



DESPLIEGUE TAREAS

Manual de instalación y verificación



17 DE MARZO DE 2022

JEAN CRISTHOER CANTORAL FLORIAN
2 DAW DESPLIEGUE

CREACION DE LA INSTANCIA:

Primero tenemos que crear una nueva instancia ,que es con la que trabajaremos .

En la imagen de Amazon Machine elegimos la de Ubuntu

Se le ha invitado a probar una iteración anticipada del asistente de nueva instancia de lanzamiento. Seguiremos mejorando la experiencia en los próximos meses. Le pedimos a un pequeño grupo de clientes que nos envíe sus comentarios sobre esta versión anticipada. Para salir del asistente de nueva instancia de lanzamiento en cualquier momento, elija el botón **Cancelar**.

1. Elija AMI 2. Elegir tipo de instancia 3. Configurar la instancia 4. Adición de almacenamiento 5. Agregar etiquetas 6. Página Configure Security Group 7. Análisis

Paso 1: Elegir una imagen de Amazon Machine (AMI)

Una AMI es una plantilla que contiene la configuración de software (sistema operativo, servidor de aplicaciones y aplicaciones) necesaria para lanzar la instancia. Puede seleccionar una AMI proporcionada por AWS, nuestra comunidad de usuarios o AWS Marketplace, o puede seleccionar una de sus propias AMI.

Q Para buscar una AMI, escriba un término de búsqueda; por ejemplo, "Windows"

Buscar por parámetro de Systems Manager

Inicio rápido < 1 a 45 de 45 AMI >

Mis AMI
AWS Marketplace
AMI de la comunidad

☐ Solo capa gratuita

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type - ami-0c02fb55956c7d316 (64 bits x86) / ami-03190fe20ef6b1419 (64 bits Arm) **Seleccionar**

Amazon Linux **Apto para la capa**

Amazon Linux 2 incluye cinco años de soporte. Proporciona el kernel de Linux 5.10 adaptado para un rendimiento óptimo en Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1 y en los últimos paquetes de software a través de complementos.

Tipo de dispositivo raíz: ebs Tipo de virtualización: hvm Habilitado para ENA: Sí

Amazon Linux 2 AMI (HVM) - Kernel 4.14, SSD Volume Type - ami-03e0b06f01d45a4eb (64 bits x86) / ami-018d50b368e796499 (64 bits Arm) **Seleccionar**

Amazon Linux **Apto para la capa**

Amazon Linux 2 incluye cinco años de soporte. Proporciona el kernel de Linux 4.14 adaptado para un rendimiento óptimo en Amazon EC2, systemd 219, GCC 7.3, Glibc 2.26, Binutils 2.29.1 y en los últimos paquetes de software a través de complementos.

Tipo de dispositivo raíz: ebs Tipo de virtualización: hvm Habilitado para ENA: Sí

macOS Monterey 12.2.1 - ami-03f795d99e0a6256e **Seleccionar**

The macOS Monterey AMI is an EBS-backed, AWS-supported image. This AMI includes the AWS Command Line Interface, Command Line Tools for Xcode, Amazon SSM Agent, and Homebrew. The AWS Homebrew Tap includes the latest versions of Ruby, Python, and Node.js.

64 bits (Mac)

Paso siguiente tenemos que configurar los grupos de seguridad donde añadiremos tipo https y el tipo http .

1. Elija AMI 2. Elegir tipo de instancia 3. Configurar la instancia 4. Adición de almacenamiento 5. Agregar etiquetas 6. Página Configure Security Group 7. Análisis

Paso 6: Página Configure Security Group

Un grupo de seguridad es un conjunto de reglas del firewall que controlan el tráfico de la instancia. En esta página, puede agregar reglas para permitir que determinado tráfico llegue a la instancia. Por ejemplo, si desea configurar un servidor web y permitir que el tráfico de Internet llegue a la instancia, agregue reglas que permitan el acceso sin restricción a los puertos HTTP y HTTPS. Puede crear un nuevo grupo de seguridad o seleccionar uno existente a continuación. [Más información](#) sobre los grupos de seguridad de Amazon EC2.

Asignar un grupo de seguridad: ☒ Crear un nuevo grupo de seguridad ☐ Seleccionar un grupo de seguridad existente

Nombre del grupo de seguridad:

Descripción:

Tipo	Protocolo	Rango de puertos	Origen	Descripción
SSH	TCP	22	Personaliz... 0.0.0.0/0	por ejemplo SSH for Admin Desk
HTTPS	TCP	443	Cualquier li... 0.0.0.0/0, :::0	por ejemplo SSH for Admin Desk
HTTP	TCP	80	Cualquier li... 0.0.0.0/0, :::0	por ejemplo SSH for Admin Desk

Añadir regla

Aviso

Las reglas con el origen 0.0.0.0/0 permiten que todas las direcciones IP tengan acceso a la instancia. Le recomendamos que configure las reglas del grupo de seguridad para permitir el acceso únicamente desde direcciones IP conocidas.

Creamos un par de claves

×

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

Y ya estaría creada la instancia .

 Servicios

Buscar servicios, características, blogs, document [Alt+S]

Norte de Virginia

voclabs/user1669540-Jean_Cantoral @ 0672-7175-9696

Página Launch Status

✓ Se está lanzando su instancia

Se ha iniciado el siguiente lanzamiento de instancia: i-0a929964d2fc5d6f2 Ver log de lanzamiento

ℹ Recibir notificaciones de los cargos estimados

Crear alertas de facturación para obtener una notificación por correo electrónico cuando los cargos estimados de su factura de AWS superen el importe definido (por ejemplo, cuando se excede la capa de uso gratuita).

