OnlyOffice

Aplicativo de edição e colaboração

- OnlyOffice Workspace
  - Instalação e Configurações OnlyOffice Workspace
  - Instalação e Configurações OnlyOffice Workspace Github
- OnlyOffice DocSpace
  - Instalação e Configuração OnlyOffice DocSpace
- OnlyOffice Apps
  - Instalação e Configuração OnlyOffice Apps
- OnlyOffice Community Server
  - Instalação e Configuração OnlyOffice Community Server

OnlyOffice Workspace

Aplicativos de colaboração

Instalação e Configurações OnlyOffice Workspace

Link: https://helpcenter.onlyoffice.com/workspace/installation/workspace-install-docker-compose.aspx

Em 02/04/2026

Installing ONLYOFFICE Workspace Community using Docker Compose

Introduction

If you need to quickly install the **ONLYOFFICE Workspace Community**, then the installation via Docker Compose will suit you. Please, follow the steps below to launch all components of **ONLYOFFICE Workspace Community** at once.

Step 1. Install Docker Compose

Visit the official Docker Compose page and follow the installation instructions.

Step 2. Clone the GitHub repository

After Docker Compose has been installed, need to clone the GitHub repository:

```
git clone https://github.com/ONLYOFFICE/Docker-CommunityServer
```

Step 3. Install ONLYOFFICE Workspace Community

Then you need to switch to the repository folder:

```
cd Docker-CommunityServer
```

For the mail server correct work, open one of the files depending on the product you use:

- docker-compose.groups.yml file for Community Server (distributed as ONLYOFFICE Groups)
- docker-compose.workspace.yml file for ONLYOFFICE Workspace Community
- docker-compose.workspace_enterprise.yml file for ONLYOFFICE Workspace Enterprise

For working on Ubuntu 22.04 and Debian 11 or later, you need to use docker-compose versions v2.16.0 or later and uncomment the cgroup line in the yml file.

Then replace the `${MAIL_SERVER_HOSTNAME}` variable with your own hostname for the **Mail Server**. After that, assuming you have docker-compose installed, execute the following command specifying the necessary yml file:

```
docker-compose -f docker-compose.workspace.yml up -d
```

To find out more options of **ONLYOFFICE Workspace Community** installation using Docker, visit this page.

Host ONLYOFFICE Workspace on your own server

Download

Instalação e Configurações OnlyOffice Workspace Github

Link: https://github.com/ONLYOFFICE/Docker-CommunityServer

git clone https://github.com/ONLYOFFICE/Docker-CommunityServer.git

Última versão do OnlyOffice Workspace v4.17.0.1936 publicada em 30/10/2024

https://www.onlyoffice.com/

Website | ONLYOFFICE Workspace | Documentation | API | About

https://www.facebook.com/ONLYOFFICE-833032526736775/ https://twitter.com/ONLY_OFFICE
https://www.youtube.com/user/onlyofficeTV https://www.instagram.com/the_onlyoffice/

http://www.apache.org/licenses/LICENSE-2.0
https://github.com/ONLYOFFICE/CommunityServer/releases

- Overview
- Functionality
- Recommended System Requirements
- Installing Prerequisites
- Installing MySQL
- Installing Community Server
- Configuring Docker Image
  - Storing Data
  - Running ONLYOFFICE Community Server on Different Port
  - Exposing Additional Ports

    - Running ONLYOFFICE Community Server using HTTPS
        - Using the automatically generated Let's Encrypt SSL Certificates
        - Generation of Self Signed Certificates
        - Strengthening the Server Security
        - Installation of the SSL Certificates
        - Available Configuration Parameters
- Installing ONLYOFFICE Workspace
- Upgrading ONLYOFFICE Community Server
- Connecting Your Own Modules
- Project Information
- User Feedback and Support

Overview

ONLYOFFICE Community Server is a free open-source collaborative system developed to manage documents, projects, customer relationship and email correspondence, all in one place.

Starting from version 11.0 Community Server, is distributed as ONLYOFFICE Groups on terms of Apache License.

Functionality

    - Cross platform solution: Linux, Windows
    - Document management
- Integration with Google Drive, Box, Dropbox, OneDrive, OwnCloud
- File sharing
- Document embedding
- Access rights management
- Customizable CRM
- Web-to-lead form
- Invoicing system
- Project Management
- Gantt Chart
- Milestones, task dependencies and subtasks
- Time tracking
- Automated reports
- Blogs, forums, polls, wiki
- Calendar

- Email Aggregator
- People module (employee database)
- Support of more than 20 languages

Community Server (distributed as ONLYOFFICE Groups) is a part of **ONLYOFFICE Workspace** that also includes Document Server (distributed as ONLYOFFICE Docs), Mail Server, Talk (instant messaging app).

Control Panel for administrating **ONLYOFFICE Workspace** can be found in this repo.

