

Repo Info (/ /nextcloud/)

Short Description

A safe home for all your data

Full Description

Note: the description for this image is longer than the Hub length limit of 25000, so has been trimmed. The full description can be found at <https://github.com/docker-library/docs/tree/master/nextcloud/README.md> (<https://github.com/docker-library/docs/tree/master/nextcloud/README.md>). See [docker/hub-beta-feedback#238](https://github.com/docker/hub-beta-feedback/issues/238) (<https://github.com/docker/hub-beta-feedback/issues/238>) for more information.

Supported tags and respective Dockerfile links

- [12.0.13RC2-apache](#) , [12.0.13-rc-apache](#) , [12.0-rc-apache](#) , [12-rc-apache](#) , [12.0.13RC2](#) , [12.0.13-rc](#) , [12.0-rc](#) , [12-rc](#) ([12.0-rc/apache/Dockerfile](#)) (<https://github.com/nextcloud/docker/blob/b4c914d530e9f309189eaab912244a15561641f7/12.0-rc/apache/Dockerfile>)
- [12.0.13RC2-fpm-alpine](#) , [12.0.13-rc-fpm-alpine](#) , [12.0-rc-fpm-alpine](#) , [12-rc-fpm-alpine](#) ([12.0-rc/fpm-alpine/Dockerfile](#)) (<https://github.com/nextcloud/docker/blob/b4c914d530e9f309189eaab912244a15561641f7/12.0-rc/fpm-alpine/Dockerfile>)
- [12.0.13RC2-fpm](#) , [12.0.13-rc-fpm](#) , [12.0-rc-fpm](#) , [12-rc-fpm](#) ([12.0-rc/fpm/Dockerfile](#)) (<https://github.com/nextcloud/docker/blob/b4c914d530e9f309189eaab912244a15561641f7/12.0-rc/fpm/Dockerfile>)
- [12.0.12-apache](#) , [12.0-apache](#) , [12-apache](#) , [12.0.12](#) , [12.0](#) , [12](#) ([12.0/apache/Dockerfile](#)) (<https://github.com/nextcloud/docker/blob/ae849f190066e68156554d908cce5fa3e38fa05f/12.0/apache/Dockerfile>)
- [12.0.12-fpm-alpine](#) , [12.0-fpm-alpine](#) , [12-fpm-alpine](#) ([12.0/fpm-alpine/Dockerfile](#)) (<https://github.com/nextcloud/docker/blob/ae849f190066e68156554d908cce5fa3e38fa05f/12.0/fpm-alpine/Dockerfile>)
- [12.0.12-fpm](#) , [12.0-fpm](#) , [12-fpm](#) ([12.0/fpm/Dockerfile](#)) (<https://github.com/nextcloud/docker/blob/ae849f190066e68156554d908cce5fa3e38fa05f/12.0/fpm/Dockerfile>)
- [13.0.8RC2-apache](#) , [13.0.8-rc-apache](#) , [13.0-rc-apache](#) , [13-rc-apache](#) , [13.0.8RC2](#) , [13.0.8-rc](#) , [13.0-rc](#) , [13-rc](#) ([13.0-rc/apache/Dockerfile](#)) (<https://github.com/nextcloud/docker/blob/b4c914d530e9f309189eaab912244a15561641f7/13.0-rc/apache/Dockerfile>)
- [13.0.8RC2-fpm-alpine](#) , [13.0.8-rc-fpm-alpine](#) , [13.0-rc-fpm-alpine](#) , [13-rc-fpm-alpine](#) ([13.0-rc/fpm-alpine/Dockerfile](#))

