

# GUIA DESPLIEGUE Y CONFIGURACIÓN PITCH DREAMERS

## Índice:

### GUIA DESPLIEGUE Y CONFIGURACIÓN PITCH DREAMERS

DESPLIEGUE MEDIANTE CLOUDFLARE	1
Cosas a tener en cuenta Respecto con este despliegue	3
Despliegue mediante AWS LightSail	4
Resultado final:	5
Cosas a tener en cuenta con el despliegue de AWS LightSail	12
Cambios Para alternar entre modo desarrollo y producción (despliegue)	12
	13

## PITCH DREAMERS



## CONFIGURACIÓN INICIAL PROYECTO

1. Configurar el proyecto para que funcione correctamente localmente, pasos de instalación en el Readme.md
2. Configurar y crear los ficheros para el correcto funcionamiento del despliegue
  - a. Ficheros Docker:
    - docker-compose en el back (recomendable ruta raíz pero por temas de subida de repositorio, lo creo aquí).
    - dockerfile uno el front y otro en el back
    - .dockeringore para que no copie el node\_modules ya que el dockerfile al construir la imagen lo hace según las dependencias del package.json
  - b. Ficheros de configuración:
    - Incluir en el fichero package.json del back esta línea: "build": "tsc" para poder hacer el build
    - El fichero nginx.conf en el frontend sirve para configurar el servidor Nginx que se usa como reverse proxy y servidor estático dentro del contenedor Docker del frontend.
    - Las imágenes tenerlo en la carpeta public del frontend
  - c. Configuración de conexión a la bd en docker y en local.

En mi caso he necesitado configurar estos ficheros:

    - .env en la ruta back (mismo motivo que el docker-compose): con las variables de entorno de la base de datos de docker
    - database.ts (credenciales en el código)
    - seedDatabase.ts (credenciales en el código)
    - initializeDatabase.ts (credenciales en el código))
3. Tener Docker Desktop instalado y ejecutar el siguiente comando: docker-compose up --build -d
4. Arreglar errores del proyecto para que se compile correctamente, normalmente los errores son por cosas de tags de typescript o dependencias, elementos no usados, como por ejemplo este:

```
=> [frontend build 5/6] COPY . .
=> ERROR [frontend build 6/6] RUN npm run build

-----
> [frontend build 6/6] RUN npm run build:
0.309
0.309 > frontend@0.0.0 build
0.309 > tsc -b && vite build
0.309
1.329 src/pages/AdminDashboard.tsx(151,35): error TS2322: Type 'unknown' is not assignable to type 'ReactNode'.
-----
failed to solve: process "/bin/sh -c npm run build" did not complete successfully: exit code: 2
```

5. Una vez ya todo funcional el proyecto para que se despliegue correctamente en Docker procedemos a configurar lo necesario para el despliegue

## DESPLIEGUE MEDIANTE CLOUDFLARE

Despliegue Accesible a todos mediante https y dominio será por la aplicación cloudflare Utilizado cloudflare ya que es gratuita, fácil de configurar, proporciona dominio o puedes utilizar otro dominio gratuito, viene incluido ya el certificado SSL y no tengo que exponer el puerto 80 de mi router y de mi pc.

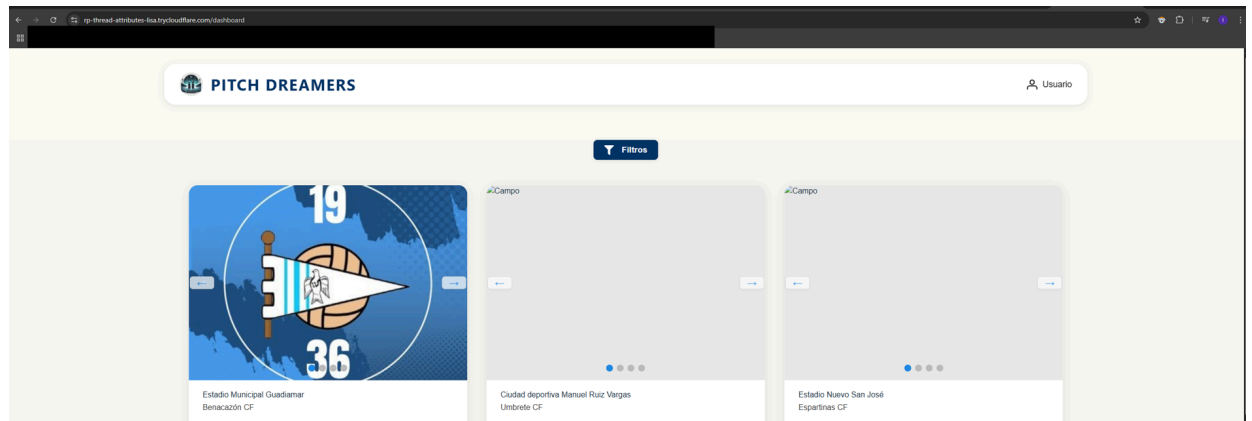
1. Crea una cuenta gratuita en Cloudflare:
  - Ve a <https://dash.cloudflare.com/sign-up> y regístrate.
2. Instalar cloudflare en tu PC:
  - Descarga el ejecutable para Windows desde:  
<https://developers.cloudflare.com/cloudflare-one/connections/connect-apps/install-and-setup/installation/>
  - O ejecuta en PowerShell: `iwr -outf cloudflared.exe https://github.com/cloudflare/cloudflared/releases/latest/download/cloudflared-windows-amd64.exe`
3. En mi caso lo he instalado mediante Powershell. Una vez instalado, mover el cloudflare.exe a la carpeta Windows: C:\Windows
4. Una vez ya en una ruta para poder ejecutarse, en powershell haremos este comando `cloudflare tunnel login` y nos abrirá una pestaña en el navegador, donde aqui podremos poner nuestro dominio, pero como no tenemos ninguno no haremos nada, pudiendo cerrar esta pestaña
5. Ahora tendremos que ejecutar el docker en caso de que lo tengamos parado o eliminado, de igual forma que hicimos antes.
6. Por último expondremos el puerto 80 con su certificado ssl permitiendo conexión https. En la misma terminal de PowerShell haremos este comando:

```
cloudflared tunnel --url http://localhost:80
```

