ejemplo nginx.

Nos quedará como en la siguiente imagen:

Settings	
Type:	<input checked="" type="radio"/> Customization
Interface:	INTERNET_TR069_R_8_35 ▾
Protocol:	TCP ▾
Remote host:	<input type="text"/>
External start port:	80
External end port:	80
Internal host:	192.168.1.142
Internal port:	80
Mapping name:	Nginx

Guardamos los cambios y reiniciamos el router. Si todo ha funcionado correctamente si introducimos la dirección IP (externa) de nuestra conexión a internet desde nuestro navegador debe llevarnos al balanceador de carga y este a una de las raspberry con servidor web.

Posteriormente en otro apartado asignaremos un dominio a la IP de nuestra conexión a internet para acceder de una forma más sencilla y evitar los problemas que surgen al tener IP dinámica.

Benchmarks

Para comprobar que realizar una granja web merece la pena en términos de rendimiento hemos realizado dos tipos de benchmarks que demuestran la superioridad frente a una sola raspberry. En el fondo es lo esperado ya que con el procesamiento de más máquinas, un mayor rendimiento se debe conseguir.

Para la realización de los benchmarks se ha usado el script php del profesor de la asignatura. El script es el siguiente:

```
<?php
$tiempo_inicio = microtime(true);
```

```

for ($i=0; $i<300000; $i++){
    $a = $i * $i;
    $b = $a - $i;
    $c = $a / $b;
    $d = 1 / $c;
}
$tiempo_fin = microtime(true);
echo "Tiempo empleado: " . round($tiempo_fin - $tiempo_inicio, 4) ;
?>

```

El primer benchmark realizado es el Apache benchmark. En este benchmark se realizan 2000 peticiones en concurrencias de 20. Por tanto el comando de ejecución será similar al siguiente:

Ab -n 2000 -c 20 <http://192.168.1.142/script.php>

Lo debemos aplicar en la raspberry con el balanceador de carga y en una de las que actúan como servidor. De las mediciones nos interesan cuatro resultados: time taken for tests, failed requests, requests per second y time per request.

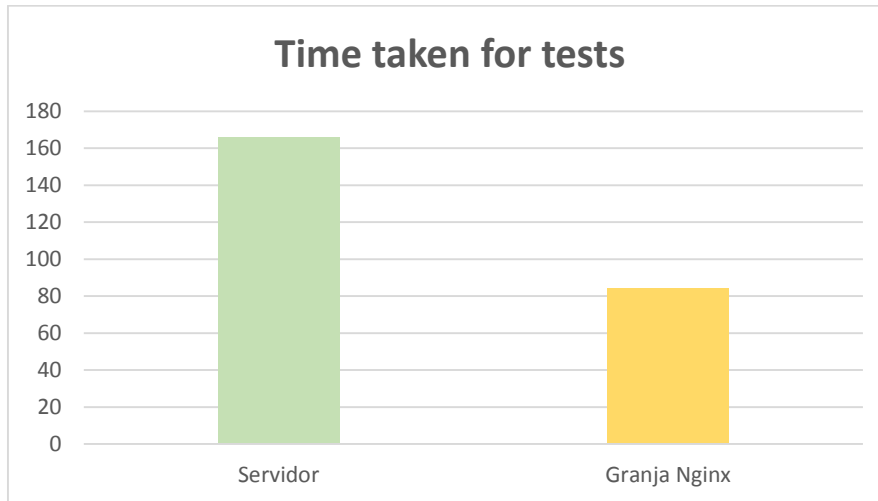
Realizamos diversas mediciones y las introducimos en una tabla para calcular la media y la desviación. Los resultados que hemos obtenido son los siguientes:

Apache Benchmark		1	2	3	4	5
Servidor	Time taken for tests	165,748537	165,727623	166,623384	165,843523	165,709536
	Failed requests	202	191	203	200	205
	Requests per second	12,07	12,07	12	12,06	12,07
	Time per request (ms)	82,874	82,864	83,312	82,922	82,855
Granja Nginx	Time taken for tests	83,32962	87,974251	82,953204	82,952484	82,874283
	Failed requests	223	198	263	214	190
	Requests per second	24,09	22,73	24,11	24,11	24,13
	Time per request (ms)	41,516	43,987	41,447	41,476	41,437

