



## Render

### Cómo hacer un deploy

#### Objetivo

Siguiendo estos pasos vamos a poder desplegar **nuestro proyecto de Mercado Libre** a [Render](#) para que todos lo puedan visualizar en la nube.



#### Antes de empezar...

Antes de hacer cualquier cosa, asegurémonos de tener listos estos **4 puntos**:

1. Un proyecto subido a un repositorio de [GitHub](#).
2. **Adecuar la configuración** del archivo **Package.json**, para poder ejecutar nuestro proyecto desde la nube.
3. **Puertos y variables de entornos** configurados.\*

## \* Configuración del archivo package.json

En la sección scripts del archivo package.json:

```
"scripts": {  
  
  "test": "echo \"Error: no test specified\" && exit 1"  
  
},
```

En base a cómo queremos ejecutar nuestro proyecto, haremos lo siguiente:

1. **Si deseamos ejecutar nuestro proyecto de manera local**, podemos efectuar el siguiente cambio en la propiedad **"test"**, señalando la ruta y el nombre de nuestro archivo definido como Punto de Entrada (**Entry Point**), tal como se muestra en la siguiente imagen:

```
"scripts": {  
  
  "test": "nodemon src/index.js"  
  
}
```

Al querer ejecutar nuestro proyecto, en la consola solo debemos colocar el comando **npm test** y se activa el servidor local para la ejecución del mismo.

2. **Para poder ejecutar nuestro proyecto desde la nube**, es necesario agregar la propiedad **"start"** con el valor **"node src/index.js"** o **"node src/app.js"**, dependiendo del nombre de tu archivo de Punto de Entrada (**Entry Point**), tal como lo puedes observar en la siguiente imagen:

```
"scripts": {  
  
  "test": "nodemon src/index.js",  
  
  "start": "node src/index.js"  
  
},
```

## \* Configuración del puerto

Cuando iniciamos nuestro proyecto instalando el framework **Express** a través del comando **npm install express** debemos hacer un pequeño cambio dentro de la configuración del puerto donde se va a ejecutar nuestro proyecto.

Normalmente utilizamos el puerto **3000 o 3001 o el 3030**, pero [Render](#) establece otro diferente. En ese caso, vamos a cambiar el número de puerto dentro del **listen** a la siguiente línea:

**// CAMBIAR ESTE BLOQUE:**

```
app.listen(3001, function() {  
  console.log(`Servidor corriendo en el puerto 3001`);  
})
```

**// POR ESTE BLOQUE**

```
//Levantar nuestro servidor  
const port = process.env.PORT || 3001;  
app.listen(port, ()=>console.log(`Servidor corriendo en el puerto  
${port}`));
```

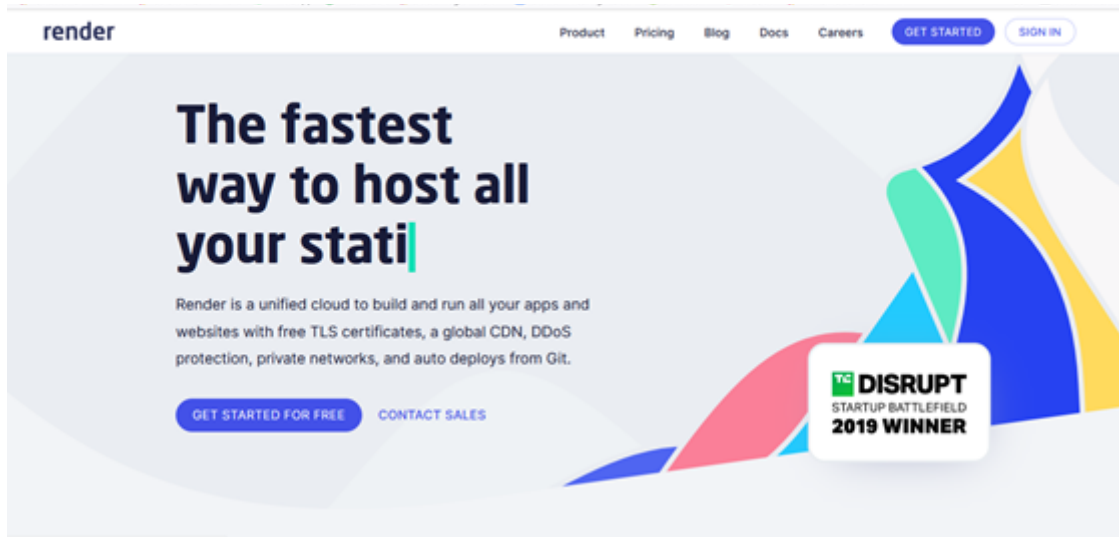
De esta forma realizamos una configuración dinámica usando [variables de entorno](#), útiles para guardar la información que necesitamos. Como se puede observar en la variable de entorno llamada PORT, estará guardado el puerto que utilice [Render](#) y en el caso contrario se utilizará nuestro puerto 3001.