Cómo conectarse a la instancia

Se está lanzando su instancia. Pueden transcurrir unos minutos hasta que tenga el estado **en ejecución**, momento en el cual estará lista para poder usarla. Las horas de uso de la nueva instancia comenzarán inmediatamente y seguirán devengando gastos hasta que detenga o termine la instancia.

Haga clic en **Ver las instancias** para monitorizar el estado de su instancia. Cuando la instancia tenga el estado **en ejecución**, podrá **conectarse** a ella desde la pantalla Instancias. [Más información](#) cómo conectarse a la instancia.

▼ Aquí tiene algunos recursos útiles que le ayudarán a comenzar

- Cómo conectarse a la instancia Linux
- Más información sobre la capa de uso gratuita de AWS

- Amazon EC2: Guía del usuario
- Amazon EC2: Foro de debate

Mientras se están lanzando sus instancias, también puede

- Crear alarmas de comprobación de estado para recibir notificaciones cuando estas instancias no superen las comprobaciones de estado. (Podrían aplicarse cargos adicionales)
- Crear y asociar volúmenes de EBS adicionales (Podrían aplicarse cargos adicionales)
- Administrar grupos de seguridad

Ver instancias

Para poder entrar a nuestra maquina virtual con nuestra clave ssh primero tenemos que darle permisos a nuestra llave .

```
jeanc@DESKTOP-DUGCA02 MINGW64 ~  
$ chmod 777 jean.pem
```

Una vez dado los permisos ya podemos acceder

```
jeanc@DESKTOP-DUGCA02 MINGW64 ~  
$ ssh -i jean.pem ubuntu@3.95.165.245  
The authenticity of host '3.95.165.245 (3.95.165.245)' can't be established.  
ED25519 key fingerprint is SHA256:6UgIFF9Y2Av9efq15aaUVHsKdZzqJQGHYUz0t6DC9Fw.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes  
Warning: Permanently added '3.95.165.245' (ED25519) to the list of known hosts.  
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.11.0-1022-aws x86_64)  
  
* Documentation:  https://help.ubuntu.com  
* Management:    https://landscape.canonical.com  
* Support:        https://ubuntu.com/advantage  
  
System information as of Wed Mar 16 16:30:57 UTC 2022  
  
System load:  0.08             Processes:            109  
Usage of /:   18.2% of 7.69GB   Users logged in:     0  
Memory usage: 5%              IPv4 address for eth0: 172.31.94.13
```

INSTALACION DE PHP:

Antes de cualquier tipo de instalación tenemos que actualizar nuestra maquina virtual con el siguiente comando `sudo apt update` , y paso siguiente hacemos un `sudo apt upgrade` para actualizar el grupo de paquetes descargados con el anterior comando , en concreto el upgrade hará una instalación de las nuevas versiones respetando la configuración del software.

Instalamos php-fpm

```
ubuntu@ip-172-31-94-13:~$ sudo apt install php-fpm  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following package was automatically installed and is no longer required:  
  libfwupdplugin1  
Use 'sudo apt autoremove' to remove it.  
The following additional packages will be installed:  
  php-common php7.4-cli php7.4-common php7.4-fpm php7.4-json php7.4-opcache  
  php7.4-readline  
Suggested packages:  
  php-pear  
The following NEW packages will be installed:  
  php-common php-fpm php7.4-cli php7.4-common php7.4-fpm php7.4-json
```

Instalamos php-mysql

```
ubuntu@ip-172-31-94-13:~$ sudo apt install php-mysql  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following package was automatically installed and is no longer required:  
  libfwupdplugin1  
Use 'sudo apt autoremove' to remove it.  
The following additional packages will be installed:  
  php7.4-mysql  
The following NEW packages will be installed:  
  php-mysql php7.4-mysql  
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.  
Need to get 123 kB of archives.  
After this operation, 487 kB of additional disk space will be used.  
Do you want to continue? [Y/n] |
```

Una vez instalados para poder arrancar el php tenemos que utilizar el comando enable de tal manera: `sudo systemctl enable --now php (versión del php que se haya instalado)-fpm.service`

Y par comprobar que no ha habido ningún error comprobamos el estado con un `sudo systemctl status php (versión del php que se haya instalado)-fpm.service`

```
ubuntu@ip-172-31-94-13:~$ sudo systemctl enable --now php-fpm.service
Failed to enable unit: Unit file php-fpm.service does not exist.
ubuntu@ip-172-31-94-13:~$ sudo systemctl enable --now php7.4-fpm.service
Synchronizing state of php7.4-fpm.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable php7.4-fpm
ubuntu@ip-172-31-94-13:~$ sudo 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 Wed 2022-03-16 16:44:28 UTC; 1min 0s ago
     Docs: man:php-fpm7.4(8)
    Main PID: 26570 (php-fpm7.4)
    Status: "Processes active: 0, idle: 2, Requests: 0, slow: 0, Traffic: 0req"
      Tasks: 3 (limit: 4693)
     Memory: 6.8M
    CGroup: /system.slice/php7.4-fpm.service
            └─26570 php-fpm: master process (/etc/php/7.4/fpm/php-fpm.conf)
              └─26571 php-fpm: pool www
```

INSTALACION NGINX

Es el mismo procedimiento a la hora de instalar que PHP .

