
Práctica 1.1

Despliegue de Aplicaciones Web

José Juan Sánchez Hernández

Índice

- 1 Instalación de la pila LAMP en una instancia de OpenStack 1**
 - 1.1 Arquitectura web con un único servidor 1
 - 1.2 Tareas a realizar 2
 - 1.3 Entregables 2
 - 1.3.1 Documento técnico 3
 - 1.3.2 Scripts de Bash 3
 - 1.3.3 Actividad de ampliación 4
 - 1.4 Anexo. ¿Cómo gestionar los archivos .env? 4

- 2 Referencias 5**

- 3 Licencia 6**

Índice de figuras

Índice de cuadros

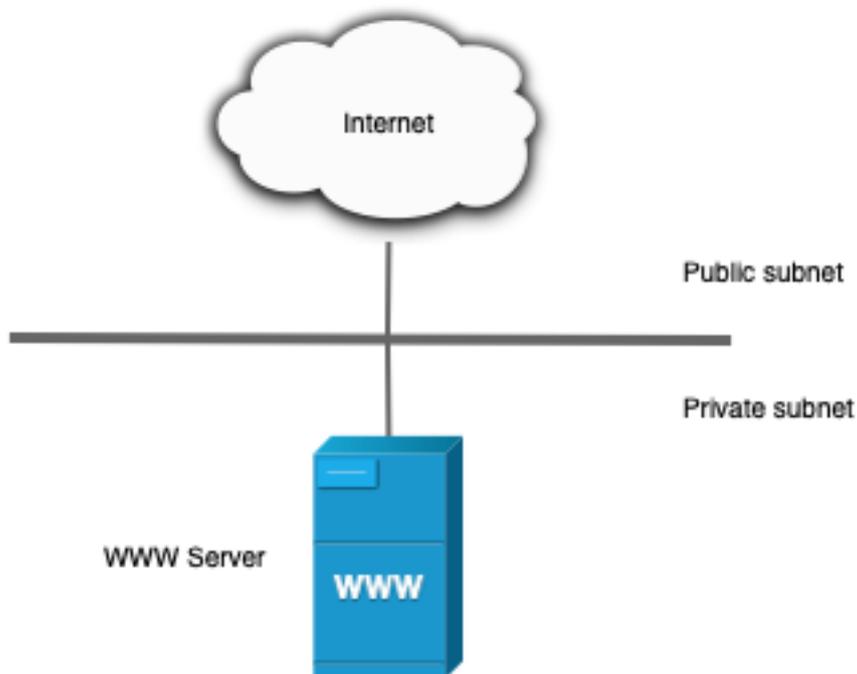
1 Instalación de la pila LAMP en una instancia de OpenStack

En esta práctica tendremos que crear una instancia en [OpenStack](#) con la última versión de [Ubuntu Server](#), e instalar todos los paquetes necesarios para tener una [pila LAMP](#) y todas las herramientas adicionales que hemos estudiado en el apartado de teoría de esta práctica.

Deberá automatizar el proceso de instalación y configuración de la pila LAMP, haciendo uso de todos los **scripts de Bash** que sean necesarios para llevarlo a cabo.

1.1 Arquitectura web con un único servidor

En esta práctica vamos a trabajar con una arquitectura web basada en un único servidor.



Ventajas:

- Es una solución sencilla de implementar ya que sólo tenemos un único servidor.
- Puede ser útil para aplicaciones que van a tener poca carga de trabajo.

Inconvenientes:

- Esta solución puede provocar problemas de rendimiento ante una carga de trabajo elevada, ya que todos los servicios se están ejecutando sobre la misma máquina y tienen que competir por los mismos recursos.
- Al estar todos los servicios ejecutándose sobre la misma máquina, no va a ser posible escalar nuestro sistema horizontalmente, es decir, añadiendo más nodos al sistema para repartir la carga. En este caso habrá que escalar verticalmente, añadiendo más recursos sobre la máquina donde están todos los servicios.

1.2 Tareas a realizar

Tendremos que realizar la instalación de la pila LAMP para el sistema operativo [Ubuntu Server](#), que se estará ejecutando en una instancia de [OpenStack](#).

A continuación se describen **muy brevemente** algunas de las tareas que tendrá que realizar.

1. Crea una máquina instancia EC2 en AWS.
2. La **Amazon Machine Image (AMI)** que vamos a seleccionar para esta práctica será una **Community AMI** con la última versión de **Ubuntu Server**.
3. Cuando esté creando la instancia deberá configurar los puertos que estarán abiertos para poder conectarnos por SSH y para poder acceder por HTTP/HTTPS.
 - SSH (TCP)
 - HTTP (TCP)
 - HTTPS (TCP)
4. Crea un par de claves (pública y privada) para conectar por SSH con la instancia. También puedes hacer uso de las claves que te proporciona AWS Academy (*vockey.pem*).
5. Crea una dirección **IP elástica** y asígnala a la instancia EC2.
6. Realice la instalación automática de la pila LAMP y todas las herramientas adicionales propuestas en la instancia EC2.
7. Busque cuál es la dirección IP elástica de su instancia y compruebe que puede acceder a ella desde un navegador web.