## Pasos para desplegar nuestro proyecto:

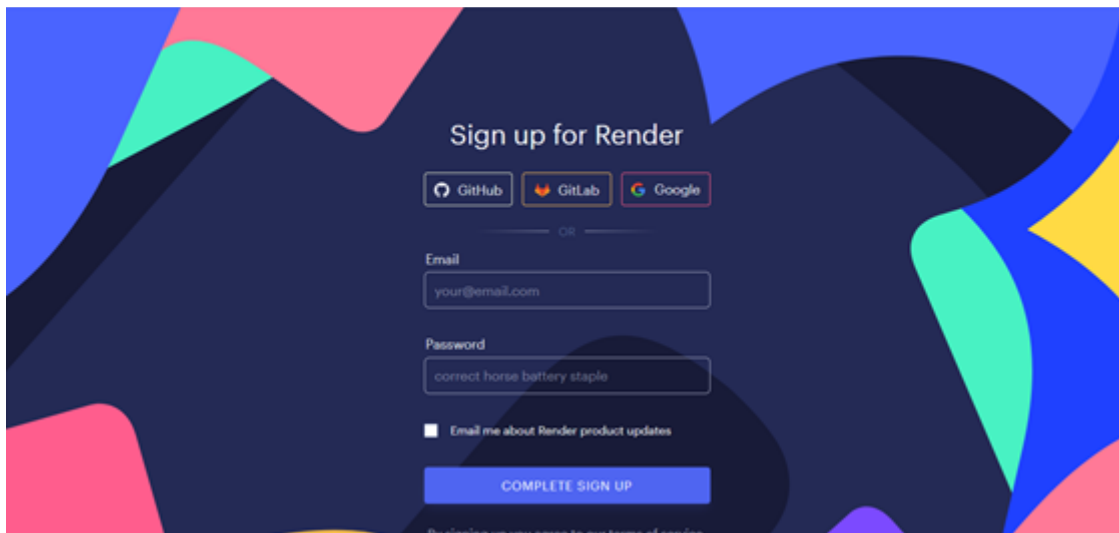


### Paso 1

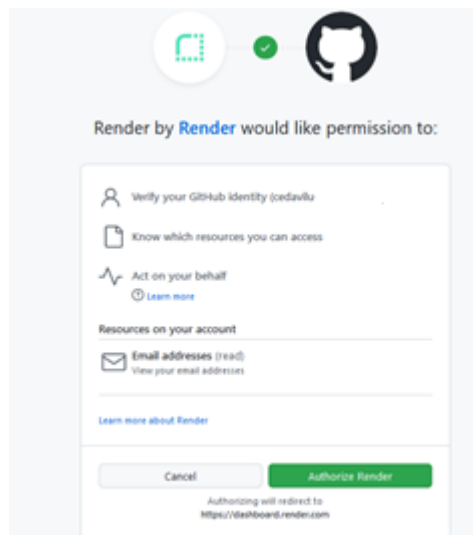
Ingresa a la plataforma [Render](https://render.com/) (<https://render.com/>)



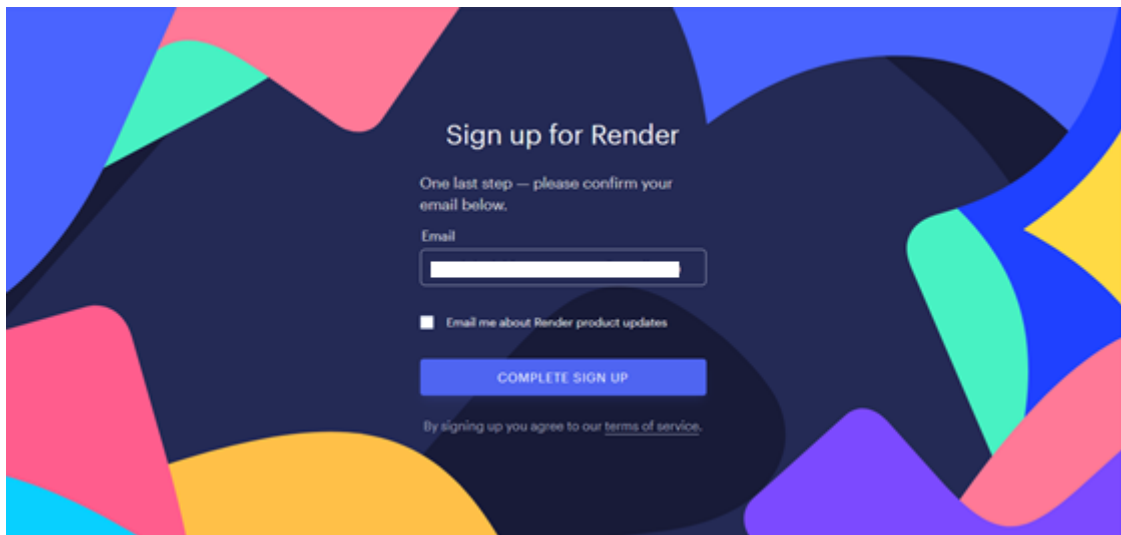
Una vez allí damos click sobre el botón **GET STARTED FOR FREE**. Te aparecerá la siguiente pantalla:



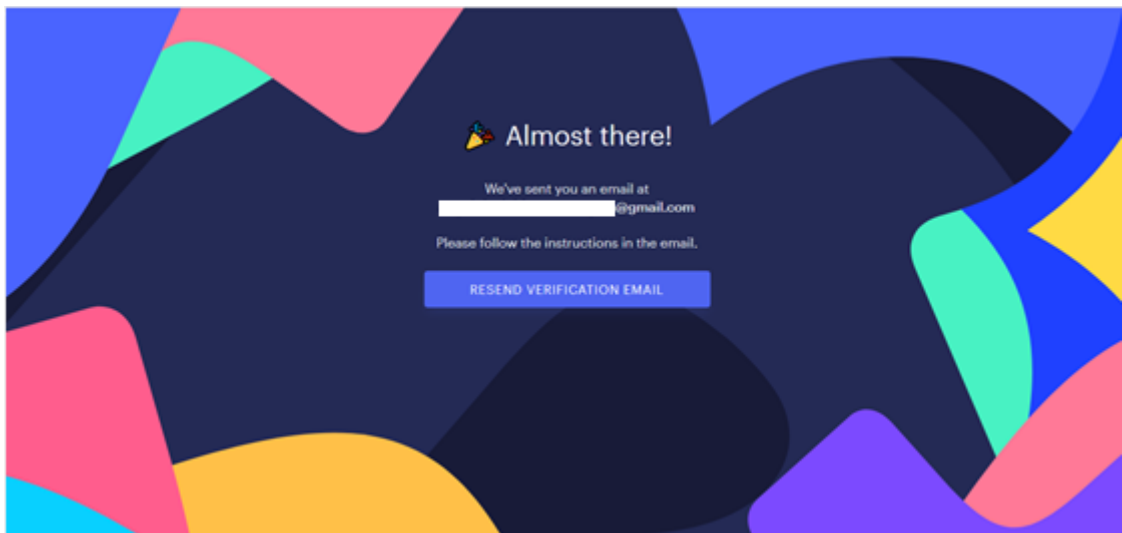
En esta pantalla, deberás accionar click sobre el botón **GitHub** para crear tu cuenta en Render.



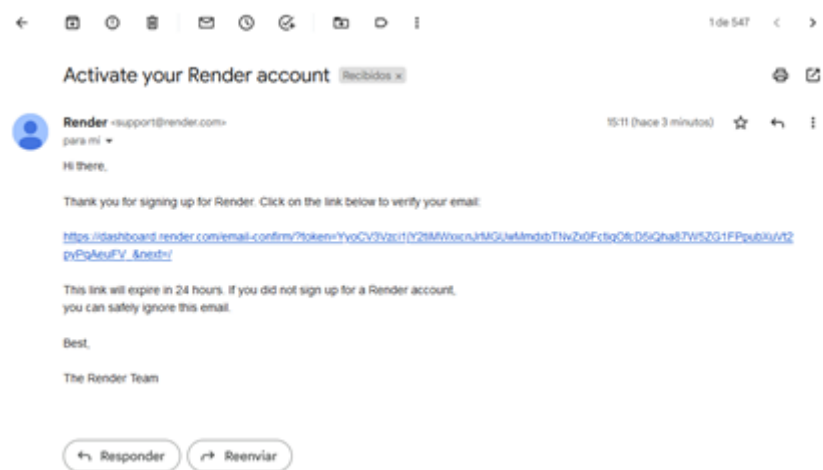
Aquí es necesario que des permiso a **Render** para que se comuniquen con tu repositorio en **GitHub**, a través del botón **Authorize Render**. Luego, te aparecerá la siguiente pantalla:



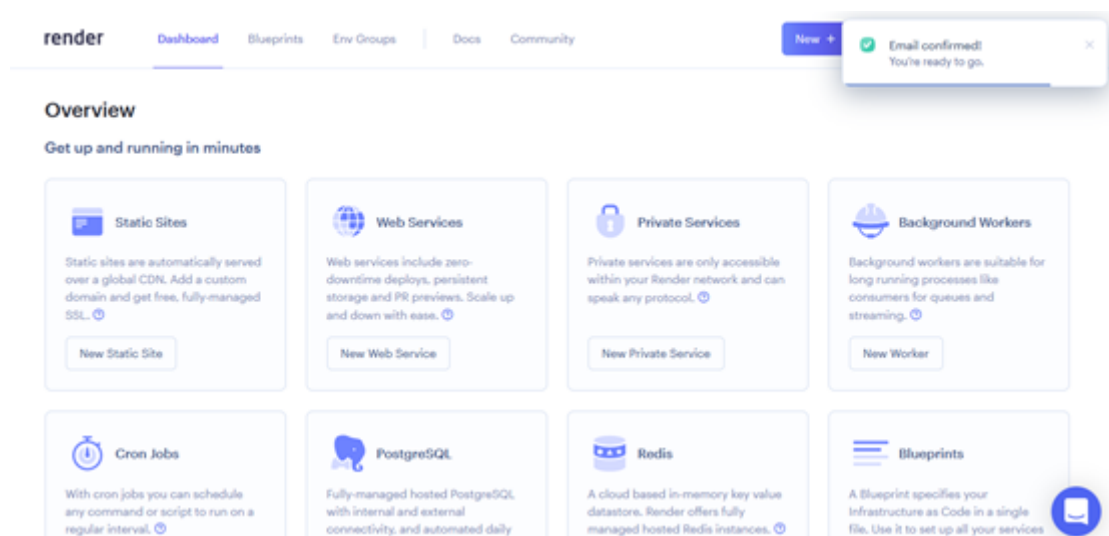
Aquí, confirma tu correo electrónico y da click al botón **COMPLETE SIGN UP**. Te enviarán un mail para confirmar tu cuenta.



A continuación, ingresa a tu correo electrónico y clickea el enlace que te enviaron para activar tu cuenta en **Render**.