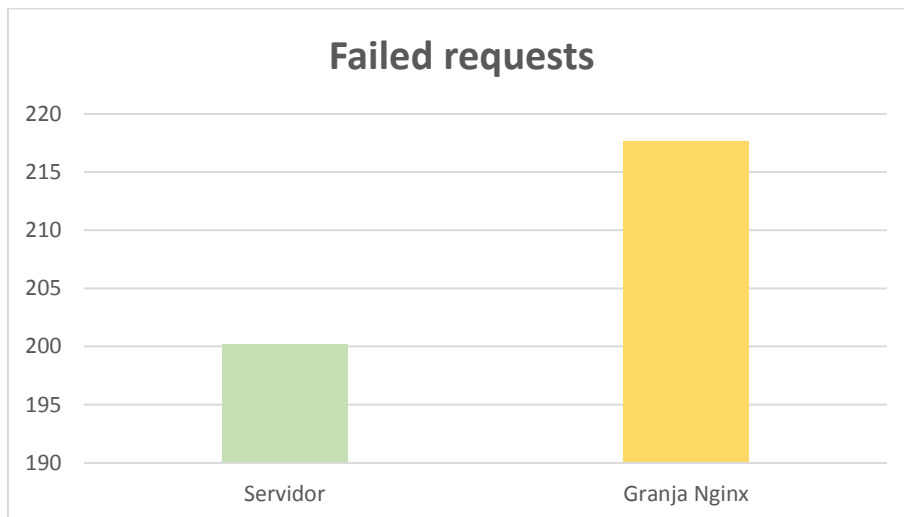
		Media	Desviación
Servidor	Time taken for tests	165,930521	0,39075182
	Failed requests	200,2	5,44977064
	Requests per second	12,054	0,0304959
	Time per request (ms)	82,9654	0,19547327
Granja Nginx	Time taken for tests	84,0167684	2,21940222
	Failed requests	217,6	28,5008772
	Requests per second	23,834	0,61731677

Time per request (ms)	41,9726	1,1265018
-----------------------	---------	-----------

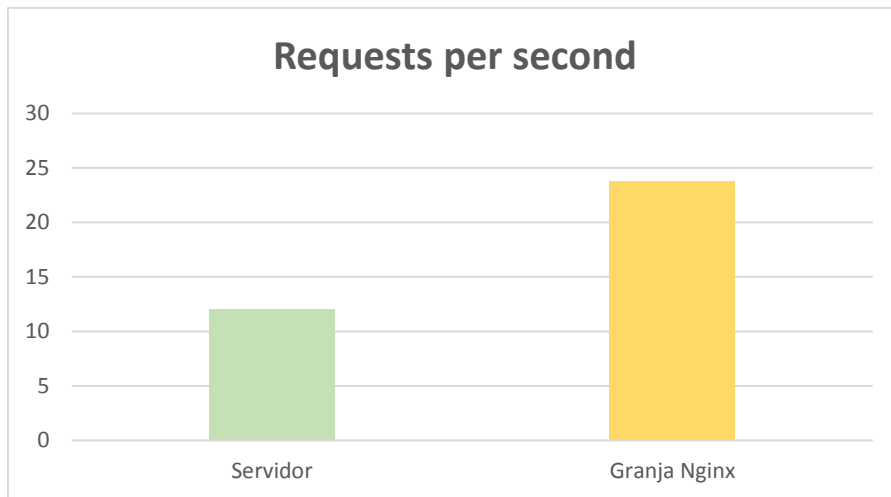
Apreciamos un mejor rendimiento con la granja como era de esperar, unos menores tiempos de respuesta y un mayor número de peticiones por unidad de tiempo. Tal vez en los resultados vistos en tablas no se aprecie la diferencia de rendimiento entre ambos métodos por lo que lo mejor es crear unas gráficas para hacer más visible la gran diferencia entre ambas formas.