- <https://github.com/nextcloud/docker/blob/b4c914d530e9f309189eaab912244a15561641f7/13.0-rc/fpm-alpine/Dockerfile>
- [13.0.8RC2-fpm , 13.0.8-rc-fpm , 13.0-rc-fpm , 13-rc-fpm \(13.0-rc/fpm/Dockerfile\)](https://github.com/nextcloud/docker/blob/b4c914d530e9f309189eaab912244a15561641f7/13.0-rc/fpm/Dockerfile)
(<https://github.com/nextcloud/docker/blob/b4c914d530e9f309189eaab912244a15561641f7/13.0-rc/fpm/Dockerfile>)
- [13.0.7-apache , 13.0-apache , 13-apache , stable-apache , production-apache , 13.0.7 , 13.0 , 13 , stable , production \(13.0/apache/Dockerfile\)](https://github.com/nextcloud/docker/blob/ae849f190066e68156554d908cce5fa3e38fa05f/13.0/apache/Dockerfile)
(<https://github.com/nextcloud/docker/blob/ae849f190066e68156554d908cce5fa3e38fa05f/13.0/apache/Dockerfile>)
- [13.0.7-fpm-alpine , 13.0-fpm-alpine , 13-fpm-alpine , stable-fpm-alpine , production-fpm-alpine \(13.0/fpm-alpine/Dockerfile\)](https://github.com/nextcloud/docker/blob/ae849f190066e68156554d908cce5fa3e38fa05f/13.0/fpm-alpine/Dockerfile)
(<https://github.com/nextcloud/docker/blob/ae849f190066e68156554d908cce5fa3e38fa05f/13.0/fpm-alpine/Dockerfile>)
- [13.0.7-fpm , 13.0-fpm , 13-fpm , stable-fpm , production-fpm \(13.0/fpm/Dockerfile\)](https://github.com/nextcloud/docker/blob/ae849f190066e68156554d908cce5fa3e38fa05f/13.0/fpm/Dockerfile)
(<https://github.com/nextcloud/docker/blob/ae849f190066e68156554d908cce5fa3e38fa05f/13.0/fpm/Dockerfile>)
- [14.0.4RC2-apache , 14.0.4-rc-apache , 14.0-rc-apache , 14-rc-apache , rc-apache , 14.0.4RC2 , 14.0.4-rc , 14.0-rc , 14-rc , rc \(14.0-rc/apache/Dockerfile\)](https://github.com/nextcloud/docker/blob/b4c914d530e9f309189eaab912244a15561641f7/14.0-rc/apache/Dockerfile)
(<https://github.com/nextcloud/docker/blob/b4c914d530e9f309189eaab912244a15561641f7/14.0-rc/apache/Dockerfile>)
- [14.0.4RC2-fpm-alpine , 14.0.4-rc-fpm-alpine , 14.0-rc-fpm-alpine , 14-rc-fpm-alpine , rc-fpm-alpine \(14.0-rc/fpm-alpine/Dockerfile\)](https://github.com/nextcloud/docker/blob/b4c914d530e9f309189eaab912244a15561641f7/14.0-rc/fpm-alpine/Dockerfile)
(<https://github.com/nextcloud/docker/blob/b4c914d530e9f309189eaab912244a15561641f7/14.0-rc/fpm-alpine/Dockerfile>)
- [14.0.4RC2-fpm , 14.0.4-rc-fpm , 14.0-rc-fpm , 14-rc-fpm , rc-fpm \(14.0-rc/fpm/Dockerfile\)](https://github.com/nextcloud/docker/blob/b4c914d530e9f309189eaab912244a15561641f7/14.0-rc/fpm/Dockerfile)
(<https://github.com/nextcloud/docker/blob/b4c914d530e9f309189eaab912244a15561641f7/14.0-rc/fpm/Dockerfile>)
- [14.0.3-apache , 14.0-apache , 14-apache , apache , 14.0.3 , 14.0 , 14 , latest \(14.0/apache/Dockerfile\)](https://github.com/nextcloud/docker/blob/ae849f190066e68156554d908cce5fa3e38fa05f/14.0/apache/Dockerfile)
(<https://github.com/nextcloud/docker/blob/ae849f190066e68156554d908cce5fa3e38fa05f/14.0/apache/Dockerfile>)
- [14.0.3-fpm-alpine , 14.0-fpm-alpine , 14-fpm-alpine , fpm-alpine \(14.0/fpm-alpine/Dockerfile\)](https://github.com/nextcloud/docker/blob/ae849f190066e68156554d908cce5fa3e38fa05f/14.0/fpm-alpine/Dockerfile)
(<https://github.com/nextcloud/docker/blob/ae849f190066e68156554d908cce5fa3e38fa05f/14.0/fpm-alpine/Dockerfile>)
- [14.0.3-fpm , 14.0-fpm , 14-fpm , fpm \(14.0/fpm/Dockerfile\)](https://github.com/nextcloud/docker/blob/ae849f190066e68156554d908cce5fa3e38fa05f/14.0/fpm/Dockerfile)
(<https://github.com/nextcloud/docker/blob/ae849f190066e68156554d908cce5fa3e38fa05f/14.0/fpm/Dockerfile>)
- [15.0.0beta2-apache , 15.0.0-beta-apache , 15.0-beta-apache , 15-beta-apache , beta-apache , 15.0.0beta2 , 15.0.0-beta , 15.0-beta , 15-beta , beta \(15.0-beta/apache/Dockerfile\)](https://github.com/nextcloud/docker/blob/ae849f190066e68156554d908cce5fa3e38fa05f/15.0-beta/apache/Dockerfile)
(<https://github.com/nextcloud/docker/blob/ae849f190066e68156554d908cce5fa3e38fa05f/15.0-beta/apache/Dockerfile>)
- [15.0.0beta2-fpm-alpine , 15.0.0-beta-fpm-alpine , 15.0-beta-fpm-alpine , 15-beta-fpm-alpine , beta-fpm-alpine \(15.0-beta/fpm-alpine/Dockerfile\)](https://github.com/nextcloud/docker/blob/ae849f190066e68156554d908cce5fa3e38fa05f/15.0-beta/fpm-alpine/Dockerfile)
(<https://github.com/nextcloud/docker/blob/ae849f190066e68156554d908cce5fa3e38fa05f/15.0-beta/fpm-alpine/Dockerfile>)
- [15.0.0beta2-fpm , 15.0.0-beta-fpm , 15.0-beta-fpm , 15-beta-fpm , beta-fpm \(15.0-beta/fpm/Dockerfile\)](https://github.com/nextcloud/docker/blob/ae849f190066e68156554d908cce5fa3e38fa05f/15.0-beta/fpm/Dockerfile)
(<https://github.com/nextcloud/docker/blob/ae849f190066e68156554d908cce5fa3e38fa05f/15.0-beta/fpm/Dockerfile>)