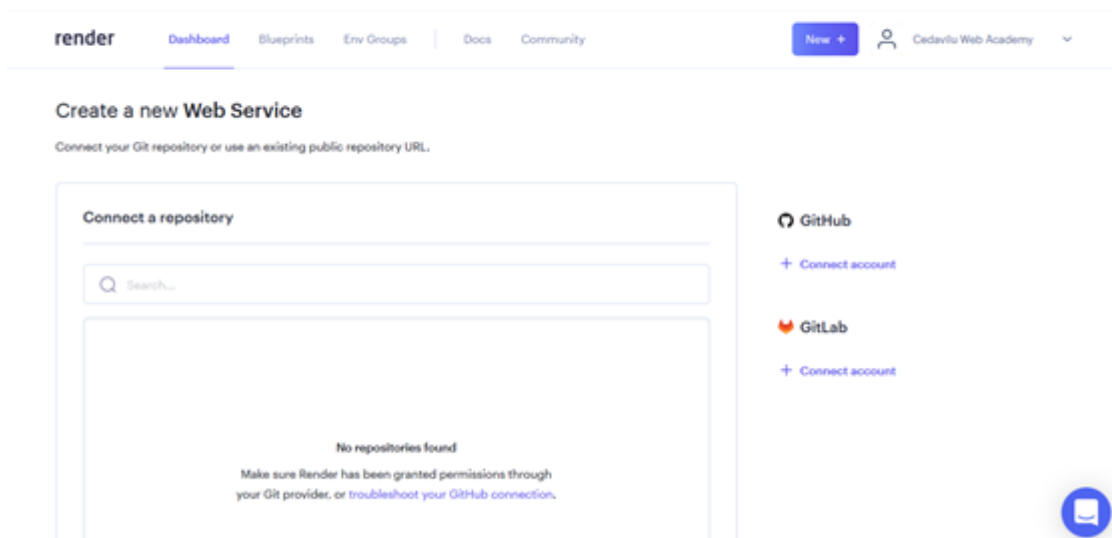
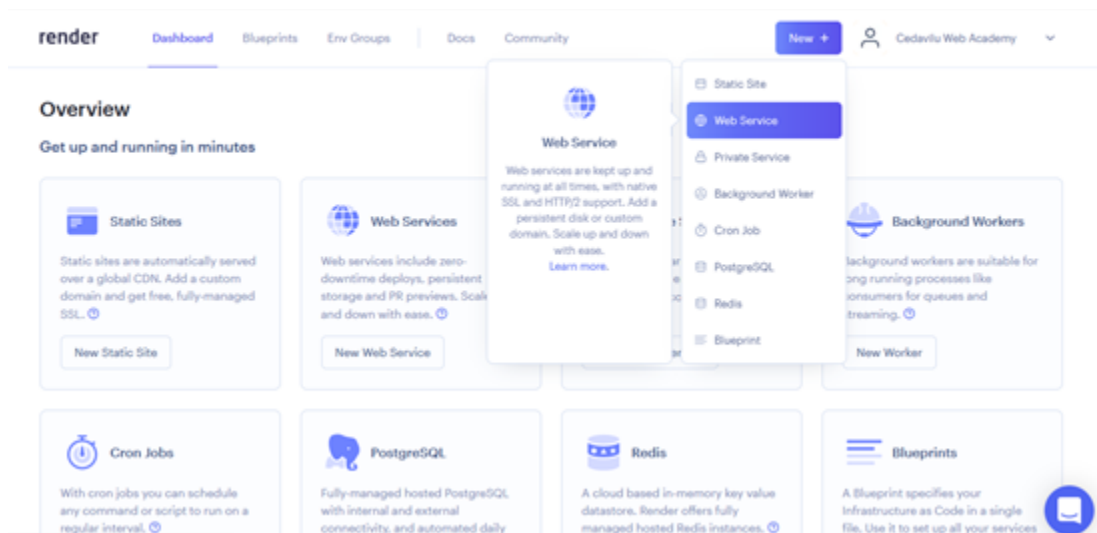
Serás redireccionado a la plataforma Render, con la cuenta activada:





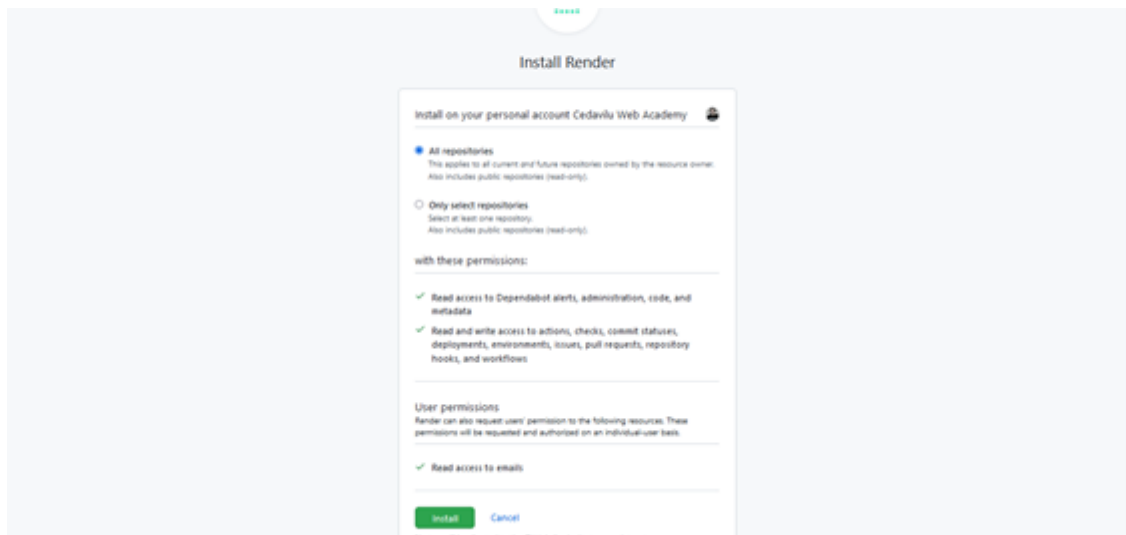
## Paso 2

Una vez activada nuestra cuenta, damos click sobre el botón **New +** y luego sobre la opción **Web Services**.

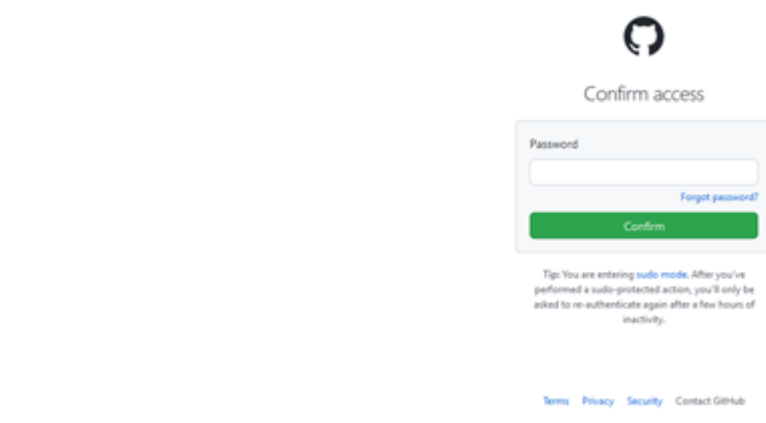


Aquí tenemos la posibilidad de conectarnos con nuestra cuenta de Github, clickeando en la opción **+ Connect Account** que aparece debajo del ícono de Github.

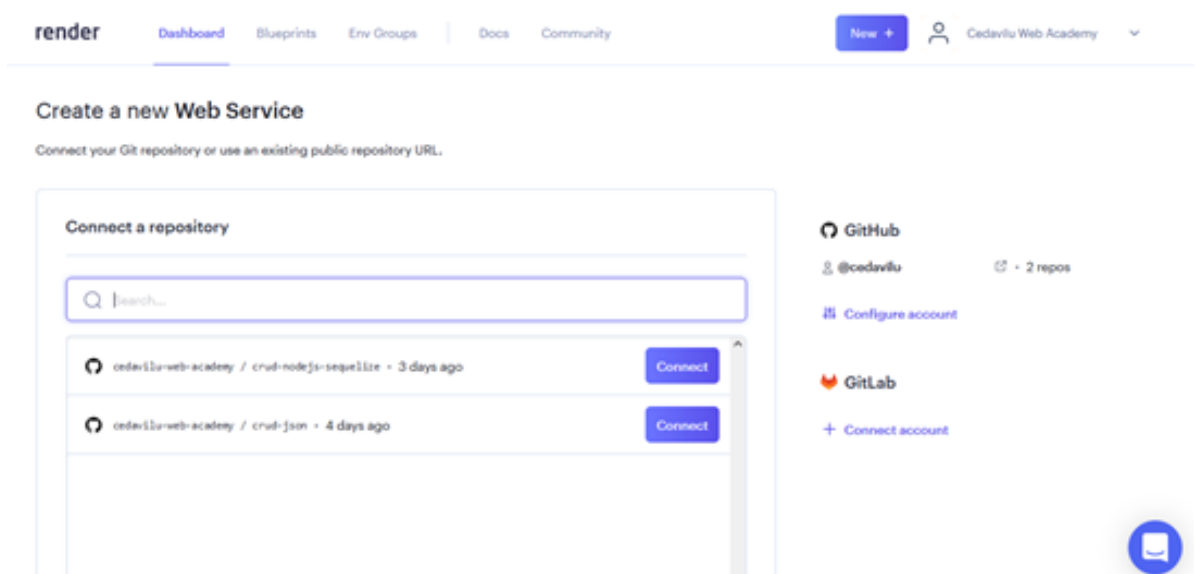
Podrás elegir entre dos opciones: que Render pueda acceder a todos tus repositorios de Github o seleccionar un único repositorio.



Una vez aquí hay que dar click sobre el botón **Install**. Luego, como se ve en la captura de abajo, confirmar nuestra contraseña de Github y clicar **Confirm**.



Hecho esto, podrás observar los datos de tus repositorios de Github desde Render:







## Paso 3

Una vez conectada nuestra cuenta de Github, seleccionar el botón **Connect** para efectivamente crear nuestro servicio web y desplegar nuestro proyecto en Render.

Tendremos que completar algunos datos:

The screenshot shows the Render dashboard with the 'New' button highlighted. Below the navigation bar, it states 'You are deploying a web service for cedavilu/crud-nodejs-sequelize.' The form contains four fields: 'Name' (required, with a red border and 'Required' error message), 'Root Directory' (optional, with a hint 'e.g., src'), 'Environment' (set to 'Node'), and 'Region' (set to 'Oregon (US West)'). A blue chat bubble icon is visible in the bottom right corner of the form area.

