Nextcloud und Onlyoffice mit Docker.

Installing Document Server for Docker on a local server

Document Serverv.5.4 Document Server changelog

- local server
- server version
- install server solution
- Document Server
- Docker version

Introduction

Document Server is an online office suite comprising viewers and editors for texts, spreadsheets and presentations, fully compatible with Office Open XML formats: .docx, .xlsx, .pptx and enabling collaborative editing in real time.

Functionality

- Document Editor
- Spreadsheet Editor
- Presentation Editor
- Documents application for iOS and Android
- Collaborative editing
- Hieroglyph support
- Support for all the popular formats: DOC, DOCX, TXT, ODT, RTF, ODP, EPUB, ODS, XLS, XLSX, CSV, PPTX, HTML

Integrating it with **Community Server** you will be able to:

- view and edit files stored on Drive, Box, Dropbox, OneDrive, OwnCloud connected to ONLYOFFICE;
- share files;
- embed documents on a website;
- manage access rights to documents.

This guide will show you how to install **Document Server** Docker version to your machine.

System requirements

- CPU

dual core **2 GHz** or better

- RAM

  2 GB or more

- HDD

  at least **40 GB** of free space

- Additional requirements

  at least **4 GB** of swap

- OS

  amd64 Linux distribution with kernel version **3.10** or later

- Additional requirements

  - **Docker**: version **1.10** or later

Installing Document Server

You need the latest Docker version installed. If you do not have it, please see the Installation section on Docker website to learn how to get it.

Docker specifies Google DNS servers by default. If your **Document Server** is not going to have access to the Internet, we recommend that you change the default Docker DNS address to the address of your local DNS server. To do that go to the /etc/default/docker file and change the IP address in the following line to the IP address of a DNS server in your local network:

```
docker_OPTS="--dns 8.8.8.8"
```

After you have Docker installed, run it and execute the following command:

```
sudo docker run -i -t -d -p 80:80 --restart=always onlyoffice/documentserver
```

Use this command if you wish to install **Document Server** separately. To install **Document Server** integrated with **Community** and **Mail Servers**, refer to the corresponding instructions below.

By default **Document Server** listens to the incoming connections using port **80**. Starting with version **4.3** you can change the port for **Document Server** if you plan to use it instead of the default one. This can be easily done changing the mapping port during the installation:

```
sudo docker run -i -t -d -p <PORT_NUMBER>:80 --restart=
always onlyoffice/documentserver
```

Where is the number of port you want **Document Server** to use.

This will install **Document Server** and all the dependencies it needs (the list of what is being done by the script can be found here).

Storing data outside containers

All the data are stored in the specially-designated directories, **data volumes**, at the following location:

- /var/log/onlyoffice for **Document Server** logs
- /var/www/onlyoffice/Data for certificates
- /var/lib/onlyoffice for file cache
- /var/lib/postgresql for database

We recommend We strongly recommend that you store the data outside the Docker containers on the host machine as it allows you to easily update **Document Server** once the new version is released without losing your data.

To get access to your data located outside the container, you need to mount the volumes. It can be done by specifying the **-v** option in the docker run command.

```
sudo docker run -i -t -d -p 80:80 --restart=always \
    -v /app/onlyoffice/DocumentServer/logs:/var/log/onlyoffice  \
    -v /app/onlyoffice/DocumentServer/data:/var/www/onlyoffice/Data  \
    -v /app/onlyoffice/DocumentServer/lib:/var/lib/onlyoffice \
    -v /app/onlyoffice/DocumentServer/db:/var/lib/
postgresql  onlyoffice/documentserver
```

Please note, that in case you are trying to mount the folders which are not yet created, these folders will be created but the access to them will be limited. You will need to change their access rights manually.

Normally, you do not need to store container data because the container operation does not depend on its state. Saving data will be useful:

- for easy access to container data, such as logs;
- to remove the limit on the size of the data inside the container;
- when using services launched outside the container such as PostgreSQL, Redis, RabbitMQ.

Running Document Server using HTTPS

```
sudo docker run -i -t -d -p 443:443 --restart=always \
-v /app/onlyoffice/DocumentServer/data:/var/www/onlyoffice/Data
  onlyoffice/documentserver
```

Access to the onlyoffice application can be secured using SSL so as to prevent unauthorized access. While a CA certified SSL certificate allows for verification of trust via the CA, self signed certificates can also provide an equal level of trust verification as long as each client takes some additional steps to verify the identity of your website. Below the instructions on achieving this are provided.

To secure the application via SSL basically two things are needed:

- **Private key (.key)**
- **SSL certificate (.crt)**

So you need to create and install the following files:

```
/app/onlyoffice/DocumentServer/data/certs/onlyoffice.key
/app/onlyoffice/DocumentServer/data/certs/onlyoffice.crt
```

When using CA certified certificates, these files are provided to you by the CA. When using self-signed certificates you need to generate these files yourself. Skip the following section if you are have CA certified SSL certificates.

Generation of self signed certificates

Generation of self-signed SSL certificates involves a simple 3 step procedure

STEP 1: Create the server private key

```
openssl genrsa -out onlyoffice.key 2048
```

STEP 2: Create the certificate signing request (CSR)

```
openssl req -new -key onlyoffice.key -out onlyoffice.csr
```

STEP 3: Sign the certificate using the private key and CSR

```
openssl x509 -req -days 365 -in onlyoffice.csr -signkey onlyoffice.key -out
 onlyoffice.crt
```

You have now generated an SSL certificate that's valid for 365 days.