Instalamos nginx con el siguiente comando `sudo apt install -y nginx`

```
ubuntu@ip-172-31-94-13:~$ sudo apt install -y nginx
```

Una vez instalado la iniciamos y comprobamos su estado con el comando `systemctl --no y el status .`

```
ubuntu@ip-172-31-94-13:~$ sudo 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 Wed 2022-03-16 16:51:10 UTC; 32s ago
     Docs: man:nginx(8)
    Main PID: 27144 (nginx)
    Tasks: 3 (limit: 4693)
     Memory: 3.5M
    CGroup: /system.slice/nginx.service
            └─27144 nginx: master process /usr/sbin/nginx -g daemon on; master_process
              └─27145 nginx: worker process
                └─27146 nginx: worker process

Mar 16 16:51:10 ip-172-31-94-13 systemd[1]: Starting A high performance web server and a reverse proxy server: nginx.
Mar 16 16:51:10 ip-172-31-94-13 systemd[1]: Started A high performance web server and a reverse proxy server: nginx.
```

Instalación de MariaDB:

Instalamos el paquete de mariaDB con el comando sudo apt install mariadb-server

```
ubuntu@ip-172-31-94-13:~$ sudo apt install mariadb-server
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfwupdplugin1
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  galera-3 libbcgi-fast-perl libbcgi-pm-perl libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl libencode-locale-perl
  libfcgi-perl libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl
  libio-html-perl liblwp-mediatypes-perl libmysqlclient21 libsnappy1v5 libterm-readkey-perl libtimedate-perl liburi-perl
  mariadb-client-10.3 mariadb-client-core-10.3 mariadb-common mariadb-server-10.3 mariadb-server-core-10.3 mysql-common socat
Suggested packages:
  libclone-perl libldb-perl libnet-daemon-perl libsql-statement-perl libdata-dump-perl libipc-sharedcache-perl libwww-perl
  mailx mariadb-test tinyca
The following NEW packages will be installed:
  galera-3 libbcgi-fast-perl libbcgi-pm-perl libconfig-inifiles-perl libdbd-mysql-perl libdbi-perl libencode-locale-perl
  libfcgi-perl libhtml-parser-perl libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl
  libio-html-perl liblwp-mediatypes-perl libmysqlclient21 libsnappy1v5 libterm-readkey-perl libtimedate-perl liburi-perl
  mariadb-client-10.3 mariadb-client-core-10.3 mariadb-common mariadb-server mariadb-server-10.3 mariadb-server-core-10.3
  mysql-common socat
0 upgraded, 28 newly installed, 0 to remove and 0 not upgraded.
Need to get 21.3 MB of archives.
After this operation, 174 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Una vez instalado la iniciamos con enable y acto seguido tenemos securizar a MariaDB así le ponemos contraseña a root . El comando para securizar es sudo mysql_secure_installation

```
ubuntu@ip-172-31-94-13:~$ sudo mysql_secure_installation

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

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

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

Setting the root password ensures that nobody can log into the MariaDB
root user without the proper authorisation.

Set root password? [Y/n] n
... skipping.

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

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

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

Disallow root login remotely? [Y/n] n
... skipping.
```

Acto seguido comprobamos en estado de MariaDB con el status

```
ubuntu@ip-172-31-94-13:~$ sudo systemctl status mariadb.service
● mariadb.service - MariaDB 10.3.34 database server
   Loaded: loaded (/lib/systemd/system/mariadb.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2022-03-16 16:56:32 UTC; 1min 56s ago
     Docs: man:mysqld(8)
           https://mariadb.com/kb/en/library/systemd/
   Main PID: 27968 (mysqld)
    Status: "Taking your SQL requests now..."
     Tasks: 31 (limit: 4693)
    Memory: 63.0M
   CGroup: /system.slice/mariadb.service
           └─27968 /usr/sbin/mysqld

Mar 16 16:56:32 ip-172-31-94-13 systemd[1]: Starting MariaDB 10.3.34 database server...
Mar 16 16:56:32 ip-172-31-94-13 mysqld[27968]: 2022-03-16 16:56:32 0 [Note] /usr/sbin/mysqld (mysqld 10.3.34-MariaDB-0ubuntu0.22.03.1) started
Mar 16 16:56:32 ip-172-31-94-13 systemd[1]: Started MariaDB 10.3.34 database server.
Mar 16 16:56:32 ip-172-31-94-13 /etc/mysql/debian-start[28003]: Upgrading MySQL tables if necessary.
Mar 16 16:56:32 ip-172-31-94-13 /etc/mysql/debian-start[28006]: Looking for 'mysql' as: /usr/bin/mysql
Mar 16 16:56:32 ip-172-31-94-13 /etc/mysql/debian-start[28006]: Looking for 'mysqlcheck' as: /usr/bin/mysqlcheck
Mar 16 16:56:32 ip-172-31-94-13 /etc/mysql/debian-start[28006]: This installation of MariaDB is already upgraded to 10.3.34-MariaDB-0ubuntu0.22.03.1
Mar 16 16:56:32 ip-172-31-94-13 /etc/mysql/debian-start[28006]: There is no need to run mysql_upgrade again for 10.3.34-MariaDB-0ubuntu0.22.03.1
Mar 16 16:56:32 ip-172-31-94-13 /etc/mysql/debian-start[28006]: You can use --force if you still want to run mysql_upgrade
Mar 16 16:56:32 ip-172-31-94-13 /etc/mysql/debian-start[28018]: Triggering myisam-recover for all MyISAM tables and aria-recover
```

Instalación de WordPress:

Para descargar la última versión de WordPress lo podemos hacer desde la pagina oficial , donde nos aparece descargar le damos a click derecho copiamos la dirección de enlace y con el comando curl -OL <https://es.wordpress.org/latest.tar.gz> acto seguido lo descomprimos con el comando tar xzvf latest.tar.gz y para finalizar movemos la carpeta WordPress que se ha generado al descomprimir el tar a la ruta /var/www .podemos comprobar haciendo un ls hacia ese directorio para ver que hemos movido correctamente la carpeta.

