

# Installation of Nextcloud

- [Setting Up Nextcloud with Docker Containers](#)
  - [Setting Up Nextcloud with Docker Containers](#)
  - [Securing Nextcloud with Apache and SSL Using Docker Containers](#)

# Setting Up Nextcloud with Docker Containers

# Setting Up Nextcloud with Docker Containers

Nextcloud is a powerful and open-source file hosting service that allows you to store and sync your files, contacts, calendars, and more. One of the easiest ways to deploy Nextcloud is by using Docker containers. In this tutorial, we'll guide you through the process of setting up Nextcloud with Docker.

## Prerequisites

Before you begin, ensure that you have the following prerequisites installed on your machine:

- Docker
- Docker Compose

## Step 1: Create a Docker Compose File

Start by creating a `docker-compose.yml` file in a directory of your choice. Use a text editor to paste the following basic configuration:

```
version: '3'

services:
  nextcloud:
    image: nextcloud
    ports:
      - 8080:80
    volumes:
      - nextcloud:/var/www/html
    environment:
      - MYSQL_HOST=nextcloud-db
      - MYSQL_DATABASE=nextcloud
      - MYSQL_USER=nextcloud
      - MYSQL_PASSWORD=your_mysql_password
    depends_on:
      - nextcloud-db
```

```
nextcloud-db:
  image: mariadb
  environment:
    - MYSQL_ROOT_PASSWORD=your_mysql_root_password
    - MYSQL_DATABASE=nextcloud
    - MYSQL_USER=nextcloud
    - MYSQL_PASSWORD=your_mysql_password
  volumes:
    - nextcloud-db: /var/lib/mysql
```

```
volumes:
  nextcloud:
  nextcloud-db:
```

Replace `your_mysql_password` and `your_mysql_root_password` with secure passwords.

## Step 2: Start Docker Containers

Open a terminal, navigate to the directory with your `docker-compose.yml` file, and run the following command:

```
bash Copy code
docker-compose up -d
```

This command downloads the required images and starts the Nextcloud and MariaDB containers in the background.

## Step 3: Access Nextcloud Setup

Open your web browser and go to `http://localhost:8080`. You'll be directed to the Nextcloud setup page. Complete the installation by providing the necessary information, including database details.

## Step 4: Finish Installation

Complete the setup process by creating an admin account and configuring additional settings as needed.

## Step 5: Access Nextcloud

After the setup is complete, you can access Nextcloud by visiting `http://localhost:8080` in your web browser.

Congratulations! You've successfully installed Nextcloud using Docker containers. For more detailed configuration options or updates, refer to the official Nextcloud documentation.

# Securing Nextcloud with Apache and SSL Using Docker Containers

## Prerequisites:

1. A domain name pointing to your server's IP address.
2. Docker installed on your machine.
3. Docker Compose installed on your machine.
4. A valid SSL certificate for your domain.

## Step 1: Obtain an SSL Certificate

Ensure you have an SSL certificate and its corresponding private key for your domain. You can obtain a free certificate from Let's Encrypt using certbot or any other SSL certificate provider.

## Step 2: Create Docker Compose File

Create a `docker-compose.yml` file with the following modifications:

```
version: '3'

services:
  nextcloud:
    image: nextcloud
    ports:
      - 8080: 80
    volumes:
      - nextcloud: /var/www/html
    environment:
      - MYSQL_HOST=nextcloud-db
      - MYSQL_DATABASE=nextcloud
      - MYSQL_USER=nextcloud
      - MYSQL_PASSWORD=your_mysql_password
    depends_on:
```

```
- nextcloud-db
```

```
nextcloud-db:
```

```
image: mariadb
```

```
environment:
```

- MYSQL\_ROOT\_PASSWORD=your\_mysql\_root\_password
- MYSQL\_DATABASE=nextcloud
- MYSQL\_USER=nextcloud
- MYSQL\_PASSWORD=your\_mysql\_password

```
volumes:
```

- nextcloud-db: /var/lib/mysql

```
web:
```

```
image: httpd:2.4
```

```
ports:
```

- 443:443

```
volumes:
```

- ./apache-config: /usr/local/apache2/conf
- nextcloud: /var/www/html

```
volumes:
```

```
nextcloud:
```

```
nextcloud-db:
```

## Step 3: Create Apache Configuration

Create a directory named `apache-config` in the same directory as your `docker-compose.yml` file. Inside this directory, create a file named `httpd.conf` with the following configuration:

```
LoadModule proxy_module modules/mod_proxy.so
LoadModule proxy_http_module modules/mod_proxy_http.so
LoadModule ssl_module modules/mod_ssl.so

<VirtualHost *:443>
    ServerName your-domain.com

    SSLEngine on
    SSLCertificateFile /usr/local/apache2/conf/cert.crt
    SSLCertificateKeyFile /usr/local/apache2/conf/cert.key
```

```
<Location />
    ProxyPass http://nextcloud:80/
    ProxyPassReverse http://nextcloud:80/
</Location>
</VirtualHost>
```

Replace `your-domain.com` with your actual domain, and replace `/usr/local/apache2/conf/cert.crt` and `/usr/local/apache2/conf/cert.key` with the paths to your SSL certificate and private key.

## Step 4: Start Docker Containers

Run the following command in the terminal to start the Docker containers:

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

## Step 5: Access Nextcloud over HTTPS

Visit `https://your-domain.com` in your web browser. You should now be able to access Nextcloud securely over HTTPS.

Note: Ensure that your firewall allows traffic on port 443, and adjust security groups if you are using a cloud provider.

This setup creates an Apache container as a reverse proxy, forwarding requests to the Nextcloud container while handling SSL termination. Adjust the configurations according to your specific needs and security considerations.