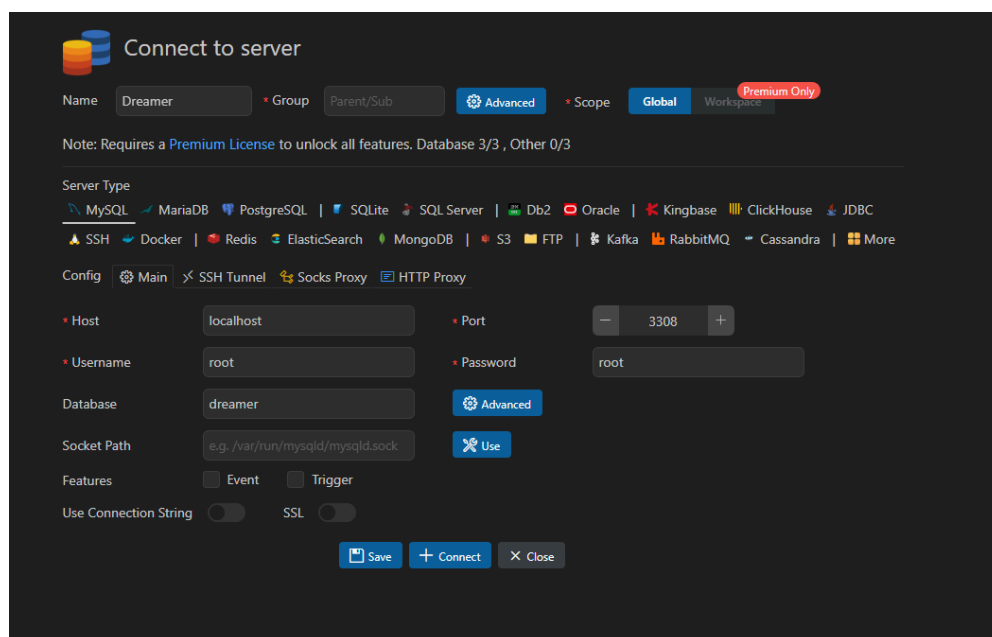
y nos dará una url con un dominio generado automáticamente:

```
PS C:\Users\Ismael> cloudflared tunnel --url http://localhost:80
2025-06-06T00:35:34Z [INFO] Thank you for trying Cloudflare Tunnel. Doing so, without a Cloudflare account, is a quick way
to experiment and try it out. However, be aware that these account-less Tunnels have no uptime guarantee, are subject to
the Cloudflare Online Services Terms of Use (https://www.cloudflare.com/website-terms/), and Cloudflare reserves the r
ght to investigate your use of Tunnels for violations of such terms. If you intend to use Tunnels in production you sho
ld use a pre-created named tunnel by following: https://developers.cloudflare.com/cloudflare-one/connections/connect-ap
s
2025-06-06T00:35:34Z [INFO] Requesting new quick Tunnel on trycloudflare.com...
2025-06-06T00:35:39Z [INFO] Your quick Tunnel has been created! Visit it at (it may take some time to be reachable):
2025-06-06T00:35:39Z [INFO] | https://cp-thread-attributes-lisa.trycloudflare.com
2025-06-06T00:35:39Z [INFO] Cannot determine default configuration path. No file [config.yml config.yaml] in [~/cloudflar
d ~/.cloudflare-warp ~/.cloudflare-warp]
2025-06-06T00:35:39Z [INFO] Version 2025.5.0 (Checksum 27c8c4b706fbc12035eb55b4da208523928e91386c91fd89d1557fecb35cefe6)
2025-06-06T00:35:39Z [INFO] GOOS: windows, GOVersion: go1.22.5-devel-cf, GoArch: amd64
2025-06-06T00:35:39Z [INFO] Settings: map[ha-connections:1 protocol:quic url:http://localhost:80]
2025-06-06T00:35:39Z [INFO] cloudflared will not automatically update on Windows systems.
2025-06-06T00:35:39Z [INFO] Generated Connector ID: b5411969-3388-4278-a419-ecdbeadea246
2025-06-06T00:35:39Z [INFO] Initial protocol quic
2025-06-06T00:35:39Z [INFO] ICMP proxy will use 192.168.1.25 as source for IPv4
2025-06-06T00:35:39Z [INFO] ICMP proxy will use fe80::72c3:3e1a:dd7d:2611 in zone Ethernet as source for IPv6
2025-06-06T00:35:39Z [INFO] cloudflared does not support loading the system root certificate pool on Windows. Please use
origin-ca-pool <PATH> to specify the path to the certificate pool
2025-06-06T00:35:39Z [INFO] ICMP proxy will use 192.168.1.25 as source for IPv4
2025-06-06T00:35:39Z [INFO] Tunnel connection curve preferences: [CurveID(4588) CurveID(25497) CurveP256] connIndex=0 even
0 ip=198.41.200.23
2025-06-06T00:35:39Z [INFO] ICMP proxy will use fe80::72c3:3e1a:dd7d:2611 in zone Ethernet as source for IPv6
2025-06-06T00:35:39Z [INFO] Starting metrics server on 127.0.0.1:20241/metrics
2025-06-06T00:35:40Z [INFO] Registered tunnel connection connIndex=0 connectionId45ea23c7-6a34-4df5-9a8a-dbb6ab632cd9 even
0 ip=198.41.200.23 location=mad04 protocol=quic
```