```
ubuntu@ip-172-31-94-13:~$ sudo mv wordpress /var/www/
ubuntu@ip-172-31-94-13:~$ ls /var/www/
```

Para que PHP pueda tener acceso a nuestra carpeta WordPress tenemos que cambiar tanto como grupo y como usuario y esto se le hace con el comando chown

```
ubuntu@ip-172-31-94-13:/var/www$ sudo chown -R www-data:www-data /var/www/wordpress
ubuntu@ip-172-31-94-13:/var/www$ ls -al
total 16
drwxr-xr-x  4 root    root    4096 Mar 16 17:04 .
drwxr-xr-x 14 root    root    4096 Mar 16 16:51 ..
drwxr-xr-x  2 root    root    4096 Mar 16 16:51 html
drwxr-xr-x  5 www-data www-data 4096 Mar 11 00:39 wordpress
ubuntu@ip-172-31-94-13:/var/www$ |
```

1.Una vez instalado tenemos que configurar nuestro servidor:

Primero tenemos que eliminar el enlace por defecto que se crea en sites-enabled ,lo eliminamos con sudo rm es necesario poner el sudo porque tenemos que tener permisos para eliminarlo.

```
ubuntu@ip-172-31-94-13:~$ ls /etc/nginx/sites-enabled/
default
ubuntu@ip-172-31-94-13:~$ |
```

2.Tenemos que crear un archivo con la configuración del servidor:

Creamos en nginx/sites-available un fichero en donde va a ir la siguiente configuración

```
server unix:/run/php/php7.4-fpm.sock;
listen 443 ssl;
ssl_certificate /etc/ssl/certs/wp-manual.crt;
ssl_certificate_key /etc/ssl/private/wp-manual.key;
}
server {
    server_name _;
    root /var/www/wordpress;
    index index.php;
    location = /favicon.ico {
        log_not_found off;
        access_log off;
    }
    location = /fichero.txt {
        allow all;
        log_not_found off;
        access_log off;
    }
    location / {
        try_files $uri $uri/ /index.php?$args;
    }
    location ~ \.php$ {
        include fastcgi.conf;
        fastcgi_intercept_errors on;
        fastcgi_pass php;
    }
    location ~* \.(js|css|png|jpg|jpeg|gif|ico)$ {
        expires max;
        log_not_found off;
    }
}
```


Una vez añadida la configuración tenemos que reiniciar los servicios , para comprobar que la configuración ha sido correcta con `sudo nginx -t` podemos asegurarnos acto seguido comprobamos con un `status` si no ha habido algún problema a la hora de iniciar .

```
ubuntu@ip-172-31-94-13:/etc/nginx/sites-available$ 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-172-31-94-13:/etc/nginx/sites-available$ sudo systemctl reload nginx.service
ubuntu@ip-172-31-94-13:/etc/nginx/sites-available$ sudo systemctl restart php7.4-fpm.service
ubuntu@ip-172-31-94-13:/etc/nginx/sites-available$ sudo systemctl status nginx-service
Unit nginx-service.service could not be found.
ubuntu@ip-172-31-94-13:/etc/nginx/sites-available$ sudo systemctl status nginx
● 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 Wed 2022-03-16 16:51:10 UTC; 1h 19min ago
     Docs: man:nginx(8)
   Process: 29511 ExecReload=/usr/sbin/nginx -g daemon on; master_process on; -s reload (code=exited, status=0)
   Main PID: 27144 (nginx)
    Tasks: 3 (limit: 4693)
   Memory: 3.6M
   CGroup: /system.slice/nginx.service
           └─27144 nginx: master process /usr/sbin/nginx -g daemon on; master_process on;
             └─29512 nginx: worker process
               └─29513 nginx: worker process

Mar 16 16:51:10 ip-172-31-94-13 systemd[1]: Starting A high performance web server and a reverse proxy server..
Mar 16 16:51:10 ip-172-31-94-13 systemd[1]: Started A high performance web server and a reverse proxy server.
Mar 16 18:09:37 ip-172-31-94-13 systemd[1]: Reloading A high performance web server and a reverse proxy server.
Mar 16 18:09:37 ip-172-31-94-13 systemd[1]: Reloaded A high performance web server and a reverse proxy server.
```

3.Una vez configurado WordPress creamos la base de datos y damos permisos al usuario creado.

