

# 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.

---

Revision #2

Created 2 February 2024 09:00:47 by Admin

Updated 2 February 2024 09:47:58 by Admin