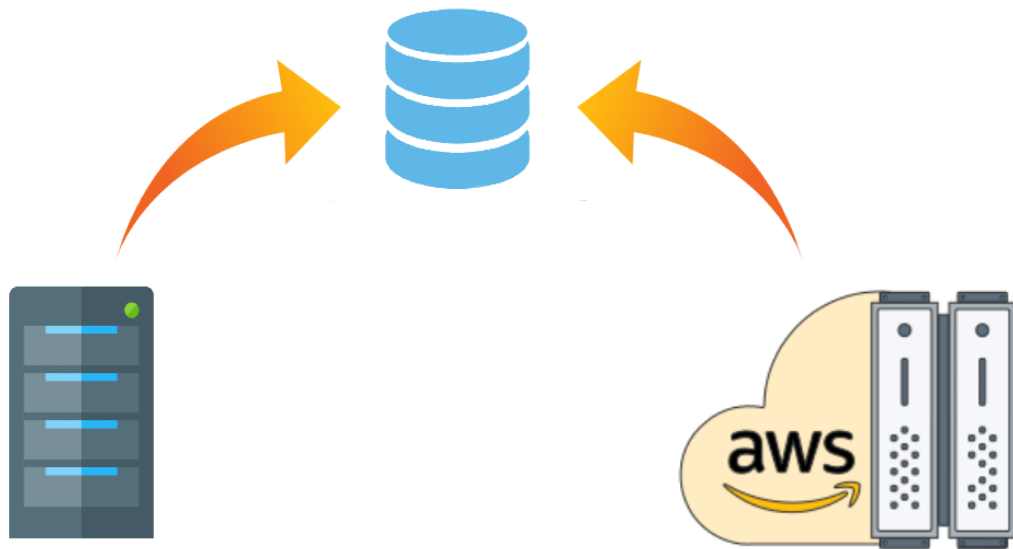


VERSIÓN 0.0

FECHA



## PRESTASHOP & DUAL SERVER

DANIEL PANDO MORENO / JAVIER RUIZ ROMERA

|                                        |           |
|----------------------------------------|-----------|
| <b>Objetivo</b>                        | <b>2</b>  |
| Introducción                           | 2         |
| HAProxy                                | 2         |
| Instancia AWS                          | 2         |
| PrestaShop                             | 2         |
| OwnCloud                               | 2         |
| <b>Investigación</b>                   | <b>3</b>  |
| <b>Público objetivo</b>                | <b>4</b>  |
| <b>Servicios</b>                       | <b>5</b>  |
| Amazon Web Service                     | 5         |
| Prestaciones Free Tier (Instancia AWS) | 5         |
| Prestashop                             | 6         |
| HaProxy                                | 7         |
| Owncloud                               | 7         |
| <b>Configuración</b>                   | <b>8</b>  |
| Configuración HAProxy                  | 8         |
| FRONTEND                               | 8         |
| BACKEND                                | 8         |
| Configuración Instancias AWS           | 10        |
| Instalación PrestaShop                 | 11        |
| Dominio + SSL                          | 18        |
| Réplica entre 2 base de datos          | 22        |
| Master                                 | 22        |
| Slave                                  | 23        |
| OwnCloud (BACKUPS)                     | 24        |
| Automatización de backups              | 27        |
| Servidor principal AWS                 | 27        |
| Servidor OwnCloud                      | 27        |
| <b>Planificación</b>                   | <b>28</b> |
| Diagrama de Gantt                      | 28        |
| Estudio económico y presupuestario     | 29        |
| <b>Conclusión</b>                      | <b>31</b> |
| <b>Anexo</b>                           | <b>32</b> |
| Errores                                | 32        |
| Glosario                               | 36        |
| WebGrafía                              | 37        |

## OBJETIVO

El objetivo de este proyecto es crear y configurar 2 servidores web que tengan implementado *PrestaShop*. Un servidor principal donde se lleve la mayor carga, y un servidor secundario alojado en la nube para no tener un corte de conexión si el principal falla y poder seguir dando servicio a los clientes.

Además de crear y configurar lo anterior, tener un apartado de backups utilizando ownCloud ubicado en un disco duro externo a los servidores para tener una mayor seguridad.

---

## INTRODUCCIÓN

El proyecto se divide en 3 partes; servidor principal alojando el balanceador de carga (HAproxy), los 2 servidores AWS que están en la nube, los cuales tienen el servicio PrestaShop instalado y configurado y un tercer servidor para las copias de seguridad (OwnCloud).

### HAPROXY

HAproxy es un balanceador de carga gratuito de código abierto, el cual nos ha permitido configurar 2 servidores, para que cuando uno de ellos tiene un fallo de conexión o está muy saturado de peticiones redirija automáticamente al otro servidor que supuestamente no ha tenido ningún corte de conexión y está más libre de carga.

### INSTANCIA AWS

Una *instancia AWS* es una máquina física alojada en la nube proporcionada por Amazon, que tiene unas ciertas especificaciones de hardware según el Tier que escojamos.

### PRESTASHOP

PrestaShop es una plataforma eCommerce completa, innovadora y eficaz que permite crear tu tienda online y desarrollar tu negocio en internet.

### OWNCLOUD

El servicio OwnCloud es un servicio de alojamiento de archivos, el cual hemos utilizado para hacer copias de seguridad de la base de datos de PrestaShop. Al estar las bases de datos sincronizadas, solo hace falta hacer la copia de seguridad en un servidor AWS.

## INVESTIGACIÓN

A la hora de empezar el proyecto nos hemos tomado el tiempo necesario para poder investigar sobre las tecnologías y las métricas que utilizaremos durante el transcurso del mismo.

A la hora de analizar la situación, nos hemos decantado por utilizar una serie de tecnologías con las cuales no estamos familiarizados y así ponernos un objetivo más complicado a la vez que obtenemos nuevos conocimientos.

Ante la gran variedad de servicios que se ofrecen para alojar servidores en la nube, hemos decidido implementar parte del proyecto en 'Amazon Web Service' en la cual está únicamente dedicado para los servidores web (PrestaShop). Una de las tecnologías que hemos escogido es HAProxy, para implementarla en un servidor local el cual se encargará de balancear la carga de los dos servidores web.

Ya que en clase hemos visto *WordPress* y a nosotros nos gustan los retos, hemos decidido utilizar *PrestaShop*, porque según investigaciones que hemos realizado en diferentes sitios web, es uno de los e-commerce más utilizados a nivel de España.



PrestaShop



## PÚBLICO OBJETIVO

El público objetivo de este proyecto es gente de cualquier edad que tiene una base de conocimientos sobre mecánica, ya que el principal producto que se vende en la web son materiales para la reparación de distintos vehículos.

Esta página web tiene un mercado bastante reducido al solo ser de materiales de mecánica, pero esto no quiere decir que no pueda entrar cualquiera a visitarnos, y que haya sido recomendado por una persona con conocimientos sobre mecánica para hacer una compra, o una persona que simplemente desea obtener una primera idea sobre el costo de una posible reparación para su vehículo.



## SERVICIOS

**AMAZON WEB SERVICE**

**Amazon Web Services** (AWS) es la plataforma en la nube más adoptada y completa en el mundo, que ofrece más de 175 servicios integrales de centros de datos a nivel global.

Cuenta con una cantidad de servicios y de características incluidas en ellos, ofreciendo desde tecnologías de infraestructura como cómputo, almacenamiento y bases de datos hasta tecnologías emergentes como aprendizaje automático e inteligencia artificial, lagos de datos y análisis e internet de las cosas.



Después de una larga investigación y bastantes pruebas sobre los diferentes servicios de cloud computing que hay en el mercado, nos hemos decantado por *Amazon Web Services*.

El primer servicio que probamos fue este, *AWS*, porque es el más famoso y supuestamente la mejor plataforma en la nube, pero no queríamos ser conformistas y quedarnos con el primero que veíamos.

Gracias a nuestra pequeña experiencia trabajando en la informática, conocíamos de varios servicios, uno de ellos es **Microsoft Azure**, el cual probamos e intentamos testear para ver cómo se manejaba y si estaba a la altura de AWS. Después de varias pruebas, vimos que si estaba al nivel y ofrecían las mismas características, tanto uno como otro, pero no nos convenció por su aspecto y panel de control que tenía, no nos sentíamos cómodos. El tercer servicio que probamos fue **Google Cloud**, el cual también ofrecía las mismas características que los demás servicios, y como no, siendo *Google*, estaba al nivel, pero tampoco nos convencía del todo como para utilizarlo en el proyecto. Siendo así que todos estos servicios ofrecen lo mismo en la capa gratuita, escogimos al gran Amazon básicamente por su aspecto y manera de manejarlo, o puede ser que nos haya influido simplemente por su nombre.

**PRESTACIONES FREE TIER (INSTANCIA AWS)**

| EC2           | Limitaciones                                                                                                   |
|---------------|----------------------------------------------------------------------------------------------------------------|
| Free Tier     | 2GB Ram, 30 GB SSD, 750 h/mes                                                                                  |
| Personalizado | Según qué hardware seleccionemos el precio varía, además de la cantidad de datos que entra y sale del servidor |

---

## PRESTASHOP

**Prestashop** es una plataforma *eCommerce* principalmente diseñada en PHP, la cual ofrece como función principal poder crear una tienda online a nuestro gusto sin la necesidad de programar desde cero. Además de esta función, tiene distintos apartados donde podemos gestionar nuestro negocio en un par de clicks, como por ejemplo: análisis de ventas, el cual permite tener a nuestro alcance información sobre las compras que realizan los clientes y así poder utilizar la zona de marketing que viene incorporada para poder realizar newsletter personalizadas y automatizadas para cada cliente.

Una característica divertida que tiene, es que podemos instalar e activar módulos con un solo click. Esto nos facilita bastante personalizar la web y a la vez que sea potente, y por supuesto nos ahorra tiempo, que es muy importante.



# PrestaShop

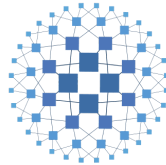
Hemos escogido **PrestaShop** porque queríamos obtener diferentes puntos de vista sobre las plataformas de creación de *eCommerce*, y así ampliar nuestro conocimiento para después tener un abanico de conocimientos más amplios hacia el sector laboral.

Además en múltiples búsquedas hemos visto que, prestashop es uno de los que tiene una interfaz más sencilla pero también permite realizar una completa configuración sobre el frontend y en el backoffice.

---

## HAPROXY

**HAProxy**, que significa High Availability Proxy (Proxy de alta disponibilidad), es un popular software de código abierto TCP/HTTP Balanceador de carga y una solución de proxy que se puede ejecutar en Linux, Solaris y FreeBSD. Su uso más común es mejorar el rendimiento y la confiabilidad de un entorno de servidor distribuyendo la carga de trabajo entre múltiples servidores (por ejemplo, web, aplicación, base de datos). Se utiliza en muchos entornos de alto perfil, incluidos: GitHub, Imgur, Instagram y Twitter.



## HAPROXY

Hemos escogido HAProxy principalmente porque fue una de las diferentes opciones que se nos propuso como ideas. Este es el que más nos gustó por la simplicidad que tiene y a la misma vez nos permitía enlazarlo con los servidores externos a la red local donde está el balanceador de carga.

---

## OWNCLOUD

**OwnCloud** es un servicio de almacenamiento y sincronización de archivos multiplataforma que se puede instalar en nuestro servidor, donde cualquier usuario con una cuenta puede subir información y se sincronizará con los demás usuarios en cualquiera de sus dispositivos.



Hemos escogido *OwnCloud* porque al estar la web montada en 2 servidores de *Amazon* los cuales cada uno está en una región/país diferente, *OwnCloud* nos permite acceder a las copias de seguridad desde cualquier dispositivo en cualquier localización en la que estemos en ese momento. Además, este servicio ya lo conocíamos y habíamos trabajado con él, por eso decidimos apostar en él, ya que siempre nos ha funcionado sin problemas.



## CONFIGURACIÓN

### CONFIGURACIÓN HAPROXY

Para empezar a configurar HAProxy, antes de nada tenemos que tener la máquina actualizada y entonces instalamos el servicio.

```
sudo apt update && apt upgrade
sudo apt install haproxy
```

Una vez instalado, el archivo de configuración lo podemos encontrar en su correspondiente directorio : `/etc/haproxy/`

Dentro de este directorio se encuentra el servicio haproxy, por lo que solo necesitaremos editar el archivo de configuración '`haproxy.cfg`'. Para no tener sustos, primero de todo hacemos un backup de este fichero y ahora si que si empezamos a configurar.

Nos encontraremos con una configuración por defecto, la cual no hace la función de un balanceador de carga pero define los parámetros a nivel de sistema, como la localización de los ficheros y el usuario y grupo sobre el que HAProxy está siendo ejecutado, además de definir parámetros de log adicionales relacionados a *timeouts* i/o *errores*. Esta última opción la podemos deshabilitar, pero puede ser una herramienta útil a la hora de configurar.

Una vez sabido lo que hace esta configuración por defecto, nos adentramos a la configuración del balanceador de carga.

Cuando configuramos HAProxy hay dos tipos de nodos que deben ser definidos; frontend y backend.

### FRONTEND

Este nodo es por el cual HAProxy escucha las conexiones.

```
frontend haproxynode
  bind *:80
  mode http
  default_backend backendnodes
```

Esta configuración de frontend especifica el nombre del nodo (haproxynode), el cual está vinculado a todas las interfaces de red en el puerto 80. Escuchará las conexiones HTTP y usará el backend '`backendnodes`'.

### BACKEND

Este nodo es donde HAProxy puede enviar solicitudes.

```
backend backendnodes
  balance roundrobin
  option forwardfor
  http-request set-header X-Forwarded-Port %[dst_port]
  http-request add-header X-Forwarded-Proto https if { ssl_fc }
  option httpchk HEAD / HTTP/1.1\r\nHost:localhost
  server Server_aws_1 IP AWS 1:8080 check
  server Server_aws_2 IP AWS 2:8080 check
```

- *balance* : especifica la estrategia de balanceo de carga. En nuestro caso hemos utilizado la estrategia *roundrobin*, la cual utiliza cada servidor a su vez, pero permite asignar pesos a cada servidor (los servidores con pesos más altos se usan con mayor frecuencia).
- *forwardfor* : este parámetro garantiza que la solicitud reenviada incluya la dirección IP del cliente real.
- El primer *http-request* permite que la solicitud enviada incluya el puerto de la solicitud HTTP del cliente. El segundo agrega el proto-encabezado que contiene https en caso de que la variable *ssl\_fc* (variable del sistema HAProxy) devuelve verdadero.
- *httpchk* : define la comprobación que HAProxy utiliza para probar si un servidor web sigue siendo válido para reenviar solicitudes. Si el servidor no responde a la solicitud definida, no se utilizará para el equilibrio de carga hasta que pase la prueba.
- *server* : estas líneas definen los nodos reales del servidor y sus direcciones IP, a las cuales se reenviarán las direcciones IP. En nuestro caso hemos puesto 2 nodos, ya que tenemos dos servidores, donde cada uno utilizará la comprobación de estado que ha definido.

Por otra parte tenemos el nodo *stats*. Este nodo puede ser utilizado para monitorear el balanceador de carga y los nodos frontend y backend.

```
listen stats
bind :32700
stats enable
stats uri /
stats hide-version
stats auth user:password
```

En este nodo definimos que escuche por el puerto 32700 (podemos definir otro puerto el cual esté libre) para conexiones además de estar configurado para esconder la versión de HAProxy.

Con el parámetro *auth* definimos un usuario con contraseña para poder hacer login en el monitor del servicio utilizando el puerto que hemos definido.

**HAProxy**

**Statistics Report for pid 967**

> General process information

pid = 967 (process #1, nbproc = 1, hltthread = 2)  
 uptime = 52.2450m37s  
 system limit: memmax = unlimited; ulimit = 1023  
 maxsock = 1023; maxconn = 400; maxpipes = 0  
 current conn = 1; current pipe = 0; conn rate = 1/soc; b/s rate = 0.271 kbps  
 Running task: 1/12; idle = 100 %

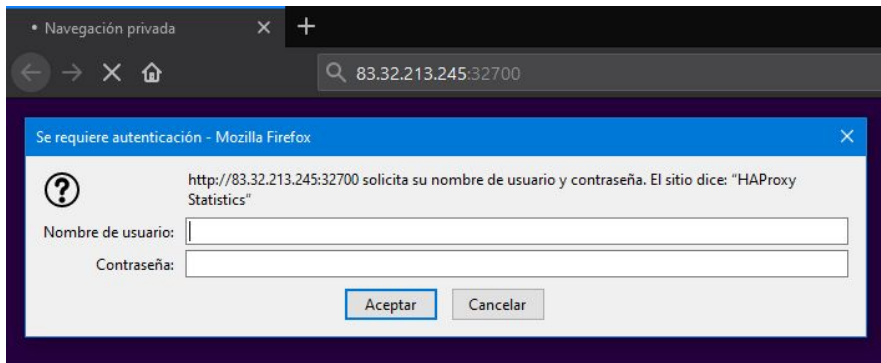
Legend: active UP, backup UP, active UP going down, backup UP going down, active DOWN, going up, backup DOWN, going up, active or backup DOWN, not checked, active or backup DOWN for maintenance (MAINT), active or backup SOFT STOPPED for maintenance, Note: "NOUP/DRAIN" = UP with loadbalancing disabled

| haproxysocks |       | Sessions |     |       |     |     |       |     |     |       |       | Errors |      |     |     | Warnings |      |     |      | Status |      |      |      |      |        |         |      |     |      |     |      |          |        |
|--------------|-------|----------|-----|-------|-----|-----|-------|-----|-----|-------|-------|--------|------|-----|-----|----------|------|-----|------|--------|------|------|------|------|--------|---------|------|-----|------|-----|------|----------|--------|
| Frontend     | Queue | Cur      | Max | Limit | Cur | Max | Limit | Cur | Max | Limit | Total | 1/30s  | 1/1m | 1/s | Out | Req      | Resp | Req | Conn | Resp   | Rate | Rate | Rate | Rate | Status | LastChk | Wght | Act | Back | Chk | Down | Overtime | Health |
| Frontend     | 0     | 0        | 1   | 0     | 0   | 0   | 0     | 0   | 0   | 0     | 490   | 0      | 0    | 0   | 699 | 964      | 0    | 0   | 0    | 0      | 0    | 0    | 0    | 0    | OPEN   |         |      |     |      |     |      |          |        |

| backendsocks |       | Sessions |     |       |     |     |       |     |     |       |       | Errors |       |     |     | Warnings |      |     |      | Status |      |      |      |      |         |                    |      |     |      |     |      |          |        |
|--------------|-------|----------|-----|-------|-----|-----|-------|-----|-----|-------|-------|--------|-------|-----|-----|----------|------|-----|------|--------|------|------|------|------|---------|--------------------|------|-----|------|-----|------|----------|--------|
| node1        | Queue | Cur      | Max | Limit | Cur | Max | Limit | Cur | Max | Limit | Total | 1/30s  | 1/1m  | 1/s | Out | Req      | Resp | Req | Conn | Resp   | Rate | Rate | Rate | Rate | Status  | LastChk            | Wght | Act | Back | Chk | Down | Overtime | Health |
| node1        | 0     | 0        | 0   | 0     | 1   | 0   | 1     | 0   | 1   | 0     | 2     | 2      | 1615m | 884 | 366 | 0        | 0    | 0   | 0    | 0      | 0    | 0    | 0    | 0    | 2b/s UP | L7OK/302 in 225ms  | 1    | Y   | -    | 0   | 0    | 0s       | -      |
| node2        | 0     | 0        | 0   | 0     | 1   | 0   | 1     | 0   | 1   | 0     | 1     | 1      | 1647m | 115 | 206 | 0        | 0    | 0   | 0    | 0      | 0    | 0    | 0    | 0    | 2b/s UP | *L7OK/302 in 225ms | 1    | Y   | -    | 1   | 0    | 0s       | -      |
| Backend      | 0     | 0        | 0   | 1     | 0   | 1   | 49    | 0   | 1   | 0     | 49    | 3      | 1616m | 496 | 572 | 0        | 0    | 0   | 0    | 0      | 0    | 0    | 0    | 0    | 2b/s UP |                    | 2    | Z   | 0    | 0   | 0s   | -        |        |

| stats    |       | Sessions |     |       |     |     |       |     |     |       |       | Errors |      |     |     | Warnings |        |     |      | Status |      |      |      |      |         |         |      |     |      |     |      |          |        |
|----------|-------|----------|-----|-------|-----|-----|-------|-----|-----|-------|-------|--------|------|-----|-----|----------|--------|-----|------|--------|------|------|------|------|---------|---------|------|-----|------|-----|------|----------|--------|
| Frontend | Queue | Cur      | Max | Limit | Cur | Max | Limit | Cur | Max | Limit | Total | 1/30s  | 1/1m | 1/s | Out | Req      | Resp   | Req | Conn | Resp   | Rate | Rate | Rate | Rate | Status  | LastChk | Wght | Act | Back | Chk | Down | Overtime | Health |
| Frontend | 0     | 0        | 1   | 0     | 0   | 0   | 0     | 0   | 0   | 0     | 49    | 0      | 0    | 0   | 0   | 2 099    | 94 524 | 0   | 0    | 0      | 0    | 0    | 0    | 0    | OPEN    |         |      |     |      |     |      |          |        |
| Backend  | 0     | 0        | 0   | 0     | 0   | 0   | 0     | 0   | 0   | 0     | 49    | 0      | 0    | 0   | 0   | 2 099    | 94 524 | 0   | 0    | 0      | 0    | 0    | 0    | 0    | 2b/s UP |         | 0    | 0   | 0    | 0   | 0    | 0s       | -      |

Para entrar al monitor de haproxy simplemente debemos acceder al navegador con la IP donde esta alojado el servicio con el puerto que hemos definido en la configuración, y nos debería dejar hacer login para poder acceder.



Con esta configuración ya tendríamos el balanceador de carga configurado y listo para funcionar.

## CONFIGURACIÓN INSTANCIAS AWS

Para configurar nuestra instancia AWS, debemos estar informados de los Tier que nos ofrecen. En esta configuración utilizamos la capa gratuita ([free tier](#)).

Empezamos creando la instancia desde el panel de control y nos aseguramos que seleccionamos las diferentes prestaciones de la capa gratuita.

Click the Launch Instance button to start your own server.

Launch Instance

A continuación podremos escoger diferentes sistemas operativos, pero cuidado, porque no todos son gratuitos, por lo tanto activamos la casilla *free tier*.

Free tier only ⓘ

En esta configuración utilizaremos el S.O *Ubuntu Server 18.04 LTS* de 64 bits.

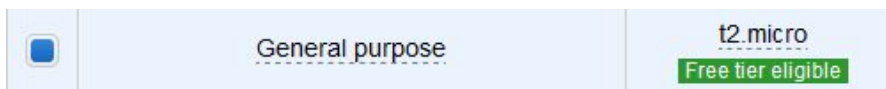
 **Ubuntu Server 18.04 LTS (HVM), SSD Volume Type** - ami-07c1207a9d40bc3bd (64-bit x86) / ami-0a5ee0336de62011b (64-bit Arm)

Free tier eligible

Ubuntu Server 18.04 LTS (HVM),EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Root device type: ebs    Virtualization type: hvm    ENA Enabled: Yes

Ahora nos pedirá escoger el tier que queremos utilizar. No nos olvidamos, capa gratuita.



Como la capa gratuita nos permite almacenar hasta 30 GB, porque no aprovechar al máximo.

| Volume Type ⓘ | Device ⓘ  | Snapshot ⓘ             | Size (GiB) ⓘ | Volume Type ⓘ               |
|---------------|-----------|------------------------|--------------|-----------------------------|
| Root          | /dev/sda1 | snap-0d2a2e7855e80a405 | 30           | General Purpose SSD (gp2) ▾ |

Para poder acceder a nuestra instancia de AWS configuraremos el grupo de seguridad para que cualquiera pueda entrar de momento. Más adelante podemos cambiar el grupo y por lo tanto, reforzar la seguridad de nuestros servidores.

| Type ⓘ | Protocol ⓘ | Port Range ⓘ | Source ⓘ                |
|--------|------------|--------------|-------------------------|
| SSH ▾  | TCP        | 22           | Anywhere ▾ 0.0.0.0, ::0 |

Por último paso debemos configurar la clave SSH para acceder a nuestra instancia.


Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair ▼

**Key pair name**

ssh\_aws

**Download Key Pair**

 You have to download the **private key file** (\*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

---

**Cancel** **Launch Instances**

## INSTALACIÓN PRESTASHOP

Para empezar a instalar PrestaShop, nos conectaremos mediante SSH a nuestra instancia AWS con la key ssh que descargamos al crear la instancia anteriormente.

Una vez conectado actualizaremos la máquina.

```
sudo apt update && apt upgrade
```

Después de actualizar nuestra máquina procedemos a instalar el paquete LAMP.

```
sudo apt install apache2 libapache2-mod-php mysql-server
sudo apt install php7.2-cli php7.2-common php7.2-curl php7.2-zip php7.2-gd php7.2-mysql
php7.2-xml php7.2-mbstring php7.2-json php7.2-intl
```

Al instalar el paquete LAMP debemos activar el módulo Apache, así que ejecutaremos el siguiente comando:

```
sudo a2enmod write
```

Reiniciamos el servicio Apache para que surtan los efectos.

```
sudo systemctl restart apache2
```

Seguidamente crearemos la base de datos para PrestaShop. Para ello nos conectamos a mysql con el usuario root y una vez dentro creamos la base de datos.

```
mysql -u root -p
CREATE DATABASE prestashop;
```

Añadiremos un usuario exclusivo para interactuar con la base de datos que acabamos de crear.

```
GRANT ALL PRIVILEGES ON prestashop.* to 'prestashop'@'localhost' IDENTIFIED BY
'5tr0ng_Pa55w0rD';
```

Y refrescamos los privilegios para que se apliquen los cambios.

```
FLUSH PRIVILEGES;
```

Una vez creada la base de datos con su usuario ya podemos instalar PrestaShop. Primero empezaremos descargando la última versión estable de este.

```
wget https://download.prestashop.com/download/releases/prestashop_1.7.6.5.zip
```

Y lo descomprimos en el directorio de apache

```
unzip prestashop_1.7.6.5.zip
```

```
unzip prestashop_1.7.6.5.zip -d /var/www/html/prestashop/
```

Estableceremos los permisos y la propiedad de los archivos apropiados (en nuestro caso es el usuario y el grupo www-data porque estamos utilizando el servidor web Apache en Ubuntu).

```
chown -R www-data: /var/www/html/prestashop/
```

Por casi último paso, nos queda configurar Apache. Tenemos que crear un nuevo host virtual en el servidor web para que pueda servir el directorio PrestaShop. Para ello creamos el fichero de configuración en su correspondiente ruta :

```
sudo nano /etc/apache2/sites-available/prestashop.conf
```

```
<VirtualHost *:80>
  ServerAdmin admin@rebuildcars.duckdns.org
  ServerName rebuildcars.duckdns.org
  ServerAlias rebuildcars.duckdns.org
  DocumentRoot /var/www/html/prestashop

  <Directory /var/www/html/prestashop>
    Options +FollowSymlinks
    AllowOverride All
    Require all granted
  </Directory>

  ErrorLog /var/log/apache2/prestashop-error_log
  CustomLog /var/log/apache2/prestashop-access_log common
</VirtualHost>
```

Después de configurar el archivo de apache de PrestaShop, activamos dicho archivo :

```
sudo a2ensite prestashop.conf
```

También borraremos el archivo de configuración por defecto de Apache :

```
sudo rm /etc/apache2/sites-enabled/000-default.conf
```

Una vez hecho todos estos pasos, reiniciamos el servicio Apache para que se realicen los cambios y ya podremos proceder a la instalación en la web de PrestaShop. Accederemos por nuestra IP del servidor donde acabamos de instalar PrestaShop

Cuando hayamos accedido, nos saldrá un asistente para proceder con la instalación.

Foro | Soporte | Documentación | Blog

## Asistente de instalación ○ ○ ○ ○ ○ ○

▶ **Selección de idioma**

Acuerdos de licencia

Compatibilidad del sistema

Información de la tienda

Configuración del sistema

Instalación de la tienda

### Bienvenido al instalador de PrestaShop 1.7.6.5

Instalar PrestaShop es rápido y fácil. En tan solo unos minutos, puedes formar parte de una comunidad formada por más de 250.000 comerciantes. Estás en el camino de crear tu propia tienda en línea única que puedes gestionar fácilmente cada día.

Si necesita ayuda no dude en [ver este corto tutorial](#), o [consulte nuestra documentación](#).

**Continuar el proceso de instalación en:**

Español (Spanish) ▼

La anterior selección de idioma afecta únicamente al Asistente de Instalación. Cuando su tienda se encuentre instalada, podrá elegir el idioma de su tienda entre más de 60 traducciones disponibles, ¡todos ellos gratuitos!

[Siguiente](#)

Si necesita ayuda, puede [solicitar ayuda](#) a nuestro equipo de soporte. Además, [la documentación oficial](#) está aquí para guiarle.

**¿Necesita ayuda?**

[Obtenga un plan de soporte técnico](#)

Foro | Soporte | Documentación | Blog

## Asistente de instalación ✔ ○ ○ ○ ○ ○

✔ Selección de idioma

▶ **Acuerdos de licencia**

Compatibilidad del sistema

Información de la tienda

Configuración del sistema

Instalación de la tienda

### Términos y condiciones del acuerdo de licencia

Para disfrutar de las numerosas funcionalidades ofrecidas de forma gratuita por PrestaShop, por favor, lee los términos y condiciones de la licencia descritos a continuación. El núcleo de PrestaShop está disponible bajo la licencia OSL 3.0, mientras que los módulos y los temas están licenciados bajo la licencia AFL 3.0.

**Core: Open Software License ("OSL") v. 3.0**

This Open Software License (the "License") applies to any original work of authorship (the "Original Work") whose owner (the "Licensor") has placed the following licensing notice adjacent to the copyright notice for the Original Work:

**Licensed under the Open Software License version 3.0**

**1. Grant of Copyright License.** Licensor grants You a worldwide, royalty-free, non-exclusive, sublicensable license, for the duration of the copyright, to do the following:

**Acepto los términos y condiciones arriba indicados.**

[Siguiente](#)

[Atrás](#)

Si necesita ayuda, puede [solicitar ayuda](#) a nuestro equipo de soporte. Además, [la documentación oficial](#) está aquí para guiarle.

**¿Necesita ayuda?**

[Obtenga un plan de soporte técnico](#)



## Asistente de instalación




- ✓ Selección de idioma
- ✓ Acuerdos de licencia
- ▶ **Compatibilidad del sistema**
- Información de la tienda
- Configuración del sistema
- Instalación de la tienda



### Estamos comprobando la compatibilidad de PrestaShop con el entorno del sistema

Si tiene alguna pregunta, por favor, visita nuestra [documentación](#) y [nuestro foro de la comunidad](#).

✓ **¡La compatibilidad de PrestaShop con el entorno del sistema ha sido verificada correctamente!**

To get the latest internationalization data upgrade the ICU system package and the intl PHP extension. 

Actualizar

Atrás


Siguiente

Si necesita ayuda, puede [solicitar ayuda](#) a nuestro equipo de soporte. Además, [la documentación oficial](#) está aquí para guiarle.



Este paso es importante porque, estamos definiendo el nombre que va a tener nuestra tienda y la actividad principal, para así poder tener una orientación óptima y mejoras funcionales por parte de PrestaShop.


Además de definir lo anterior, estamos creando una cuenta administrador, lo cual quiere decir que pondremos una contraseña bastante fuerte y sobretodo que no se nos olvide.



[Foro](#) | [Soporte](#) | [Documentación](#) | [Blog](#)

## Asistente de instalación ✔ ✔ ✔

- ✔ Selección de idioma
- ✔ Acuerdos de licencia
- ✔ Compatibilidad del sistema
- ▶ **Información de la tienda**
- Configuración del sistema
- Instalación de la tienda



### Información sobre tu tienda

Nombre de la tienda  \*

Actividad principal  Ayúdanos a aprender más acerca de su tienda, ¡para que le podamos ofrecer una orientación óptima y mejoras funcionales para su negocio!

País  \*

### Tu cuenta

Nombre  \*

Apellidos  \*

Dirección de correo electrónico  \* Esta dirección de correo electrónico corresponderá a tu usuario en el acceso al interfaz de administración de tu tienda Online.

Contraseña de la tienda  \* Mínimo 8 caracteres

Confirmar la contraseña  \*

Todos los datos recopilados podrán ser tratados y utilizados con fines estadísticos. Tus datos personales podrán comunicarse a proveedores de servicios y socios comerciales. En virtud de la "Ley de Procesamiento de Datos, Archivos de Datos y Libertades Individuales" en vigor, puedes ejercer los derechos de acceso, rectificación y oposición al tratamiento de tus datos personales a través del siguiente [link](#).

Atrás
Siguiente

En el siguiente paso debemos configurar la conexión a la base de datos, la cual hemos creado anteriormente. Le indicamos la dirección IP donde se encuentra la base de datos, que en nuestro caso es local, y le decimos el nombre y usuario de esta, además de la contraseña para que PrestaShop pueda acceder a ella.

Foro | Soporte | Documentación | Blog

## Asistente de instalación

✓ Selección de idioma  
✓ Acuerdos de licencia  
✓ Compatibilidad del sistema  
✓ Información de la tienda  
▶ **Configuración del sistema**  
Instalación de la tienda

**¿Necesita ayuda?**  
Obtenga un plan de soporte técnico

**Configura la conexión a la base de datos rellenando los siguientes campos**

Para utilizar PrestaShop, debes **crear una base de datos** para guardar todos los datos relacionados con las actividades de tu tienda.  
Por favor, rellena los siguientes campos para conectar PrestaShop a tu base de datos.

Dirección del servidor de la base de datos   
El puerto predeterminado es el 3306. Si utilizas un puerto diferente, añade el número del puerto al final de la dirección del tu servidor, por ejemplo ":4242".

Nombre de la base de datos

Usuario de la base de datos

Contraseña de la base de datos

Prefijo de las tablas

**¡Comprobar la conexión con tu base de datos!**

✓ La base de datos está conectada

Si necesita ayuda, puede **solicitar ayuda** a nuestro equipo de soporte. Además, la **documentación oficial** está aquí para guiarle.

Foro oficial | PrestaShop.com | Soporte | Documentación | Contacte con nosotros | © 2007-2020

[Foro](#) | [Soporte](#) | [Documentación](#) | [Blog](#)

## Asistente de instalación

- ✓ Selección de idioma
- ✓ Acuerdos de licencia
- ✓ Compatibilidad del sistema
- ✓ Información de la tienda
- ✓ Configuración del sistema
- ▶ **Instalación de la tienda**

**¿Necesita ayuda?**  
Obtenga un plan de soporte técnico

### ¡La instalación ha finalizado!

El proceso de instalación de tu tienda ha finalizado. ¡Gracias por utilizar PrestaShop!

Por favor, recuerda tus datos de acceso a la tienda:

Dirección de correo electrónico: rebuildcars20@gmail.com [Imprimir mis datos de acceso a la tienda](#)

Contraseña: \*\*\*\*\* (Mostrar)

Por razones de seguridad, debe eliminar la carpeta "install". [?](#)

#### Interfaz de administración (Back Office)

*Administra tu tienda utilizando la interfaz de administración. Gestiona tus pedidos y clientes, añade módulos, cambia el tema predeterminado, etc.*

Administra tu tienda

#### Interfaz de usuario (Front Office)

*¡Descubre tu tienda tal y cómo la verán tus clientes!*

Visita tu tienda

¡Comparte tu experiencia con tus amigos!

Tuitear

Compartir

Google+

Pinterest

LinkedIn

Después de completar el asistente de instalación, debemos eliminar el archivo install de prestashop.

```
sudo rm -rf /var/www/html/prestashop/install/
```

## DOMINIO + SSL

En este proyecto hemos implementado el dominio utilizando *'duckdns.org'*, ya que la máquina que redirige a los servidores AWS tiene IP pública dinámica.

Para implementar el dominio debemos crear una carpeta donde vamos a crear un script el cual se va a ejecutar cada cierto tiempo mediante crontab.

Teniendo registrado nuestro dominio, nos darán un comando el cual debemos introducir en el script que va dentro de la carpeta que hemos creado. El comando es el siguiente :

```
echo
url="https://www.duckdns.org/update?domains=rebuildcars&token=61360c9a-2f88-48b2-89d4-47e890cd6dd9&ip=" | curl -k -o ~/duckdns/duck.log -K -
```

Cambiamos los permisos al script : `chmod 700 duck.sh`

Y ejecutamos crontab -e para hacer que este script se ejecute cada 5 minutos. Si queremos podemos cambiar el tiempo.

```
# m h dom mon dow   command
*/5 * * * * ~/duckdns/duck.sh >/dev/null 2>&1
```

Por último ejecutamos el script manualmente y comprobamos que funciona correctamente. Si da cualquier fallo, el script está configurado para que los errores vayan a un fichero .log.

```
awsbalancer@awsbalancer:~/duckdns$ cat duck.log
OKawsbalancer@awsbalancer:~/duckdns$
```

Vemos que no hay ningún fallo en el script.

Una vez configurado el dominio pasamos al apartado de crear el certificado SSL, para ello empezamos ejecutando el siguiente comando :

```
sudo openssl req -new /etc/ssl/prestashop/prestashop.key -out /etc/ssl/prestashop/prestashop.crt
```

Nos pedirá información la cual debemos rellenar conforme los datos que tiene nuestra página web.

```
awsbalancer@awsbalancer:~$ sudo openssl req -new -key /etc/ssl/xip.io/xip.io.key -out /etc/ssl/xip.io/xip.io.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]:ES
State or Province Name (full name) [Some-State]:Barcelona
Locality Name (eg, city) []:Barcelona
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Rebuild Cars
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:rebuildcars.duckdns.org
Email Address []:rebuildcars20@gmail.com

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

Una vez creado el certificado vamos a darle una fecha de expiración :

```
sudo openssl x509 -req -days 365 -in /etc/ssl/prestashop/prestashop.crt \
-signkey /etc/ssl/prestashop/prestashop.key \
-out /etc/ssl/prestashop/prestashop.crt
```

```
awsbalancer@awsbalancer:~$ sudo openssl x509 -req -days 365 -in /etc/ssl/xip.io/xip.io.csr \
-signkey /etc/ssl/xip.io/xip.io.key \
-out /etc/ssl/xip.io/xip.io.crt
Signature ok
subject=C = ES, ST = Barcelona, L = Barcelona, O = Rebuild Cars, CN = rebuildcars.duckdns.org, emailAddress = rebuildcars20@gmail.com
Getting Private key
```

Para exportar la key a las máquinas requeridas debemos ejecutar un par de comandos. Nos pedirá una contraseña de exportación por seguridad :

```
openssl pkcs12 -export -out SSL247Backup.pfx -inkey yourprivatekey.txt -in yourSSLCertificate.crt
-certfile intermediario.crt
```

```
awsbalancer@awsbalancer: /etc/ssl/prestashop$ sudo openssl pkcs12 -export -out SSL247Backup.pfx -inkey prestashop.key -in
prestashop.crt -certfile prestashop.crt
Enter Export Password:
Verifying - Enter Export Password:
```

```
awsbalancer@awsbalancer: /etc/ssl/prestashop$ ls
prestashop.crt  prestashop.csr  prestashop.key  prestashop.pem  SSL247Backup.pfx
```

Cuando tengamos este fichero creado nos vamos a las máquinas donde queremos implementar SSL para importar este fichero el cual contiene la clave del certificado.

Para descargar dicha clave, con el comando scp nos servirá :

```
scp -P 37805 awsbalancer@83.32.213.245:/etc/ssl/prestashop/SSL247Backup.pfx /home/ubuntu/
```

```
ubuntu@ip-172-31-16-82: $ scp -P 37805 awsbalancer@83.32.213.245:/etc/ssl/prestashop/SSL247Backup.pfx /home/ubuntu/
awsbalancer@83.32.213.245's password:
SSL247Backup.pfx 100% 3549 29.1KB/s 00:00
```

Una vez descargado, para importarlo necesitamos ejecutar el siguiente comando :

```
openssl pkcs12 -in SSL247Backup.pfx -out prestashop.key -nodes
```

```
ubuntu@ip-172-31-16-82: /etc/ssl/prestashop$ sudo openssl pkcs12 -in SSL247Backup.pfx -out prestashop.key -nodes
Enter Import Password:
ubuntu@ip-172-31-16-82: /etc/ssl/prestashop$ ls
SSL247Backup.pfx  prestashop.key
```

Para los demás ficheros seguiremos utilizando el comando scp.

```
ubuntu@ip-172-31-16-82: /etc/ssl/prestashop$ sudo scp -P 37805 awsbalancer@83.32.213.245:/etc/ssl/prestashop/prestashop.crt /etc/ssl/prestashop/
awsbalancer@83.32.213.245's password:
prestashop.crt 100% 1346 11.2KB/s 00:00
ubuntu@ip-172-31-16-82: /etc/ssl/prestashop$ sudo scp -P 37805 awsbalancer@83.32.213.245:/etc/ssl/prestashop/prestashop.pem /etc/ssl/prestashop/
awsbalancer@83.32.213.245's password:
prestashop.pem 100% 3025 25.3KB/s 00:00
```

Una vez exportado los ficheros necesarios para implementar SSL (.crt, .key, .pem) a las máquinas donde se aloja PrestaShop, debemos activar dicho protocolo con el siguiente comando :

```
sudo a2enmod ssl
```

Una vez activado debemos ir al archivo de configuración de apache e activar el puerto 443 para que escuche por este, además de indicarle la ruta donde se encuentran los ficheros del certificado SSL.

Dentro del archivo de configuración tenemos que crear un apartado nuevo con el puerto 443 e indicarle donde se encuentran los ficheros del certificado (.crt, .key, .pem)

```
<VirtualHost *:443>
    ServerAdmin admin@18.191.218.27
    ServerName 18.191.218.27
    ServerAlias rebuildcars.duckdns.org
    DocumentRoot /var/www/html/prestashop

    <Directory /var/www/html/prestashop>
        Options +FollowSymlinks
        AllowOverride All
        Require all granted
    </Directory>

    SSLEngine on
    SSLProtocol all
    SSLCertificateFile /etc/ssl/prestashop/prestashop.crt
    SSLCertificateKeyFile /etc/ssl/prestashop/prestashop.key
    SSLCACertificateFile /etc/ssl/prestashop/prestashop.pem


    ErrorLog /var/log/apache2/prestashop-error_log
    CustomLog /var/log/apache2/prestashop-access_log common
</VirtualHost>
```



Reiniciamos apache para que se apliquen los cambios y nos vamos a PrestaShop para activar SSL en la página web.


Nos iremos al apartado de : **Parámetros de la tienda -> Configuración**, y comprobaremos si nuestra página web soporta HTTPS.

## Configuración

Activar SSL 

Por favor, haz clic aquí para comprobar si tu tienda soporta HTTPS.

Nos saldrán un par de avisos. Aceptamos y entendemos los riesgos y continuamos.



### Advertencia: riesgo potencial de seguridad a continuación

Firefox ha detectado una posible amenaza de seguridad y no ha cargado 3.22.221.201. Si visita este sitio, los atacantes podrían intentar robar información como sus contraseñas, correos electrónicos o detalles de su tarjeta de crédito.

[Más información...](#)

Retroceder (recomendado)
Avanzado...

Los sitios web prueban su identidad mediante certificados. Firefox no confía en este sitio porque utiliza un certificado que no es válido para 3.22.221.201.

Código de error: MOZILLA\_PKIX\_ERROR\_SELF\_SIGNED\_CERT

[Ver certificado](#)

Retroceder (recomendado)
Aceptar el riesgo y continuar

 **Token no válido:** el acceso directo a este enlace puede ocasionar una violación de seguridad potencial.

**¿Deseas mostrar esta página?**

Sí, entiendo los riesgos

¡Salir de aquí!

Si nuestra página web soporta HTTPS nos dejará activar SSL. Aprovechamos y activamos SSL en todas las páginas, ya que queremos que nuestra página web sea segura en cualquier parte.

## Configuración

Activar SSL 

NO  SÍ

Si dispones un certificado SSL propio para el dominio de tu tienda, puedes activar el cifrado SSL (https://) para la identificación de las cuentas de cliente y el proceso de compra.

Activar SSL en todas las páginas

NO  SÍ

Cuando está activada, todas las páginas de tu tienda estarán protegidas por SSL.

Una vez activado podemos ver el certificado :

## Certificado

[rebuildcars.duckdns.org](https://rebuildcars.duckdns.org)

---

|                                        |                         |
|----------------------------------------|-------------------------|
| <b>Nombre del asunto</b>               | _____                   |
| <b>País</b>                            | ES                      |
| <b>Estado/Provincia</b>                | Barcelona               |
| <b>Localidad</b>                       | Barcelona               |
| <b>Organización</b>                    | Rebuild Cars            |
| <b>Nombre común</b>                    | rebuildcars.duckdns.org |
| <b>Dirección de correo electrónico</b> | rebuildcars20@gmail.com |

---

|                                        |                         |
|----------------------------------------|-------------------------|
| <b>Nombre del emisor</b>               | _____                   |
| <b>País</b>                            | ES                      |
| <b>Estado/Provincia</b>                | Barcelona               |
| <b>Localidad</b>                       | Barcelona               |
| <b>Organización</b>                    | Rebuild Cars            |
| <b>Nombre común</b>                    | rebuildcars.duckdns.org |
| <b>Dirección de correo electrónico</b> | rebuildcars20@gmail.com |

---

|                   |                                                      |
|-------------------|------------------------------------------------------|
| <b>Validez</b>    | _____                                                |
| <b>No antes</b>   | 22/5/2020 0:05:10 (hora de verano de Europa central) |
| <b>No después</b> | 22/5/2021 0:05:10 (hora de verano de Europa central) |

Si queremos asegurarnos de que nuestra máquina tiene SSL activado, con esta herramienta muy útil podemos hacer un check → <https://decoder.link/sslchecker/>

Simplemente debemos indicarle la IP o dominio de la máquina y nos dice toda la información del certificado.

Hostname\*  
rebuildcars.duckdns.org

Port\*  
443

CHECK

---

**Report**

|                         |                                                                                      |
|-------------------------|--------------------------------------------------------------------------------------|
| <b>Hostname:</b>        | ✓ Matches Common Name or/and SAN                                                     |
| <b>Expired:</b>         | ✓ No (363 days till expiration)                                                      |
| <b>Public Key:</b>      | ✓ We were unable to find any issues in the public key of end-entity certificate      |
| <b>Trusted:</b>         | ✗ The end-entity certificate is self-signed                                          |
| <b>Self-Signed:</b>     | ✗ Yes, the end-entity certificate is self-signed                                     |
| <b>Chain Issues:</b>    | ✗ The chain doesn't contain any intermediate certificates                            |
| <b>Weak signatures:</b> | ✓ No, certificates sent by the server were not signed utilizing a weak hash function |

## RÉPLICA ENTRE 2 BASE DE DATOS

Para hacer la réplica entre las 2 base de datos que tenemos (una en cada servidor donde está alojado PrestaShop) hemos utilizado MySQL. Esto es así como dos servidores 'Master - Slave'.

### MASTER

Para poder hacer esto, necesitamos obviamente configurarlo, y para ello nos iremos al archivo de configuración de mysql : `/etc/mysql/mysql.conf.d/mysqld.conf`

Debemos cambiar unos cuantos parámetros, empezaremos cambiando la línea 'bind-address' por la siguiente IP :

```
bind-address = 0.0.0.0
```

Esto hará que permita conexiones entrantes para así más tarde hacer la réplica.

Seguidamente para poder hacer la réplica debemos indicar una id a cada servidor, en este caso estamos configurando el servidor el cual va a ofrecer la réplica, por lo tanto le ponemos '1'.

```
server-id = 1
log_bin = /var/log/mysql/mysql-bin.log
log_bin_index = /var/log/mysql/mysql-bin.log.index
relay_log = /var/log/mysql/mysql-relay-bin
relay_log_index = /var/log/mysql/mysql-relay-bin.index
```

Después de modificar el archivo de configuración, crearemos un usuario con permisos específicos para que pueda hacer la réplica :

```
mysql > create user 'replication_user'@'%' identified by 'PASSWORD';
mysql > GRANT REPLICATION SLAVE ON *.* TO 'replication_user'@'%' ;
mysql > FLUSH PRIVILEGES;
```

Antes de ir al otro servidor a configurarlo, ejecutamos el siguiente comando, ya que necesitaremos esta información para poder realizar la réplica :

```
mysql> show master status \g
+-----+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB | Executed_Gtid_Set |
+-----+-----+-----+-----+-----+
| mysql-bin.000125 | 15135252 | prestashop   |                    |                    |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Teniendo esta información en mano, ya podemos ir al otro servidor.



## SLAVE

Para el segundo servidor debemos hacer lo mismo, ir al archivo de configuración y editar los siguientes parámetros :

```
bind-address = 0.0.0.0
```

```
server-id = 2
log_bin = /var/log/mysql/mysql-bin.log
log_bin_index = /var/log/mysql/mysql-bin.log.index
relay_log = /var/log/mysql/mysql-relay-bin
relay_log_index = /var/log/mysql/mysql-relay-bin.index
```

Estando en el otro servidor le pondremos otra id para diferenciarlos.

A continuación cambiaremos la configuración Master-Slave. Para hacer esto necesitamos la información que hemos advertido antes de tener en mano (la del comando `'show master status\g`).

Antes de ejecutar cualquier comando, hemos de parar el *slave*. Es muy simple :

```
mysql> stop slave;
```

Una vez parado, ahora ya podemos ejecutar el verdadero comando que nos va ayudar a realizar la réplica:

```
mysql> CHANGE MASTER TO MASTER_HOST = '18.191.248.27', MASTER_USER =
'replication_user', MASTER_PASSWORD = 'PASSWORD', MASTER_LOG_FILE =
'mysql-bin.000125', MASTER_LOG_POS = 15135252;
mysql > start slave;
mysql > Quit;
```

Debemos indicarle al comando la IP del servidor *'Master'* (servidor con la id-1) además de el usuario con su contraseña, el cual hemos creado anteriormente específicamente para hacer esto, el log del fichero *mysql.bin*. y la posición en la que se encuentra.

Lo que hacemos aquí es cambiar el *master\_host* al servidor con id 1 (configurado previamente) para que cuando haya algún cambio los replique en este servidor.

---

## OwnCloud (BACKUPS)

Hemos escogido OwnCloud para realizar copias de seguridad de la base de datos de PrestaShop.

OwnCloud está hecho en PHP y se accede normalmente por interfaz web, así que necesitamos tener instalado; apache + PHP + Módulos PHP necesarios para OwnCloud.

```
sudo apt install apache2 libapache2-mod-php7.2 openssl php-imagick php7.2-common php7.2-curl  
php7.2-gd php7.2-imap php7.2-intl php7.2-json php7.2-ldap php7.2-mbstring php7.2-mysql php7.2-pgsql  
php-smbclient php-ssh2 php7.2-sqlite3 php7.2-xml php7.2-zip
```

Cuando tengamos instalado lo necesario debemos crear una base de datos para almacenar ficheros durante y después de la instalación. Para ello nos conectamos a mysql y creamos dicha base de datos :

```
CREATE DATABASE owncloud_db;  
GRANT ALL ON owncloud_db.* TO 'owncloud_user'@'localhost' IDENTIFIED BY 'StrongP@ssword';  
FLUSH PRIVILEGES;  
EXIT;
```

Una vez creada la base de datos y dado permisos a un usuario específico para hacer las copias de seguridad en owncloud, procedemos a descargar owncloud.

```
sudo wget https://download.owncloud.org/community/owncloud-10.4.0.zip
```

Después de descargar, descomprimos el zip en la siguiente ruta :

```
sudo unzip owncloud-10.4.0.zip -d /var/www/
```

Cuanto esté descomprimido, cambiamos los permisos a los necesarios :

```
sudo chown -R www-data:www-data /var/www/owncloud/  
sudo chmod -R 755 /var/www/owncloud/
```

A continuación configuraremos apache para OwnCloud, para ello crearemos un fichero de configuración para dicho servicio.

```
sudo nano /etc/apache2/conf-available/owncloud.conf
```

Deberá contener la siguiente configuración :

```
Alias /owncloud "/var/www/owncloud/"

<Directory /var/www/owncloud/>
  Options +FollowSymlinks
  AllowOverride All

  <IfModule mod_dav.c>
    Dav off
  </IfModule>

  SetEnv HOME /var/www/owncloud
  SetEnv HTTP_HOME /var/www/owncloud

</Directory>
```

Una vez creado y configurado el fichero, debemos habilitar dicha configuración además de los módulos necesarios para apache.

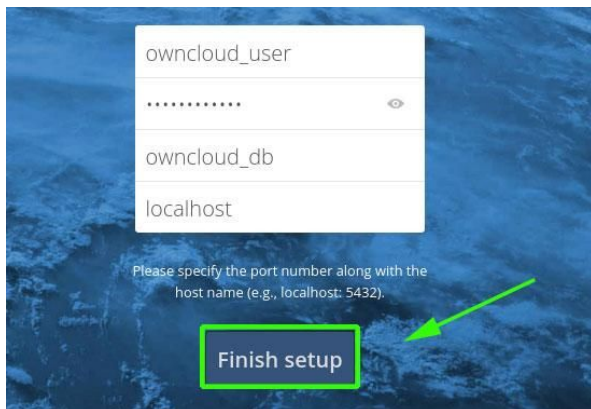
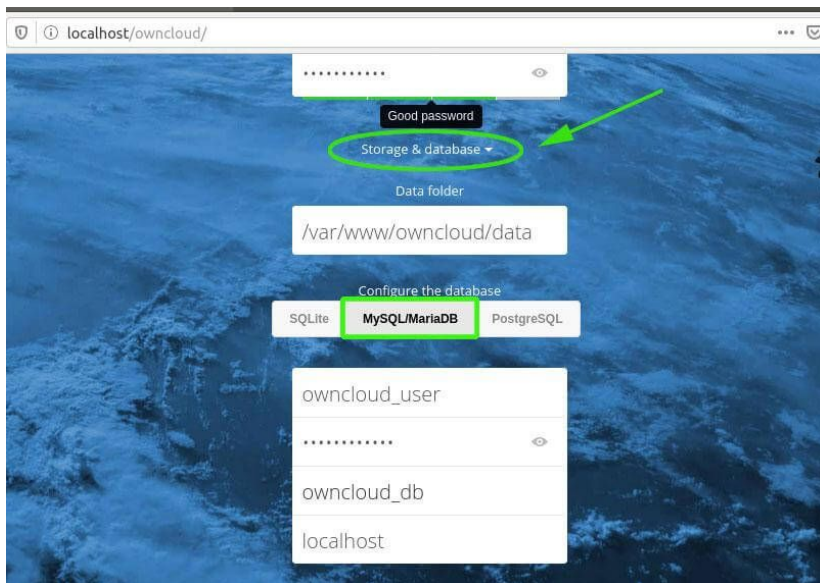
```
sudo a2enconf owncloud
sudo a2enmod rewrite
sudo a2enmod headers
sudo a2enmod env
sudo a2enmod dir
sudo a2enmod mime
```

Por casi último paso reiniciamos el servicio apache para que se realicen los cambios.

```
sudo systemctl restart apache2
```

Para finalizar accederemos a OwnCloud mediante la siguiente url : **localhost/owncloud/**

Cuando estemos dentro hemos de configurar la base de datos para que se conecte a ella.



## AUTOMATIZACIÓN DE BACKUPS

Este apartado lo hemos implementado completamente utilizando *'crontab'*, con la finalidad de no tener que estar ejecutando comandos manualmente y así tener un sistema de backups automatizado.

Para hacer esto posible debemos ejecutar el comando *'crontab -e'* y se nos abrirá un archivo de configuración en el cual se introducen las tareas que queramos que las ejecute cada cierto tiempo, ya sea diariamente, semanalmente o cada 5 minutos.

### SERVIDOR PRINCIPAL AWS

```
5 * * * * mysqldump -u prestashop -pprestashop prestashop > ~/backup/prestashop-dump.sql >/dev/null 2>&1
```

- Se conecta a mysql con el usuario *'prestashop'* con su debida contraseña a la base de datos *'prestashop'*.

```
*/30 * 1 * * * zip -r ~/backup/prestashop.zip /var/www/html/prestashop >/dev/null 2>&1
```

- Crea un *zip* recursivamente del directorio */var/www/html/prestashop*.

```
** */6 * * * * zip -r ~/mysqldump_prestashop.zip ~/backup >/dev/null 2>&1 >/dev/null 2>&1
```

- Crea un *zip* recursivamente del directorio *~/backup*.

```
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
5 * * * * mysqldump -u prestashop -pprestashop prestashop > ~/backup/prestashop-dump.sql >/dev/null 2>&1
*/30 * 1 * * * zip -r ~/backup/prestashop.zip /var/www/html/prestashop >/dev/null 2>&1 >/dev/null 2>&1
** */6 * * * * zip -r ~/mysqldump_prestashop.zip ~/backup >/dev/null 2>&1 >/dev/null 2>&1
```

### SERVIDOR OWN CLOUD

```
* * * * */6 * * * * scp -i /root/ssh_aws.pem ubuntu@18.191.248.27:/home/ubuntu/mysqldump_prestashop.zip /var/www/owncloud/data/admin/files/Backup_Prestashop/ >/dev/null 2>&1
```

- Se conecta mediante ssh utilizando una clave para obtener el *zip* creado anteriormente en la otra máquina, y lo obtiene directamente en el directorio de OwnCloud.

```
** */3 * * * * cd /var/www/owncloud && sudo -u www-data php occ files:scan --all >/dev/null 2>&1
```

- Accede al directorio de OwnCloud para poder forzar una actualización en la base de datos y así se muestren los archivos en la interfaz web.

```
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
* * * * */6 * * * * scp -i /root/ssh_aws.pem ubuntu@18.191.248.27:/home/ubuntu/mysqldump_prestashop.zip /var/www/owncloud/data/admin/files/Backup_Prestashop/ >/dev/null 2>&1
** */3 * * * * cd /var/www/owncloud && sudo -u www-data php occ files:scan --all >/dev/null 2>&1
```



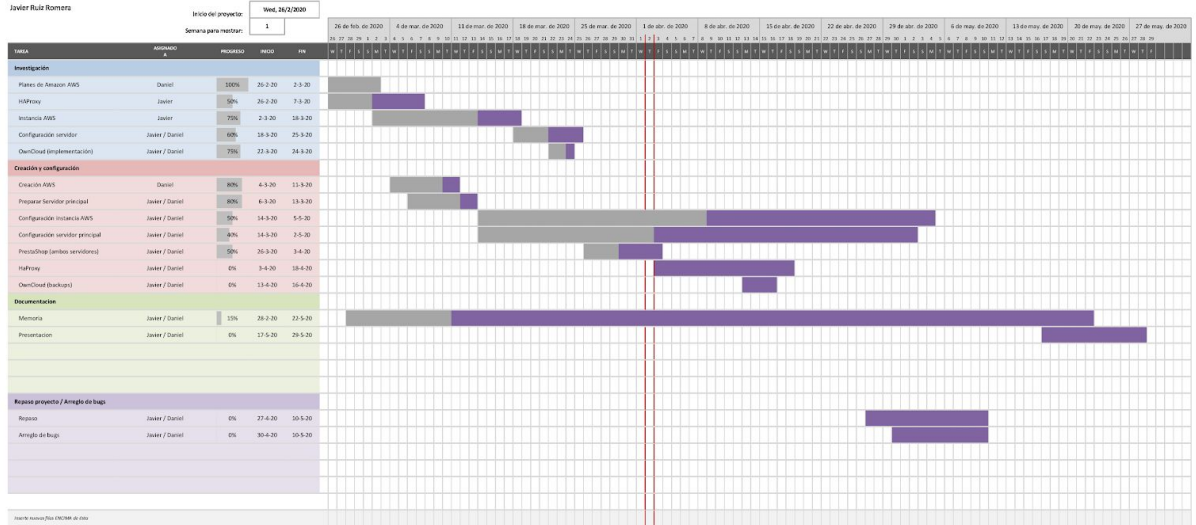
PLANIFICACIÓN

DIAGRAMA DE GANTT

Prestashop & Dual Server

Daniel Fando Moreno

Javier Ruiz Romero



**ESTUDIO ECONÓMICO Y PRESUPUESTARIO**

A continuación detallamos un presupuesto en el que calculamos los gastos iniciales en material necesario para implementar el proyecto en una empresa, además también incluimos la mensualidad del servicio.



**Dirección de la compañía**  
Calle, ciudad, provincia, código postal

**Fecha** 21/05/2020  
**N.º de presupuesto** 1

**Comentarios o instrucciones especiales:**

Presupuesto inicial sobre costes de mantenimiento y montaje de la estructura del proyecto.

| Vendedor | Número de orden de Número | Fecha de envío | PUERTO DE Punto | Condiciones         |
|----------|---------------------------|----------------|-----------------|---------------------|
|          |                           |                |                 | Pago a la recepción |

| Cantidad | Descripción                  | Precio por unidad | ¿Se le aplican impuestos? | Importe    |
|----------|------------------------------|-------------------|---------------------------|------------|
| 2        | Servidores (Instancias) AWS  | 534,36 €          | Sí                        | 1.068,72 € |
| 1        | Servidor Haproxy             | 422,00 €          | Sí                        | 422,00 €   |
| 1        | Servidor copias de seguridad | 783,00 €          | Sí                        | 783,00 €   |
|          |                              |                   |                           |            |
|          |                              |                   |                           |            |

|                           |                   |
|---------------------------|-------------------|
| Subtotal                  | 2.273,72 €        |
| Tasa impositiva           | 21,00%            |
| Impuesto sobre las ventas | 477,48 €          |
| Otros                     |                   |
| <b>TOTAL</b>              | <b>2.751,20 €</b> |

| Mensualidad | Descripción                 |
|-------------|-----------------------------|
| 2           | Servidores (Instancias) AWS |

|                           |                |
|---------------------------|----------------|
| Subtotal                  | 78,00 €        |
| Tasa impositiva           | 21,00%         |
| Impuesto sobre las ventas | 16,38 €        |
| Otros                     |                |
| <b>TOTAL</b>              | <b>94,38 €</b> |

Como vemos la inversión inicial que se necesitaría sería un total de 2.751,20€, para crear/montar los servidores necesarios que son los dos de AWS que contienen la web, 2 físicos que contiene Haproxy y otro contiene el servicio para crear y almacenar las copias de seguridad.

En el apartado de mensualidad vemos que los dos servidores saldrían por un total de 94,38€ , lo cual cada uno de los servidores cuenta con 8GB de memoria RAM, 240 GB de almacenamiento SSD, 2 vCPU, además de contar con la posibilidad de realizar una snapshot diaria de la máquina.



## CONCLUSIÓN

A la hora de empezar con este proyecto íbamos con miedo, ya que no estábamos realmente informados de cómo realizar / configurar algunos servicios. Pero mientras íbamos investigando, el miedo se nos iba yendo poco a poco además de coger seguridad en nosotros mismos y sacar este proyecto adelante. Somos inexpertos en el tema de saber cómo organizar cada cosa y que esté en su sitio (como por ejemplo el servidor HAProxy en local), pero gracias a esto tenemos la oportunidad de aprender, y lo más importante, fallar, ya que de los errores se aprende y en este caso no hay una persona detrás de todo esto la cual nos pueda despedir si ocurre algo (despedir no, pero suspender si...). Esto hace que se nos quite un poco el miedo a la hora de probar cosas nuevas, ya que no hay clientes queriendo acceder al servicio, y esto nos permite tener fallos a la hora de configurar servicios, reinicio de servidores, etc.

Además de todo esto, al ser un proyecto personal que la idea nace de nosotros, lo hacemos con un plus de apego, lo que quiere decir que aprendemos más rápido, ya que con este plus no nos importa la cantidad de errores que salgan porque vamos a poner ímpetu para solucionarlos.

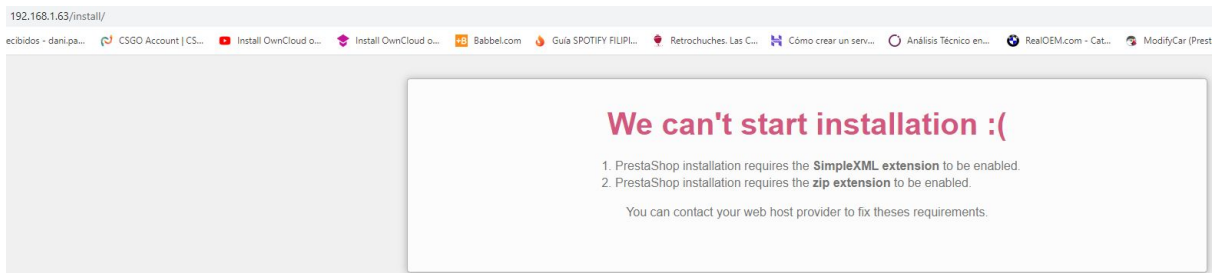
Durante este proyecto hemos aprendido bastantes cosas. Una de ellas es saber organizarse y repartirse el trabajo según las capacidades de cada uno. Luego entraríamos a la parte técnica, donde hemos aprendido sobre los servicios de Cloud Computing (más en profundidad el de Amazon Web Service), balanceadores de carga (como funcionan y los métodos que utilizan a la hora de redirigir) y eCommerce (cómo implementar un comercio en un servidor y las funciones que te da). Una cosa que no teníamos en mente para este proyecto, era tener que implementar un ddns, por temas de ip pública dinámica. Finalmente ha resultado exitosamente haber optado por instalar un ddns sin conocimientos previos.

Si tuviésemos que cambiar algo de este proyecto, hubiésemos optado por obtener más recursos para aumentar medidas de seguridad como puede ser un sistema RAID en los servidores, pero no ha sido posible por las circunstancias actuales.

