
Práctica Plesk

Despliegue de Aplicaciones Web

José Juan Sánchez Hernández

Curso 2024/2025

Índice

1	Despliegue de aplicaciones web con Plesk	1
1.1	¿Qué es Plesk?	1
1.2	AWS	1
1.2.1	Creación de un grupo de seguridad en AWS	1
1.2.2	Creación de una instancia EC2 en AWS	2
1.2.3	Creación de una IP elástica en AWS	2
1.2.4	Instalación de Plesk en una instancia EC2 de AWS	2
1.2.5	¿Cómo recuperar el password del usuario admin de Plesk?	5
1.3	OpenStack	5
1.3.1	Instalación de Plesk en OpenStack	5
1.3.2	Acceso al panel de administración de Plesk	6
1.4	Gestión de dominios y subdominios en Plesk	7
1.4.1	Configuración de entradas DNS locales en su sistema operativo	7
1.4.2	Gestionar los dominios con algún proveedor de dominios gratuitos	9
1.4.3	Gestionar los dominios con el DNS wildcard del servicio nip.io	9
1.5	Despliegue de una aplicación subiendo los archivos desde el panel de control	10
1.6	Despliegue de una aplicación WordPress	22
1.7	Despliegue de una aplicación web con el asistente de Plesk	23
1.8	Despliegue de una aplicación web con Git	23
1.9	Despliegue de una aplicación web subiendo los archivos por FTP	41
1.10	Despliegue de una aplicación Laravel	45
1.11	Extensiones de Plesk	47
1.12	Entregables	47
2	Referencias	48
3	Licencia	49

Índice de figuras

Índice de cuadros

1 Despliegue de aplicaciones web con Plesk

En esta práctica tendrá que realizar el despliegue de varias aplicaciones web LAMP en un servidor con el software [Plesk](#). El servidor [Plesk](#) estará ejecutándose en una instancia de [OpenStack](#) del cloud privado del instituto, con la última versión de [Ubuntu Server](#).

1.1 ¿Qué es Plesk?

[Plesk](#) es una plataforma software de hosting que permite alojar y administrar sus aplicaciones web en un [servidor virtual privado \(VPS\)](#), en un [servidor dedicado](#) o en un [servidor de un proveedor de servicios cloud](#).

Existen diferentes alternativas de paneles de control para gestionar sitios webs, algunas de ellas son:

- [cPanel](#).
- [DirectAdmin](#).
- [Vesta](#).
- [Webmin](#).

1.2 AWS

1.2.1 Creación de un grupo de seguridad en AWS

En primer lugar vamos a crear un grupo de seguridad llamado: `sg_plesk`.

A continuación, se describen las reglas de entrada y salida que tendrá que configurar en este grupo.

Reglas de entrada

Deberá configurar los puertos que estarán abiertos para poder conectarnos por [SSH](#) y para poder acceder por [HTTP/HTTPS](#). También se recomienda crear una regla para permitir el tráfico [ICMP](#) para poder hacerle [ping](#) a la instancia.

- 22: SSH (TCP)
- 80: HTTP (TCP)
- 443: HTTPS (TCP)
- ICMP

Si vamos a conectarnos por [FTP](#) al servidor, es necesario abrir los puertos 20 Y 21. Plesk utiliza por defecto el **modo activo** para las conexiones [FTP](#) pero si queremos configurar el **modo pasivo**, es necesario abrir un rango de puertos.

- 21: FTP (TCP) - Puerto de control en los modos activo y pasivo
- 20: FTP (TCP) - Puerto de datos en el modo activo
- 49152 – 65535: Rango de puertos dinámicos para el modo pasivo (TCP)

Si vamos a configurar el servidor de correo electrónico, es necesario abrir los puertos de los protocolos de correo saliente (SMTP) y correo entrante (IMAP y POP3).

- 25: SMTP (TCP)
- 465: SMTPS (TCP)
- 143: IMAP (TCP)
- 993: IMAPS (TCP)
- 110: POP3 (TCP)
- 995: POP3S (TCP)

Si va a utilizar [Webhooks](#) de GitHub para hacer [pulls](#) y despliegues automáticos en Plesk, es necesario abrir el puerto:

- 8443: (TCP)

Reglas de salida

Como reglas de salida tendremos una única regla donde vamos a permitir todo el tráfico de salida para cualquier dirección IP destino (0.0.0.0/0).

1.2.2 Creación de una instancia EC2 en AWS

Las características de la instancia EC2 que vamos a crear son las siguientes:

- Nombre de la instancia: [plesk](#).
- Imagen (AMI): Última versión disponible de Ubuntu Server.
- Arquitectura: [x86](#).
- Tipo de instancia: [t2.medium](#) (2 vCPUs, 4 GB de RAM).
- Par de claves: [vockey](#).
- Grupo de seguridad: Crear un grupo de seguridad para plesk.
- Almacenamiento: 30 GB de disco EBS.

1.2.3 Creación de una IP elástica en AWS

Una vez que haya creado la instancia EC2, tendrá que crear una dirección IP elástica y asociarla a la instancia.

1.2.4 Instalación de Plesk en una instancia EC2 de AWS

Una vez que haya creado la instancia EC2, tendrá que conectarse a ella por SSH y realizar la instalación de [Plesk](#).

Existen varias formas de instalar **Plesk** en un servidor, pero en esta práctica vamos a utilizar el script de **instalación en modo desatendido** que proporciona **Plesk**. En la documentación oficial de Plesk puede encontrar un artículo titulado: *Installing Plesk for Linux in Unattended Mode*, con la descripción detallada.

Pasos para realizar la instalación desatendida de Plesk

Como **root** o con un usuario con permisos de **sudo**, ejecute los siguientes comandos:

1. Descargue el script de instalación desatendida de Plesk.

```
1 wget https://autoinstall.plesk.com/plesk-installer
```

2. Le asignamos permisos de ejecución al script.

```
1 chmod +x plesk-installer
```

3. Ejecutamos el script de instalación.

```
1 ./plesk-installer install plesk
```

El proceso de instalación puede tardar unos **10 minutos**. Una vez que finalice el proceso de instalación le aparecerá un mensaje similar a este:

```
1 Congratulations!
2
3 The installation has been finished. Plesk is now running on your server.
4
5 To complete the configuration process, browse either of URLs:
6 * https://frosty-antonelli.54-90-74-86.plesk.page/login?secret=
   rNECTAYl7gCSNhDCraxGuQD_ANbwMIeAP7JLtkUymSTChS5_
7 * https://54.90.74.86/login?secret=
   rNECTAYl7gCSNhDCraxGuQD_ANbwMIeAP7JLtkUymSTChS5_
8
9 Use the username 'admin' to log in. To log in as 'admin', use the 'plesk login'
  command.
10 You can also log in as 'root' using your 'root' password.
11
12 Use the 'plesk' command to manage the server. Run 'plesk help' for more info.
13
14 Use the following commands to start and stop the Plesk web interface:
15 'systemctl start psa.service' and 'systemctl stop psa.service' respectively.
16
17 If you would like to migrate your subscriptions from other hosting panel
18 or older Plesk version to this server, please check out our assistance
19 options: https://www.plesk.com/professional-services/
```

Para completar el proceso de instalación habrá que acceder a una de las URLs que aparecen en el mensaje anterior. Cuando acceda le aparecerá un formulario donde puede configurar los siguientes parámetros:

- **Correo electrónico** del administrador.
- **Contraseña** del usuario **admin**.
- **Licencia**. En nuestro seleccionaremos la licencia de prueba que cuenta con todas las opciones disponibles durante un período de **15 días**.

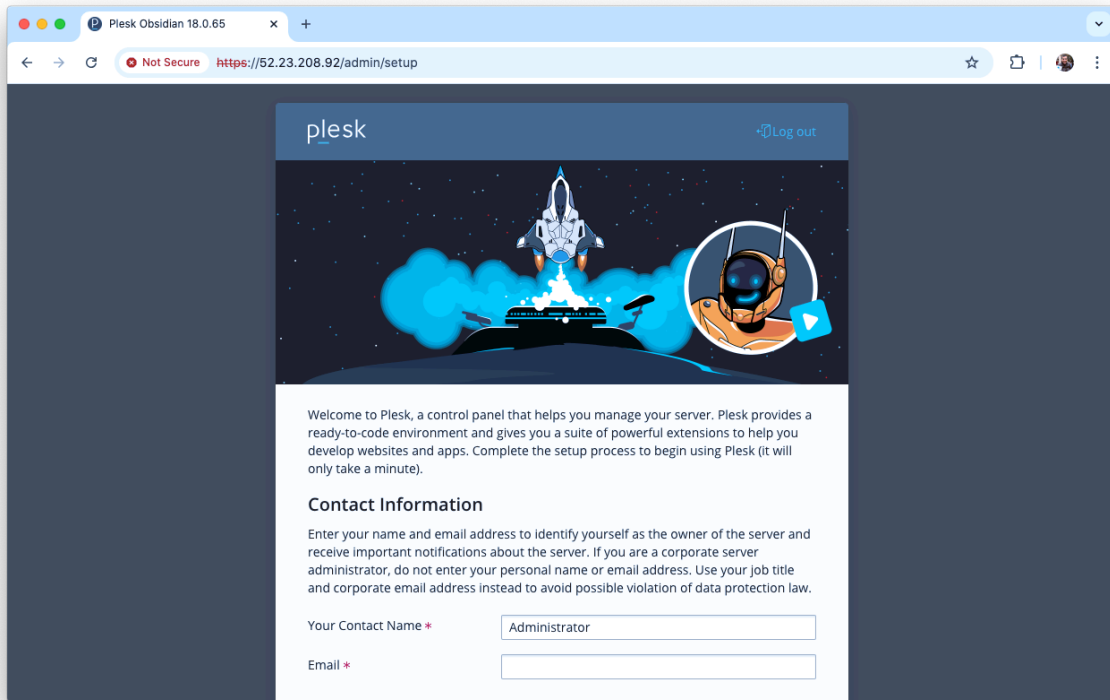


Imagen: Formulario para completar el proceso de instalación de Plesk.

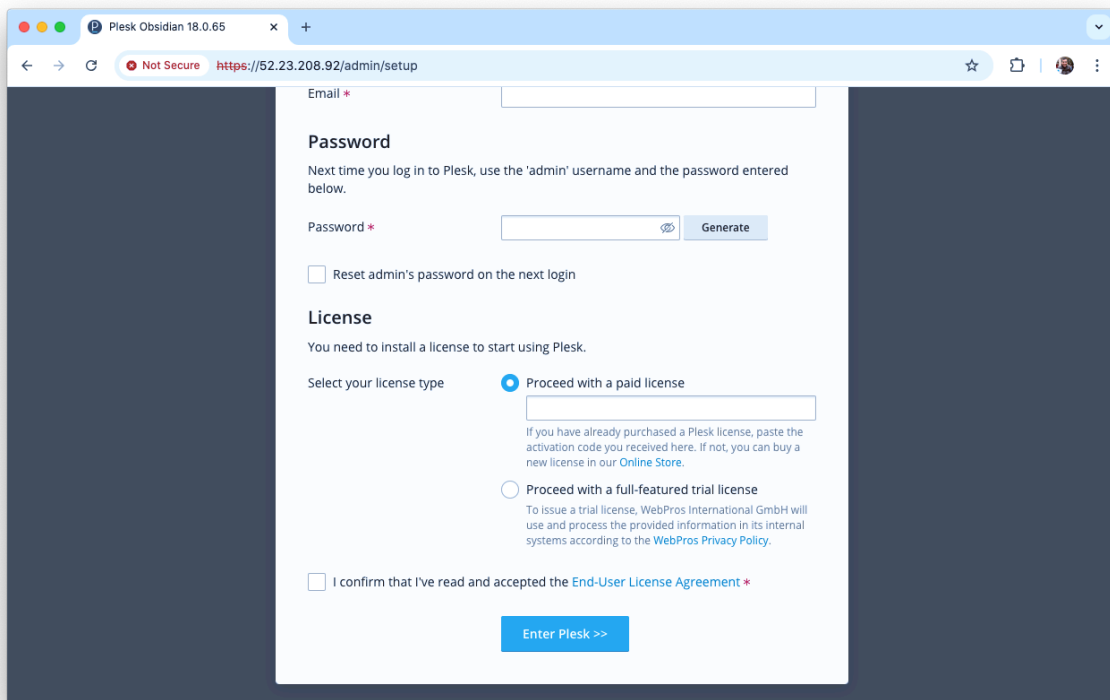


Imagen: Formulario para completar el proceso de instalación de Plesk.

1.2.5 ¿Cómo recuperar el password del usuario admin de Plesk?

Si no recuerda la contraseña del usuario `admin` de Plesk, puede recuperarla accediendo por SSH a la instancia y **ejecutando algunos de los siguientes comandos como `root`**.

Solución 1

Este comando nos devolverá una URL para acceder a Plesk.

```
1 # plesk login
```

Solución 2

Este comando nos permite cambiar la contraseña del usuario `admin`. Tendrá que sustituir la cadena `your_new_password` por la nueva contraseña que quiera configurar.

```
1 # export PSA_PASSWORD=your_new_password && plesk bin admin --set-admin-password  
-passwd "$PSA_PASSWORD"
```

Referencia

- [How to get or reset lost Plesk Administrator password?](#)

1.3 OpenStack

1.3.1 Instalación de Plesk en OpenStack

A continuación se describen **muy brevemente** algunas de las tareas que tendrá que realizar para instalar **Plesk** en el **OpenStack** del instituto.

1. Crea una instancia (máquina virtual) en **OpenStack**.
2. La **imagen** que vamos a utilizar en esta práctica será **`plesk-ubuntu-22-18.0.56`** que está basada Ubuntu Server.
3. Seleccione el *flavor* **`m1.large`** con 4 vCPUs, 4 GB de RAM y 20 GB de disco.
4. Configure el grupo de seguridad para configurar los puertos que estarán abiertos para poder conectarnos por **SSH** y para poder acceder por **HTTP/HTTPS**. También se recomienda crear una regla para permitir el tráfico **ICMP** para poder hacerle **ping** a la instancia.
 - 22: SSH (TCP)
 - 80: HTTP (TCP)
 - 443: HTTPS (TCP)
 - ICMP

5. Si vamos a conectarnos por **FTP** al servidor, es necesario abrir los puertos 20 Y 21. Plesk utiliza por defecto el **modo activo** para las conexiones **FTP** pero si queremos configurar el **modo pasivo**, es necesario abrir un rango de puertos.
 - 21: FTP (TCP) - Puerto de control en los modos activo y pasivo
 - 20: FTP (TCP) - Puerto de datos en el modo activo
 - 49152 – 65535: Rango de puertos dinámicos para el modo pasivo (TCP)
6. Si vamos a configurar el servidor de correo electrónico, es necesario abrir los puertos de los protocolos de correo saliente (SMTP) y correo entrante (IMAP y POP3).
 - 25: SMTP (TCP)
 - 465: SMTPS (TCP)
 - 143: IMAP (TCP)
 - 993: IMAPS (TCP)
 - 110: POP3 (TCP)
 - 995: POP3S (TCP)
7. Si va a utilizar **Webhooks** de GitHub para hacer **pulls** y despliegues automáticos en Plesk, es necesario abrir el puerto:
 - 8443: (TCP)
8. Crea un par de claves (pública y privada) para conectar por SSH con su instancia.
9. Crea una dirección **IP flotante** y asígnala a la instancia.
10. Busque cuál es la dirección IP flotante de su instancia y compruebe que puede acceder a ella desde una navegador web.

1.3.2 Acceso al panel de administración de Plesk

Las máquinas virtuales creadas a partir de una imagen de tipo **QCOW2** no tienen preconfigurada una contraseña para el usuario **root**. Por lo tanto, para acceder al panel de administración web de Plesk es necesario **iniciar la máquina virtual en modo de recuperación** y modificar la contraseña del usuario **root**.

1. Acceda al panel de control web de OpenStack.
2. Seleccione la instancia y acceda a la pestaña **Console**.
3. Reinicie la instancia pulsando sobre el texto **Send CtrlAltDel**.
4. Pulse la tecla **Shift** para que aparezca el menú de arranque.
5. Seleccione **Advanced options for Ubuntu**.
6. Seleccione **Ubuntu, with Linux 5.15.0-88-generic (recovery mode)**.
7. Seleccione **root - Drop to root shell prompt**.
8. La partición raíz se monta en modo de solo lectura. Para montarla en modo de lectura y escritura, introduzca el comando:

```
1 mount -o remount,rw /
```

9. Modifique la contraseña del usuario `root`:

```
1 passwd root
```

10. Reinicie la máquina virtual:

```
1 reboot
```

Una vez realizados estos pasos podrá acceder al panel de administración web de Plesk accediendo por [HTTPS](#), a través de un nombre de dominio o de la dirección IP flotante de la máquina.

```
1 https://IP_FLOTANTE
```

Para acceder al panel de administración web puede utilizar el usuario `root` y la contraseña que ha configurado en el paso anterior.

Referencia:

- [Logging in to the Created Virtual Machine](#). Documentación oficial de Plesk.

1.4 Gestión de dominios y subdominios en Plesk

En esta práctica tendrá que crear dominios y subdominios en el panel de [Plesk](#) para desplegar varias aplicaciones web.

Se proponen dos soluciones para la creación de los dominios:

- Gestionar las entradas DNS locales en su sistema operativo en el archivo `/etc/hosts`.
- Gestionar los dominios con algún proveedor de dominios gratuitos, como [No-IP](#).
- Gestionar los dominios con el DNS wildcard que nos proporciona el servicio [nip.io](#).

Se recomienda la lectura de la [documentación oficial de Plesk](#) para aprender a realizar estas tareas.

1.4.1 Configuración de entradas DNS locales en su sistema operativo

En este caso, los dominios y subdominios propuestos son los siguientes:

Dominio	Descripción
<code>web.celia</code>	En este dominio realizará el despliegue utilizando la subida de archivos desde el panel de control de Plesk.

