# Deploy Nextcloud with docker compose - (Medium site)

Link:
8 Setembro 2023.

In the spirit of DIY, experimenting, and learning, I set up a Nextcloud server to replace Google Drive and One Drive. Over the years my Nextcloud instance has become home to many of my backed up files, contacts, calendar, notes, and my server even has an office suite thanks to Collabora.

Nextcloud can be a pretty awesome, open source alternative to many services.

For better or for worse, I decided to deploy Nextcloud on my home server using a Docker container. If you're looking to get experience in Docker or trying to find out how to set up a Docker Compose file to deploy Nextcloud you're in the right spot.

The docker-compose.yml

```yaml
---
version: '3'

services:
  nextcloud:
    image: nextcloud
    container_name: nextcloud
    restart: unless-stopped
    networks:
      - cloud
    depends_on:
      - nextclouddb
      - redis
    ports:
      - 8081:80
    volumes:
      - ./html:/var/www/html
      - ./custom_apps:/var/www/html/custom_apps
      - ./config:/var/www/html/config
      - ./data:/var/www/html/data
    environment:
```

```yaml
      - PUID=1000
      - PGID=1000
      - TZ=America/Los_Angeles
      - MYSQL_DATABASE=nextcloud
      - MYSQL_USER=nextcloud
      - MYSQL_PASSWORD=dbpassword
      - MYSQL_HOST=nextclouddb
      - REDIS_HOST=redis

nextclouddb:
  image: mariadb
  container_name: nextcloud-db
  restart: unless-stopped
  command: --transaction-isolation=READ-COMMITTED --binlog-format=ROW
  networks:
    - cloud
  volumes:
    - ./nextclouddb:/var/lib/mysql
  environment:
    - PUID=1000
    - PGID=1000
    - TZ=America/Los_Angeles
    - MYSQL_RANDOM_ROOT_PASSWORD=true
    - MYSQL_PASSWORD=dbpassword
    - MYSQL_DATABASE=nextcloud
    - MYSQL_USER=nextcloud

collabora:
  image: collabora/code
  container_name: collabora
  restart: unless-stopped
  networks:
    - cloud
  environment:
    - PUID=1000
    - PGID=1000
    - TZ=America/Los_Angeles
    - password=password
    - username=nextcloud
    - domain=example.com
    - extra_params=--o:ssl.enable=true
  ports:
    - 9980:9980

redis:
  image: redis:alpine
  container_name: redis
  volumes:
    - ./redis:/data
  networks:
    - cloud

nginx-proxy:
```

```
    image: 'jc21/nginx-proxy-manager:latest'
    container_name: nginx-proxy
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=America/Los_Angeles
    restart: unless-stopped
    ports:
      - '80:80'
      - '81:81'
      - '443:443'
    volumes:
      - ./data:/data
      - ./letsencrypt:/etc/letsencrypt

networks:
  cloud:
    name: cloud
    driver: bridge
```

Alright. Lets break this down.

This Docker Compose file will deploy 5 containers. They are:

1. Nextcloud
2. MySQL database required for Nextcloud
3. Collabora — A awesome open source office suite similar to google docs. Collabora Office also has mobile apps.
4. Redis — Memory Caching. If you're going to rely on next cloud for important files, I highly recommend setting up Redis.
5. Nginx Proxy Manager — A reverse proxy manager to handle incoming requests to the server.

# Docker Compose

A little explanation for what each line does.

Nextcloud:

```
nextcloud:
    image: nextcloud # The image that will be used. The official nextcloud docker
    container_name: nextcloud # Just the name of the container. Help you identify it
    restart: unless-stopped # If something happens like the container crashes then we want the container to start u
    networks: # Link all the containers through the network "cloud"
      - cloud
    depends_on: # Wait for the database and redis containers before starting nextcloud
      - nextclouddb
      - redis
```

```
  ports: # If you have multiple web service on your server you need to change the port. I am directing nextcloud 1
    - 8081:80
  volumes: # These are important. This will map a file directory inside the container to a directory on your actual
    - ./html:/var/www/html # Map the /var/www/html directory in the container to the html folder in the same folde
    - ./custom_apps:/var/www/html/custom_apps # These volumes allow us to easily interact with the files in the c
    - ./config:/var/www/html/config
    - ./data:/var/www/html/data
  environment:
    - PUID=1000 # The user ids. Most likely both should be 1000. Incorrectly setting these will led to file permissio
    - PGID=1000 # Set these to whatever your user is.
    - TZ=America/Los_Angeles # Set this to your timezone
    - MYSQL_DATABASE=nextcloud # This is the database information we will set up in the next section
    - MYSQL_USER=nextcloud
    - MYSQL_PASSWORD=dbpassword
    - MYSQL_HOST=nextclouddb
    - REDIS_HOST=redis # The redis container to use
```

## Nextcloud Database:

```
nextclouddb:
  image: mariadb # offical mariadb image
  container_name: nextcloud-db
  restart: unless-stopped
  command: --transaction-isolation=READ-COMMITTED --binlog-format=ROW # I honestly cant remember. If you l
  networks:
    - cloud
  volumes:
    - ./nextclouddb:/var/lib/mysql
  environment:
    - PUID=1000 # Should be the same as the other containers
    - PGID=1000
    - TZ=America/Los_Angeles
    - MYSQL_RANDOM_ROOT_PASSWORD=true
    - MYSQL_PASSWORD=dbpassword # Same information that was entered in the nextcloud portion
    - MYSQL_DATABASE=nextcloud
    - MYSQL_USER=nextcloud
```

## Collabora (Optional, but really cool):

```
collabora:
  image: collabora/code:latest
  container_name: collabora
  restart: unless-stopped
  networks:
    - cloud
  environment:
    - PUID=1000
    - PGID=1000
    - TZ=America/Los_Angeles # Should be the same as the others
    - password=password
    - username=nextcloud
    - domain=example.com # domain your nextcloud is on
```

```
      - extra_params=--o:ssl.enable=true # Use if have ssl. You should.
    ports:
      - 9980:9980
```

Redis (Optional, but seriously, just add it. You will run into file lock issues eventually and Redis will prevent it):

```
  redis:
    image: redis:alpine
    container_name: redis
    volumes:
      - ./redis:/data
    networks:
      - cloud
```

Nginx Proxy Manager:

```
  nginx-proxy:
    image: 'jc21/nginx-proxy-manager:latest'
    container_name: nginx-proxy
    environment:
      - PUID=1000
      - PGID=1000
      - TZ=America/Los_Angeles
    restart: unless-stopped
    ports:
      - '80:80'
      - '81:81'
      - '443:443'
    volumes:
      - ./data:/data
      - ./letsencrypt:/etc/letsencrypt
```

Nginx Proxy manager is an awesome program that will direct incoming traffic to the server towards Nextcloud. Add a new host.

On the details panel enter your domain name.

Scheme = http, Forward Hostname = whatever your machines local IP address is, (for example 192.168.1.35) and forward port = 80.

Cache Assets, Block Common Exploits, Websockets support all should be on.

Next on the Custom Locations tab we will enable caldav and carddav to allow remote access to your calendars and contacts.

Location 1:

1. location = /.well-known/caldav
2. scheme = html

3. Forward Hostname = <local IP>/ remote.php/dav
4. Forward Port 80

Location 2:

1. location = /.well-known/carddav
2. scheme = html
3. Forward Hostname = <local IP>/ remote.php/dav
4. Forward Port 80

# Additional Configuration

Now if you are setting up Nextcloud to work with your custom domain you will need to open the config.php file and change trusted domains to whatever your domain is.

If you are trying to access your Nextcloud from your network it could be useful to add your Nextcloud's local IP address.

```
'trusted_domains' =>
array (
0 => 'example.com',
1 => '192.168.1.12:8081',
),
'overwritehost' => 'example.com',
'overwriteprotocol' => 'https',
```

Since Niginx Proxy Manager is set up, the following needs to be added to the config.php file:

```
'default_phone_region' => 'US',
'trustedproxies' =>
array (
0 => 'NginxProxyManager',
1 => '192.168.0.145',
),
```

To solve some of the warnings you need to do the following:

```
'default_phone_region' => 'US',
```

To set up mail alerts you will need to add the following to the config file. The values will need to be obtained from your email provider.

```
'mail_from_address' => 'user', # insert your emails user
'mail_smtpmode' => 'smtp',
'mail_sendmailmode' => 'smtp',
```

```
'mail_domain' => 'example.com', # Your email domain
'mail_smtphost' => 'smtp.example.com',
'mail_smtpport' => '465',
'mail_smtpauth' => 1,
'mail_smtpsecure' => 'ssl',
'mail_smtpname' => 'user@example.com',
'mail_smtppassword' => 'secretpassword',
```

# Run the container

```
docker-compose up -d
```

Congratulations! Nextcloud is setup using docker containers and docker compose! Let me know if you have any questions.

# Potential Issues

- If you run into a 502 Gateway Error try clearing the cookies in your browser for the domain your server is hosted. It works most of the time for me.
- Make sure to update the docker images on a regular basis. Nextcloud in a Docker cannot whole number skip versions. For example if your version is 24 and the newest version is 26, **DO NOT** update straight to 26. I learned this the hard way. Update first to 25. So run "docker-compose pull" somewhat regularly.

---

Revision #3
Created 12 May 2024 12:55:52 by Administrador
Updated 4 July 2024 18:33:49 by Administrador