

WordPress en AWS

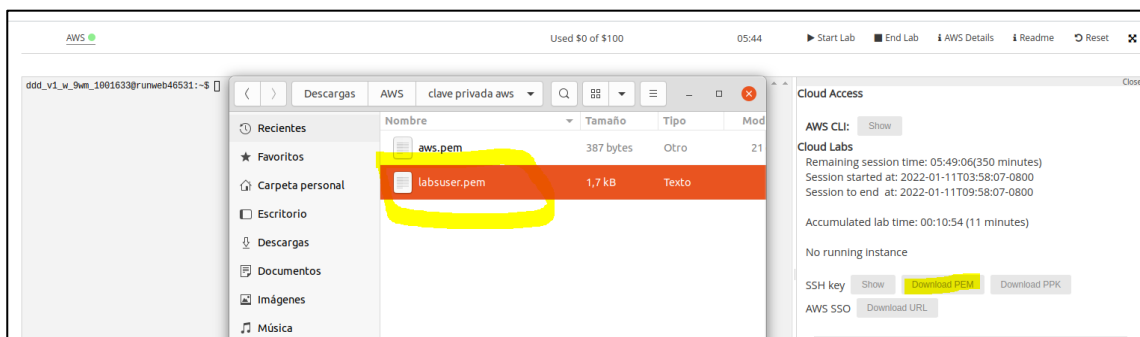
INDICE

1.	<i>CREACIÓN DE UNA NUEVA INSTANCIA EN AWS</i>	2
2.	<i>ACCEDIENDO AL SERVIDOR DE AWS DESDE UBUNTU</i>	4
3.	<i>CONFIGURACIONES PREVIAS A LA INSTALACIÓN DE WORDPRESS</i>	6
4.	<i>DESCARGA E INSTALACIÓN DE WORDPRESS</i>	9
5.	<i>CREACIÓN DE LA BASE DE DATOS DE WORDPRESS</i>	14
6.	<i>CREACIÓN DE UNA CONEXIÓN HTTPS</i>	16
7.	<i>DNS AUTOMÁTICO CON DUCKDNS</i>	21
8.	<i>DESPLIEGUE MANUAL DE UN PROYECTO PHP EN AWS</i>	24
9.	<i>BIBLIOGRAFÍA</i>	31

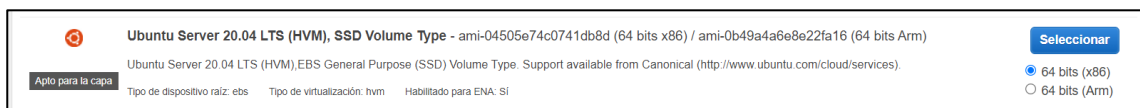
Manual de Wordpress en AWS

1. CREACIÓN DE UNA NUEVA INSTANCIA EN AWS

Antes de crear nuestra nueva instancia, debemos realizar un paso previo. Para poder acceder remotamente a nuestra máquina tendremos que identificarnos para lo que necesitaremos una clave privada de seguridad que descargaremos de la página con los detalles de AWS. La clave se podrá descargar y su nombre por defecto es labsuser.pem.



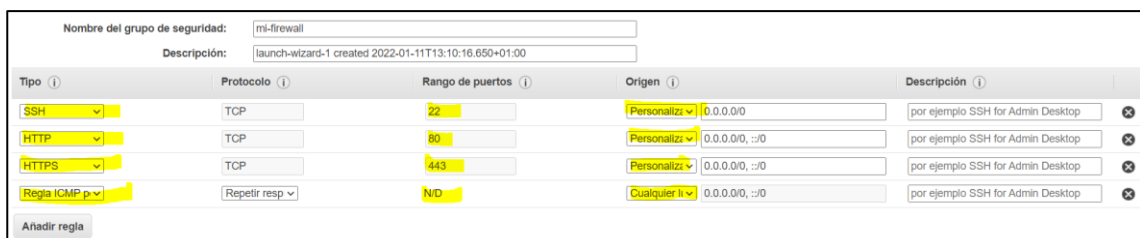
En AWS creamos una máquina con S.O. UBUNTU



Esta será la configuración del tipo máquina que se puede elegir.

t2	t2.medium	2	4	EBS solo	-	De bajo a moderado	Si
----	-----------	---	---	----------	---	--------------------	----

Esta sería la configuración de los servicios de seguridad. Es importante que la regla ICMP sea la de IPV4.



Tras configurar nuestra instancia y tras lanzarla nos pedirá que enlacemos la misma a un par de claves de seguridad. Una publica que almacenará AWS y una privada que es la clave que descargamos antes de iniciar la creación de la nueva instancia.

Esta será la configuración que debemos elegir.

Seleccione un par de claves existente o cree un nuevo par de claves

Un par de claves consta de una **clave pública** que AWS almacena y un **archivo de claves privadas** que usted almacena. Juntas, le permiten conectarse a su instancia de forma segura. Para las AMI de Windows, el archivo de claves privadas es necesario para obtener la contraseña usada para iniciar sesión en la instancia. Para las AMI de Linux, el archivo de claves privadas le permite establecer una conexión SSH segura con su instancia. Amazon EC2 es compatible con los tipos de clave RSA y ED25519.

Nota: El par de claves seleccionado se añadirá al conjunto de claves autorizadas para esta instancia. Obtenga más información sobre [cómo eliminar pares de claves existentes de una AMI pública](#).

Elegir un par de claves existente ▼

Seleccionar un par de claves

vockey | RSA ▼

☒ Confirmo que tengo acceso al archivo de clave privada correspondiente, y que sin este archivo, no podré iniciar sesión en mi instancia.

[Cancelar](#)

[Lanzar instancias](#)

2. ACCEDIENDO AL SERVIDOR DE AWS DESDE UBUNTU

Para acceder a AWS desde nuestra máquina local abrimos una terminal y establecemos la ruta dónde se almacenó la clave privada que se descargó en el paso anterior (labsuser.pem).

```
alberto@alberto:~/Descargas/AWS/clave privada aws$ ls -la
total 16
drwxrwxrwx 2 alberto alberto 4096 ene 11 13:09
drwxrwxrwx 3 alberto alberto 4096 ene 11 13:09
-r----- 1 alberto alberto 387 oct 21 09:51 aws.pem
-rw-rw-r-- 1 alberto alberto 1674 ene 11 13:09 labsuser.pem
```

Para poder acceder a la clave para la conexión debemos, previamente, darle derechos de lectura al propietario utilizando el comando `chmod`.

```
alberto@alberto:~/Descargas/AWS/clave privada aws$ sudo chmod 400 labsuser.pem
alberto@alberto:~/Descargas/AWS/clave privada aws$
```

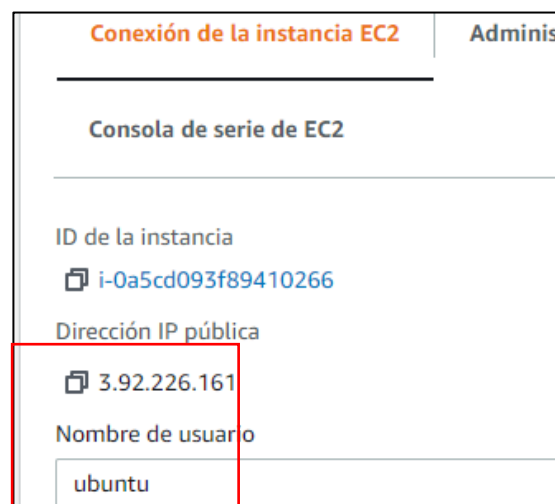
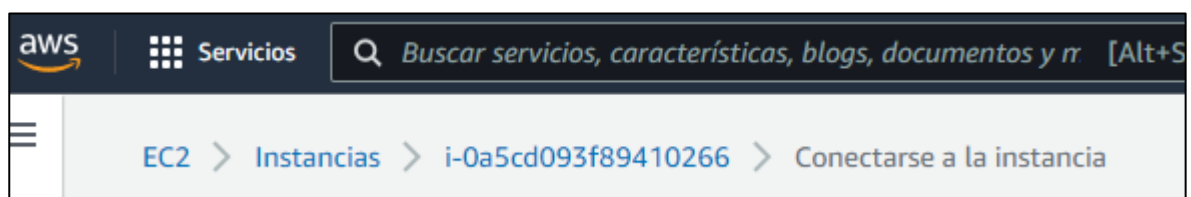
Para acceder a AWS desde nuestra máquina local utilizaremos la línea de comandos,

ssh [-i(opción de clave privada) clave privada] [usuario@][nombre_host]

ssh -i labsuser.pem ubuntu@3.90.50.22

```
alberto@alberto:~/Descargas/AWS/clave privada aws$ ssh -i labsuser.pem ubuntu@3.90.50.22
The authenticity of host '3.90.50.22 (3.90.50.22)' can't be established.
ECDSA key fingerprint is SHA256:AwRV8+CVr2dnAE7jFBmNGXavDu+Uxd0Ya0oyB3fB76I.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

Para conocer el nombre del usuario y el nombre del host podemos mirarlo en AWS accediendo a la instancia que vamos a ejecutar.



Si queremos más comodidad podemos hacer que el acceso a de la máquina local al servidor sea más sencillo. Para ello tenemos que dirigirnos a la carpeta .ssh (en home ~) de nuestra máquina local y crear un fichero de configuración llamado config con los siguientes detalles.

Host despliegue

HostName bravoga.duckdns.org
User ubuntu
#Port 22
IdentityFile ~/daw/labsuser.pem
ServerAliveInterval 60
ServerAliveCountMax 6

```
GNU nano 4.8 config
Host despliegue
  HostName bravoga.duckdns.org
  User ubuntu
  #Port 22
  IdentityFile /home/alberto/daw/labsuser.pem
  ServerAliveInterval 60
  ServerAliveCountMax 6
```

Ahora bastará con acceder a la máquina remota con, ssh despliegue

```
alberto@alberto:/$ ssh despliegue
Warning: Permanently added the ECDSA host key for IP address '44.200.100.100'.
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1028-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

System information as of Thu Feb 24 16:37:03 UTC 2022

System load:  0.1          Processes:      131
Usage of /:   38.0% of 7.69GB Users logged in:   0
Memory usage: 6%          IPv4 address for eth0: 172.31.80.1
Swap usage:   0%

26 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Last login: Thu Feb 24 14:58:42 2022 from 2.153.93.204
ubuntu@ip-172-31-80-200:~$
```

Una vez dentro del servidor de AWS con el comando lscpu podemos verificar los datos de la máquina activa.

