

# MEMORIA PRUEBA TÉCNICA

Juan Crespín Valero

<b>INTRODUCCIÓN</b>	<b>3</b>
<b>SET UP MÁQUINAS VIRTUALES</b>	<b>4</b>
<b>CONFIGURACIÓN INSTANCIAS AWS</b>	<b>5</b>
<b>CONFIGURACIÓN BALANCEADOR</b>	<b>7</b>
1- INSTALACIÓN Y CONFIGURACIÓN SSH	7
2- INSTALACIÓN Y CONFIGURACIÓN NGINX	8
<b>CONFIGURACIÓN WORDPRESS</b>	<b>9</b>
1- INSTALACIÓN Y CONFIGURACIÓN BASE DE DATOS	9
2- INSTALACIÓN Y CONFIGURACIÓN APACHE	11
3- INSTALACIÓN Y CONFIGURACIÓN WORDPRESS	12
<b>ESTRUCTURA DE GITHUB</b>	<b>16</b>
<b>BIBLIOGRAFÍA</b>	<b>17</b>

# INTRODUCCIÓN

Para esta prueba técnica de crear una infraestructura con un servidor que actuará como un balanceador de carga que redirige las peticiones a otro servidor con un wordpress instalados he decidido hacer el mismo enfoque pero uno de forma local mediante máquinas virtuales de virtualbox y otro enfoque mediante las instancias ec2 de aws por lo que los pasos serán los mismos pero el contexto será diferente.

Para solucionar la prueba técnica he creado dos máquinas virtuales/instancias ambas con debian 11 y con acceso ssh.

- Para el acceso ssh he decidido usar el paquete de Debian OpenSSH y configurarlo para securizarlo , en el caso del entorno de máquinas virtuales será mediante contraseña y en el entorno de aws mediante claves, como se indica en el apartado correspondiente más abajo.
- Para las redirecciones del balanceador de carga usaré el paquete Nginx, la configuración está en el apartado correspondiente.
- Para la instalación y configuración del servidor Wordpress, está montado encima de un Apache para así limitar el acceso solo a las peticiones enviadas por el balanceador de carga.

- Acceso url al balanceador de carga que redirige las peticiones al servidor con Wordpress:

<http://ec2-15-188-11-173.eu-west-3.compute.amazonaws.com>

<https://ec2-15-188-11-173.eu-west-3.compute.amazonaws.com>

- Acceso mediante ssh a las distintas instancias AWS:

acceso balanceador: **15.188.11.173** / puerto: **2200** / usuario : **admin**

acceso wordpress: **13.39.84.137** / puerto: **2200** / usuario : **admin**

**Para acceder mediante ssh es necesario usar la clave adjunta en el correo.**

- Repositorio Github donde están las máquinas virtuales y su configuración:

[https://github.com/juancrespin/load\\_balancer\\_wordpress](https://github.com/juancrespin/load_balancer_wordpress)

# SET UP MÁQUINAS VIRTUALES

Para este supuesto es necesario lo siguiente:

## **INFORMACION DE LA INSTALACIÓN EN LOCAL**

Máquina virtual 1 (Balanceador de carga)

- Image: debían 11
- Hostname: balanceador
- IP red privada: 192.168.1.101
- IP adaptador acceso internet 192.168.0.101
- Usuario: administrator
- contraseña: administrator
- puerto escucha ssh 2200

Máquina virtual 2 (Wordpress)

- Image: debían 11
- Hostname: wordpress
- IP red privada: 192.168.1.225
- IP adaptador acceso internet 192.168.0.225
- Usuario: administrator
- contraseña: administrator
- puerto escucha ssh 2200

## **INFORMACION DE LAS INSTANCIAS EC2 AWS**

Instancia ec2 nº 1 (Balanceador de carga)

- Image: debían 11
- IP red privada: 172.31.38.150
- IP publica: 15.188.11.173
- Usuario: admin
- puerto escucha ssh 2200
- url acceso: ec2-15-188-11-173.eu-west-3.compute.amazonaws.com

Instancia ec2 nº 2 (Wordpress)

- Image: debían 11
- IP red privada: 172.31.34.17
- IP publica: 13.39.84.137
- Usuario: admin
- puerto escucha ssh 2200
- url acceso: ec2-13-39-84-137.eu-west-3.compute.amazonaws.com

Para acceder mediante ssh a las instancias es necesario obtener la clave de acceso que se adjuntan en el correo.