Dominio	Descripción
<code>wordpress.web.celia</code>	En este subdominio realizará el despliegue de un sitio web WordPress utilizando el instalador de aplicaciones web de Plesk.
<code>git.web.celia</code>	En este subdominio realizará el despliegue utilizando un repositorio privado de Git que esté alojado en GitHub.
<code>ftp.web.celia</code>	En este subdominio realizará el despliegue de una aplicación web subiendo los archivos a Plesk por FTP.
<code>laravel.web.celia</code>	En este subdominio realizará el despliegue de una aplicación web Laravel.

Para poder acceder a los dominios y subdominios que ha creado en el panel de Plesk, vamos a configurar las entradas DNS locales **en su sistema operativo** para que pueda acceder a ellos desde su navegador web.

Linux

Si su sistema operativo está basado en Linux, tendrá que editar el archivo:

```
1 /etc/hosts
```

Tendrá que editar el archivo como `root` y añadir los dominios que ha dado de alta en Plesk para que apunten a la dirección IP flotante de la instancia de OpenStack.

Ejemplo:

```
1 IP_FLOTANTE_PLESK web.celia
2 IP_FLOTANTE_PLESK git.web.celia
3 IP_FLOTANTE_PLESK ftp.web.celia
4 IP_FLOTANTE_PLESK wordpress.web.celia
```

Tenga en cuenta que tendrá que sustituir `IP_FLOTANTE_PLESK` por la dirección IP flotante de su instancia. Por ejemplo, si la IP flotante fuese `172.16.14.199` el archivo quedaría así.

```
1 172.16.14.199 web.celia
2 172.16.14.199 git.web.celia
3 172.16.14.199 ftp.web.celia
4 172.16.14.199 wordpress.web.celia
```

Windows

En el sistema operativo Windows tendrá que realizar los mismos pasos pero el archivo de configuración se encuentra en la ruta:

```
1 C:\Windows\System32\drivers\etc\hosts
```

1.4.2 Gestionar los dominios con algún proveedor de dominios gratuitos

En este caso puede utilizar el proveedor de dominios gratuitos [No-IP](#). Tenga en cuenta que con la versión gratuita sólo podrá crear un dominio asociado a una dirección de correo electrónico, por lo tanto, tendrá que utilizar varias cuentas de correo.

Para evitar tener que gestionar varias cuentas de correo, se propone utilizar la siguiente solución que aparece en esta guía.

1.4.3 Gestionar los dominios con el DNS wildcard del servicio nip.io

El servicio [nip.io](#) permite gestionar dominios DNS wildcard de una forma muy sencilla, sin tener que registrarnos en ningún sitio. Este servicio mapea una dirección IP a un nombre de dominio utilizando dos formatos:

Formato sin nombre

Dominio	Descripción
<code>10.0.0.1.nip.io</code>	En este dominio resuelve la dirección IP <code>10.0.0.1</code> .
<code>10-0-0-1.nip.io</code>	En este dominio resuelve la dirección IP <code>10.0.0.1</code> .

Formato con nombre

Dominio	Descripción
<code>wordpress.10.0.0.1.nip.io</code>	En este dominio resuelve la dirección IP <code>10.0.0.1</code> .
<code>wordpress.10-0-0-1.nip.io</code>	En este dominio resuelve la dirección IP <code>10.0.0.1</code> .

En este caso, los dominios y subdominios propuestos son los siguientes:

Dominio	Descripción
<code>IP_ELÁSTICA.nip.io</code>	En este dominio realizará el despliegue utilizando la subida de archivos desde el panel de control de Plesk.
<code>wordpress.IP_ELÁSTICA.nip.io</code>	En este subdominio realizará el despliegue de un sitio web WordPress utilizando el instalador de aplicaciones web de Plesk.
<code>git.IP_ELÁSTICA.nip.io</code>	En este subdominio realizará el despliegue utilizando un repositorio privado de Git que esté alojado en GitHub.

Dominio	Descripción
<code>ftp.IP_ELÁSTICA.nip.io</code>	En este subdominio realizará el despliegue de una aplicación web subiendo los archivos a Plesk por FTP.

1.5 Despliegue de una aplicación subiendo los archivos desde el panel de control

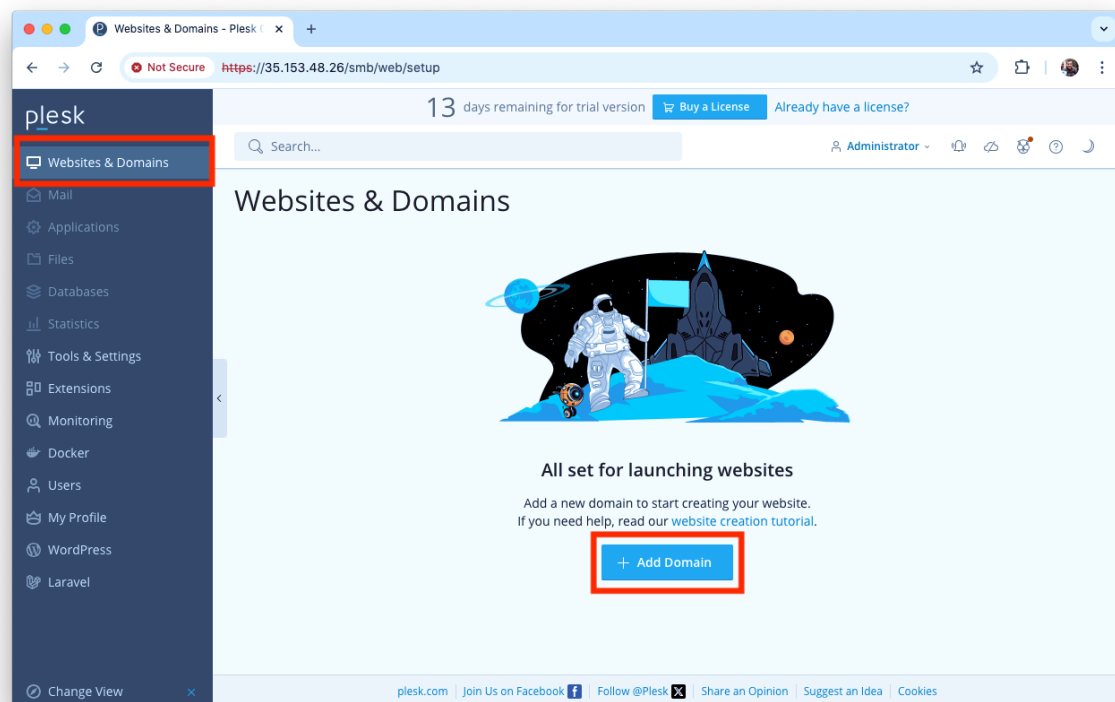
En este ejemplo vamos a dar de alta un dominio con el siguiente formato:

```
1 IP_ESLÁSTICA.nip.io
```

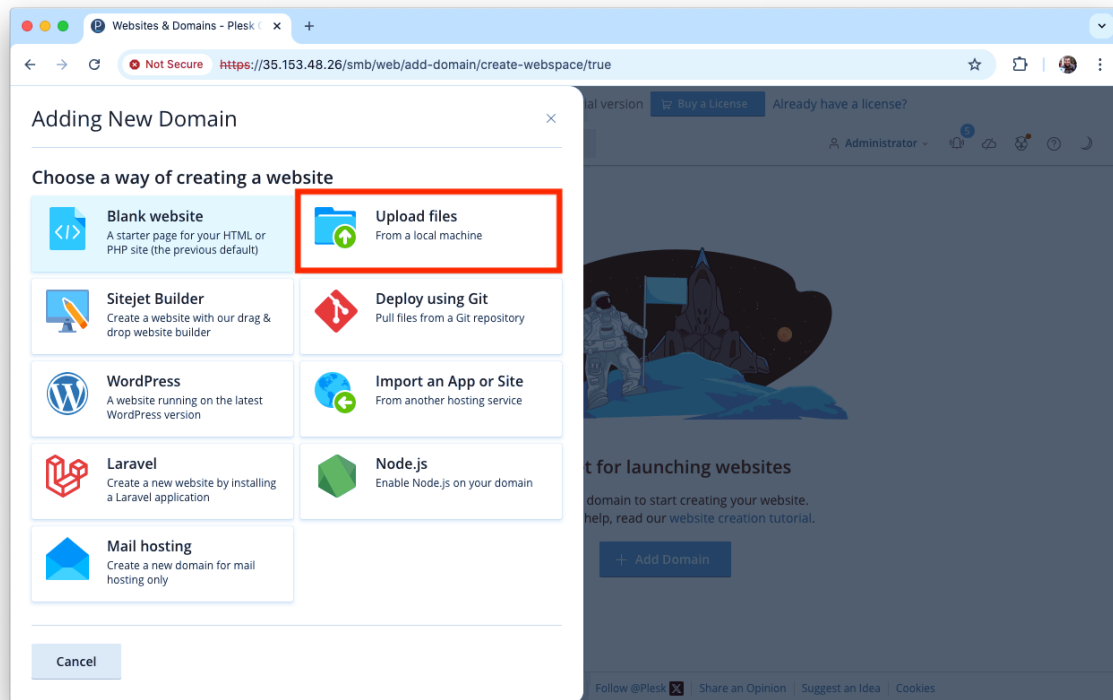
Por ejemplo, si la IP elástica fuera 35 . 153 . 48 . 26 el dominio sería:

```
1 35.153.48.26.nip.io
```

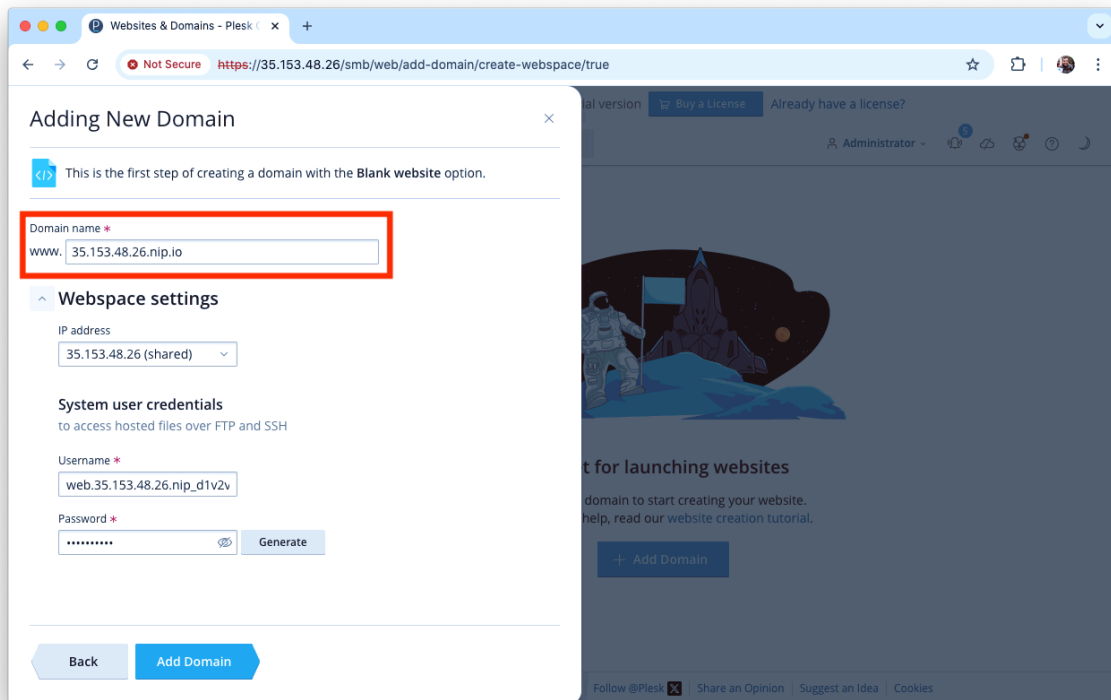
Paso 1. Seleccionamos la opción **Websites & Domains** y hacemos clic en el botón **Add Domain**.



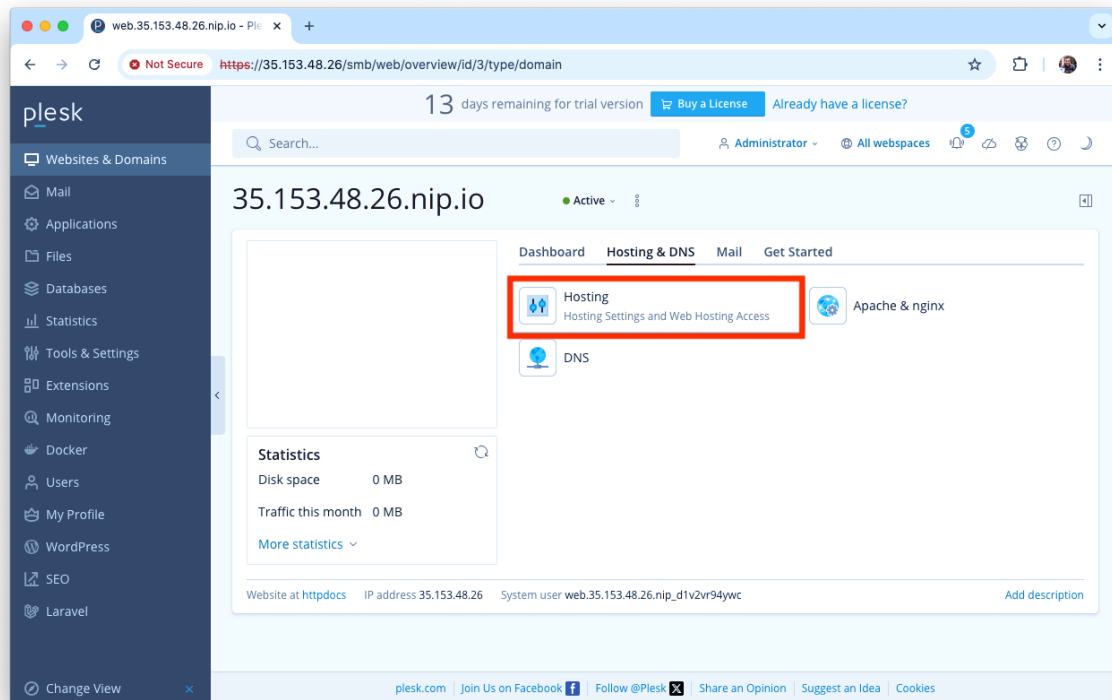
Paso 2. Para indicar que vamos a desplegar un sitio web subiendo los archivos desde nuestra máquina local, seleccionamos la opción **Upload files**.



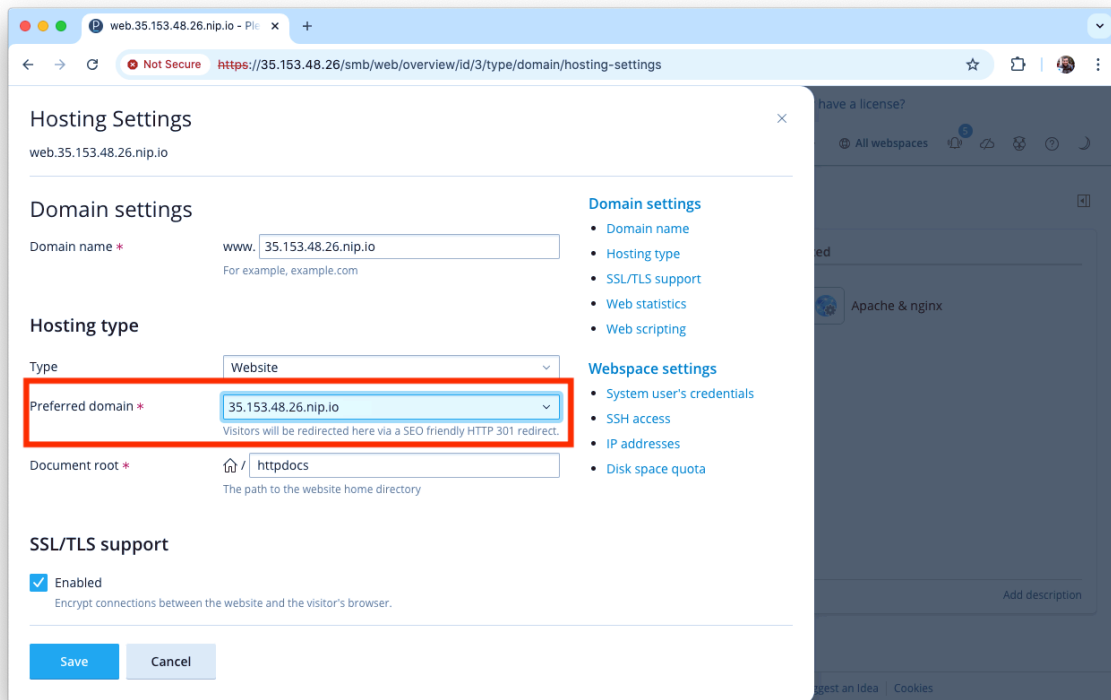
Paso 3. Indicamos el nombre del dominio que queremos añadir. Tenga en cuenta que Plesk añadirá automáticamente el prefijo `www`.