- Ahora ya tenemos acceso a la aplicación aunque sin dominio, con certificado SSL y público para todo el mundo sin comprometer nuestro ordenador y sin ningún tipo de pago



- Conexión al front y back es de igual forma con la url que nos proporciona la consola de cloudflare solo que cambiando el / por ejemplo para el swagger sería <url automática>/v3/api-docs
- Conexión a la bd:



## Cosas a tener en cuenta Respecto con este despliegue

- El dominio es generado automáticamente y no uno gratuito o personalizado, aunque su versión premium sí permite poner dominios.
- Es completamente gratis.
- Muy fácil de desplegar.
- Es necesario tener la terminal abierta o el pc encendido y en una herramienta de cloud siempre estaría corriendo.
- Necesario tener el docker abierto y los contenedores ejecutándose.
- Hace el certificado https de forma automática.

## Despliegue mediante AWS LightSail

El despliegue mediante AWS LightSail arregla el problema de tener que estar ejecutando el programa en la terminal y tener docker abierto como hace cloudflare, permitiendo tener todo en la nube, también tiene un coste del más básico de 5€ aunque 90 días son gratis, pero son pocos recursos. Requiere una cuenta de AWS teniendo que insertar datos bancarios. Gracias a la cuenta personal mía de AWS de mi empresa Ceroone, puedo realizar este despliegue. Los pasos serán los siguientes:















1. Para desplegar tendremos que poner en el Buscador Lighstasil y una vez dentro a crear instancia.
2. La configuración de la instancia será de la siguiente forma:  
La localización puede ser cualquiera pero todos nuestros datos están alojados en París

### Create an instance

#### Select your instance location [Info](#)

##### Select a Region

The closer your instance is to your users, the less latency they will experience. [Learn more about Regions](#)

<input type="radio"/>  Frankfurt eu-central-1	<input checked="" type="radio"/>  Paris eu-west-3	<input type="radio"/>  London eu-west-2	<input type="radio"/>  Ireland eu-west-1
<input type="radio"/>  Stockholm eu-north-1	<input type="radio"/>  Mumbai ap-south-1	<input type="radio"/>  Montreal ca-central-1	<input type="radio"/>  Virginia us-east-1
<input type="radio"/>  Ohio us-east-2	<input type="radio"/>  Singapore ap-southeast-1	<input type="radio"/>  Seoul ap-northeast-2	<input type="radio"/>  Tokyo ap-northeast-1
<input type="radio"/>  Oregon us-west-2	<input type="radio"/>  Sydney ap-southeast-2		

#### Availability Zone - optional [Info](#)

You are creating this instance in **Zone A** (eu-west-3a)

[Change your Availability Zone](#)

#### Pick your instance image [Info](#)












The instance image you pick determines the operating system and whether there are any included applications in your instance.