## CONFIGURACIÓN INSTANCIAS AWS

Para que las instancias puedan comunicarse entre sí es necesario que estén en el mismo VPC y en la misma subred, en este caso he seleccionado esta subred:

ID de subred

 [subnet-0d950c65a0bdeae9c](#) 

Zona de disponibilidad

 eu-west-3c

Y este VPC:

ID de VPC

 [vpc-0919538f8e08bf8ea](#) 

Y que la configuración del grupo de seguridad acepte las peticiones entrantes por estos puertos, debido a que si no, no podríamos acceder.

sgr-0666856e3ecbf55da	HTTP	TCP	80	Person...	0.0.0.0/0	Eliminar
sgr-033c5e428c9299f5a	SSH	TCP	22	Person...	0.0.0.0/0	Eliminar
sgr-07afb562c5ea13a5	TCP personalizado	TCP	2200	Person...	0.0.0.0/0	Eliminar
sgr-068809db29fcc120f	HTTPS	TCP	443	Person...	0.0.0.0/0	Eliminar
Agregar regla						

Para comprobar si las instancias son capaces de comunicarse y que la configuración funciona correctamente se cambia el fichero /etc/hosts de cada una de ellas para hacer el ping por nombre en vez de por ip para que sea más cómodo a la hora de configurar los pasos siguientes.

Fichero de configuración /etc/hosts para el balanceador:

```
GNU nano 5.4 /etc/hosts
# Your system has configured 'manage_etc_hosts' as True.
# As a result, if you wish for changes to this file to persist
# then you will need to either
# a.) make changes to the master file in /etc/cloud/templates/hosts.debian.tpl
# b.) change or remove the value of 'manage_etc_hosts' in
#    /etc/cloud/cloud.cfg or cloud-config from user-data
#
127.0.1.1 ip-172-31-38-150.eu-west-3.compute.internal ip-172-31-38-150
127.0.0.1 localhost
172.31.34.17 wordpress
```

Fichero de configuración /etc/hosts para el wordpress:

```
GNU nano 5.4 /etc/hosts *
# Your system has configured 'manage_etc_hosts' as True.
# As a result, if you wish for changes to this file to persist
# then you will need to either
# a.) make changes to the master file in /etc/cloud/templates/hosts.debian.tpl
# b.) change or remove the value of 'manage_etc_hosts' in
#    /etc/cloud/cloud.cfg or cloud-config from user-data
#
127.0.1.1 ip-172-31-34-17.eu-west-3.compute.internal ip-172-31-34-17
127.0.0.1 localhost
172.31.38.150 balanceador
```

Para comprobar que son capaces de comunicarse hacemos un ping:

```
admin@ip-172-31-34-17:~$ ping balanceador
PING balanceador (172.31.38.150) 56(84) bytes of data:
64 bytes from balanceador (172.31.38.150): icmp_seq=1 ttl=64 time=0.435 ms
64 bytes from balanceador (172.31.38.150): icmp_seq=2 ttl=64 time=0.570 ms
64 bytes from balanceador (172.31.38.150): icmp_seq=3 ttl=64 time=0.445 ms
64 bytes from balanceador (172.31.38.150): icmp_seq=4 ttl=64 time=0.459 ms
```

Una vez las instancias se pueden comunicar podemos pasar a la configuración de cada una.

