

Despliegue a producción con Railway.app

Julian F. Latorre

13 de septiembre de 2024

Índice

1. Introducción	2
2. Preparación del Proyecto	2
2.1. Estructura del Proyecto	2
2.2. Configuración del Root package.json	2
2.3. Configuración del Procfile	2
3. Preparación del Frontend (React + Tailwind)	3
3.1. Optimización de la Construcción	3
3.2. Configuración de Entorno	3
4. Preparación del Backend (Node.js + Express)	3
4.1. Configuración del Servidor	3
5. Configuración de las Bases de Datos	4
5.1. MongoDB	4
5.2. MySQL	4
6. Despliegue en Railway	4
6.1. Preparación para el Despliegue	4
6.2. Proceso de Despliegue	5
6.3. Configuración de Variables de Entorno	5
6.4. Verificación Post-Despliegue	5
7. Configuración de Dominio Personalizado (Opcional)	5
8. Mantenimiento y Actualizaciones	6
9. Escalado y Optimización	6
10. Backups y Seguridad	6

1. Introducción

Esta guía proporciona instrucciones detalladas paso a paso para desplegar una aplicación fullstack desarrollada con React, Tailwind CSS, Node.js, Express, MongoDB y MySQL en Railway. La guía asume que ya tienes tu aplicación desarrollada y funcionando localmente.

2. Preparación del Proyecto

2.1. Estructura del Proyecto

Asegúrate de que tu proyecto tenga una estructura similar a esta:

```
my-fullstack-app/  
  client/           # Frontend React  
    src/  
    package.json  
    ...  
  server/          # Backend Node.js/Express  
    src/  
    package.json  
    ...  
  package.json      # Root package.json  
  Procfile          # Para Railway  
  .gitignore
```

2.2. Configuración del Root package.json

En el directorio raíz del proyecto, asegúrate de que tu `package.json` incluya scripts para instalar dependencias, construir y ejecutar tanto el frontend como el backend:

```
{  
  "scripts": {  
    "install:client": "cd client && npm install",  
    "install:server": "cd server && npm install",  
    "build:client": "cd client && npm run build",  
    "build": "npm run install:client && npm run install:server &&  
      npm run build:client",  
    "start": "cd server && npm start"  
  }  
}
```

2.3. Configuración del Procfile

Crea un archivo `Procfile` en la raíz del proyecto con el siguiente contenido:

```
web: npm start
```

Este archivo le indica a Railway cómo iniciar tu aplicación.

3. Preparación del Frontend (React + Tailwind)

3.1. Optimización de la Construcción

En el directorio `client`, asegúrate de que tu `package.json` tenga un script de construcción:

```
{
  "scripts": {
    "build": "react-scripts build"
  }
}
```

3.2. Configuración de Entorno

Crea un archivo `.env.production` en el directorio `client` para las variables de entorno de producción:

```
REACT_APP_API_URL=${RAILWAY_STATIC_URL}/api
```

Railway proporcionará la variable `RAILWAY_STATIC_URL` automáticamente.

4. Preparación del Backend (Node.js + Express)

4.1. Configuración del Servidor

En tu archivo principal del servidor (por ejemplo, `server/src/index.js`), asegúrate de que el servidor escuche en el puerto proporcionado por Railway:

```
const express = require('express');
const path = require('path');
const app = express();
const port = process.env.PORT || 3000;

// Sirve los archivos estáticos del frontend
app.use(express.static(path.join(__dirname, '../..client/build')));

// Tus rutas API aquí
app.get('/api/hello', (req, res) => {
  res.json({ message: 'Hello from server!' });
});

// Maneja las rutas del SPA
app.get('*', (req, res) => {
  res.sendFile(path.join(__dirname, '../..client/build/index.html'));
});

app.listen(port, () => {
  console.log(`Server running on port ${port}`);
});
```

5. Configuración de las Bases de Datos

5.1. MongoDB

Railway ofrece una integración directa con MongoDB. Sigue estos pasos:

1. En el dashboard de Railway, ve a tu proyecto.
2. Haz clic en "New" selecciona "MongoDB".
3. Railway configurará automáticamente una instancia de MongoDB y proporcionará las variables de entorno necesarias.
4. En tu backend, usa la variable de entorno `MONGODB_URL` para la conexión:

```
const mongoose = require('mongoose');
mongoose.connect(process.env.MONGODB_URL, {
  useNewUrlParser: true,
  useUnifiedTopology: true
});
```

5.2. MySQL

Railway también ofrece una integración con MySQL. Sigue estos pasos:

1. En el dashboard de Railway, ve a tu proyecto.
2. Haz clic en "New" selecciona "MySQL".
3. Railway configurará automáticamente una instancia de MySQL y proporcionará las variables de entorno necesarias.
4. En tu backend, usa las variables de entorno proporcionadas por Railway para la conexión:

```
const mysql = require('mysql2');
const connection = mysql.createConnection({
  host: process.env.MYSQLHOST,
  user: process.env.MYSQLUSER,
  password: process.env.MYSQLPASSWORD,
  database: process.env.MYSQLDATABASE,
  port: process.env.MYSQLPORT
});
```

6. Despliegue en Railway

6.1. Preparación para el Despliegue

1. Asegúrate de que todo tu código esté "commitado" y "pushado" a un repositorio de GitHub.
2. Crea una cuenta en Railway si aún no tienes una (<https://railway.app>).

6.2. Proceso de Despliegue

1. Inicia sesión en tu cuenta de Railway.
2. Haz clic en "New Project".
3. Selecciona "Deploy from GitHub repo".
4. Elige el repositorio que contiene tu aplicación.
5. En la configuración del proyecto:
 - Root Directory: Deja en blanco para usar la raíz del repositorio.
 - Build Command: Usa `npm run build`.
 - Start Command: Usa `npm start`.
6. Railway detectará automáticamente que es una aplicación Node.js y configurará el entorno adecuadamente.
7. Haz clic en "Deploy" para iniciar el despliegue.

6.3. Configuración de Variables de Entorno

1. En el dashboard de tu proyecto en Railway, ve a la pestaña "Variables".
2. Aquí verás las variables de entorno para tus bases de datos MongoDB y MySQL.
3. Añade cualquier variable de entorno adicional que tu aplicación necesite.

6.4. Verificación Post-Despliegue

1. Una vez completado el despliegue, Railway proporcionará una URL para tu aplicación.
2. Visita la URL y verifica que tanto el frontend como el backend funcionen correctamente.
3. Revisa los logs en el dashboard de Railway para detectar cualquier error.

7. Configuración de Dominio Personalizado (Opcional)

1. En el dashboard de Railway, ve a la pestaña "Settings" de tu proyecto.
2. En la sección "Domains", haz clic en "Generate Domain." o "Add Custom Domain".
3. Si eliges un dominio personalizado, sigue las instrucciones de Railway para configurar los registros DNS de tu dominio.

8. Mantenimiento y Actualizaciones

- Railway configurará automáticamente despliegues continuos desde tu repositorio de GitHub.
- Cada vez que hagas push a la rama principal, Railway iniciará un nuevo despliegue.
- Utiliza el sistema de monitoreo integrado de Railway para supervisar el rendimiento de tu aplicación.
- Revisa regularmente los logs de tu aplicación en el dashboard de Railway.

9. Escalado y Optimización

- Railway ofrece opciones de escalado vertical y horizontal.
- Para escalar verticalmente, puedes aumentar los recursos (CPU, RAM) asignados a tu aplicación desde el dashboard.
- Para escalar horizontalmente, puedes configurar múltiples instancias de tu aplicación.
- Utiliza la característica de "Metrics" de Railway para identificar cuellos de botella y optimizar el rendimiento.

10. Backups y Seguridad

- Railway realiza backups automáticos de tus bases de datos.
- Configura backups adicionales manualmente si es necesario.
- Revisa y actualiza regularmente las dependencias de tu proyecto para mantener la seguridad.
- Utiliza variables de entorno para todas las configuraciones sensibles y nunca las incluyas en el control de versiones.