

Docker yaml templates

- [paperless-ngx](#)
- [watchtower](#)
- [vaultwarden](#)

paperless-ngx

```
# docker-compose file for running paperless from the Docker Hub.
# This file contains everything paperless needs to run.
# Paperless supports amd64, arm and arm64 hardware.
#
# All compose files of paperless configure paperless in the following way:
#
# - Paperless is (re)started on system boot, if it was running before shutdown.
# - Docker volumes for storing data are managed by Docker.
# - Folders for importing and exporting files are created in the same directory
#   as this file and mounted to the correct folders inside the container.
# - Paperless listens on port 8010.
#
# In addition to that, this docker-compose file adds the following optional
# configurations:
#
# - Instead of SQLite (default), PostgreSQL is used as the database server.
#
# To install and update paperless with this file, do the following:
#
# - Open portainer Stacks list and click 'Add stack'
# - Paste the contents of this file and assign a name, e.g. 'Paperless'
# - Click 'Deploy the stack' and wait for it to be deployed
# - Open the list of containers, select paperless_webserver_1
# - Click 'Console' and then 'Connect' to open the command line inside the container
# - Run 'python3 manage.py createsuperuser' to create a user
# - Exit the console
#
# For more extensive installation and update instructions, refer to the
# documentation.

version: "3.4"

services:
  broker:
    image: docker.io/library/redis:7
```

restart: always

volumes:

- /opt/paperless/redisdata:/data

db:

image: docker.io/library/postgres:13

restart: always

volumes:

- /opt/paperless/pgdata:/var/lib/postgresql/data

environment:

POSTGRES_DB: paperless

POSTGRES_USER: paperless

POSTGRES_PASSWORD: paperless

webserver:

image: ghcr.io/paperless-ngx/paperless-ngx:latest

restart: always

depends_on:

- db

- broker

ports:

- 8000:8000

healthcheck:

test: ["CMD", "curl", "-fs", "-S", "--max-time", "2", "http://localhost:8000"]

interval: 30s

timeout: 10s

retries: 5

volumes:

- /opt/paperless/data:/usr/src/paperless/data

- /opt/paperless/media:/usr/src/paperless/media

- /opt/paperless/export:/usr/src/paperless/export

- /consume:/usr/src/paperless/consume

environment:

PAPERLESS_REDIS: redis://broker:6379

PAPERLESS_DBHOST: db

PAPERLESS_TIME_ZONE: Europe/Berlin

PAPERLESS_OCR_IMAGE_DPI: 300

PAPERLESS_OCR_LANGUAGES: deu

PAPERLESS_OCR_LANGUAGE: deu

The UID and GID of the user used to run paperless in the container. Set this

```
# to your UID and GID on the host so that you have write access to the
# consumption directory.
    USERMAP_UID: 1000
    USERMAP_GID: 100
# Additional languages to install for text recognition, separated by a
# whitespace. Note that this is
# different from PAPERLESS_OCR_LANGUAGE (default=eng), which defines the
# language used for OCR.
# The container installs English, German, Italian, Spanish and French by
# default.
# See https://packages.debian.org/search?keywords=tesseract-ocr-&searchon=names&suite=buster
# for available languages.
    #PAPERLESS_OCR_LANGUAGES: tur ces
# Adjust this key if you plan to make paperless available publicly. It should
# be a very long sequence of random characters. You don't need to remember it.
    #PAPERLESS_SECRET_KEY: change-me
# Use this variable to set a timezone for the Paperless Docker containers. If not specified, defaults to UTC.
    #PAPERLESS_TIME_ZONE: America/Los_Angeles
# The default language to use for OCR. Set this to the language most of your
# documents are written in.
    #PAPERLESS_OCR_LANGUAGE: eng
```

watchtower

version: '3'

services:

app:

container_name: WatchTower

restart: always

image: containrrr/watchtower:latest

volumes:

- /var/run/docker.sock:/var/run/docker.sock
- /etc/timezone:/etc/timezone:ro

environment:

- WATCHTOWER_DEBUG=true
- WATCHTOWER_CLEANUP=true
- WATCHTOWER_INCLUDE_STOPPED=true
- WATCHTOWER_ROLLING_RESTART=true
- WATCHTOWER_SCHEDULE=0 0 18 * * *
- WATCHTOWER_NOTIFICATIONS=gotify
- WATCHTOWER_NOTIFICATION_GOTIFY_URL=<https://gotifyURL>
- WATCHTOWER_NOTIFICATION_GOTIFY_TOKEN=<token>

vaultwarden

version: '3'

services:

bitwarden:

image: vaultwarden/server:latest

container_name: vaultwarden

restart: always

environment:

- WEBSOCKET_ENABLED=true
- SIGNUPS_ALLOWED=true
- INVITATIONS_ALLOWED=false
- DOMAIN=https://<YourDynamicDomainName>

ports:

- 8080:80

volumes:

- /opt/vaultwarden/data:/data