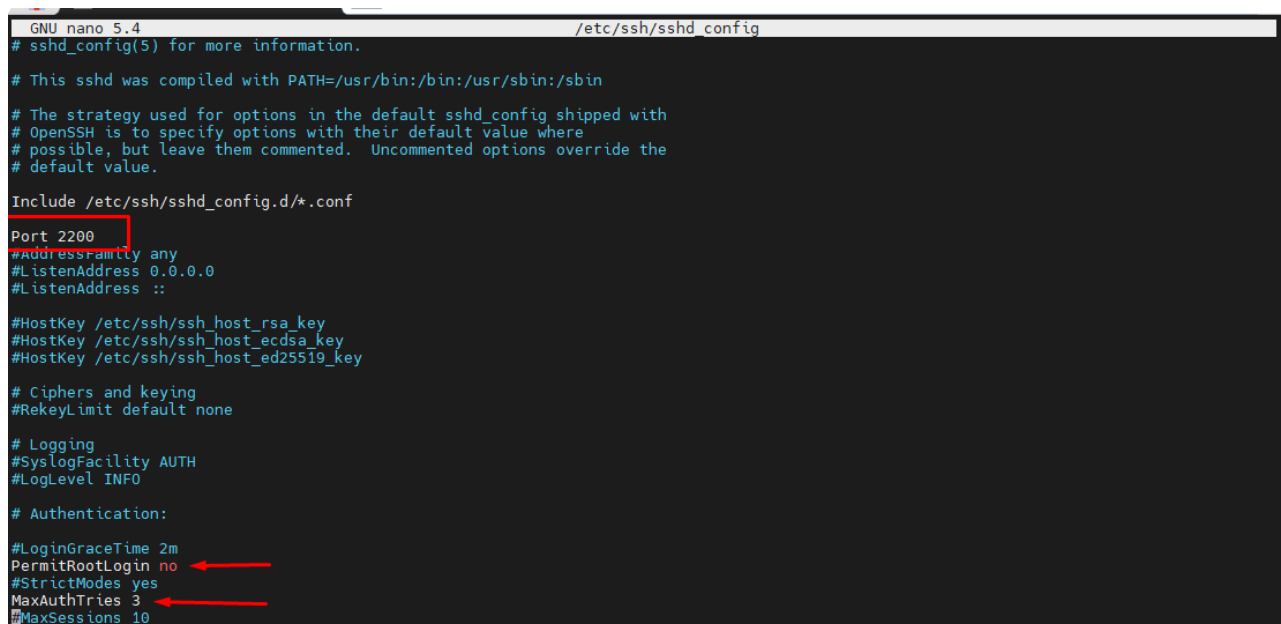
# CONFIGURACIÓN BALANCEADOR

## 1- INSTALACIÓN Y CONFIGURACIÓN SSH

El primer paso será instalar el servicio SSH:

```
administrator@balanceador:~$ sudo apt install openssh-server
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
```

Para securizar el acceso ssh al balanceador se decide cambiar los puntos seleccionados del fichero de configuración (que se encuentra en la ruta `/etc/ssh/sshd_config`) en la imagen siguiente:



```
GNU nano 5.4 /etc/ssh/sshd_config
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented. Uncommented options override the
# default value.

Include /etc/ssh/sshd_config.d/*.conf

Port 2200
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes
MaxAuthTries 3
MaxSessions 10
```

1. Para que el servicio ssh escuche en otro puerto es necesario cambiar el apartado de **Port**.
2. El apartado de **PermitRootLogin** se utiliza para configurar el acceso de root mediante ssh.
3. El último apartado **MaxAuthTries** es para configurar el máximo de intentos de acceso.

Para securizar más sería necesario configurar la autenticación basada en llave RSA en vez de por contraseña pero debido al caso, creo que no es necesario para este supuesto.

## 2- INSTALACIÓN Y CONFIGURACIÓN NGINX

Para las redirecciones se va a utilizar nginx ya que creo que es lo más simple para configurar un balanceador de carga y redireccionar.

El primer paso será instalar nginx:

```
administrator@balanceador:~$ sudo apt install nginx
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core geoip-database libdeflate0 libfontconfig1 libgd3 libgeoip1 libjpeg62-turbo
  libnginx-mod-http-geoip libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream
  libnginx-mod-stream-geoip libtiff5 libwebp6 libxpm4 libxslt1.1 nginx-common nginx-core
```

Debido a que se van a redireccionar peticiones http y https es necesario crear certificados ssl para que sea capaz de redireccionar a https, debido a que es un supuesto práctico se van a utilizar certificados autofirmados.

