

- 2) Статья установки OnlyOffice Document Server <https://helpcenter.onlyoffice.com/ru/>
- 3) Создаем папки `sudo mkdir /app /app/onlyoffice /app/onlyoffice/DocumentServer && cd /app/o*/D* && sudo mkdir logs data data/certs lib db`
- 4) Строку для конфигурационного файла nextcloud не пропустил ютуб, смотрите в видео
- 5) Команда если после жесткой перезагрузки не запускается nextcloud:
`sudo snap disable nextcloud && sudo snap enable nextcloud`
- 6) `sudo docker run -i -t -d -p 8443:443 --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`

Установка ONLYOFFICE Docs

У вас должна быть установлена последняя версия Docker. В противном случае обратитесь к разделу **Installation** на сайте Docker за инструкциями по установке.

В Docker по умолчанию указаны DNS-серверы Google. Если ваш **ONLYOFFICE Docs** не будет иметь доступа в Интернет, рекомендуется изменить заданный в Docker по умолчанию адрес DNS-сервера на адрес вашего локального DNS-сервера. Для этого откройте файл `/etc/default/docker` и замените IP-адрес в следующей строке на IP-адрес DNS-сервера в вашей локальной сети:

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

Для операционных систем на основе RPM, например, CentOS, файл конфигурации может находиться здесь: `/etc/docker/daemon.json`

После того, как Docker будет установлен, запустите его и выполните следующую команду:

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

Используйте эту команду, если вы хотите установить **ONLYOFFICE Docs** отдельно.

Чтобы установить **ONLYOFFICE Docs** с интегрированными модулями **ONLYOFFICE Groups** и **ONLYOFFICE Mail**, обратитесь к соответствующим инструкциям ниже.

По умолчанию **ONLYOFFICE Docs** слушает входящие соединения на порту **80**. Начиная с версии **4.3** вы можете сменить порт для **ONLYOFFICE Docs** в случае, если вы планируете использовать его вместо того, который идёт по умолчанию. Это можно легко сделать, сменив порт во время установки:

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

Где `<PORT_NUMBER>` - это номер порта, который вы хотите использовать для **ONLYOFFICE Docs**.

В результате будет установлена версия **ONLYOFFICE Docs** со всеми необходимыми зависимостями (список команд, которые выполняет скрипт, можно найти [здесь](#)).

Сохранение данных вне контейнеров

Все данные хранятся в специально отведенных для этого каталогах, **томах данных**, которые находятся по следующему адресу:

- /var/log/onlyoffice для лог-файлов **ONLYOFFICE Docs**
- /var/www/onlyoffice/Data для сертификатов
- /var/lib/onlyoffice для кэша файлов
- /var/lib/postgresql для базы данных

Рекомендация Настоятельно рекомендуется сохранять данные вне контейнеров Docker на хост-компьютере, так как это позволяет при выходе новой версии легко обновлять **ONLYOFFICE Docs** без потери данных.

Для получения доступа к данным извне контейнера надо монтировать папку хоста к контейнеру. Это можно сделать, указав значение параметра **-v** в команде docker run.

```
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
```

Пожалуйста, обратите внимание на то, что, если вы пытаетесь монтировать еще не созданные папки, эти папки будут созданы, но доступ к ним будет ограничен. Вам понадобится изменить права доступа к ним вручную.

Обычно нет необходимости хранить данные контейнера, поскольку работа контейнера не зависит от их состояния. Сохранение данных будет полезно:

- для обеспечения быстрого доступа к данным контейнера, таким как лог-файлы;
- для снятия ограничений по объему данных внутри контейнера;
- при использовании сервисов, запускаемых извне контейнера, таких как PostgreSQL, Redis, RabbitMQ.

Запуск ONLYOFFICE Docs с использованием HTTPS

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

Для предотвращения несанкционированного доступа к информации вы можете обезопасить доступ к приложению onlyoffice, используя протокол SSL. SSL-сертификат, заверенный центром сертификации, предусматривает проверку отношений доверия через центр сертификации, но самоподписанные сертификаты могут обеспечить такой же уровень проверки отношений доверия, поскольку каждый клиент предпринимает ряд дополнительных мер для проверки подлинности вашего сайта. Ниже приведены инструкции о том, как обеспечить безопасный доступ к приложению onlyoffice.

Для обеспечения безопасного доступа к приложению с помощью SSL в принципе нужны две вещи:

- **Закрытый ключ (.key)**
- **SSL-сертификат (.crt)**

Итак, вам надо создать и установить следующие файлы:

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

При использовании сертификатов, заверенных центром сертификации, эти файлы вам предоставляет центр сертификации (в таком случае пропустите следующий раздел). При использовании самоподписанных сертификатов вам надо сгенерировать эти файлы самостоятельно.