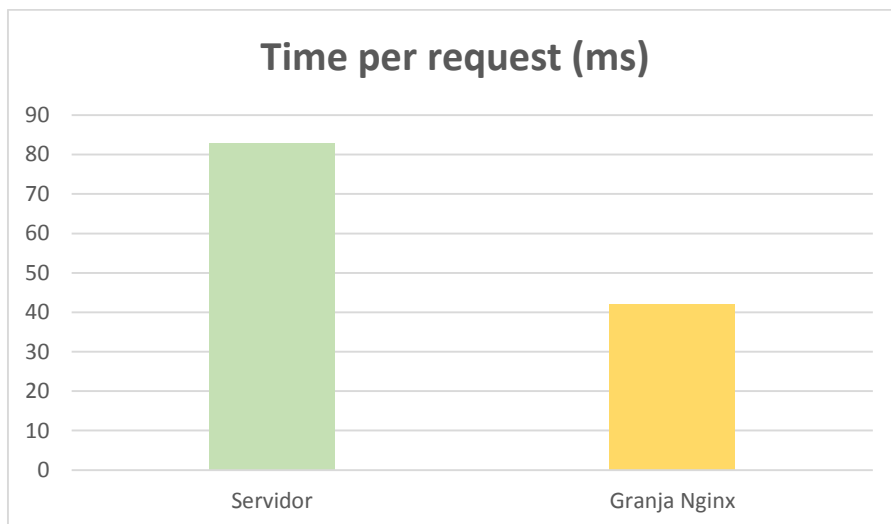
Se aprecia claramente que el tiempo necesitado para realizar los tests ha sido casi del doble en la raspberry que actúa individualmente como servidor frente a nuestro cluster con nginx.



Tal vez sea el único punto negativo de nuestra granja. Ésta ha registrado un mayor número de peticiones fallidas pero la diferencia entre ella y el servidor individual es muy poco significativa.



En las peticiones por segundo el cluster casi dobla al servidor individual como era predecible.



El tiempo por petición es prácticamente la mitad en la granja frente a una única raspberry. Por tanto al menos en los benchmarks realizados mediante Apache benchmark podemos sacar como conclusión que nuestra granja tiene un rendimiento muy superior frente a una única raspberry como servidor.

Para corroborar esto también se han realizado diversos test con otro software de benchmark, concretamente OpenWebLoad.

En este benchmark simularemos 20 peticiones simultáneas hacia la url de nuestro script en la granja y en una raspberry. Por tanto el comando a ejecutar sería similar al siguiente modificando la IP en función de la máquina destino (raspberry servidor o balanceador).

```
openload.exe 192.168.1.142/script.php 20
```

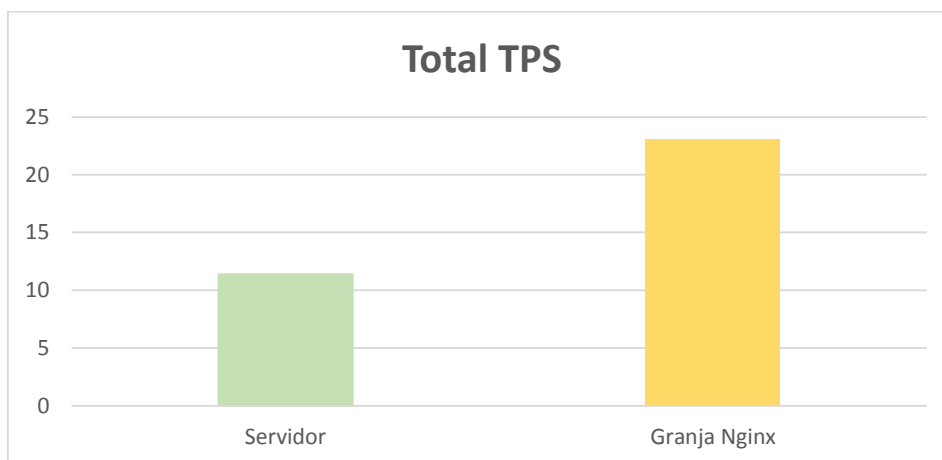
Con los resultados obtenidos seleccionamos los siguientes parámetros: Total TPS, Avg. Response time y Max Response time.

