



Cómo crear tu propio servidor LAMP

15 MARCH 2016 on servidores, optimizacion, digitalocean, vps, hosting, apache

Venimos discutiendo hace rato sobre el futuro del desarrollo web y cómo estamos convencidos de que hoy por hoy, **el hosting compartido va cediendo terreno frente a**

los servidores privados virtuales (VPS).

Y por eso, decidimos armar este tutorial para que puedas empezar a **migrar tus proyectos actuales a tu propio servidor!**

Paso 1 - Registrarse en DigitalOcean

Primero que nada, necesitamos registrarnos en DigitalOcean, nuestro proveedor de hosting en la nube. **Haciendo click en éste link, tenemos US\$ 10 de regalo**, suficiente para nuestros primeros dos meses!

DigitalOcean nos enviará un mail de confirmación, lo seguimos, y actualizamos nuestros datos bancarios para que Digital pueda cobrarnos.

No se preocupen porque el costo es muy bajo, y el rendimiento es muy superior al de los hostings compartidos, además de la ventaja de que van a poder modificar a gusto su servidor, alojar todos los proyectos que quieran, usar todos los comandos de Linux, y demás.

Paso 2 - Crear tu primer Droplet

La interfaz gráfica de DigitalOcean es de las grandes ventajas que tiene éste proveedor: simple, intuitiva y rápida.

Una vez que tenemos nuestra cuenta y hemos ingresado los datos de nuestra tarjeta de crédito, crear nuestro primer servidor es de lo más sencillo!

La interfaz gráfica de DigitalOcean es de las grandes ventajas que tiene éste proveedor: **simple, intuitiva y rápida.**

Para crear nuestro primer Droplet (server en lenguaje DigitalOcean), vamos a clickear en el botón Create Droplet, luego elegiremos la pestaña One-Click Apps; veremos que tenemos un montón de imágenes ya pre-configuradas para utilizar.

En este caso, **instalaremos la imagen LAMP** (Linux, Apache, MySQL, PHP) on 14.04 (La versión de Ubuntu), elegiremos el tamaño del droplet (US\$ 5/mes será más que suficiente), un nombre para nuestro servidor, y le daremos click a Create.

DigitalOcean comenzará a construir nuestro droplet (no tarda más que 1 minuto!), y nos mostrará el progreso en el **Dashboard.**

Cuando termine de crearlo, nos llegará un mail con la información del mismo:

Nombre del Droplet: Para identificarlo en caso de que tengamos varios.

Dirección IP: Para acceder al servidor por SSH

Usuario: root, el superusuario en los sistemas Unix.

Password: Una clave para acceder por única vez con root, y luego cambiarla por una nuestra.

Paso 3 - Importar un proyecto al servidor

Una de las grandes ventajas de pasar a un VPS, es que podemos **dejar de depender de FTP** para transferir archivos y actualizar nuestros proyectos, y **empezar a usar git**

Ahora que tenemos nuestro servidor andando y ya estamos logueados, podemos comprobar que tanto Apache, como PHP y MySQL están instalados:

apache -v nos muestra la versión de Apache que estamos corriendo (2.4.7 Ubuntu)

php -v nos muestra la versión de PHP (5.5.9)

top nos muestra los procesos corriendo en nuestro servidor, apretando **Shift + M** podemos ordenarlos por consumo de memoria (MySQL estará primero generalmente)

Una de las grandes ventajas de pasar a un VPS, es que podemos **dejar de depender de FTP** para transferir archivos y actualizar nuestros proyectos, y **empezar a usar git** para ir actualizando sólo los últimos cambios, de manera rápida, eficiente y siguiendo un control de versiones.

Antes de importar nuestro proyecto, necesitaremos instalar **git** en el servidor:

apt-get update (actualizamos los repositorios de donde descargamos los paquetes)

apt-get install git

Para importar un proyecto a nuestro servidor, basta con pararnos en `/var/www` escribiendo **cd /var/www** y clonar el repositorio:

git clone protocol://githost.org/extuser/repo.git

Con **cd nombre-repo** podemos acceder al directorio del repositorio.

Paso 4 - Configurar DNS y Apache

Ahora, todos los pedidos que lleguen a nuestro dominio, se enviarán desde nuestro Domain Registrar a DigitalOcean, y de DigitalOcean, a nuestro servidor en particular.

Ahora que el proyecto ya está en nuestro servidor, nos queda configurar los DNS para que apunten a él y crear un virtual host en Apache para que maneje los requests que lleguen de ese dominio.

Supongamos que nuestro dominio es dominio.com, en DigitalOcean vamos a la sección de Networking / Domains, escribimos dominio.com donde dice Domain, elegimos del desplegable nuestro droplet, y seleccionamos Create Record.

En nuestro Domain Registrar (GoDaddy, Nic.ar, etc), tenemos que apuntar nuestros DNS a los de DigitalOcean, existentes en nuestro registro de dominio:

ns1.digitalocean.com
ns2.digitalocean.com
ns3.digitalocean.com

Ahora, todos los pedidos que lleguen a nuestro dominio, se enviarán desde nuestro Domain Registrar a DigitalOcean, y de DigitalOcean, a nuestro servidor en particular.

Por último, tenemos que configurar Apache para que redireccione los pedidos que entren desde dominio.com a la carpeta específica de nuestro proyecto `/var/www/proyecto`.

Para ello, nos dirigimos a `/etc/apache2/sites-available/` y creamos un nuevo archivo: **nano dominio.com.conf**
Ahí pegamos lo siguiente:

```
<VirtualHost *:80>
    ServerName dominio.com
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/proyecto
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Habilitamos nuestro sitio usando **a2ensite dominio.com.conf**

Recargamos Apache con **service apache2 reload** y ya estamos listos para servir nuestro sitio!

Lo que sigue

Ahora ya podemos repetir los pasos 3 y 4 para todos nuestros proyectos y **en poco tiempo habremos migrado a un servidor propio**, con la ventaja de que tenemos total control sobre los servicios que corren en el mismo, y podemos escalarlo para que cumpla con las exigencias que necesitemos, actualizar PHP a la versión 7 (con la consiguiente mejora de performance), modificar como maneja Apache los procesos de PHP (usar PHP-Fpm para

mejorar el uso de memoria), cambiar MySQL por MariaDB (también mejor uso de memoria), agregar monitores de uptime de servicios, *y mucho pero mucho más!*

En [CoffeeDevs](#), **estamos comprometidos a una web más rápida, eficiente y con una gran experiencia de usuario**, y trabajar usando VPS y Git en lugar de hosting compartido y FTP es una parte importantísima para lograrlo!

Si todavía no estás dispuesto a iniciarte en el mundo de los servidores, pero estás cansado de luchar con los hosting compartidos y estás buscando una solución económica y performante, [ponte en contacto](#) con nosotros y te ayudaremos!

Facundo Goñi

Read [more posts](#) by this author.

Share this post



YOU MIGHT ENJOY

Un año de DigitalOcean - Nuestra experiencia

Como agencia de desarrollo web, en CoffeeDevs tuvimos que decidir desde el primer día, qué tipo de servicio de...