

# Tarea 1: Continuous deployment (CD)

## 1. Elección de la Plataforma

### 1.1 Comparativa de las opciones más destacadas y sencillas:

Plataforma	Recursos Gratuitos	Mejor Uso	Capa de Pago Necesaria
Render	512 MB RAM, 0.5 vCPU, 750 MB disk	Backend y servicios Dockerizados	Mayor RAM/CPU o uptime
Vercel	100 GB ancho de banda, serverless	Frontend estático o Next.js	Más builds o tráfico
Netlify	125k invocaciones, 100 GB ancho	Frontend estático y serverless ligero. No soporta docker	Más tráfico/invocaciones
Heroku	Sin plan gratuito (mínimo 5\$)	Prototipos backend o front simple, dockerizados	Actividad constante
Railway	\$5 créditos (~500 horas)	Servicios Dockerizados y bases de datos	Más créditos
Fly.io	Sin plan gratuito pay per use	Backend avanzado y distribuido	Más RAM/CPU/storage

### 1.2. Elección Final:

La plataforma elegida es **RENDER** por ser la única que tiene un plan gratuito a coste 0 y que además tiene soporte para desplegar apps dockerizadas desde dockerhub. El plan más generoso es el de VERCEL, pero sin embargo este hace el despliegue mediante la clonación del repositorio de github, similar al github action checkout.

He decidido dockerizar y desplegar una pequeña app que hice en verano para aprender a usar el framework next.js. Utiliza la librería fakers.js.

**Repositorio de github:** <https://github.com/bmascot/data-generator>

**Último commit:** feat(workflows): add condition to deploy for events different of pull request ( c0e2bfee50c528a8f9a62dbbb3251830776ce7ca )

## 2. Configuración en Render

### 1. Registro en Render:

- Crea una cuenta en [Render](#) si aún no tienes una.



Sign In to Render

 GitHub	 GitLab
 Bitbucket	 Google

or

Email

Password

Sign in

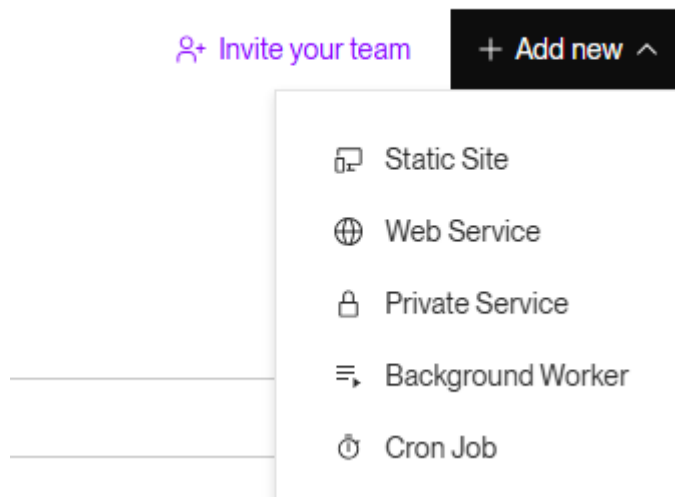
[Sign in with SSO](#)

Need an account? [Sign up](#)

Forgot your password? [Reset it](#)

### 2. Nuevo Servicio Web:

- En tu dashboard de Render, haz clic en **+ Add new > Web Service**.




- Selecciona la opción **Existing Image** y añadimos el link a nuestra imagen en el registro de Docker Hub (por ejemplo, `docker.io/<dockerhub-username>/<repository-name>:latest`).

Git Provider

Public Git Repository

Existing Image

Image URL  
Deploy an image from a Docker registry

 docker.io/bmascat/data-generator:latest

Credential (Optional)

No credential

Connect →

### 3. Conectar con Docker Hub:

- Configura los siguientes parámetros:
  - **Instance Type:** Selecciona la instancia gratuita.

#### Instance Type

For hobby projects

Free

\$0 / month

512 MB (RAM)

0.1 CPU

⚠ Upgrade to enable more features

Free instances spin down after periods of inactivity. They do not support SSH access, scaling, one-off jobs, or persistent disks. Select any paid instance type to enable these features.

For professional use

For more power and to get the most out of Render, we recommend using one of our paid instance types. All paid instances support:

- Zero Downtime
- SSH Access
- Scaling
- One-off jobs
- Support for persistent disks

Starter

\$7 / month

512 MB (RAM)

0.5 CPU

Standard

\$25 / month

2 GB (RAM)

1 CPU

Pro

\$85 / month

4 GB (RAM)

2 CPU

Pro Plus

\$175 / month

8 GB (RAM)

4 CPU

Pro Max

\$225 / month

16 GB (RAM)

4 CPU

Pro Ultra

\$450 / month

32 GB (RAM)

8 CPU

### 4. Despliegue:

- Render hará el pull de la imagen y desplegará automáticamente el servicio.
- Una vez finalizado, Render generará una URL pública para tu aplicación.
- En mi caso es: <https://data-generator-4ghg.onrender.com/>

WEB SERVICE

**data-generator**

Image

Free

[Upgrade your instance →](#)

Connect

Manual Deploy

bmascat / data-generator

latest

<https://data-generator-4ghg.onrender.com>

ⓘ Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more. [Upgrade now](#)

January 21, 2025 at 5:58 PM Live  
Deploy for 86db6ae 

latest

All logs

Search

Jan 21, 5:57 PM - 5:59 PM

GMT+1

↑

↕

```

Jan 21 05:58:37 PM ➤ ==> Starting service...
Jan 21 05:58:47 PM ➤ ▲ Next.js 14.2.13
Jan 21 05:58:47 PM ➤ - Local:      http://localhost:10000
Jan 21 05:58:47 PM ➤ - Network:   http://0.0.0.0:10000
Jan 21 05:58:47 PM ➤
Jan 21 05:58:47 PM ➤ ✓ Starting...
Jan 21 05:58:48 PM ➤ ✓ Ready in 681ms
Jan 21 05:58:58 PM ➤ ==> Your service is live 🎉
  
```

- En render el primer despliegue es manual, para automatizar futuros despliegues lo hacemos a través de un webhook configurado en el .yaml del github action.

Pre-Deploy Command

Optional

\$

Edit

Render runs this command before the start command. Useful for database migrations and static asset uploads.

Deploy hook

.....

Regenerate hook

Your private URL to trigger a deploy for this server. Remember to keep this a secret.

- Configuramos un secret en el repo de github con la url proporcionada por RENDER y creamos el job en el .yaml

```

57 |   deploy:
58 |     needs: build-and-push
59 |     runs-on: ubuntu-latest
60 |     if: github.event_name != 'pull_request'
61 |     steps:
62 |       - name: Deploy to Render
63 |         run: curl ${ secrets.RENDER_DEPLOY_HOOK }
  
```

deploy

succeeded 6 minutes ago in 2s

Search logs

Refresh

Settings

Set up job

0s

Deploy to Render

2s

Run curl \*\*\*

% Total	% Received	Xferd	Average Speed	Time	Time	Time	Current
			Dload	Upload	Total	Spent	Left
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0:00:01	0
100	44	100	44	0	31	0:00:01	31

Complete job

0s