```
ubuntu@ip-172-31-94-13:/var/www$ sudo mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 43
Server version: 10.3.34-MariaDB-0ubuntu0.20.04.1 Ubuntu 20.04

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

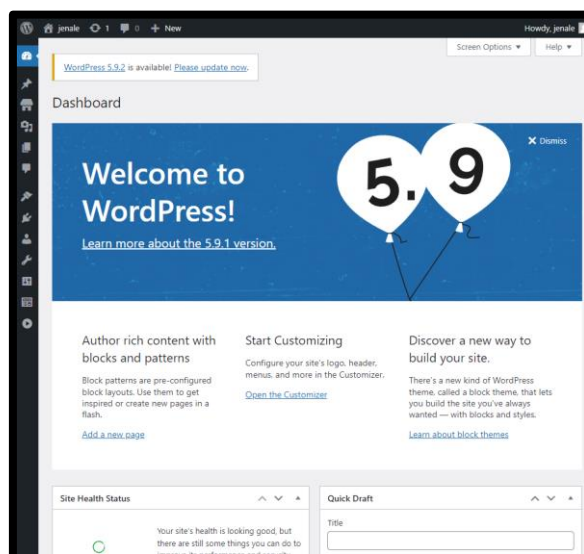
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> GRANT ALL PRIVILEGES ON WordPress.* TO "Jena"@"localhost" IDENTIFIED by "1234";
Query OK, 0 rows affected (0.000 sec)

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

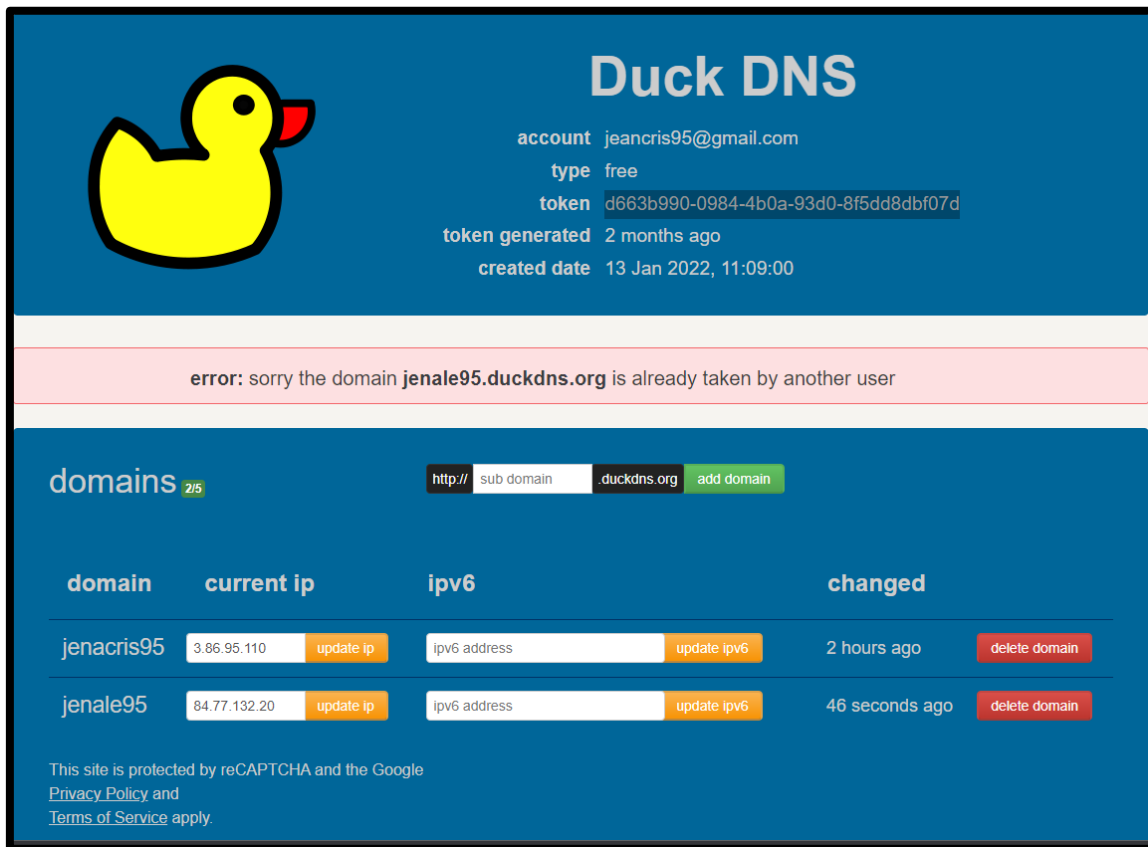
MariaDB [(none)]> |
```

Si todo ha sido correcto para poder entrar a WP lo podremos hacer poniendo nuestra dirección de ip publica en el navegador y debería aparecer el panel para poder rellenar los campos necesarios para empezar con la instalación , una vez entrado nos aparecerá el dashboard



.DNS dinámico con duckdns y creación de un servicio de Systemd para su actualización al arranque de la máquina

Creamos un nuevo dominio en la pagina DUCK DNS



The screenshot shows the Duck DNS website interface. At the top left is a yellow duck logo. To its right, the account details are displayed: account (jeancris95@gmail.com), type (free), token (d663b990-0984-4b0a-93d0-8f5dd8dbf07d), token generated (2 months ago), and created date (13 Jan 2022, 11:09:00). Below this, a red error message states: "error: sorry the domain jenale95.duckdns.org is already taken by another user". The main section is titled "domains 2/5" and features a form to add a new domain with fields for "http://", "sub domain", ".duckdns.org", and an "add domain" button. Below the form is a table listing existing domains:

domain	current ip	ipv6	changed
jenacris95	3.86.95.110 update ip	<input type="text" value="ipv6 address"/> update ipv6	2 hours ago delete domain
jenale95	84.77.132.20 update ip	<input type="text" value="ipv6 address"/> update ipv6	46 seconds ago delete domain

At the bottom, there is a notice: "This site is protected by reCAPTCHA and the Google Privacy Policy and Terms of Service apply."

Acto seguido lo comprobamos si responde ok es que esta todo correcto si pone ko hay algo mal

