

Actividad I (Parte A) - Instalación y configuración de un servidor de aplicaciones y su interacción con un servidor web

Índice

Actividad 1 - Instalación y configuración de un servidor de aplicaciones y su interacción con un servidor web.....	1
1. Introducción.....	1
2. Instalación de Nginx.....	1
3. Instalación del servidor de aplicaciones PHP- FPM.....	2
3.1 Instalación del servicio.....	3
3.2 Estructura de directorios.....	3
3.2 Estructura de ficheros.....	4
3.3 Definición de un nuevo pool de conexiones.....	5
3.3 Configuración del server block.....	6
3.4 Prueba del servicio.....	6
4 Trabajo a realizar.....	7
5. Referencias.....	8

1. Introducción

El objetivo de esta práctica es analizar el funcionamiento del servicio **PHP-FPM** y las ventajas que proporciona respecto a los métodos tradicionales y monolíticos de despliegue de aplicaciones web mediante el módulo interno **mod_php** de apache.

Además, identificaremos los mecanismos que permiten la interacción entre un servidor web y un servidor de aplicaciones. Para ello, instalaremos Nginx y lo configuraremos para que colabore con el servidor de aplicaciones **PHP-FPM**.

Contesta las preguntas que irás encontrando y documenta aquellos procesos señalados de forma explícita con las capturas de pantalla necesarias.

2. Instalación de Nginx

Como ya vimos en la unidad anterior, nginx es un **servidor web ligero** de alto rendimiento y que también puede hacer de **proxy** inverso. Se trata de un software de código abierto que pertenece al llamado "Software Libre" y que se distribuye bajo la licencia **BDS** simplificada, es multiplataforma y funciona sobre sistemas Unix, Like-Unix (GNU / Linux, BSD, Solaris, Mac OS X, etc) y Windows.

Actualmente, podemos encontrar una larga lista de sitios web conocidos que lo utilizan como; **Worpress**, **Netflix**, **GitHub**, **SourceForge** e incluso a algunas partes de

Facebook.

La configuración del servicio es muy parecida a la del servidor web **apache**. La diferencia principal radica en que mientras en el servidor **apache**, la configuración se aplica para directorios específicos mediante la directiva `<Directory>`, en **nginx** se hace **para url** concretas `/image`, sin importar el directorio en el que se encuentran alojados los documentos o recursos web.

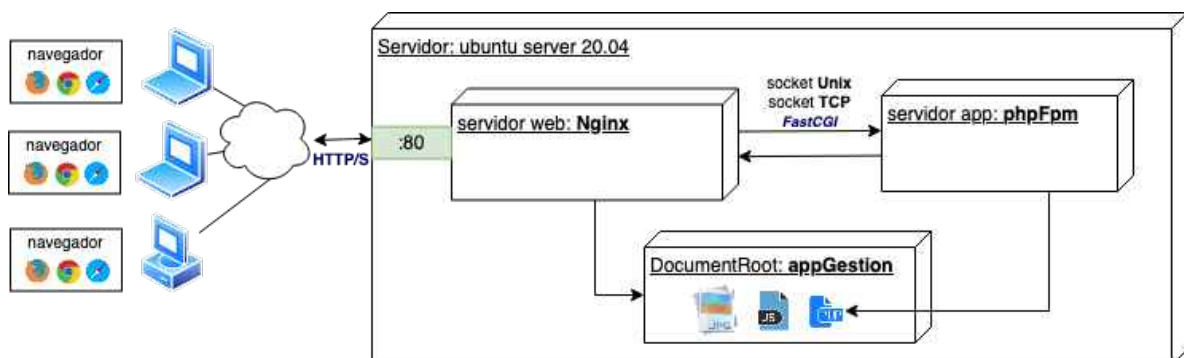
Actividad 1 (0.5 puntos)

Lee el [siguiente documento](#) sobre la instalación y configuración básica del servicio Nginx. Instala (en tu máquina virtual) Nginx según lo indicado en el documento (no cambies la configuración) y contesta las preguntas planteadas:

- ¿Cuál es el equivalente en Nginx a los **host virtuales** de apache? ¿Qué directiva se utiliza para configurar el dominio al que responderá cada **VirtualHost en Nginx?** y el **DocumentRoot?**
- ¿Cuál es el archivo de **configuración principal** de nginx?