Генерация самоподписанных сертификатов

Генерация самоподписанных SSL-сертификатов включает в себя три простых шага:

ШАГ 1: Создайте закрытый ключ сервера

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

ШАГ 2: Создайте запрос подписи сертификата (CSR)

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

ШАГ 3: Подпишите сертификат с помощью закрытого ключа и CSR

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

Вы только что сгенерировали SSL-сертификат со сроком действия 365 дней.

Усиление безопасности сервера

В этом разделе приводятся инструкции по **усилению безопасности вашего сервера**. Для этого необходимо сгенерировать параметры Диффи-Хеллмана (DHE), обеспечивающие более высокую стойкость.

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

Установка SSL-сертификатов

Из четырех ранее сгенерированных файлов на сервер onlyoffice необходимо установить файлы onlyoffice.key, onlyoffice.crt и dhparam.pem. Файл CSR не требуется, но убедитесь, что вы создали его резервную копию (на случай, если он вам снова понадобится).

По умолчанию приложение onlyoffice настроено для поиска SSL-сертификатов в каталоге /var/www/onlyoffice/Data/certs. Этот путь можно изменить, используя параметры настройки SSL_KEY_PATH, SSL_CERTIFICATE_PATH и SSL_DHPARAM_PATH.

Путь /var/www/onlyoffice/Data/ - это путь к хранилищу данных. Вам надо создать папку с именем certs в каталоге /app/onlyoffice/DocumentServer/data/ и скопировать файлы в нее, а также в качестве меры безопасности необходимо изменить права доступа к файлу onlyoffice.key, чтобы он был доступен на чтение только владельцу.

```
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
```

И перезапустите Docker-контейнер:

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

Остался всего один шаг, чтобы обеспечить безопасный доступ к приложению.

Доступные параметры настройки

Флаг `--env-file` в параметрах команды `docker run` позволяет задать все необходимые переменные среды в одном файле. Так можно избежать написания потенциально длинной команды `docker run`.

Ниже приводится полный список параметров, которые можно настроить, используя переменные среды.

- **ONLYOFFICE_HTTPS_HSTS_ENABLED**: Дополнительный параметр настройки для отключения HSTS (HTTP Strict Transport Security). Применяется только в тех случаях, когда используется SSL. По умолчанию задано значение `true`.
- **ONLYOFFICE_HTTPS_HSTS_MAXAGE**: Дополнительный параметр настройки для задания параметра `max-age` HSTS в конфигурации виртуального хоста NGINX для onlyoffice. Применяется только в тех случаях, когда используется SSL. По умолчанию задано значение `31536000`.
- **SSL_CERTIFICATE_PATH**: Путь к используемому SSL-сертификату. По умолчанию задано значение `/var/www/onlyoffice/Data/certs/onlyoffice.crt`.
- **SSL_KEY_PATH**: Путь к закрытому ключу SSL-сертификата. По умолчанию задано значение `/var/www/onlyoffice/Data/certs/onlyoffice.key`.
- **SSL_DHPARAM_PATH**: Путь к параметру Диффи-Хеллмана. По умолчанию задано значение `/var/www/onlyoffice/Data/certs/dhparam.pem`.
- **SSL_VERIFY_CLIENT**: Включение проверки сертификатов клиента с помощью файла `CA_CERTIFICATES_PATH` file. По умолчанию задано значение `false`.
- **POSTGRESQL_SERVER_HOST**: IP-адрес или имя хоста, на котором запущен сервер PostgreSQL.
- **POSTGRESQL_SERVER_PORT**: Номер порта сервера PostgreSQL.
- **POSTGRESQL_SERVER_DB_NAME**: Имя базы данных PostgreSQL, которая должна создаваться при запуске образа.
- **POSTGRESQL_SERVER_USER**: Имя нового пользователя с правами суперпользователя для учетной записи PostgreSQL.
- **POSTGRESQL_SERVER_PASS**: Пароль, заданный для учетной записи PostgreSQL.
- **AMQP_SERVER_URL**: **AMQP URL** для подключения к серверу менеджера сообщений.
- **AMQP_SERVER_TYPE**: Тип менеджера сообщений. Поддерживаются значения `rabbitmq` или `activemq`. По умолчанию задано значение `rabbitmq`.
- **REDIS_SERVER_HOST**: IP-адрес или имя хоста, на котором запущен Redis.
- **REDIS_SERVER_PORT**: Номер порта сервера Redis.
- **NGINX_WORKER_PROCESSES**: Определяет число рабочих процессов NGINX.
- **NGINX_WORKER_CONNECTIONS**: Задаёт максимальное число соединений, которые одновременно может открыть рабочий процесс NGINX.
- **JWT_ENABLED**: Управляет включением валидации посредством JSON веб-токена **ONLYOFFICE Docs**. По умолчанию задано значение `false`.
- **JWT_SECRET**: Определяет секретный ключ для валидации JSON веб-токена при запросе к **ONLYOFFICE Docs**. По умолчанию задано значение `secret`.
- **JWT_HEADER**: Определяет HTTP-заголовок, который будет использоваться для отправки JSON веб-токена. По умолчанию задано значение `Authorization`.