- **Name:** el nombre que queremos darle a nuestro servicio. Recuerda crear el nombre sin usar caracteres especiales o espacios en blanco. Puedes usar el guión del medio, **por ejemplo: "crud-nodejs-sequelize"**.
- **Root directory:** **No es necesario completar este campo.** Render considera como directorio raíz la misma donde se encuentra su proyecto en GitHub.
- **Environment:** **Node.**
- **Región:** La región en la que va a correr nuestro servicio. **Puede dejar la que Render le dispone por defecto.**

Cuando ya esté completa esta información, vamos a especificar la rama con la que vamos a desplegar nuestro proyecto.

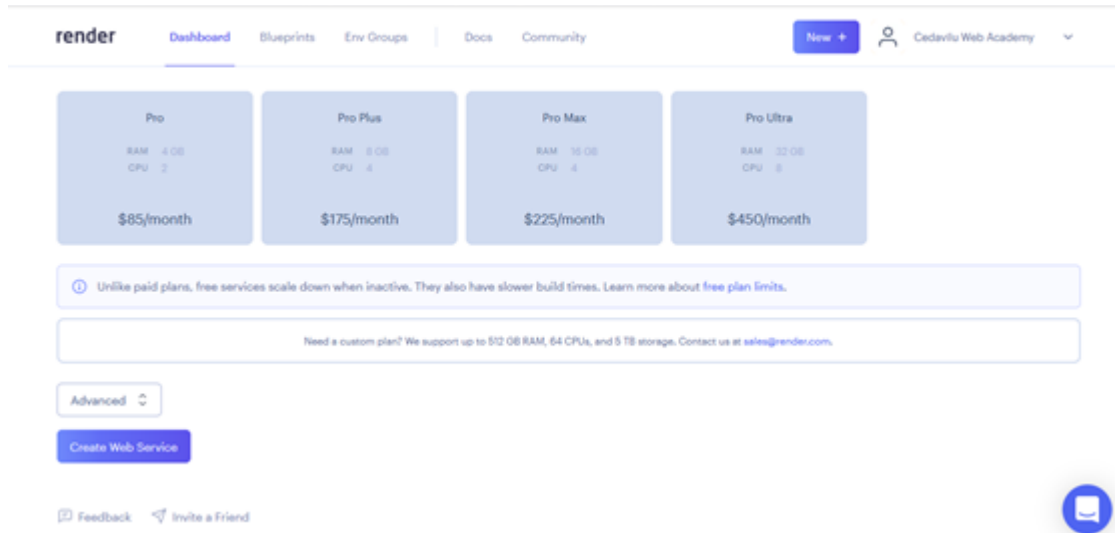
The screenshot shows the Render dashboard with the following configuration fields:

- Branch:** A dropdown menu with "master" selected. Description: "The repository branch used for your web service."
- Build Command:** A text input field containing "\$ npm install". Description: "This command runs in the root directory of your repository when a new version of your code is pushed, or when you deploy manually. It is typically a script that installs libraries, runs migrations, or compiles resources needed by your app."
- Start Command:** A text input field containing "\$ npm start". Description: "This command runs in the root directory of your app and is responsible for starting its processes. It is typically used to start a webserver for your app. It can access environment variables defined by you in Render."
- Plans:** A section with a message: "Please enter your payment information to select a plan with higher limits." and a button with a mail icon.

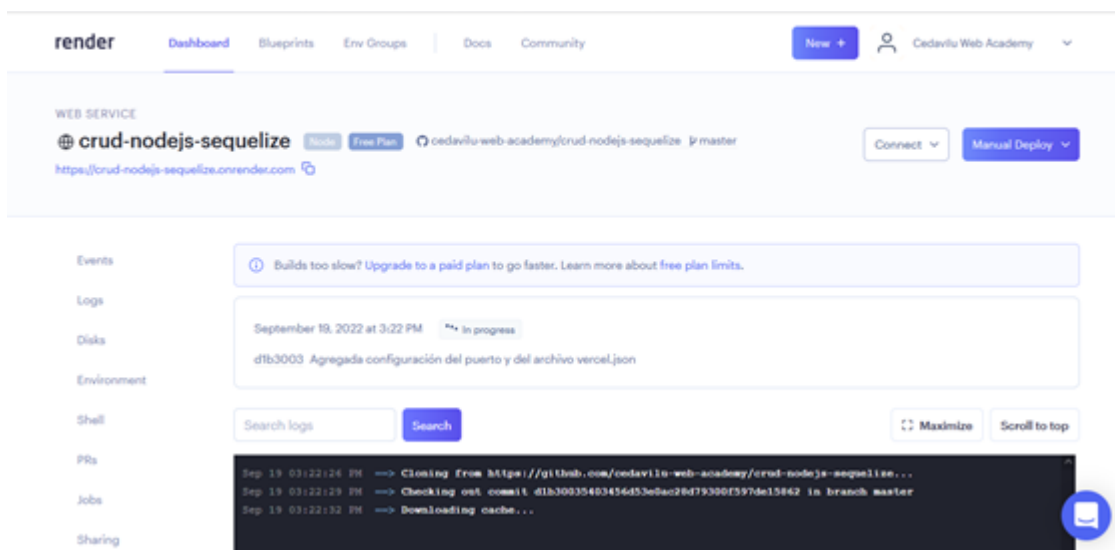
- **Branch:** "master"
- **Build Command:** Comando utilizado para efectuar la instalación de conjunto de dependencias que se encuentran en nuestro archivo package.json: "npm install"
- **Start Command:** Comando que utilizamos para ejecutar nuestro proyecto, "npm start". Si por algún motivo en tu archivo package.json, no tienes configurado el script de carga del proyecto, entonces podrás colocar: "node scr/index.js" o "node src/app.js". Dependerá del nombre de su archivo de punto de entrada creado (**Entry Point**), tal como se puede observar en la imagen debajo.