```
ubuntu@ip-172-31-94-13:/var/www$ curl 'https://www.duckdns.org/update?domains=jenale95.duckdns.org&token=d663b990-0984-4b0a-93d0-8f5dd8dbf07d'
OKubuntu@ip-172-31-94-13:/var/www$ |
```

Una vez comprobado todo eso comprobamos los servicios para eso es recomendable crear una carpeta y crear un fichero .sh

En este caso yo he creado mi fichero serviciodns.sh y he colocado mi dominio y mi token , en el caso de que funcione me saldría el nombre del dominio.

```
ubuntu@ip-172-31-94-13:~/servicio$ cat serviciodns.sh
#!/usr/bin/env bash
DUCKDNS_TOKEN='d663b990-0984-4b0a-93d0-8f5dd8dbf07d'
DUCKDNS_DOMAINS='jenale95.duckdns.org'
echo "${DUCKDNS_DOMAINS}"
ubuntu@ip-172-31-94-13:~/servicio$ |
```

Le damos permisos sudo chmod +x serviciodns.sh así de esta manera lo haremos ejecutable.

Y para ejecutarlo solo nos bastaría con sudo ./serviciodns.sh

Acto seguido creamos un fichero .service en mi caso he creado un servicio.service ,muy importante poner en el ExecStart el nombre del fichero.sh creado previamente si no mas adelante dará un error.

```
ubuntu@ip-172-31-94-13:~/servicio$ cat servicio.service
[Unit]
Description=DuckDNS update
After=network.target
[Service]
Type=simple
ExecStart=/usr/local/bin/serviciodns.sh
RemainAfterExit=false
[Install]
WantedBy=multi-user.target
ubuntu@ip-172-31-94-13:~/servicio$ |
```

Por ultimo creamos un fichero cuya finalidad será copiar lo que se tenga en el.sh al .service

En mi caso he creado un copia.sh

```
ubuntu@ip-172-31-94-13:~/servicio$ cat copia.sh
cp serviciodns.sh /usr/local/bin/
cp servicio.service /etc/systemd/system/
ubuntu@ip-172-31-94-13:~/servicio$ |
```

Por último, le damos los mismos permisos que al .sh el chmod +x para convertirlo en un ejecutable y lo ejecutamos con sudo ./copia.sh .

Una vez tengamos todo esto iniciamos el servicio , lo habilitamos y comprobamos con el status para ver que no haya ningún tipo de error .

```
ubuntu@ip-172-31-94-13:~/servicio$ sudo ./copia.sh
ubuntu@ip-172-31-94-13:~/servicio$ sudo systemctl start servicio.service
ubuntu@ip-172-31-94-13:~/servicio$ sudo systemctl enable --now servicio.service
Created symlink /etc/systemd/system/multi-user.target.wants/servicio.service - /etc/systemd/system/servicio.service.
ubuntu@ip-172-31-94-13:~/servicio$ sudo systemctl status servicio.service
● servicio.service - DuckDNS update
   Loaded: loaded (/etc/systemd/system/servicio.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Wed 2022-03-16 22:36:18 UTC; 12s ago
     Process: 2825 ExecStart=/usr/local/bin/serviciodns.sh (code=exited, status=0/SUCCESS)
    Main PID: 2825 (code=exited, status=0/SUCCESS)

Mar 16 22:36:18 ip-172-31-94-13 systemd[1]: Started DuckDNS update.
Mar 16 22:36:18 ip-172-31-94-13 serviciodns.sh[2825]: jenale95.duckdns.org
Mar 16 22:36:18 ip-172-31-94-13 systemd[1]: servicio.service: Succeeded.
ubuntu@ip-172-31-94-13:~/servicio$ |
```

Ahora tenemos que ponerle un temporizador systemd al Dns

Creamos un archivo temporizador.timer

```
ubuntu@ip-172-31-94-13:~/servicio$ cat temporizador.timer
[Unit]
Description=Run servicio.service every minute

[Timer]
OnBootSec=30
OnUnitActiveSec=1min

[Install]
WantedBy=timers.target
ubuntu@ip-172-31-94-13:~/servicio$ |
```

Agregamos en el archivo copia un cp temporizador.timer /etc/systemd/system/

Acto seguido lo ejecutamos e iniciamos el temporizador.

Con systemctl enable --now temporizador.timer

Funcionamiento HTTPS creando un certificado

Certificado autoafirmado

Primero comenzaremos instalando certbot

