

# Varnish Cache, un sistema que te ayuda a mejorar la velocidad de tu web



## Introducción

Hoy en día, las aplicaciones web requieren de decenas de consultas SQL a base de datos para conseguir la información necesaria que genere una página solicitada por un usuario. Esta cantidad de consultas suele ser un gran problema en aquellas webs que son visitadas por un alto número de usuarios, lo que suele provocar que aumente el tiempo de respuesta, y por tanto, una penalización en los principales buscadores web.

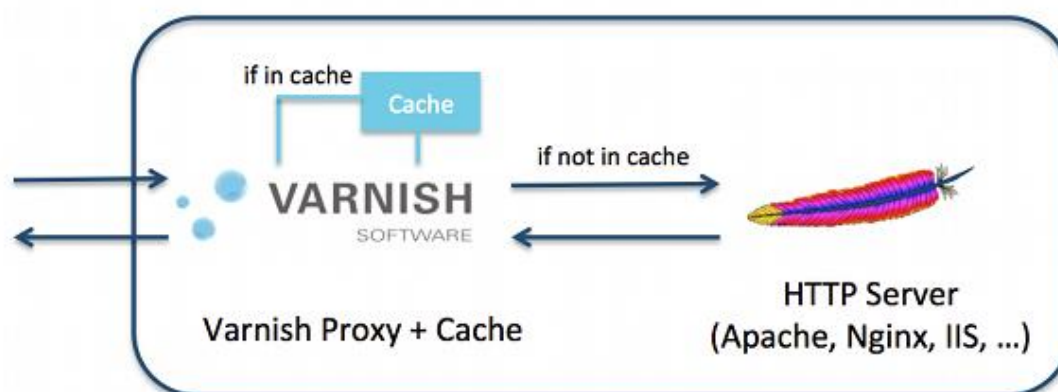
Para dar solución a este problema de velocidad de carga, podemos hacer uso de varias herramientas como puede ser el uso de Memcached o bien el uso de aceleradores HTTP como Varnish Caché, herramienta sobre la que hablaremos a lo largo de nuestro White Paper.

## Qué es Varnish Cache

**Varnish Cache** es un potente acelerador HTTP de código abierto capaz de cachear el contenido de una página web y servirlo de forma rápida y eficaz. Al servir páginas cacheadas, Varnish consigue reducir el número de llamadas a la base de datos cada vez que alguien realice una visita a nuestra web. Esto trae consigo una reducción en el tiempo de carga de nuestra web, lo que mejora la experiencia del usuario.

Varnish Cache trabaja como proxy inverso, situándose delante del servidor HTTP, cacheando la respuesta de dicho servidor web en memoria. De esta forma, la próxima vez que un visitante visite la misma url, la página será servida desde el servidor Varnish en vez de desde servidor web, ahorrando en recursos y permitiendo más conexiones simultáneas.

Mediante el uso de esta herramienta, el esquema de peticiones de una url de una página web quedaría de la siguiente manera.



El esquema anterior funcionaría de la siguiente manera. Un usuario solicita ver una determinada url. La petición de la información llega a Varnish, que comprueba si esa información la tiene en caché. De ser así, la recupera y se la muestra al visitante. Si no la tuviera en cacheada, pasaría la información al **servidor web** para que generase la respuesta. Este servidor web se la devolvería a Varnish que la almacenaría en caché a la vez que se la muestra al usuario.

Cuando Varnish recibe una solicitud de información, podrá actuar de varias formas, según haya sido configurado. Entre las opciones que tendrá están:

- Aceptar o denegar la solicitud, dependiendo de varios criterios.
- Pasar la solicitud al servidor web, para que éste le devuelva la respuesta que será mostrada al usuario.
- Cachear la información devuelta por el servidor web.

Servir la información almacenada en caché en vez de pasar la petición al servidor web.

## Principales características de Varnish Cache

Entre las principales características que nos presenta Varnish, podemos destacar:

- Se trata de una herramienta muy estable y rápida, capaz de servir miles de peticiones por segundo.
- Ofrece mucha flexibilidad a la hora de su configuración gracias a que incorpora su propio lenguaje de configuración denominado VCL (Varnish Configuration Language) con el que es posible establecer reglas para el cacheo de la información.
- Posibilidad de ampliar su funcionalidad por medio de módulos llamados VMOD's, [listado de módulos que puedes ver en esta dirección](#).
- Soporte para GZIP y ESI (Edge Side Includes). También permite la generación de respuestas a peticiones de forma dinámica mediante el uso del lenguaje de marcado ESI, lo que permite cachear fragmentos de las páginas usando tiempos de expiración diferentes, sin la necesidad de tener que borrar de caché las secciones menos cambiantes de un portal para mostrar información fresca. Por ejemplo, podríamos tener una web donde la columna de la derecha fuera siempre fija que fuera cacheada con un tiempo de expiración de 3.600 segundos, mientras que el centro podría cambiar acorde a las noticias publicadas, pudiéndole asignar un tiempo de expiración de 300 segundos.
- Consulta de logs y estadísticas en tiempo real.

## Instalando Varnish Cache en nuestro servidor

El proceso de instalación de Varnish Cache en un servidor es muy similar para cualquier distribución Linux, pero en nuestro caso explicaremos el proceso de instalarlo en un servidor que funciona con el sistema operativo Ubuntu.