3. Instalación del servidor de aplicaciones PHP- FPM

Puesto que **Nginx** no puede ejecutar **código PHP** de forma nativa, deberemos hacer uso de un servicio independiente, en este caso hemos elegido un servidor **PHP-FPM**. Cuando **Nginx** reciba una petición que requiera la ejecución de código **php**, hará uso de este servicio, reenviando las peticiones para que sean interpretadas. Una vez ejecutadas son devueltas como, un documento web que el servidor nginx servirá a los clientes.



3.1 Instalación del servicio

En distribuciones **linux** utilizaremos el gestor de paquetes, ejecutando la orden:

```
$ sudo apt install php-fpm
```

Tras ejecutar el comando, deberíamos tener instalado el servicio **php-fpm** en la versión de php que esté definida en nuestros repositorio. Para ubuntu 20.04 Server, se instalará la versión **php 7.4**. Este servicio se gestiona de forma independiente al del servidor web, mediante el comando **systemctl** o su correspondiente **service**.

<code>systemctl [start stop restart] php7.4-fpm</code>	<code>service php7.4-fpm [start stop restart]</code>
--------------------------------------------------------	------------------------------------------------------

La instalación por defecto deja abierto un **socket unix** en la ruta `/run/php/php{version}.sock`. Este será el que utilizaremos para comunicarnos con el **servidor de aplicaciones**.

Actividad 2 (0.5 puntos)

- ¿Qué permisos y propietario tiene asignado el fichero que representa el socket? ¿Por qué crees que tiene asignados dicho usuario y permisos?
- ¿Qué diferencia existe entre un socket Unix y un socket TCP?

3.2 Estructura de directorios

Antes de empezar a configurar el servicio, vamos a analizar la estructura de directorios creada durante el proceso de instalación del **servicio php-fpm** y la funcionalidad de cada una de ellas.

Fichero/directorio	Descripción
<code>/etc/php/{version}/fpm/conf.d</code>	<ul style="list-style-type: none"> - Contiene los módulos y configuraciones que tendremos disponibles para el intérprete de php. <i>Se trata de enlaces simbólicos a <code>/etc/php/{version}/mods-available</code></i>
<code>/etc/php/{version}/fpm/php-fpm.conf</code>	<ul style="list-style-type: none"> - Configuración general del servicio php-fpm; <i>error-log, log-</i>

	<i>level</i>
<code>/etc/php/{version}/fpm/php.ini</code>	- Configuración específica de php para este escenario.
<code>/etc/php/{version}/fpm/pool.d</code>	<ul style="list-style-type: none"> - Directorio con configuraciones para los diferentes pool de conexiones que atenderán las peticiones de nginx. - Podemos configurar diferentes pools (procesos distintos de php-fpm) para cada aplicación que queramos desplegar y, por tanto, configuraciones distintas.



Disponer de diferentes **pools de conexiones** para cada definición de **bloque de servidor** nos va a permitir aislar la ejecución de las aplicaciones y llevar a cabo configuraciones específicas para cada una de ellas.

3.2 Estructura de ficheros

La **instalación básica** del servicio crea un pool de conexiones, cuyo archivo de configuración podemos encontrar en la siguiente ruta. `/etc/php/{version-
php}/fpm/pool.d/www.conf`.

En la siguiente tabla, se explican algunas de las directivas de configuración más usuales. Puedes obtener una lista completa en la [documentación oficial](#).

<code>/etc/php/7.4/fpm/pool.d/www.conf</code>	
parámetros	Descripción
[identificador] <code>[www]</code>	identificador/nombre del pool , Si Tenemos varios, cada uno debe tener un número distinto
user, group <code>user = www-data</code> <code>group = www-data</code>	Usuario y grupo con el que se van ejecutar los procesos de apache (Determinará los permisos que debería tener cada DocumentRoot)
Listen <code>//Socket unix</code> <code>listen = /var/run/php/php7.2-fpm.sock</code> <code>//socket tcp</code> <code>listen = 127.0.0.1:9000</code>	Se indica el socket unix o el socket TCP que atenderá las peticiones.