Recommended System Requirements

- **RAM**: 4 GB or more
- **CPU**: dual-core 2 GHz or higher
- **Swap file**: at least 2 GB
- **HDD**: at least 2 GB of free space
- **Distributive**: 64-bit Red Hat, CentOS or other compatible distributive with kernel version 3.8 or later, 64-bit Debian, Ubuntu or other compatible distributive with kernel version 3.8 or later
- **Docker**: version 1.9.0 or later

Installing Prerequisites

Before you start **ONLYOFFICE Community Server**, you need to create the following folders:

1. For MySQL server

```
sudo mkdir -p "/app/onlyoffice/mysql/conf.d";
sudo mkdir -p "/app/onlyoffice/mysql/data";
sudo mkdir -p "/app/onlyoffice/mysql/initdb";
```

2. For **Community Server** data and logs

```
sudo mkdir -p "/app/onlyoffice/CommunityServer/data";
sudo mkdir -p "/app/onlyoffice/CommunityServer/logs";
sudo mkdir -p "/app/onlyoffice/CommunityServer/letsencrypt";
```

3. For **Document server** data and logs

```
sudo mkdir -p "/app/onlyoffice/DocumentServer/data";
sudo mkdir -p "/app/onlyoffice/DocumentServer/logs";
```

4. And for **Mail Server** data and logs

```
sudo mkdir -p "/app/onlyoffice/MailServer/data/certs";
sudo mkdir -p "/app/onlyoffice/MailServer/logs";
```

5. For **Control Panel**:

```
sudo mkdir -p "/app/onlyoffice/ControlPanel/data";
sudo mkdir -p "/app/onlyoffice/ControlPanel/logs";
```

Then create the `onlyoffice` network:

```
sudo docker network create --driver bridge onlyoffice
```

Installing MySQL

After that you need to create MySQL server Docker container. Create the configuration file:

```
echo "[mysqld]
sql_mode = 'NO_ENGINE_SUBSTITUTION'
max_connections = 1000
max_allowed_packet = 1048576000
group_concat_max_len = 2048" > /app/onlyoffice/mysql/conf.d/onlyoffice.cnf
```

Create the SQL script which will generate the users and issue the rights to them. The `onlyoffice_user` is required for **ONLYOFFICE Community Server**, and the `mail_admin` is required for **ONLYOFFICE Mail Server** in case it is going to be installed:

```
echo "ALTER USER 'root'@'%' IDENTIFIED WITH mysql_native_password BY 'my-secret-pw';
CREATE USER IF NOT EXISTS 'onlyoffice_user'@'%' IDENTIFIED WITH mysql_native_passwor
CREATE USER IF NOT EXISTS 'mail_admin'@'%' IDENTIFIED WITH mysql_native_password BY '
GRANT ALL PRIVILEGES ON *.* TO 'root'@'%';
GRANT ALL PRIVILEGES ON *.* TO 'onlyoffice_user'@'%';
GRANT ALL PRIVILEGES ON *.* TO 'mail_admin'@'%';
FLUSH PRIVILEGES;" > /app/onlyoffice/mysql/initdb/setup.sql
```

Please note, that the above script will set permissions to access SQL server from any domains (`%`). If you want to limit the access, you can specify hosts which will have access to SQL server.

Now you can create MySQL container setting MySQL version to 8.0.29:

```
sudo docker run --net onlyoffice -i -t -d --restart=always --name onlyoffice-mysql-se
 -v /app/onlyoffice/mysql/conf.d:/etc/mysql/conf.d \
 -v /app/onlyoffice/mysql/data:/var/lib/mysql \
 -v /app/onlyoffice/mysql/initdb:/docker-entrypoint-initdb.d \
 -e MYSQL_ROOT_PASSWORD=my-secret-pw \
 -e MYSQL_DATABASE=onlyoffice \
 mysql:8.0.29
```

Installing Community Server

Use this command to install **ONLYOFFICE Community Server**:

```
sudo docker run --net onlyoffice -i -t -d --privileged --restart=always --name onlyof
 -e MYSQL_SERVER_ROOT_PASSWORD=my-secret-pw \
 -e MYSQL_SERVER_DB_NAME=onlyoffice \
 -e MYSQL_SERVER_HOST=onlyoffice-mysql-server \
 -e MYSQL_SERVER_USER=onlyoffice_user \
 -e MYSQL_SERVER_PASS=onlyoffice_pass \
 -v /app/onlyoffice/CommunityServer/data:/var/www/onlyoffice/Data \
 -v /app/onlyoffice/CommunityServer/logs:/var/log/onlyoffice \
 -v /app/onlyoffice/CommunityServer/letsencrypt:/etc/letsencrypt \
 -v /sys/fs/cgroup:/sys/fs/cgroup:rw \
 onlyoffice/communityserver
```

Configuring Docker Image

Storing Data

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

- **/var/log/onlyoffice** for ONLYOFFICE Community Server logs
- **/var/www/onlyoffice/Data** for ONLYOFFICE Community Server data
- **/etc/letsencrypt** for information on generated certificates

To get access to your data from 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 --cgroupns=host \
    -v /app/onlyoffice/CommunityServer/logs:/var/log/onlyoffice \
    -v /app/onlyoffice/CommunityServer/data:/var/www/onlyoffice/Data \
 -v /app/onlyoffice/CommunityServer/letsencrypt:/etc/letsencrypt \
    -v /sys/fs/cgroup:/sys/fs/cgroup:rw onlyoffice/communityserver
```

Storing the data on the host machine allows you to easily update ONLYOFFICE once the new version is released without losing your data.

Running ONLYOFFICE Community Server on Different Port

To change the port, use the -p command. E.g.: to make your portal accessible via port 8080 execute the following command:

```
sudo docker run -i -t -d --privileged -p 8080:80 --cgroupns=host \
-v /app/onlyoffice/CommunityServer/logs:/var/log/onlyoffice \
-v /app/onlyoffice/CommunityServer/data:/var/www/onlyoffice/Data \
-v /app/onlyoffice/CommunityServer/letsencrypt:/etc/letsencrypt \
-v /sys/fs/cgroup:/sys/fs/cgroup:rw onlyoffice/communityserver
```

Exposing Additional Ports

The container ports to be exposed for **incoming connections** are the folloing:

- **80** for plain HTTP
- **443** when HTTPS is enabled (see below)
- **5222** for XMPP-compatible instant messaging client (for ONLYOFFICE Talk correct work)

You can expose ports by specifying the '-p' option in the docker run command.