```
ubuntu@ip-172-31-80-200:~$ lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Byte Order:            Little Endian
Address sizes:          46 bits physical, 48 bits virtual
CPU(s):                 2
On-line CPU(s) list:   0,1
Thread(s) per core:     1
Core(s) per socket:     2
Socket(s):              1
NUMA node(s):          1
Vendor ID:              GenuineIntel
CPU family:             6
Model:                  70
```

Como último paso actualizamos el repositorio de Ubuntu con, sudo apt update

```
ubuntu@ip-172-31-80-200:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [51.9 kB]
Hit:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-security InRelease
```

3. CONFIGURACIONES PREVIAS A LA INSTALACIÓN DE WORDPRESS

Guía de ayuda si hay dudas:

<https://linuxconfig.org/ubuntu-20-04-wordpress-with-nginx-installation>

1. Instalamos mariadb y verificamos su estatus

sudo apt install mariadb-server

systemctl status mariadb.service

```
ubuntu@ip-172-31-80-200:~$ systemctl status mariadb.service
● mariadb.service - MariaDB 10.3.32 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2022-01-11 12:40:08 UTC; 1min 28s ago
     Docs: man:mysqld(8)
           https://mariadb.com/kb/en/library/systemd/
  Main PID: 2352 (mysqld)
    Status: "Taking your SQL requests now..."
     Tasks: 31 (limit: 4693)
    Memory: 64.5M
   CGroup: /system.slice/mariadb.service
           └─2352 /usr/sbin/mysqld

Jan 11 12:40:09 ip-172-31-80-200 /etc/mysql/debian-start[2399]: Processing databases
Jan 11 12:40:09 ip-172-31-80-200 /etc/mysql/debian-start[2399]: information_schema
Jan 11 12:40:09 ip-172-31-80-200 /etc/mysql/debian-start[2399]: mysql
Jan 11 12:40:09 ip-172-31-80-200 /etc/mysql/debian-start[2399]: performance_schema
Jan 11 12:40:09 ip-172-31-80-200 /etc/mysql/debian-start[2399]: Phase 6/7: Checking and upgrading tables
Jan 11 12:40:09 ip-172-31-80-200 /etc/mysql/debian-start[2399]: Processing databases
Jan 11 12:40:09 ip-172-31-80-200 /etc/mysql/debian-start[2399]: information_schema
```

2. securizamos mariadb

sudo mysql_secure_installation

```
ubuntu@ip-172-31-80-200:/etc/php/7.4/fpm/pool.d$ 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!
```

en la contraseña del root decimos que no queremos que haya y todo lo demás yes.

3. Instalamos nginx y verificamos su estatus

sudo apt install nginx

systemctl status nginx.service

```
ubuntu@ip-172-31-80-200:~$ systemctl status nginx.service
● nginx.service - A high performance web server and a reverse proxy server
   Loaded: loaded (/lib/systemd/system/nginx.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2022-01-11 12:42:25 UTC; 2min 8s ago
     Docs: man:nginx(8)
  Main PID: 14921 (nginx)
     Tasks: 3 (limit: 4693)
    Memory: 4.3M
   CGroup: /system.slice/nginx.service
           └─14921 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
             └─14922 nginx: worker process
               └─14923 nginx: worker process

Jan 11 12:42:25 ip-172-31-80-200 systemd[1]: Starting A high performance web server and a reverse proxy server...
Jan 11 12:42:25 ip-172-31-80-200 systemd[1]: Started A high performance web server and a reverse proxy server.
```

4. Instalamos php-fpm y verificamos su estatus

sudo apt install php-fpm

systemctl status php7.4-fpm.service

```
ubuntu@ip-172-31-80-200:/etc/nginx/sites-enabled$ systemctl status php7.4-fpm.service
● php7.4-fpm.service - The PHP 7.4 FastCGI Process Manager
   Loaded: loaded (/lib/systemd/system/php7.4-fpm.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2022-01-11 13:03:19 UTC; 3h 43min ago
     Docs: man:php-fpm7.4(8)
   Process: 22771 ExecStartPost=/usr/lib/php/php-fpm-socket-helper install /run/php/php-fpm.sock
   Main PID: 22767 (php-fpm7.4)
    Status: "Processes active: 0, idle: 2, Requests: 8, slow: 0, Traffic: 0req/sec"
```

5. Instalamos php-MySQL

sudo apt install php-mysql

```
ubuntu@ip-172-31-80-200:/etc/php/7.4/fpm/pool.d$ sudo apt install php-mysql
```

6. Ahora revisamos el servidor de nginx en, /etc/nginx/nginx.conf

```
ubuntu@ip-172-31-80-200:/etc/nginx/conf.d$ cd ..
ubuntu@ip-172-31-80-200:/etc/nginx$ nano nginx.conf
ubuntu@ip-172-31-80-200:/etc/nginx$
```

vemos que el usuario para user y group será www-data en lugar de nginx como usábamos en las máquinas anteriores que no eran Ubuntu

```
GNU nano 4.8 nginx.conf
user www-data;
worker_processes auto;
pid /run/nginx.pid;
include /etc/nginx/modules-enabled/*.conf;

events {
    worker_connections 768;
    # multi_accept on;
}

http {
```

```
##
# Virtual Host Configs
##

include /etc/nginx/conf.d/*.conf;
include /etc/nginx/sites-enabled/*;
}
```

Nos indica que guardará las páginas en estas posibles ubicaciones (vemos q hay en cada uno de ellos)

include /etc/nginx/conf.d/*.conf;

include /etc/nginx/sites-enabled/*;

verificamos el contenido de las carpetas conf.d y sites_enabled.

ls -la conf.d/

```
ubuntu@ip-172-31-80-200:/etc/nginx$ ls -la conf.d/
total 8
drwxr-xr-x 2 root root 4096 May 25  2021 .
drwxr-xr-x 8 root root 4096 Jan 11 12:42 ..
```

ls -la sites-enabled/

```
ubuntu@ip-172-31-80-200:/etc/nginx$ ls -la sites-enabled/
total 8
drwxr-xr-x 2 root root 4096 Jan 11 12:42 .
drwxr-xr-x 8 root root 4096 Jan 11 12:42 ..
lrwxrwxrwx 1 root root   34 Jan 11 12:42 default -> /etc/nginx/sites-available/default
```

Accedemos para ver el contenido de default

vim sites-available/default

```
ubuntu@ip-172-31-80-200:/etc/nginx$ vim sites-available/default
```

Ahora buscamos la configuración de PHP y cambiamos el usuario en user y group tal y como aparece en el fichero de configuración de nginx (www-data).

```
ubuntu@ip-172-31-80-200:/etc/php/7.4/fpm/pool.d$ nano www.conf
```

```
; - 'chdir'
; - 'php_values'
; - 'php_admin_values'
; When not set, the global prefix (or /usr) applies i
; Note: This directive can also be relative to the gl
; Default Value: none
;prefix = /path/to/pools/$pool

; Unix user/group of processes
; Note: The user is mandatory. If the group is not se
; will be used.
user = www-data
group = www-data

; The address on which to accept FastCGI requests.
; Valid syntaxes are:
;   'ip.add.re.ss:port'    - to listen on a TCP socke
;                           a specific port;
;   '[ip:6:addr:ess]:port' - to listen on a TCP socke
;                           a specific port;
;   'port'                 - to listen on a TCP socke
;                           (IPv6 and IPv4-mapped) o
;   '/path/to/unix/socket' - to listen on a unix sock
; Note: This value is mandatory.
listen = /run/php/php7.4-fpm.sock
```

IMPORTANTE, copiamos los datos de la directiva listen ya que habrá que ponerla luego en el servidor de nginx, /run/php/php7.4-fpm.sock

4. DESCARGA E INSTALACIÓN DE WORDPRESS

Vamos a descargar wordpress directamente de la web, pero es importante solicitar el paquete desde home `~` con el comando,

`curl [-L (location)][-O (output)]...`

`curl -LO https://wordpress.org/latest.tar.gz`

```
ubuntu@ip-172-31-80-200:~$ curl -LO https://wordpress.org/latest.tar.gz
% Total    % Received % Xferd Average Speed   Time    Time     Time  Current
           % Done    Dload  Upload    Total   Spent    Left   Speed
```

descomprimos el paquete tar con, `tar xvf latest.tar.gz` (si es necesario podemos instalar un descompresor)

