Instalação e Configuração OnlyOffice DocSpace

Link: https://helpcenter.onlyoffice.com/docspace/installation/docspace-community-running-docker.aspx

Running ONLYOFFICE DocSpace in Docker

Introduction

ONLYOFFICE DocSpace is a document hub where you can connect users and documents in one place to boost collaboration.

This guide will show you how to run **ONLYOFFICE DocSpace** on your machine using **Docker**.

This guide deploys a development/testing build of **ONLYOFFICE DocSpace** and is **not intended for production use**. This build serves over plain HTTP only, ships without identity/OAuth services, and runs a reduced feature set. For production deployments, use the production version of ONLYOFFICE DocSpace.

Overview

Unlike the standard multi-container deployment, this build runs all **ONLYOFFICE DocSpace** services inside a single container instead of as separate per-service containers. The full stack consists of the following containers:

Container	Role
onlyoffice-docspace	All **ONLYOFFICE DocSpace** services (consolidated)
onlyoffice-document-server	Document Server (editors)
onlyoffice-mysql-server	MySQL database
onlyoffice-opensearch	OpenSearch

Compared to the standard multi-container deployment, this build has the following differences:

- All **ONLYOFFICE DocSpace** services are consolidated into a single container.
- Simplified search.
- No thumbnail generation.
- HTTP only, no TLS/HTTPS support out of the box.
- No identity/OAuth services.

System requirements

- **CPU**: at least **6-core** (**8-core** recommended)
- **RAM**: at least **12 GB** (**16 GB** recommended)
- **HDD**: at least **40 GB** of free space
- **OS**: **amd64** Linux distribution with kernel version **3.10** or later
- **Additional requirements**:
  - at least **6 GB** of swap
  - **Docker Engine** with the **Compose** plugin (`docker compose` command)

Cloning the repository

Clone the `DocSpace-buildtools` repository using the following command:

```
git clone https://github.com/ONLYOFFICE/DocSpace-buildtools.git
```

Once the repository is cloned, change into the Compose directory:

```
cd DocSpace-buildtools/install/docker/community
```

Running prebuilt images

To start **ONLYOFFICE DocSpace** in detached mode using the prebuilt images, run the following command:

```
docker compose up -d
```

Once the containers are up and running, open a browser and enter **http://localhost** or **http://your-ip-address** in the address bar to access **ONLYOFFICE DocSpace**.

Building images from source

Use this option if you want to build the **ONLYOFFICE DocSpace** images yourself or test changes from a specific branch.

Build and start the containers using the following command:

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

By default, the images are built from the **master** branch. To build from another branch, specify the `GIT_BRANCH` build argument: `GIT_BRANCH=your-branch docker compose up -d --build`.

Once the build is complete and the containers are running, open **http://localhost** or **http://your-ip-address** in your browser to access **ONLYOFFICE DocSpace**.

Running with SSL

ONLYOFFICE DocSpace supports both **Let's Encrypt** and custom SSL certificates.

Using a Let's Encrypt certificate

To run **ONLYOFFICE DocSpace** with a **Let's Encrypt** certificate, run the following command, specifying your own domain, email, and portal URL:

```
SSL_MODE="letsencrypt" \
SSL_DOMAIN="example.com,portal.example.com,api.example.com" \
SSL_EMAIL="admin@example.com" \
APP_URL_PORTAL="https://example.com/" \
docker compose \
  -f docker-compose.yml \
  -f ssl.yml \
  up -d
```

The following parameters are used:

- **SSL_MODE** – the SSL certificate mode.
- **SSL_DOMAIN** – one or more domains separated by commas.
- **SSL_EMAIL** – the email address used for the **Let's Encrypt** registration.
- **APP_URL_PORTAL** – the public HTTPS URL of your portal.

Using a custom or private CA certificate

If you are using a self-signed certificate or a certificate issued by a private Certificate Authority (CA), the **ONLYOFFICE Document Server** must also trust this certificate.

Uncomment the following volume in the `docker-compose.yml` file:

```
services:
   onlyoffice-document-server:
     volumes:
       - ${CERTIFICATE_PATH}:/var/www/onlyoffice/Data/certs/extra-ca-certs.pem
```

Then specify the certificate path on the host and run the following command:

```
SSL_MODE="custom" \
SSL_DOMAIN="example.com" \
SSL_CERT_PATH="/etc/nginx/certs/fullchain.crt" \
SSL_KEY_PATH="/etc/nginx/certs/private.key" \
CERTIFICATE_PATH="./config/nginx/certs/fullchain.crt" \
APP_URL_PORTAL="https://example.com/" \
docker compose \
  -f docker-compose.yml \
  -f ssl.yml \
  up -d
```

`CERTIFICATE_PATH` must point to the certificate file **on the Docker host**, not to the path inside the container. This option is typically required only for self-signed certificates or certificates issued by a private CA. Certificates issued by public CAs (for example, **Let's Encrypt**, **DigiCert**, or **GoDaddy**) usually do not require this additional configuration.

The following parameters are used:

- **SSL_MODE** – the SSL certificate mode (**custom**).
- **SSL_DOMAIN** – the portal domain name.
- **SSL_CERT_PATH** – the path to the SSL certificate.
- **SSL_KEY_PATH** – the path to the private key.
- **APP_URL_PORTAL** – the public HTTPS URL of your portal.

By default, the `ssl.yml` configuration mounts the local `./config/nginx/certs` directory to `/etc/nginx/certs` inside the container.

Once the containers are up and running, open your portal address, for example **https://example.com/**, in a browser to access **ONLYOFFICE DocSpace**.

Host ONLYOFFICE DocSpace on your own server or use it in the cloud

Get it nowUse in the cloud

Revision #1
Created 2026-10-02 11:39:03 UTC by Administrador
Updated 2026-10-02 11:41:20 UTC by Administrador