CI/CD con GitHub Actions y GHCR (GitHub Container Registry)

Este proyecto utiliza un flujo CI/CD automatizado para construir y desplegar el backend en Kubernetes usando GitHub Actions y MetalLB.

■ Autenticación en GHCR con GHCR_PAT

En lugar de usar el `GITHUB_TOKEN` (que puede expirar o limitarse tras reinicios), configuramos un **token personal (GHCR_PAT)** con permisos mínimos.

Permisos mínimos requeridos:

• **read:packages** → para que Kubernetes pueda descargar imágenes desde GHCR.

• **write:packages** → para que el pipeline pueda subir nuevas imágenes.

■ Configuración de GitHub Secrets

En tu repositorio de GitHub:

1. Ve a **Settings > Secrets and variables > Actions**.

2. Agrega un secreto llamado:

• - `GHCR_PAT` → tu token personal de GitHub con permisos de `read:packages` y `write:packages`.

3. Ajusta el workflow (`.github/workflows/deploy.yml`) para usar `GHCR_PAT` en lugar de `GITHUB_TOKEN`:

• name: Log in to Container Registry

uses: docker/login-action@v3

with:

registry: ghcr.io

username: ${{ github.actor }}

password: ${{ secrets.GHCR_PAT }}

■■ Flujo CI/CD

1. **Push a main** → El pipeline se dispara automáticamente.

2. **Build & Push** → Construye la imagen Docker y la sube a GHCR (`ghcr.io/poravv/message-sender`).

3. **Deploy** → Aplica los manifests (`namespace.yaml`, `configmap.yaml`, `backend-deployment.yaml`, `ingress.yaml`).

4. **Rollout status** → Verifica que el despliegue fue exitoso.

5. **Health Check** → Hace un curl al endpoint `/health` del servicio.

■ Persistencia después de reinicios

• El runner se configuró como un **servicio systemd**, por lo que seguirá activo después de reinicios:

sudo systemctl enable actions.runner.poravv-message-sender.k8s-master.service

sudo systemctl status actions.runner.poravv-message-sender.k8s-master.service

• El secreto `GHCR_PAT` se almacena en **GitHub Secrets**, no en el servidor, por lo que siempre tendrá acceso a las imágenes después de reinicios.

■ Acceso a la aplicación

El acceso público se gestiona vía **Ingress + MetalLB**:

• Backend expuesto en: `https://sender.mindtechpy.net`

• Longhorn UI en: `http://192.168.100.230/#/dashboard` (red interna).

■ Con esta configuración el CI/CD queda estable, persistente y seguro frente a reinicios.