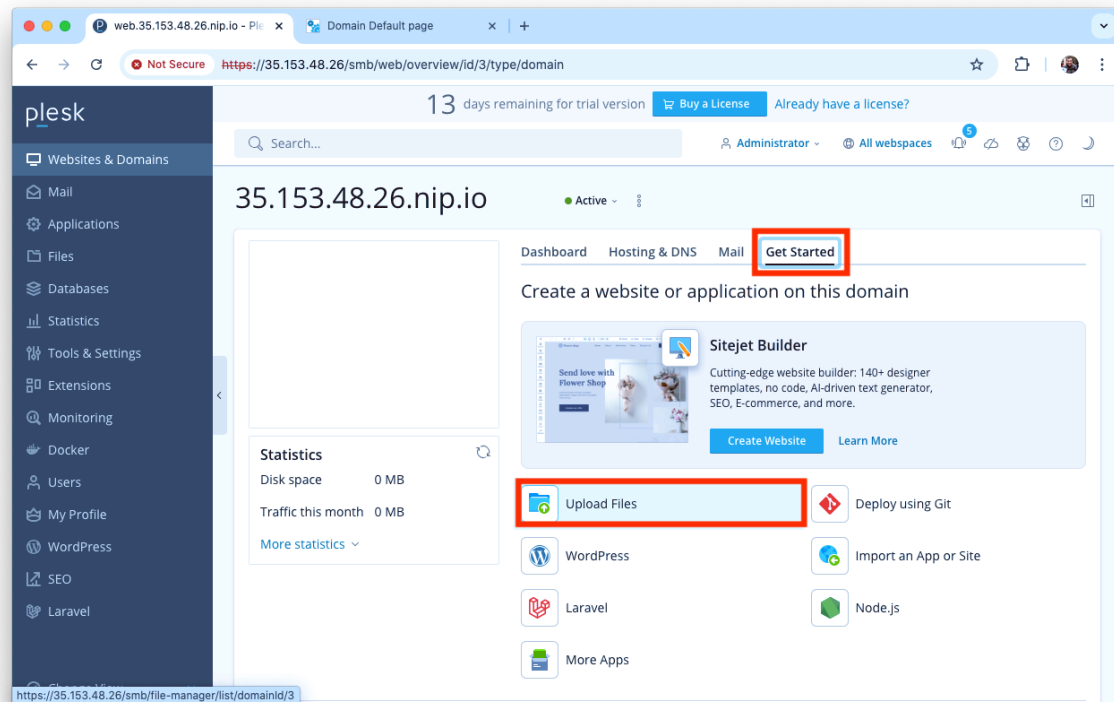
Paso 4. Si queremos que los usuarios sean redirigidos automáticamente al dominio sin `www`, accedemos a la sección **Hosting & DNS** y seleccionamos la opción **Hosting**.



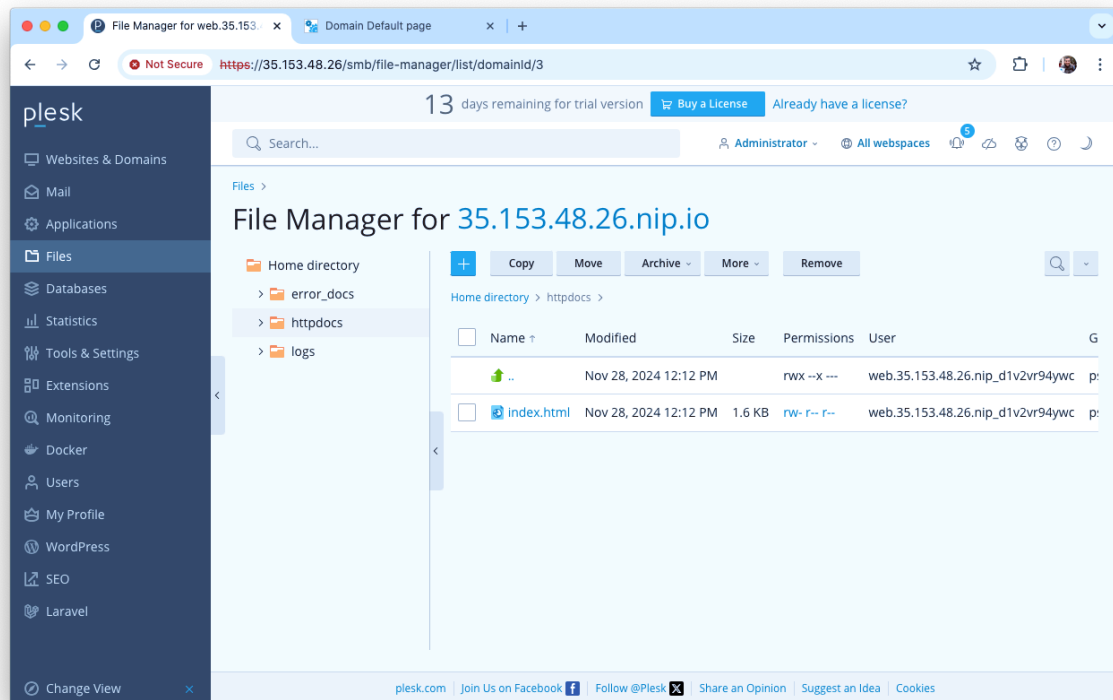
Paso 5. En el desplegable que aparece en la opción Preferred domain seleccionamos cuál será nuestro dominio preferido.



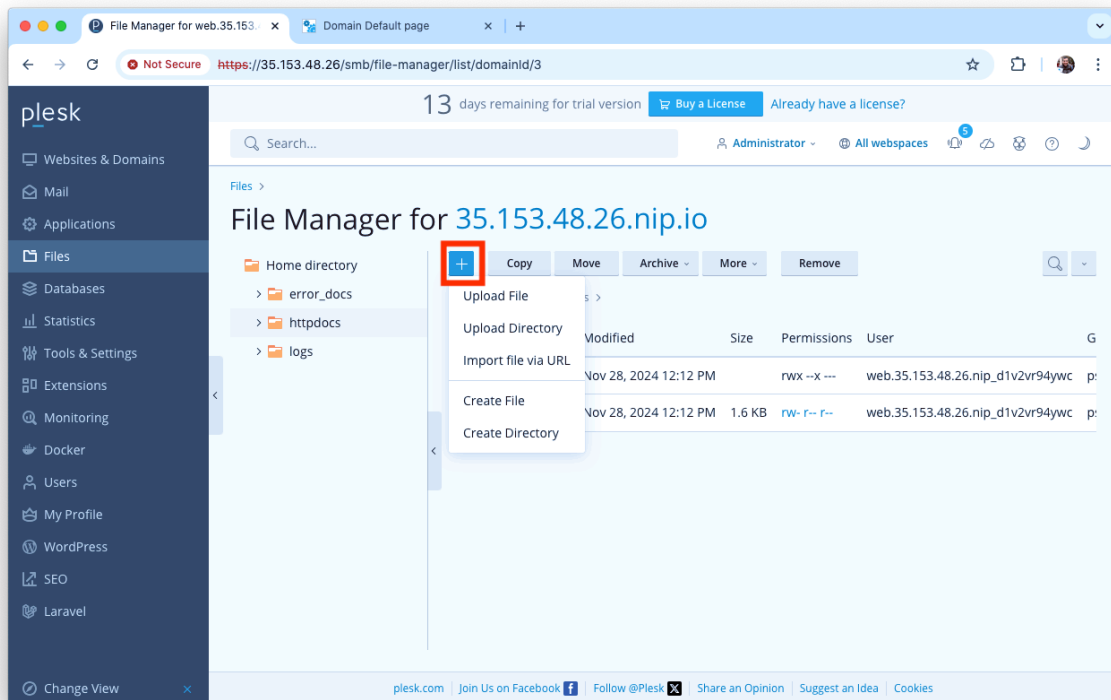
Paso 6. Para subir los archivos de la aplicación web seleccionamos la opción **Get Started** y dentro de esta sección seleccionamos la opción **Upload Files**.



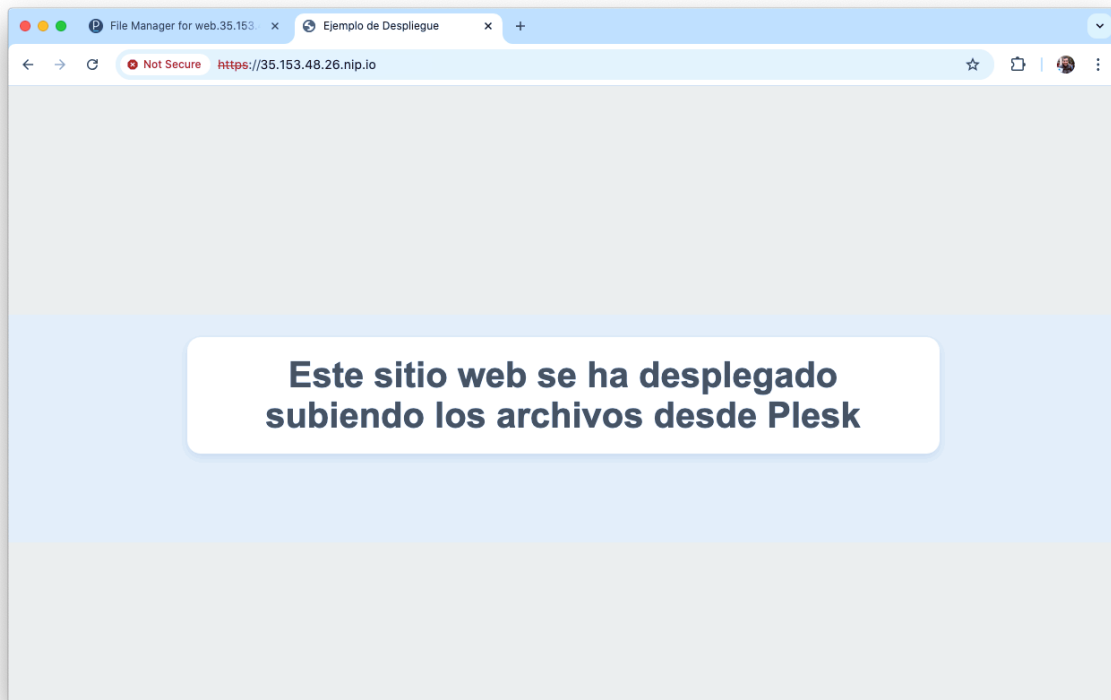
Paso 7. Nos aparecerá el árbol de directorios del servidor y dentro del directorio `httpdocs` subiremos los archivos de la aplicación web.



Paso 8. Para subir nuevos archivos pinchamos sobre el botón con el icono del + y nos aparecerá un menú con varias opciones.

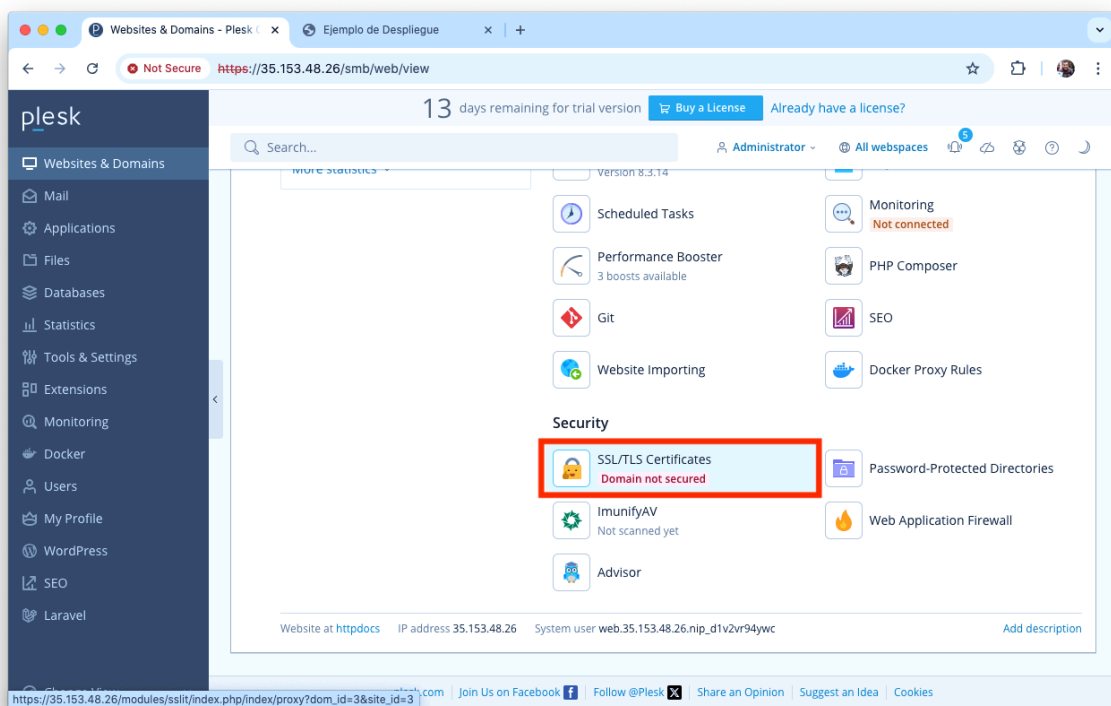
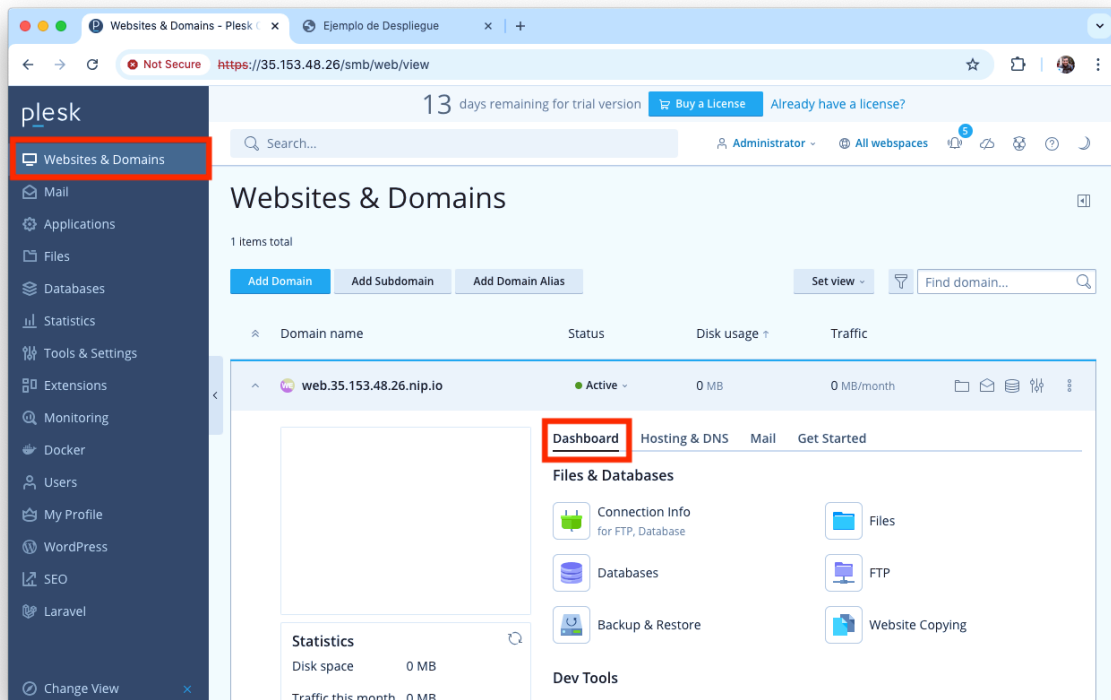


Paso 9. Una vez que hayamos subido nuestros archivos, accederemos desde otra pestaña del navegador al dominio que hemos dado de alta en Plesk.



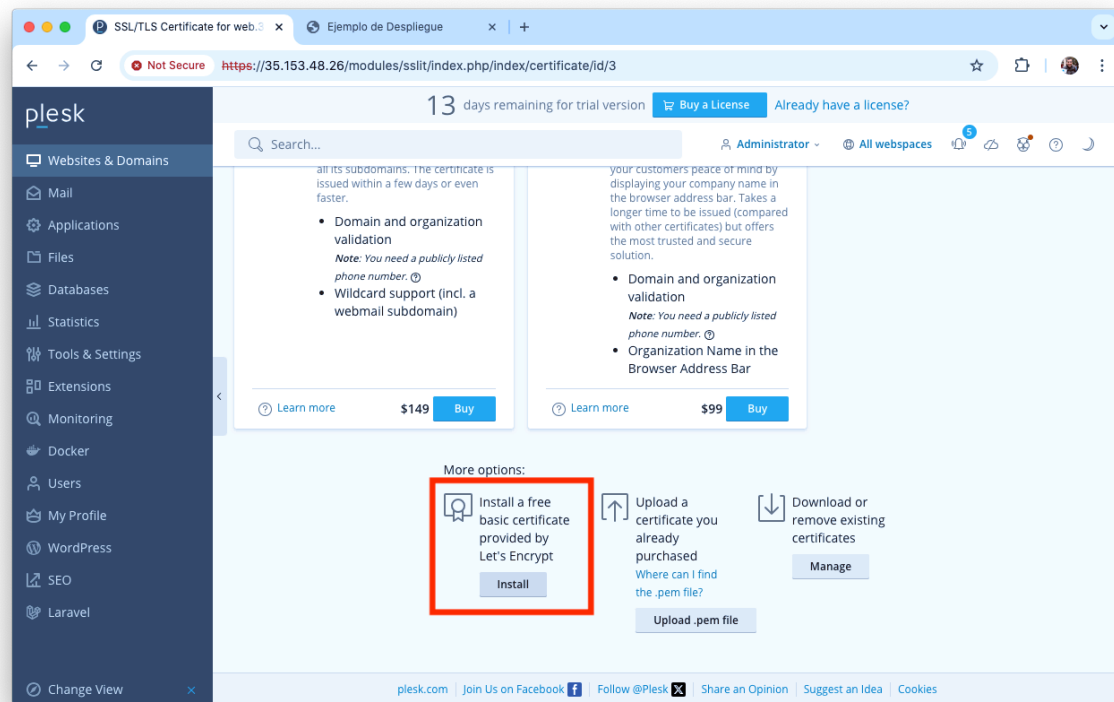
Paso 10. Podemos observar que nuestro sitio sólo está disponible por [HTTP](#). Vamos a solucionarlo solicitando un certificado SSL gratuito a Let's Encrypt.

Seleccionamos en el menú de la izquierda la opción [Websites & Domains](#) y en la sección [Dashboard](#) seleccionamos la opción [SSL/TLS Certificates](#).