##### Select a platform

<input checked="" type="radio"/>  Linux/Unix 27 blueprints	<input type="radio"/>  Microsoft Windows 6 blueprints
--	---

##### Select a blueprint

Apps + OS | **Operating System (OS) only**

<input type="radio"/>  Amazon Linux 2023 2023.7.20250527.1	<input type="radio"/>  Amazon Linux 2 2.0.20250603.0	<input checked="" type="radio"/>  Ubuntu 24.04 LTS	<input type="radio"/>  Ubuntu 22.04 LTS
<input type="radio"/>  Debian 12.8	<input type="radio"/>  Debian 11.11	<input type="radio"/>  FreeBSD 14.2	<input type="radio"/>  FreeBSD 13.4
<input type="radio"/>  openSUSE 15.6	<input type="radio"/>  AlmaLinux 9.4	<input type="radio"/>  CentOS CS9-20230110	

Ubuntu 24.04 LTS

### 3. Luego seleccionamos el plan, el nombre de la instancia y la creamos:

**Choose your instance plan** [Info](#)  
Instance plans include the instance's network type and a size which determines what bundled compute resources the instance will have.

**Select a network type** [Info](#)

☒ **Dual-stack** Recommended  
For workloads that require full network compatibility. Includes a public IPv4 and a public IPv6 address.

☐ **IPv6-only**  
For workloads that do not require a public IPv4 address. Includes a public IPv6 address.

**Select a size**

Sort by Price per month ▾

<input type="radio"/> <b>\$5</b> USD per month  512 MB Memory 2 vCPUs Processing 20 GB SSD Storage 1 TB Transfer <a href="#">First 90 days free</a>	<input type="radio"/> <b>\$7</b> USD per month  1 GB Memory 2 vCPUs Processing 40 GB SSD Storage 2 TB Transfer <a href="#">First 90 days free</a>	<input checked="" type="radio"/> <b>\$12</b> USD per month  2 GB Memory 2 vCPUs Processing 60 GB SSD Storage 3 TB Transfer <a href="#">First 90 days free</a>	<input type="radio"/> <b>\$24</b> USD per month  4 GB Memory 2 vCPUs Processing 80 GB SSD Storage 4 TB Transfer
<input type="radio"/> <b>\$44</b> USD per month  8 GB Memory 2 vCPUs Processing 160 GB SSD Storage 5 TB Transfer	<input type="radio"/> <b>\$84</b> USD per month  16 GB Memory 4 vCPUs Processing 320 GB SSD Storage 6 TB Transfer	<input type="radio"/> <b>\$164</b> USD per month  32 GB Memory 8 vCPUs Processing 640 GB SSD Storage 7 TB Transfer	<input type="radio"/> <b>\$384</b> USD per month  64 GB Memory 16 vCPUs Processing 1,280 GB SSD Storage 8 TB Transfer <span>Largest plan</span>

For a limited time, new Lightsail customers can try the selected plan for free for 90 days. [Learn more about the free trial in Lightsail](#)

---

**Identify your instance**

**Instance name**  
Instance names help you identify an instance once it's created. The instance name must be unique in the AWS Region for your Lightsail account.

×

**Tagging options - optional** [Info](#)  
You can specify tags to assign to this resource after it's created. Key-value tags are used to filter and organize your resources, organize your billing, and control access to your resources in the Lightsail console.

No tags associated with the resource.

### 4. Una vez, creada entramos a la instancia y en el apartado Networking agregaremos la regla HTTPS y el puerto 3308 para la BD. dándole a Add rule

#### IPv4 Firewall [?](#)

Create rules to open ports to the internet, or to a specific IPv4 address or range.

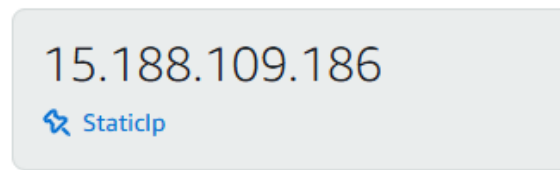
[Learn more about firewall rules](#)