```
ubuntu@ip-172-31-80-200:~$ tar xvf latest.tar.gz
wordpress/
wordpress/xmlrpc.php
wordpress/wp-blog-header.php
```

debe haberlo bajado en home

```
ubuntu@ip-172-31-80-200:~$ ls
latest.tar.gz  wordpress
```

lo muevo a la carpeta www con, `sudo mv wordpress /var/www/`

```
ubuntu@ip-172-31-80-200:~$ sudo mv wordpress /var/www/
```

ahora tengo que cambiar el propietario del archivo con el comando `chown`,

`sudo chown -R www-data:www-data /var/www/wordpress`

```
ubuntu@ip-172-31-80-200:~$ sudo chown -R www-data:www-data /var/www/wordpress
ubuntu@ip-172-31-80-200:~$ ls -la /var/www
total 16
drwxr-xr-x  4 root    root    4096 Jan 11 13:11 .
drwxr-xr-x 14 root    root    4096 Jan 11 12:42 ..
drwxr-xr-x  2 root    root    4096 Jan 11 12:42 html
drwxr-xr-x  5 www-data www-data 4096 Jan  6 19:13 wordpress
ubuntu@ip-172-31-80-200:~$
```

Ahora configuramos nginx con un servidor nuevo y para ello vamos a eliminar el servidor por defecto actual de nginx de la carpeta, sites-enabled

```
ubuntu@ip-172-31-80-200:~$ cd /etc/nginx/sites-enabled/
ubuntu@ip-172-31-80-200:/etc/nginx/sites-enabled$ ls -la
total 8
drwxr-xr-x  2 root    root    4096 Jan 11 12:42 .
drwxr-xr-x  8 root    root    4096 Jan 11 12:42 ..
lrwxrwxrwx  1 root    root      34 Jan 11 12:42 default -> /etc/nginx/sites-available/default
ubuntu@ip-172-31-80-200:/etc/nginx/sites-enabled$ sudo rm default
ubuntu@ip-172-31-80-200:/etc/nginx/sites-enabled$ ls -la
total 8
drwxr-xr-x  2 root    root    4096 Jan 11 13:14 .
drwxr-xr-x  8 root    root    4096 Jan 11 12:42 ..
ubuntu@ip-172-31-80-200:/etc/nginx/sites-enabled$ ls -la ../sites-available/
total 12
drwxr-xr-x  2 root    root    4096 Jan 11 12:42 .
drwxr-xr-x  8 root    root    4096 Jan 11 12:42 ..
-rw-r--r--  1 root    root    2416 Mar 26  2020 default
```

Creamos el nuevo fichero, dónde configuraremos el servidor para Wordpress. Se llamará, wordpress.

`sudo nano ../sites-available/wordpress`

```
ubuntu@ip-172-31-80-200:/etc/nginx/sites-enabled$ ls -la
total 8
drwxr-xr-x 2 root root 4096 Jan 11 13:14 .
drwxr-xr-x 8 root root 4096 Jan 11 12:42 ..
ubuntu@ip-172-31-80-200:/etc/nginx/sites-enabled$ sudo nano ../sites-available/wordpress
```

estos son los datos que debe contener este fichero.

<https://wordpress.org/support/article/nginx/#main-generic-statartup-file>

General WordPress rules #

For single site installations, here is the 'global/wordpress.conf' file:

```
1 # WordPress single site rules.
2 # Designed to be included in any server {} block.
3 # Upstream to abstract backend connection(s) for php
4 upstream php {
5     server unix:/tmp/php-cgi.socket;
6     server 127.0.0.1:9000;
7 }
8
9 server {
10     ## Your website name goes here.
11     server_name domain.tld;
12     ## Your only path reference.
13     root /var/www/wordpress;
14     ## This should be in your http block and if it is, it
15     index index.php;
16
17     location = /favicon.ico {
18         log_not_found off;
19         access_log off;
20     }
21
22     location = /robots.txt {
```

Ahora cambiamos el socket de [php \(ir a la nota con el aviso\)](#) y el nombre del usuario

```
Upstream to abstract backend connection(s) for php
upstream php {
    #server unix:/tmp/php-cgi.socket;
    #server 127.0.0.1:9000;
    server unix:/run/php/php7.4-fpm.sock;

server {
    ## Your website name goes here.
    server_name _;
    ## Your only path reference.
    root /var/www/wordpress;
    ## This should be in your http block and if it
```

SE PUEDE AÑADIR EN EL SERVER listen 80;

```
server {  
    listen 80;  
    ## Your website name
```

como siempre, tras modificar un fichero de configuración de nginx verifico que todo es correcto con el comando, nginx -t

```
ubuntu@ip-172-31-80-200:/etc/nginx/sites-enabled$ sudo nginx -t  
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok  
nginx: configuration file /etc/nginx/nginx.conf test is successful  
ubuntu@ip-172-31-80-200:/etc/nginx/sites-enabled$
```

y reiniciamos nginx con sudo systemctl restart nginx

```
ubuntu@ip-172-31-80-200:/etc/nginx/sites-enabled$ sudo systemctl restart nginx
```

creamos un enlace simbólico al archivo de configuración con

sudo ln -s ../sites-available/wordpress

(el punto es el directorio de destino, pero podría ser opcional)

```
ubuntu@ip-172-31-80-200:/etc/nginx/sites-available$ sudo ln -s ../sites-available/wordpress .
```

```
ubuntu@ip-172-31-80-200:/etc/nginx/sites-enabled$ sudo ln -s ../sites-available/wordpress
```

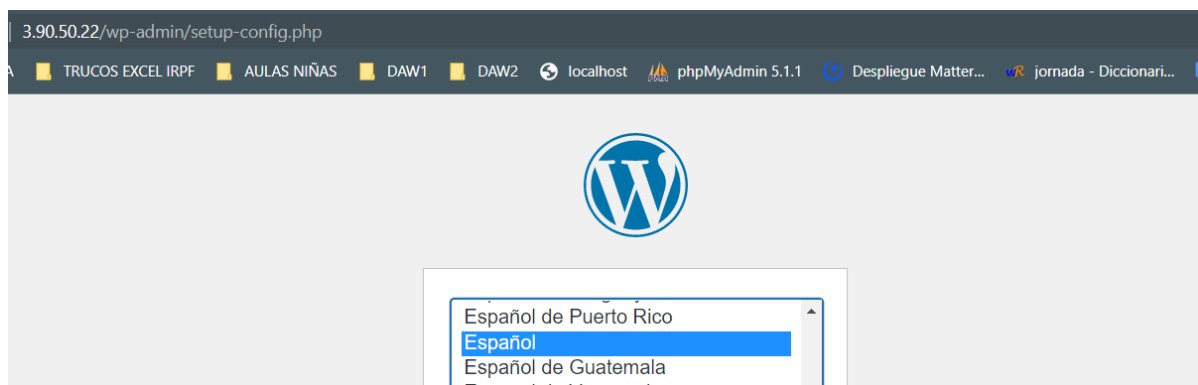
probamos si conectamos con, curl localhost -v

```
ubuntu@ip-172-31-80-200:~$ curl localhost -v  
* Trying 127.0.0.1:80...  
* TCP_NODELAY set  
* Connected to localhost (127.0.0.1) port 80 (#0)  
> GET / HTTP/1.1  
> Host: localhost  
> User-Agent: curl/7.68.0  
> Accept: */*  
>  
* Mark bundle as not supporting multiuse  
< HTTP/1.1 302 Found  
< Server: nginx/1.18.0 (Ubuntu)  
< Date: Thu, 13 Jan 2022 09:43:03 GMT  
< Content-Type: text/html; charset=UTF-8  
< Transfer-Encoding: chunked  
< Connection: keep-alive  
< Location: http://localhost/wp-admin/setup-config.php  
<  
* Connection #0 to host localhost left intact  
ubuntu@ip-172-31-80-200:~$
```

sí detectamos algún problema, podemos verificar los puertos activos con sudo ss -tlnp

```
ubuntu@ip-172-31-80-200:~$ sudo ss -tlnp
State      Recv-Q    Send-Q    Local Address:Port    Peer Address:Port    Process
LISTEN     0          80        127.0.0.1:3306        0.0.0.0:*            users:((("mysqld",pid=646,fd=21))
LISTEN     0          511       0.0.0.0:80           0.0.0.0:*            users:((("nginx",pid=531,fd=6),("nginx",pid=530,fd=0),("nginx",pid=519,fd=0))
LISTEN     0          4096      127.0.0.53:53        0.0.0.0:*            users:((("systemd-resolve",pid=389,fd=13))
LISTEN     0          128       0.0.0.0:22           0.0.0.0:*            users:((("sshd",pid=727,fd=3))
LISTEN     0          128       :::22                :::*                  users:((("sshd",pid=727,fd=4))
```

Para acabar, accedemos en el navegador a la ip de nuestro servidor y ya tendremos Wordpress instalado (importante que el acceso sea con http y **NO** con https



A modo de resumen puedo ver los comandos insertados para instalar con history | grep install (con grep filtramos por palabra), aparecerán todas las veces que hemos usado la palabra install con lo que puede listar paquetes llamados erróneamente.

