

Administrar VPS sin panel de control

escrito por Andy Garcia | 07/07/2014



Conozco varios servidores de [hosting VPS con IP española](#), con 2 GB de RAM y licencia Plesk o cPanel por 20 €/mes, por ejemplo [CUBENODE](#), pero sin panel de control te puede salir por menos de la mitad...

Gracias a [GinerNet](#) y a sus VPS SSD me he propuesto **optimizar el presupuesto de Hosting** y también **optimizar el rendimiento del servidor**, he tenido que hacer muchas pruebas y re-instalar el VPS varias veces (cada reinstalación no lleva más que un par de minutos y puedes

elegir entre varias distribuciones de linux).

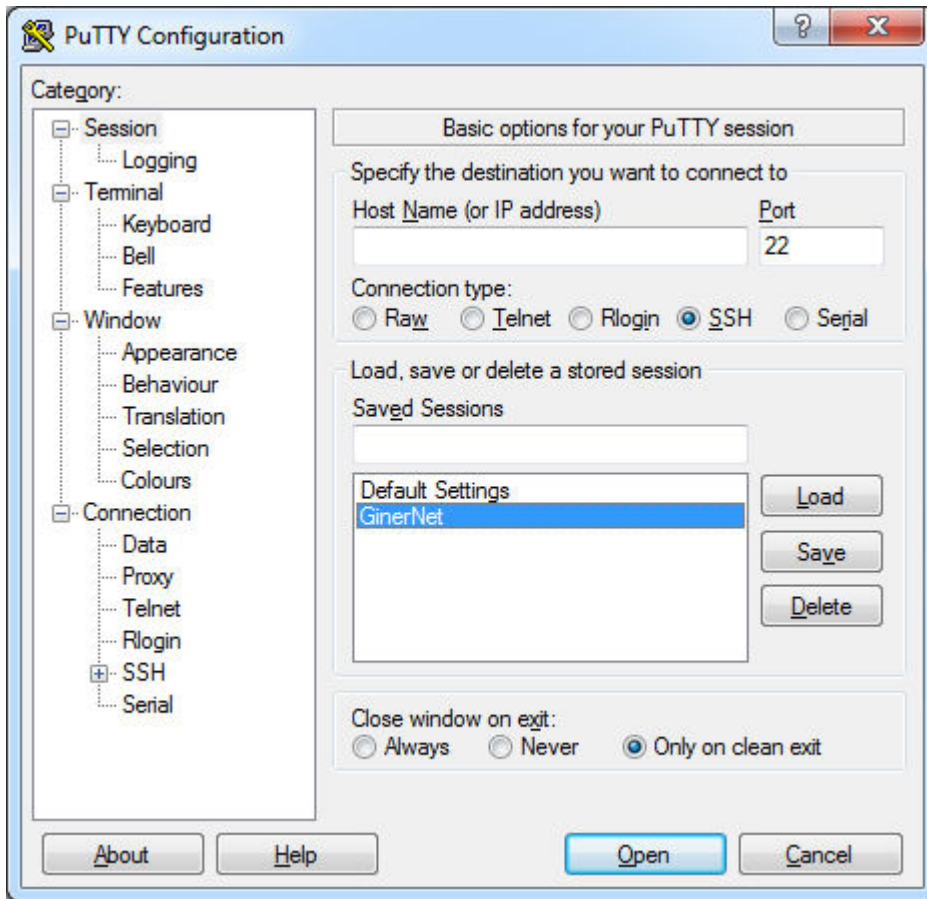
Además me propuse un reto adicional porque quería hospedar, entre otras, una web que no se lleva bien con las versiones modernas de PHP, así que como máximo debía instalar PHP 5.2.