```
sudo docker run -i -t -d --privileged -p 80:80 -p 443:443 -p 5222:5222 --cgroupns=hos
-v /app/onlyoffice/CommunityServer/logs:/var/log/onlyoffice \
-v /app/onlyoffice/CommunityServer/data:/var/www/onlyoffice/Data \
-v /app/onlyoffice/CommunityServer/letsencrypt:/etc/letsencrypt \
-v /sys/fs/cgroup:/sys/fs/cgroup:rw onlyoffice/communityserver
```

For **outgoing connections** you need to expose the following ports:

- **80** for HTTP
- **443** for HTTPS

Additional ports to be exposed for the mail client correct work:

- **25** for SMTP
- **465** for SMTPS
- **143** for IMAP
- **993** for IMAPS
- **110** for POP3
- **995** for POP3S

Running ONLYOFFICE Community Server using HTTPS

```
sudo docker run -i -t -d -p 80:80  -p 443:443 \
    -v /app/onlyoffice/CommunityServer/logs:/var/log/onlyoffice \
    -v /app/onlyoffice/CommunityServer/data:/var/www/onlyoffice/Data onlyoffice/commu
```

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, a 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/CommunityServer/data/certs/onlyoffice.key
    /app/onlyoffice/CommunityServer/data/certs/onlyoffice.crt
```

When using CA certified certificates (e.g. Let's Encrypt, these files are provided to you by the CA. When using self-signed certificates you need to generate these files yourself.

Using the automatically generated Let's Encrypt SSL Certificates

```
sudo docker exec -it onlyoffice-community-server bash
bash /var/www/onlyoffice/Tools/letsencrypt.sh yourdomain.com subdomain1.yourdomain.co
```

Where `yourdomain.com` is the address of the domain where your ONLYOFFICE Workspace is installed, and `subdomain1.yourdomain.com` and `subdomain2.yourdomain.com` (and any other subdomains separated with a space) are the subdomains for the main domain which you use.

The script will automatically create and install the letsencrypt.org CA-signed certificate to your server and restart the NGINX service for the changes to take effect.

Now your portal should be available using the `https://` address.

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/CommunityServer/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/CommunityServer/data/certs
cp onlyoffice.key /app/onlyoffice/CommunityServer/data/certs/
cp onlyoffice.crt /app/onlyoffice/CommunityServer/data/certs/
cp dhparam.pem /app/onlyoffice/CommunityServer/data/certs/
chmod 400 /app/onlyoffice/CommunityServer/data/certs/onlyoffice.key
```

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's 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`
- **MYSQL_SERVER_HOST**: The IP address or the name of the host where the server is running.
- **MYSQL_SERVER_PORT**: The port number.
- **MYSQL_SERVER_DB_NAME**: The name of a MySQL database to be created on image startup.
- **MYSQL_SERVER_USER**: The new user name with superuser permissions for the MySQL account.
- **MYSQL_SERVER_PASS**: The password set for the MySQL account.

Installing ONLYOFFICE Workspace

ONLYOFFICE Community Server is a part of ONLYOFFICE Community Edition that comprises also Document Server and Mail Server. To install them, follow these easy steps:

STEP 1: Create the `onlyoffice` network.

```
docker network create --driver bridge onlyoffice
```

Then launch containers on it using the 'docker run --net onlyoffice' option:

STEP 2: Install MySQL.

Follow these steps to install MySQL server.

STEP 3: Generate JWT Secret

JWT secret defines the secret key to validate the JSON Web Token in the request to the **ONLYOFFICE Document Server**. You can specify it yourself or easily get it using the command:

```
JWT_SECRET=$(cat /dev/urandom | tr -dc A-Za-z0-9 | head -c 32);
```

STEP 4: Install ONLYOFFICE Document Server.

```
sudo docker run --net onlyoffice -i -t -d --restart=always --name onlyoffice-document-server \
 -e JWT_ENABLED=true \
 -e JWT_SECRET=${JWT_SECRET} \
 -e JWT_HEADER=AuthorizationJwt \
 -v /app/onlyoffice/DocumentServer/logs:/var/log/onlyoffice  \
 -v /app/onlyoffice/DocumentServer/data:/var/www/onlyoffice/Data  \
 -v /app/onlyoffice/DocumentServer/fonts:/usr/share/fonts/truetype/custom \
 -v /app/onlyoffice/DocumentServer/forgotten:/var/lib/onlyoffice/documentserver/App_Data/cache/f
 onlyoffice/documentserver
```

To learn more, refer to the ONLYOFFICE Document Server documentation.

STEP 5: Install ONLYOFFICE Mail Server.

For the mail server correct work you need to specify its hostname 'yourdomain.com'. To learn more, refer to the ONLYOFFICE Mail Server documentation.