```
ubuntu@ip-172-31-80-200:~$ history | grep install
 2  sudo apt install mariadb-service
 3  sudo apt install mariadb-server
 8  sudo apt install nginx
 9  sudo apt install ftl-fpm
10  sudo apt install php-fpm
41  sudo apt install php-mysql
42  sudo mysql_secure_installation
138 history | grep install
ubuntu@ip-172-31-80-200:~$
```

A modo de información, si podemos poner sólo history y veremos todos los comandos introducidos

```
ubuntu@ip-172-31-80-200:~$ history
 1  sudo apt update
 2  sudo apt install mariadb-service
 3  sudo apt install mariadb-server
 4  sudo systemctl mariadb.service
 5  sudo systemctl status mariadb.service
 6  sudo systemctl status mariadb
 7  systemctl status mariadb.service
 8  sudo apt install nginx
```

Para ver si los paquetes se instalaron con apt show [nombre paquete]

```
ubuntu@ip-172-31-80-200:~$ apt show php-fpm
Package: php-fpm
Version: 2:7.4+75
Priority: optional
Section: universe/php
Source: php-defaults (75)
Origin: Ubuntu
```

5. CREACIÓN DE LA BASE DE DATOS DE WORDPRESS

Dentro del servidor,

1. Accedemos a MySQL con sudo ya que así accedemos como si fuéramos root.

```
ubuntu@ip-172-31-80-200:/etc/nginx/sites-available$ sudo mysql
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 36
Server version: 10.3.32-MariaDB-0ubuntu0.20.04.1 Ubuntu 20.04

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)]> 
```

2. Creamos un usuario

create user 'wordpress'@'localhost' identified by 'password';

```
MariaDB [(none)]> create user 'wordpress'@'localhost' identified by 'password';
Query OK, 0 rows affected (0.000 sec)
```

3. Le damos todos los privilegios

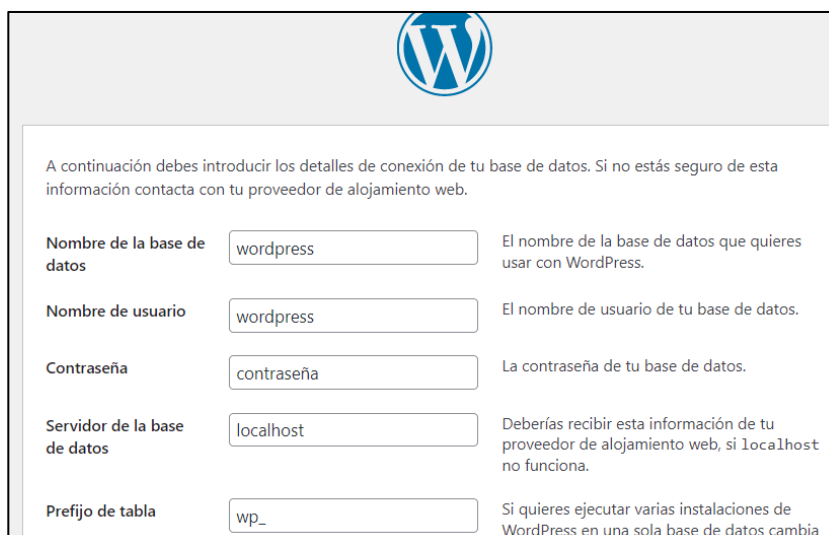
grant all privileges on wordpress.* to 'wordpress'@'localhost';

```
MariaDB [(none)]> grant all privileges on wordpress.* to 'wordpress'@'localhost';
Query OK, 0 rows affected (0.000 sec)
```

Si hay errores en la creación de un usuario, lo podemos borrar con drop user [usuario]

