DESPLIEGUE A PRODUCCIÓN

Miguel Antonio Amaya Hernández

Prueba técnica NOVIEMBRE,2024

Software Libre:

Preparación del Servidor de Producción

Requerimientos

- **Sistema Operativo**: Ubuntu 20.04/22.04 LTS.
- **Recursos mínimos recomendados**:
  - 2 CPU.
  - 4GB de RAM.
  - 20GB de almacenamiento.
- **Software necesario**:
  - **Java JDK**: Versión 11 o superior.
  - **Node.js y npm**: Para construir y servir el frontend.
  - **MySQL**: Base de datos para el backend.

Pasos para preparar el servidor

1. **Actualizar el servidor**:

```
sudo apt update && sudo apt upgrade -y
```

2. **Instalar Java JDK**:

```
sudo apt install openjdk-11-jdk -y
java -version
```

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

```
curl -fsSL https://deb.nodesource.com/setup_16.x | sudo -E bash -
sudo apt install -y nodejs
node -v
npm -v
```

4. **Instalar MySQL**:

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

Configurar la base de datos:

- ✓ Accede a MySQL:

```
sudo mysql -u root -p
```

- ✓ Crea la base de datos y usuario:

```
CREATE DATABASE gestion_productos;
CREATE USER 'admin'@'%' IDENTIFIED BY 'password';
GRANT ALL PRIVILEGES ON gestion_productos.* TO 'admin'@'%';
FLUSH PRIVILEGES;
EXIT;
```

2. Despliegue del Backend (Spring Boot)

Generar el archivo JAR

Desde tu proyecto de Spring Boot:

```
mvn clean package
```

Esto generará un archivo gestion-productos-0.0.1-SNAPSHOT.jar en la carpeta target.

Configurar el backend en el servidor

1. **Copiar el archivo JAR al servidor**: Usa scp para copiar el archivo:

```
scp target/gestion-productos-0.0.1-SNAPSHOT.jar
usuario@servidor:/home/usuario/
```

2. **Configurar el servicio con systemd**: Crea un archivo de servicio para gestionar el backend:

```
sudo nano /etc/systemd/system/gestion-productos.service
```

Contenido:

```
[Unit]
Description=Aplicación Spring Boot
After=syslog.target
[Service]
User=usuario
ExecStart=/usr/bin/java -jar /home/usuario/gestion-productos-0.0.1-
SNAPSHOT.jar
Restart=always
[Install]
WantedBy=multi-user.target
```

3. **Iniciar el servicio**:

```
sudo systemctl daemon-reload
sudo systemctl start gestion-productos
sudo systemctl enable gestion-productos
```

4. **Configurar el archivo application.properties para producción**: Ajusta las credenciales y URL de la base de datos:

```
spring.datasource.url=jdbc:mysql://localhost:3306/gestion_productos
spring.datasource.username=admin
spring.datasource.password=password
```

3. Despliegue del Frontend (React)

Construir el frontend

Desde la carpeta de tu proyecto React:

```
npm run build
```

Esto generará una carpeta build/ con los archivos estáticos listos para producción.

Configurar Nginx para servir el frontend

1. **Instalar Nginx**:
   `sudo apt install nginx -y`
2. **Copiar los archivos del build al servidor**:
   `scp -r build/ usuario@servidor:/home/usuario/build`
3. **Configurar Nginx**:
   - Crea un nuevo bloque de servidor:
     sudo nano /etc/nginx/sites-available/gestion-productos
   - Contenido del archivo:

```
server {
    listen 80;
    server_name ejemplo.com;

    root /home/usuario/build;
    index index.html;

    location / {
        try_files $uri /index.html;
    }

    location /api/ {
        proxy_pass http://localhost:8080/;
        proxy_http_version 1.1;
        proxy_set_header Upgrade $http_upgrade;
        proxy_set_header Connection 'upgrade';
        proxy_set_header Host $host;
        proxy_cache_bypass $http_upgrade;
  }
}
```

4. **Habilitar el bloque de servidor**:

```
sudo ln -s /etc/nginx/sites-available/gestion-productos
/etc/nginx/sites-enabled/
sudo nginx -t
sudo systemctl restart nginx
```

4. Configurar un Dominio y HTTPS

Apuntar el dominio al servidor

- Configura los registros DNS de tu dominio para apuntar a la dirección IP de tu servidor.

Instalar Certbot para HTTPS:

1. **Instalar Certbot**:
   `sudo apt install certbot python3-certbot-nginx -y`
2. **Generar el certificado SSL**:
   `sudo certbot --nginx -d ejemplo.com -d www.ejemplo.com`
3. **Renovar automáticamente el certificado**:
   `sudo crontab -e`

Agrega:

```
0 0 * * * certbot renew --quiet
```

5. Pruebas y Monitoreo

Verificar que todo funciona correctamente

- Abre tu navegador y accede a tu dominio.
- Prueba las rutas de la API y asegúrate de que el frontend puede interactuar con el backend.

