# GitHub Sync Webhook Service – Documentation

## Overview

This project automates Git repository syncing and container restarts via a webhook. When you push code to GitHub, the system automatically pulls the latest changes and restarts any Docker containers associated with the updated repository.

## Features

- Supports multiple repositories (private & public)

- Restarts Docker containers based on labels

- Uses GitHub webhooks to trigger updates

- Private GitHub repo support via mounted SSH key

- Lightweight Flask-based service inside a Docker container

## Requirements

1. Docker & Docker Compose  
2. SSH Access to Private Repos

## Project Structure

.  
├── .env # Environment variables for each repo  
├── repos/ # Cloned repositories will live here  
├── Dockerfile # Container setup  
├── docker-compose.yml # Deployment configuration  
├── requirements.txt # Python dependencies  
├── sync.py # Webhook Flask server  
└── README.md # GitHub Readme

## .env Configuration

REPOS=github-sync-test,another-repo  
  
REPO\_URL\_github-sync-test=git@github.com:ahmedkhamis12/github-sync-test.git  
CONTAINER\_LABEL\_github-sync-test=odoo\_group  
  
REPO\_URL\_another-repo=git@github.com:username/another-repo.git  
CONTAINER\_LABEL\_another-repo=another\_group  
  
BASE\_REPO\_DIR=/repos

## How to Use

1. Clone and configure the project:

git clone git@github.com:your-org/github-sync-tool.git  
 cd github-sync-tool  
 cp .env.example .env

2. Run the service:

docker-compose up -d --build

3. Set up GitHub Webhook:

- Go to GitHub repo → Settings → Webhooks → Add webhook

- Payload URL: http://<your-server-ip>:8000/webhook

- Content type: application/json

- Trigger: Just the push event

## Docker Compose Overview

version: "3.8"  
services:  
 github-sync:  
 image: github-sync  
 container\_name: github-sync  
 build: .  
 ports:  
 - "8000:8000"  
 volumes:  
 - ~/.ssh:/root/.ssh  
 - /var/run/docker.sock:/var/run/docker.sock  
 - ./repos:/repos  
 - ./.env:/app/.env:ro  
 restart: unless-stopped

## Private Repository Access

Make sure your ~/.ssh directory contains:

- A valid private key

- Proper known\_hosts entry or use: ssh-keyscan github.com >> ~/.ssh/known\_hosts

- Optional: SSH config if needed

## Testing

Push to your GitHub repo and check logs:

docker logs github-sync -f

## Support

For support or customization, contact Ahmed Khamis (akhamis574@gmail.com)