Quick reference

- **Where to get help:**
[the Docker Community Forums \(https://forums.docker.com/\)](https://forums.docker.com/), [the Docker Community Slack \(https://blog.docker.com/2016/11/introducing-docker-community-directory-docker-community-slack/\)](https://blog.docker.com/2016/11/introducing-docker-community-directory-docker-community-slack/), or [Stack Overflow \(https://stackoverflow.com/search?tab=newest&q=docker\)](https://stackoverflow.com/search?tab=newest&q=docker)
- **Where to file issues:**
<https://github.com/nextcloud/docker/issues> (<https://github.com/nextcloud/docker/issues>)
- **Maintained by:**
[Nextcloud \(https://github.com/nextcloud/docker\)](https://github.com/nextcloud/docker)
- **Supported architectures:** (more info (<https://github.com/docker-library/official-images#architectures-other-than-amd64>))
[amd64 \(https://hub.docker.com/r/amd64/nextcloud/\)](https://hub.docker.com/r/amd64/nextcloud/), [arm32v5 \(https://hub.docker.com/r/arm32v5/nextcloud/\)](https://hub.docker.com/r/arm32v5/nextcloud/), [arm32v6 \(https://hub.docker.com/r/arm32v6/nextcloud/\)](https://hub.docker.com/r/arm32v6/nextcloud/), [arm32v7 \(https://hub.docker.com/r/arm32v7/nextcloud/\)](https://hub.docker.com/r/arm32v7/nextcloud/), [arm64v8 \(https://hub.docker.com/r/arm64v8/nextcloud/\)](https://hub.docker.com/r/arm64v8/nextcloud/), [i386 \(https://hub.docker.com/r/i386/nextcloud/\)](https://hub.docker.com/r/i386/nextcloud/), [ppc64le \(https://hub.docker.com/r/ppc64le/nextcloud/\)](https://hub.docker.com/r/ppc64le/nextcloud/), [s390x \(https://hub.docker.com/r/s390x/nextcloud/\)](https://hub.docker.com/r/s390x/nextcloud/)
- **Published image artifact details:**
[repo-info repo's repos/nextcloud/ directory \(https://github.com/docker-library/repo-info/blob/master/repos/nextcloud\)](https://github.com/docker-library/repo-info/blob/master/repos/nextcloud/) ([history \(https://github.com/docker-library/repo-info/commits/master/repos/nextcloud\)](https://github.com/docker-library/repo-info/commits/master/repos/nextcloud/))
(image metadata, transfer size, etc)
- **Image updates:**
[official-images PRs with label library/nextcloud \(https://github.com/docker-library/official-images/pulls?q=label%3Alibrary%2Fnextcloud\)](https://github.com/docker-library/official-images/pulls?q=label%3Alibrary%2Fnextcloud)
[official-images repo's library/nextcloud file \(https://github.com/docker-library/official-images/blob/master/library/nextcloud\)](https://github.com/docker-library/official-images/blob/master/library/nextcloud) ([history \(https://github.com/docker-library/official-images/commits/master/library/nextcloud\)](https://github.com/docker-library/official-images/commits/master/library/nextcloud/))
- **Source of this description:**
[docs repo's nextcloud/ directory \(https://github.com/docker-library/docs/tree/master/nextcloud\)](https://github.com/docker-library/docs/tree/master/nextcloud)
([history \(https://github.com/docker-library/docs/commits/master/nextcloud\)](https://github.com/docker-library/docs/commits/master/nextcloud/))
- **Supported Docker versions:**
[the latest release \(https://github.com/docker/docker-ce/releases/latest\)](https://github.com/docker/docker-ce/releases/latest) (down to 1.6 on a best-effort basis)