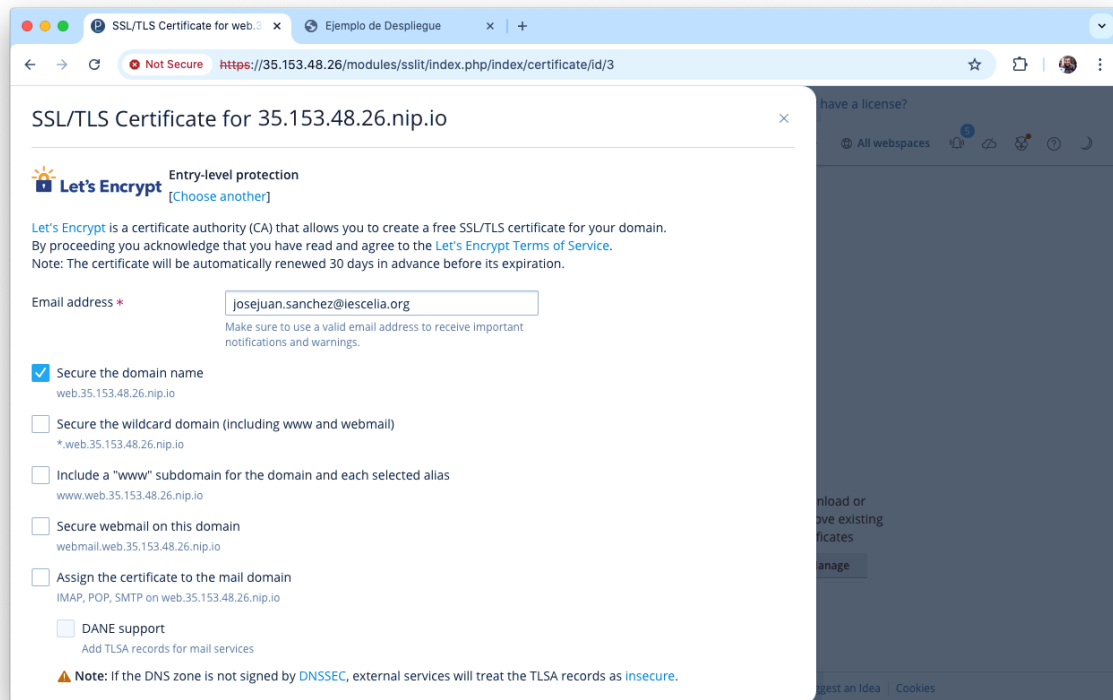


Paso 11. Seleccionamos la opción Install a free basic certificate provided by Let's

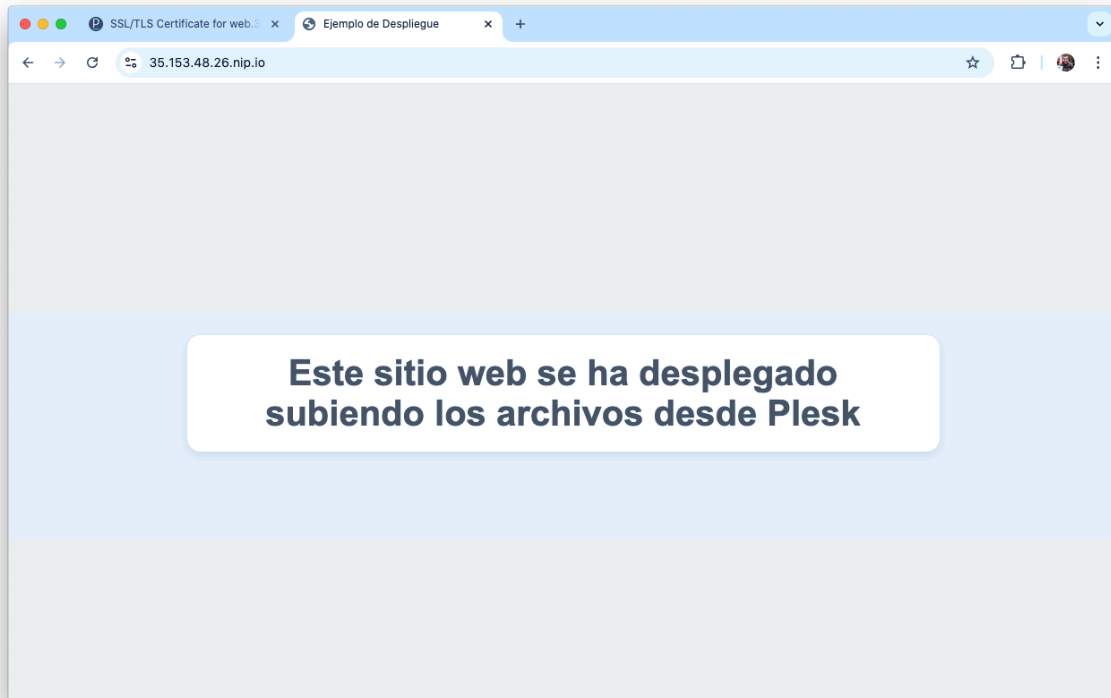
Encrypt.



Paso 12. Indicamos que vamos a solicitar un certificado para nuestro dominio.



Paso 13. Una vez que haya finalizado el proceso de instalación del certificado, podemos acceder a nuestro sitio web por **HTTPS**.



1.6 Despliegue de una aplicación WordPress

En este ejemplo vamos a dar de alta un **subdominio** con el siguiente formato:

```
1 wordpress.IP_ESLÁSTICA.nip.io
```

Por ejemplo, si la IP elástica fuera 35 . 153 . 48 . 26 el subdominio sería:

```
1 wordpress.35.153.48.26.nip.io
```

Plesk dispone de la utilidad [WP Toolkit](#) que permite instalar, configurar y administrar WordPress de una forma sencilla.

En esta práctica tendrá que crear el subdominio y desplegar una aplicación WordPress.

Una vez que haya realizado la instalación de WordPress, tendrá que aplicar las medidas de seguridad recomendadas por Plesk para proteger la aplicación.

Referencias:

- [WP Toolkit](#). Documentación oficial de Plesk.

1.7 Despliegue de una aplicación web con el asistente de Plesk

Plesk nos permite realizar la instalación de numerosas aplicaciones web de una forma muy sencilla. Entre las aplicaciones que podemos instalar se encuentran:

- [WordPress](#)
- [Joomla](#)
- [Drupal](#)
- [PrestaShop](#)
- [Moodle](#)
- [MediaWiki](#)

Realice el despliegue de una de estas aplicaciones web utilizando el asistente de Plesk.

1.8 Despliegue de una aplicación web con Git

En este ejemplo vamos a dar de alta un **subdominio** con el siguiente formato:

```
1 git.IP_ESLÁSTICA.nip.io
```

Por ejemplo, si la IP elástica fuera 35.153.48.26 el dominio sería:

```
1 git.35.153.48.26.nip.io
```

En este apartado tendrá que realizar el despliegue de una aplicación web LAMP que está alojada en un repositorio de GitHub. La aplicación web está desarrollada en PHP y utiliza una base de datos MySQL. La URL de la aplicación es la siguiente:

- <https://github.com/josejuansanchez/iaw-practica-lamp>

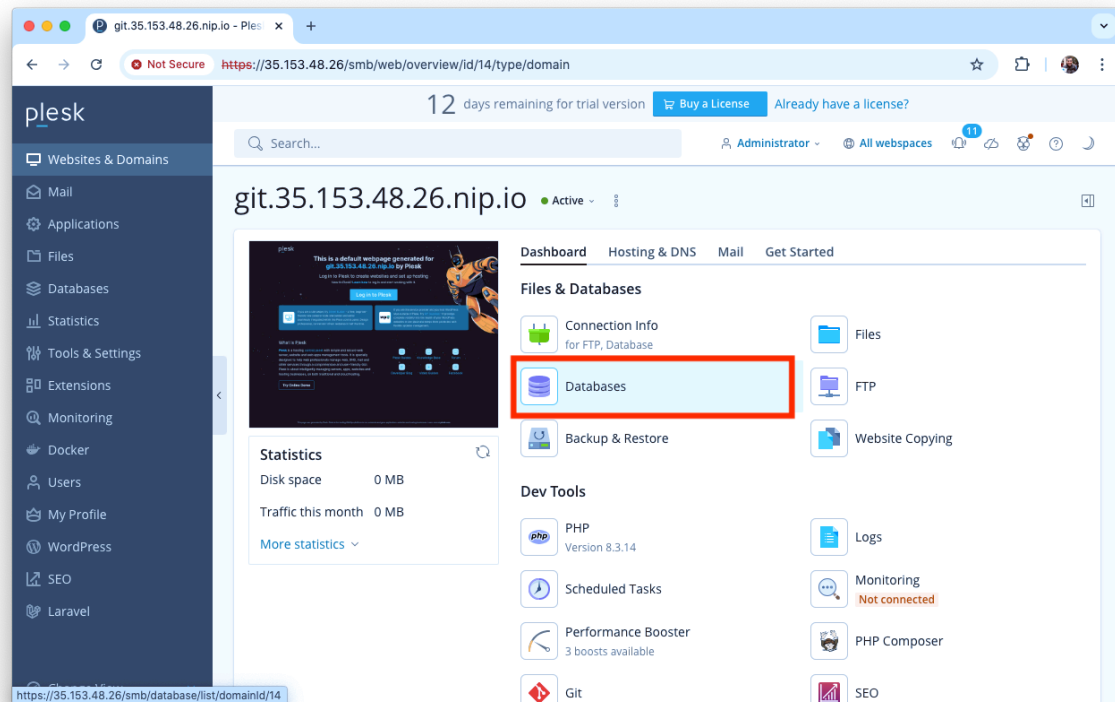
Nota: Se recomienda que haga un **fork** del repositorio en su cuenta de GitHub para que más adelante pueda configurar el clonado mediante una **clave SSH** y pueda configurar un **Webhook**.

Para realizar el despliegue tendrá que realizar los siguientes pasos:

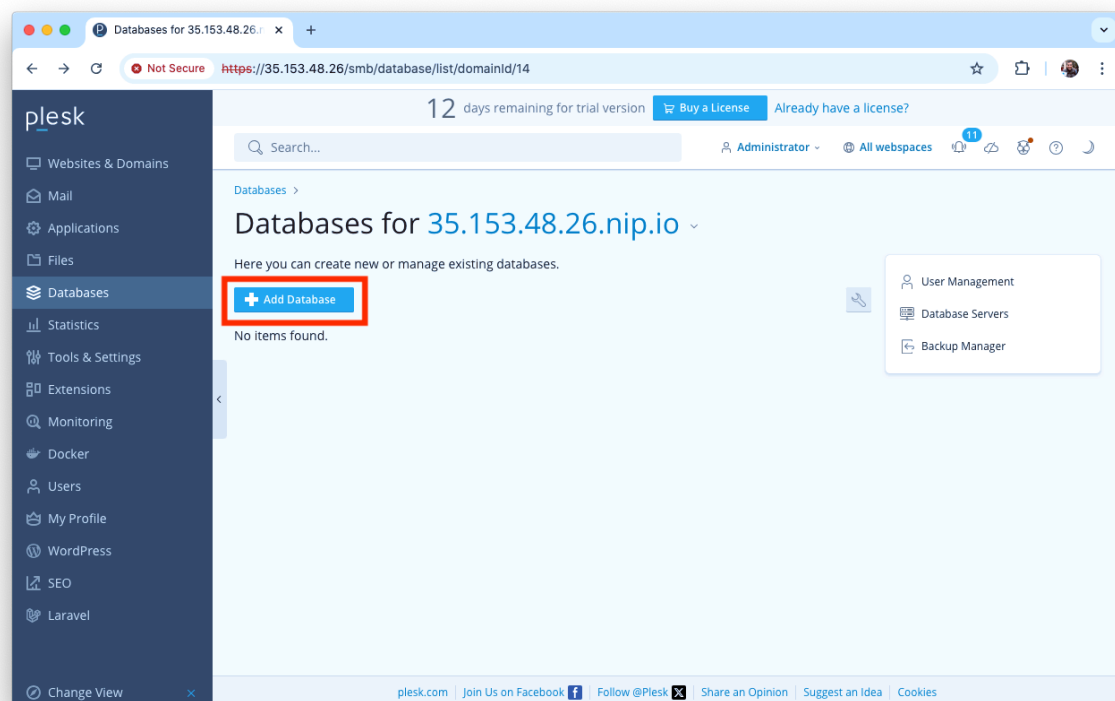
Paso 1. En primer lugar, tendrá que añadir el nuevo dominio y configurar un certificado SSL/TLS de Let's Encrypt.

Paso 2. Crea una base de datos MySQL para la aplicación web y un usuario con permisos sobre la base de datos.

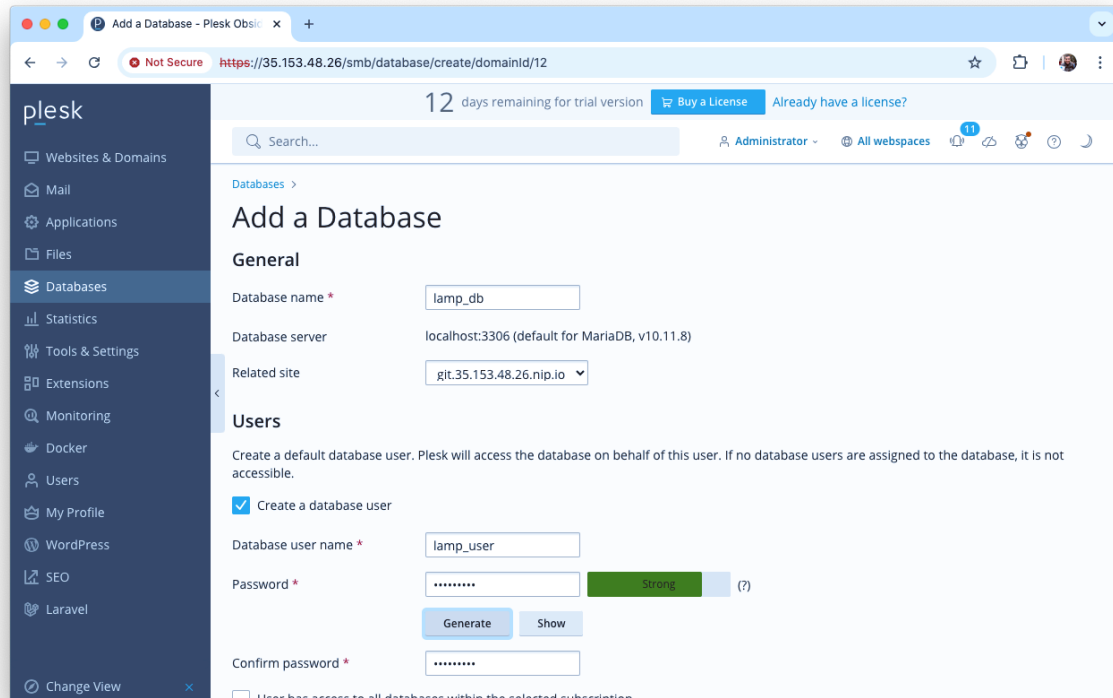
Para crear la base de datos seleccionamos la sección [Dashboard](#) y dentro de esta sección seleccionamos la opción [Databases](#).



Pulsamos sobre el botón Add Database.



En este paso tenemos que indicar el nombre de la base de datos, el nombre del usuario, la contraseña y desde que hosts se puede conectar el usuario.



The screenshot shows the Plesk control panel interface for adding a new database. The page is titled "Add a Database" and is divided into two main sections: "General" and "Users".

General Section:

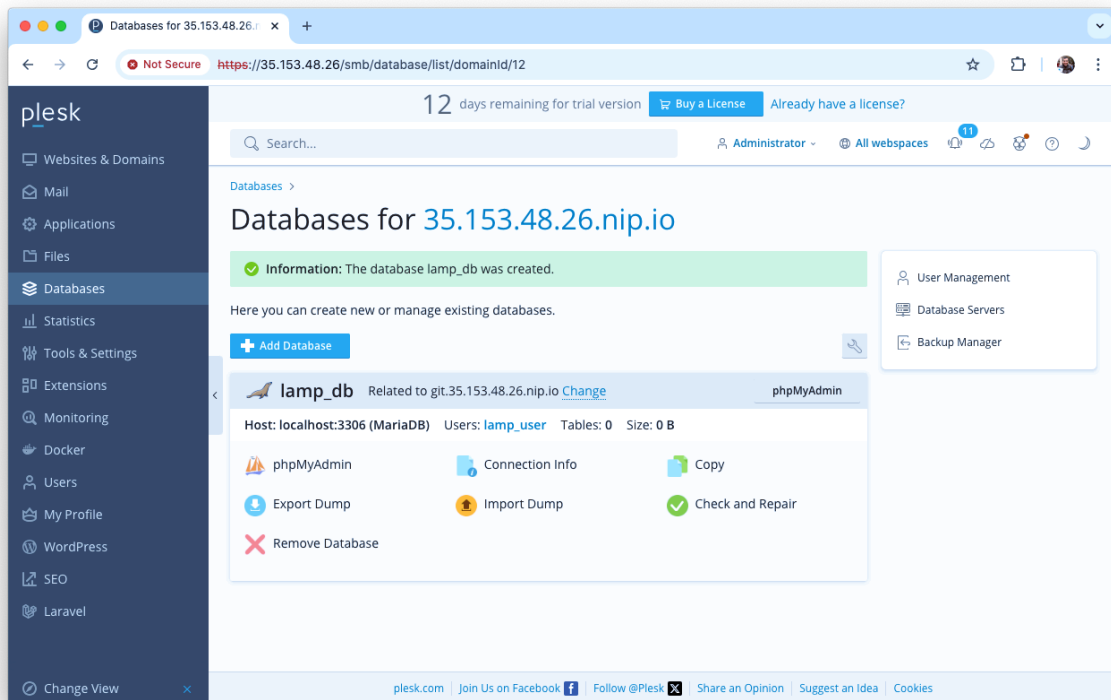
- Database name ***: Input field containing "lamp_db".
- Database server**: Text "localhost:3306 (default for MariaDB, v10.11.8)".
- Related site**: Dropdown menu showing "git.35.153.48.26.nip.io".

