Инструкция по установке MFlash в Dockerконтейнерах на ОС Альт.

0. Предварительные действия

Обратите внимание!

Дальнейшие действия будут выполняться от имени пользователя root!

su – root

1. Установка Docker

1) Перед установкой Docker обновите список пакетов и перезагрузите систему:

```
apt-get update
apt-get dist-upgrade
apt-get clean
reboot
```

2) Установите Docker:

```
apt-get install docker-engine wget -y || apt-get install
docker-ce wget -y
```

3) Включите и запустите Docker-сервис:

```
systemctl start docker
```

2.Загрузка Docker-образов

Перенесите необходимые архивы Docker-образов на сервер, где будет проводиться установка.

1) Загрузите Docker-образы из архивов в:

```
docker load < mf-app-8.2-*.tar.gz
docker load < mf-svc-8.2-*.tar.gz
docker load < mf-ws-8.2-*.tar.gz
docker pull nginx
```

3. Установка Postgresql

Для работы сервера необходимо развернуть сервер СУБД PostgreSQL версии 15 или выше. Для установки и настройки сервера СУБД обратитесь к документации <u>вендора ОС Альт</u>.

Сервер СУБ может быть установлен как на одном хосте с сервером приложений, так и раздельно с ним, в любой конфигурации (standalone, fall-tolerant).

После установки сервера СУБД необходимо произвести следующие действия по его настройке:

1. Отредактируйте файл pg_hba.conf для настройки аутентификации клиентов и файл postgresql.conf для общей настройки сервера PostgreSQL.

vim /etc/postgresql/15/main/pg_hba.conf

# Database administrative login by Unix domain socket							
local	all	postgres		peer			
# TYPE	DATABASE	USER	ADDRESS	METHOD			
				10.000/0220-0220			
# "loca	l" is for Unix o	domain socket con	nections only				
local	all	all		peer			
# IPv4 local connections:							
host	all	all	192.168.1.89/24	md5			
host	all	all	192.168.1.0/24	md5			
host	all	all	127.0.0.1/32	scram-sha-256			
# IPv6 local connections:							
host	all	all	::1/128	scram-sha-256			

Добавьте или измените следующие строки, чтобы разрешить подключения:

Разрешить подключения с любого IP-адреса с использованием md5-аутентификации host all all 0.0.0/0 md5

2. Откройте файл конфигурации PostgreSQL:

vim /etc/postgresql/15/main/postgresql.conf

3. Найдите параметр listen_addresses и измените его значение на '*':

listen_addresses = '*'

# CONNECTIONS AND AUTHENTICATION					
#					
# - Connection Settings -					
listen_addresses = '*' # what	IP address(es) to listen on;				
	<pre># comma-separated list of addresses;</pre>				
	<pre># defaults to 'localhost'; use '*' for all</pre>				
	<pre># (change requires restart)</pre>				
port = 5432	# (change requires restart)				
<pre>max_connections = 100</pre>	# (change requires restart)				
<pre>#superuser_reserved_connections = 3</pre>	<pre># (change requires restart)</pre>				
unix_socket_directories = '/var/run/postgresql' # comma-separated list of directories					
	<pre># (change requires restart)</pre>				
<pre>#unix_socket_group = ''</pre>	<pre># (change requires restart)</pre>				
<pre>#unix_socket_permissions = 0777</pre>	<pre># begin with 0 to use octal notation</pre>				
	<pre># (change requires restart)</pre>				
<pre>#bonjour = off</pre>	<pre># advertise server via Bonjour</pre>				

4. Откройте psql под пользователем postgres и измените его пароль.

sudo -u postgres psql

5. Измените пароль пользователя postgres на mflash01

alter user postgres password 'mflash01';

- 6. Закройте psql командой \q.
- 7. Перезапустите службу PostgreSQL, чтобы применить изменения в конфигурационных файлах:

systemctl restart postgresql

8. Включите PostgreSQL, чтобы он запускался при загрузке системы:

systemctl enable postgresql

3. Настройка Docker и шифрование строки подключения

1. Создайте Docker volume для веб-приложения:

docker volume create mflash_web_space

 Зашифруйте строку подключения к базе данных с помощью следующей команды:

/usr/bin/docker run --rm --name mflash-web-test \

- -e appToken=asdwerejwekjhrekhjdsgjhgjdsfsf \
- -e appSecret=dsadasddewrwerkjhaskjdhmnljlkajs \
- -e saltPass=asdsadasdsaeerwerwer $\$
- -e saltToken=sdasdasdasd

-e dbConnKey=rEdgm5ElK5Pt6fWarRWgwwAOcc5PEZRi \



 После того как получили
 Entered encrypted string, его необходимо

 указать
 в
 переменной
 dbConnEncr:
 \

Примечание: Параметры APP_TOKEN, APP_SECRET, SALT_PASS, SALT_TOKEN, DB_CONN_KEY должны быть одинаковыми при шифровании строки подключения к БД и при запуске контейнера

4.Создание и настройка сервиса mflash-web.service

1. Создайте файл mflash-web.service сервиса для systemd.

vim /etc/systemd/system/mflash-web.service

2. Вставьте следующий текст:

[Unit] Description=MFlash Web After=docker.service Requires=docker.service						
<pre>[Service] TimeoutStartSec=0 Restart=always ExecStartPre=-/usr/bin/docker container inspect %n >/dev/null 2>&1 && -/usr/bin/docker exec %n stop true ExecStart=/usr/bin/docker runrmname %n \ -v /opt/mflash/logs:/var/log/ \ -v mflash_web_space:/srv/mflash \ -v /opt/mflash/ssl/:/srv/mflash/ssl/ \ -e TZ="Europe/Moscow" \</pre>						
<pre>dbConnEncr=rAymOLjOGeTeYd093UdjmKznhrMZZRBStdz7CffAPdmrBTSme0 yOB+akLs7EYjm8mDbad9+0+5bcm0PSfiaFN5Ae2XgG99ZUC7bjeaM/E/ mnaG/e7ZbnmnZMLpfJw8kn2++I+M6sjUmieBtrTQNVHR0fem5dQ/ c0cQHiUelnDAA= \ -e appToken=asdwerejwekjhrekhjdsgjhgjdsfsf \ -e appSecret=dsadasddewrwerkjhaskjdhmnljlkajs \ -e saltPass=asdsadasdsaeerwerwer \ -e saltToken=sdasdasdasdasd \ -e dbConnKey=rEdgm5ElK5Pt6fWarRWgwwAOcc5PEZRi \</pre>						

```
-p 9000:9000 \
--network=host \
hub.msoftgroup.ru:5000/mf-app:8.2-344
```

[Install] WantedBy=default.target

- Замените значение dbConnEncr на полученную ранее зашифрованную строку.
- Значения переменных appToken, appSecret, saltPass, saltToken, dbConnKey должны быть одинаковыми при шифровании строки подключения и при запуске контейнера.

5. Настройка конфигурационного файла Nginx

1. Создайте файл сервиса для systemd:

vim /etc/systemd/system/nginx-docker.service

2. Добавьте следующее содержимое в файл:

```
[Unit]
Description=Nginx
After=docker.service
Requires=docker.service
[Service]
TimeoutStartSec=0
Restart=always
ExecStartPre=-/usr/bin/docker container inspect %n >/dev/null
2>&1 && -/usr/bin/docker exec %n stop || true
ExecStart=/usr/bin/docker run --rm --name %n \
     -e TZ="Europe/Moscow" \
     -v /opt/mflash/logs:/var/log/nginx \
     -v mflash_web_space:/srv/mflash
                                       \mathbf{i}
     - V
/opt/mflash/mf_conf/mflash.conf:/etc/nginx/conf.d/mflash.conf
     -v /opt/mflash/mf conf/:/etc/ssl/certs/ \
     -p 80:80 \
     -p 443:443 \
     --network=host \
     nginx:latest
[Install]
WantedBy=default.target
```