What is Nextcloud?

A safe home for all your data. Access & share your files, calendars, contacts, mail & more from any device, on your terms.

[Nextcloud.com \(https://nextcloud.com/\)](https://nextcloud.com/)



How to use this image

This image is designed to be used in a micro-service environment. There are two versions of the image you can choose from.

The `apache` tag contains a full Nextcloud installation including an apache web server. It is designed to be easy to use and gets you running pretty fast. This is also the default for the `latest` tag and version tags that are not further specified.

The second option is a `fpm` container. It is based on the `php-fpm` (https://hub.docker.com/_/php/) image and runs a fastCGI-Process that serves your Nextcloud page. To use this image it must be combined with any webserver that can proxy the http requests to the FastCGI-port of the container.

Using the apache image

The apache image contains a webserver and exposes port 80. To start the container type:

```
$ docker run -d -p 8080:80 nextcloud
```

Now you can access Nextcloud at <http://localhost:8080/> (<http://localhost:8080/>) from your host system.

Using the fpm image

To use the fpm image you need an additional web server that can proxy http-request to the fpm-port of the container. For fpm connection this container exposes port 9000. In most cases you might want use another container or your host as proxy. If you use your host you can address your Nextcloud container directly on port 9000. If you use another container, make sure that you add them to the same docker network (via `docker run --network <NAME> ...` or a `docker-compose` file). In both cases you don't want to map the fpm port to you host.

```
$ docker run -d nextcloud:fpm
```

As the fastCGI-Process is not capable of serving static files (style sheets, images, ...) the webserver needs access to these files. This can be achieved with the `volumes-from` option. You can find more information in the `docker-compose` section.

Using an external database

By default this container uses SQLite for data storage, but the Nextcloud setup wizard (appears on first run) allows connecting to an existing MySQL/MariaDB or PostgreSQL database. You can also link a database container, e. g. `--link my-mysql:mysql`, and then use `mysql` as the database host on setup. More info is in the `docker-compose` section.

Persistent data

The Nextcloud installation and all data beyond what lives in the database (file uploads, etc) is stored in the unnamed docker volume (<https://docs.docker.com/engine/tutorials/dockervolumes/#adding-a-data-volume>) volume `/var/www/html`. The docker daemon will store that data within the docker directory `/var/lib/docker/volumes/...`. That means your data is saved even if the container crashes, is stopped or deleted.

To make your data persistent to upgrading and get access for backups is using named docker volume or mount a host folder. To achieve this you need one volume for your database container and Nextcloud.

Nextcloud:

- `/var/www/html/` folder where all Nextcloud data lives

```
$ docker run -d \
-v nextcloud:/var/www/html \
nextcloud
```

Database:

- `/var/lib/mysql` MySQL / MariaDB Data
- `/var/lib/postgresql/data` PostgreSQL Data

```
$ docker run -d \
-v db:/var/lib/mysql \
mariadb
```

If you want to get fine grained access to your individual files, you can mount additional volumes for data, config, your theme and custom apps. The `data`, `config` are stored in respective subfolders inside `/var/www/html/`. The apps are split into `core apps` (which are shipped with Nextcloud and you don't need to take care of) and a `custom_apps` folder. If you use a custom theme it would go into the `themes` subfolder.