Generamos como en el caso anterior una tabla con los resultados para calcular las medias y las desviaciones. El resultado ha sido este:

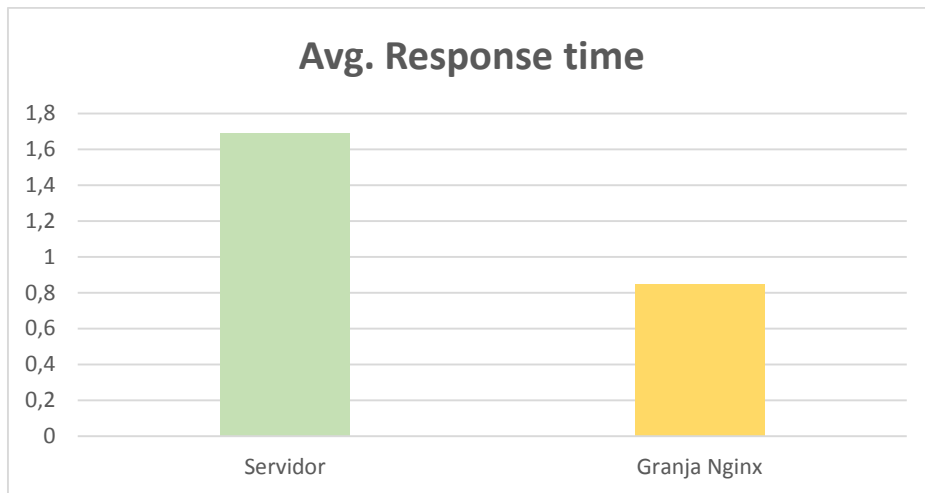
OpenWebLoad						
		1	2	3	4	5
Servidor	Total TPS	10,74	11,74	11,45	11,43	11,58
	Avg. Response time	1,798	1,66	1,664	1,664	1,662
	Max Response time	5,093	2,597	2,526	2,44	2,299
Granja Nginx	Total TPS	21,59	23,71	23,72	23,31	23,18
	Avg. Response time	0,913	0,83	0,829	0,829	0,84
	Max Response time	4,048	1,202	1,133	1,26	1,614

		Media	Desviación
Servidor	Total TPS	11,388	0,38284462
	Avg. Response time	1,6896	0,06062013
	Max Response time	2,991	1,18028916
Granja Nginx	Total TPS	23,102	0,87850441
	Avg. Response time	0,8482	0,03651986
	Max Response time	1,8514	1,24187189

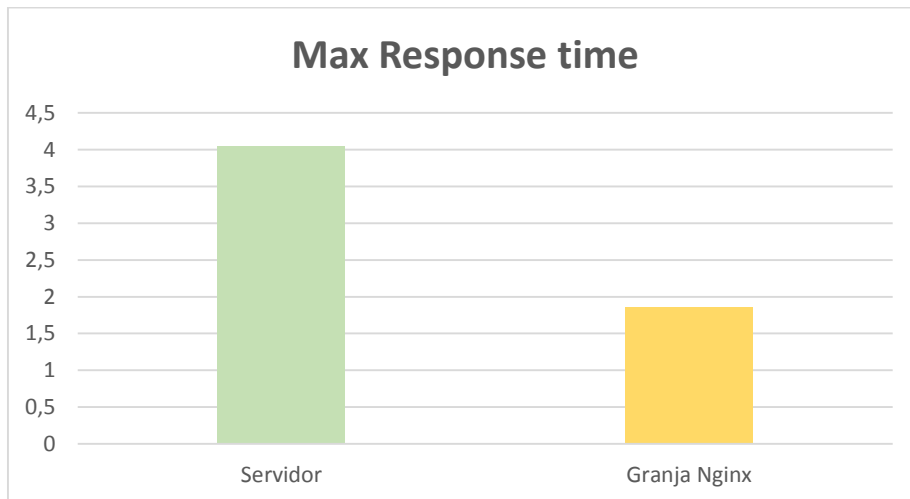
Como en el benchmark anterior la tónica es similar demostrando un mejor rendimiento la granja web. Para representar la información de una forma más visual pasamos los datos a gráficas.



