

# Cómo Instalar Y Utilizar Lithium En VPS Con Ubuntu 12.04

Antes de comenzar este tutorial, daremos una breve explicación de lo que es **Lithium**, para que se usa, y como instalarlo en un **VPS** con **Ubuntu 12.04**.

## Que Es Lithium?

**Lithium** es un **framework de PHP** completa para el desarrollo de aplicaciones web. Basado en la arquitectura Modelo-Vista-Controlador (MVC), que está construido para PHP 5.3 y versiones superiores; se integra con las últimas tecnologías de almacenamiento como **MongoDB** o **CouchDB**.

Está diseñado para ofrecer tanto una gran organización del proyecto, así como la posibilidad de codificar fuera del **framework** a medida que desarrolle su propia **aplicación web**. Además, cuenta con un sistema de plugins robusta que le permite utilizar los componentes preferidos fuera del **framework** (como Twig for templating ó Doctrine2 for ORM).

En este articulo, el tutorial será un paso a paso en un **VPS**, si no tienes el tuyo, te invitamos a que contrates uno, en el mercado habrán muchos, pero pocos con un buen soporte, te recomendamos la [siguiente web para contratar tu VPS](#).

## Configuración De Apache

Ya que estamos usando **Apache** como **servidor web** y **Lithium** hace uso intensivo del **.htaccess**. Para la reescritura de URL, necesitaremos también asegurarnos de que Apache, de hecho, dejar que haga eso. Si usted no ha hecho esto, son pasos que ahora tendrás que hacer.

Edite el archivo del host virtual que es responsable de la carpeta en la que tendrá la aplicación (en nuestro caso, vamos a decir el valor por defecto de root de Apache: `/var/www`):

```
[js]sudo nano /etc/apache2/sites-available/default[/js]
```

Dentro del archivo añadimos lo siguiente:

```
[js]
<Directory /var/www/>
[/js]
```

Asegurate de que en lugar de **AllowOverride None** tienes **AllowOverride All**. Lo siguiente que tenemos que hacer es activar `mod_rewrite` (de nuevo, si usted no lo tiene activado). Para comprobar si ya está activada, utilice el siguiente comando:

```
[js]
apache2ctl -M
[/js]
```

Si ve «**rewrite\_module**» en la lista, usted está muy bien. Si no es así, utilice el comando siguiente para habilitar el módulo:

```
[js] a2enmod rewrite [/js]
```

Después de realizar los cambios necesarios, habrá que reiniciar apache:

```
[js] sudo service apache2 restart [/js]
```

## Instalando Lithium

Antes de comenzar la instalación de **Lithium**, vamos a instalar Git para que podamos usarlo para **descargar el framework de GitHub**. Usted puede hacer esto con los siguientes 2 comandos:

```
[js]
```

```
sudo apt-get update
sudo apt-get install git-core
```

```
[/js]
```

A continuación, podemos clonar el repositorio git de **Lithium** en nuestro servidor (mientras se está en la raíz de documentos de nuestro servidor web: /var/www para Apache):

```
[js]git clone git://github.com/UnionOfRAD/framework.git
site[/js]
```

Esto clonara el repositorio del **framework** y lo pondra en una carpeta llamada **site**. Ahora podemos instalar **Lithium** como un submódulo:

```
[js]
```

```
cd site
git submodule init
git submodule update
[/js]
```

Esto ahora clonar la biblioteca de **Lithium** también sobre nuestro servidor en la carpeta **libraries/lithium/**. Esto será necesario para poder instalar la aplicación.