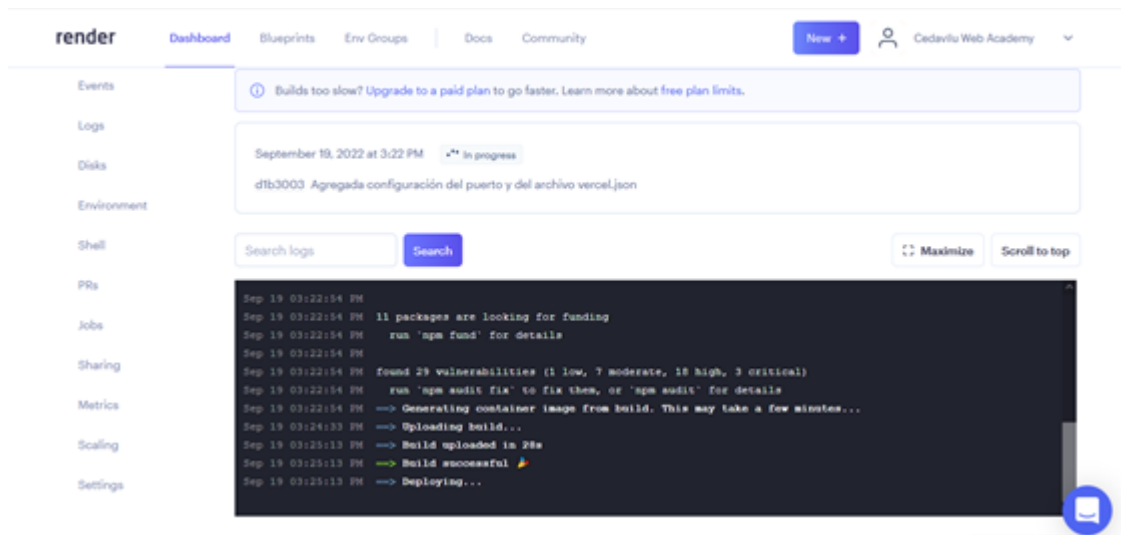
```
"scripts": {
  "test": "nodemon src/index.js",
  "start": "node src/index.js"
},
```

Finalmente, daremos click sobre el botón **Create Web Services**:

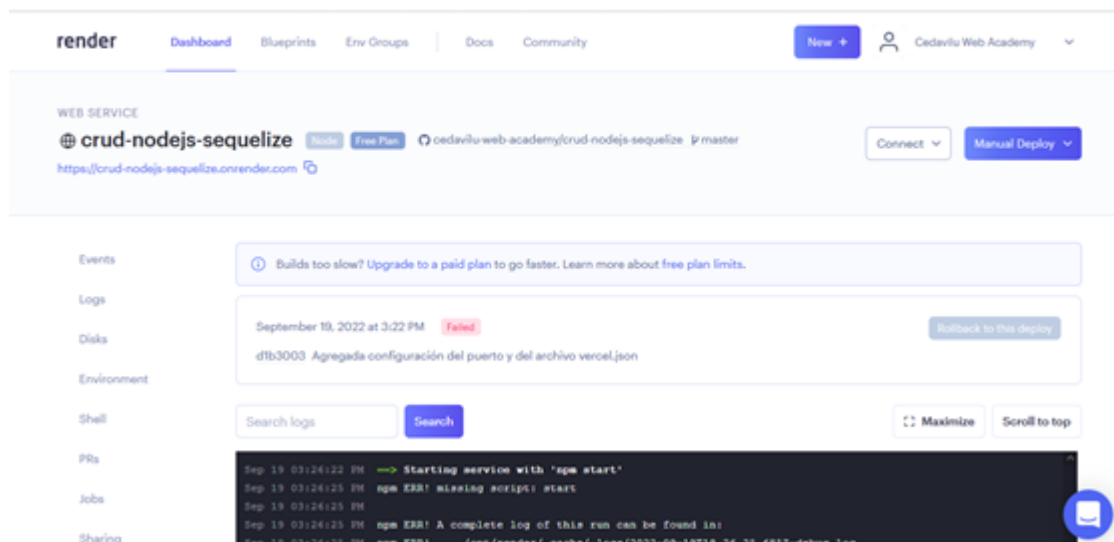


**Render** iniciará el despliegue del repositorio seleccionado de nuestra cuenta de GitHub, tal como se puede observar en las siguientes imágenes:



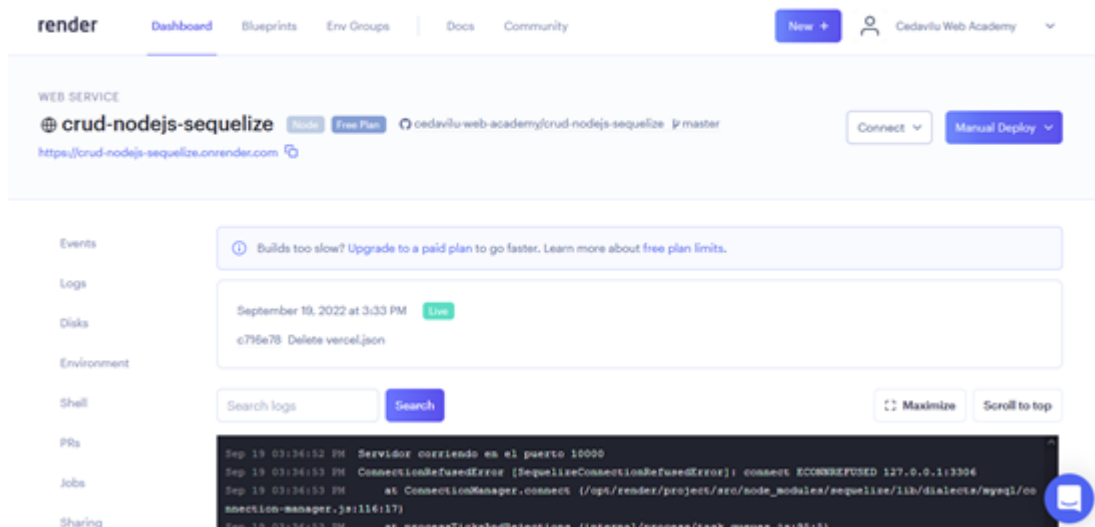


Una vez que finaliza el despliegue del proyecto, nos aparecerá una pantalla donde podemos observar si hubo errores en el proceso.



Si nos aparece algún error, como en la imagen debemos realizar los ajustes que sean necesarios en nuestro proyecto. Una vez hechos, debemos dar click sobre el botón que aparece en la parte superior derecha, **Manual Deploy**. Esto podrá hacerse tantas veces como sea necesario en base a los ajustes que realice sobre su proyecto en GitHub.

De no encontrarse ningún error al momento de desplegar nuestro proyecto, nos aparece la siguiente pantalla:

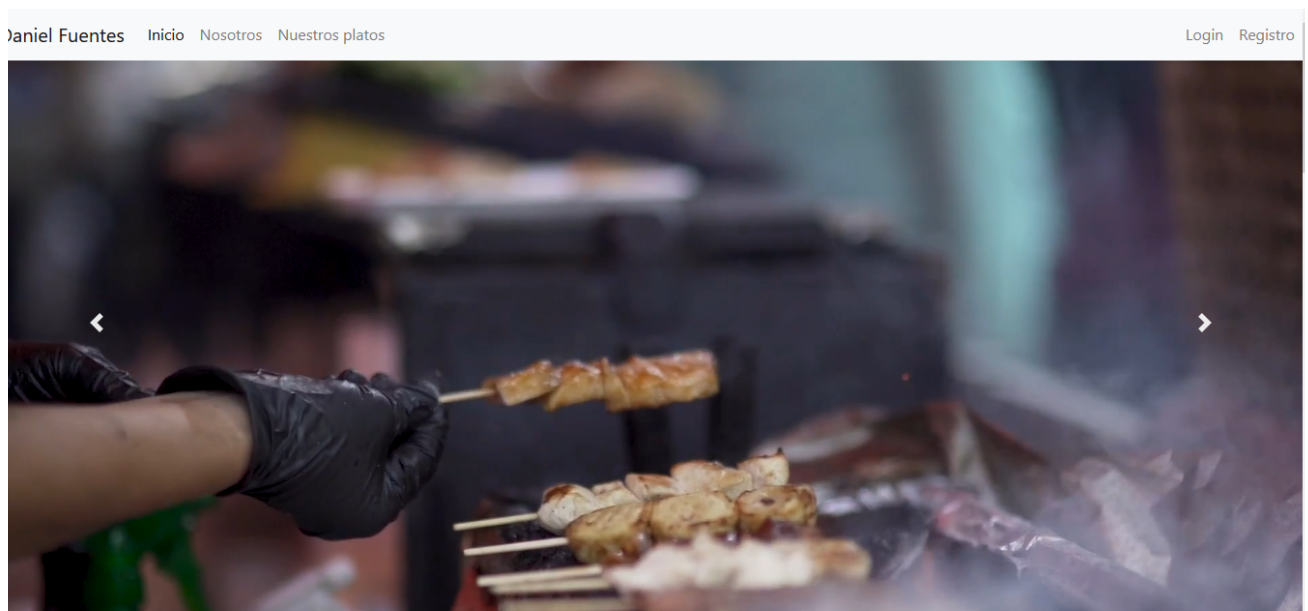


En el centro de la pantalla vemos el mensaje **Live** que nos indica que el proyecto fue desplegado correctamente.

En la parte superior izquierda de la pantalla, se podrá observar el localizador universal de recurso de nuestro servicio web desplegado en Render. En el caso de esta guía vemos:

<https://crud-nodejs-sequelize.onrender.com/>

Si hacemos click en ella, accedemos a nuestro proyecto en la nube.



**¡Happy Deploying!**