Overview of the folders that can be mounted as volumes:

- `/var/www/html` Main folder, needed for updating
- `/var/www/html/custom_apps` installed / modified apps
- `/var/www/html/config` local configuration
- `/var/www/html/data` the actual data of your Nextcloud
- `/var/www/html/themes/<YOU_CUSTOM_THEME>` theming/branding

If you want to use named volumes for all of these it would look like this

```
$ docker run -d \
-v nextcloud:/var/www/html \
-v apps:/var/www/html/custom_apps \
-v config:/var/www/html/config \
-v data:/var/www/html/data \
-v theme:/var/www/html/themes/<YOUR_CUSTOM_THEME> \
nextcloud
```

Using the Nextcloud command-line interface

To use the [Nextcloud command-line interface](#)

(https://docs.nextcloud.com/server/12/admin_manual/configuration_server/occ_command.html) (aka. `occ` command):

```
$ docker exec --user www-data CONTAINER_ID php occ
```

or for docker-compose:

```
$ docker-compose exec --user www-data app php occ
```

Auto configuration via environment variables

The nextcloud image supports auto configuration via environment variables. You can preconfigure everything that is asked on the install page on first run. To enable auto configuration, set your database connection via the following environment variables. ONLY use one database type!

SQLITE_DATABASE:

- `SQLITE_DATABASE` Name of the database using sqlite

MYSQL/MariaDB:

- `MYSQL_DATABASE` Name of the database using mysql / mariadb.
- `MYSQL_USER` Username for the database using mysql / mariadb.
- `MYSQL_PASSWORD` Password for the database user using mysql / mariadb.
- `MYSQL_HOST` Hostname of the database server using mysql / mariadb.

PostgreSQL:

- `POSTGRES_DB` Name of the database using postgres.
- `POSTGRES_USER` Username for the database using postgres.
- `POSTGRES_PASSWORD` Password for the database user using postgres.
- `POSTGRES_HOST` Hostname of the database server using postgres.

If you set any values, they will not be asked in the install page on first run. With a complete configuration by using all variables for your database type, you can additionally configure your Nextcloud instance by setting admin user and password (only works if you set both):

- `NEXTCLOUD_ADMIN_USER` Name of the Nextcloud admin user.
- `NEXTCLOUD_ADMIN_PASSWORD` Password for the Nextcloud admin user.

If you want you can set the data directory and table prefix, otherwise default values will be used.

- `NEXTCLOUD_DATA_DIR` (default: `/var/www/html/data`) Configures the data directory where nextcloud stores all files from the users.
- `NEXTCLOUD_TABLE_PREFIX` (default: `""`) Optional prefix for the tables. Used to be `oc_` in the past

Running this image with docker-compose

The easiest way to get a fully featured and functional setup is using a `docker-compose` file. There are too many different possibilities to setup your system, so here are only some examples what you have to look for.

At first make sure you have chosen the right base image (fpm or apache) and added the features you wanted (see below). In every case you want to add a database container and docker volumes to get easy access to your persistent data. When you want to have your server reachable from the internet adding HTTPS-encryption is mandatory! See below for more information.

Base version - apache

This version will use the apache image and add a mariaDB container. The volumes are set to keep your data persistent. This setup provides **no ssl encryption** and is intended to run behind a proxy.

Make sure to set the variables `MYSQL_ROOT_PASSWORD` and `MYSQL_PASSWORD` before you run this setup.

```
version: '2'

volumes:
  nextcloud:
  db:

services:
  db:
    image: mariadb
    restart: always
    volumes:
      - db:/var/lib/mysql
    environment:
      - MYSQL_ROOT_PASSWORD=
      - MYSQL_PASSWORD=
      - MYSQL_DATABASE=nextcloud
      - MYSQL_USER=nextcloud

  app:
    image: nextcloud
    ports:
      - 8080:80
    links:
      - db
    volumes:
      - nextcloud:/var/www/html
    restart: always
```

Then run `docker-compose up -d`, now you can access Nextcloud at <http://localhost:8080/> (<http://localhost:8080/>) from your host system.

Base version - FPM

When using the FPM image you need another container that acts as web server on port 80 and proxies the requests to the Nextcloud container. In this example a simple nginx container is combined with the Nextcloud-fpm image and a MariaDB database container. The data is stored in docker volumes. The nginx container also need access to static files from your Nextcloud installation. It gets access to all the volumes mounted to Nextcloud via the `volumes_from` option. The configuration for nginx is stored in the configuration file `nginx.conf`, that is mounted into the container. An example can be found in the examples section [here \(https://github.com/nextcloud/docker/tree/master/examples\)](https://github.com/nextcloud/docker/tree/master/examples).