1.3 Entregables

Deberá crear un repositorio en [GitHub](#) con el nombre de la práctica y añadir al profesor como colaborador.

El repositorio debe tener el siguiente contenido:

- Un **documento técnico** con la descripción de todos los pasos que se han llevado a cabo.
- Los **scripts de Bash** que se han utilizado para automatizar la instalación y configuración de la pila LAMP, así como de las herramientas adicionales.

Además del contenido anterior puede ser necesario crear otros archivos de configuración. A continuación se muestra un ejemplo de cómo puede ser la estructura del repositorio:

```
1  .|—
2  README.md|—
3  conf|
4     └─ 000-default.conf|—
5  htaccess|
6     └─ .htaccess|—
7  php|
8     └─ index.php|—
9  scripts|—
10     .env|—
11     install_lamp.sh|—
12     install_tools.sh
```

1.3.1 Documento técnico

El documento técnico `README.md` tiene que estar escrito en [Markdown](#) y debe incluir **como mínimo** los siguientes contenidos:

- Descripción de la instalación de [Apache HTTP Server](#), [PHP](#), [MySQL Server](#) en la última versión [Ubuntu Server](#).
- Descripción de la instalación de [phpMyAdmin](#).
- Descripción de la instalación de [Adminer](#).
- Instalación del analizador de logs [GoAccess](#) para Apache Server.
- [Control de acceso a un directorio con .htaccess](#).
- Cada descripción debe ir acompañada de **alguna/s captura/s de pantalla**, donde se pueda ver claramente los pasos que se han llevado a cabo.

1.3.2 Scripts de Bash

El directorio `scripts` debe incluir los siguientes archivos:

- `.env`: Este archivo contiene todas las variables de configuración que se utilizarán en los scripts de Bash.
- `install_lamp.sh`: Script de Bash con la automatización del proceso de instalación de la pila LAMP.
- `install_tools.sh`: Script de Bash con la automatización del proceso de instalación de las herramientas adicionales.

1.3.3 Actividad de ampliación

Como **actividad de ampliación** se propone:

- Instalación del analizador de logs [AWStats](#) para Apache Server.

1.4 Anexo. ¿Cómo gestionar los archivos `.env`?

Los archivos `.env` son archivos de texto que se suelen utilizar para almacenar variables de entorno con información sensible, con contraseñas, claves de API, etc.

Estos archivos no se deberían subir a un repositorio público, porque cualquier persona podría acceder a ellos y obtener las contraseñas que contienen.

Paso 1. Crear un archivo `.gitignore` en el repositorio.

Para evitar que estos archivos se guarden en el repositorio podemos crear un archivo `.gitignore` en la raíz de nuestro repositorio y añadir el nombre del archivo que no queremos tener bajo el control de versiones, que en este caso es el archivo `.env`.

```
1 .env
```

Paso 2. Crear un archivo `.env.example` y añadirlo al repositorio.

Podemos crear un archivo `.env.example` y añadirlo al repositorio. Este archivo almacenará todas las variables que se tienen que configurar en el archivo `.env`, pero estarán sin inicializar.

Ejemplo:

```
1 PHPMYADMIN_APP_PASSWORD=  
2 APP_USER=  
3 APP_PASSWORD=  
4 STATS_USERNAME=  
5 STATS_PASSWORD=
```

De esta forma, cuando alguien quiera desplegar el proyecto que contiene el repositorio, sólo tendrá que copiar el archivo `.env.example`, renombrarlo a `.env` y actualizar las variables de entorno con los valores correctos.

Para gestionar estas variables en un entorno de producción existen varias soluciones:

- Se pueden crear directamente en el servidor.
- Sí se utiliza una plataforma de despliegue como [Vercel](#), [Netlify](#) o [Heroku](#), se pueden configurar en la interfaz de usuario web de la plataforma.
- Se pueden utilizar servicios para almacenar secretos como [AWS Secrets Manager](#) o [HashiCorp Vault](#).

2 Referencias

- [Amazon Web Services \(AWS\)](#)
- [Ubuntu Server](#)
- [LAMP Stack](#)
- [PHP](#)
- [Apache HTTP Server](#)
- [MySQL Server](#)
- [Configuraciones comunes para aplicaciones web](#)
- [Markdown](#)
- [PHPMyAdmin](#)
- [Adminer](#)
- [GoAccess](#)
- [AWStats](#)
- [Tutorial del Servidor Apache HTTP: Ficheros .htaccess](#)

3 Licencia

Esta página forma parte del curso Despliegue de Aplicaciones Web de José Juan Sánchez Hernández y su contenido se distribuye bajo una licencia Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional.