

Mettre en place Nextcloud Docker

Prérequis :

- Avoir une base de donnée `Postgres` déployé en local
 - Voir [déploiement Postgres](#)
- Avoir un nom de domaine
 - Dans notre cas le nom de domaine utilisé est : `miraceti.net`

Il existe plusieurs manière de mettre en place `Nextcloud` Docker. Dans notre cas nous allons utiliser une base de données `Postgres` déjà déployé avec Docker

Configuration nom de domaine

Grâce à notre nom de domaine, nous allons pouvoir créer des sous domaine (en CNAME `A`).

Sur le site de nom de domaine respectif, dans la section DNS, ajouter l'entrée suivante :

- `cloud.miraceti.net` `A` `ip.adress.server`

Configuration Docker Compose

Pour accéder à notre base de données externe, nous devons importer le `network` lié à la base de donnée (dans notre cas le network est "`bddnetwork`").

De plus dans la variable d'environnement `POSTGRES_HOST` du conteneur de nextcloud **on doit renseigner le nom du conteneur BDD externe** (Dans notre cas "`postgres`")

Fichier `docker-compose.yml` :

```
services:

  nextcloud:
    image: ${NEXTCLOUD_IMAGE_TAG}
    container_name: nextcloud
```

```
networks:
  - bddnetwork
ports:
  - 8082:80
volumes:
  - /path/to/data/nextcloud:/var/www/html
  - /path/to/conf/nextcloud/config:/var/www/html/config
restart: unless-stopped
environment:
  TZ: ${NEXTCLOUD_TIMEZONE}
  POSTGRES_HOST: postgres
  DB_PORT: 5432
  POSTGRES_DB: ${NEXTCLOUD_DB_NAME}
  POSTGRES_USER: ${NEXTCLOUD_DB_USER}
  POSTGRES_PASSWORD: ${NEXTCLOUD_DB_PASSWORD}
  REDIS_HOST: redis-nextcloud
  REDIS_HOST_PORT: 6379
  REDIS_HOST_PASSWORD: ${NEXTCLOUD_REDIS_PASSWORD}
#   TRUSTED_PROXIES: ${NEXTCLOUD_HOSTNAME}
  OVERWRITECLIURL: ${NEXTCLOUD_URL}
  OVERWRITEPROTOCOL: https
  OVERWRITEHOST: ${NEXTCLOUD_HOSTNAME}
#   PHP_MEMORY_LIMIT: 2048M## If you want to overwrite Php Memory Limit
#   PHP_UPLOAD_LIMIT: 2048M## If you want to overwrite Php Upload Limit

redis-nextcloud:
  image: ${NEXTCLOUD_REDIS_IMAGE_TAG}
  container_name: redis-nextcloud
  command: ["redis-server", "--requirepass", "${NEXTCLOUD_REDIS_PASSWORD}"]
  volumes:
    - /path/to/data/redis-nextcloud:/data
  networks:
    - bddnetwork
  healthcheck:
    test: ["CMD", "redis-cli", "ping"]
    interval: 10s
    timeout: 5s
    retries: 3
    start_period: 60s
  restart: unless-stopped
```

```
volumes:
  nextcloud:
  redis-nextcloud:

networks:
  bddnetwork:
    external: true
```

Fichier `.env` :

```
# Nextcloud Variables
NEXTCLOUD_REDIS_IMAGE_TAG=redis:7.2
NEXTCLOUD_IMAGE_TAG=nextcloud:29.0
NEXTCLOUD_REDIS_PASSWORD=5454fdsLJDSqd45
NEXTCLOUD_DB_NAME=nextcloudb
NEXTCLOUD_DB_USER=nextcloudbuser
NEXTCLOUD_DB_PASSWORD=q5s4dazelqksdjfga
NEXTCLOUD_URL=https://cloud.miraceti.net
NEXTCLOUD_HOSTNAME=cloud.miraceti.net
# Timezone inside container
# A list of these tz database names can be looked up at Wikipedia
# https://en.wikipedia.org/wiki/List_of_tz_database_time_zones
NEXTCLOUD_TIMEZONE=Europe/Paris
```

Démarrer le docker compose devrait rendre Nextcloud accessible en local, ou sur l'adresse IP de votre serveur sur le réseau local, sur le port `8082`.

Configuration Apache2

Création d'un nouveau host dans `/etc/apache2/sites-available` avec le nom suivant "`cloud.miraceti.net.conf`":

```
<VirtualHost *:80>
    ServerName cloud.miraceti.net

    ErrorLog ${APACHE_LOG_DIR}/cloud.miraceti.net.log
    CustomLog ${APACHE_LOG_DIR}/cloud.miraceti.net.log combined

    ProxyPreserveHost On
    ProxyRequests Off
    ProxyPass / http://localhost:8082/[]# Port Used by container Nextcloud
    ProxyPassReverse / http://localhost:8082/[]# Port Used by container Nextcloud

    <IfModule mod_dav.c>
        Dav off
    </IfModule>

    <Proxy *>
        Order deny,allow
        Allow from all
    </Proxy>
</VirtualHost>
```

Vérification de la nouvelle configuration :

```
apachectl configtest
```

Activation de la nouvelle configuration :

```
sudo a2ensite cloud.miraceti.net.conf
```

Revision #16

Created 17 July 2024 14:46:56 by gpatruno

Updated 17 July 2024 15:46:48 by gpatruno