As this setup does **not include encryption** it should to be run behind a proxy.

Make sure to set the variables `MYSQL_ROOT_PASSWORD` and `MYSQL_PASSWORD` before you run this setup.

```
version: '2'

volumes:
  nextcloud:
  db:

services:
  db:
    image: mariadb
    restart: always
    volumes:
      - db:/var/lib/mysql
    environment:
      - MYSQL_ROOT_PASSWORD=
      - MYSQL_PASSWORD=
      - MYSQL_DATABASE=nextcloud
      - MYSQL_USER=nextcloud

  app:
    image: nextcloud:fpm
    links:
      - db
    volumes:
      - nextcloud:/var/www/html
    restart: always

  web:
    image: nginx
    ports:
      - 8080:80
    links:
      - app
    volumes:
      - ./nginx.conf:/etc/nginx/nginx.conf:ro
    volumes_from:
      - app
    restart: always
```

Then run `docker-compose up -d`, now you can access Nextcloud at <http://localhost:8080/> (<http://localhost:8080/>) from your host system.

Make your Nextcloud available from the internet

Until here your Nextcloud is just available from you docker host. If you want you Nextcloud available from the internet adding SSL encryption is mandatory.

HTTPS - SSL encryption

There are many different possibilities to introduce encryption depending on your setup.

We recommend using a reverse proxy in front of our Nextcloud installation. Your Nextcloud will only be reachable through the proxy, which encrypts all traffic to the clients. You can mount your manually generated certificates to the proxy or use a fully automated solution, which generates and renews the certificates for you.

In our [examples \(https://github.com/nextcloud/docker/tree/master/.examples\)](https://github.com/nextcloud/docker/tree/master/.examples) section we have an example for a fully automated setup using a reverse proxy, a container for [Let's Encrypt \(https://letsencrypt.org/\)](https://letsencrypt.org/) certificate handling, database and Nextcloud. It uses the popular [nginx-proxy \(https://github.com/jwilder/nginx-proxy\)](https://github.com/jwilder/nginx-proxy) and [docker-letsencrypt-nginx-proxy-companion \(https://github.com/JrCs/docker-letsencrypt-nginx-proxy-companion\)](https://github.com/JrCs/docker-letsencrypt-nginx-proxy-companion) containers. Please check the according documentations before using this setup.

First use

When you first access your Nextcloud, the setup wizard will appear and ask you to choose an administrator account, password and the database connection. For the database use `db` as host and `nextcloud` as table and user name. Also enter the password you chose in your `docker-compose.yml` file.

Update to a newer version

Updating the Nextcloud container is done by pulling the new image, throwing away the old container and starting the new one. Since all data is stored in volumes, nothing gets lost. The startup script will check for the version in your volume and the installed docker version. If it finds a mismatch, it automatically starts the upgrade process. Don't forget to add all the volumes to your new container, so it works as expected.

```
$ docker pull nextcloud
$ docker stop <your_nextcloud_container>
$ docker rm <your_nextcloud_container>
$ docker run <OPTIONS> -d nextcloud
```

Beware that you have to run the same command with the options that you used to initially start your Nextcloud. That includes volumes, port mapping.

When using docker-compose your compose file takes care of your configuration, so you just have to run:

```
$ docker-compose pull
$ docker-compose up -d
```

Adding Features

A lot of people want to use additional functionality inside their Nextcloud installation. If the image does not include the packages you need, you can easily build your own image on top of it. Start your derived image with the `FROM` statement and add whatever you like.

```
FROM nextcloud:apache

RUN ...
```

The [examples folder \(https://github.com/nextcloud/docker/blob/master/.examples\)](https://github.com/nextcloud/docker/blob/master/.examples) gives a few examples on how to add certain functionalities, like including the cron job, smb-support or imap-authentication.