```
sudo docker run --init --net onlyoffice --privileged -i -t -d --restart=always --name onlyoffice
 -e MYSQL_SERVER=onlyoffice-mysql-server \
 -e MYSQL_SERVER_PORT=3306 \
 -e MYSQL_ROOT_USER=root \
 -e MYSQL_ROOT_PASSWD=my-secret-pw \
 -e MYSQL_SERVER_DB_NAME=onlyoffice_mailserver \
 -v /app/onlyoffice/MailServer/data:/var/vmail \
 -v /app/onlyoffice/MailServer/data/certs:/etc/pki/tls/mailserver \
 -v /app/onlyoffice/MailServer/logs:/var/log \
 -h yourdomain.com \
 onlyoffice/mailserver
```

The additional parameters for mail server are available here.

STEP 6: Install ONLYOFFICE Control Panel

```
docker run --net onlyoffice -i -t -d --restart=always --name onlyoffice-control-panel \
-v /var/run/docker.sock:/var/run/docker.sock \
-v /app/onlyoffice/CommunityServer/data:/app/onlyoffice/CommunityServer/data \
-v /app/onlyoffice/ControlPanel/data:/var/www/onlyoffice/Data \
-v /app/onlyoffice/ControlPanel/logs:/var/log/onlyoffice onlyoffice/controlpanel
```

STEP 7: Install ONLYOFFICE Community Server

```
sudo docker run --net onlyoffice -i -t -d --privileged --restart=always --name onlyoffice-commur
 -e MYSQL_SERVER_ROOT_PASSWORD=my-secret-pw \
 -e MYSQL_SERVER_DB_NAME=onlyoffice \
 -e MYSQL_SERVER_HOST=onlyoffice-mysql-server \
 -e MYSQL_SERVER_USER=onlyoffice_user \
 -e MYSQL_SERVER_PASS=onlyoffice_pass \
 -e DOCUMENT_SERVER_PORT_80_TCP_ADDR=onlyoffice-document-server \
 -e DOCUMENT_SERVER_JWT_ENABLED=true \
 -e DOCUMENT_SERVER_JWT_SECRET=${JWT_SECRET} \
 -e DOCUMENT_SERVER_JWT_HEADER=AuthorizationJwt \
 -e MAIL_SERVER_API_HOST=${MAIL_SERVER_IP} \
 -e MAIL_SERVER_DB_HOST=onlyoffice-mysql-server \
 -e MAIL_SERVER_DB_NAME=onlyoffice_mailserver \
 -e MAIL_SERVER_DB_PORT=3306 \
 -e MAIL_SERVER_DB_USER=root \
 -e MAIL_SERVER_DB_PASS=my-secret-pw \
 -e CONTROL_PANEL_PORT_80_TCP=80 \
 -e CONTROL_PANEL_PORT_80_TCP_ADDR=onlyoffice-control-panel \
 -v /app/onlyoffice/CommunityServer/data:/var/www/onlyoffice/Data \
 -v /app/onlyoffice/CommunityServer/logs:/var/log/onlyoffice \
 -v /app/onlyoffice/CommunityServer/letsencrypt:/etc/letsencrypt \
 -v /sys/fs/cgroup:/sys/fs/cgroup:rw \
 onlyoffice/communityserver
```

Where `${MAIL_SERVER_IP}` is the IP address for **ONLYOFFICE Mail Server**. You can easily get it using the command:

```
MAIL_SERVER_IP=$(docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}
```

Alternatively, you can use an automatic installation script to install ONLYOFFICE Workspace at once. For the mail server correct work you need to specify its hostname 'yourdomain.com'.

STEP 1: Download the ONLYOFFICE Workspace Docker script file

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

STEP 2: Install ONLYOFFICE Workspace executing the following command:

```
workspace-install.sh -md yourdomain.com
```

Or use docker-compose.

First you need to clone this GitHub repository:

```
git clone https://github.com/ONLYOFFICE/Docker-CommunityServer
```

After that switch to the repository folder:

```
cd Docker-CommunityServer
```

For the mail server correct work, you need to set a variable with your own hostname for the **Mail Server**:

```
export MAIL_SERVER_HOSTNAME="your-own-hostname"
```

After that, assuming you have docker-compose version 2.16.0 or later installed, execute the following command:

```
docker-compose -f docker-compose.workspace_enterprise.yml up -d
```

The instructions for running ONLYOFFICE Workspace Enterprise Edition are used as an example, if you need to run a different version, specify a different file:

- docker-compose.groups.yml for Community Server (distributed as ONLYOFFICE Groups)
- docker-compose.workspace.yml for ONLYOFFICE Workspace Community Edition
- docker-compose.workspace_enterprise.yml for ONLYOFFICE Workspace Enterprise Edition

Upgrading ONLYOFFICE Community Server

To upgrade to a newer release, please follow these easy steps:

STEP 1: Make sure that all the container volumes are mounted following the **Storing Data** section instructions:

```
sudo docker inspect --format='{{range $p,$conf:=.HostConfig.Binds}}{{$conf}};{{end}}'
```

where {{COMMUNITY_SERVER_ID}} stands for a container name or ID

STEP 2 Remove the current container sudo docker rm -f {{COMMUNITY_SERVER_ID}}

STEP 3 Remove the current image sudo docker rmi -f $(sudo docker images | grep onlyoffice/communityserver | awk '{ print $3 }')

STEP 4 Run the new image with the same map paths

