RENDIMIENTO WEB

Ejercicio performance



**Autor**: Albert Delgado Roda

Instalación en máquina local

El sistema operativo con el que realizo esta práctica es: **Fedora 25 Workstation.**

Lo primero que he hecho tras ver el repositorio, ha sido realizar un fork de este a mi cuenta de [github](https://github.com/devcorpio).

Al clonar el proyecto en mi local e intentar lanzar un composer install, he encontrado mi **primer problema** y es que el composer.json espera que trabaje con la version **5.6** de **PHP**. En mi máquina tengo instalada la versión **7.1.4** de **PHP**

He valorado diferentes opciones:

1. Instalarme un **PHP** switcher, que me permita elegir la versión de **PHP** con la que trabajo, y que para esta práctica me permita pasar a tener **5.6** sin tener que desinstalarme la versión actual.
2. Crear una máquina virtual y instalarle directamente la versión **5.6** de **PHP.**
3. Pensar de la siguiente manera “esto esta bajo un framework **PHP**, y se espera que el cambio de **5.6** a **7** sea mero trámite porque se espera a que no hagan uso de funciones/constructores del lenguaje que hayan sido deprecadas/eliminadas en la versión **7** de **PHP**”, por tanto indicarle a composer que la versión requerida ahora sea a partir de la **7.0**

He optado por la opción **3**.

Pasos realizados:

1. En el composer.json, cambiar "php": "~5.6" por "php": "~7.0".
2. composer update

Ahora si se han instalado las dependencias. Por tanto lo siguiente que quiero hacer es ver en mi navegador la web ejecutandose. Para ello vía terminal y estando en el la raiz del proyecto lanzo el siguiente comando:

* php -S localhost:8000 -t web/

Al ir al navegador y poner la dirección **http://localhost:8000/** veo que me da un error 500. Por lo que me dice el servidor de php en ejecución, hay un problema con una template de twig que no encuentra.

Trás unos pocos minutos de depuración he podido ver en realidad donde está el problema. Este se encuentra en que se require crear una base de datos, con una serie de tablas encontradas en el fichero dataset.sql, por tanto realizo la pertinente creación de manera manual (aun siendo consciente que para cuando se necesito desployar en un servidor, ese paso manual debe ser automatizado, pero voy partido a partido)

Tras crear la estructura, me indica el que el servidor “mysql” es incapaz de encontrarlo, con lo cual valoro dos rápidas opciones:

1. cambiar mysql por 127.0.0.1
2. crear en /etc/hosts una referencia para mysql, evitando tener que tocar el código.

Opto por la opción **2**.

Tras realizar el paso elegido, la aplicación parece ya estar en funcionamiento:

# **There are no articles to show**

Hago un commit con todos los cambios realizados hasta el momento. Antes de ello veo que el directorio vendor me aparecer como commiteable, para evitar commits con cosas que no tocan, creo un .gitignore indicando precisamente eso, que ignore el directorio.

Commit realizado. **¡Ya está la aplicación arrancada en local!**

Deployar con ansistrano

Tras haber instalado la aplicación en local, ha llegado el momento de hacer el deploy de esta con ansistrano a mi servidor de amazon.

Tras haber visto el ejemplo que hay en la cuenta github, he optado por hacer una pequeña readaptacion de la estructura de directorios dentro del proyecto git, creando dos directorios padre, llamados **deployment** y **application**, toda la configuración de ansistrano ira en deployment, en applicatión estará el aplicativo php a deployar en amazon.

Para instalar ansistrano he lanzado el siguiente comando:

**sudo ansible-galaxy install carlosbuenosvinos.ansistrano-deploy carlosbuenosvinos.ansistrano-rollback**

El deployment en amazon siguiente la estrategia git, por tanto he instalado git en el server de amazon mediante el comando

**sudo yum install git**

Para conectarme a amazon necesito el .pem, por tanto he creado un directorio keys, en donde se encuentra el .pem.

Al haber configurado el host de ansible, y el playbook y haber lanzado el comando:

**ansible-playbook -i hosts deploy.yml**

El servidor de amazon me da un error indicando que el .pem, debería tener permisos 600 con lo cual mediante el comando chmod 600, cambio los permisos del .pem y vuelvo a lanzar el comando ansible.

