Despliegue a Producción

Tomás Eduardo Romero Cañas

INTRODUCCIÓN

El objetivo de este documento es detallar los pasos necesarios para desplegar la aplicación desarrollada en un servidor de producción. El despliegue se realiza utilizando tecnologías como Spring Boot para el backend, React para el frontend y MySQL como base de datos. La guía considera la configuración de un servidor Linux, pero puede ser adaptada a otros entornos.

Pasos para el Despliegue

Requisitos Previos

1. ***Servidor:***
    - *Acceso a un servidor Linux (ejemplo: Ubuntu 22.04) con permisos de administrador.*
    - *Puerto abierto para la aplicación web (por defecto 8080) y para el frontend (por defecto 3000).*
2. ***Software:***
    - *Java JDK 17+ instalado en el servidor.*
    - *Node.js 18+ instalado para construir la aplicación frontend.*
    - *MySQL 8.1+ configurado y corriendo.*
    - *Herramienta de empaquetado (Maven) instalada en el servidor.*
3. ***Otros:***
    - *Un dominio y certificado SSL (opcional, pero recomendado para producción).*
    - *Docker (opcional para contenerización).*

Configuración del Entorno

1. **Instalar Java:**

   sudo apt update
   sudo apt install openjdk-17-jdk -y
   java -version

2. **Instalar Node.js y npm:**

   curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash –
   sudo apt install -y nodejs
   node -v
   npm -v

3. **Instalar MySQL:**

   sudo apt update
   sudo apt install mysql-server -y
   sudo mysql_secure_installation

4. **Configurar la base de datos:**

   sudo mysql -u root -p

   Crear la base de datos y usuario:

   ```
   CREATE DATABASE productos_db;
   CREATE USER 'appuser'@'%' IDENTIFIED BY 'securepassword';
   GRANT ALL PRIVILEGES ON productos_db.* TO 'appuser'@'%';
   FLUSH PRIVILEGES;
   EXIT;
   ```

5. **Configurar el firewall:**

   sudo ufw allow 8080/tcp
   sudo ufw allow 3000/tcp
   sudo ufw enable

Despliegue del Backend

1. **Clonar el repositorio en el servidor:**

   git clone <repositorio-url>
   cd <directorio-del-proyecto>

2. **Actualizar las credenciales en application.properties:**

   spring.datasource.url=jdbc:mysql://localhost:3306/productos_db
   spring.datasource.username=appuser
   spring.datasource.password=securepassword
   server.port=8080

3. **Construir y empaquetar la aplicación:**

   cd backend
   mvn clean package

4. **Ejecutar el backend:**
   - Sin Docker:

   java -jar target/<nombre-del-jar>.jar

   - Con Docker:

     - Crear un archivo Dockerfile en el directorio del backend:

   FROM openjdk:17-jdk-slim
   VOLUME /tmp
   ADD target/<nombre-del-jar>.jar app.jar
   ENTRYPOINT ["java", "-jar", "/app.jar"]

     - Construir y ejecutar el contenedor:

   docker build -t backend-app .
   docker run -d -p 8080:8080 backend-app

Despliegue del Frontend

1. **Construir la aplicación React:**

   cd frontend
   npm install
   npm run build

2. **Configurar un servidor para archivos estáticos:**

   - Instalar Nginx:

   sudo apt install nginx -y

   - Configurar Nginx para servir los archivos:

   sudo nano /etc/nginx/sites-available/default

   - Reemplazar el contenido por:

```
server {
    listen 80;

    server_name yourdomain.com;

    location / {
        root /path/to/frontend/build;
        index index.html;
        try_files $uri /index.html;
    }
}
```

   - Reiniciar Nginx:

   sudo systemctl restart nginx

3. **Opcional: Configurar HTTPS con Certbot:**

   - Instalar Certbot:

   sudo apt install certbot python3-certbot-nginx -y

   - Generar y configurar certificados SSL:

   sudo certbot --nginx -d yourdomain.com

Verificación

1. **Pruebas:**
   - Acceder a http://<tu-servidor>:8080 para probar el backend.
   - Acceder a http://<tu-dominio> para probar el frontend.
2. **Logs:**
   - Verificar los logs del backend:

```
tail -f logs/app.log
```

   - Verificar los logs de Nginx:

```
sudo tail -f /var/log/nginx/access.log
```