Users Section:

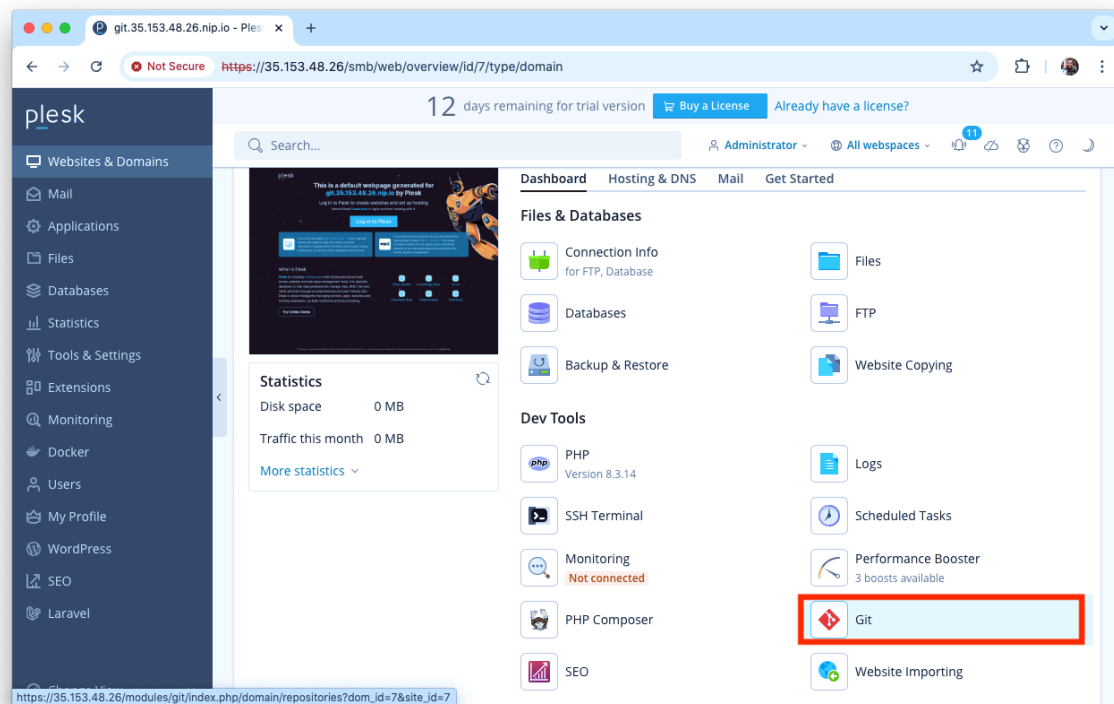
- Text: "Create a default database user. Plesk will access the database on behalf of this user. If no database users are assigned to the database, it is not accessible."
- Create a database user**
- Database user name ***: Input field containing "lamp_user".
- Password ***: Input field with masked characters "*****". A strength indicator shows "Strong" in green. There are "Generate" and "Show" buttons next to it.
- Confirm password ***: Input field with masked characters "*****".
- User has access to all databases within the selected subscription**

The left sidebar contains navigation options: Websites & Domains, Mail, Applications, Files, Databases (selected), Statistics, Tools & Settings, Extensions, Monitoring, Docker, Users, My Profile, WordPress, SEO, and Laravel. The top navigation bar shows "12 days remaining for trial version", "Buy a License", and "Already have a license?".

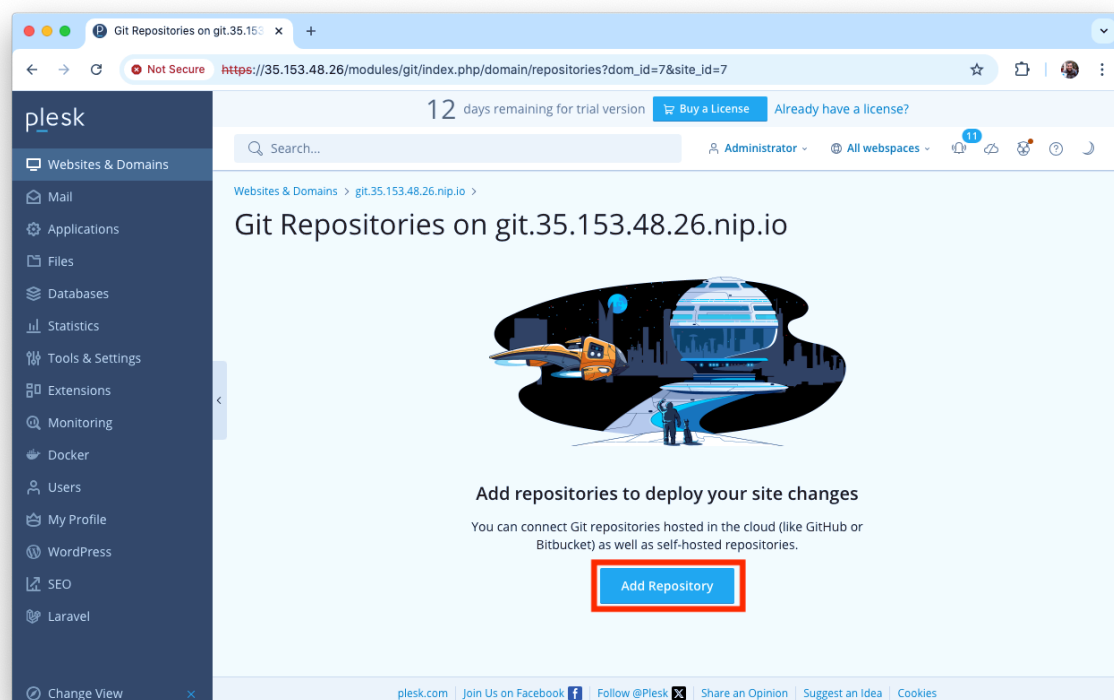
Una vez que hemos creado la base de datos y el usuario, nos aparecerán una serie de opciones para gestionarla. Entre estas opciones tenemos un enlace a phpMyAdmin.



Paso 3. El siguiente paso consiste en clonar el repositorio de GitHub de la aplicación propuesta. Para hacerlo tenemos que acceder a la sección **Dashboard** y en ella seleccionamos la opción **Git**.



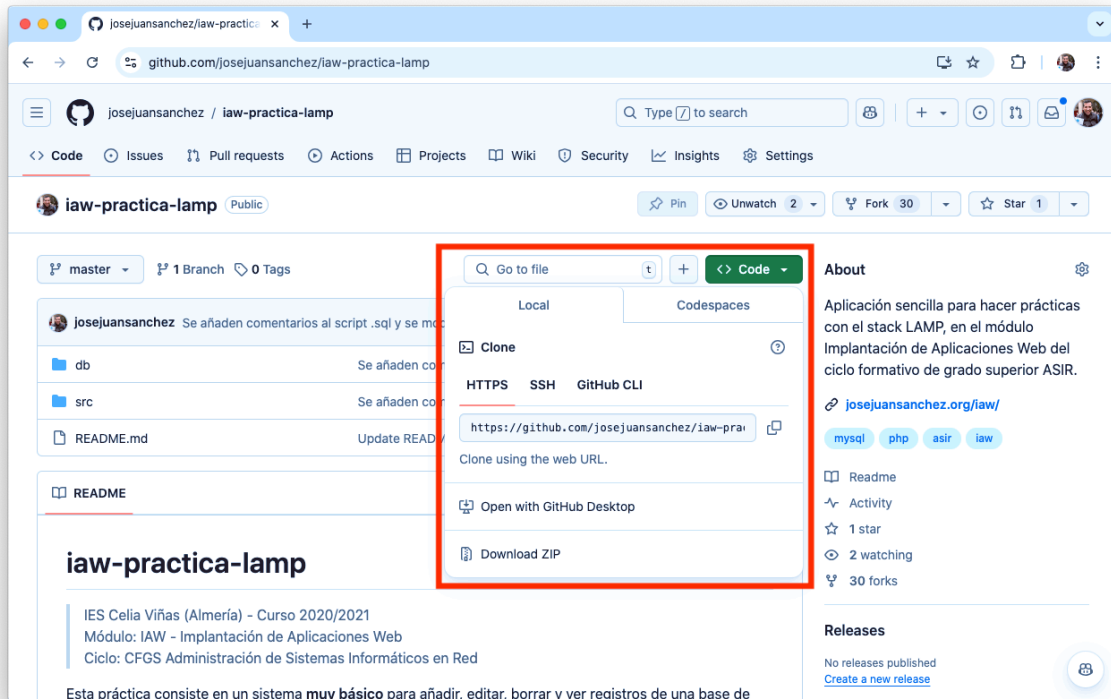
Pinchamos sobre el botón `Add Repository`.



Plesk nos permite clonar repositorios remotos y repositorios locales, y el clonado se puede realizar para cada repositorio será por **HTTPS** o por **SSH**.

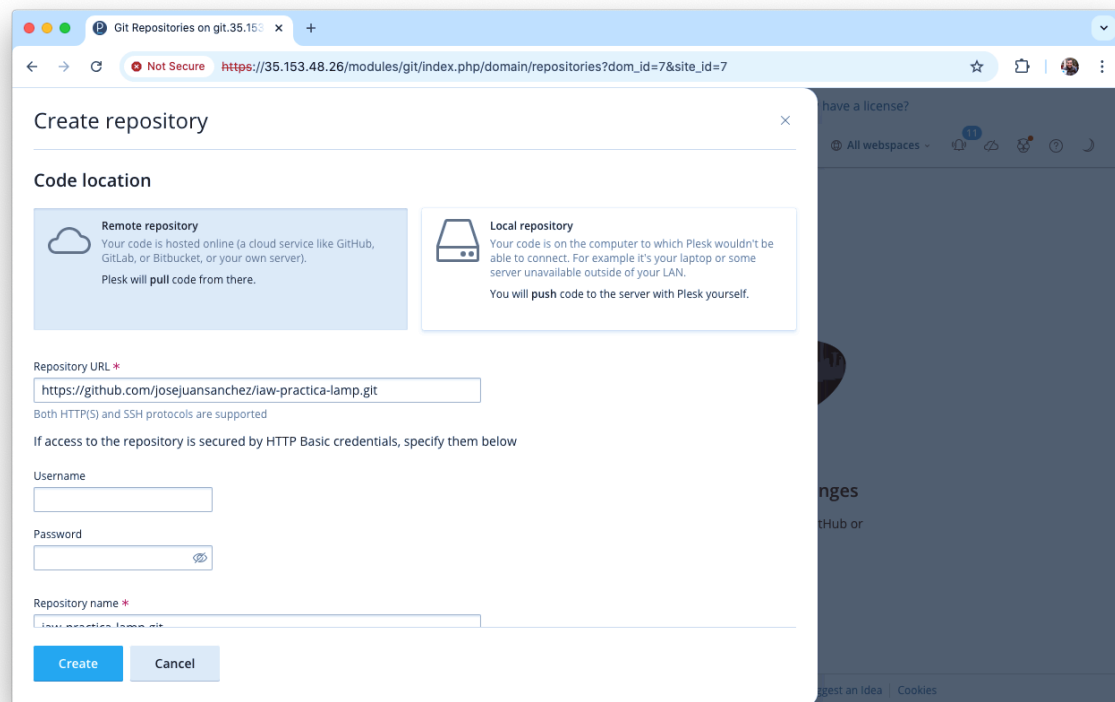
Las URLs para cada caso las puede obtener en la web del repositorio de GitHub. Recuerde, que en este caso la URL del repositorio es la siguiente:

- <https://github.com/josejuansanchez/iaw-practica-lamp>

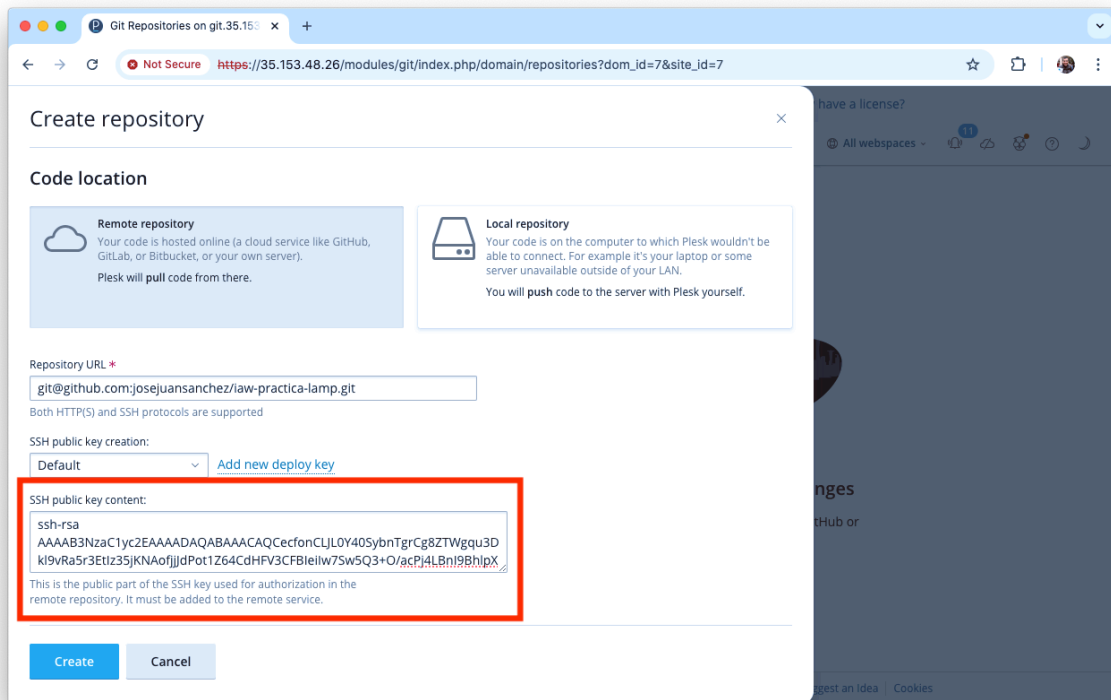


Veamos cómo sería el proceso de clonado del repositorio en Plesk para cada una de las opciones.

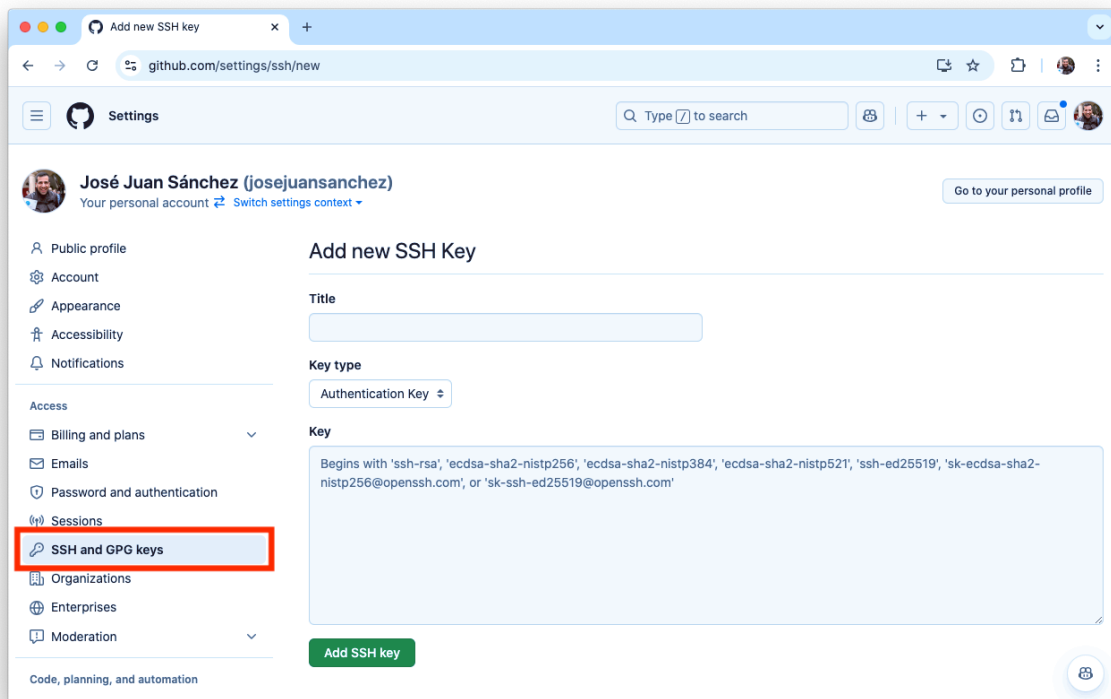
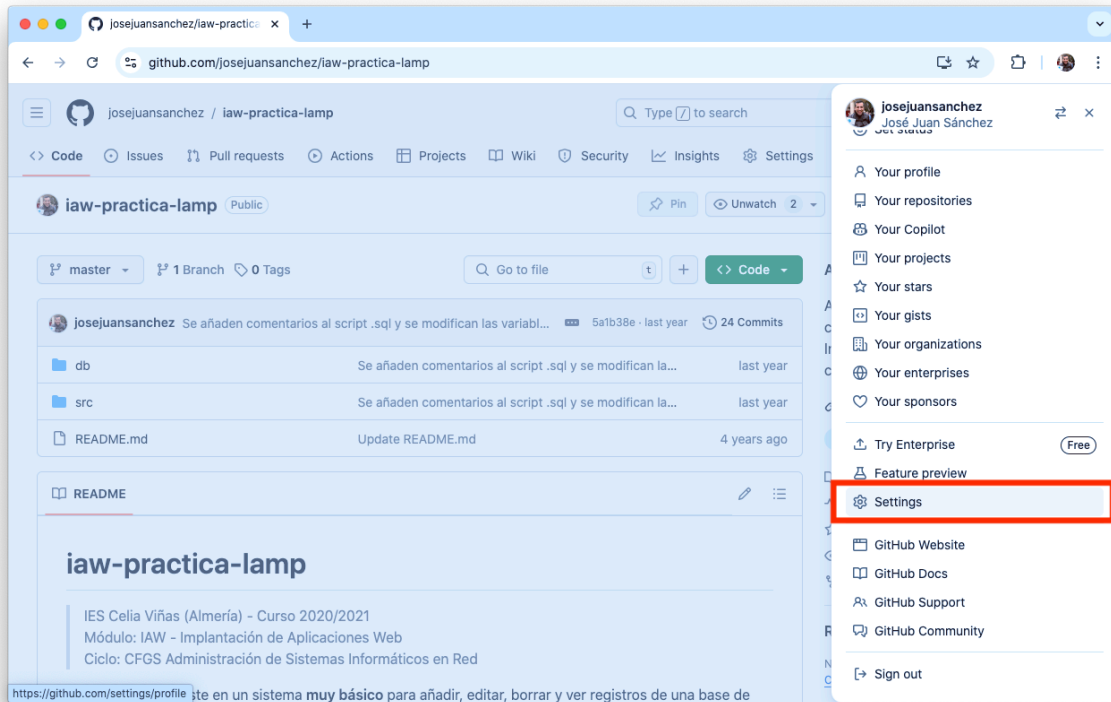
- (a) Si se trata de un repositorio público, se recomienda utilizar clonado por **HTTPS** ya que no será necesario configurar las credenciales para clonarlo.