```
MariaDB [(none)]> drop user 'wordpress'@'localhost';
Query OK, 0 rows affected (0.000 sec)
```

Accedemos a nuestra web con [http://\[ip actual\]](http://[ip actual]) proporcionada en AWS lab.



The image shows the WordPress database configuration screen. At the top is the WordPress logo. Below it is a message: "A continuación debes introducir los detalles de conexión de tu base de datos. Si no estás seguro de esta información contacta con tu proveedor de alojamiento web." There are five input fields with labels and descriptions to their right:

Field Label	Input Value	Description
Nombre de la base de datos	wordpress	El nombre de la base de datos que quieres usar con WordPress.
Nombre de usuario	wordpress	El nombre de usuario de tu base de datos.
Contraseña	contraseña	La contraseña de tu base de datos.
Servidor de la base de datos	localhost	Deberías recibir esta información de tu proveedor de alojamiento web, si localhost no funciona.
Prefijo de tabla	wp_	Si quieres ejecutar varias instalaciones de WordPress en una sola base de datos cambia

Nos pedirá instalar y le damos que sí y llegamos a la fase de instalación

WordPress

Bienvenido al famoso proceso de instalación de WordPress en cinco minutos! Simplemente completa la información siguiente y estarás a punto de usar la más enriquecedora y potente plataforma de publicación personal del mundo.

Información necesaria

Por favor, debes facilitarnos los siguientes datos. No te preocupes, siempre podrás cambiar estos ajustes más tarde.

Título del sitio

Nombre de usuario
Los nombres de usuario pueden tener únicamente caracteres alfanuméricos, espacios, guiones bajos, guiones medios, puntos y el símbolo @.

Contraseña [Ocultar](#)
Muy débil

Importante: Necesitas esta contraseña para acceder. Por favor, guárdala en un lugar seguro.

Confirma la contraseña ☒ Confirma el uso de una contraseña débil.

Tu correo electrónico
Comprueba bien tu dirección de correo electrónico antes de continuar.

Visibilidad en los motores de búsqueda ☒ Disuadir a los motores de búsqueda de indexar este sitio
Depende de los motores de búsqueda atender esta petición o no.

[Instalar WordPress](#)

¡Lo lograste!

WordPress ya está instalado. ¡Gracias, y que lo disfrutes!

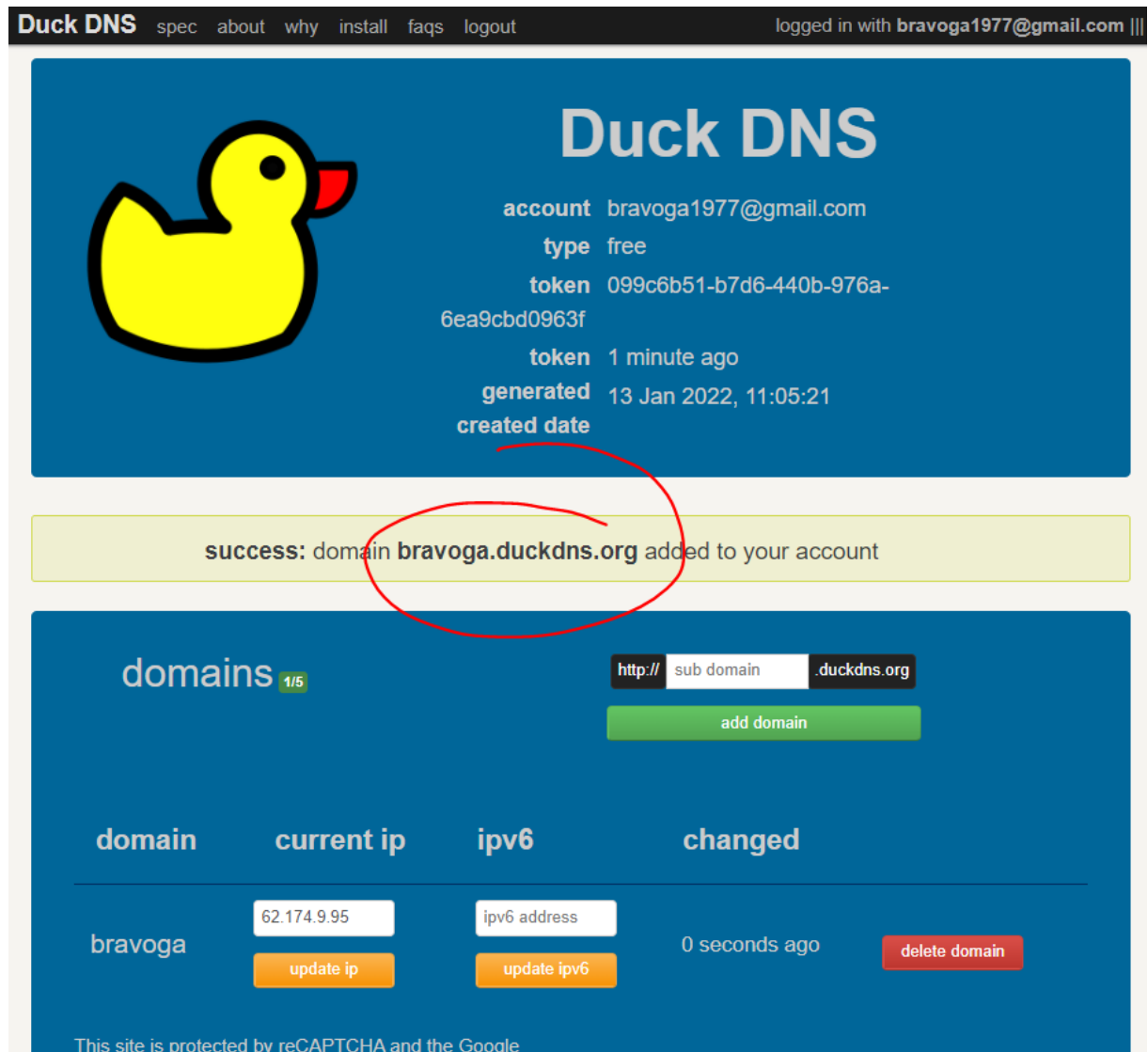
Nombre de usuario alberto

Contraseña La contraseña que has elegido.

[Acceder](#)

6. CREACIÓN DE UNA CONEXIÓN HTTPS

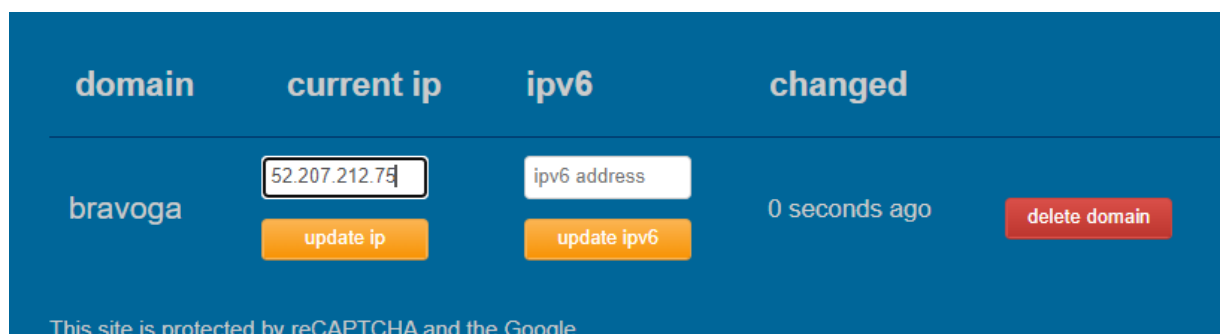
Para poder configurar este tipo de conexión, previamente necesitamos crear un certificado de seguridad, pero ese certificado debe estar unido a un nombre de dominio. Podemos crear un dominio gratuito en la web <https://www.duckdns.org/>



The screenshot shows the Duck DNS website interface. At the top, there's a navigation bar with links: spec, about, why, install, faqs, logout. The user is logged in with **bravoga1977@gmail.com**. The main header features a yellow duck logo and the text "Duck DNS". Below this, account details are listed: **account** bravoga1977@gmail.com, **type** free, **token** 099c6b51-b7d6-440b-976a-6ea9cbd0963f, **token** 1 minute ago, **generated** 13 Jan 2022, 11:05:21, and **created date**. A green success message states: "success: domain **bravoga.duckdns.org** added to your account". Below this, the "domains" section shows a table with columns: domain, current ip, ipv6, and changed. The table has one entry for "bravoga" with current ip 62.174.9.95 and a "0 seconds ago" timestamp. A red circle highlights the domain name "bravoga.duckdns.org" in the success message and the "bravoga" domain in the table.

domain	current ip	ipv6	changed
bravoga	62.174.9.95	ipv6 address	0 seconds ago

Ahora ponemos la IP de AWS y actualizamos en "Update ip"



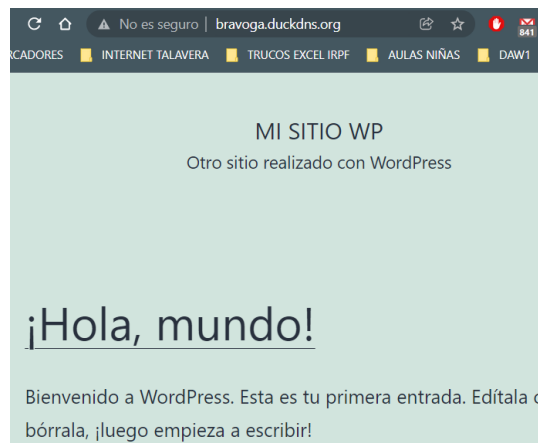
This screenshot shows the same Duck DNS interface, but the "current ip" for the "bravoga" domain has been updated to 52.207.212.75. The "update ip" button is highlighted in orange. The "changed" column still shows "0 seconds ago".

domain	current ip	ipv6	changed
bravoga	52.207.212.75	ipv6 address	0 seconds ago

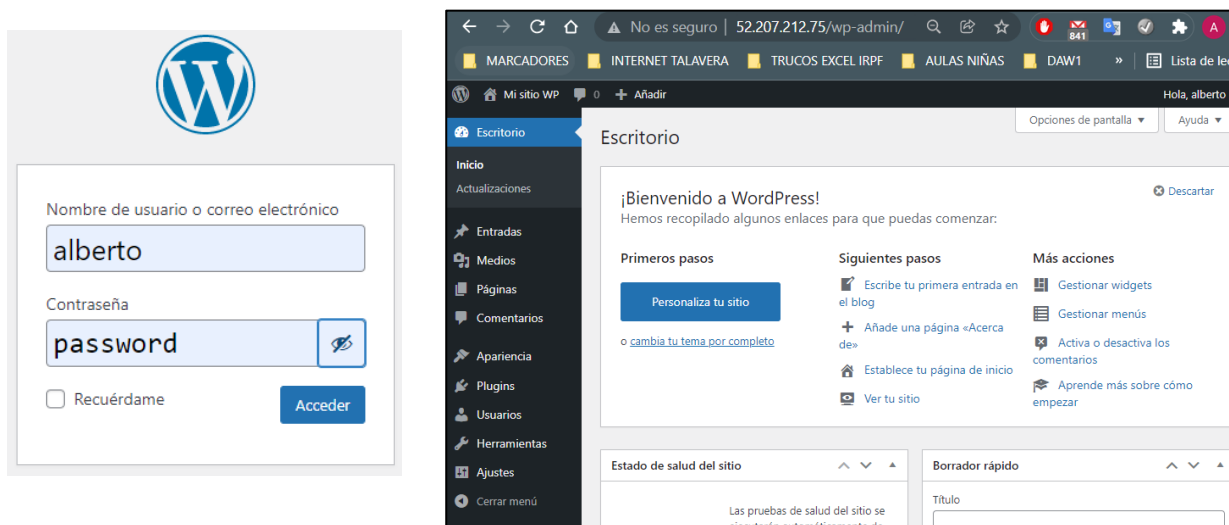
Como la IP la cambiará AWS cada vez que lo reiniciemos, más adelante lo automatizaremos para que duck DNS actualice su IP de forma automática.

Ahora podremos acceder con https al servidor usando nuestro dominio.

<http://bravoga.duckdns.org>



Ahora, si entramos al menú del administrador y nos logueamos desde nuestro dominio, <http://bravoga.duckdns.org/wp-admin> podremos ver el dashboard de WordPress y empezar a utilizarlo al 100%.



Ahora vamos a crear el certificado para preparar el acceso seguro por https. Para ello instalaremos en nuestra máquina remota certbot.

`sudo apt install certbot`

Con certbot creamos un certificado para un nombre de dominio específico.

```
ubuntu@ip-172-31-80-200:~$ sudo apt install certbot
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
```

Para ver los certificados creados utilizaremos sudo certbot certificates

```
ubuntu@ip-172-31-80-200:~$ sudo certbot certificates
Saving debug log to /var/log/letsencrypt/letsencrypt.log

-----
Found the following certs:
Certificate Name: bravoga.duckdns.org
Domains: bravoga.duckdns.org
Expiry Date: 2022-04-13 10:37:39+00:00 (VALID: 89 days)
Certificate Path: /etc/letsencrypt/live/bravoga.duckdns.org/fullchain.pem
Private Key Path: /etc/letsencrypt/live/bravoga.duckdns.org/privkey.pem
-----
ubuntu@ip-172-31-80-200:~$ AC
```

Ahora instalamos el plugin de certbot para nginx

sudo apt install python3-certbot-nginx

```
ubuntu@ip-172-31-80-200:~$ sudo apt install python3-certbot-nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
```

Hechas todas las instalaciones vamos y con nuestro dominio ya creado procedemos con la creación de un nuevo servidor al que llamaremos certificado para poder generar un certificado.

vamos a la ruta cd /etc/nginx/sites-available y creamos el fichero y lo editamos

sudo touch certificado

sudo nano certificado

```
GNU nano 4.8
server{
    server_name bravoga.duckdns.org;
    root /var/www/certificado;

    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/bravoga.duckdns.org/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/bravoga.duckdns.org/privkey.pem;
}
```

Vamos a /etc/nginx/sites-enabled y borramos el fichero de configuración por defecto

```
total 8
drwxr-xr-x 2 root root 4096 Jan 31 15:56 .
drwxr-xr-x 8 root root 4096 Jan 31 15:56 ..
lrwxrwxrwx 1 root root   34 Jan 31 15:56 default -> /etc/nginx/sites-available/default
```

```
ubuntu@ip-172-31-80-200:/etc/nginx/sites-enabled$ sudo rm default
```

Creamos un enlace al fichero certificado

sudo ln -s /etc/nginx/sites-available/certificado

```
ubuntu@ip-172-31-94-176:/etc/nginx/sites-enabled$ sudo ln -s /etc/nginx/sites-available/certificado
ubuntu@ip-172-31-94-176:/etc/nginx/sites-enabled$ ls -al
total 8
drwxr-xr-x 2 root root 4096 Jan 31 16:27 .
drwxr-xr-x 8 root root 4096 Jan 31 15:56 ..
lrwxrwxrwx 1 root root   38 Jan 31 16:27 certificado -> /etc/nginx/sites-available/certificado
ubuntu@ip-172-31-94-176:/etc/nginx/sites-enabled$
```

Comprobamos que la configuración es correcta y reiniciamos nginx:

sudo nginx -t

sudo systemctl restart nginx

```
ubuntu@ip-172-31-94-176:/etc/nginx/sites-enabled$ sudo nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
ubuntu@ip-172-31-94-176:/etc/nginx/sites-enabled$ sudo systemctl restart nginx
ubuntu@ip-172-31-94-176:/etc/nginx/sites-enabled$
```

Vamos a /var/www donde creamos un directorio con el mismo nombre que hemos puesto en la root del servidor:

```
ubuntu@ip-172-31-94-176:/etc/nginx/sites-enabled$ cd /var/www
ubuntu@ip-172-31-94-176:/var/www$ ls
html
ubuntu@ip-172-31-94-176:/var/www$ sudo mkdir certificado
ubuntu@ip-172-31-94-176:/var/www$ ls
certificado html
ubuntu@ip-172-31-94-176:/var/www$
```

Creamos en el directorio certificado un index básico.