Mi intención es instalar cuantas menos cosas mejor y tocar el servidor lo mínimo, para así **optimizar el uso de los recursos y conseguir máximo rendimiento.**

He probado varias distribuciones de linux para decantarme por CentOS 5, el único motivo para ello es que incluye PHP 5.1 y MySQL 5.0 en sus repositorios oficiales, si te decides por CentOS 6 tendrás PHP 5.3 y MySQL 5.1 en su lugar.

AVISO: la rama 5.2 de PHP no dispone de soporte desde el año 2011.

Para seguir este tutorial debes tener un mínimo de experiencia con el entorno SSH y la línea de comandos de linux.



Una vez instalado el VPS, y conectado como root vía SSH (usando [Putty](#)), los pasos a seguir para instalar Apache, PHP, MySQL y las dependencias necesarias son los siguientes:

- El primer paso es opcional y sirve para actualizar el sistema operativo y algunos de sus componentes:

```
yum update -y
```

El parámetro '-y' sirve para evitar que el sistema pida confirmación, puedes omitirlo y confirmar manualmente.

Instalar Apache

```
httpd -v  
yum install httpd -y  
service httpd start  
chkconfig httpd on  
vi /etc/httpd/conf/httpd.conf
```

```
service httpd restart
```

La primera línea comprueba si está ya instalado, si el sistema de te devuelve la versión de Apache puedes saltar directamente a la penúltima línea (del código anterior) para configurar algunas opciones.

Para editar un archivo de configuración usando vi puedes usar /palabra para buscar lo que quieres cambiar, posteriormente pulsas 'i' para entrar en el modo de edición, haces los cambios y pulsas 'escape' para salir del modo edición tecleando :wq! para salir guardando los cambios o :q! sin guardar.

Los cambios a realizar en la configuración de apache son los siguientes:

- descomentar NameVirtualHost *:80
(para poder usar varios VirtualHost y hospedar varios dominios)
- comentar AddDefaultCharset UTF-8
(para evitar que UTF-8 sea la opción por defecto)
- cambiar «AllowOverride None» por «AllowOverride All»
(para facilitar que mod_rewrite haga su trabajo y tener URLs amigables)

Instalar FTP

```
yum install vsftpd -y  
service vsftpd start  
chkconfig vsftpd on
```

Aunque no es obligatorio tener un servidor FTP te puede resultar cómodo para subir los archivos, si deseas tener más control sobre este asunto puedes configurar las opciones usando este completo manual de FTP y vsftpd.

Instalar PHP

```
yum install php php-mysql php-mbstring -y
service httpd restart
```

Si quieres cambiar la configuración de PHP, por ejemplo «magic_quotes_gpc = Off» por «magic_quotes_gpc = On», sólo tienes que modificar el archivo de texto «/etc/php.ini» usando el mismo procedimiento descrito en un párrafo anterior con la configuración de Apache.

Instalar GD

```
yum install gd gd-devel php-gd -y
service httpd restart
```

GD es una librería gráfica que le permite a PHP manipular imágenes, es prácticamente imprescindible hoy en día ya que la usan la mayoría de los gestores de contenidos.

Instalar MySQL

```
yum install mysql mysql-server -y
service mysqld start
chkconfig mysqld on
mysql_secure_installation
service mysqld restart
```

El comando `chkconfig` seguido de un servicio y `on` sirve para predeterminar que el servicio se inicie de forma automática al reiniciar el sistema, mientras que «`mysql_secure_installation`» sirve para configurar algunas opciones de MySQL y asignar una contraseña a root, es una especie de Wizard, una vez dentro puedes pulsar *ENTER* en todos los casos para dejar la opción por defecto, pero no te olvides de introducir una contraseña apropiada para el usuario root de MySQL.

Podríamos instalar phpMyAdmin para gestionar las bases de datos, pero hemos decidido hacer estas tareas desde la línea de comandos, una vez que sabes como se hace es todo más

rápido y limpio.

