

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