docker rm $(docker ps -a -q)

docker rmi $(docker images -q)

docker stop OnlyOffice-DocumentServer

docker rm OnlyOffice-DocumentServer

docker rmi onlyoffice/documentserver

You can pull the image on a computer that have access to the internet.

sudo docker pull onlyoffice/documentserver

Then you can save this image to a file

sudo docker save -o onlyoffice/documentserver\_image.docker onlyoffice/documentserver

Transfer the file on the offline computer (USB/CD/whatever) and load the image from the file:

sudo docker load -i /documentserver\_image.docker

bzw. bei Paket –Installation:

sudo apt-get install onlyoffice-documentserver

##### 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](https://raymii.org/s/tutorials/Strong_SSL_Security_On_nginx.html).

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

Add your selfsigned or wildcard-certificate to the ca-bundle of ownCloud/Nextcloud

cat /app/onlyoffice/DocumentServer/data/certs/onlyoffice.crt >> /var/www/25\_NextCloud/resources/config/ca-bundle.crt

And restart Docker container:

sudo docker restart {{DOCUMENT\_SERVER\_ID}}

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

docker run --name=ONLYOFFICEDOCKER -i -t -d -p 8443:443 --restart=always \

 -e JWT\_ENABLED='true' -e JWT\_SECRET='97jfbeil05' \

 -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 \

 -v /app/onlyoffice/DocumentServer/etc/hosts:/etc/hosts \

onlyoffice/documentserver

offene ports scannen

netstat -lntp

##### 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.
* **RABBITMQ\_SERVER\_HOST**: The IP address or the name of the host where the RabbitMQ server is running.
* **RABBITMQ\_SERVER\_USER**: The RabbitMQ server user name.
* **RABBITMQ\_SERVER\_PASS**: The password set for the RabbitMQ account.
* **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.