[+](#) Add rule

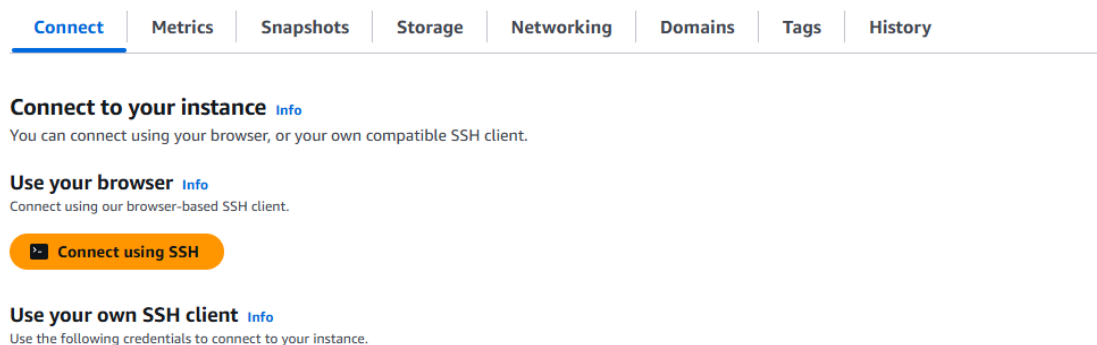
Application	Protocol	Port or range / Code	Restricted to		
SSH	TCP	22	Any IPv4 address Lightsail browser SSH/RDP <a href="#">?</a>	<a href="#">✎</a>	<a href="#">🗑</a>
HTTP	TCP	80	Any IPv4 address	<a href="#">✎</a>	<a href="#">🗑</a>
HTTPS	TCP	443	Any IPv4 address	<a href="#">✎</a>	<a href="#">🗑</a>
Custom	TCP	3308	Any IPv4 address	<a href="#">✎</a>	<a href="#">🗑</a>

5. Aquí mismo también crearemos una ip estática para que el dominio siempre apunte a la misma ip aunque se reinicie la instancia, tendremos que darle en el botón naranja de attach static ip y asignarle un nombre único.

PUBLIC IPV4



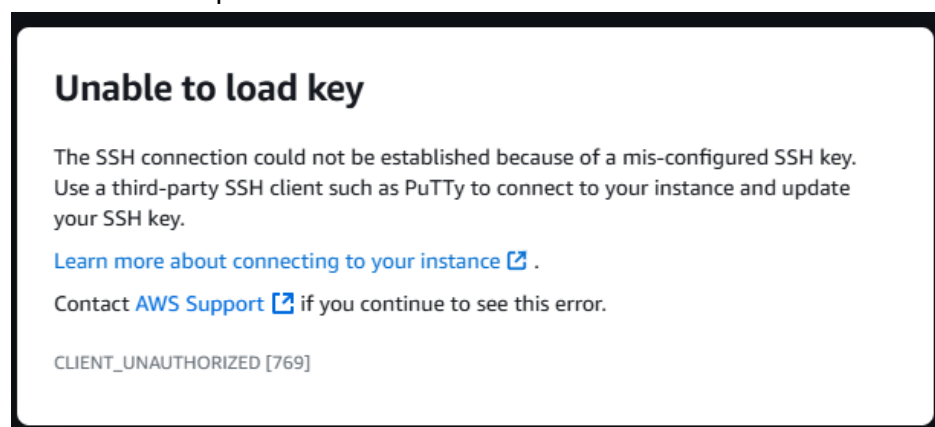
6. Ahora tendremos que conectarnos a la instancia ubuntu mediante SSH, abriendo la terminal botón connect using SSH



7. Una vez dentro de la terminal tendremos que ejecutar todos estos comandos:

```
sudo apt update
sudo apt install docker.io docker-compose -y
sudo usermod -aG docker $USER
```

8. Una vez ejecutado e instalado todo, cerramos y abrimos la terminal la instancia para que la virtualización se realice.
9. Solo en caso de que salga este error hacer todo el paso 9, si no ir directamente al paso 10:



Tendremos que conectarnos mediante ssh de nuestra terminal de la siguiente forma:



-Descargamos la clave ssh en el mismo apartado donde nos conectamos a la terminal que habra un boton llamado

#### SSH key

This instance uses the default SSH key for the Paris (eu-west-3) Region.

[↓ Download default key](#)

-También tendremos que tener instalado OpenSSH Client para poder hacerlo en nuestra terminal en Windows siguiendo estos pasos:

- Abre el menú de inicio y busca "Características opcionales".
- Haz clic en "Agregar una característica"
- Busca **OpenSSH Client**.
- Haz clic en "Instalar"

- Luego en la terminal pondremos el siguiente comando, `ssh -i <clave> ubuntu@<ip static> ubuntu` es el username por el tipo de instancia

```
PS C:\Users\Ismael\Downloads> ssh -i .\LightsailDefaultKey-eu-west-3.pem ubuntu@15.188.109.186
The authenticity of host '15.188.109.186 (15.188.109.186)' can't be established.
ED25519 key fingerprint is SHA256:wNBpAyYcq6h3eLEfsvSyHo889CQ81boZ2z1v5QoiTMA.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '15.188.109.186' (ED25519) to the list of known hosts.
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1018-aws x86_64)

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

System information as of Fri Jun  6 13:28:40 UTC 2025

System load:          0.0
Usage of /:           13.3% of 18.33GB
Memory usage:         58%
Swap usage:           0%
Temperature:         -273.1 C
Processes:            110
Users logged in:      0
IPv4 address for ens5: 172.26.0.122
IPv6 address for ens5: 2a05:d012:931:1500:feec:f15f:5a00:3266

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

1 additional security update can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

*** System restart required ***
Last login: Fri Jun  6 13:13:26 2025 from 52.46.83.56
ubuntu@ip-172-26-0-122:~$
```



10. Una vez dentro de nuevo haremos este procedimiento para tener el proyecto en la instancia:

```
Last login: Fri Jun  6 14:21:45 2025 from 52.46.83.60
ubuntu@ip-172-26-14-245:~$ docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
ubuntu@ip-172-26-14-245:~$ ls
ubuntu@ip-172-26-14-245:~$ mkdir PitchDreamers
ubuntu@ip-172-26-14-245:~$ ls
PitchDreamers
ubuntu@ip-172-26-14-245:~$ cd PitchDreamers/
ubuntu@ip-172-26-14-245:~/PitchDreamers$ git clone https://github.com/ismaelsevidanes/back
end.git
Cloning into 'backend'...
remote: Enumerating objects: 361, done.
remote: Counting objects: 100% (361/361), done.
remote: Compressing objects: 100% (187/187), done.
remote: Total 361 (delta 157), reused 295 (delta 96), pack-reused 0 (from 0)
Receiving objects: 100% (361/361), 822.57 KiB | 12.28 MiB/s, done.
Resolving deltas: 100% (157/157), done.
ubuntu@ip-172-26-14-245:~/PitchDreamers$ git clone https://github.com/ismaelsevidanes/fron
tend.git
Cloning into 'frontend'...
remote: Enumerating objects: 321, done.
remote: Counting objects: 100% (321/321), done.
remote: Compressing objects: 100% (222/222), done.
remote: Total 321 (delta 187), reused 215 (delta 86), pack-reused 0 (from 0)
Receiving objects: 100% (321/321), 468.16 KiB | 8.36 MiB/s, done.
Resolving deltas: 100% (187/187), done.
ubuntu@ip-172-26-14-245:~/PitchDreamers$ ls
backend  frontend
ubuntu@ip-172-26-14-245:~/PitchDreamers$
```

11. A continuación, instalaremos el paquete para los certificados en la instancia

```
sudo apt install certbot
```

12. A continuación vamos a generar un dominio como el proyecto es de pruebas será uno gratuito en <https://www.duckdns.org/> .Es muy fácil generar un dominio solo loguearse y poner el nombre del dominio y una vez creado poner en el current ip la ip estática de la instancia de LightSail

success: ip address for pitchdreamers.duckdns.org updated to 15.188.109.186

domains 1/5

http:// sub domain .duckdns.org add domain

domain	current ip	ipv6	changed
pitchdreamers	15.188.109.186 <a href="#">update ip</a>	ipv6 address <a href="#">update ipv6</a>	0 seconds ago <a href="#">delete domain</a>

This site is protected by reCAPTCHA and the Google

[Privacy Policy](#) and [Terms of Service](#) apply.

13. Volvemos A la terminal de la instancia y ponemos este comando:

```
sudo certbot certonly --standalone -d pitchdreamers.duckdns.org
```

Esto nos pedirá un email y aceptar los términos y si hemos hecho todo bien nos saldrá un mensaje como este. ismaelshenshu3000@gmail.com

```
ubuntu@ip-172-26-14-245:~/PitchDreamers$ sudo certbot certonly --standalone -d pitchdreamers.duckdns.org
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Requesting a certificate for pitchdreamers.duckdns.org

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/pitchdreamers.duckdns.org/fullchain.pem
Key is saved at: /etc/letsencrypt/live/pitchdreamers.duckdns.org/privkey.pem
This certificate expires on 2025-09-04.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

- - - - -
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-172-26-14-245:~/PitchDreamers$
This certificate expires on 2025-09-04.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.

No containers need to be restarted.
No user sessions are running outdated binaries.
ubuntu@ip-172-26-14-245:~/PitchDreamers$ sudo certbot certonly --standalone -d pitchdreamers.duckdns.org
Saving debug log to /var/log/letsencrypt/letsencrypt.log
Requesting a certificate for pitchdreamers.duckdns.org

Successfully received certificate.
Certificate is saved at: /etc/letsencrypt/live/pitchdreamers.duckdns.org/fullchain.pem
Key is saved at: /etc/letsencrypt/live/pitchdreamers.duckdns.org/privkey.pem
This certificate expires on 2025-09-04.
These files will be updated when the certificate renews.
Certbot has set up a scheduled task to automatically renew this certificate in the background.
```