## ANEXO

## ERRORES

## 1. No se puede acceder a la página de instalación de PrestaShop.



**Solución:** Instalar la extensión '*SimpleXML*' y '*Zip extension*' de PHP además de añadir 2 líneas en el fichero de configuración de PHP indicando estas dos extensiones para que las utilice.

```
apt-get install php-zip php-simplexml
```

And THEN you have to add it manually to /etc/php/7.0/apache2/php.ini (apache2 or whatever web server you're using).

Just add those two lines at the end of the file :

```
extension=simplexml.so
extension=zip.so
```

Restart apache and you should be done.

## 2. No se puede conectar a la página web desde un cliente que no sea local.



**Solución:** En la configuración de PrestaShop debemos indicar; el dominio de la tienda (IP), dominio SSL (IP) y la base URL.

SEO y URLs

| SEO y URLs                  | Motores de búsqueda | Afilados |
|-----------------------------|---------------------|----------|
| Establecer URL de la tienda |                     |          |
| Dominio de la tienda        | 79.154.51.12        |          |
| Dominio SSL                 | 79.154.51.12        |          |
| Base URI                    | /                   |          |

### 3. No se puede acceder a ciertas zonas de la página web de PrestaShop.

|                 |                |                                                                                                                                                                                                                                                         |
|-----------------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>CRITICAL</b> | <b>php</b>     | Call to undefined function mb_split()                                                                                                                                                                                                                   |
| 22:40:58        |                |                                                                                                                                                                                                                                                         |
| <b>CRITICAL</b> | <b>request</b> | Uncaught PHP Exception Symfony\Component\Debug\Exception\UndefinedFunctionException: "Attempted to call function "mb_split" from the global namespace." at /var/www/html/prestashop/vendor/smarty/smarty/libs/plugins/shared.mb_str_replace.php line 47 |
| 22:40:58        |                |                                                                                                                                                                                                                                                         |

**Solución:** Instalar la extensión de php 'mbstring' (sudo apt-get install php7.2-mbstring)

### 4. No podemos acceder a la página web por cambio de ip pública

**Solución:** Cambiar el valor en la base de datos.

```
mysql> select * from ps_shop_url
-> ;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id_shop_url | id_shop | domain      | domain_ssl | physical_uri | virtual_uri | main | active |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          1 |        1 | 79.154.50.39 | 79.154.50.39 | /           |             |    1 |       1 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
mysql> UPDATE ps_shop_url SET domain=83.41.235.193,domain_ssl=83.41.235.193 WHERE id_shop_url=1;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version
for the right syntax to use near '.235.193,domain_ssl=83.41.235.193 WHERE id_shop_url=1' at line 1
mysql> UPDATE ps_shop_url SET domain='83.41.235.193', domain_ssl='83.41.235.193' WHERE id_shop_url=1;
Query OK, 1 row affected (0.05 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> select * from ps_shop_url;
+-----+-----+-----+-----+-----+-----+-----+-----+
| id_shop_url | id_shop | domain      | domain_ssl | physical_uri | virtual_uri | main | active |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          1 |        1 | 83.41.235.193 | 83.41.235.193 | /           |             |    1 |       1 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

## 5. Error al instalar el servicio haproxy

```

haproxy@haproxy: $ sudo apt install haproxy
[sudo] password for haproxy:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
 liblua5.3-0
Se instalarán los siguientes paquetes NUEVOS:
 haproxy liblua5.3-0
0 actualizados, 2 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 0 B/1.703 kB de archivos.
Se utilizarán 3.945 kB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
Seleccionando el paquete liblua5.3-0:amd64 previamente no seleccionado.
(Leyendo la base de datos ... 72988 ficheros o directorios instalados actualmente.)
Preparando para desempaquetar ../liblua5.3-0_5.3.3-1.ubuntu2_amd64.deb ...
Desempaquetando liblua5.3-0:amd64 (5.3.3-1.ubuntu2) ...
Seleccionando el paquete haproxy previamente no seleccionado.
Preparando para desempaquetar ../haproxy_2.1.4-1ppa1~focal_amd64.deb ...
Desempaquetando haproxy (2.1.4-1ppa1~focal) ...
Configurando liblua5.3-0:amd64 (5.3.3-1.ubuntu2) ...
Configurando haproxy (2.1.4-1ppa1~focal) ...
adduser: The user `haproxy' already exists, but is not a system user. Exiting.
dpkg: error al procesar el paquete haproxy (--configure):
 el subproceso instalado paquete haproxy script post-installation devolvió el código de salida de error 1
Procesando disparadores para man-db (2.9.1-1) ...
Procesando disparadores para libc-bin (2.31-0ubuntu9) ...
Procesando disparadores para rsyslog (8.2001.0-1ubuntu1) ...
Procesando disparadores para systemd (245.4-4ubuntu3) ...
Se encontraron errores al procesar:
 haproxy
E: Sub-process /usr/bin/dpkg returned an error code (1)
haproxy@haproxy: $

```

**Solución:** Crear un usuario el cual no se llame haproxy.

## Error installing haproxy 2.0 #12



RedWolf3121 opened this issue 17 days ago · 2 comments



RedWolf3121 commented 17 days ago

I try to install haproxy with your ppa, and i get this error.

```

haproxy@haproxy: $ sudo apt install haproxy
Reading package lists... Done
Building dependency tree
Reading state information... Done
haproxy is already the newest version (2.0.14-1ppa1~focal).
0 upgraded, 0 newly installed, 0 to remove and 10 not upgraded.
1 not fully installed or removed.
After this operation, 0 B of additional disk space will be used.
Do you want to continue? [Y/n] y
Setting up haproxy (2.0.14-1ppa1~focal) ...
adduser: The user `haproxy' already exists, but is not a system user. Exiting.
dpkg: error processing package haproxy (--configure):
 installed haproxy package post-installation script subprocess returned error exit status 1
Errors were encountered while processing:
 haproxy
E: Sub-process /usr/bin/dpkg returned an error code (1)
haproxy@haproxy: $

```



vincentbernat commented 17 days ago

Owner ...

You seem to have already installed haproxy in the past using an alternative method. There is a `haproxy` user remaining you should delete. You can use `deluser haproxy`.

6. Error al hacer la réplica entre bases de datos

**Solución:** cambiar el parámetro *'bind-address'* por la siguiente IP: 0.0.0.0, para así permitir conexiones entrantes.

7. Error con apache y haproxy que escuchan por el mismo puerto.

```
GNU nano 4.8 /etc/apache2/ports.conf
# If you just change the port or add more ports here, you will likely also
# have to change the VirtualHost statement in
# /etc/apache2/sites-enabled/000-default.conf

Listen 81

<IfModule ssl_module>
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
```

**Solución:** cambiar el puerto por el que escucha *apache* en el fichero de configuración *'/etc/apache2/ports.conf'*.

---

## GLOSARIO

**Prestashop:** Gestor de contenidos (CMS) específico que sirve para crear tiendas virtuales o tiendas online.

**AWS:** es una plataforma de servicios de nube que proporciona una variedad de servicios de infraestructura.

**Instancia:** es un servidor virtual en la nube de AWS. Con Amazon EC2, puede instalar y configurar el sistema operativo y las aplicaciones que se ejecutan en la instancia.

**HAProxy:** es un software libre que actúa como balanceador de carga (load balancer) ofreciendo alta disponibilidad, balanceo de carga y proxy para comunicaciones TCP y HTTP.

**Owncloud:** es una aplicación de software libre del tipo Servicio de alojamiento de archivos, que permite el almacenamiento en línea y aplicaciones en línea (cloud computing).

**CronTab:** es un archivo de configuración que especifica comando shell para ejecutarse periódicamente a una hora específica.

**Apache:** es un software de servidor web gratuito y de código abierto para plataformas Unix.

**MySQL replication:** es un proceso que permite que los datos de un servidor de bases de datos MySQL (master) se copien automáticamente en uno o más servidores de bases de datos MySQL (slave).

**Cloud Computing:** es un paradigma que permite ofrecer servicios de computación a través de una red, que usualmente es Internet.

**Base de datos:** es una colección de información organizada de forma que un programa de ordenador pueda seleccionar rápidamente los fragmentos de datos que necesite.

**eCommerce:** consiste en la distribución, venta, compra, marketing y suministro de información de productos o servicios a través de Internet.

**DDNS:** es un servicio que permite la actualización en tiempo real de la información sobre nombres de dominio situada en un servidor de nombres.

**Balanceador de carga:** es un dispositivo que actúa como proxy inverso distribuyendo el tráfico de red o de una aplicación a varios servidores.



---

## WEBGRAFÍA

### Instalación Del servidor web

<https://www.digitalocean.com/community/tutorials/como-instalar-en-ubuntu-18-04-la-pila-lamp-linux-apache-mysql-y-php-es>

### Instalación y configuración de prestashop

<https://linuxhostsupport.com/blog/how-to-install-prestashop-on-ubuntu-18-04/>

### Consultar información PrestaShop

<https://www.prestashop.com/forums/>

### HaProxy

<https://www.linode.com/docs/uptime/loadbalancing/how-to-use-haproxy-for-load-balancing/>

### Réplica BB.DD

[https://medium.com/@Alibaba\\_Cloud/how-to-configure-mysql-master-slave-replication-on-ubuntu-16-04-3bf97b961730](https://medium.com/@Alibaba_Cloud/how-to-configure-mysql-master-slave-replication-on-ubuntu-16-04-3bf97b961730)

### Instalación duckdns

<https://www.duckdns.org/install.jsp>

### Instalación OwnCloud

<https://www.tecmint.com/install-owncloud-on-ubuntu/>

### SSL Checker

<https://decoder.link/sslchecker/rebuildcars.duckdns.org/443>