El número de transacciones por segundo supera el doble en la granja web frente a una sola raspberry. Es una clara demostración de la diferencia de rendimiento.



En el tiempo medio de respuesta pasa como en el caso anterior, y la granja es el doble de rápida que un único servidor web.



Como tiempo máximo de respuesta es mejor la granja aunque no llega a ser la mitad de tiempo que una sola raspberry. De todas formas es un gran resultado.

La conclusión es bastante clara, la granja posee un rendimiento muy superior frente a usar una sola raspberry como servidor web. De todas formas lo normal es estudiar las necesidades en cada caso para así montar el servidor web de una forma u otra.

Réplica de datos entre Raspberrys

Debemos replicar los datos que tenemos alojados en el directorio `/var/www/` de una raspberry a la otra.

Para ello en una de las máquinas generamos una clave pública y una privada. Lo hacemos con `ssh-keygen -t dsa`.

```

root@raspberrypi:/home/pi# ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/root/.ssh/id_dsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_dsa.
Your public key has been saved in /root/.ssh/id_dsa.pub.
The key fingerprint is:
01:03:13:02:ec:81:68:32:49:af:eb:d3:93:a2:6f:81 root@raspberrypi
The key's randomart image is:
+--[ DSA 1024]-----+
*+. +00
*+.. . 0
+... .
.. .
o      S
E o
....
.O.+
O++ .
+-----+

```

Con `ssh-copy-id -i .ssh/id_dsa.pub root@192.168.1.39` la copiamos en la otra máquina. De esta forma podrá conectarse sin necesitar clave.

Finalmente automatizamos la tarea de copia de ficheros del directorio `/var/www/` editando el fichero `/etc/crontab` y añadiendo:

```

5 * * * * root rsync -avz --delete --exclude=**/stats --exclude=**/error --
exclude=**/files/pictures -e "ssh -l root" root@192.168.1.39:/var/www/ /var/www/

```

De esta forma se replicarán los ficheros en intervalos de 5 minutos.

```

root@raspberrypi:/home/pi# cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 * * * * root    cd / && run-parts --report /etc/cron.hourly
25 6 * * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
t /etc/cron.daily )
47 6 * * 7 root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
t /etc/cron.weekly )
52 6 1 * * root    test -x /usr/sbin/anacron || ( cd / && run-parts --report
t /etc/cron.monthly )
5 * * * * root rsync -avz --delete --exclude=**/stats --exclude=**/error --excl
de=**/files/pictures -e "ssh -l root" root@192.168.1.39:/var/www/ /var/www/
#
root@raspberrypi:/home/pi#

```

MySQL en modo master-master

Al igual que replicamos los datos de una raspberry en la otra debemos hacer lo mismo con las bases de datos. Como nuestra intención es montar un foro y distribuir la carga mediante el balanceador deberemos replicar la base de datos en ambos sentidos.

Editamos el fichero `/etc/mysql/my.cnf` de las 2 raspberrys.

En una de ellas lo modificamos de la siguiente forma:

```
server-id = 1
```

```
log_bin = /var/log/mysql/mysql-bin.log
```

```
binlog_do_db = foro
```

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

Y en la segunda sería igual salvo server-id=2.

Accedemos a mysql desde la primera máquina con `mysql -u root -p`. Le damos permisos a root para que pueda replicar datos.

```
grant replication slave on *.* to root@'192.168.1.39' identified by 'raspberry';
```

Hacemos exactamente igual en la segunda máquina. A dicha máquina le pasamos la IP maestra el usuario y la posición del fichero de base de datos. Sería así:

```
CHANGE MASTER TO MASTER_HOST = '192.168.1.39', MASTER_USER = 'root',  
MASTER_PASSWORD = 'raspberry', MASTER_LOG_FILE = 'mysql-bin.000001',  
MASTER_LOG_POS = 4546;
```

En la primera haríamos lo mismo pero con:

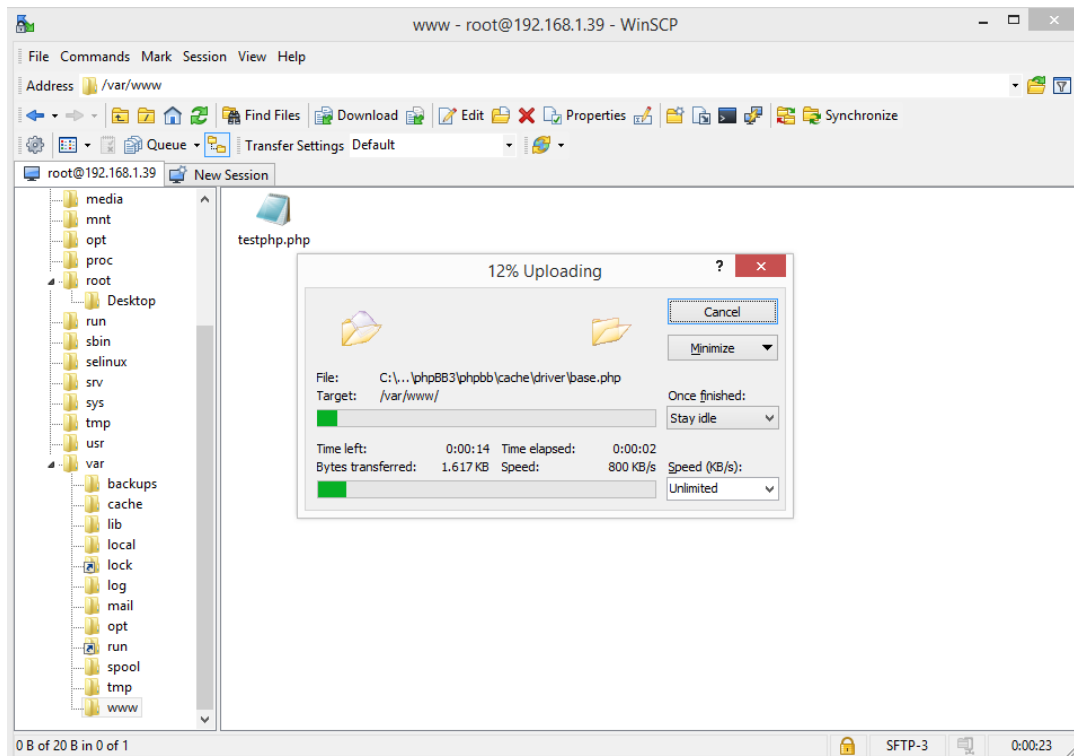
```
CHANGE MASTER TO MASTER_HOST = '192.168.1.144', MASTER_USER = 'root',  
MASTER_PASSWORD = 'raspberry', MASTER_LOG_FILE = 'mysql-bin.000001',  
MASTER_LOG_POS = 226;
```

Recordar que para hacer esto hay que parar primero los esclavos.

Creación y configuración del foro PHPBB3

Descargamos la última versión de phpBB3 de su web oficial. Descomprimos el fichero y nos aparecerán varios directorios y ficheros php. Estos deberemos subirlos a las dos raspberrys que actúan como servidor web.

Subimos los ficheros en las raspberrys que poseen apache en la ruta `/var/www/`. Podemos hacerlo mediante FTP o con scp. En nuestro caso lo hemos realizado por scp con la aplicación WinSCP.



Accedemos a las IPs locales de cada máquina y nos llevarán a la página de instalación de phpBB3.

Procedemos a instalar y nos llevará a la configuración de la base de datos. Le introducimos la IP del servidor de bases de datos que al ser el mismo que el web es localhost. El nombre de la base de datos que tenemos creada a través de phpmyadmin y nuestro usuario y contraseña.