5. Настройка конфигурационного файла Nginx

1. Откройте конфигурационный файл Nginx для редактирования:

vim /opt/mflash/mf_conf/mflash.conf

2. Добавьте следующее содержимое в файл:

```
upstream mf-ws {
     server 127.0.0.1:8083;
}
server {
     listen 80;
     server_name localhost 192.168.1.89;
     rewrite (.*) https://192.168.1.89$1;
}
server {
     listen 443 default_server ssl;
     ssl_certificate /etc/ssl/certs/nginx-selfsigned.crt;
     ssl_certificate_key /etc/ssl/certs/nginx-selfsigned.key;
     ssl_protocols TLSv1.2 TLSv1.3;
     ssl_ciphers
                          ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-
AES128-GCM-SHA256: ECDHE-ECDSA-AES256-GCM-SHA384: ECDHE-RSA-AES256-
GCM-SHA384:DHE-RSA-AES128-GCM-SHA256:DHE-RSA-AES256-GCM-SHA384;
     client_max_body_size 2g;
     proxy_connect_timeout 159s;
     proxy_send_timeout
                           600;
     proxy_read_timeout
                           600;
     server_name localhost 192.168.1.89;
     location ~ /(admin|mflash|share)/ {
           root /srv/mflash/web/;
           index index.php index.html index.htm;
           location ~ ^/(admin|mflash|share)/(.*)\.php$ {
                fastcgi_read_timeout 200;
                fastcgi_pass 192.168.1.89:9000;
     fastcgi_index index.php;
                fastcgi_param SCRIPT_FILENAME
/srv/mflash/web$fastcgi_script_name;
                include fastcgi_params;
           }
     }
     location ~ /api_v2(.*) {
           root /srv/mflash/api_v2/public;
           try_files $uri /index.php$1$is_args$args;
     }
     location ~ ^/index\.php(.*)$ {
           fastcgi_read_timeout 6400;
```



Замените 192.168.1.89 на нужный ІР-адрес.

Создание самоподписанных сертификатов nginx:

1. Сгенерируйте самоподписанный сертификат с помощью следующей команды:

openssl req -x509 -nodes -days 365 -newkey rsa:2048 \ -subj

"/C=RU/ST=Moscow/L=Moscow/O=MSOFT/OU=MFLASH/CN=10.1.1.65/emai
lAddress=support@10.1.1.65" \

- -out /opt/mflash/mf_conf/nginx-selfsigned.crt
- Поменяйте значение CN = 192.168.1.89 на локальный адрес сервера.
- Убедитесь, что файлы сертификата и ключа находятся в директории. /opt/mflash/mf_conf/.

6. Шифрование строки подключения к базе данных

1. Для генерации зашифрованной строки подключения к базе данных используйте следующую команду:

/usr/bin/docker run \
hub.msoftgroup.ru:5000/mf-svc:8.2-61 \
dotnet /opt/mflash/data/cryptutil/cryptutil.dll \
'server=10.1.1.65;
port=5432;
username=postgres;
password=mflash01;
database=mflash;
Search Path=mflash;
options=--lc_messages=C.UTF-8;'

Сохраните переменную connectionString для дальнейшего использования <value> </value>

6. Создание и настройка сервиса mflash-service

1. Создайте файл сервиса для systemd:

vim /etc/systemd/system/mflash-service.service

2. Добавьте следующее содержимое в файл сервиса:

```
[Unit]
Description=MFlash Service
After=docker.service
Requires=docker.service
[Service]
TimeoutStartSec=0
Restart=always
ExecStartPre=-/usr/bin/docker container inspect %n >/dev/null
2>&1 && -/usr/bin/docker exec %n stop || true
ExecStart=/usr/bin/docker run --rm --name %n \
     -v /opt/mflash/smw:/srv/mflash/smw \
     -w /opt/mflash/data/service/
     -e TZ="Europe/Moscow" \
     - e
connectionString=N4PTBl1A6oSB05d3u4eqmxZCJpxr2+5sLZk9Jh7GZDc0
0IYQ/
jM0oo9lTG5oTRk55Hjej4GLUuSAX1gnvxzb632fdQ2+gC68r1fPUu0Yn1a3TI
HooW009lAcEFed5AqDn/9KtrQg3Fzbceod36m/0bvNuTFGLIJK/
GpgFLErfK/Tb/iPxLHePMCKHDC8qp8W \
     -e role=service \
     -e appToken=asdwerejwekjhrekhjdsgjhgjdsfsf \
     -e workerUrl=http://127.0.0.1:11090 \
```