```
ubuntu@ip-172-31-94-176:/var/www$ cd certificado/
ubuntu@ip-172-31-94-176:/var/www/certificado$ sudo touch index.html
ubuntu@ip-172-31-94-176:/var/www/certificado$ sudo nano index.html
ubuntu@ip-172-31-94-176:/var/www/certificado$ cat index.html
Hola
Soy tu pagina web
ubuntu@ip-172-31-94-176:/var/www/certificado$
```

Cambiamos al propietario del directorio certificado y su contenido.

sudo chown -R www-data:www-data .

```
ubuntu@ip-172-31-94-176:/var/www/certificado$ sudo chown -R www-data:www-data .
ubuntu@ip-172-31-94-176:/var/www/certificado$ cd ..
ubuntu@ip-172-31-94-176:/var/www$ ls -al
total 16
drwxr-xr-x 4 root root 4096 Jan 31 16:31 .
drwxr-xr-x 14 root root 4096 Jan 31 15:56 ..
drwxr-xr-x 2 www-data www-data 4096 Jan 31 16:34 certificado
drwxr-xr-x 2 root root 4096 Jan 31 15:56 html
ubuntu@ip-172-31-94-176:/var/www$
```

Creamos el certificado firmado por la CA let's encrypt con el comando,

sudo certbot --nginx -d bravoga.duckdns.org

completamos los datos y aceptaremos los términos y condiciones.

```

Setting up python3-certbot-nginx (0.40.0-0ubuntu0.1) ...
ubuntu@ip-172-31-80-200:~$ sudo certbot --nginx -d bravoga.duckdns.org
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator nginx, Installer nginx
Enter email address (used for urgent renewal and security notices) (Enter 'c' to
cancel): bravoga1977@gmail.com

. . . . .
Please read the Terms of Service at
https://letsencrypt.org/documents/LE-SA-v1.2-November-15-2017.pdf. You must
agree in order to register with the ACME server at
https://acme-v02.api.letsencrypt.org/directory
. . . . .
(Agree/(C)ancel): a

. . . . .
Would you be willing to share your email address with the Electronic Frontier
Foundation, a founding partner of the Let's Encrypt project and the non-profit
organization that develops Certbot? We'd like to send you email about our work
encrypting the web, EFF news, campaigns, and ways to support digital freedom.
. . . . .
(Y)es/(N)o: y
Obtaining a new certificate
Performing the following challenges:
http-01 challenge for bravoga.duckdns.org
Using default address 80 for authentication.
Waiting for verification...
Cleaning up challenges
Could not automatically find a matching server block for bravoga.duckdns.org. Set the 'server_name' directive to use the Nginx installer.

IMPORTANT NOTES:
- Unable to install the certificate
- Congratulations! Your certificate and chain have been saved at:
  /etc/letsencrypt/live/bravoga.duckdns.org/fullchain.pem
  Your key file has been saved at:
  /etc/letsencrypt/live/bravoga.duckdns.org/privkey.pem
  Your cert will expire on 2022-04-13. To obtain a new or tweaked
  version of this certificate in the future, simply run certbot again
  with the "certonly" option. To non-interactively renew *all* of
  your certificates, run "certbot renew"
- Your account credentials have been saved in your Certbot
  configuration directory at /etc/letsencrypt. You should make a
  secure backup of this folder now. This configuration directory will
  also contain certificates and private keys obtained by Certbot so
  making regular backups of this folder is ideal.
ubuntu@ip-172-31-80-200:~$

```

Volvemos a configurar el fichero del servidor. Debería ser idéntico a lo siguiente.

```

server{
    server_name bravoga.duckdns.org;
    root /var/www/certificado;

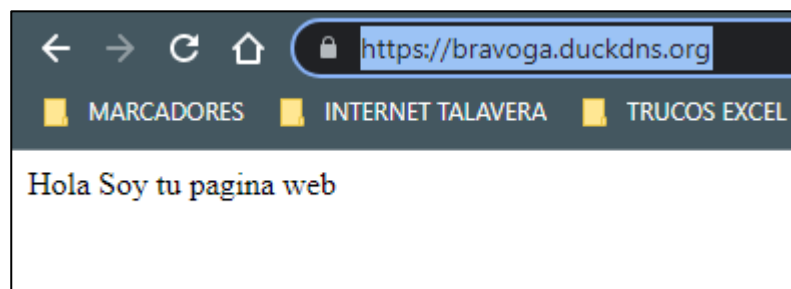
    listen 443 ssl; # managed by Certbot
    ssl_certificate /etc/letsencrypt/live/bravoga.duckdns.org/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/bravoga.duckdns.org/privkey.pem;
    include /etc/letsencrypt/options-ssl-nginx.conf; #managed by Certbot
    ssl_dhparam /etc/letsencrypt/ssl-dhparams.pem;
}

server{
    if ($host = bravoga.duckdns.org) {
        return 301 https://$host$request_uri;
    }

    listen 80;
    server_name bravoga.duckdns.org;
    return 404;
}

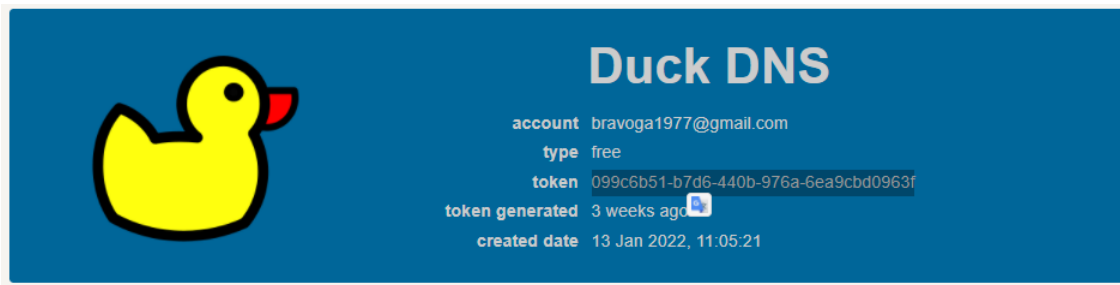
```

Accedemos a nuestro dominio con https:// y comprobamos que nos ha redirigido correctamente y que la página tiene una conexión segura.



7. DNS AUTOMÁTICO CON DUCKDNS

Para configurar la DNS de modo automático accedemos a nuestra cuenta de Duck DNS y en la página principal copiamos el token que aparece.



Verificamos con curl, el token y la localización de duckdns.

`curl -v 'https://www.duckdns.org/update?domains=bravoga.duckdns.org&token=099c6b51-b7d6-440b-976a-6ea9cbd0963f'`

```
ubuntu@ip-172-31-80-200:/var/www/certificado$ curl -v 'https://www.duckdns.org/update?domains=bravoga.duckdns.org&token=099c6b51-b7d6-440b-976a-6ea9cbd0963f'
* Trying 3.96.249.198:443...
* TCP_NODELAY set
* Connected to www.duckdns.org (3.96.249.198) port 443 (#0)
* ALPN, offering h2
* ALPN, offering http/1.1
* successfully set certificate verify locations:
```

Ahora debemos realizar 2 procesos. Crear un servicio y un timer asociados al nombre duckdns.

1. Creación del servicio

Primero creamos un directorio donde se guardarán todos los ficheros para el servicio y el timer.

```
ubuntu@ip-172-31-80-200:~$ mkdir duckdns
```

```
ubuntu@ip-172-31-80-200:~$ cd duckdns
```

Ahora crearemos un fichero llamado duckdns con extensión *.sh

```
ubuntu@ip-172-31-80-200:~/duckdns$ nano duckdns.sh
```

editamos el fichero duckdns.sh con el siguiente contenido.

```
#!/usr/bin/env bash
DUCKDNS_TOKEN='099c6b51-b7d6-440b-976a-6ea9cbd0963f'
DUCKDNS_DOMAINS='bravoga.duckdns.org'
echo "DuckDNS updating domains: ${DUCKDNS_DOMAINS}"
curl https://www.duckdns.org/update?domains=${DUCKDNS_DOMAINS}&token=${DUCKDNS_TOKEN}
```

```
#!/usr/bin/env bash
DUCKDNS_TOKEN='099c6b51-b7d6-440b-976a-6ea9cbd0963f'
DUCKDNS_DOMAINS='bravoga.duckdns.org'
echo "DuckDNS updating domains: ${DUCKDNS_DOMAINS}"
curl "https://www.duckdns.org/update?domains=${DUCKDNS_DOMAINS}&token=${DUCKDNS_TOKEN}"
```

Necesitamos proporcionarle derechos de ejecución con el comando chmod

sudo chmod +x duckdns.sh

y creamos el fichero de servicio (duckdns.service) con el siguiente contenido

```
#DuckDNS update service
[Unit]
Description=DuckDNS update
After=network.target

[Service]
Type=simple
ExecStart=/usr/local/bin/duckdns.sh
RemainAfterExit=false

[Install]
WantedBy=multi-user.target
```

```
#DuckDNS update service
[Unit]
Description=DuckDNS update
After=network.target

[Service]
Type=simple
ExecStart=/usr/local/bin/duckdns.sh
RemainAfterExit=false

[Install]
WantedBy=multi-user.target
```

y para poder instalar el servicio creamos dentro del directorio en el que estamos (~/duckdns\$) el siguiente script llamado install.sh con el siguiente contenido.