Strengthening the server security

This section provides you with instructions to strengthen your server security.

To achieve this you need to generate stronger DHE parameters.

```
openssl dhparam -out dhparam.pem 2048
```

Installation of the SSL certificates

Out of the four files generated above, you need to install the onlyoffice.key, onlyoffice.crt and dhparam.pem files at the onlyoffice server. The CSR file is not needed, but do make sure you safely backup the file (in case you ever need it again).

The default path that the onlyoffice application is configured to look for the SSL certificates is at /var/www/onlyoffice/Data/certs, this can however be changed using the SSL_KEY_PATH, SSL_CERTIFICATE_PATH and SSL_DHPARAM_PATH configuration options.

The /var/www/onlyoffice/Data/ path is the path of the data store, which means that you have to create a folder named certs inside /app/onlyoffice/DocumentServer/data/ and copy the files into it and as a measure of security you will update the permission on the onlyoffice.key file to only be readable by the owner.

```
mkdir -p /app/onlyoffice/DocumentServer/data/certs
cp onlyoffice.key /app/onlyoffice/DocumentServer/data/certs/
cp onlyoffice.crt /app/onlyoffice/DocumentServer/data/certs/
cp dhparam.pem /app/onlyoffice/DocumentServer/data/certs/
chmod 400 /app/onlyoffice/DocumentServer/data/certs/onlyoffice.key
```

And restart Docker container:

```
sudo docker restart {{DOCUMENT_SERVER_ID}}
```

You are now just one step away from having our application secured.

Available configuration parameters

Please refer the docker run command options for the --env-file flag where you can specify all required environment variables in a single file. This will save you from writing a potentially long docker run command.

Below is the complete list of parameters that can be set using environment variables.

- **ONLYOFFICE_HTTPS_HSTS_ENABLED**: Advanced configuration option for turning off the HSTS configuration. Applicable only when SSL is in use. Defaults to true.
- **ONLYOFFICE_HTTPS_HSTS_MAXAGE**: Advanced configuration option for setting the HSTS max-age in the onlyoffice NGINX vHost configuration. Applicable only when SSL is in use. Defaults to 31536000.
- **SSL_CERTIFICATE_PATH**: The path to the SSL certificate to use. Defaults to /var/www/onlyoffice/Data/certs/onlyoffice.crt.
- **SSL_KEY_PATH**: The path to the SSL certificate private key. Defaults to /var/www/onlyoffice/Data/certs/onlyoffice.key.
- **SSL_DHPARAM_PATH**: The path to the Diffie-Hellman parameter. Defaults to /var/www/onlyoffice/Data/certs/dhparam.pem.
- **SSL_VERIFY_CLIENT**: Enable verification of client certificates using the CA_CERTIFICATES_PATH file. Defaults to false.
- **POSTGRESQL_SERVER_HOST**: The IP address or the name of the host where the PostgreSQL server is running.
- **POSTGRESQL_SERVER_PORT**: The PostgreSQL server port number.

- **POSTGRESQL_SERVER_DB_NAME**: The name of a PostgreSQL database to be created on the image startup.
- **POSTGRESQL_SERVER_USER**: The new user name with superuser permissions for the PostgreSQL account.
- **POSTGRESQL_SERVER_PASS**: The password set for the PostgreSQL account.
- **AMQP_SERVER_URL**: The AMQP URL to connect to the message broker server.
- **AMQP_SERVER_TYPE**: The message broker type. Supported values are rabbitmq or activemq. Defaults to rabbitmq.
- **REDIS_SERVER_HOST**: The IP address or the name of the host where the Redis server is running.
- **REDIS_SERVER_PORT**: The Redis server port number.
- **NGINX_WORKER_PROCESSES**: Defines the number of NGINX worker processes.
- **NGINX_WORKER_CONNECTIONS**: Sets the maximum number of simultaneous connections that can be opened by a NGINX worker process.
- **JWT_ENABLED**: Specifies the enabling the JSON web token validation by **Document Server**. Defaults to false.
- **JWT_SECRET**: Defines the secret key to validate the JSON web token in the request to **Document Server**. Defaults to secret.
- **JWT_HEADER**: Defines the HTTP header that will be used to send the JSON web token. Defaults to Authorization.

Installing Document Server integrated with Community and Mail Servers

Document Server is a part of **ONLYOFFICE Community Edition** that comprises also **Community Server** and **Mail Server**. In case you want to install all of them and integrate with each other, read these instructions.

Alternative ways to install Document Server

Alternatively, you can use an automatic installation script to install the whole **Community Edition** at once. For the mail server correct work you need to specify its hostname yourdomain.com.

STEP 1: Download the installation script file

Assuming you have docker-compose installed, execute the following command:

```
wget https://download.onlyoffice.com/install/opensource-install.sh
```

STEP 2: Install Community Edition

Run the following command to do that:

```
bash opensource-install.sh -md yourdomain.com
```

Or you can use docker-compose to install **Document Server**. See the instructions here on how to do that.

Source: https://helpcenter.onlyoffice.com/server/docker/document/docker-installation.aspx

Eindeutige ID: #1093
Verfasser: n/a
Letzte Änderung: 2025-07-30 14:37