-e adminSafeList=127.0.0.1;192.168.1.89 \

```
-e freshclam=false \
-p 11080:11080 \
--network=host \
hub.msoftgroup.ru:5000/mf-svc:8.2-61
```

[Install] WantedBy=default.target

3. Поменяйте значение connectionString на полученное на предыдущем этапе

7. Создание и настройка сервис файла mflash-worker

1. Создайте файл сервиса для systemd:

vim /etc/systemd/system/mflash-worker.service

2. Добавьте следующее содержимое в файл сервиса:

```
[Unit]
Description=MFlash Worker
After=docker.service
Requires=docker.service
[Service]
TimeoutStartSec=0
Restart=always
ExecStartPre=-/usr/bin/docker container inspect %n >/dev/null 2>&1
&& -/usr/bin/docker exec %n stop || true
ExecStart=/usr/bin/docker run --rm --name %n \
     -v /opt/mflash/smw:/srv/mflash/smw \
     -w /opt/mflash/data/worker/
     -e TZ="Europe/Moscow" ∖
     - e
connectionString=N4PTBl1A6oSB05d3u4eqmxZCJpxr2+5sLZk9Jh7GZDc0QIYQ/
jM0oo9lTG5oTRk55Hjej4GLUuSAX1gnvxzb632fdQ2+gC68r1fPUu0Yn1a3TIHooW0
09lAcEFed5AqDn/9KtrQq3Fzbceod36m/0bvNuTFGLIJK/GpqFLErfK/Tb/
iPxLHePMCKHDC8qp8W
     -e role=worker \
     --network=host \
     -p 11090:11090 \
     hub.msoftgroup.ru:5000/mf-svc:8.2-61
[Install]
WantedBy=default.target
```

- 4. Поменяйте значение connectionString на полученное на предыдущем этапе
- 8. Создание файла сервиса mflash-ws

1. Шифрование строки подключения к базе данных

Для шифрования строки подключения к базе данных используйте следующую команду:



Скопируйте зашифрованную строку подключения и используйте её для переменной **DBConn** в следующем шаге.

h01:@#4124 sslmode=disable' +lKYQ7PqIKgoqrwoZo8hQZBKXmf2E9GFFvVibXy7Bli1X3IJS8kMNvgS/ztHgAHToMMHuKrz+t78cxB6HA5KR/YgoQEz9yjGG51djRnIRL7wrw1lumh1GnxQBGFOq2QOphLFWVAEd6W4Trb8Dcdg1a1+ZoMIZ7VQGNHCfQRd

Создание файла сервиса mflash-ws

1. Создайте файл сервиса для systemd:

vim /etc/systemd/system/mflash-ws.service

2. Создайте файл сервиса для systemd:

[Unit] Description=MFlash-WS After=docker.service Requires=docker.service
[Service] TimeoutStartSec=0
ExecStartPre=-/usr/bin/docker container inspect %n >/dev/null 2>&1 && -/usr/bin/docker exec %n stop true
ExecStart=/usr/bin/docker runrmname %n \ -e MFlashHost=https://192.168.1.89 \
-e MFlashToken=asdwerejwekjhrekhjdsgjhgjdsfsf \ -e MFlashInsecure=true \
-e Port=8083 \ -v /opt/mflash/ssl/jwt_public.pem:/app/public.pem \ -e JWTPublic="/app/public.pem" \ -e Host=0.0.0.0 \
-e
DBConn="arnlw6ALfW7iVTIDryYID6W3YadBkZ3Bt1BgIR9b44j8ydRyr4re/ 4x9tZJmM8NSFfiQ3gqqeiQZ7Sm3pjD4GbHtfg0In9DVgVdCtXB5He4P6o/ IuTbiS98X80JqpBTn6k5KUKu3bPzhX31cTffClobWVYDIV+bxJXsOjw==" \



WantedBy=default.target

Установка планировщика PG_CRON

На выбор вы можете установить один из двух планировщиков. Мы рекомендуем ставить PG_CRON. В случае если установить PG_CRON нет возможности, то установите планировщик systemd(mflash-db-job).