Crear base de datos

```
mysql -u root -pcontraseña
CREATE DATABASE basedatos;
SHOW DATABASES;
CREATE USER 'usuario'@'localhost' IDENTIFIED BY
'contraseña';
GRANT ALL PRIVILEGES ON basedatos.* TO
'usuario'@'localhost';
exit
```

En la primera línea debes indicar tu contraseña de root para MySQL (tal y como está, es decir seguida de la letra p y sin espacios en blanco que la separe).

También debes cambiar 'basedatos' por el nombre de la base de datos que estás creando, 'usuario' y 'contraseña' por el usuario y su contraseña que tendrá acceso a esa base de datos (evitando usar root por seguridad), fíjate que la primera contraseña es de root y la siguiente del usuario nuevo que tendrá acceso a la base de datos nueva.

Importar base de datos

En primer lugar debes subir el archivo con las ordenes SQL (puede ser exportado desde otro servidor o una copia de seguridad) a una ruta del servidor para tener acceso desde la línea de comandos, puede ser un archivo comprimido o sin comprimir y puedes usar FTP para subirlo.

```
gunzip /home/basedatos.sql.gz
mysql -u usuario -p'contraseña' basedatos <
/home/basedatos.sql
```

Si tu archivo tiene la extensión .sql.gz tienes que ejecutar la primera línea para descomprimirlo, en caso contrario debes pasar directamente a la segunda línea para importar el contenido, cambiando 'usuario', 'contraseña' y 'basedatos' por los datos apropiados.

Comprobar base de datos

```
mysql -u usuario -p'contraseña'  
SHOW DATABASES;  
USER basedatos;  
SHOW TABLES;  
exit
```

Puedes repetir los pasos anteriores para crear tantas bases de datos como necesites, así como importar los datos correspondientes.

Hemos indicado en mayúsculas las ordenes SQL para diferenciarlas del resto de comandos propios de linux, fíjate que todas las ordenes SQL deben acabar con un punto y coma.

Configurar los dominios

```
mkdir /home/dominio.tld  
vi /etc/httpd/conf.d/dominio.tld.conf
```

En el código anterior hay que cambiar 'dominio.tld' por el nombre de tu dominio y en el archivo de configuración que acabas de crear (con el segundo comando) hay que introducir el siguiente código, cambiando 'email' por una dirección de e-mail válida y 'dominio.tld' por tu nombre de dominio:

```
<VirtualHost *:80>  
ServerAdmin email  
DocumentRoot «/home/dominio.tld»  
ServerName dominio.tld  
ServerAlias www.dominio.tld  
</VirtualHost>
```

Puedes repetir el paso anterior tantas veces como dominios tengas y para terminar ejecutar lo siguiente para reiniciar apache y validar la configuración:

```
service httpd restart
```

Puesto que ya has creado la base de datos e importado sus

datos, lo único que te falta es subir por FTP los archivos a la carpeta que has indicado cuando configuraste los dominios y cambiar los permisos que sean necesarios para ver tu web online, después de cambiar los [DNS record o nameservers](#) de tu dominio o modificar tu archivo host.

Si tuvieras algún fallo y quisieras identificarlo puedes echar un vistazo a los logs del sistema, con alguno de los siguientes comandos:

```
cat /var/log/httpd/access_log  
cat /var/log/httpd/error_log
```

Eso es todo, no hemos querido entrar en la configuración del e-mail porque consideramos mucho más interesante no tener tus e-mails en la misma máquina que tu web, la mayoría de los registradores de dominio te facilitan una o varias direcciones de e-mail con tu dominio, gratis o casi gratis y de esa forma cada vez que cambias de hosting no tienes que crear las cuentas de e-mail ni perder ningún mensaje, en otra ocasión publicaremos un post para explicar como se hace esto de forma segura, fácil y rápida.

Todas estas pruebas han sido ejecutadas en un VPS SSD de [GinerNet](#) y el resultado ha sido que las páginas web cargan muy rápido y el precio es muy competitivo, debido a no tener que pagar la licencia del panel de control.