

3.11 PaperMerge. Gestión documental



Imagen obtenida de <https://twitter.com/papermerge>

Esta herramienta sirve para...

realizar la gestión documental a través de una interface web sencilla de utilizar. Cuenta con OCR por lo que podremos buscar textos dentro de estos documentos.

Web de proyecto y otros enlaces de interés

Página web: <https://papermerge.com/>

Repositorio de los proyectos que componen esta herramienta <https://github.com/papermerge>

Puesta en marcha

La documentación del proyecto <https://docs.papermerge.io/Installation/docker-compose.html> recomienda no utilizar docker-compose para un sistema en producción.

Como en ocasiones anteriores vamos a hacer con docker-compose para ello accedemos al terminal y escribimos

```
cd $HOME  
mkdir papermerge  
cd papermerge  
nano .env
```

y dentro del fichero copiaremos el siguiente contenido:

```
APP_IMAGE=papermerge/papermerge  
APP_TAG=latest  
PAPERMERGE_JS_IMAGE=papermerge/papermerge.js  
PAPERMERGE_JS_TAG=latest  
  
TIMEZONE=Europe/Madrid  
  
DB_USER=postgres  
DB_NAME=postgres  
DB_PASSWORD=postgres  
DB_HOST=db  
DB_PORT=5432  
  
USE_HOSTNAME=papermerge.local  
  
REDIS_HOST=redis  
REDIS_PORT=6379  
  
SECRET_KEY=12345abcdxyz
```

```
SUPERUSER_USERNAME=admin  
SUPERUSER_EMAIL=admin@example.com  
SUPERUSER_PASSWORD=admin
```

como en ocasiones anteriores, para guardar los cambios pulsaremos `control + x` y cuando nos pregunte aceptaremos.

Hasta ahora nunca habíamos utilizado ningún fichero `.env` pero lo correcto es establecer determinadas configuraciones en los mismos. Los ficheros `.env` son ficheros que no se comparten en el repositorio de código mientras que los ficheros `docker-compose.yml` si. Lo que suele hacerse es compartirse algún fichero como `env-sample` o de nombre similar de modo que cualquier persona pueda ver ahí los parámetros que debe configurar pero sin ver tus valores reales y le basta con renombrarlo a `.env`.

Ahora si. Vamos a crear el fichero `docker-compose.yml` para ello escribimos en el terminal

```
nano docker-compose.yml
```

y dentro del fichero copiamos el siguiente contenido:

```
version: '3.7'  
  
# Any top-level key starting with x- in a Docker Compose file will be  
# ignored  
x-backend: &backend # yaml anchor definition  
  
image: ${APP_IMAGE}:${APP_TAG}  
  
volumes:  
  - media_root:/app/media  
  - xapien_index:/app/xapien_index  
  
environment:  
  # PAPERMERGE__<section>__<variable>  
  - PAPERMERGE__MAIN__SECRET_KEY=${SECRET_KEY}  
  - PAPERMERGE__DATABASE__TYPE=postgres  
  - PAPERMERGE__DATABASE__USER=${DB_USER}  
  - PAPERMERGE__DATABASE__NAME=${DB_NAME}  
  - PAPERMERGE__DATABASE__PASSWORD=${DB_PASSWORD}  
  - PAPERMERGE__DATABASE__HOST=${DB_HOST}  
  - PAPERMERGE__REDIS__HOST=${REDIS_HOST}  
  - PAPERMERGE__REDIS__PORT=${REDIS_PORT}
```

```
- PAPERMERGE__MAIN__TIMEZONE=${TIMEZONE}
# path where xapian index data is stored
- PAPERMERGE__SEARCH__PATH=/app/xapian_index
- DJANGO_SUPERUSER_USERNAME=${SUPERUSER_USERNAME}
- DJANGO_SUPERUSER_EMAIL=${SUPERUSER_EMAIL}
- DJANGO_SUPERUSER_PASSWORD=${SUPERUSER_PASSWORD}
- DJANGO_SETTINGS_MODULE=config.settings
```

services:

worker: # celery worker

```
<<: *backend
```

```
command: worker
```

ws_server: # websockets server / daphne

```
<<: *backend
```

```
command: ws_server
```

labels:

```
- "traefik.enable=true"
```

```
- "traefik.http.routers.ws_server.rule=Host(`${USE_HOSTNAME}`) && PathPrefix(`/ws/`)"
```

backend: # rest api backend / uwsgi

```
<<: *backend
```

labels:

```
- "traefik.enable=true"
```

```
- "traefik.http.routers.backend.rule=Host(`${USE_HOSTNAME}`) && PathPrefix(`/api/`)"
```

db:

```
image: postgres:14.4
```

volumes:

```
- postgres_data:/var/lib/postgresql/data/
```

environment:

```
- POSTGRES_USER=${DB_USER}
```

```
- POSTGRES_DB=${DB_NAME}
```

```
- POSTGRES_PASSWORD=${DB_PASSWORD}
```

redis:

```
image: 'redis:6'
```

ports:

```
- '6379:6379'
```

volumes:

```
- redis_data:/data
```

```
traefik:
  image: "traefik:v2.6"
  command:
    #- "--log.level=DEBUG"
    - "--api.insecure=true"
    - "--providers.docker=true"
    - "--providers.docker.exposedbydefault=false"
    - "--entrypoints.web.address=:80"
  ports:
    - "80:80"
    - "8080:8080"
  volumes:
    - "/var/run/docker.sock:/var/run/docker.sock:ro"

frontend: # emberjs
  image: ${PAPERMERGE_JS_IMAGE}:${PAPERMERGE_JS_TAG}
  labels:
    - "traefik.enable=true"
    - "traefik.http.routers.traefik.rule=Host(`${USE_HOSTNAME}`) && PathPrefix(`/)"
  volumes:
    postgres_data:
    media_root:
    xapián_index:
    redis_data:
```

como en ocasiones anteriores, para guardar los cambios pulsaremos `control + x` y cuando nos pregunte aceptaremos.

Ahora en el equipo desde el que vayamos a acceder al servicio en el terminal escribiremos:

```
sudo nano /etc/hosts
```

Y en dicho fichero añade al final del mismo el texto

```
192.168.0.201    papermerge.local
# En lugar de 192.168.0.201 vosotros/a pondréis la IP de vuestra Raspberry Pi
```

como en ocasiones anteriores, para guardar los cambios pulsaremos `control + x` y cuando nos pregunte aceptaremos.