- (b) Cuando quiera clonar un repositorio privado se recomienda utilizar clonado por **SSH**. En este caso, al clonar la URL en Plesk, nos devolverá una **clave pública SSH** que tendremos que dar alta en la configuración de nuestra cuenta de usuario de GitHub para permitir el acceso desde Plesk.



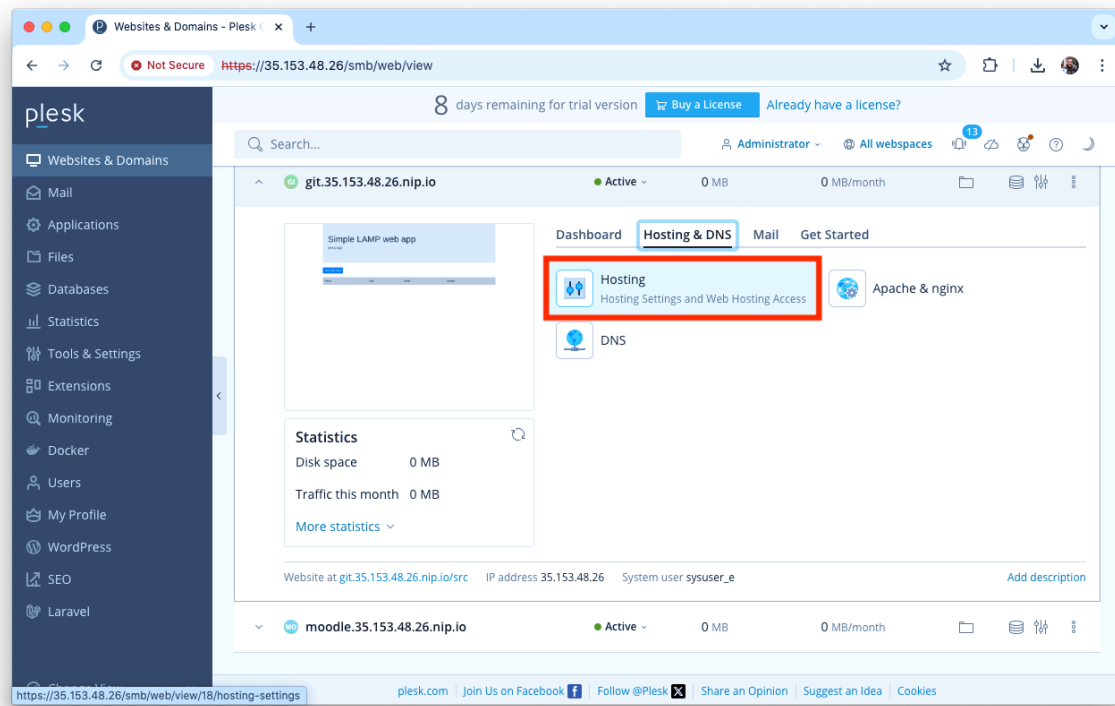
Para dar alta en GitHub la clave pública SSH que nos devuelve Plesk tenemos que acceder a la sección [Settings](#) de nuestra cuenta de GitHub y dentro de esta sección seleccionar la opción [SSH and GPG keys](#)



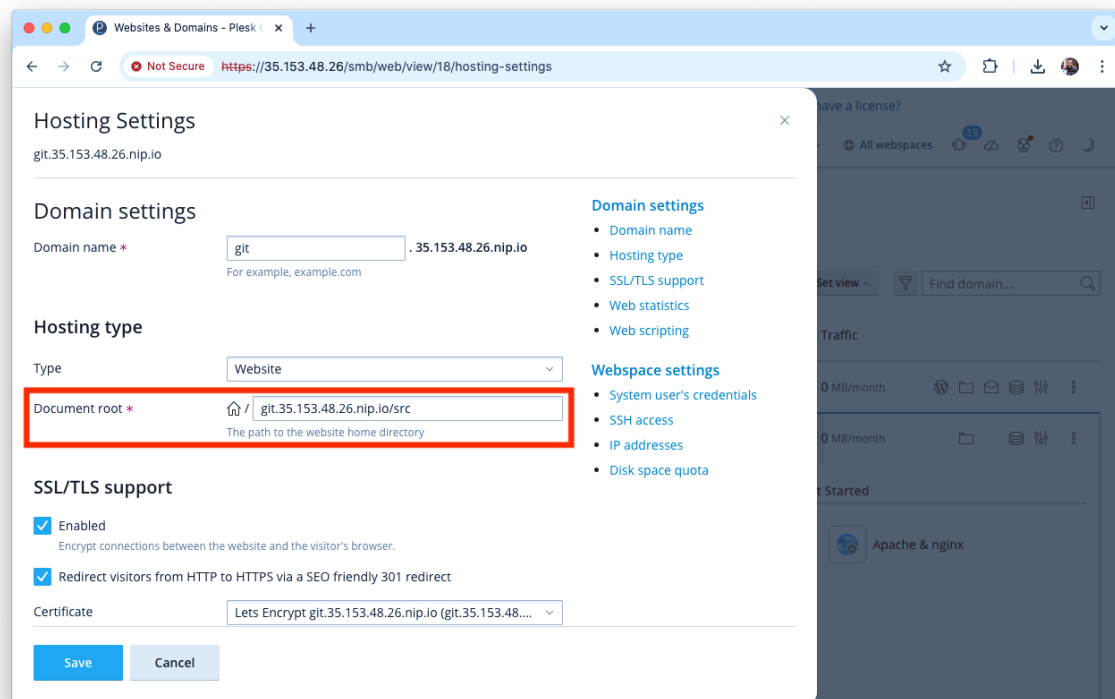
Paso 4. Configuramos el valor de DocumentRoot, que será el directorio que utilizará el servidor web para

servir los archivos de la aplicación web. Accedemos a la ruta:

- [Websites & Domains](#) > [Hosting & DNS](#) > [Hosting](#)



En el apartado `DocumentRoot` hay que indicar la ruta que queremos configurar, que en este caso será la carpeta `src` del repositorio que hemos clonado.

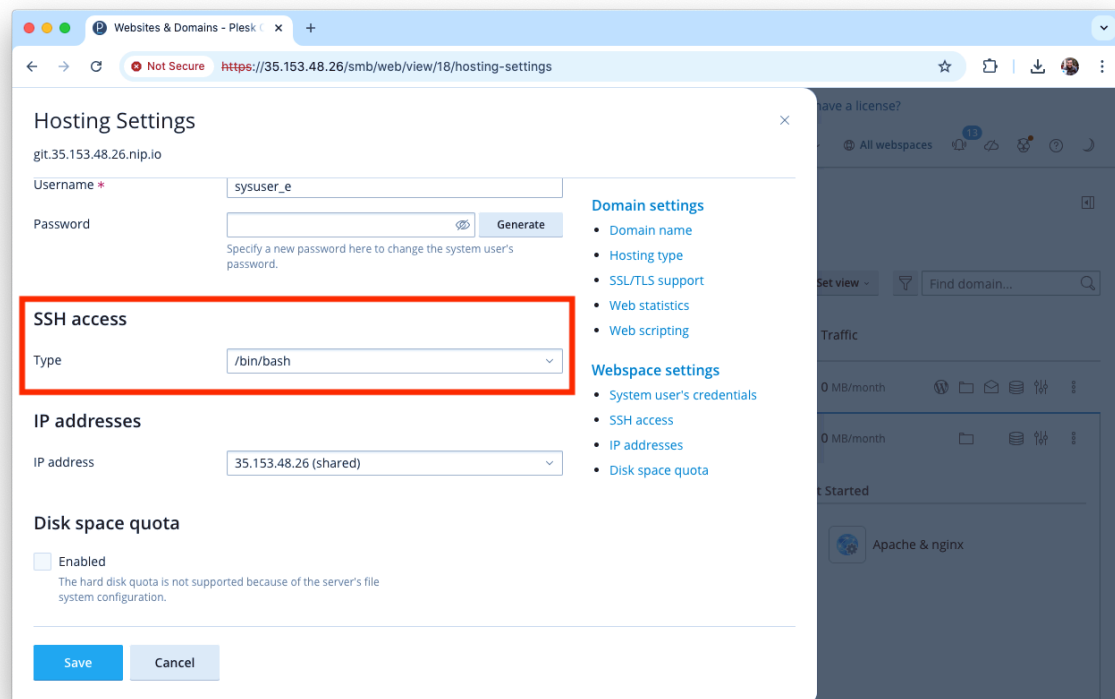


Paso 5. Podemos hacer uso de las `deploy` actions para ejecutar comandos en el servidor después de cada `pull`.

Para poder utilizar todos los comandos del sistema en las `deploy` actions acceda a la ruta:

- `Websites & Domains > Hosting & DNS > Hosting`

Y configure el `SSH access` como `/bin/bash`.

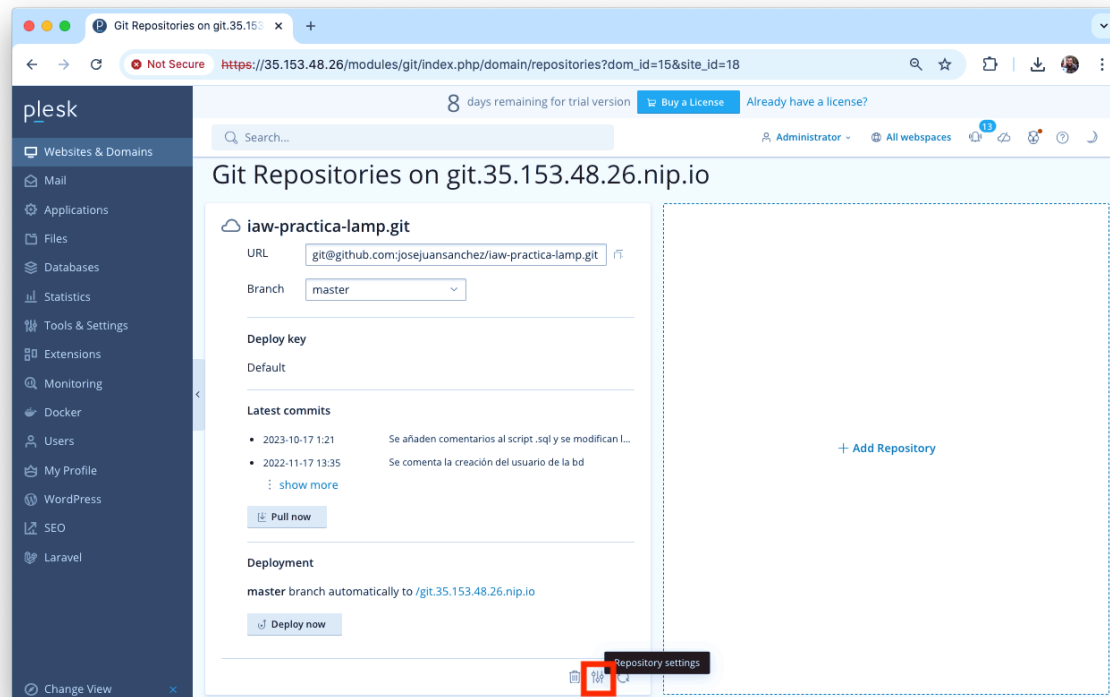


Paso 6. Ahora vamos a configurar qué acciones queremos que se ejecuten después de cada `pull` en las `deploy actions`

Acceda a la ruta:

- `Websites & Domains > Dashboard > Git`

Pinche sobre el icono que permite acceder a los ajustes del repositorio que ha clonado.



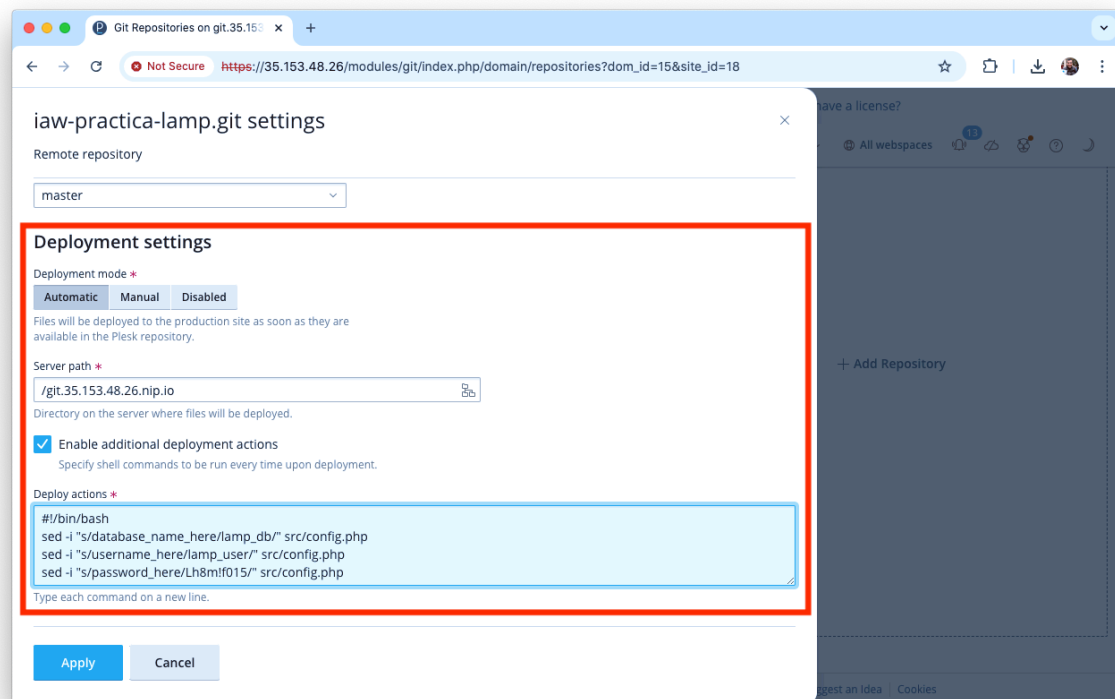
En las `deploy actions` podrá configurar el archivo `config.php` de la aplicación que ha desplegado, para que reemplace automáticamente los valores de configuración de la base de datos por los valores que ha creado en Plesk.

Ejemplo:

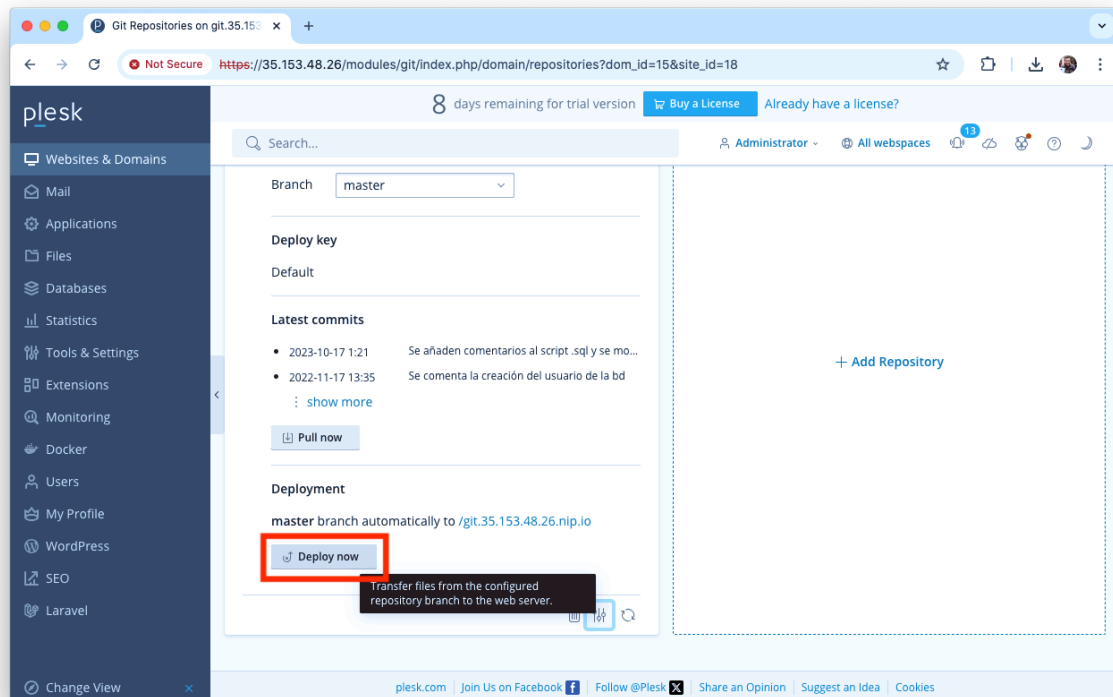
A continuación, se muestran los comandos con algunos valores de ejemplo que se podrían ejecutar en las `deploy actions` para configurar el archivo `config.php` de la aplicación.

```
1 #!/bin/bash
2 sed -i "s/database_name_here/lamp_db/" src/config.php
3 sed -i "s/username_here/lamp_user/" src/config.php
4 sed -i "s/password_here/Lh8m!f015/" src/config.php
```

Nota: Tendrá que indicar el nombre de la base de datos, el nombre del usuario y la contraseña que haya creado previamente en Plesk.