```
sudo docker run -i -t -d --privileged -p 80:80 --cgroupns=host \
-e MYSQL_SERVER_ROOT_PASSWORD=my-secret-pw \
-e MYSQL_SERVER_DB_NAME=onlyoffice \
-e MYSQL_SERVER_HOST=onlyoffice-mysql-server \
-e MYSQL_SERVER_USER=onlyoffice_user \
-e MYSQL_SERVER_PASS=onlyoffice_pass \
-v /app/onlyoffice/CommunityServer/logs:/var/log/onlyoffice  \
-v /app/onlyoffice/CommunityServer/data:/var/www/onlyoffice/Data \
-v /app/onlyoffice/CommunityServer/letsencrypt:/etc/letsencrypt \
-v /sys/fs/cgroup:/sys/fs/cgroup:rw onlyoffice/communityserver
```

This will update Community Server container only and will not connect Document Server and Mail Server to it. You will need to use the additional parameters (like those used during installation) to connect them.

Or you can use ONLYOFFICE Workspace script file to upgrade your current installation:

```
bash workspace-install.sh -u true
```

It will update all the installed components automatically. If you want to update **Community Server** only, use the following command:

```
bash workspace-install.sh -u true -cv 9.1.0.393 -ids false -ims false
```

Where `9.1.0.393` is the number of **Community Server** version which you are going to update to.

Connecting Your Own Modules

You can now create your own modules and connect them to ONLYOFFICE Community Server. See this instruction for more details.

Project Information

Official website: https://www.onlyoffice.com

Code repository: https://github.com/ONLYOFFICE/CommunityServer

License: Apache 2.0

ONLYOFFICE Workspace: https://www.onlyoffice.com/workspace.aspx

User feedback and support

If you have any problems with or questions about this image, please visit our official forum to find answers to your questions: dev.onlyoffice.org or you can ask and answer ONLYOFFICE development questions on Stack Overflow.

OnlyOffice DocSpace

Suite Colaboração OnlyOffice

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

OnlyOffice Apps

Link: https://hub.docker.com/r/onlyoffice/apps

Instalação e Configuração OnlyOffice Apps

Link: https://hub.docker.com/r/onlyoffice/apps/tags

Página para atualização da versão DocSpace para a versão OnlyOffice Apps.

OnlyOffice Community Server

Instalação e Configuração OnlyOffice Community Server

Link: https://hub.docker.com/r/onlyoffice/communityserver

Em 20/10/2026

onlyoffice/communityserver

Verified Publisher

By Ascensio System SIA

•Updated 5 months ago

Collaborative system for managing documents, projects, customer relations and emails in one place

onlyoffice/communityserver repository overview

- Overview
- Functionality
- Recommended System Requirements
- Installing Prerequisites

- Installing MySQL
- Installing Community Server
- Configuring Docker Image
  - Storing Data
  - Running ONLYOFFICE Community Server on Different Port
  - Exposing Additional Ports
  - Running ONLYOFFICE Community Server using HTTPS
    - Using the automatically generated Let's Encrypt SSL Certificates
    - Generation of Self Signed Certificates
    - Strengthening the Server Security
    - Installation of the SSL Certificates
    - Available Configuration Parameters
- Installing ONLYOFFICE Workspace
- Upgrading ONLYOFFICE Community Server
- Connecting Your Own Modules
- Project Information
- User Feedback and Support

Overview

ONLYOFFICE Community Server is a free open-source collaborative system developed to manage documents, projects, customer relationship and email correspondence, all in one place.

Starting from version 11.0 Community Server, is distributed as ONLYOFFICE Groups on terms of Apache License.

Functionality

- Cross platform solution: Linux, Windows
- Document management
- Integration with Google Drive, Box, Dropbox, OneDrive, OwnCloud
- File sharing
- Document embedding
- Access rights management
- Customizable CRM
- Web-to-lead form
- Invoicing system
- Project Management
- Gantt Chart

- Milestones, task dependencies and subtasks
- Time tracking
- Automated reports
- Blogs, forums, polls, wiki
- Calendar
- Email Aggregator
- People module (employee database)
- Support of more than 20 languages

Community Server (distributed as ONLYOFFICE Groups) is a part of **ONLYOFFICE Workspace** that also includes Document Server (distributed as ONLYOFFICE Docs), Mail Server, Talk (instant messaging app).

Control Panel for administrating **ONLYOFFICE Workspace** can be found in this repo.

Recommended System Requirements

- **RAM**: 4 GB or more
- **CPU**: dual-core 2 GHz or higher
- **Swap file**: at least 2 GB
- **HDD**: at least 2 GB of free space
- **Distributive**: 64-bit Red Hat, CentOS or other compatible distributive with kernel version 3.8 or later, 64-bit Debian, Ubuntu or other compatible distributive with kernel version 3.8 or later
- **Docker**: version 1.9.0 or later

Installing Prerequisites

Before you start **ONLYOFFICE Community Server**, you need to create the following folders:

1. For MySQL server

```
sudo mkdir -p "/app/onlyoffice/mysql/conf.d";
sudo mkdir -p "/app/onlyoffice/mysql/data";
sudo mkdir -p "/app/onlyoffice/mysql/initdb";
```

1. For **Community Server** data and logs

```
sudo mkdir -p "/app/onlyoffice/CommunityServer/data";
sudo mkdir -p "/app/onlyoffice/CommunityServer/logs";
sudo mkdir -p "/app/onlyoffice/CommunityServer/letsencrypt";
```

1. For **Document server** data and logs

```
sudo mkdir -p "/app/onlyoffice/DocumentServer/data";
sudo mkdir -p "/app/onlyoffice/DocumentServer/logs";
```

1. And for **Mail Server** data and logs

```
sudo mkdir -p "/app/onlyoffice/MailServer/data/certs";
sudo mkdir -p "/app/onlyoffice/MailServer/logs";
```

1. For **Control Panel**:

```
sudo mkdir -p "/app/onlyoffice/ControlPanel/data";
sudo mkdir -p "/app/onlyoffice/ControlPanel/logs";
```

Then create the `onlyoffice` network:

```
sudo docker network create --driver bridge onlyoffice
```

Installing MySQL

After that you need to create MySQL server Docker container. Create the configuration file:

```
echo "[mysqld]
sql_mode = 'NO_ENGINE_SUBSTITUTION'
max_connections = 1000
max_allowed_packet = 1048576000
group_concat_max_len = 2048
log-error = /var/log/mysql/error.log" > /app/onlyoffice/mysql/conf.d/onlyoffice.cnf
```

Create the SQL script which will generate the users and issue the rights to them. The `onlyoffice_user` is required for **ONLYOFFICE Community Server**, and the `mail_admin` is required for **ONLYOFFICE Mail Server** in case it is going to be installed:

```
echo "CREATE USER 'onlyoffice_user'@'localhost' IDENTIFIED BY 'onlyoffice_pass';
CREATE USER 'mail_admin'@'localhost' IDENTIFIED BY 'Isadmin123';
GRANT ALL PRIVILEGES ON * . * TO 'root'@'%' IDENTIFIED BY 'my-secret-pw';
GRANT ALL PRIVILEGES ON * . * TO 'onlyoffice_user'@'%' IDENTIFIED BY 'onlyoffice_pass';
GRANT ALL PRIVILEGES ON * . * TO 'mail_admin'@'%' IDENTIFIED BY 'Isadmin123';
FLUSH PRIVILEGES;" > /app/onlyoffice/mysql/initdb/setup.sql
```

Please note, that the above script will set permissions to access SQL server from any domains (`%`). If you want to limit the access, you can specify hosts which will have access to SQL server.

Now you can create MySQL container setting MySQL version to 5.7:

```
sudo docker run --net onlyoffice -i -t -d --restart=always --name onlyoffice-mysql-server \
 -v /app/onlyoffice/mysql/conf.d:/etc/mysql/conf.d \
 -v /app/onlyoffice/mysql/data:/var/lib/mysql \
 -v /app/onlyoffice/mysql/initdb:/docker-entrypoint-initdb.d \
 -e MYSQL_ROOT_PASSWORD=my-secret-pw \
 -e MYSQL_DATABASE=onlyoffice \
 mysql:5.7
```

Installing Community Server

Use this command to install **ONLYOFFICE Community Server**:

```
sudo docker run --net onlyoffice -i -t -d --privileged --restart=always --name onlyoffice-
community-server -p 80:80 -p 443:443 -p 5222:5222 \
 -e MYSQL_SERVER_ROOT_PASSWORD=my-secret-pw \
 -e MYSQL_SERVER_DB_NAME=onlyoffice \
 -e MYSQL_SERVER_HOST=onlyoffice-mysql-server \
 -e MYSQL_SERVER_USER=onlyoffice_user \
 -e MYSQL_SERVER_PASS=onlyoffice_pass \
 -v /app/onlyoffice/CommunityServer/data:/var/www/onlyoffice/Data \
 -v /app/onlyoffice/CommunityServer/logs:/var/log/onlyoffice \
 -v /app/onlyoffice/CommunityServer/letsencrypt:/etc/letsencrypt \
 -v /sys/fs/cgroup:/sys/fs/cgroup:ro \
 onlyoffice/communityserver
```

Configuring Docker Image

Storing Data

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

- **/var/log/onlyoffice** for ONLYOFFICE Community Server logs
- **/var/www/onlyoffice/Data** for ONLYOFFICE Community Server data
- **/etc/letsencrypt** for information on generated certificates

To get access to your data from 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 \
    -v /app/onlyoffice/CommunityServer/logs:/var/log/onlyoffice \
    -v /app/onlyoffice/CommunityServer/data:/var/www/onlyoffice/Data \
-v /app/onlyoffice/CommunityServer/letsencrypt:/etc/letsencrypt \
    -v /sys/fs/cgroup:/sys/fs/cgroup:ro onlyoffice/communityserver
```

Storing the data on the host machine allows you to easily update ONLYOFFICE once the new version is released without losing your data.

Running ONLYOFFICE Community Server on Different Port

To change the port, use the -p command. E.g.: to make your portal accessible via port 8080 execute the following command:

```
sudo docker run -i -t -d --privileged -p 8080:80 \
-v /app/onlyoffice/CommunityServer/logs:/var/log/onlyoffice \
-v /app/onlyoffice/CommunityServer/data:/var/www/onlyoffice/Data \
-v /app/onlyoffice/CommunityServer/letsencrypt:/etc/letsencrypt \
-v /sys/fs/cgroup:/sys/fs/cgroup:ro onlyoffice/communityserver
```

Exposing Additional Ports

The container ports to be exposed for **incoming connections** are the folloing:

- **80** for plain HTTP
- **443** when HTTPS is enabled (see below)
- **5222** for XMPP-compatible instant messaging client (for ONLYOFFICE Talk correct work)

You can expose ports by specifying the '-p' option in the docker run command.

```
sudo docker run -i -t -d --privileged -p 80:80 -p 443:443 -p 5222:5222 \
-v /app/onlyoffice/CommunityServer/logs:/var/log/onlyoffice \
-v /app/onlyoffice/CommunityServer/data:/var/www/onlyoffice/Data \
-v /app/onlyoffice/CommunityServer/letsencrypt:/etc/letsencrypt \
-v /sys/fs/cgroup:/sys/fs/cgroup:ro onlyoffice/communityserver
```