Creación de carpeta para tener todos los certificados en una ruta:

```
administrator@balanceador:~$ sudo mkdir -p /etc/nginx/ssl/certificados
administrator@balanceador:~$ cd /etc/nginx/ssl/certificados
```

Creación de la clave privada:

```
administrator@balanceador:~$ sudo openssl genrsa -des3 -out server.key 2048
```

Creación CSR(petición de firma certificada):

```
administrator@balanceador:/etc/nginx/ssl/certificados$ sudo openssl req -new -key server.key -out server.csr
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:SP
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:
Organization Name (eg, company) [Internet Widgits Pty Ltd]:
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:balanceador
Email Address []:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Al ser un certificado autofirmado es necesario autofirmar la clave mediante el siguiente comando:

```
administrator@balanceador:/etc/nginx/ssl/certificados$ sudo openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
Signature ok
subject=C = SP, ST = Some-State, O = Internet Widgits Pty Ltd, CN = balanceador
Getting Private key
```



modificamos el fichero por defecto de nginx (en la ruta `/etc/nginx/sites-available/default`) para que quede de esta forma (dejo la explicación en los comentarios del archivo):

```
GNU nano 5.4 /etc/nginx/sites-available/default
#Coleccion donde estaran los servidores a los que se van a redireccionar(puede ser ip o nombre)
upstream backend {
    server wordpress;
}
#Configuracion
server {
    #configuracion puerto http
    listen 80;

    #configuracion puerto https
    listen 443 ssl;

    #certificados ssl y su ruta
    ssl_certificate      /etc/nginx/ssl/certificados/server.crt;
    ssl_certificate_key  /etc/nginx/ssl/certificados/server.key;

    server_name _;

    #redireccionamiento al wordpress
    location / {
        proxy_pass http://backend;
        proxy_set_header    Host $host;
    }
}
```

Y con esta configuración ya estaría configurado el balanceador de carga, por lo que pasamos a configurar el wordpress.

## CONFIGURACIÓN WORDPRESS

Esta configuración se basará en Apache y la base de datos MariaDB.

### 1- INSTALACIÓN Y CONFIGURACIÓN BASE DE DATOS

El primer paso será instalar mysql , descargando el fichero mediante el comando **sudo wget https://dev.mysql.com/get/mysql-apt-config\_0.8.22-1\_all.deb**:

```
administrator@wordpressapp:~$ sudo wget https://dev.mysql.com/get/mysql-apt-config_0.8.22-1_all.deb
[sudo] password for administrator:
--2022-10-12 17:48:24-- https://dev.mysql.com/get/mysql-apt-config_0.8.22-1_all.deb
Resolving dev.mysql.com (dev.mysql.com)... 92.122.234.204, 2a02:26f0:e0:594::2e31, 2a02:26f0:e0:58b::2e31
Connecting to dev.mysql.com (dev.mysql.com)|92.122.234.204|:443 ... connected.
HTTP request sent, awaiting response... 302 Moved Temporarily
Location: https://repo.mysql.com//mysql-apt-config_0.8.22-1_all.deb [following]
--2022-10-12 17:48:25-- https://repo.mysql.com//mysql-apt-config_0.8.22-1_all.deb
Resolving repo.mysql.com (repo.mysql.com)... 23.214.73.59
Connecting to repo.mysql.com (repo.mysql.com)|23.214.73.59|:443 ... connected.
HTTP request sent, awaiting response... 200 OK
Length: 18012 (18K) [application/x-debian-package]
Saving to: 'mysql-apt-config_0.8.22-1_all.deb'

mysql-apt-config_0.8.22-1_all.d 100%[=====] 17.59K --KB/s in 0.001s
2022-10-12 17:48:25 (31.9 MB/s) - 'mysql-apt-config_0.8.22-1_all.deb' saved [18012/18012]
```

E instalandolo mediante el siguiente comando:

```
administrator@wordpressapp:~$ sudo apt install ./mysql-apt-config_0.8.22-1_all.deb
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
```

Por último instalamos el mysql-server:

```
administrator@wordpressapp:~$ sudo apt install mysql-server
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
```

Y ejecutamos el script **mysql\_secure\_installation** que servirá para configurar la seguridad, los puntos más importantes son los siguientes:

```
administrator@wordpressapp:~$ sudo mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...
```

Los puntos más importantes de este paso son los siguientes, para no permitir el acceso remoto del usuario root y también para que se borren los usuarios anónimos:

```
You already have your root account protected, so you can safely answer 'n'

Change the root password? [Y/n] n
... skipping.