```
ubuntu@ip-172-31-94-13:~$ sudo apt install python3-certbot-nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  libfupdplugin1
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  certbot python3-acme python3-certbot python3-certbot-nginx python3-configargparse python3-future python3-icu python3-josepy python3-mock
  python3-parsedatetime python3-pbr python3-pyparsing python3-requests-toolbelt python3-rfc3339 python3-tz
  python3-zope.component python3-zope.event python3-zope.hookable
Suggested packages:
  python3-certbot-apache python-certbot-doc python-acme-doc python-certbot-nginx-doc python-future-doc python-mock-doc
  python-pyparsing-doc
The following NEW packages will be installed:
  certbot python3-acme python3-certbot python3-certbot-nginx python3-configargparse python3-future python3-icu python3-josepy
  python3-mock python3-parsedatetime python3-pbr python3-pyparsing python3-requests-toolbelt python3-rfc3339 python3-tz
  python3-zope.component python3-zope.event python3-zope.hookable
0 upgraded, 18 newly installed, 0 to remove and 0 not upgraded.
Need to get 1264 kB of archives.
After this operation, 6657 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 python3-josepy all 1.2.0-2 [28.1 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 python3-pbr all 5.4.5-0ubuntu1 [64.0 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 python3-mock all 3.0.5-1build1 [25.6 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 python3-requests-toolbelt all 0.8.0-1.1 [35.2 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 python3-tz all 2019.3-1 [24.4 kB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 python3-rfc3339 all 1.1-2 [6808 B]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 python3-acme all 1.1.0-1 [29.6 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 python3-configargparse all 0.13.0-2 [22.6 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 python3-future all 0.18.2-2 [336 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 python3-parsedatetime all 2.4-5 [32.6 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 python3-zope.hookable amd64 5.0.0-1build1 [11.2 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 python3-zope.event all 4.4-2build1 [7704 B]
Get:13 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 python3-zope.component all 4.3.0-3 [38.3 kB]
Get:14 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 python3-certbot all 0.40.0-1ubuntu0.1 [223 kB]
Get:15 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 certbot all 0.40.0-1ubuntu0.1 [17.9 kB]
Get:16 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 python3-pyparsing all 2.4.6-1 [61.3 kB]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 python3-certbot-nginx all 0.40.0-0ubuntu0.1 [5.8 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu focal/main amd64 python3-icu amd64 7.2.4-2.0ubuntu2 [5350 kB]
```

Una vez instalado necesitamos crear una llave

openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout clave_cerbot.key -out clave_cerbot.crt . Una vez creado la llave nos pedirá que rellenemos una serie de datos.

```
ubuntu@ip-172-31-94-13:~$ openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout clave_cerbot.key -out clave_cerbot.crt
Generating a RSA private key
.....+++++
.....+++++
writing new private key to 'clave_cerbot.key'
-----
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]:Madrid
Locality Name (eg, city) []:Madrid
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Despliegue
Organizational Unit Name (eg, section) []:Despliegue
Common Name (e.g. server FQDN or YOUR name) []:jean cristoher cantoral
Email Address []:jeancris@gmail.com
ubuntu@ip-172-31-94-13:~$
```

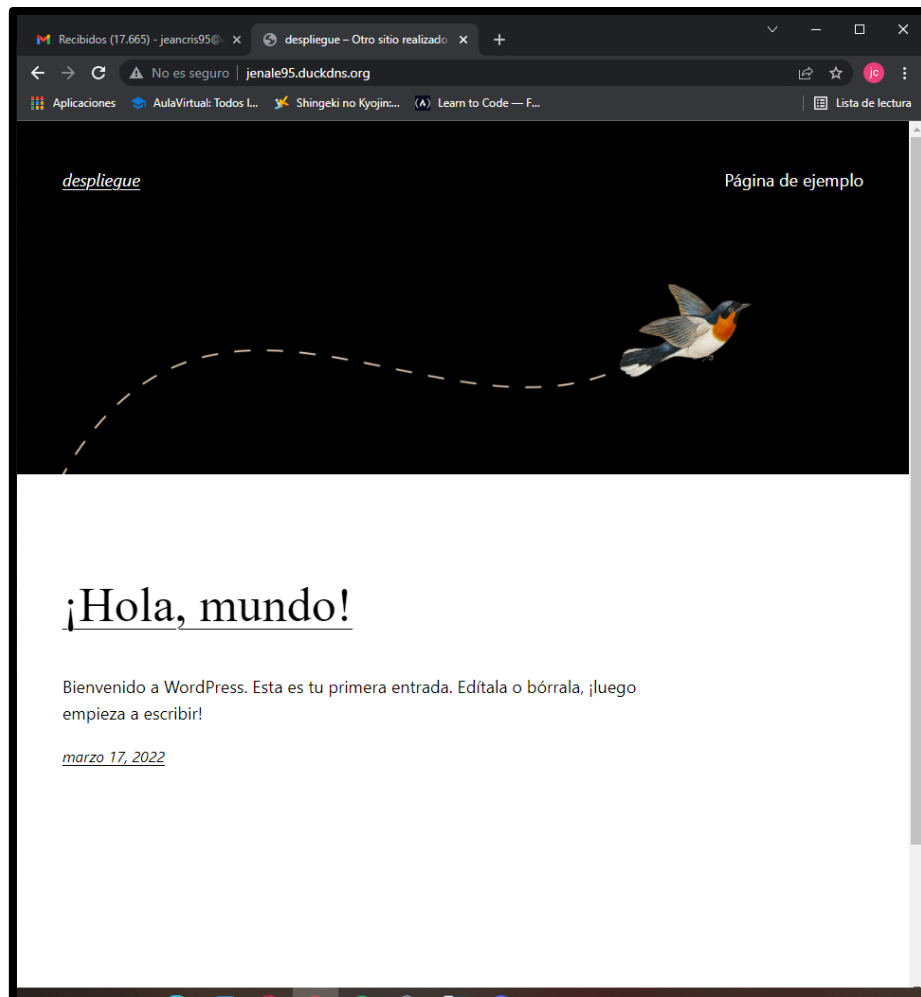
Una vez que se nos haya creado tanto la clave_cerbot.key y la .crt tenemos que moverla cada uno a su directorio correspondiente .