Configuración de base de datos

Tipo de base de datos:	MySQL
Nombre del servidor de la base de datos o DSN: <small>DSN significa Data Source Name y es relevante solo para instalaciones ODBC. En PostgreSQL, use localhost para conectar con el servidor local a través del dominio del socket UNIX y 127.0.0.1 para conectar via TCP. En SQLite, introduzca la ruta completa a su base de datos.</small>	localhost
Puerto en el servidor de la base de datos: <small>Déjelo en blanco a menos que quiera que el servidor opere en un puerto distinto al estándar.</small>	
Nombre de base de datos:	foro
Usuario de base de datos:	root
Clave de base de datos:
Prefijo para tablas en la base de datos: <small>El prefijo debe comenzar con una letra y sólo debe contener letras, números y subrayados.</small>	p1

Proceder al siguiente paso

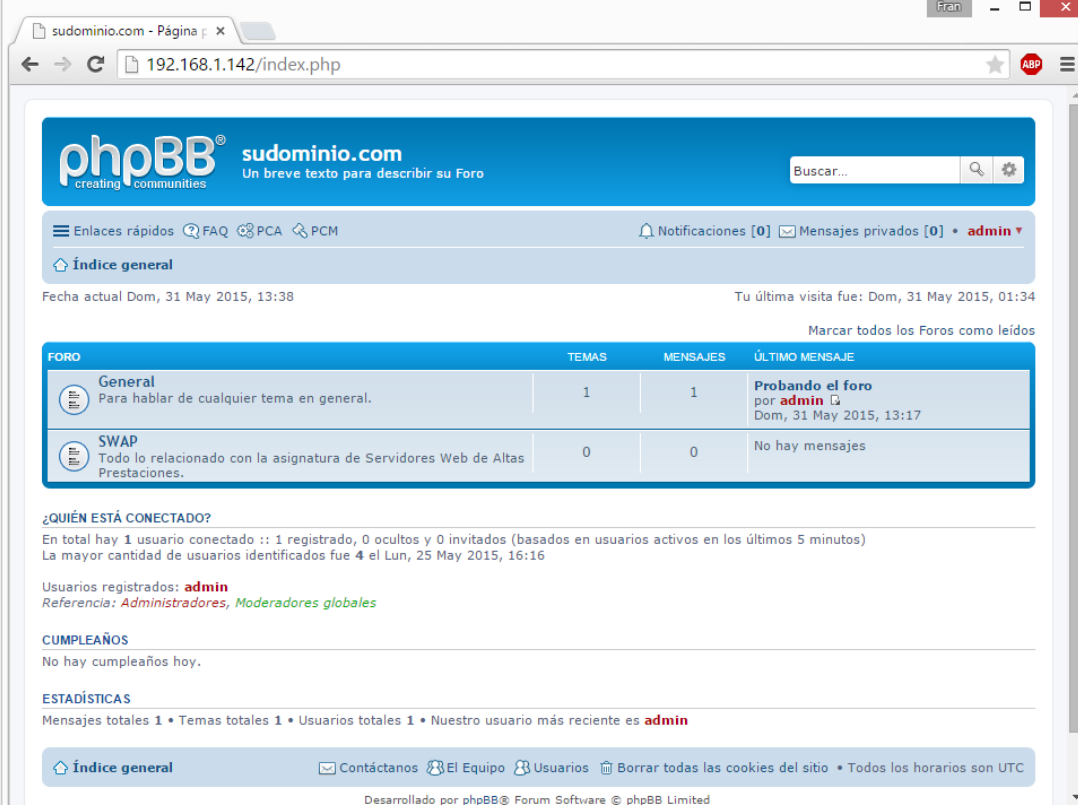
Continuamos y nos pedirá el nombre, email y contraseña que deseamos que tenga el administrador.

Lo siguiente está relacionado con el tema de url, protocolos https y redireccionamientos, envíos de email smtp... Como realmente no es lo que nos interesa en este momento lo dejamos por defecto.

Se subirán las tablas de la base de datos y tendremos el foro en funcionamiento.

La configuración debe ser exactamente igual en los dos servidores para no tener problemas con la base de datos en sus replicaciones.

