

Guía de Deployment – Sistema de Seguridad IoT

1. Requisitos Previos

- Raspberry Pi 4 con Raspberry Pi OS 64 bits.
- Docker y Docker Compose instalados.
- Usuario con permisos para ejecutar docker sin sudo.
- Acceso a Internet para descargar imágenes base.

2. Estructura de Docker Compose

Un archivo docker-compose.yml orquesta todos los servicios: broker MQTT, backend REST, worker MQTT, servicio IA, frontend y base de datos PostgreSQL. Cada servicio tiene su propia imagen y configuración.

```
version: '3.8'
services:
  mqtt:
    image: eclipse-mosquitto:2
    container_name: mqtt
    ports:
      - "1883:1883"
    volumes:
      - ./mosquitto/conf:/mosquitto/config
      - ./mosquitto/data:/mosquitto/data

  db:
    image: postgres:14
    container_name: postgres
    environment:
      - POSTGRES_DB=seguridad
      - POSTGRES_USER=seguridad
      - POSTGRES_PASSWORD=seguridad123
    volumes:
      - ./postgres/data:/var/lib/postgresql/data

  backend:
    build: ./backend
    container_name: backend
    depends_on:
      - db
    environment:
      - DATABASE_URL=postgresql://seguridad:seguridad123@db:5432/seguridad
    ports:
      - "8000:8000"

  worker:
    build: ./worker
    container_name: worker
    depends_on:
      - mqtt
      - db

  ai:
    build: ./ai
    container_name: ai
    depends_on:
      - backend

  frontend:
    build: ./frontend
    container_name: frontend
```

```
ports:  
  - "3000:80"
```

3. CI/CD con GitHub Actions (Ejemplo)

```
name: CI  
  
on:  
  push:  
    branches: [ main ]  
  
jobs:  
  build-and-test:  
    runs-on: ubuntu-latest  
    steps:  
      - uses: actions/checkout@v4  
      - name: Set up Docker Buildx  
        uses: docker/setup-buildx-action@v3  
      - name: Build backend image  
        run: docker build -t backend:latest backend  
      - name: Build worker image  
        run: docker build -t worker:latest worker  
      - name: Run unit tests  
        run:  
          docker run --rm backend:latest pytest
```