By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
... Success!

Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y
```

Una vez instalado mysql procedemos a crear la base de datos con la que se conectara wordpress mediante los siguientes comandos, también crearemos un usuario para acceder.

```
administrator@wordpressapp:~$ sudo mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 36
Server version: 10.5.15-MariaDB-0+deb11u1 Debian 11

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE wordpress_db
→ ;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> CREATE USER 'admin'@'localhost' IDENTIFIED BY 'admin';
Query OK, 0 rows affected (0.003 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.000 sec)
```

Para que el usuario tenga permisos en esta base de datos es necesario otorgarselos y refrescar los permisos:

```
MariaDB [(none)]> GRANT ALL PRIVILEGES ON wordpress_db.* to admin@'localhost';
Query OK, 0 rows affected (0.003 sec)

MariaDB [(none)]> flush privileges;
Query OK, 0 rows affected (0.000 sec)
```

## 2- INSTALACIÓN Y CONFIGURACIÓN APACHE

Instalamos apache2 mediante el siguiente comando:

```
administrator@wordpressapp:~$ sudo apt-get install apache2
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
```

Es necesario instalar los siguientes paquetes para después instalar wordpress:

```
administrator@wordpressapp:~$ sudo apt install php libapache2-mod-php mariadb-server mariadb-client php-mysql php-curl php-xml p
hp-mbstring php-imagick php-zip php-gd
```

Creamos un fichero de configuración para wordpress con la configuración señalada en las capturas de abajo:

```
administrator@wordpressapp:~$ sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/wordpress.conf
```

```
GNU nano 5.4 /etc/apache2/sites-available/wordpress.conf
<VirtualHost *:80>
    # The ServerName directive sets the request scheme, hostname and port that
    # the server uses to identify itself. This is used when creating
    # redirection URLs. In the context of virtual hosts, the ServerName
    # specifies what hostname must appear in the request's Host: header to
    # match this virtual host. For the default virtual host (this file) this
    # value is not decisive as it is used as a last resort host regardless.
    # However, you must set it for any further virtual host explicitly.
    #ServerName www.example.com

    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/wordpress
    ServerName localhost

    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
    # error, crit, alert, emerg.
    # It is also possible to configure the loglevel for particular
    # modules, e.g.
    #LogLevel info ssl:warn
```

para que salga wordpress en vez de el html por defecto de apache es necesario activar la nueva configuración y desactivar la configuración por defecto con los comandos de la imagen:

```
administrator@wordpressapp:~$ sudo a2ensite wordpress.conf
Enabling site wordpress.
To activate the new configuration, you need to run:
  systemctl reload apache2
administrator@wordpressapp:~$ sudo a2dissite 000-default.conf
Site 000-default disabled.
To activate the new configuration, you need to run:
  systemctl reload apache2
```

### 3- INSTALACIÓN Y CONFIGURACIÓN WORDPRESS

Para descargar wordpress e instalarlo es necesario descargar un paquete con el comando wget:

```
administrator@wordpressapp:~$ wget -O /tmp/wordpress.tar.gz https://wordpress.org/latest.tar.gz
--2022-10-11 16:46:03-- https://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 21172479 (20M) [application/octet-stream]
Saving to: '/tmp/wordpress.tar.gz'

/tmp/wordpress.tar.gz 100%[=====] 20.19M 5.34MB/s in 4.3s
2022-10-11 16:46:08 (4.72 MB/s) - '/tmp/wordpress.tar.gz' saved [21172479/21172479]
```

y la descomprimos en la ruta **sudo tar -xzf /tmp/wordpress.tar.gz -C /var/www**

Le damos los permisos necesarios