## Línea De Comando

**Lithium** viene con una utilidad de línea de comandos (**li3**) que ayuda a la generación de código, documentación, etc, pero para que sea utilizable desde cualquier lugar, tendremos que añadir la biblioteca de la consola a la ruta del shell. Abra el archivo **.bash\_profile** ubicado en la **carpeta home** (si usted no tiene uno puede crearlo):

```
[js]nano ~/.bash_profile  [/js]
```

Y pegue el siguiente código en él:

```
[js] PATH=$PATH:/path/to/docroot/lithium/libraries/lithium/console[/js]
```

Asegúrese de reemplazar la ruta con el camino correcto que conduce a la consola. Así que en nuestro caso sería:

```
[js]PATH=$PATH:/var/www/site/libraries/lithium/console[/js]
```

Después de cualquier medida de este tipo, se debe ejecutar el siguiente comando para asegurarse de que el comando de bash tendrá efecto:

```
[js]source ~/.bash_profile[/js]
```

Y ahora pruebe el comando para asegurarse de que está trabajando, sin ninguna opción mostrara texto de ayuda:

```
[js] li3 [/js]
```

## Conexión De Base De Datos

La mayoría de las **aplicaciones web** necesitan una base de datos de la cual dependen para su almacenamiento. Con **Lithium** , puede utilizar una amplia gama de motores de bases de datos como **MySQL** , **MariaDB** , **MongoDB** , **CouchDB**, etc. A los efectos de la creación de nuestra aplicación de prueba vamos a utilizar **MySQL**, pero usted es libre de experimentar con la que usted se sienta más cómodo. Hay más información [aquí sobre su puesta en marcha con MongoDB](#).

Para configurar una conexión de base de datos, primero edite el archivo **Bootstrap.php** ubicado en la carpeta app/config de la aplicación (site/):

```
[js] nano /var/www/site/app/config/bootstrap.php [/js]
```

Dentro de este archivo, si comentó, descomentar la siguiente línea:

```
[js]require __DIR__ . '/bootstrap/connections.php';[/js]
```

A continuación, edite el archivo siguiente:

```
[js]nano  
/var/www/site/app/config/bootstrap/connections.php[/js]
```

Y quite de la configuración de la base de datos encontrado en el siguiente bloque:

```
[js]  
  
/**  
* Uncomment this configuration to use MySQL as your default  
database.  
*/  
  
[/js]
```

Se dará cuenta de varios bloques como éste para los diferentes motores de bases de datos. Además, establece la información de **conexión de MySQL** en su caso.

## La Aplicación

Es hora de visitar el navegador y ver lo que tenemos hasta ahora. Puede hacerlo ingresando a tu ip/sitio. Deberías ver la aplicación de Lithium funcionando con alguna información acerca de su estado y la configuración del servidor que necesita para que funcione.

Si aparece el siguiente mensaje:

```
[js]Magic quotes are enabled in your PHP configuration[/js]
```

Habra que editar el **archivo php.ini** en su servidor:

```
[js]sudo nano /etc/php5/apache2/php.ini[/js]
```

Y pegue la siguiente línea en él:

```
[js]magic_quotes_gpc = Off[/js]
```

A continuación, guarde el archivo y reinicie Apache:

```
[js]sudo service apache2 restart[/js]
```

## Modelo-Vista-Controlador

**Lithium** es un **framework MVC**, se verá en la estructura de 3 carpetas importantes para ello: `controllers/`, `models/` y `views/`. Vamos a crear rápidamente nuestro primer controlador e imprimir el típico **iHola, mundo !**.

Crear un nuevo archivo en la carpeta `controllers/` llamado **HelloController.php** con el siguiente contenido :

```
[php]
<?php

namespace app\controllers;

class HelloController extends \lithium\action\Controller {
public function index() {
echo "Hello World!";
}
}

?>

[/php]
```

Puede guardar el archivo. Lo que hicimos aquí fue crear una nueva clase de controlador que se encuentra en un archivo llamado cuidadosamente (basado en el nombre de la clase ) y que se extiende a la clase del controlador de Lithium. En el interior, hemos creado un método inicial que se llama si no se pasan parámetros al llamar este controlador. Dentro de este método acabamos de imprimir el mensaje.

Para acceder a esto en el navegador, tenemos que desplazarnos a `ip/site/hello` y usted debería ver **Hello World!** como mensaje

en el navegador.

En el siguiente **tutorial**, vamos a ir un poco más allá y ver cómo funciona la arquitectura **MVC con Lithium**. Usaremos Vistas al igual que los modelos (para ilustrar la interacción con nuestro motor de almacenamiento de MySQL).