Entramos en la IP del balanceador y ya veremos nuestro foro.



phpBB® sudominio.com
Un breve texto para describir su Foro

Enlaces rápidos: FAQ, PCA, PCM

Notificaciones [0] Mensajes privados [0] • admin

Fecha actual Dom, 31 May 2015, 13:38 Tu última visita fue: Dom, 31 May 2015, 01:34

Marcar todos los Foros como leídos

FORO	TEMAS	MENSAJES	ÚLTIMO MENSAJE
General Para hablar de cualquier tema en general.	1	1	Probando el foro por admin Dom, 31 May 2015, 13:17
SWAP Todo lo relacionado con la asignatura de Servidores Web de Altas Prestaciones.	0	0	No hay mensajes

¿QUIÉN ESTÁ CONECTADO?
En total hay 1 usuario conectado :: 1 registrado, 0 ocultos y 0 invitados (basados en usuarios activos en los últimos 5 minutos)
La mayor cantidad de usuarios identificados fue 4 el Lun, 25 May 2015, 16:16

Usuarios registrados: **admin**
Referencia: *Administradores, Moderadores globales*

CUMPLEAÑOS
No hay cumpleaños hoy.

ESTADÍSTICAS
Mensajes totales 1 • Temas totales 1 • Usuarios totales 1 • Nuestro usuario más reciente es **admin**

Índice general Contáctanos El Equipo Usuarios Borrar todas las cookies del sitio • Todos los horarios son UTC

Desarrollado por phpBB® Forum Software © phpBB Limited







DNS Dinámico

Para siempre poder acceder de forma sencilla a nuestra granja web debemos asignarle un dominio a la IP en la que esté trabajando nuestra granja. Como normalmente en una conexión ADSL convencional nuestra IP es dinámica nos surge un gran problema, ¿cómo saber en todo momento nuestra IP y cómo asignarla automáticamente a un dominio?

Lo primero que debemos realizar es crearnos una cuenta gratuita en alguna empresa que ofrezca servicios de DNS dinámicos. Usaremos NoIP por lo que nos registraremos en <http://www.noip.com/>.

En la web accedemos a la sección de Hosts / Redirects y pinchamos en “Add a Host”. Seleccionamos un nombre de dominio gratuito. Nosotros hemos elegido swap.noip.me. Como Host type marcamos DNS Host (A).

La dirección IP no es necesaria introducirla ya que un demonio en nuestro balanceador lo hará de forma automática. Guardamos cambios y ya tenemos nuestro dominio listo.

Hostname Information	
Hostname:	swap.noip.me 
Host Type:	<input checked="" type="radio"/> DNS Host (A) <input type="radio"/> DNS Host (Round Robin) <input type="radio"/> DNS Alias (CNAME) 
	<input type="radio"/> Port 80 Redirect <input type="radio"/> Web Redirect
IP Address:	<input type="text" value="87.219.57.30"/> Last Update: 2015-05-29 17:32:00 PDT 
Assign to Group:	<input type="text" value="- No Group -"/>  Configure Groups
Enable Wildcard:	Wildcards are a Plus / Enhanced feature. Upgrade Now! 
Advanced Records:	TXT, SPF, and SRV records and the use of some special clients are Plus / Enhanced features. Upgrade now to use them. 

Ahora vamos a instalar en el balanceador el demonio que acabamos de mencionar.

Creamos el directorio /home/pi/noip y nos ubicamos en él. Ahí guardaremos el fichero necesario que obtendremos con:

```
wget http://www.no-ip.com/client/linux/noip-duc-linux.tar.gz
```

Lo descomprimos y accedemos al directorio. En el hacemos un make y cuando esté listo un make install.

Nos pedirá que introduzcamos nuestro usuario y contraseña de noip y que seleccionemos el dominio a usar. También el intervalo de actualización.

Lanzamos el servicio con sudo /usr/local/bin/noip2.

Para que se ejecute cada vez que iniciemos el sistema modificamos el fichero /etc/rc.local y le introducimos /usr/local/bin/noip2.

Guardamos y ya lo tenemos listo para cuando cambie nuestra IP o reiniciemos el balanceador.

Referencias

<http://www.geothread.net/building-a-lamp-server-on-the-raspberry-pi-computer/>

<http://www.awesomeweirdness.com/projects-diy/raspberrypi/setup-noip-client-raspberry-pi/>

<http://www.phpbb-es.com/foro/viewtopic.php?t=39887>

<http://www.modmypi.com/blog/tutorial-how-to-give-your-raspberry-pi-a-static-ip-address>