For **outgoing connections** you need to expose the following ports:

- **80** for HTTP
- **443** for HTTPS

Additional ports to be exposed for the mail client correct work:

- **25** for SMTP
- **465** for SMTPS
- **143** for IMAP
- **993** for IMAPS
- **110** for POP3
- **995** for POP3S

Running ONLYOFFICE Community Server using HTTPS

```
sudo docker run -i -t -d -p 80:80  -p 443:443 \
    -v /app/onlyoffice/CommunityServer/logs:/var/log/onlyoffice \
    -v /app/onlyoffice/CommunityServer/data:/var/www/onlyoffice/Data
onlyoffice/communityserver
```

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, a 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/CommunityServer/data/certs/onlyoffice.key
    /app/onlyoffice/CommunityServer/data/certs/onlyoffice.crt
```

When using CA certified certificates (e.g. Let's Encrypt, these files are provided to you by the CA. When using self-signed certificates you need to generate these files yourself.

Using the automatically generated Let's Encrypt SSL Certificates

```
sudo docker exec -it onlyoffice-community-server bash
bash /var/www/onlyoffice/Tools/letsencrypt.sh yourdomain.com subdomain1.yourdomain.com
subdomain2.yourdomain.com
```

Where `yourdomain.com` is the address of the domain where your ONLYOFFICE Workspace is installed, and `subdomain1.yourdomain.com` and `subdomain2.yourdomain.com` (and any other subdomains separated with a space) are the subdomains for the main domain which you use.

The script will automatically create and install the letsencrypt.org CA-signed certificate to your server and restart the NGINX service for the changes to take effect.

Now your portal should be available using the `https://` address.

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/CommunityServer/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/CommunityServer/data/certs
cp onlyoffice.key /app/onlyoffice/CommunityServer/data/certs/
```

```
cp onlyoffice.crt /app/onlyoffice/CommunityServer/data/certs/
cp dhparam.pem /app/onlyoffice/CommunityServer/data/certs/
chmod 400 /app/onlyoffice/CommunityServer/data/certs/onlyoffice.key
```

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's 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`
- **MYSQL_SERVER_HOST**: The IP address or the name of the host where the server is running.
- **MYSQL_SERVER_PORT**: The port number.
- **MYSQL_SERVER_DB_NAME**: The name of a MySQL database to be created on image startup.
- **MYSQL_SERVER_USER**: The new user name with superuser permissions for the MySQL account.
- **MYSQL_SERVER_PASS**: The password set for the MySQL account.

Installing ONLYOFFICE Workspace

ONLYOFFICE Community Server is a part of ONLYOFFICE Community Edition that comprises also Document Server and Mail Server. To install them, follow these easy steps:

STEP 1: Create the `onlyoffice` network.

```
docker network create --driver bridge onlyoffice
```

Then launch containers on it using the 'docker run --net onlyoffice' option:

STEP 2: Install MySQL.

Follow these steps to install MySQL server.

STEP 3: Install ONLYOFFICE Document Server.

```
sudo docker run --net onlyoffice -i -t -d --restart=always --name onlyoffice-document-server \
-v /app/onlyoffice/DocumentServer/logs:/var/log/onlyoffice  \
-v /app/onlyoffice/DocumentServer/data:/var/www/onlyoffice/Data  \
-v /app/onlyoffice/DocumentServer/fonts:/usr/share/fonts/truetype/custom \
-v
/app/onlyoffice/DocumentServer/forgotten:/var/lib/onlyoffice/documentserver/App_Data/cache/fil
es/forgotten \
onlyoffice/documentserver
```

To learn more, refer to the ONLYOFFICE Document Server documentation.

STEP 4: Install ONLYOFFICE Mail Server.

For the mail server correct work you need to specify its hostname 'yourdomain.com'. To learn more, refer to the ONLYOFFICE Mail Server documentation.

```
sudo docker run --init --net onlyoffice --privileged -i -t -d --restart=always --name
onlyoffice-mail-server -p 25:25 -p 143:143 -p 587:587 \
 -e MYSQL_SERVER=onlyoffice-mysql-server \
 -e MYSQL_SERVER_PORT=3306 \
 -e MYSQL_ROOT_USER=root \
 -e MYSQL_ROOT_PASSWD=my-secret-pw \
 -e MYSQL_SERVER_DB_NAME=onlyoffice_mailserver \
 -v /app/onlyoffice/MailServer/data:/var/vmail \
 -v /app/onlyoffice/MailServer/data/certs:/etc/pki/tls/mailserver \
 -v /app/onlyoffice/MailServer/logs:/var/log \
 -h yourdomain.com \
 onlyoffice/mailserver
```

The additional parameters for mail server are available here.