<pre>//socket tcp en cualquier interfaz de red local a la máquina listen = 9000</pre>	
<p>pm</p> <pre>pm.max_children = 64 pm.max_requests = 500 pm.status_path = /status</pre>	<p>Directivas para la gestión de procesos</p> <p>En el ejemplo se habilita una página de monitorización del servicio php-fpm, se establece un máximo de 64 procesos activos y un máximo de 500 peticiones concurrentes</p>
<p>access</p> <pre>access.log = /var/log/\$pool.access.log access.format= "%R - %u %t \"%m %r%Q%q\""</pre>	<p>Directivas para la gestión del archivo de log para las peticiones que atiende el servidor php-fpm</p>

3.3 Definición de un nuevo pool de conexiones

Aunque no siempre es necesario, si necesitamos configurar un nuevo **pool de conexiones** para una segunda aplicación, solo deberemos crear una **copia del archivo** anterior.

```
$ sudo cp /etc/php/7.4/fpm/pool.d/www.conf /etc/php/7.4/fpm/pool.d/001-
ddaw-es
```

seguidamente estableceremos **un nuevo identificador** y **un nuevo socket** desde el que el servidor de aplicaciones se comunicará con el servidor web.

<pre>/etc/php/7.4/fpm/pool.d/001-ddaw-es</pre>
<pre>#Identificador del pool [001-ddaw-es] #Usuario y grupo con el se leerán y ejecutarán los documentos php user = www-data group = www-data #Socket de conexión listen = /var/run/php/php7.4-fpm-001-ddaw-es.sock</pre>

Una vez reiniciado el servicio, se abrirá el **socket** de forma automática y estará listo para su utilización

```
$ sudo service php7.4-fpm restart
```

Para comprobar que todo está correcto:

```
$ sudo service php7.4-fpm status
```

3.3 Configuración del server block

El siguiente paso será la configuración del **server block** de Nginx para que, cuando se le solicite una página escrita en php, haga uso del servidor de aplicaciones cuando .

En primer lugar, editaremos el fichero que contiene la configuración del sitio `/etc/nginx/sites-available/001-com-example` e incluiremos la siguiente directiva.

```

/etc/nginx/sites-available/001-com-example

server {
    listen 80;
    root /var/www/html;
    index index.php index.html;
    server_name example.com;

    #Definimos un nuevo bloque de configuración que se aplicará cuando
    #se acceda a cualquier fichero con extensión .php
    location ~ /\.php$ {
        #Incluimos configuración específica para el servicio
        include snippets/fastcgi-php.conf;

        #Establecemos la comunicación mediante Unix Socket
        fastcgi_pass unix:/var/run/php/php7.4-fpm-001-ddaw-es.sock;
    }
}

```



Es muy probable que ya dispongamos en el de configuración del **server block** las **directivas comentadas**, por lo que tan solo deberemos **comprobar que son correctas** y descomentarlas.

NOTA: Asegúrate que no exista ningún otro server block que esté entrando en conflicto con el nuevo creado (por ejemplo, con el mismo `server_name`). Si existiese otro, es necesario deshabilitarlo en `/etc/nginx/sites_enabled`.



3.4 Prueba del servicio

El intérprete de PHP nos proporciona de forma nativa la función **phpinfo()** que nos proporciona una gran cantidad de información sobre el estado las opciones de compilación, versión de PHP, extensiones instaladas, ubicación de los ficheros de configuración.

```
<?php
    phpinfo();

?>
```

Su ejecución nos proporcionará una página como la siguiente:

PHP Version 7.2.5-0ubuntu0.18.04.1	
	
System	Linux lemp-1804-2 4.15.0-20-generic #21-Ubuntu SMP Tue Apr 24 06:16:15 UTC 2018 x86_64
Build Date	May 9 2018 17:21:02
Server API	FPM/FastCGI
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.2/fpm
Loaded Configuration File	/etc/php/7.2/fpm/php.ini
Scan this dir for additional .ini files	/etc/php/7.2/fpm/conf.d
Additional .ini files parsed	/etc/php/7.2/fpm/conf.d/10-mysqld.ini, /etc/php/7.2/fpm/conf.d/10-opcache.ini, /etc/php/7.2/fpm/conf.d/10-pdo.ini, /etc/php/7.2/fpm/conf.d/20-calendar.ini, /etc/php/7.2/fpm/conf.d/20-ctype.ini, /etc/php/7.2/fpm/conf.d/20-exif.ini, /etc/php/7.2/fpm/conf.d/20-fileinfo.ini, /etc/php/7.2/fpm/conf.d/20-ftp.ini, /etc/php/7.2/fpm/conf.d/20-gettext.ini, /etc/php/7.2/fpm/conf.d/20-iconv.ini, /etc/php/7.2/fpm/conf.d/20-ionicons.ini, /etc/php/7.2/fpm/conf.d/20-mysqli.ini, /etc/php/7.2/fpm/conf.d/20-pdo_mysql.ini, /etc/php/7.2/fpm/conf.d/20-pear.ini, /etc/php/7.2/fpm/conf.d/20-posix.ini, /etc/php/7.2/fpm/conf.d/20-readline.ini, /etc/php/7.2/fpm/conf.d/20-shmop.ini, /etc/php/7.2/fpm/conf.d/20-sockets.ini, /etc/php/7.2/fpm/conf.d/20-sysmsg.ini, /etc/php/7.2/fpm/conf.d/20-syssem.ini, /etc/php/7.2/fpm/conf.d/20-sysvshm.ini, /etc/php/7.2/fpm/conf.d/20-tkcompat.ini
PHP API	20170718
PHP Extension	20170718
Zend Extension	320170718
Zend Extension Build	API320170718.NTS
PHP Extension Build	API20170718.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
DTrace Support	available, disabled
Registered PHP Streams	https, ftps, compress.zlib, php, file, glob, data, http, ftp, phar
Registered Stream Socket Transports	tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2
Registered Stream Filters	zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, convert.iconv.*
<small>This program makes use of the Zend Scripting Language Engine: Zend Engine v3.2.0, Copyright (c) 1998-2018 Zend Technologies with Zend OPcache v7.2.5-0ubuntu0.18.04.1, Copyright (c) 1999-2018, by Zend Technologies</small>	
	

4 Trabajo a realizar

Actividad 3 – Sitios Virtuales con Nginx (1 punto)

- Instala y configura un **servidor Nginx** con las siguientes especificaciones técnicas:
 - Dispondrá de **1 bloque de servidor** que atenderá las peticiones dirigidas al dominio `app.ddaw.severoochoa.es`.
 - Crea un documento `info.php` en el document Root. ¿Qué versión de php está ejecutando el servidor `php-fpm`? ¿Qué módulos/librerías tiene activas el

intérprete? ¿Cuál es el archivo principal de configuración del servicio?

- Activa el `access_log` para el `pool por defecto` del **servicio php-fpm** y haz una captura del mismo tras atender una petición.

5. Referencias

- Digital Ocean . Instalar un entorno LEMP.
["https://www.digitalocean.com/community/tutorials/how-to-install-linux-nginx-mysql-php-lemp-stack-ubuntu-18-04"](https://www.digitalocean.com/community/tutorials/how-to-install-linux-nginx-mysql-php-lemp-stack-ubuntu-18-04)
- Php.net. Documentación oficial PHP-FPM.
["https://www.php.net/manual/es/book.fpm.php"](https://www.php.net/manual/es/book.fpm.php)
- Digital Ocean. Instalación y configuración de Nginx en Ubuntu 18.04.
["https://www.digitalocean.com/community/tutorials/como-instalar-nginx-en-ubuntu-18-04-es"](https://www.digitalocean.com/community/tutorials/como-instalar-nginx-en-ubuntu-18-04-es)