```
administrator@wordpressapp:~$ sudo chown -R www-data:www-data /var/www/wordpress
```

Se accede a la web mediante la ip **192.168.0.225** y se rellenan los datos escritos antes en la instalación de mysql, en el apartado de la creación de la base de datos y el usuario:

A continuación tendrás que introducir los detalles de tu conexión con la base de datos. Si no estás seguro de ellos, contacta con tu proveedor de alojamiento.

Nombre de la base de datos	<input type="text" value="wordpress_db"/>	El nombre de la base de datos que quieres usar con WordPress.
Nombre de usuario	<input type="text" value="admin"/>	El nombre de usuario de tu base de datos.
Contraseña	<input type="password" value="admin"/>	La contraseña de tu base de datos.
Servidor de la base de datos	<input type="text" value="localhost"/>	Si localhost no funciona, deberías poder obtener esta información de tu proveedor de alojamiento web.
Prefijo de tabla	<input type="text" value="wp_"/>	Si quieres ejecutar varias instalaciones de WordPress en una sola base de datos cambia esto.
<input type="button" value="Enviar"/>		

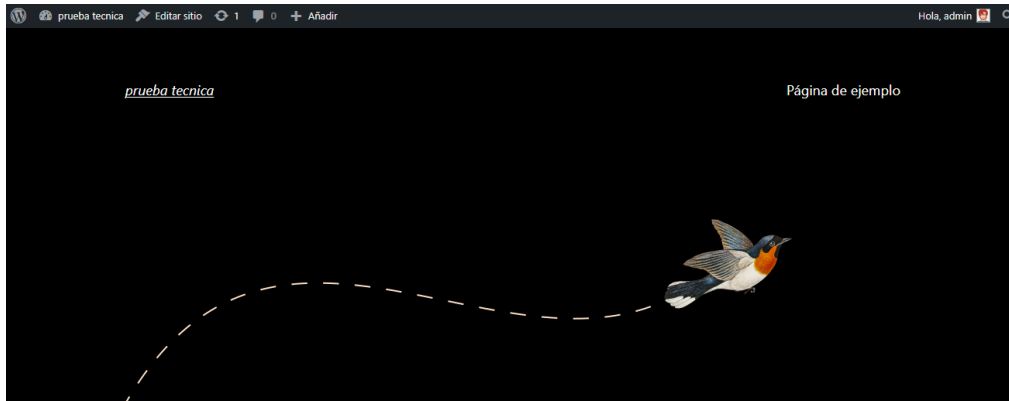
Rellenamos el siguiente formulario:

### Información necesaria

Por favor, proporciona la siguiente información. No te preocupes, siempre podrás cambiar estos ajustes más tarde.

Título del sitio	<input type="text" value="prueba tecnica"/>
Nombre de usuario	<input type="text" value="admin"/> <small>Los nombres de usuario pueden tener únicamente caracteres alfanuméricos, espacios, guiones bajos, guiones medios, puntos y el símbolo @.</small>
Contraseña	<div><input type="password" value="....."/> Very weak</div> <div><input type="button" value="Show"/></div> <p><b>Importante:</b> Necesitas esta contraseña para acceder. Por favor, guárdala en un lugar seguro.</p>
Confirma la contraseña	<input checked="" type="checkbox"/> Confirma el uso de una contraseña débil.
Tu correo electrónico	<input type="text" value="j.crespin.ies@gmail.com"/> <small>Comprueba bien tu dirección de correo electrónico antes de continuar.</small>
Visibilidad en los motores de búsqueda	<input type="checkbox"/> Pedir a los motores de búsqueda que no indexen este sitio <small>Depende de los motores de búsqueda atender esta petición o no.</small>

y al terminar la instalación iniciamos sesión y debería aparecer algo como esto al acceder a la web mediante la IP debido a que aún no hemos configurado que solo acepte las peticiones que le vengan desde el balanceador:



¡Hola, mundo!

Para configurar que acepte solo tráfico que le venga del balanceador es necesario añadir la siguiente información en el archivo de configuración de apache2, lo que hace es negar el acceso a todas las IP menos a la IP del balanceador.

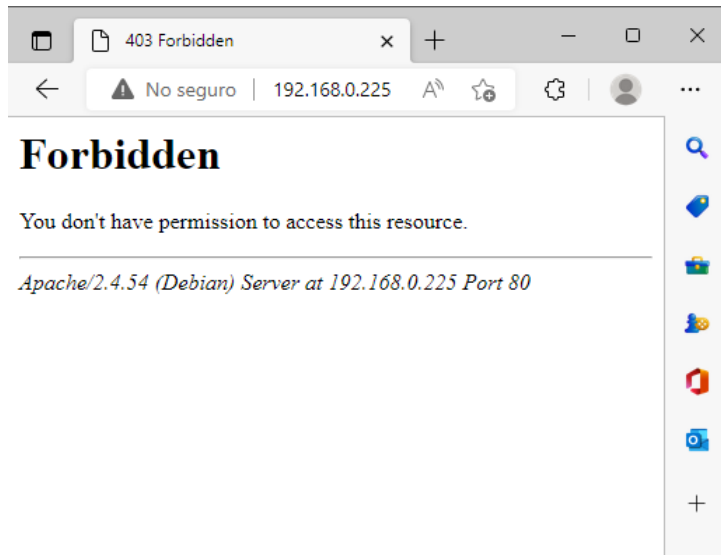
```
GNU nano 5.4
    AllowOverride None
    Require all granted
</Directory>