Instalar herramientas de monitoreo:

- **HTop** para monitorear el uso del sistema:
  ```
  sudo apt install htop -y
  htop
  ```
- **Logs de Spring Boot**:
  ```
  journalctl -u gestion-productos -f
  ```
- **Logs de Nginx**:
  ```
  sudo tail -f /var/log/nginx/access.log /var/log/nginx/error.log
  ```

Windows

1. Preparación del Entorno en Windows
Requisitos

1. **Java JDK**: Versión 11 o superior.
2. **Node.js y npm**: Para construir el frontend.
3. **MySQL**: Como base de datos del backend.
4. **Apache/Nginx** (opcional): Si deseas servir el frontend como estático y configurar un proxy inverso para el backend.

2. Instalación y Configuración
Java JDK

1. Descarga la última versión de Java JDK desde Oracle o OpenJDK.
2. Instálalo y configura la variable de entorno JAVA_HOME:
   - Ve a **Configuración del Sistema → Variables de Entorno**.
   - Agrega JAVA_HOME apuntando a la ruta donde instalaste el JDK.
   - Agrega %JAVA_HOME%\bin al PATH.

Verifica la instalación:
java -version
Node.js

1. Descarga Node.js desde Node.js Official.
2. Durante la instalación, habilita la opción de agregar Node.js al PATH.
3. Verifica la instalación:
   ```
   node -v
   npm -v
   ```

MySQL

1. Descarga MySQL desde MySQL Community Downloads.
2. Durante la instalación, selecciona la opción "Developer Default" y configura una contraseña para el usuario root.

Crea la base de datos:

- Abre MySQL Workbench o usa la línea de comandos:
  ```
  CREATE DATABASE gestion_productos;
  CREATE USER 'admin'@'localhost' IDENTIFIED BY 'password';
  GRANT ALL PRIVILEGES ON gestion_productos.* TO
  'admin'@'localhost';
  FLUSH PRIVILEGES;
  ```

3. Despliegue del Backend (Spring Boot)
Generar el archivo JAR

1. Ve a tu proyecto de Spring Boot y ejecuta:
   ```
   mvn clean package
   ```

Esto generará un archivo gestion-productos-0.0.1-SNAPSHOT.jar en la carpeta target.
Configurar el archivo de propiedades para producción

1. Abre application.properties o application.yml y ajusta:
   ```
   spring.datasource.url=jdbc:mysql://localhost:3306/gestion_productos
   spring.datasource.username=admin
   spring.datasource.password=password
   ```

Ejecutar el backend

1. Ejecuta el archivo JAR:
   ```
   java -jar target/gestion-productos-0.0.1-SNAPSHOT.jar
   ```
2. Asegúrate de que el backend esté corriendo en http://localhost:8080.

4. Despliegue del Frontend (React)

Construir el frontend

1. Ve a tu proyecto React y ejecuta:
   ```
   npm run build
   ```

Esto generará una carpeta build con los archivos estáticos.

Servir el frontend

Puedes usar un servidor HTTP simple o un servidor web como **IIS** (Internet Information Services).

Opción 1: Servidor HTTP simple

1. Instala el paquete serve:
   ```
   npm install -g serve
   Sirve los archivos:
   serve -s build
   ```

El frontend estará disponible en http://localhost:5000.

Opción 2: Usar IIS

1. Habilita IIS en tu sistema desde **Panel de Control → Activar o desactivar características de Windows**.
2. Copia la carpeta build a una ubicación como C:\inetpub\wwwroot.
3. Configura un sitio web en IIS apuntando a esa carpeta.

5. Configurar Proxy Inverso (opcional)

Para conectar React con Spring Boot y evitar problemas de CORS, puedes configurar un proxy en IIS o usar **http-proxy-middleware** en el frontend.

Usar http-proxy-middleware

1. Instala el paquete:
   ```
   npm install http-proxy-middleware --save
   ```
2. Crea un archivo src/setupProxy.js con el contenido:
   ```
   const { createProxyMiddleware } = require('http-proxy-middleware');

   module.exports = function (app) {
     app.use(
       '/api',
       createProxyMiddleware({
         target: 'http://localhost:8080',
         changeOrigin: true,
       })
     );
   };
   ```

6. Configurar HTTPS (opcional)
Si necesitas HTTPS, puedes usar herramientas como **mkcert** para generar certificados locales o configurar HTTPS en IIS.

7. Estructura del Proyecto en GitHub
Si deseas mantener el frontend y el backend en un mismo repositorio, organiza tu proyecto así:

```
/gestion-productos
  /frontend-react
  /backend-springboot
```

Cada carpeta contiene los archivos específicos del frontend y el backend. En el archivo README.md puedes agregar instrucciones detalladas para instalar y ejecutar ambas partes.

8. Pruebas Finales
Probar la integración

- Abre el navegador en http://localhost:5000 y verifica que puedas realizar operaciones como:
  - Iniciar sesión/registrarte.
  - Agregar, listar, buscar y eliminar productos.

Monitorear el backend
Si hay problemas con Spring Boot, revisa los logs:

```
java -jar target/gestion-productos-0.0.1-SNAPSHOT.jar
```