14. Una vez ya con los certificados le daremos los permisos al contenedor con estos dos comandos:

```
sudo chmod -R a+r
/etc/letsencrypt/live/pitchdreamers.duckdns.org
sudo chmod -R a+r
/etc/letsencrypt/archive/pitchdreamers.duckdns.org
```

15. Modificar el fichero nginx.conf y en la línea server\_name poner el dominio y toda la configuración de los certificados

```
server {
    listen 80;
    server_name pitchdreamers.duckdns.org;
    return 301 https://$host$request_uri;
}

server {
    listen 443 ssl;
    server_name pitchdreamers.duckdns.org;

    You, 1 hour ago via PR #3 • Ultimos cambios del nginx.conf para despliegue...

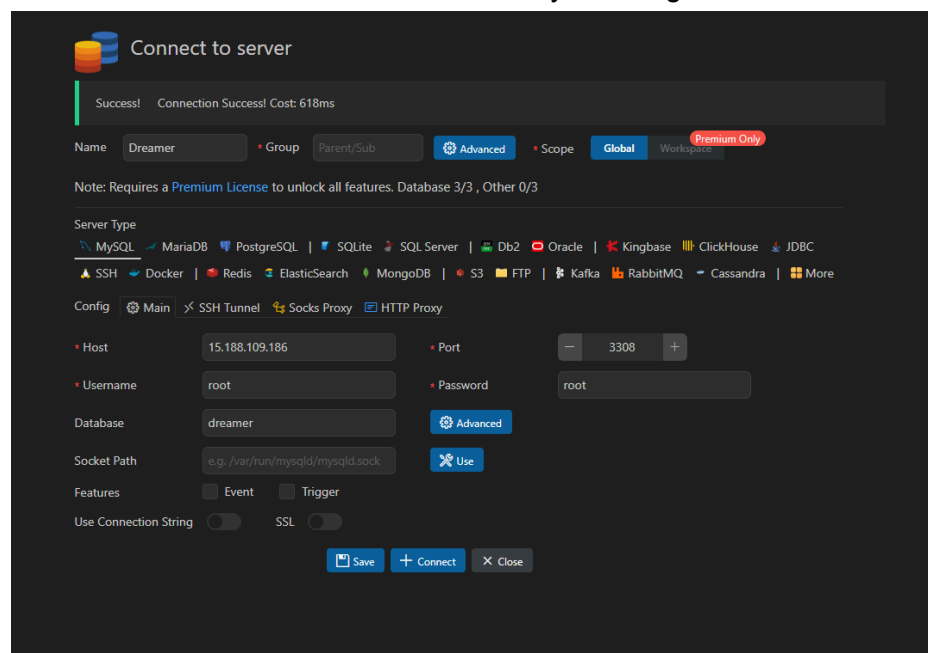
    ssl_certificate /etc/letsencrypt/live/pitchdreamers.duckdns.org/fullchain.pem;
    ssl_certificate_key /etc/letsencrypt/live/pitchdreamers.duckdns.org/privkey.pem;

    ssl_protocols TLSv1.2 TLSv1.3;
}
```