```
#!/usr/bin/env bash
cp duckdns.sh /usr/local/bin/
cp duckdns.service /etc/systemd/system/
```

```
#!/usr/bin/env bash
cp duckdns.sh /usr/local/bin/
cp duckdns.service /etc/systemd/system/
```

le damos permisos de ejecución y lo instalamos

sudo chmod +x install.sh

```
ubuntu@ip-172-31-94-176:~/duckdns$ chmod +x install.sh
```

sudo ./install.sh

```
ubuntu@ip-172-31-94-176:~/duckdns$ sudo ./install.sh
```

Iniciamos el servicio y verificamos su estado con,

sudo Systemctl start duckdns.service

sudo Systemctl status duckdns.service

```
● duckdns.service - DuckDNS update
   Loaded: loaded (/etc/systemd/system/duckdns.service; disabled; vendor p
   Active: inactive (dead)

Jan 31 17:20:21 ip-172-31-94-176 duckdns.sh[14755]: [158B blob data]
Jan 31 17:20:21 ip-172-31-94-176 duckdns.sh[14755]: OK
Jan 31 17:20:21 ip-172-31-94-176 systemd[1]: duckdns.service: Succeeded.
Jan 31 17:20:28 ip-172-31-94-176 systemd[1]: Started DuckDNS update.
Jan 31 17:20:28 ip-172-31-94-176 duckdns.sh[14764]: DuckDNS updating domain:
Jan 31 17:20:28 ip-172-31-94-176 duckdns.sh[14765]: % Total % Received
Jan 31 17:20:28 ip-172-31-94-176 duckdns.sh[14765]:
Jan 31 17:20:28 ip-172-31-94-176 duckdns.sh[14765]: [158B blob data]
```

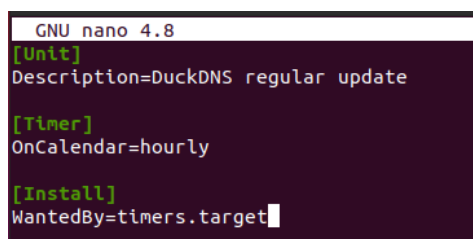

2. Creación del timer

Creamos un nuevo fichero en la misma carpeta que en el paso anterior y lo editamos con este contenido.

sudo touch duckdns.timer

sudo nano duckdns.timer

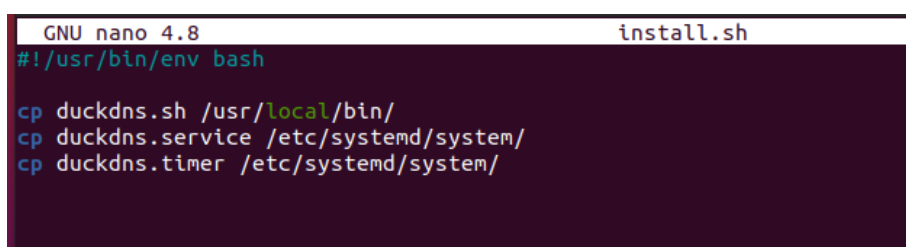
```
[Unit]
Description=DuckDNS regular update
[Timer]
OnCalendar=hourly
[Install]
WantedBy=timers.target
```

A screenshot of the GNU nano 4.8 text editor. The file being edited is duckdns.timer. The content shown is: [Unit] Description=DuckDNS regular update [Timer] OnCalendar=hourly [Install] WantedBy=timers.target. The cursor is at the end of the last line.

```
GNU nano 4.8
[Unit]
Description=DuckDNS regular update
[Timer]
OnCalendar=hourly
[Install]
WantedBy=timers.target
```

Ahora debemos añadir en el fichero install.sh la ruta al timer de la siguiente manera.

```
#!/usr/bin/env bash
cp duckdns.sh /usr/local/bin/
cp duckdns.service /etc/systemd/system/
cp duckdns.timer /etc/systemd/system/
```

A screenshot of the GNU nano 4.8 text editor. The file being edited is install.sh. The content shown is: #!/usr/bin/env bash cp duckdns.sh /usr/local/bin/ cp duckdns.service /etc/systemd/system/ cp duckdns.timer /etc/systemd/system/. The cursor is at the end of the last line.

```
GNU nano 4.8                                install.sh
#!/usr/bin/env bash
cp duckdns.sh /usr/local/bin/
cp duckdns.service /etc/systemd/system/
cp duckdns.timer /etc/systemd/system/
```

Y tras esto volvemos a instalar el script install.sh con sudo ./install.sh

Hecho esto la ip de AWS se actualizará sola en nuestro dominio bravoga.duckdns.org

8. DESPLIEGUE MANUAL DE UN PROYECTO PHP EN AWS

Para el despliegue Manual de una aplicación en PHP usando otro dominio diferente al ya creado (bravoga.duckdns.org) procedemos primero a crear ese nuevo dominio al que llamaremos, bravoga1977.duckdns.org.

domain	current ip	ipv6	changed
bravoga	34.227.161.3 update ip	<input type="text" value="ipv6 address"/> update ipv6	2 hours ago delete domain
bravoga1977	34.227.161.3 update ip	<input type="text" value="ipv6 address"/> update ipv6	0 seconds ago delete domain

This site is protected by reCAPTCHA and the Google [Privacy Policy](#) and [Terms](#)

Desde este punto tendremos que realizar diferentes procesos,

1. Crear un servicio systemctl para automatizar el arranque de la máquina y hacer que el DNS cambie dinámicamente.
2. Hacer un despliegue manual con scp del proyecto, desde nuestro equipo local al servidor de AWS.
3. Crear un certificado para que funcione con un acceso HTTPS.

1. Para hacer que la **DNS cambie de forma automática** solamente tendremos que acceder al fichero de home, duckdns con extensión *.sh

y lo editamos con el siguiente contenido (sólo habrá que añadir el nuevo subdominio en la tercera fila separándolo del anterior por una coma).

```
#!/usr/bin/env bash
DUCKDNS_TOKEN='099c6b51-b7d6-440b-976a-6ea9cbd0963f'
DUCKDNS_DOMAINS='bravoga.duckdns.org','bravoga1977.duckdns.org'
echo "DuckDNS updating domains: ${DUCKDNS_DOMAINS}"
curl https://www.duckdns.org/update?domains=${DUCKDNS_DOMAINS}&token=${DUCKDNS_TOKEN}
```

```
#!/usr/bin/env bash
DUCKDNS_TOKEN='099c6b51-b7d6-440b-976a-6ea9cbd0963f'
DUCKDNS_DOMAINS='bravoga.duckdns.org','bravoga1977.duckdns.org'
echo "DuckDNS updating domains: ${DUCKDNS_DOMAINS}"
curl https://www.duckdns.org/update?domains=${DUCKDNS_DOMAINS}&token=${DUCKDNS_TOKEN}
```

Verificamos el estado del servicio y si no está activo lo activamos con,

`sudo systemctl start duckdns.service`

`sudo systemctl status duckdns.service`

Si surgiera algún problema al inicial el servicio, bastará con utilizar el comando.

`sudo systemctl daemon-reload`

Y tras esto volvemos a instalar el script install.sh con `sudo ./install.sh`

Hecho esto la DNS de AWS se actualizará automáticamente para los subdominios,

bravoga.duckdns.org

bravoga1977.duckdns.org

domain	current ip
bravoga	44.202.17.22 update ip
bravoga1977	44.202.17.22 update ip

This site is protected by reCAPTCHA and the Google Privacy Policy and

2. Para hacer el **despliegue Manual** de un proyecto tendremos que mover el proyecto al servidor desde donde lo tengamos. En mi caso se encuentra en la máquina local.



Para enviarla al servidor necesitaremos utilizar el comando scp (Secure, Contain, Protect) utilizando el comando siguiente desde la carpeta donde se encuentre nuestro proyecto en la máquina local,

`scp -r CRUD despliegue:`

esto copiará la carpeta de nuestro proyecto en la carpeta del usuario del servidor

```

alberto@alberto:~/aplicacion$ scp -r CRUD despliegue:
crudNotas.php                                100% 2366
filtrado.php                                100% 169
singleton.php                                100% 1267
alumnos.png                                  100% 23K
asignaturas.png                             100% 130K
notas.png                                    100% 40K
alumnos.php                                  100% 7602
tablas.sql                                   100% 1407
index.php                                    100% 1345
crudAlumnos.php                             100% 1636
asignaturas.php                             100% 5809
crudAsignaturas.php                         100% 1589
notas.php                                    100% 5763
alberto@alberto:~/aplicacion$

```

```

ubuntu@ip-172-31-80-200:~$ ls
CRUD  duckdns  latest.tar.gz
ubuntu@ip-172-31-80-200:~$

```

Ahora para poder utilizar correctamente nuestra aplicación, que depende de una base de datos MySQL debemos acceder a dicha base de datos en nuestro servidor y crear la base de datos y tablas necesarias.

