

# Stacks

My favorites docker stacks

- [Utils](#)
- [Filebrowser](#)
- [Nextcloud](#)
- [JDownloader](#)
- [Sonarqube](#)

# Utils

## Services

### Nginx proxy manager

#### Env

- TZ

#### Volumes

- /config

## Shepherd

Automated service image updater

#### Env

- SHEPHERD\_SLEEP\_TIME
- SHEPHERD\_IMAGE\_AUTOCLEAN\_LIMIT
- TZ

## Ofelia

Docker cron scheduler

#### Env

- TZ

#### Volumes

- /var/run/docker.sock

## Stack

```
version: '3'

services:
  nginx_proxy_manager:
    image: jlesage/nginx-proxy-manager:latest
    restart: always
    ports:
      - '80:8080' # Public HTTP Port
      - '443:4443' # Public HTTPS Port
      #- '81:8181' # Admin Web Port
      # Add any other Stream port you want to expose
      # - '21:21' # FTP
    environment:
      TZ: ${TZ}
      # Uncomment this if IPv6 is not enabled on your host
      DISABLE_IPV6: 'true'
    volumes:
      - nginx_proxy_data:/config
    networks:
      - proxy_network

  shepherd:
    image: mazzolino/shepherd:latest
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock
    environment:
      SLEEP_TIME: ${SHEPHERD_SLEEP_TIME}
      IMAGE_AUTOCLEAN_LIMIT: ${SHEPHERD_IMAGE_AUTOCLEAN_LIMIT}
      TZ: ${TZ}
    deploy:
      placement:
        constraints:
          - node.role == manager

  scheduler:
    image: mcuadros/ofelia:latest
    command: daemon --docker
    environment:
      TZ: ${TZ}
    volumes:
      - /var/run/docker.sock:/var/run/docker.sock:ro
```

```
#labels:
```

```
#ofelia.job-local.my-test-job.schedule: "@every 5m"
```

```
#ofelia.job-local.my-test-job.command: "date"
```

```
volumes:
```

```
  nginx_proxy_data:
```

```
    external: true
```

```
networks:
```

```
  proxy_network:
```

```
    external: true
```

# Filebrowser

## Services

### Filebrowser

Filebrowser is a web file manager that allows you to manage your files in a web browser.

#### Env

- TZ
- PUID
- PGID

#### Volumes

- /config
- /database
- /srv

## Stack

```
version: '3'

services:
  app:
    image: 'filebrowser/filebrowser:s6'
    volumes:
      - filebrowser_settings:/config
      - filebrowser_data:/database
      - /mnt/disk:/srv
    environment:
      TZ: ${TZ}
      PGID: ${PGID}
      PUID: ${PUID}
    networks:
      - proxy_network
```

```
volumes:
```

```
  filebrowser_settings:
```

```
  filebrowser_data:
```

```
networks:
```

```
  proxy_network:
```

```
    external: true
```

# Nextcloud

## Services

### MariaDB

#### Env

- MYSQL\_ROOT\_PASSWORD

#### Volumes

- /var/lib/mysql

#### Command

- `--transaction-isolation=READ-COMMITTED --binlog-format=ROW`

# Nextcloud

#### Env

- APACHE\_DISABLE\_REWRITE\_IP
- OVERWRITEHOST
- OVERWRITEPROTOCOL
- OVERWRITEWEBROOT
- PHP\_MEMORY\_LIMIT
- PHP\_UPLOAD\_LIMIT
- TRUSTED\_PROXIES
- PGID
- PUID
- TZ

#### Volumes

- /var/www/html
- /var/www/html/data

## Stack

version: '3'

services:

db:

image: mariadb:10.5

restart: always

command: --transaction-isolation=READ-COMMITTED --binlog-format=ROW

volumes:

- mariadb:/var/lib/mysql

environment:

- MYSQL\_ROOT\_PASSWORD=\${MYSQL\_ROOT\_PASSWORD}

networks:

- nextcloud\_network

app:

depends\_on:

- db

image: nextcloud:latest

restart: always

volumes:

- config:/var/www/html

- \${NEXTCLOUD\_DATA\_PATH}:/var/www/html/data

environment:

APACHE\_DISABLE\_REWRITE\_IP: 1

OVERWRITEHOST: nextcloud.mydomain.fr

OVERWRITEPROTOCOL: https

OVERWRITEWEBROOT: /

PHP\_MEMORY\_LIMIT: 4G

PHP\_UPLOAD\_LIMIT: 10G

TRUSTED\_PROXIES: IPV4 Gateway - 10.0.3.1

PGID: \${PGID}

PUID: \${PUID}

TZ: \${TZ}

networks:

- proxy\_network

- nextcloud\_network

volumes:

config:

mariadb:



```
networks:
```

```
  proxy_network:
```

```
    external: true
```

```
  nextcloud_network:
```

# JDownloader

## Services

### JDownloader

#### Env

- MYJD\_DEVICE\_NAME
- MYJD\_PASSWORD
- MYJD\_USER
- XDG\_DOWNLOAD\_DIR
- PUID
- PGID

#### Volumes

- /downloads
- /opt/JDownloader
- /opt/JDownloader/cfg

## Stack

```
version: '3'

services:
  app:
    image: jaymoulin/jdownloader:latest
    ports:
      - '3129:3129'
    volumes:
      - ${DIRECT_DOWNLOAD_PATH}:/downloads
      - app:/opt/JDownloader
      - config:/opt/JDownloader/app/cfg
    environment:
      MYJD_DEVICE_NAME: ${MYJD_DEVICE_NAME}
      MYJD_PASSWORD: ${MYJD_PASSWORD}
```

MYJD\_USER: \${MYJD\_USER}

XDG\_DOWNLOAD\_DIR: /downloads

PUID: \${PUID}

PGID: \${PGID}

volumes:

config:

app:

# Sonarqube

## Services

### Postgres

#### Env

- ALLOW\_EMPTY\_PASSWORD
- POSTGRESQL\_USERNAME
- POSTGRESQL\_DATABASE
- TZ

#### Volumes

- /bitnami/postgresql
- /docker-entrypoint-initdb.d
- /docker-entrypoint-preinitdb.d

# Sonarqube

#### Env

- ALLOW\_EMPTY\_PASSWORD
- SONARQUBE\_DATABASE\_HOST
- SONARQUBE\_DATABASE\_PORT\_NUMBER
- SONARQUBE\_DATABASE\_USER
- SONARQUBE\_DATABASE\_NAME
- SONARQUBE\_USERNAME
- SONARQUBE\_PASSWORD
- SONARQUBE\_EMAIL
- TZ

#### Volumes

- /bitnami/sonarqube

## Stack

```
version: '3'

services:
  postgresql:
    image: docker.io/bitnami/postgresql:13
    volumes:
      - 'postgresql_data:/bitnami/postgresql'
      - 'entry_point:/docker-entrypoint-preinitdb.d'
      - 'entry_point:/docker-entrypoint-initdb.d'
    environment:
      # ALLOW_EMPTY_PASSWORD is recommended only for development.
      ALLOW_EMPTY_PASSWORD: 'yes'
      POSTGRESQL_USERNAME: ${SONARQUBE_DB_USER}
      POSTGRESQL_DATABASE: ${SONARQUBE_DB_NAME}
      TZ: ${TZ}
    networks:
      - proxy_network
  sonarqube:
    image: bitnami/sonarqube:9.6.1
    #ports:
    # - '8080:8080'
    volumes:
      - 'sonarqube_data:/bitnami/sonarqube'
    depends_on:
      - postgresql
    environment:
      # ALLOW_EMPTY_PASSWORD is recommended only for development.
      ALLOW_EMPTY_PASSWORD: 'yes'
      SONARQUBE_DATABASE_HOST: postgresql
      SONARQUBE_DATABASE_PORT_NUMBER: 5432
      SONARQUBE_DATABASE_USER: ${SONARQUBE_DB_USER}
      SONARQUBE_DATABASE_NAME: ${SONARQUBE_DB_NAME}
      SONARQUBE_USERNAME: ${SONARQUBE_USER}
      SONARQUBE_PASSWORD: ${SONARQUBE_PASSWORD}
      SONARQUBE_EMAIL: ${SONARQUBE_EMAIL}

      TZ: ${TZ}
    networks:
      - proxy_network
volumes:
  postgresql_data:
```

sonarqube\_data:

entry\_point:

networks:

proxy\_network:

external: true