If you use your own Dockerfile you need to configure your docker-compose file accordingly. Switch out the `image` option with `build`. You have to specify the path to your Dockerfile. (in the example it's in the same directory next to the docker-compose file)

```
app:
  build: .
  links:
    - db
  volumes:
    - data:/var/www/html/data
    - config:/var/www/html/config
    - apps:/var/www/html/apps
  restart: always
```

Updating your own derived image is also very simple. When a new version of the Nextcloud image is available run:

```
docker build -t your-name --pull .
docker run -d your-name
```

or for docker-compose:

```
docker-compose build --pull
docker-compose up -d
```

The `--pull` option tells docker to look for new versions of the base image. Then the build instructions inside your `Dockerfile` are run on top of the new image.

Migrating an existing installation

You're already using Nextcloud and want to switch to docker? Great! Here are some things to look out for:

1. Define your whole Nextcloud infrastructure in a `docker-compose` file and run it with `docker-compose up -d` to get the base installation, volumes and database. Work from there.
2. Restore your database from a mysqldump (nextcloud_db_1 is the name of your db container)

```
docker cp ./database.dmp nextcloud_db_1:/dmp
docker-compose exec db sh -c "mysql -u USER -pPASSWORD nextcloud < /dmp"
docker-compose exec db rm /dmp
```

3. Edit your config.php

1. Set database connection

```
'dbhost' => 'db:3306',
```

2. Make sure you have no configuration for the `apps_paths`. Delete lines like these

```
"apps_paths" => array (
0 => array (
    "path" => OC::$SERVERROOT."/apps",
    "url" => "/apps",
    "writable" => true,
),
),
```

3. Make sure your data directory is set to `/var/www/html/data`

```
'datadirectory' => '/var/www/html/data',
```

4. Copy your data (nextcloud_app_1 is the name of your Nextcloud container):

```
docker cp ./data/ nextcloud_app_1:/var/www/html/data
docker-compose exec app chown -R www-data:www-data /var/www/html/data
docker cp ./theming/ nextcloud_app_1:/var/www/html/theming
docker-compose exec app chown -R www-data:www-data /var/www/html/theming
docker cp ./config/config.php nextcloud_app_1:/var/www/html/config
docker-compose exec app chown -R www-data:www-data /var/www/html/config
```

5. Copy only the custom apps you use (or simply redownload them from the web interface):

```
docker cp ./apps/ nextcloud_data:/var/www/html/custom_apps
docker-compose exec app chown -R www-data:www-data /var/www/html/custom_apps
```

Questions / Issues

If you got any questions or problems using the image, please visit our [Github Repository](https://github.com/nextcloud/docker) (<https://github.com/nextcloud/docker>) and write an issue.

Image Variants

The `nextcloud` images come in many flavors, each designed for a specific use case.

`nextcloud:<version>`

This is the defacto image. If you are unsure about what your needs are, you probably want to use this one. It is designed to be used both as a throw away container (mount your source code and start the container to start your app), as well as the base to build other images off of.

`nextcloud:<version>-alpine`

This image is based on the popular [Alpine Linux project](http://alpinelinux.org) (<http://alpinelinux.org>), available in the [alpine official image](https://hub.docker.com/_/alpine) (https://hub.docker.com/_/alpine). Alpine Linux is much smaller than most distribution base images (~5MB), and thus leads to much slimmer images in general.

This variant is highly recommended when final image size being as small as possible is desired. The main caveat to note is that it does use [musl libc](http://www.musl-libc.org) (<http://www.musl-libc.org>) instead of [glibc and friends](http://www.etalabs.net/compare_libcs.html) (http://www.etalabs.net/compare_libcs.html), so certain software might run into issues depending on the depth of their libc requirements. However, most software doesn't have an issue with this, so this variant is usually a very safe choice. See [this Hacker News comment thread](https://news.ycombinator.com/item?id=10782897) (<https://news.ycombinator.com/item?id=10782897>) for more discussion of the issues that might arise and some pro/con comparisons of using Alpine-based images.

To minimize image size, it's uncommon for additional related tools (such as `git` or `bash`) to be included in Alpine-based images. Using this image as a base, add the things you need in your own Dockerfile (see the [alpine image description](https://hub.docker.com/_/alpine) (https://hub.docker.com/_/alpine) for examples of how to install packages if you are unfamiliar).

License

View [license information](https://github.com/nextcloud/server/blob/master/COPYING-README) (<https://github.com/nextcloud/server/blob/master/COPYING-README>) for the software contained in this image.

As with all Docker images, these likely also contain other software which may be under other licenses (such as Bash, etc from the base distribution, alo

Docker Pull Command



```
docker pull nextcloud
```