Установка ONLYOFFICE Docs с интегрированными модулями ONLYOFFICE Groups и ONLYOFFICE Mail

ONLYOFFICE Docs входит в состав версии **ONLYOFFICE Workspace**, которая также включает в себя **ONLYOFFICE Community Server**, **ONLYOFFICE Control Panel** и **ONLYOFFICE Mail**. Если вы хотите установить их все и интегрировать друг с другом, обратитесь к [этим инструкциям](#).

Альтернативные способы установки ONLYOFFICE Docs

Вы также можете использовать скрипт автоматической установки, чтобы установить все модули версии **Workspace** сразу. Для корректной работы почтового сервера необходимо указать имя хоста `yourdomain.com`.

ШАГ 1: Скачайте файл установочного скрипта

Выполните следующую команду:

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

ШАГ 2: Установите Workspace

Для этого выполните следующую команду:

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

Вы также можете использовать утилиту [docker-compose](#), чтобы установить **ONLYOFFICE Docs**. Обратитесь к [следующим инструкциям](#), чтобы узнать, как это сделать.

Запуск ONLYOFFICE Docs с помощью Docker Compose

ONLYOFFICE Docs [v.6.2](#) [История изменений ONLYOFFICE Docs](#)

- локальный сервер
- серверная версия
- установка серверного решения
- ONLYOFFICE Docs
- версия для Docker

Введение

Использование **Docker Compose** позволяет:

- распределенно запустить **ONLYOFFICE Docs** и всю необходимую для него инфраструктуру;
В данной инструкции рассматривается использование инструмента **Docker Compose** на одном компьютере. Для распределенного запуска контейнеров обратитесь к документации **Docker Swarm**.
- гибко масштабировать производительность **ONLYOFFICE Docs**;
- обеспечить отказоустойчивость приложения.

В данной инструкции объясняется, как установить на компьютере **ONLYOFFICE Docs** с помощью **Docker Compose**.

Системные требования

- **Процессор**

двухъядерный с тактовой частотой **2 ГГц** или лучше

- **Оперативная память**

не менее **2 Гб**

- **Свободное место на жестком диске**

не менее **40 Гб**

- **Дополнительные требования**

не менее **4 Гб** для файла подкачки

- **Операционная система**

amd64 дистрибутив Linux с версией ядра **3.10** или выше

- **Дополнительные требования**

○ **Docker**: версия **1.10** или выше

Установка Docker Compose

Установите `docker-compose`, следуя инструкциям, которые вы можете найти [здесь](#).

Получение файла docker-compose.yml

Выполните следующую команду:

```
wget https://raw.githubusercontent.com/ONLYOFFICE/Docker-DocumentServer/master/docker-compose.yml
```

Запуск и работа ONLYOFFICE Docs

Для запуска **ONLYOFFICE Docs** выполните следующую команду:

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

Для масштабирования **ONLYOFFICE Docs** выполните следующую команду:

```
sudo docker-compose scale onlyoffice-documentserver=<container_number>
```

где **container_number** - число одновременно работающих контейнеров **ONLYOFFICE Docs**.

Для остановки **ONLYOFFICE Docs** используйте следующую команду:

```
sudo docker-compose down
```

Настройка ONLYOFFICE Docs

Использование пользовательских шрифтов

Создайте мап шрифтов с локальной машины в папку шрифтов в контейнере **onlyoffice-documentserver-data**. Для этого:

1. откройте файл **docker-compose.yml**,
2. найдите раздел **services** -> **onlyoffice-documentserver-data** -> **volumes**,
3. замените строку - /usr/share/fonts на следующую:

```
- /your_local_fonts_folder:/usr/share/fonts
```

Настройка HTTPS

Для использования протокола HTTPS необходимо раскомментировать следующие строки в файле **docker-compose.yml**:

```
- FORCE_SSL=true  
- CERT_FOLDER=/certs/  
- /app/onlyoffice/DocumentServer/data/certs/onlyoffice.pem:/certs/cert1.pem
```

Создайте файл **/app/onlyoffice/DocumentServer/data/certs/onlyoffice.pem**, который содержит сертификат и приватный ключ.