STEP 5: Install ONLYOFFICE Control Panel

```
docker run --net onlyoffice -i -t -d --restart=always --name onlyoffice-control-panel \
-v /var/run/docker.sock:/var/run/docker.sock \
-v /app/onlyoffice/CommunityServer/data:/app/onlyoffice/CommunityServer/data \
-v /app/onlyoffice/ControlPanel/data:/var/www/onlyoffice/Data \
-v /app/onlyoffice/ControlPanel/logs:/var/log/onlyoffice onlyoffice/controlpanel
```

STEP 6: Install ONLYOFFICE Community Server

```
sudo docker run --net onlyoffice -i -t -d --privileged --restart=always --name onlyoffice-
community-server -p 80:80 -p 443:443 -p 5222:5222 \
 -e MYSQL_SERVER_ROOT_PASSWORD=my-secret-pw \
 -e MYSQL_SERVER_DB_NAME=onlyoffice \
 -e MYSQL_SERVER_HOST=onlyoffice-mysql-server \
 -e MYSQL_SERVER_USER=onlyoffice_user \
 -e MYSQL_SERVER_PASS=onlyoffice_pass \
 -e DOCUMENT_SERVER_PORT_80_TCP_ADDR=onlyoffice-document-server \
 -e MAIL_SERVER_API_HOST=${MAIL_SERVER_IP} \
 -e MAIL_SERVER_DB_HOST=onlyoffice-mysql-server \
 -e MAIL_SERVER_DB_NAME=onlyoffice_mailserver \
 -e MAIL_SERVER_DB_PORT=3306 \
 -e MAIL_SERVER_DB_USER=root \
 -e MAIL_SERVER_DB_PASS=my-secret-pw \
 -e CONTROL_PANEL_PORT_80_TCP=80 \
 -e CONTROL_PANEL_PORT_80_TCP_ADDR=onlyoffice-control-panel \
 -v /app/onlyoffice/CommunityServer/data:/var/www/onlyoffice/Data \
 -v /app/onlyoffice/CommunityServer/logs:/var/log/onlyoffice \
 -v /app/onlyoffice/CommunityServer/letsencrypt:/etc/letsencrypt \
 -v /sys/fs/cgroup:/sys/fs/cgroup:ro \
 onlyoffice/communityserver
```

Where `${MAIL_SERVER_IP}` is the IP address for **ONLYOFFICE Mail Server**. You can easily get it using the command:

```
docker inspect -f '{{range .NetworkSettings.Networks}}{{.IPAddress}}{{end}}' onlyoffice-mail-
server
```

Alternatively, you can use an automatic installation script to install ONLYOFFICE Workspace at once. For the mail server correct work you need to specify its hostname 'yourdomain.com'.

STEP 1: Download the ONLYOFFICE Workspace Docker script file

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

STEP 2: Install ONLYOFFICE Workspace executing the following command:

```
workspace-install.sh -md yourdomain.com
```

Or, use docker-compose. First you need to clone this GitHub repository:

```
git clone https://github.com/ONLYOFFICE/Docker-CommunityServer
```

After that switch to the repository folder:

```
cd Docker-CommunityServer
```

For the mail server correct work, open one of the files depending on the product you use:

- docker-compose.yml for Community Server (distributed as ONLYOFFICE Groups)
- docker-compose.yml for ONLYOFFICE Workspace Community Edition
- docker-compose.yml for ONLYOFFICE Workspace Enterprise Edition

Then replace the `${MAIL_SERVER_HOSTNAME}` variable with your own hostname for the **Mail Server**. After that, assuming you have docker-compose installed, execute the following command:

```
cd link-to-your-modified-docker-compose
docker-compose up -d
```

Upgrading ONLYOFFICE Community Server

To upgrade to a newer release, please follow these easy steps:

STEP 1: Make sure that all the container volumes are mounted following the **Storing Data** section instructions:

```
sudo docker inspect --format='{{range $p,$conf:=.HostConfig.Binds}}{{$conf}};{{end}}'
{{COMMUNITY_SERVER_ID}}
```

where {{COMMUNITY_SERVER_ID}} stands for a container name or ID

STEP 2 Remove the current container sudo docker rm -f {{COMMUNITY_SERVER_ID}}

STEP 3 Remove the current image sudo docker rmi -f $(sudo docker images | grep onlyoffice/communityserver | awk '{ print $3 }')

STEP 4 Run the new image with the same map paths

```
sudo docker run -i -t -d --privileged -p 80:80 \
-e MYSQL_SERVER_ROOT_PASSWORD=my-secret-pw \
-e MYSQL_SERVER_DB_NAME=onlyoffice \
-e MYSQL_SERVER_HOST=onlyoffice-mysql-server \
-e MYSQL_SERVER_USER=onlyoffice_user \
-e MYSQL_SERVER_PASS=onlyoffice_pass \
-v /app/onlyoffice/CommunityServer/logs:/var/log/onlyoffice  \
-v /app/onlyoffice/CommunityServer/data:/var/www/onlyoffice/Data \
-v /app/onlyoffice/CommunityServer/letsencrypt:/etc/letsencrypt \
-v /sys/fs/cgroup:/sys/fs/cgroup:ro onlyoffice/communityserver
```

*This will update **Community Server** container only and will not connect **Document Server** and **Mail Server** to it. You will need to use the additional parameters (like those used during installation) to connect them.*

Or you can use ONLYOFFICE Workspace script file to upgrade your current installation:

```
bash workspace-install.sh -u true
```

It will update all the installed components automatically. If you want to update **Community Server** only, use the following command:

```
bash workspace-install.sh -u true -cv 9.1.0.393 -ids false -ims false
```

Where `9.1.0.393` is the number of **Community Server** version which you are going to update to.

Connecting Your Own Modules

You can now create your own modules and connect them to ONLYOFFICE Community Server. See this instruction for more details.

Project Information

Official website: https://www.onlyoffice.com

Code repository: https://github.com/ONLYOFFICE/CommunityServer

License: Apache 2.0

ONLYOFFICE Workspace: https://www.onlyoffice.com/workspace.aspx

User feedback and support

If you have any problems with or questions about this image, please visit our official forum to find answers to your questions: dev.onlyoffice.org or you can ask and answer ONLYOFFICE development questions on Stack Overflow.