Lamentablemente, creyendo que todo va bien, recibo el siguiente error:

**fatal: [ec2-54-171-123-73.eu-west-1.compute.amazonaws.com]: UNREACHABLE! => {"changed": false, "msg": "Failed to connect to the host via ssh: Received message too long 1349281121\r\n", "unreachable": true}**

Por lo visto el problema estaba en el fichero hosts, el ansible\_ssh\_user, era incorrecto, se estaba haciendo uso de **root**, cuando se debería de haber usado **ec2-user**.

Tras haber realizado el cambio, parece que el proceso ansible se ejecuta correctamente, ya que todo sale en letra verde. Pero de nuevo, parece que hay algún problema, si entro al servidor amazon y voy a la ruta **/var/www/html/blog**, veo que no hay absolutamente nada, con lo que el deploy no se ha realizado.

He copiado literalmente, el playbook de ejemplo que tienen en la documentacion, he probado de ejecutar y he visto que tardaba más y luego daba error, esto ya me ha dejado mejor espina, parece que hacía procesos por detras, con lo que seguramente tenía algo mal puesto, parece que en el deploy via git, no tenía había puestos comillas, con lo que apunta a que eso podría ser el problema.

Tras hacer los reemplazos con los que había copiado del ejemplo de la documentación de ansistrano he conseguido que haga el deploy en amazon.

Si entro en el lugar donde está alojado el deploy hecho por ansistrano:

[**http://ec2-54-246-133-48.eu-west-1.compute.amazonaws.com/blog/current/application/web/**](http://ec2-54-171-123-73.eu-west-1.compute.amazonaws.com/blog/current/application/web/)

Veré que hay un error 500, igual que me daba en local, esto sucede porque no tenemos ningún mysql configurado para este deploy.

Para esta práctica procederé a instalar mysql en el mismo servidor de amazon.

Usaremos este [rol](https://github.com/geerlingguy/ansible-role-mysql) de ansible galaxy para instalar mysql en amazon.

En mi máquina local he ejecutado el siguiente comando:

**sudo ansible-galaxy install geerlingguy.mysql**

Tras pensarlo detenidamente, he estos pasos manuales, los podriamos obviar, por tanto he procedido a crear roles en ansible, que me automatizen las siguientes tareas:

1. instalacion del repositorio epel (para poder instalarme php 7)
2. php 7.1
3. apache, incluye un reinicio de este, por si en el paso de php7.1 añado alguna extensión que no estaba de inicio
4. composer
5. git (instalo git porque como he mencionado antes, el deploy con ansistrano es mediante git)
6. La aplicacion php tiene puesto ‘mysql’ como host, por tanto para automatizar al máximo esto, en ansible he añadido un comando para que en el fichero /etc/host añada mysql 127.0.0.1
7. Cuando tenía deployado la aplicación en amazon, **PDO** lanzaba una excepción indicando “Permission denied” código: 1002, tras investigar, acabé en este post de [stackoverflow](http://stackoverflow.com/questions/4078205/php-cant-connect-to-mysql-with-error-13-but-command-line-can). De este post salío otro rol para ansible que se encargaba de activar mediante **setbool** el flag **httpd\_can\_network\_connect**
8. Mediante el rol de ansible galaxy **geerlingguy.mysql** instalo mysql, incluyendo la creación de la base de datos y el usuario que la aplicacion **PHP** espera.
9. Deployo en amazon centos, con **ansistrano**. Al acabar el deploy, mediante un hoot de after symlink, ejecuto composer install en el directorio current (donde se encuentra el código de la aplicación **PHP**.
10. Creo las tablas mysql que la aplicación esperaba tener de buen inicio.

En este proceso he realizado varios **commits**.

Usar bootstrap

Proseguiré con el apartado de utilizar bootstrap para hacer más bonito el diseño del blog, para llevarlo a cabo, lo haré en el entorno local (para algo lo he integrado!), el primer escollo a solucionar ha sido como acceder a los recursos tipo css,javascript desde la plantilla twig.

Por lo visto era mucho mas simple de lo que me imaginaba, puesto que los assets estaban en el mismo directorio **web**.

He creado dos ficheros twigs mas, que he llamado **header** y **footer**, donde respectivamente he puesto los ficheros css y los ficheros javascript, debo decir que el layout lo he trabajado de una manera muy básica, he usado lo más basico de bootstrap, los formularios son quizás donde se vea más bonito. Todas las pantallas hacen uso mediante “includes” del footer y del header.

En este punto realizo un nuevo **commit**.