При использовании сертификатов, заверенных центром сертификации, эти файлы вам предоставляет центр сертификации (в таком случае пропустите **шаги 1-3** следующего раздела). При использовании самоподписанных сертификатов вам надо сгенерировать эти файлы самостоятельно.

Генерация самоподписанных сертификатов

Генерация самоподписанных SSL-сертификатов включает в себя три простых шага:

ШАГ 1: Создайте закрытый ключ сервера

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

ШАГ 2: Создайте запрос подписи сертификата (CSR)

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

ШАГ 3: Подпишите сертификат с помощью закрытого ключа и CSR

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

ШАГ 4: Объедините оба файла

```
cp onlyoffice.crt onlyoffice.pem  
cat onlyoffice.key >> onlyoffice.pem
```

ПРАВИЛЬНОЕ обновление NextCloud Docker

1) Останавливаем приложение:

```
sudo docker-compose stop
```

2) Правим в конфигурации версию NextCloud на ЕДЕНИЦУ больше нашей версии (если 15, то на 16)

```
sudo nano docker-compose.yml
```

и сохраняем

3)

```
sudo docker-compose pull
```

4)

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

5) Обновляем страницу, если будет ошибка nginx, то это норма, ждем 1-2 минуты и обновляем снова.

6) ПОВТОРЯЕМ все шаги до последней версии

=====

Далее в настройках NextCloud через веб-доступ в разделе Overview мы скорее всего увидим ошибки безопасности, нужно будет их исправить.

1) Команда для shell-доступа в наш контейнер nextcloud-app:

```
sudo docker exec -u www-data -it nextcloud-app
```

2) Выполняем сначала первую команду:

```
php occ db:add-missing-indices
```

3) Потом вторую команду:

```
php occ db:convert-filecache-bigint
```


NextCloud Rclone БЕСПЛАТНО расширяем NextCloud до БЕСКОНЕЧНОСТИ

1) Создайте каталог точки монтирования удаленного хранилища:

```
mkdir /home/ubuntu/clouds/firstDropbox
```

2) Запустите команду создания конфигурации для rclone:

```
rclone config
```

3) Следуйте инструкциям из видео или по ссылке:

<https://rclone.org/dropbox/>

4) После создания конфигурации можно монтировать облако:

```
rclone mount firstDropbox:/ /home/ubuntu/clouds/firstDropbox --umask 000 --allow-non-empty --allow-other --dir-cache-time 12h --buffer-size 64M --vfs-cache-mode full --vfs-read-chunk-size 40M --vfs-read-chunk-size-limit 512M --uid 1000 --vfs-cache-max-age 24h --vfs-cache-max-size 4G --log-level INFO --log-file /home/ubuntu/firstDropbox.log --daemon
```

=====

СОЗДАЕМ СЛУЖБУ:

```
vim /etc/systemd/system/firstdropbox.service
```

И копируем туда эти строки (заменить на свое):

```
#
```

```
[Service]
```

```
Type=notify
```

```
Environment=RCLONE_CONFIG=/home/ubuntu/.config/rclone/rclone.conf
```

```
KillMode=none
```

```
RestartSec=5
```

```
ExecStart=/usr/bin/rclone mount firstDropbox:/ /home/ubuntu/clouds/firstDropbox \
```

```
--umask 000 \
```

```
--allow-non-empty \
```

```
--allow-other \
```

```
--dir-cache-time 12h \
```

```
--buffer-size 64M \
```

```
--vfs-cache-mode full \
```

```
--vfs-read-chunk-size 40M \
```

```
--vfs-read-chunk-size-limit 512M \
```

```
--uid 1000 \
```

```
--vfs-cache-max-age 24h \
```

```
--vfs-cache-max-size 4G \
```

```
--log-level INFO \
```

```
--log-file /home/ubuntu/firstDropbox.log
ExecStop=/bin/fusermount -uz /home/ubuntu/clouds/firstDropbox
Restart=on-failure
User=ubuntu
Group=ubuntu
```

[Install]

```
WantedBy=multi-user.target
```

```
#####
```

```
#
```

```
=====
```

ВКЛЮЧАЕМ НАШУ СЛУЖБУ:

```
sudo systemctl start firstdropbox.service
```

```
sudo systemctl enable firstdropbox.service
```

```
=====
```