<Directory /var/www/>
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all granted
</Directory>

<Directory "/var/www/wordpress">
    Options Indexes FollowSymLinks
    AllowOverride None
    Require all denied
    Require ip 192.168.1.101
</Directory>
#<Directory /srv/>
#     Options Indexes FollowSymLinks
#     AllowOverride None
#     Require all granted
#</Directory>
```

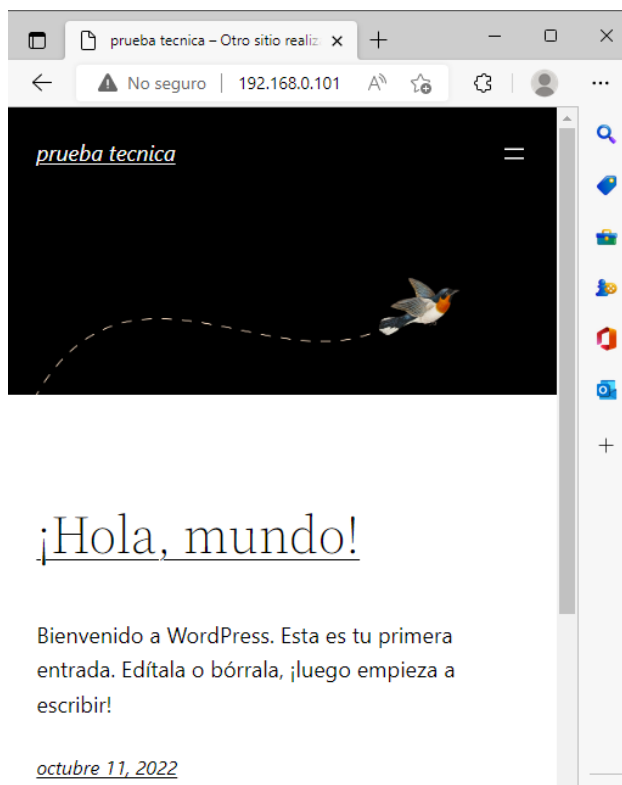
Una vez se ha configurado esto se reinician los servicios pertinentes y a la hora de acceder se tiene que acceder mediante el balanceador.

Intento de acceso directamente al wordpress:



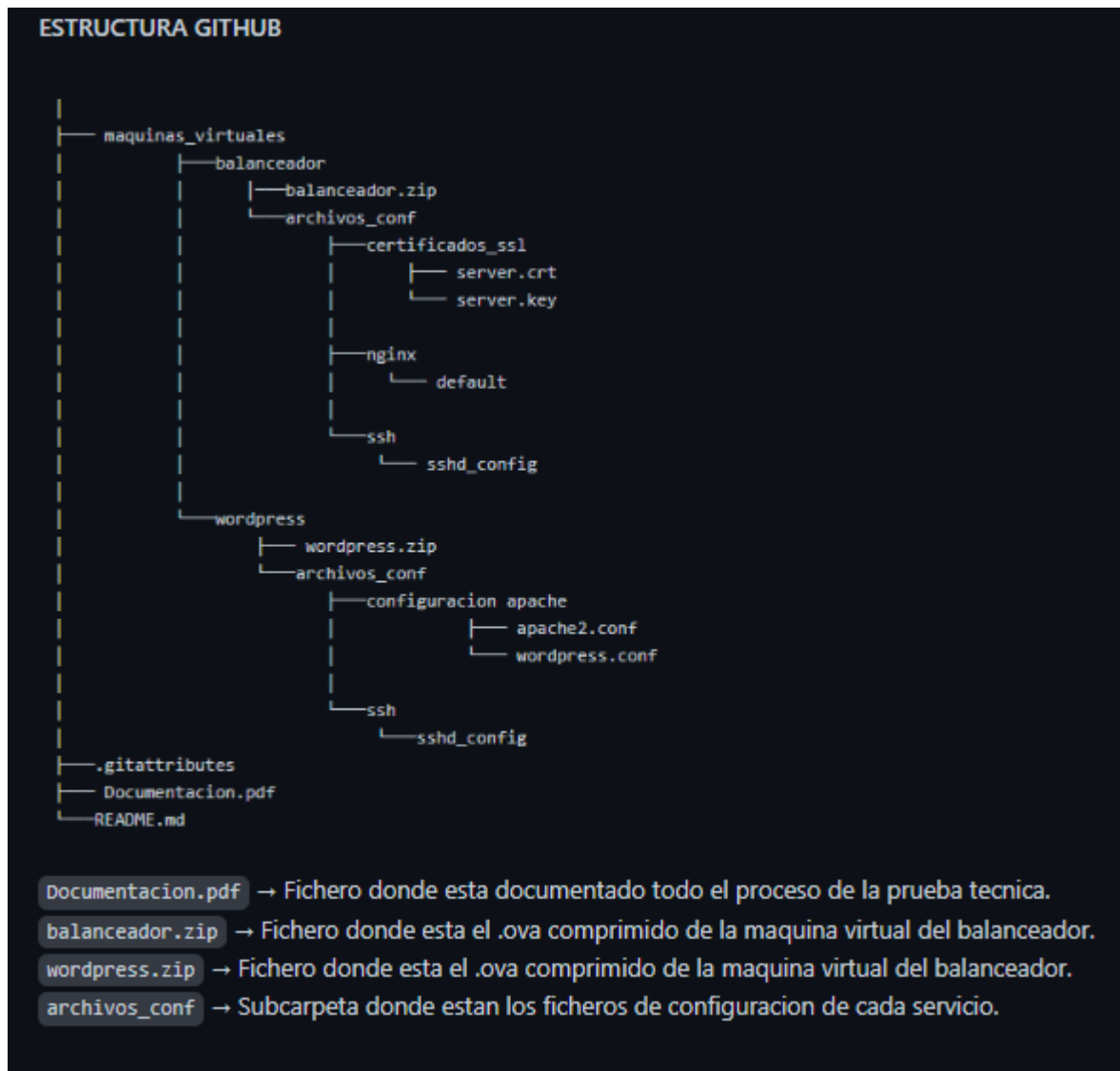
Como se puede observar no se puede acceder directamente por lo que hay que acceder mediante el balanceador.

Conexión mediante el balanceador:



# ESTRUCTURA DE GITHUB

La estructura del apartado de github es la siguiente:





# BIBLIOGRAFÍA

Instalacion wordpress:

<https://cloudinfrastructureservices.co.uk/install-wordpress-on-debian-10-11>

Creación de certificados ssl nginx:

<https://www.digitalocean.com/community/tutorials/how-to-set-up-nginx-load-balancing-with-ssl-termination>

Configuración nginx:

<https://www.digitalocean.com/community/tutorials/how-to-set-up-nginx-load-balancing>  
[http://nginx.org/en/docs/http/configuring\\_https\\_servers.html](http://nginx.org/en/docs/http/configuring_https_servers.html)

Configuracion ssh:

<https://www.howtogeek.com/443156/the-best-ways-to-secure-your-ssh-server/>

## **Configuración AWS:**

Grupos de seguridad:

[https://docs.aws.amazon.com/es\\_es/AWSEC2/latest/UserGuide/ec2-security-groups.html](https://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/ec2-security-groups.html)

Claves de seguridad(SSH):

[https://docs.aws.amazon.com/es\\_es/AWSEC2/latest/UserGuide/ec2-key-pairs.html](https://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/ec2-key-pairs.html)