Настройка PG_CRON

- 1. Установка необходимых пакетов
- apt-get install postgresql15-pg_cron -y

Haстройка PostgreSQL

2. Отредактируйте файл postgresql.conf:

vim /etc/postgresql/15/main/postgresql.conf

3. Добавьте или измените параметры:

```
shared_preload_libraries = 'pg_cron'
cron.database_name = 'mflash'
cron.use_background_workers = on
max_worker_processes = 10
```

4. Перезапустите PostgreSQL:

sudo systemctl restart postgresql

5. Подключитесь к PostgreSQL:

psql -U postgres -h 127.0.0.1 (указать нужный адрес)

6. Переключитесь на базу mflash:

\c mflash

7. Изменить search_path по умолчанию для пользователя postgres

ALTER ROLE postgres IN DATABASE mflash SET search_path TO mflash,public;

8. Создайте расширение pg_cron:

```
CREATE EXTENSION pg_cron;
```

```
9. Выполнение скрипта pg_cron_prepare.sql
```

Убедитесь, что у вас есть файл pg_cron_prepare.sql. Выполните:

```
psql -h 127.0.0.1 -U postgres -d mflash -a -
f /ПутьДоФайла/pg_cron_prepare.sql
```

10. Проверка работы pg_cron

SELECT * FROM cron.job_run_details ORDER BY runid DESC;

Также чтобы проверить, что утилита установлена корректно укажите команду: \dx.

CREATE EX mflash=#	TENSION \dx						
List of installed extensions							
Name	Version	Schema	Description				
pg_cron plpgsql	1.6 1.0	pg_catalog pg_catalog	Job scheduler for PostgreSQL PL/pgSQL procedural language				

Hастройка systemd планировщика (если pg_cron недоступен)

1. Создание файла сервиса

nano /etc/systemd/system/mflash-db-job.service

Добавьте содержимое:

[Unit] Description=mflash-jobs After=postgresql.service Requires=postgresql.service

[Service] TimeoutSec=300 User=mflash Restart=always ExecStart=/usr/lib/postgresql/15/bin/psql -h 127.0.0.1 -U postgres -d mflash -w -c "call mflash.schedulerexecutor();" RestartSec=10

[Install] WantedBy=default.target

2. Создание пользователя mflash

sudo useradd -m mflash

3. Настройка файла .pgpass

nano /home/mflash/.pgpass

4. Добавьте строку:

127.0.0.1:5432:mflash:postgres:mflash01

- **127.0.0.1** хост базы данных (localhost).
- **5432** порт PostgreSQL.
- **mflash** имя базы данных.
- **postgres** имя пользователя базы данных.
- mflash01 пароль пользователя.
- 5. Установите права доступа:

```
chown mflash:mflash /home/mflash/.pgpass
chmod 0600 /home/mflash/.pgpass
```

Запуск контейнеров

1. Перезагрузка демона systemd

systemctl daemon-reload

2. Запуск сервисов в порядке

```
systemctl start mflash-web
systemctl start mflash-ws
```

```
systemctl start nginx-docker
systemctl start mflash-service
systemctl start mflash-worker
```

3. Если вы используете планировщик systemd, запустите:

```
systemctl start mflash-db-job
```

Проверка работы сервисов

После запуска убедитесь, что все сервисы работают корректно, и проверьте их статус:

systemctl status имя_сервиса

Просмотр логов сервисов

1. При возникновении проблем с запуском сервисов просмотрите логи:

journalctl -и имя_сервиса -f

2. Например:

journalctl -u mflash-web -f

Изменение параметров файла сервисов

В случае изменения параметра сервисов необходимо перезагружать

```
systemctl daemon-reload &&
systemctl restart mflash-web.service &&
journalctl -fu mflash-web.service
```