Una vez que ha configurado las `deploy actions` puede pinchar sobre el botón `Deploy now` para que se ejecuten las acciones que ha configurado.

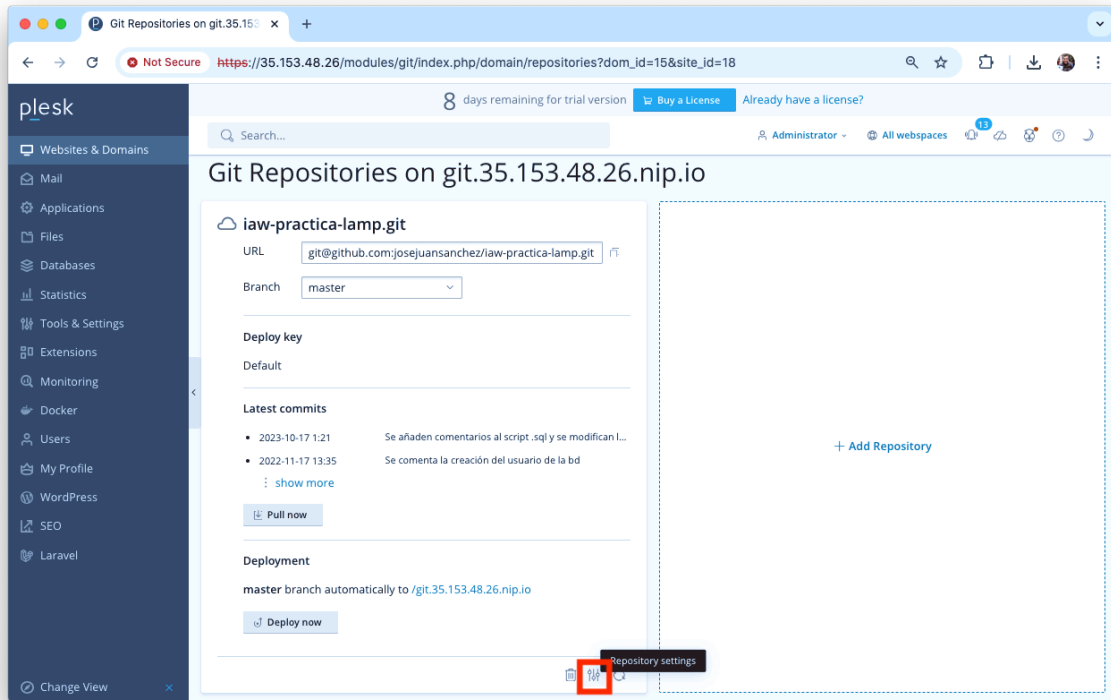


Paso 7. De forma opcional, podemos configurar el uso de [Webhooks de GitHub](#) para hacer `pulls` y despliegues automáticos en Plesk.

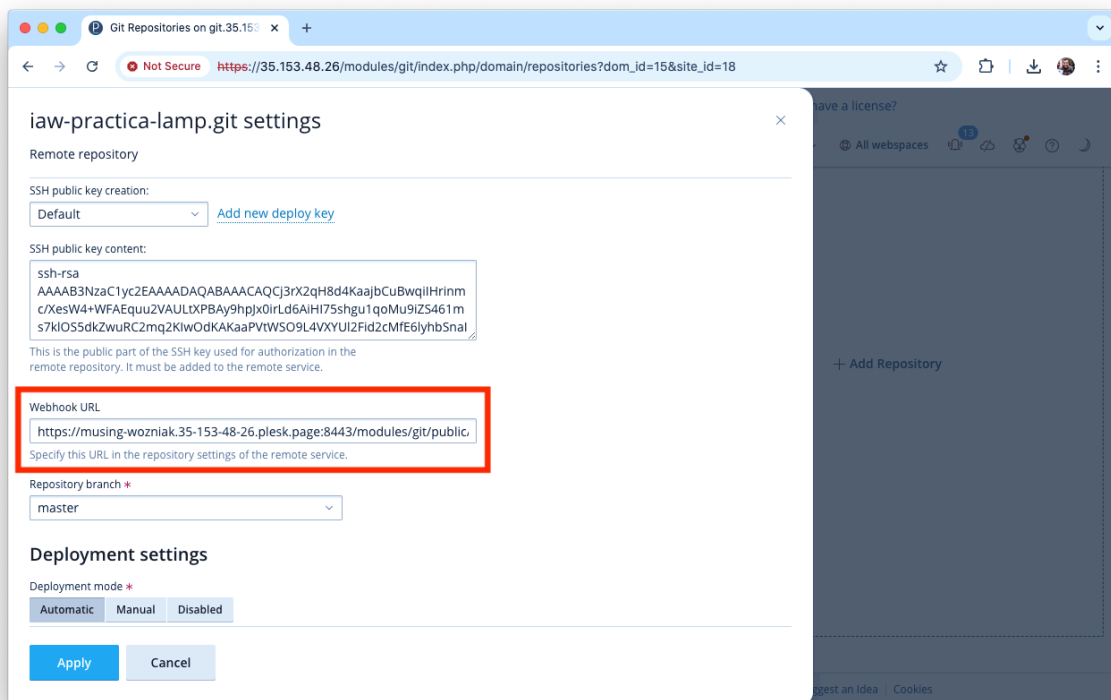
Acceda a la ruta:

- [Websites & Domains](#) > [Dashboard](#) > [Git](#)

Pinche sobre el icono que permite acceder a los ajustes del repositorio que ha clonado.



En esta sección se muestra la URL que ha creado Plesk para crear el Webhook.



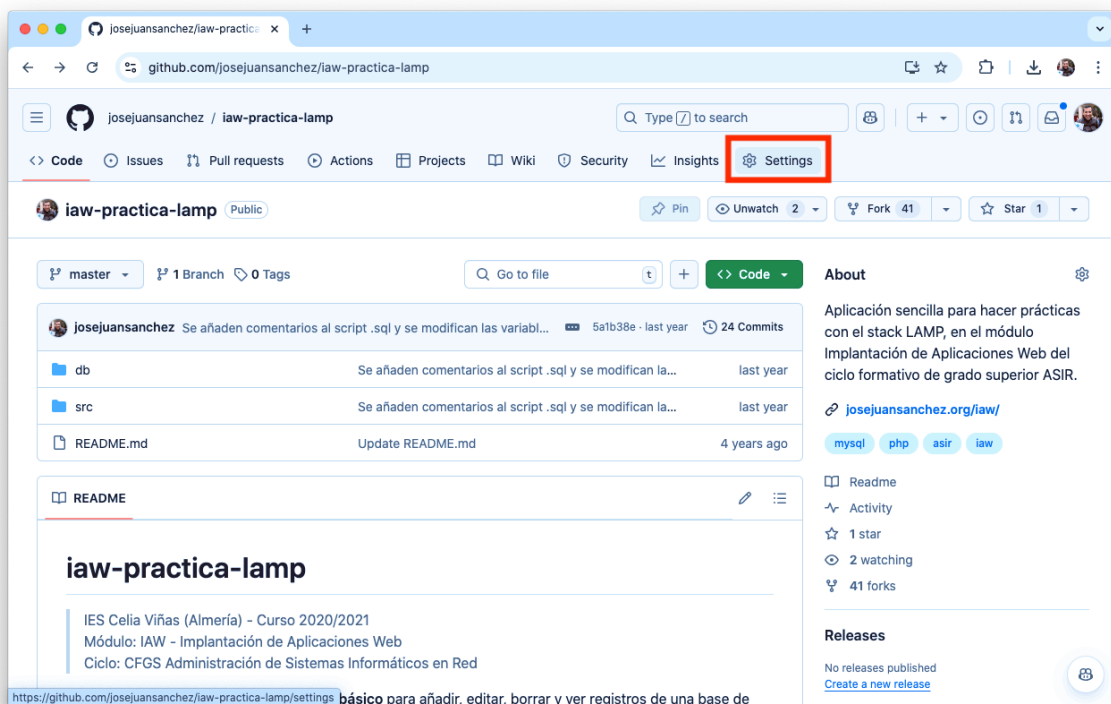
En este ejemplo, se ha creado la siguiente URL:

```
1 https://musing-wozniak.35-153-48-26.plesk.page:8443/modules/git/public/web-hook.php?uuid=7ba3fb0e-0f27-f80d-e8fa-3db472d50c3b
```

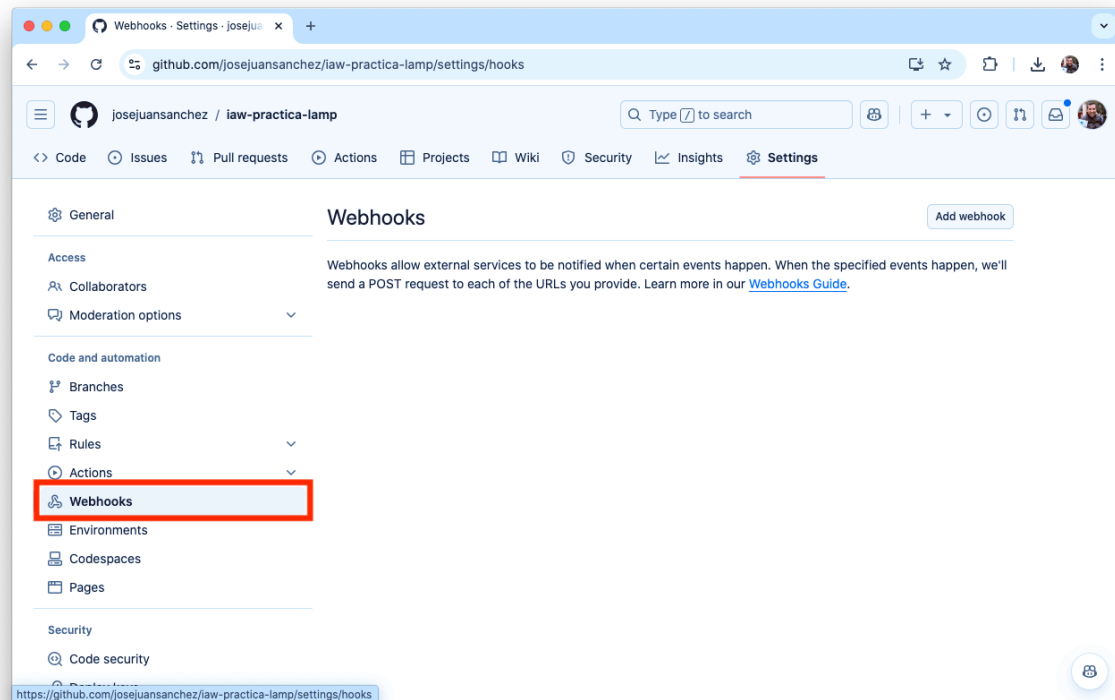
Esta URL habrá que configurarla en el repositorio de GitHub de forma que cada vez que se haga **push** en el repositorio de GitHub, GitHub hará una petición **POST** a la URL del Webhook de Plesk para informarle que tiene que hacer un nuevo **pull** para actualizar el código de la aplicación y volver a desplegarla de forma automática.

Nota: Observe que la URL del **Webhook** utiliza el **puerto 8443**. Por lo tanto, deberá asegurarse de que este puerto está abierto en el grupo de seguridad de la instancia para permitir el tráfico.

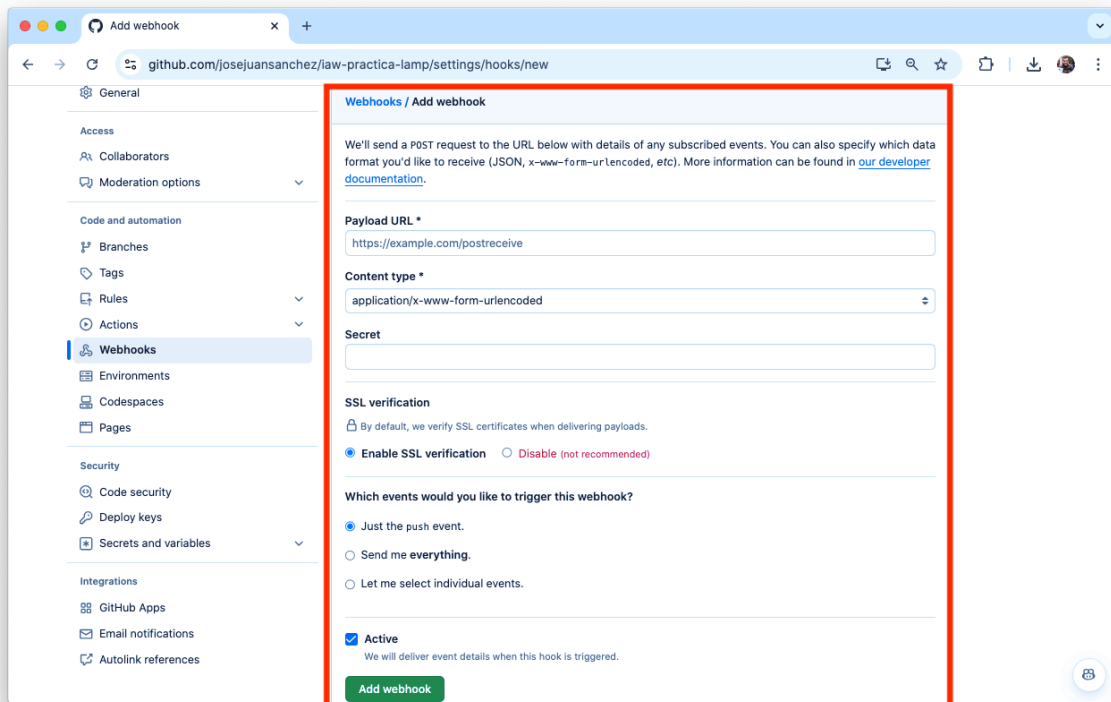
Para configurar el **Webhook** en GitHub acceda a la sección **Settings** del repositorio.



Busque la opción **Webhooks** y pinche sobre el botón **Add webhook**.



Le aparecerá un formulario donde tendrá que indicar cuál es la URL del **Webhook** y que el único que evento que disparará este **Webhook** será el evento **push**.



Una vez hecho esto ya tendremos configurado el **Webhook** en GitHub.

Referencias:

- [Soporte de Git. Documentación oficial de Plesk](#)
- [Uso de hosting remoto de Git. Documentación oficial de Plesk](#)
- [Uso de Webhooks de GitHub para hacer pulls automáticos.](#)
- [Git Deployment actions in Plesk are not executed](#)
- [Cómo clonar un repositorio de GitHub utilizando una clave SSH.](#)

1.9 Despliegue de una aplicación web subiendo los archivos por FTP

En este apartado tendrá que realizar el despliegue de una aplicación web LAMP subiendo los archivos de la aplicación por FTP a Plesk.

En este caso vamos a dar de alta un **subdominio** con el siguiente formato:

```
1 ftp.IP_ESLÁSTICA.nip.io
```

Por ejemplo, si la IP elástica fuera 35 . 153 . 48 . 26 el dominio sería:

```
1 ftp.35.153.48.26.nip.io
```

Por defecto, Plesk sólo permite conexiones por FTP en **modo activo**. Si queremos configurar el **modo pasivo**,

podemos seguir la guía de la documentación oficial, donde se explica cómo abrir un rango de puertos dinámicos para el modo pasivo (49152 – 65535).

Para conectarse al servidor por FTP tendrá que hacer uso de un cliente FTP, como por ejemplo [FileZilla](#).

Para realizar el despliegue tendrá que realizar los siguientes pasos:

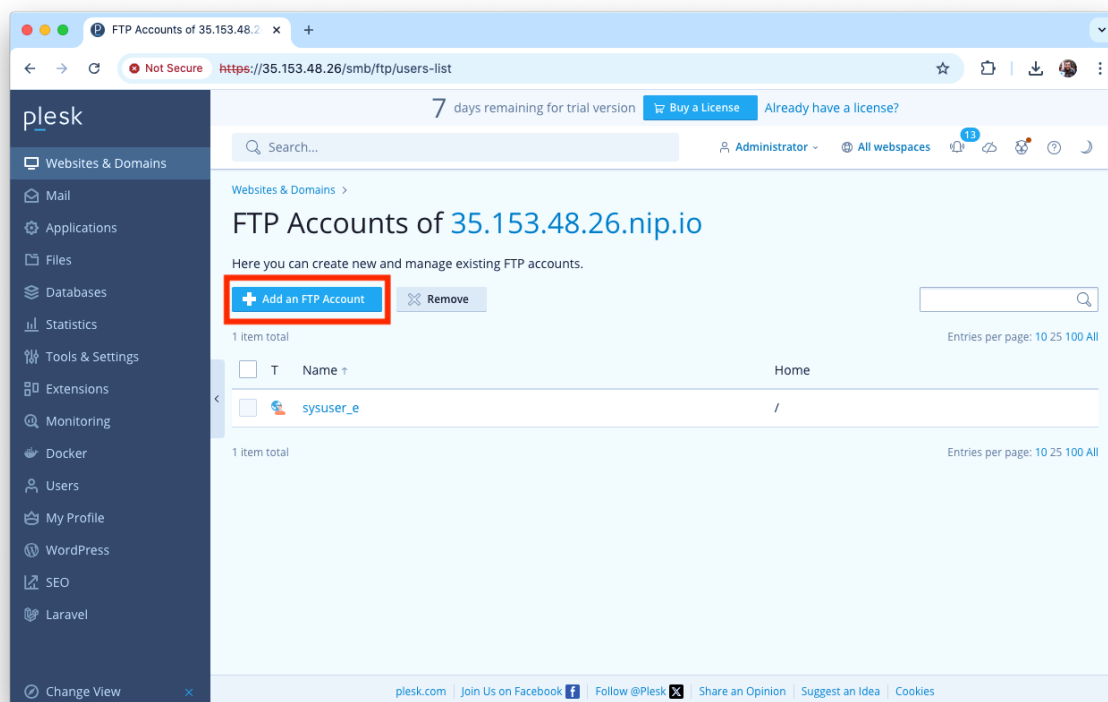
1. Crear una base de datos MySQL para la aplicación web y un usuario con permisos sobre la base de datos.
2. Subir los archivos de la aplicación web por FTP a Plesk.
3. Configurar la aplicación web para que utilice la base de datos que ha creado en Plesk.