```
ubuntu@ip-172-31-94-13:~$ sudo mv clave_cerbot.key /etc/ssl/private
ubuntu@ip-172-31-94-13:~$ sudo mv clave_cerbot.crt /etc/ssl/certs
ubuntu@ip-172-31-94-13:~$
```

Una vez tenemos todo esto tenemos que añadir en nuestro servidor de WordPress

```
ssl_certificate /etc/ssl/certs/wp-manual.crt;
```

```
ssl_certificate_key /etc/ssl/private/wp-manual.key;
```



Certificado Let's Encrypt

Para poder crear el certificado lo hacemos con el comando:

```
sudo certbot -n --nginx -d jenale95.duckdns.org
```

```

Congratulations! You have successfully enabled https://jenale95.duckdns.org

You should test your configuration at:
https://www.ssllabs.com/ssltest/analyze.html?d=jenale95.duckdns.org
-----
IMPORTANT NOTES:
- Congratulations! Your certificate and chain have been saved at:
  /etc/letsencrypt/live/jenale95.duckdns.org/fullchain.pem
  Your key file has been saved at:
  /etc/letsencrypt/live/jenale95.duckdns.org/privkey.pem
  Your cert will expire on 2022-06-15. 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"
- If you like Certbot, please consider supporting our work by:

  Donating to ISRG / Let's Encrypt: https://letsencrypt.org/donate
  Donating to EFF: https://eff.org/donate-le

ubuntu@ip-172-31-19-101: /etc/nginx/sites-available$
```

pero para que todo esto funcione tenemos que tener configurado en nuestro servidor WordPress como `server_name` nuestro dominio

```
server_name jenale95.duckdns.org;  
root /var/www/wordpress;
```

Si todo ha salido correcto se nos crea en la configuración de nuestro servidor lo siguiente :

```
ssl_certificate /etc/letsencrypt/live/jenale95.duckdns.org/fullchain.pem # managed by Certbot  
ssl_certificate_key /etc/letsencrypt/live/jenale95.duckdns.org/privkey.pem # managed by Certbot
```

Comprobamos que funcione entrando a nuestro dominio



Manual para realizar el despliegue de una aplicación cualquiera en PHP que use MySQL: Con rsync

Primero creamos un fichero config desde nuestro local , este fichero lo creamos en la ruta ./ssh/

```
Host despliegue
  HostName jeandespliegue.duckdns.org
  User ubuntu
  Identify /home/jenale/Escritorio/despliegue.pem
  ServerAliveInterval 60
  ServerAliveCountMax 60
```

Acto seguido le damos los permisos adecuados desde el servidor(tienda es la carpeta donde se guardara nuestros archivos).

```
ubuntu@ip-172-31-94-181:~$ sudo chmod 777 -R /var/www/tienda
```

Desde la maquina local hacemos el rsync

```
jenale@jenale-VirtualBox:~/.ssh$ rsync /home/jenale/Escritorio/tienda/* despliegue:/var/www/tienda/
The authenticity of host 'jeandespliegue.duckdns.org (35.174.173.190)' can't be established.
ECDSA key fingerprint is SHA256:bIXk/3tb++6E3nUlikiq2+T5LuroEx6HU3rg8NHneFs.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'jeandespliegue.duckdns.org' (ECDSA) to the list of known hosts.
```

Y como podemos comprobar los archivos han sido copiados al servidor

```
ubuntu@ip-172-31-94-181:/var/www/tienda$ ls
caja.php  compra.php  entrada.php  f_tienda.php  prueba.php  tienda.php
```

Hooks:

Primero creamos una nueva carpeta en “/var/www” a la que llamamos hook.git le damos los permisos adecuados y cambiamos el grupo al que pertenece , luego iniciamos el repositorio con un git –bare.

```
ubuntu@ip-172-31-94-181:~$ cd /var/www/
ubuntu@ip-172-31-94-181:/var/www$ sudo mkdir hook.git
ubuntu@ip-172-31-94-181:/var/www$ sudo chown ubuntu:ubuntu hook.git
ubuntu@ip-172-31-94-181:/var/www$ sudo chmod 777 hook.git
ubuntu@ip-172-31-94-181:/var/www$ cd hook.git/
ubuntu@ip-172-31-94-181:/var/www/hook.git$ sudo git init --bare
Initialized empty Git repository in /var/www/hook.git/
ubuntu@ip-172-31-94-181:/var/www/hook.git$
```

```
$ git clone despliegue:/var/www/hook
Cloning into 'hook'...
warning: You appear to have cloned an empty repository.
```

Desde Local clonamos el repositorio. en el servidor creamos un directorio en donde tendrá un post-receive con el siguiente contenido

```
#!/bin/bash

# Bare repository directory.
GIT_DIR="/home/ubuntu/escritorio/hook"
# Target directory.
TARGET="/var/www/hook "
while read oldrev newrev ref
do
    BRANCH=$(git rev-parse --symbolic --abbrev-ref $ref)
    if [[ $BRANCH == "main" ]]; then
        echo "Push received! Deploying branch: ${BRANCH}..."
        # deploy to our target directory.
        sudo git --work-tree=$TARGET --git-dir=$GIT_DIR checkout -f
$BRANCH
    else
        echo "Not master branch. Skipping."
    fi
done
```

Le damos todos los permisos con `chmod 777` .