```

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

MariaDB [(none)]> use crud
Database changed
MariaDB [crud]> CREATE TABLE `alumnos` (
  -> `ID` int(11) NOT NULL AUTO_INCREMENT,
  -> `NOMBRE` varchar(250) DEFAULT NULL,
  -> `FECHA_NACIMIENTO` date DEFAULT NULL,
  -> `MAYOR_EDAD` tinyint(4) DEFAULT NULL,
  -> PRIMARY KEY (`ID`)
  -> ) ;
Query OK, 0 rows affected (0.028 sec)

```

```

MariaDB [crud]> desc alumnos;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| ID    | int(11) | NO | PRI | NULL | auto_increment |
| NOMBRE | varchar(250) | YES | | NULL | |
| FECHA_NACIMIENTO | date | YES | | NULL | |
| MAYOR_EDAD | tinyint(4) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.002 sec)

```

```
MariaDB [crud]> CREATE TABLE `asignaturas` (
-> `ID` int(11) NOT NULL AUTO_INCREMENT,
-> `NOMBRE` varchar(250) NOT NULL,
-> `CURSO` varchar(250) NOT NULL,
-> `CICLO` varchar(250) NOT NULL,
-> PRIMARY KEY (`ID`)
-> );
Query OK, 0 rows affected (0.017 sec)
```

```
MariaDB [crud]> desc asignaturas;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| ID    | int(11)       | NO   | PRI | NULL    | auto_increment |
| NOMBRE | varchar(250)  | NO   |     | NULL    |                |
| CURSO  | varchar(250)  | NO   |     | NULL    |                |
| CICLO  | varchar(250)  | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.001 sec)
```

```
MariaDB [crud]> CREATE TABLE `notas` (
-> `ID_ALUMNO` int(11) NOT NULL,
-> `ID_ASIGNATURA` int(11) NOT NULL,
-> `NOTA` int(11) NOT NULL,
-> PRIMARY KEY (`ID_ALUMNO`, `ID_ASIGNATURA`),
-> KEY `NOTAS_ASIGNATURAS_idx` (`ID_ASIGNATURA`),
-> CONSTRAINT `NOTAS_ALUMNOS` FOREIGN KEY (`ID_ALUMNO`) REFERENCES `alumnos` (`id`),
-> CONSTRAINT `NOTAS_ASIGNATURAS` FOREIGN KEY (`ID_ASIGNATURA`) REFERENCES `asignaturas` (`id`)
-> );
Query OK, 0 rows affected (0.017 sec)
```

```
MariaDB [crud]> desc notas
-> ;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| ID_ALUMNO      | int(11)       | NO   | PRI | NULL    |                |
| ID_ASIGNATURA  | int(11)       | NO   | PRI | NULL    |                |
| NOTA           | int(11)       | NO   |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.001 sec)
```

Creamos un usuario limitado con derechos de consulta, inserción, modificación y borrado.

```
CREATE USER 'admincrud'@'localhost' IDENTIFIED BY '****';
```

```
+-----+
| Grants for admincrud@localhost
+-----+
+-----+
| GRANT USAGE ON *.* TO `admincrud`@`localhost` IDENTIFIED BY PASSWORD '*A4B6157319038724E3560894F7F932C8886EBFCF' |
| GRANT SELECT, INSERT, UPDATE, DELETE ON `crud`.* TO `admincrud`@`localhost` |
+-----+
```

y procedemos a insertar algunos datos en cada tabla.

```
MariaDB [crud]> INSERT INTO `alumnos` VALUES (1,'PEPE','1999-08-09',1),(2,'ANA',  
'2000-12-20',0);  
Query OK, 2 rows affected (0.004 sec)  
Records: 2 Duplicates: 0 Warnings: 0  
  
MariaDB [crud]> INSERT INTO `asignaturas` VALUES (1, 'DWES', 'SEGUNDO', 'DAW'),  
-> (2, 'DESPLIEGUE', 'SEGUNDO', 'DAW');  
Query OK, 2 rows affected (0.003 sec)  
Records: 2 Duplicates: 0 Warnings: 0  
  
MariaDB [crud]> INSERT INTO `notas` VALUES (1, 1, 5), (1,2, 8), (2,1,4), (2,2,7)  
;  
Query OK, 4 rows affected (0.003 sec)  
Records: 4 Duplicates: 0 Warnings: 0
```

Creando la base de datos vamos a mover nuestra aplicación a la carpeta var/www

sudo mv CRUD /var/www/crud

```
ubuntu@ip-172-31-80-200:~$ sudo mv CRUD /var/www/crud  
ubuntu@ip-172-31-80-200:~$
```

```
ubuntu@ip-172-31-80-200:~$ cd /var/www  
ubuntu@ip-172-31-80-200:/var/www$ ls  
certificado crud html web wordpress
```

Cambiamos el propietario del directorio, ahora es Ubuntu y tiene que ser www-data:

sudo chown -R www-data:www-data crud

para poder editar los ficheros deberemos volver a cambiarle el propietario a ubuntu que es el de AWS .

sudo chown -R ubuntu:ubuntu / crud

```
ubuntu@ip-172-31-80-200:/var/www$ sudo chown -R www-data:www-data crud  
ubuntu@ip-172-31-80-200:/var/www$ ls -la  
total 28  
drwxr-xr-x  7 root      root      4096 Feb 24 22:48 .  
drwxr-xr-x 14 root      root      4096 Jan 11 12:42 ..  
drwxr-xr-x  2 www-data www-data 4096 Feb  6 21:28 certificado  
drwx----- 2 www-data www-data 4096 Feb 24 22:41 crud  
drwxr-xr-x  2 root      root      4096 Jan 11 12:42 html  
drwxr-xr-x  4 www-data www-data 4096 Feb 12 20:50 web  
drwxr-xr-x  5 www-data www-data 4096 Jan 13 10:50 wordpress  
ubuntu@ip-172-31-80-200:/var/www$
```

Y ahora procedemos a crear un nuevo servidor para este subdominio.

Crearemos el servidor con el nombre de serverCrud en la carpeta,

etc/nginx/sites-available

El servidor deberá contener los siguientes datos,

```
server{  
    listen 80;  
  
    server_name bravoga1977.duckdns.org;  
  
    root /var/www/crud;  
  
    index index.php index.html;  
  
    location ~\.php$ {  
        include fastcgi.conf;  
  
        fastcgi_intercept_errors on;  
  
        fastcgi_pass unix:/run/php/php7.4-fpm.sock;  
    }  
}
```

```
tu@ip-172-31-80-200:/etc/nginx/sites-available$ sudo nano serverCrud  
tu@ip-172-31-80-200:/etc/nginx/sites-available$ cat serverCrud  
server{  
    listen 80;  
    server_name bravoga1977.duckdns.org;  
    root /var/www/crud;  
    index index.php index.html;  
  
    location ~\.php$ {  
        include fastcgi.conf;  
        fastcgi_intercept_errors on;  
        fastcgi_pass unix:/run/php/php7.4-fpm.sock;  
    }  
}
```

Ahora creamos un enlace directo al servidor en la carpeta,

etc/nginx/sites-enabled con el comando

sudo ln -s /etc/nginx/sites-available/serverCrud


```
lrwxrwxrwx 1 root root 37 Feb 24 23:07 serverCrud -> /etc/nginx/sites-available/serverCrud  
ubuntu@ip-172-31-80-200:/etc/nginx/sites-enabled$
```

y accedemos a la aplicación por medio del subdominio

bravoga1977.duckdns.org



No obstante, el acceso se ha realizado por HTTP y ahora quiero hacerlo más seguro por medio de un acceso HTTPS.

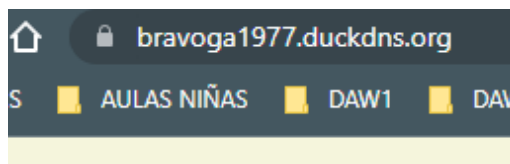
 No es seguro | bravoga1977.duckdns.org

Como ya tenemos un certificado instalado deberemos instalar uno nuevo para este subdominio. El certificado lo crearemos del siguiente modo.

sudo certbot --nginx -d bravoga1977.duckdns.org

```
ubuntu@ip-172-31-80-200:~$ sudo certbot --nginx -d bravoga1977.duckdns.org
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Plugins selected: Authenticator nginx, Installer nginx
Obtaining a new certificate
Performing the following challenges:
http-01 challenge for bravoga1977.duckdns.org
Waiting for verification...
Cleaning up challenges
Deploying Certificate to VirtualHost /etc/nginx/sites-enabled/serverCrud

Please choose whether or not to redirect HTTP traffic to HTTPS, removing HTTP access.
- - - - -
1: No redirect - Make no further changes to the webserver configuration.
2: Redirect - Make all requests redirect to secure HTTPS access. Choose this for
new sites, or if you're confident your site works on HTTPS. You can undo this
change by editing your web server's configuration.
- - - - -
Select the appropriate number [1-2] then [enter] (press 'c' to cancel): 2
Redirecting all traffic on port 80 to ssl in /etc/nginx/sites-enabled/serverCrud
- - - - -
Congratulations! You have successfully enabled https://bravoga1977.duckdns.org
```



Para más subdominios u otras aplicaciones bastará con repetir los pasos indicados anteriormente en este manual.

9. BIBLIOGRAFÍA

- Apuntes de clase.
- GONZÁLEZ, SANTIAGO. Material online de la asignatura de “Despliegue de Aplicaciones Web” (segundo curso del Grado Superior de Desarrollo de Aplicaciones Web - DAW)
<<https://sgz.es/despliegue/>>
- *Web oficial Duckdns* . < <https://www.duckdns.org/spec.jsp>>
- ¿Cómo tener varios dominios en un servidor Web?
<https://alexpro.sytes.net/varios-dominios-en-un-mismo-servidor/>