Cómo crear un usuario FTP en Plesk

Paso 1. Acceda a la ruta:

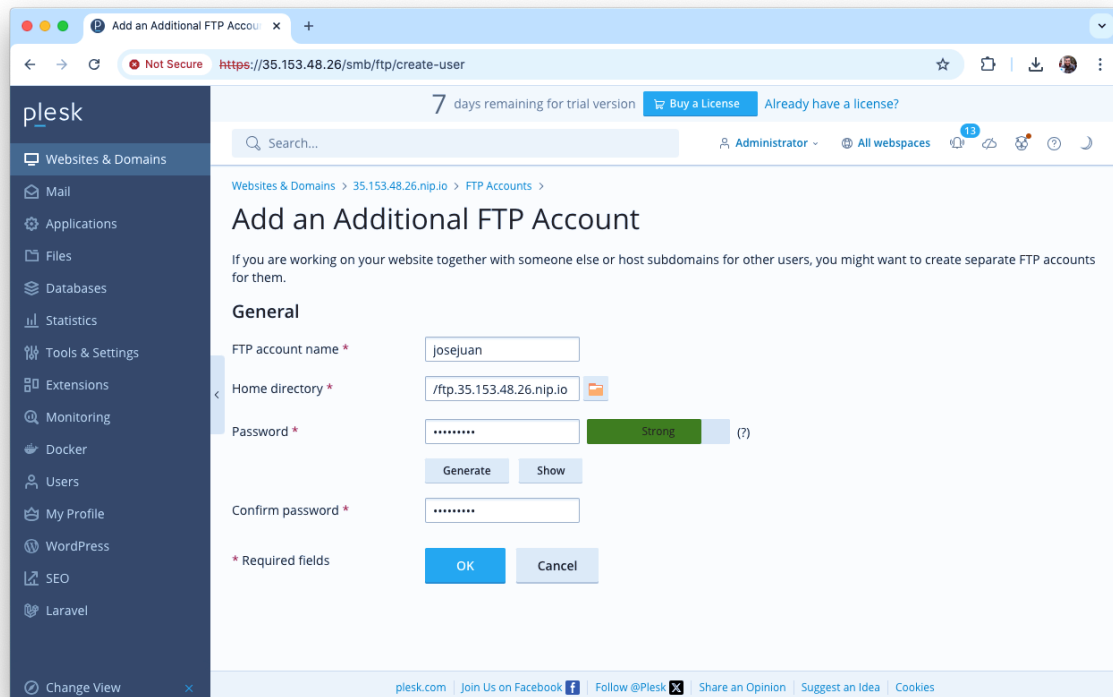
- [Websites & Domains](#) > [Dashboard](#) > [FTP](#)

Pinche sobre el botón [Add an FTP Account](#).



Paso 2. Le aparecerá un formulario donde tiene que indicar:

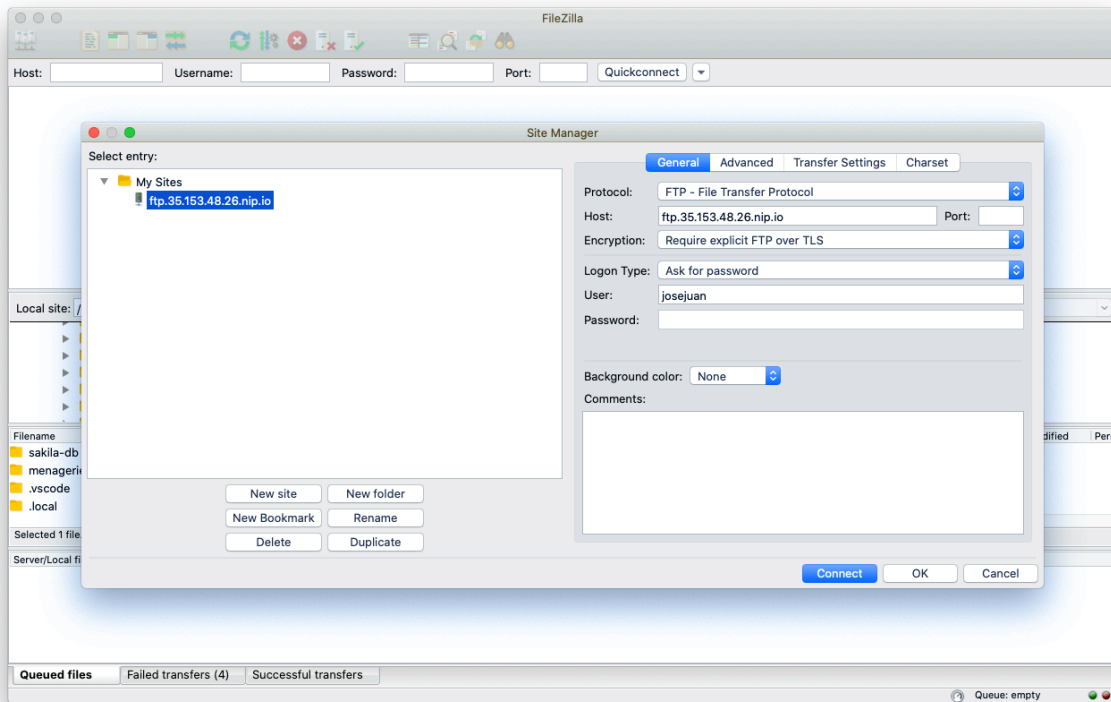
- Nombre y contraseña del usuario FTP.
- Directorio donde tendrá acceso el usuario FTP.



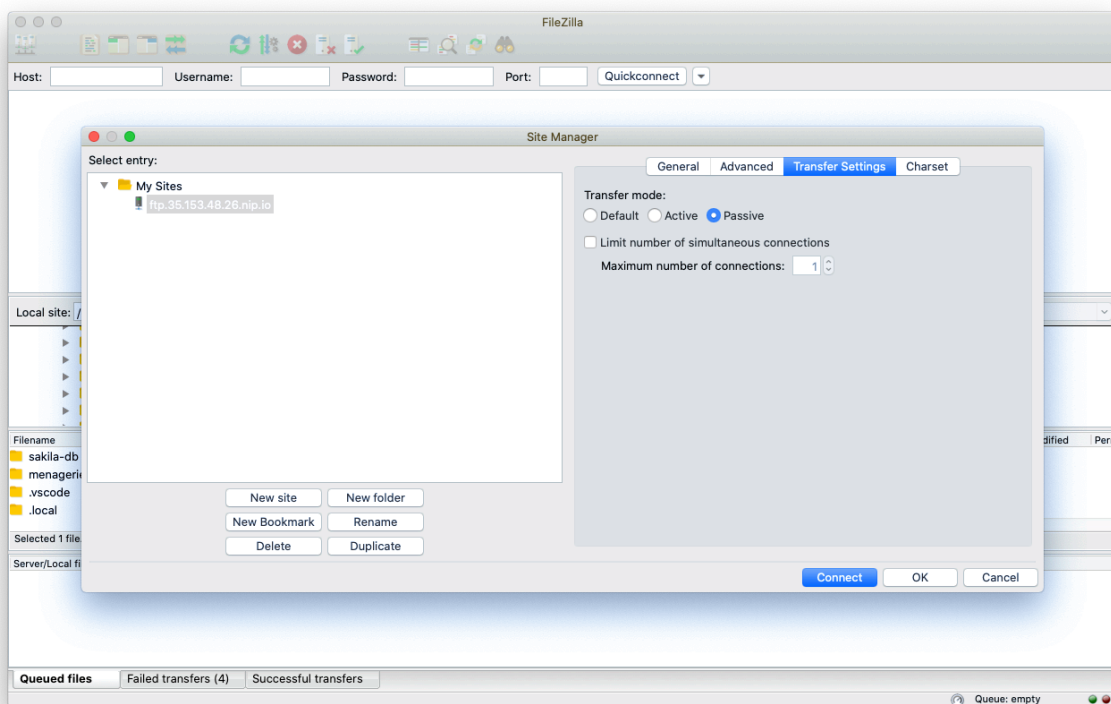
Configuración de una conexión FTP en modo pasivo en FileZilla

Para configurar una conexión por FTP en **modo pasivo** en FileZilla, primero tendrá que indicar los siguientes parámetros en la pestaña **General**:

- **Protocol:** *FTP - File Transfer Protocol*
- **Host:** El nombre de dominio del tipo `ftp.IP_ESLÁSTICA.nip.io`
- **Encryption:** *Require explicit FTP over TLS*
- **Logon type:** *Ask for password*
- **User:** El usuario que ha creado en Plesk



En la pestaña **Transfer Settings** tendrá que indicar que el modo de transferencia será **Passive**.



Configuración del modo activo de FTP en FileZilla

Para configurar el **modo activo** de FTP en **FileZilla**, acceda a la ruta:

- **Edit > Settings > Connection > FTP**

Una vez que haya seleccionado la configuración de **FTP** en el panel izquierdo, tendrá que seleccionar la opción de **Active mode** en el panel derecho dentro de la sección **Transfer mode**.

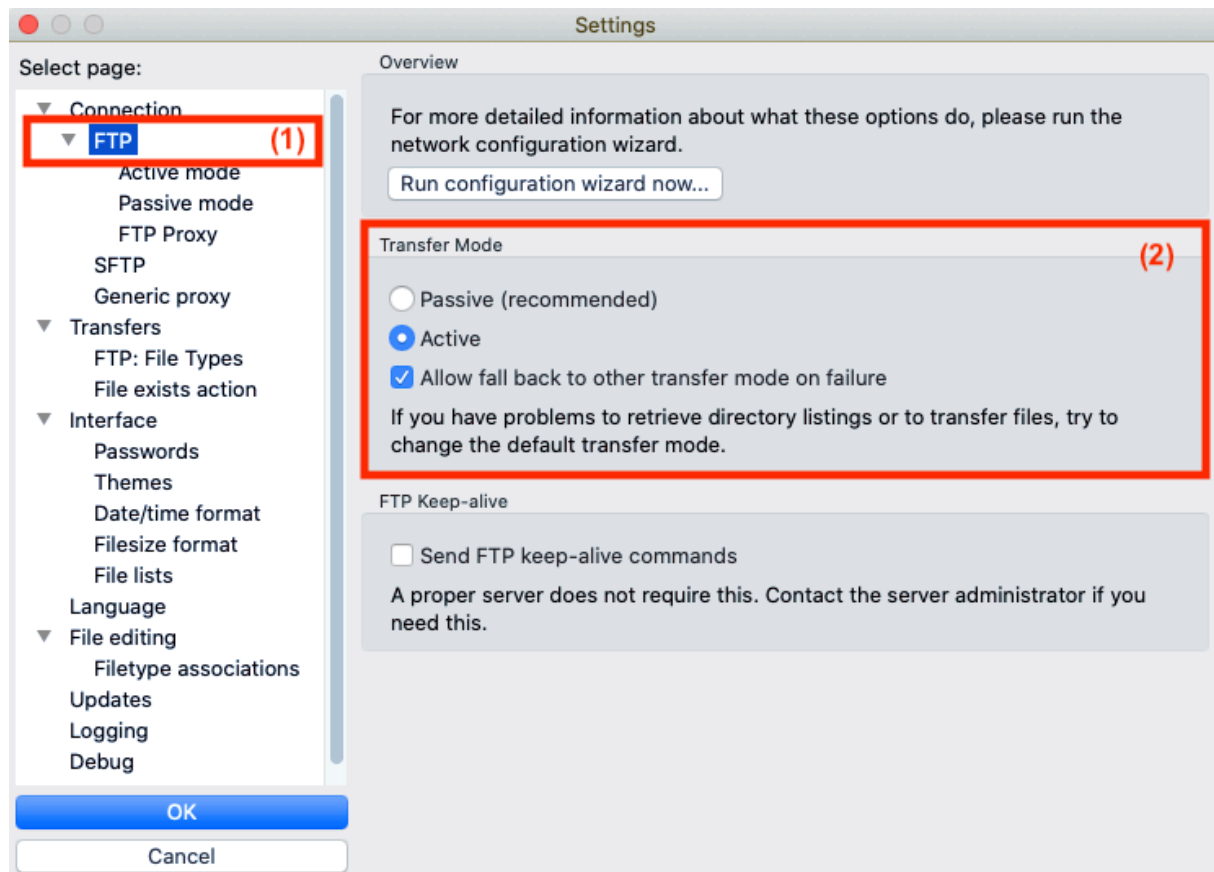


Imagen: Configuración del modo activo de FTP en FileZilla.

Referencias:

- [\(Plesk for Linux\) Configuring Passive FTP Mode](#)
- [How to enable secure FTP \(FTPS\) connections in Plesk for Linux.](#)

1.10 Despliegue de una aplicación Laravel

Plesk dispone de la utilidad **Laravel Toolkit** que permite simplificar el hosting aplicaciones desarrolladas con **Laravel** en Plesk.

En esta práctica tendrá que desplegar una aplicación desarrollada con el framework **Laravel** para que esté accesible a través del subdominio **laravel.web.celia**.

La aplicación será un CRUD muy básico que está disponible en la siguiente URL:

- <https://github.com/josejuansanchez/laravel-crud-app.git>.

Actualizar la versión de PHP de Plesk

El proyecto que vamos a desplegar necesita la versión **8.2** de **PHP** pero esta versión no está disponible en nuestra instalación de Plesk. Por lo tanto, lo primero que tendremos que hacer será instalar la versión necesaria.

Consulte la [documentación oficial de Plesk](#) para realizar esta tarea.

Configurar la versión de PHP en su dominio

Una vez que haya instalado la versión **8.2** de **PHP** tendrá que configurar su dominio para que utilice esta versión de PHP. Para configurarlo, acceda a la siguiente ruta y seleccione la versión 8.2 de PHP.

- [Websites & Domains > Dev Tools > PHP](#)

Despliegue de la aplicación Laravel

Para realizar el despliegue tendrá que realizar los siguientes pasos:

1. Instalamos la extensión de **Laravel Toolkit** y **NodeJS Toolkit**.
2. Creamos un nuevo dominio con **Laravel Toolkit**.
3. Al iniciar la instalación de la aplicación nos aparecen dos opciones:
 - *Install Skeleton.*
 - *Install from remote repository.*

Vamos a seleccionar la segunda opción **Install from remote repository**.

El repositorio que de la aplicación que vamos a desplegar es el siguiente:

- <https://github.com/josejuansanchez/laravel-crud-app.git>

Nota: Si en este paso nos aparece un error por la versión de PHP, tendrá que revisar los apartados anteriores donde se explica cómo actualizar la versión de Plesk y de un dominio en concreto.

4. Creamos la **base de datos** de la aplicación, un usuario y una contraseña.
5. Configuramos el archivo **.env** con los valores de la base de datos.
6. Instalamos las **dependencias** de la aplicación con **composer**.

```
1 composer install
```

7. Creamos la **estructura de la base de datos** con **artisan**.

```
1 php artisan migrate --force
```

8. Instalamos las **dependencias de Node.js**.

```
1 npm install
```

9. Compilamos y minimizamos archivos estáticos (CSS y JavaScript)

```
1 npm run-script build
```

Referencias:

- [Laravel Toolkit](#).
- [Laravel Application Hosting in Plesk](#).
- [How to automatically deploy Laravel in Plesk](#)

1.11 Extensiones de Plesk

Plesk cuenta con una gran cantidad de extensiones gratuitas y de pago. En esta práctica haremos uso de las siguientes extensiones:

Extensión	Descripción
WP Toolkit	Permite instalar, configurar y administrar WordPress de una forma sencilla. Viene instalada por defecto.
Docker	Permite desplegar aplicaciones web en contenedores Docker. Tendrá que instalarla de forma explícita.
Laravel Toolkit	Permite desplegar aplicaciones web Laravel de una forma sencilla. Tendrá que instalarla de forma explícita.
Node.js Toolkit	Permite desplegar aplicaciones web Node.js de una forma sencilla. Tendrá que instalarla de forma explícita.

1.12 Entregables

Deberá crear un repositorio en [GitHub](#) con el nombre de la práctica y añadir al profesor como colaborador.

El repositorio debe contener un **documento técnico** escrito en [Markdown](#) con la descripción de todos los pasos que se han llevado a cabo. Utilice capturas pantallas para mostrar los pasos más importantes.

2 Referencias

- [Plesk](#)
- [OpenStack](#)
- [Ubuntu Server](#)
- [Servidor virtual privado \(VPS\)](#)
- [Servidor dedicado](#)
- [Computación en la nube](#)
- [Imágenes QCOW2 de Plesk](#)
- [Iniciar una máquina Ubuntu en modo de recuperación](#)

3 Licencia

Esta página forma parte del curso Despliegue de Aplicaciones Web de José Juan Sánchez Hernández y su contenido se distribuye bajo una licencia Creative Commons Reconocimiento-NoComercial-CompartirIgual 4.0 Internacional.