Configuración de CI/CD y Contenerización de una Aplicación

Explicación Paso a Paso

1. Contenerización de la Aplicación con Docker

¿Qué hicimos?

1. Creamos un archivo Dockerfile:

- Este archivo define cómo construir la imagen de Docker para tu aplicación.

- Dividimos el proceso en etapas para optimizar el tiempo de construcción:

* Descargar dependencias: Preparamos las librerías necesarias.

* Construir la aplicación: Compilamos el código fuente para generar un archivo .jar.

* Optimizar capas: Dividimos el .jar en capas para mejorar la caché.

* Ejecutar la aplicación: Usamos un contenedor ligero con Java para correr la aplicación.

2. Construimos y subimos la imagen al registro de GitHub (ghcr.io):

* Construir: docker build -t ghcr.io/organizacionp6/clinica-backend:latest .

* Subir: docker push ghcr.io/organizacionp6/clinica-backend:latest

3. Probamos la imagen localmente:

* Ejecutamos la imagen en un contenedor: docker run -p 8080:8080 ghcr.io/organizacionp6/clinica-backend:latest

Configuración de CI/CD y Contenerización de una Aplicación

Archivos creados:

- Dockerfile: Define cómo construir y ejecutar la imagen Docker.

2. Configuración de docker-compose

¿Qué hicimos?

1. Creamos un archivo docker-compose.yml:

   - Este archivo orquesta varios contenedores necesarios para tu aplicación (backend, frontend y base de datos).

   - Definimos las imágenes y configuramos las variables de entorno necesarias.

2. Ejecutamos todo el sistema:

   * Construimos y levantamos los servicios con: docker compose up --build

   Esto inició:

   - El backend en http://localhost:8080

   - El frontend en http://localhost:3000

   - Una base de datos PostgreSQL.

Archivos creados:

- docker-compose.yml: Coordina los contenedores para que trabajen juntos.

3. Configuración de un Pipeline CI/CD con GitHub Actions

Configuración de CI/CD y Contenerización de una Aplicación

¿Qué hicimos?

1. Creamos un archivo .github/workflows/ci.yml:

   - Configuramos un pipeline para automatizar:

     * La compilación del proyecto.

     * La construcción y publicación de imágenes Docker en ghcr.io.

2. Describimos dos trabajos:

   - Build and Test: Descarga dependencias, compila el proyecto y ejecuta las pruebas.

   - Dockerize and Deploy: Construye y publica imágenes Docker para backend y frontend.

3. Probamos el pipeline:

   - Haciendo push a la rama, verificamos que las imágenes se publicaron automáticamente.

Archivos creados:

- .github/workflows/ci.yml: Define el pipeline CI/CD con GitHub Actions.

Resumen Final

Archivos Importantes:

1. Dockerfile: Contiene las instrucciones para construir la imagen Docker.

2. docker-compose.yml: Orquesta los contenedores para el sistema completo.

3. .github/workflows/ci.yml: Automatiza la construcción, prueba y publicación en un pipeline CI/CD.

Configuración de CI/CD y Contenerización de una Aplicación

Comandos Clave:

1. Construir la imagen Docker: docker build -t ghcr.io/organizacionp6/clinica-backend:latest .

2. Subir la imagen al registro: docker push ghcr.io/organizacionp6/clinica-backend:latest

3. Levantar todo el sistema con Docker Compose: docker compose up --build

Conceptos Básicos:

- Docker: Conteneriza tu aplicación para que pueda ejecutarse en cualquier entorno.

- Docker Compose: Coordina múltiples contenedores.

- CI/CD: Automatiza la construcción, prueba y despliegue de tu aplicación.