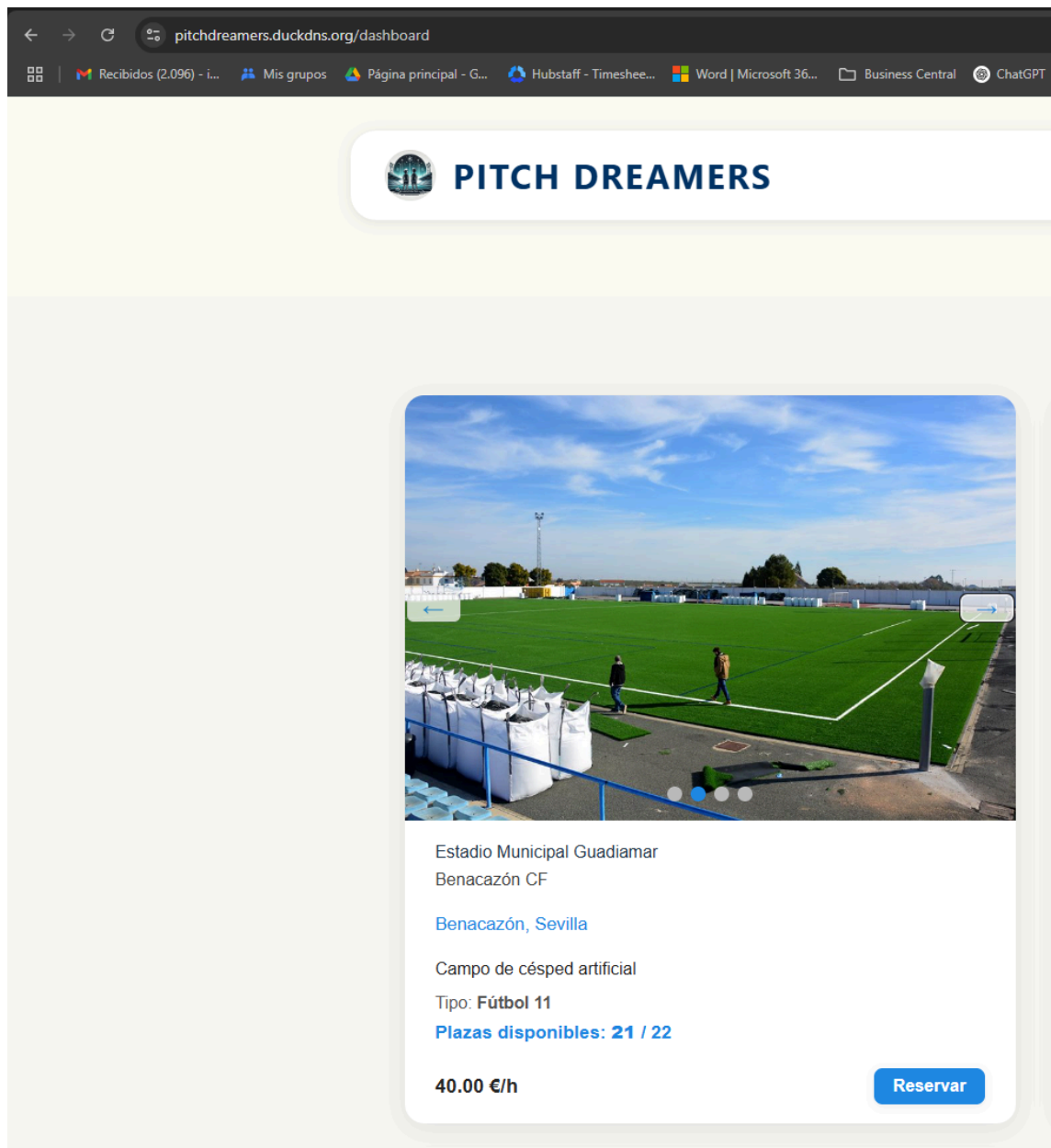
16. También modificaremos el docker-compose y pondremos esto:

```
Run Service
frontend:
  build:
    context: ../frontend
    dockerfile: Dockerfile
  depends_on:
    - backend
  ports:
    - "80:80"
  # Comentar el volumes en caso de desplegarlo mediante Cloudflare
  volumes:
    - /etc/letsencrypt/live/pitchdreamers.duckdns.org:/etc/nginx/certs:ro
    - /etc/letsencrypt/archive/pitchdreamers.duckdns.org:/etc/nginx/archive:ro
  restart: always
```

17. Y por último tendremos que actualizar el repositorio con estas últimas modificaciones y lanzar el docker-compose up --build -d . Para ello nos dirigimos a la carpeta del backend ya que es donde está el docker-compose.
18. Listo ya el despliegue está funcionando correctamente, ahora queda acceder a ella mediante url:
  - Frontend: <https://pitchdreamers.duckdns.org> | [pitchdreamers.duckdns.org](https://pitchdreamers.duckdns.org) o por la ip estatica: 15.188.109.186, aunque este no tiene el nivel de seguridad
  - Backend: <https://pitchdreamers.duckdns.org/v3/api-docs> para el swagger
  - Base de datos: Desde cualquier herramienta como DBeaver, o en mi caso una extensión de visual studio code y lo configuraremos así:



## Resultado final:



## Cosas a tener en cuenta con el despliegue de AWS LightSail

1. Necesitas tener cuenta de AWS y requiere cuentas bancarias
2. No es gratuito, cuesta 5€ pero ese plan se queda corto a nivel de recursos, lo mínimo recomendable sería el plan de 12€
3. Es bastante más complejo el despliegue que cloudflare .
4. No es necesario tener la terminal de la instancia abierta, solo que la instancia y los contenedores de docker estén ejecutándose
5. No depende de tener docker y la terminal de cloudflare abierto para que esté corriendo ya que es todo en la nube
6. Necesitas instalar los certificados de forma manual.
7. Le puedes poner el dominio tuyo que quieras, ya sea gratuito o de pago

## Cambios Para alternar entre modo desarrollo y producción (despliegue)

1. En el fichero backend/config/swagger.ts alternar el comentario entre estas líneas

```
servers: [  
  {  
    url: 'http://localhost:3000', // Cambiar esto si se usa otro puerto  
    //url: 'https://pitchdreamers.duckdns.org', // URL de producción  
  },  
],
```

**Autor:** Ismael Sevidanes Del Moral

**Tutor:** Antonio Gabriel González Casado

**Tutor del Proyecto:** Mónica María Marcos Gutiérrez