

Setup & Configuration

We need to install the service through Portainer and configure any necessary settings.

- [Preparation](#)
- [Installation](#)
- [Updating](#)

Preparation

There are some things we need to do in preparation to install this service.

Volumes

Persistent Data

This is where the service will store its own application data and ensures we can quickly update the service image.

Ensure your user has permissions to access the folder.

Media & Documents Folders

OwnCloud will need access to the folders where you store your media and document files. This will allow you to share them through the OwnCloud files and sharing app.

Ensure your user has permissions to access the folder.

Environment

General

TZ

This is the current time zone formatted using the [tz database](#).

For example: America/Vancouver

PUID

This is the numeric ID of the user account on Debian. If you are unsure, open a terminal and run:

```
id -u
```

PGID

This is the numeric ID of the user account's group on Debian. If you are unsure, open a terminal and run:

```
id -g
```

OwnCloud

OWNCLOUD_TRUSTED_DOMAINS

This is the full domain and subdomain name that will be used to access this instance from the internet, as well as the server's local IP address if you want OwnCloud to be accessible locally. They must be comma separated.

For example: 192.168.68.100, owncloud.example.com

OWNCLOUD_DOMAIN

This is the full domain and subdomain name that will be used to access this instance from the internet.

For example: owncloud.example.com

OWNCLOUD_ADMIN_USERNAME

This is the username for the administrator account that will be used to log-in through the web interface.

This should be an easy-to-remember user name.

Database

DB_USERNAME

This is the username to be used for logging into the database.

For example: owncloud_admin

Passwords

Keep these securely stored in a password manager, such as [VaultWarden](#).

DB_ROOT_PASS

This is the password that will be used for root access to the database.

It is important to use secure, randomly generated password.

You can use a random alphanumeric string from a password manager, or open the terminal and run the command:

```
tr -dc 'A-Za-z0-9!"#$%&'\'()*+,-./:;<=>?@[\\]^_`{|}~' </dev/urandom | head -c 32; echo
```

This pulls a random string from the '[urandom](#)' device, removes unwanted characters and trim it to an appropriate length.

DB_USER_PASS

This is the password that will be used for user access to the database.

It is important to use secure, randomly generated password.

You can use a random alphanumeric string from a password manager, or open the terminal and run the command:

```
tr -dc 'A-Za-z0-9!"#$%&'\'()*+,-./:;<=>?@[\]^_`{|}~' </dev/urandom | head -c 32; echo
```

This pulls a random string from the '[urandom](#)' device, removes unwanted characters and trim it to an appropriate length.

OWNCLOUD_ADMIN_PASSWORD

This is the password for the administrator account that will be used within the web interface.

It is important to use secure passphrase that is easy-to-remember.

Installation

The service can be installed through the Portainer web interface.

Learn about [creating a new stack](#).

Docker Compose

Use the following code to install the service:

```
---
services:
  owncloud:
    image: owncloud/server:latest
    container_name: owncloud
    restart: always
    ports:
      - ${HTTP_PORT}:8080
    depends_on:
      - mariadb
      - redis
    environment:
      - PGID=${PGID}
      - PUID=${PUID}
      - OWNCLOUD_DOMAIN=${OWNCLOUD_DOMAIN}
      - OWNCLOUD_TRUSTED_DOMAINS=${OWNCLOUD_TRUSTED_DOMAINS}
      - OWNCLOUD_DB_TYPE=${DB_TYPE}
      - OWNCLOUD_DB_NAME=${DB_NAME}
      - OWNCLOUD_DB_USERNAME=${DB_USERNAME}
      - OWNCLOUD_DB_PASSWORD=${DB_USER_PASS}
      - OWNCLOUD_DB_HOST=${DB_HOST}
      - OWNCLOUD_ADMIN_USERNAME=${ADMIN_USERNAME}
      - OWNCLOUD_ADMIN_PASSWORD=${ADMIN_PASSWORD}
      - OWNCLOUD_MYSQL_UTF8MB4=true
```

- OWNCLOUD_REDIS_ENABLED=true
- OWNCLOUD_REDIS_HOST=\${REDIS_HOST}

healthcheck:

```
test: ["CMD", "/usr/bin/healthcheck"]
interval: 30s
timeout: 10s
retries: 5
```

volumes:

- # Persistent Data
- /srv/owncloud/config:/mnt/data
-
- # Media Data
- /mnt/libraries:/mnt/libraries

OwnCloud Database

mariadb:

```
image: mariadb:10.6
container_name: owncloud_mariadb
restart: always
command: ["--max-allowed-packet=128M", "--innodb-log-file-size=64M"]
environment:
  - MYSQL_ROOT_PASSWORD=${DB_ROOT_PASS}
  - MYSQL_USER=${DB_USERNAME}
  - MYSQL_PASSWORD=${DB_USER_PASS}
  - MYSQL_DATABASE=${DB_NAME}
```

healthcheck:

```
test: ["CMD", "mysqladmin", "ping", "-u", "root", "--password=owncloud"]
interval: 10s
timeout: 5s
retries: 5
```

volumes:

- # Persistent Data
- /srv/owncloud/db:/var/lib/mysql

Cache Server

redis:

```
image: redis:6
container_name: owncloud_redis
restart: always
command: ["--databases", "1"]
```

```
healthcheck:
  test: ["CMD", "redis-cli", "ping"]
  interval: 10s
  timeout: 5s
  retries: 5
volumes:
  # Persistent Data
  - /srv/owncloud/redis:/data
```

Environment

Use the following environment to configure the service using the values prepared earlier:

```
# Set Run User
PGID=1000
PUID=1000

# OwnCloud Networking
OWNCLOUD_VERSION=latest
HTTP_PORT=8080
OWNCLOUD_TRUSTED_DOMAINS=192.168.68.100, owncloud.example.com
OWNCLOUD_DOMAIN=owncloud.example.com

# Cache Server
REDIS_HOST=redis

# Database Configuration
DB_HOST=mariadb
DB_TYPE=mysql
DB_NAME=owncloud
DB_USERNAME=owncloud_admin

# Passwords
# After the initial setup, the passwords should be saved elsewhere and stack should be re-
# deployed with these settings deleted.

# Database Passwords
```

```
DB_ROOT_PASS=
```

```
DB_USER_PASS=
```

```
# OwnCloud Admin Account
```

```
OWNCLOUD_ADMIN_USERNAME=
```

```
OWNCLOUD_ADMIN_PASSWORD=
```

Updating

Backup Data and Re-Deploy the Stack

OwnCloud has been optimized for running in Docker and will automatically migrate your database when a new version is installed.

This allows you to [re-deploy the stack through Portainer](#) to download the latest updates.