Lo primero que haremos será añadir el repositorio desde donde lanzaremos la instalación. Para ello ejecutamos la siguiente instrucción en nuestro [servidor dedicado](#), [cloud](#) o [VPS](#).

```
curl http://repo.varnish-cache.org/debian/GPG-key.txt | sudo apt-key add -
```

El siguiente paso será editar el fichero "source.list" para añadir el repositorio a la lista de fuentes APT. La línea que deberemos añadir es la siguiente.

```
deb http://repo.varnish-cache.org/ubuntu/ lucid varnish-3.0.4
```

Por último actualizamos “apt-get” e instalamos Varnish.

```
apt-get update
```

```
apt-get install varnish
```

## Configuración de Varnish Cache

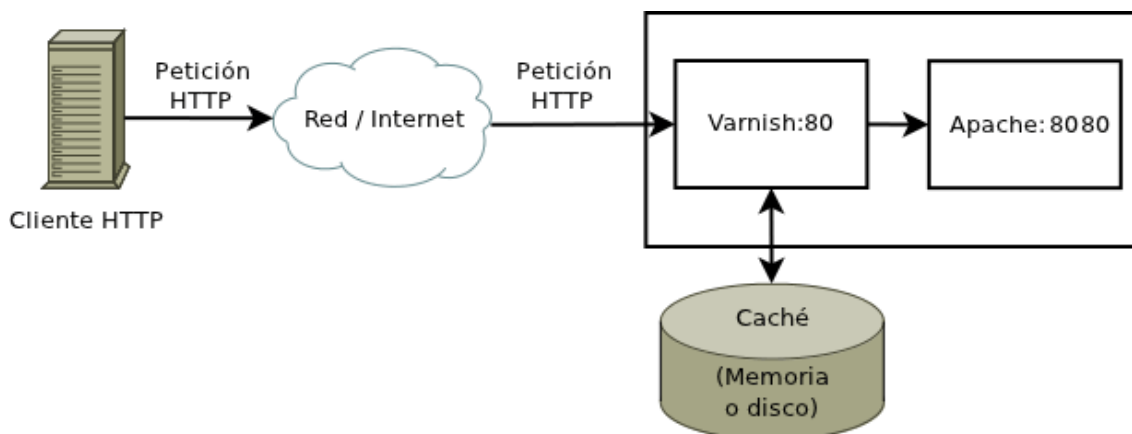
Una vez que tenemos instalada la aplicación, es necesario configurar los parámetros de arranque de Varnish. Esta configuración de arranque se lleva a cabo en la sección “DAEMON\_OPTS” que está dentro del fichero de configuración de Varnish y que podemos encontrar en la ruta “/etc/default/varnish”.

Pasamos a explicar a continuación cada uno de los parámetros que podemos utilizar y que no son de obligado uso, pudiendo utilizar únicamente aquellos que mejor se adaptan a nuestras necesidades.

- **Parámetro –a:** por medio de esta opción le podemos indicar la dirección IP y puerto por la que tiene que escuchar. Si no se indica este parámetro, escucharía por todas las IPs.
- **Parámetro –T:** indica la dirección IP y puerto por la que escuchará la interfaz de administración de Varnish Cache.
- **Parámetro –t:** permite indicar un TTL (Time To Live) por defecto.
- **Parámetro –p:** es un parámetro de uso avanzado ya que permite especificar una serie de valores como: thread\_pool\_min, thread\_pool\_max o listen\_depth entre otros.
- **Parámetro –w:** nos permite establecer separados por coma tres valores relacionados con los hilos que maneja Varnish: nº mínimo de hilos, nº máximo de hilos y timeout del hilo. Este valor tampoco será necesario tocarlo.
- **Parámetro –s:** indicamos dónde queremos que se almacene la caché y su tamaño. Lo podemos especificar de dos formas distintas:
  - **–s malloc, 512m:** con esto le indicamos que la caché se almacena en memoria y que tenga un tamaño de 512 MB.
  - **–s file, /var/lib/varnish/storage.bin, 1G:** en este caso estamos indicando que el caché se almacene en un fichero y que el tamaño sea de 1 GB.
- **Parámetro –f:** establecemos la localización del archivo de configuración de Varnish.
- **Parámetro –S:** aquí se indica la localización del archivo secreto que es utilizado para la autenticación de la consola de Varnish.

Un ejemplo de configuración podría ser el siguiente.

```
DAEMON_OPTS=" -a :80 \  
-T localhost:7007 \  
-f /etc/varnish/default.vcl \  
-S /etc/varnish/keys \  
-s malloc,512m"
```



Como hemos visto anteriormente, Varnish Cache escucha por el Puerto 80, el puerto habitual que utilizan los servidores web, por lo que será necesario configurar el servidor web para que utilice otro puerto distinto, por ejemplo el 8080.

En nuestro caso, el servidor web será Apache, por lo que editaremos el archivo de configuración de Apache con el nombre "ports.conf" que suele estar en la ruta "/etc/apache2/ports.conf". Ahí modificamos el puerto 80 por el 8080.

```
NameVirtualHost 127.0.0.1:8080
```

```
Listen 127.0.0.1:8080
```

Antes de dar por finalizada la configuración de Varnish, también será necesario modificar los ficheros de configuración de los VirtualHost que podamos tener, que suelen estar localizados en la ruta "/etc/apache2/sitios-disponibles-a-cambiar".

## Instrucciones útiles para el manejo de Varnish

Para finalizar nuestro White Paper sobre Varnish Cache, os mostraremos las instrucciones básicas para iniciar, parar, reiniciar y recargar Varnish.

- **Iniciar Varnish:** service varnish start
- **Parar Varnish:** service varnish stop
- **Reiniciar Varnish:** service varnish restart
- **Recargar Varnish:** service varnish reload

Gracias a su sencillo funcionamiento y a las muchas posibilidades que nos ofrece, Varnish se ha convertido en uno de los aceleradores HTTP más utilizado a escala mundial, sobre todo por aquellos portales web que reciben miles de visitas a lo largo del día.