VisionLabs
MACHINES CAN SEE

VISIONLABS FACE STREAM

Инструкция по установке

ООО «ВижнЛабс»
123458, г. Москва, ул. Твардовского д. 8, стр. 1

+7 (499) 399 3361
info@visionlabs.ru
www.visionlabs.ru

Содержание

Глоссарий 3
Аппаратные требования 4
Минимальные аппаратные требования 4
Программные требования 5
Введение 6
1 Подготовка к запуску 7
1.1 Распаковка архива 7
1.2 Создание символической ссылки 7
1.3 Создание директории для сохранения логов 7
1.4 Установка Docker 8
1.5 Установка зависимостей для GPU 8
1.6 Вход в registry 9
1.7 Активация лицензии 9
1.7.1 Онлайн активация лицензии 10
1.7.1.1 Процесс активации лицензии онлайн 10
1.7.2 Офлайн активация лицензии 10
1.7.2.1 Процесс офлайн активации 10
1.7.3 Описание параметров лицензионного файла 12
2 Запуск Face Stream 13
2.1 Запуск Face Stream в обычном режиме 14
2.1.1 Настройка Face Stream перед запуском 14
2.1.2 Команда запуска контейнера в обычном режиме 14
2.1.2.1 Ключи запуска в обычном режиме 15
2.1.2.2 Расшифровка параметров запуска контейнера 15
2.2 Запуск Face Stream в серверном режиме с конфигурационными файлами 17
2.2.1 Настройка Face Stream перед запуском 17
2.2.2 Команда запуска контейнера в серверном режиме с конфигурационными файлами 17
2.2.2.1 Ключи запуска в серверном режиме с конфигурационными файлами 18
2.2.2.2 Расшифровка параметров запуска контейнера 18

Глоссарий

Термин	Значение термина
Трек	Информация о положении объекта (лица) одного человека на последовательности кадров. Если объект покидает зону кадра, то трек прерывается не сразу. Некоторое время он ожидает возвращения объекта в кадр. Если объект вернулся, то трек продолжается.
Трекинг	Функция отслеживания объекта (лица) на последовательности кадров.
Детекция	Обнаружение лица в кадре.
Ракурс	Степень поворота головы (в градусах) по каждой из трех осей вращения (наклон вверх/вниз относительно горизонтальной оси; наклон влево/вправо относительно вертикальной оси; поворот относительно вертикальной оси).
Лучший кадр, лучшая детекция	Лучший кадр выбирается из всех кадров одного трека. Основными условиями выбора лучшего кадра являются приемлемое качество изображения и наличие на нём лица с наилучшим ракурсом. Условия выбора лучшего кадра задаются в настройках Face Stream.
Портрет	Фрагмент изображения лица с кадра, отобранного в соответствии с настройками алгоритма и максимально приближенный к требованиям ГОСТ 19794-5-2006 / ISO IEC 19794-5 2005(E).
Биометрический образец	Специальный формат изображения для работы с LUNA PLATFORM 5 с выровненным по горизонтали лицом. Такое изображение содержит всю необходимую информацию о лице и при этом быстро обрабатывается системой.
Биометрический шаблон	Набор уникальных свойств, получаемых в LUNA PLATFORM из биометрического образца.

Аппаратные требования

Минимальные аппаратные требования

Дальнейшие минимальные требования приведены для использования одного экземпляра FaceStream.

Для корректной работы приложения аппаратное обеспечение должно отвечать следующим минимальным требованиям:

- CPU с частотой 2 ГГц и выше;
- 4 Гб оперативной памяти и выше;
- 10 Гб свободного места на жестком диске.
- Доступ к Интернету (для контейнеров и дополнительных загрузок ПО).

На аппаратные требования влияют несколько факторов:

- Количество обрабатываемых видеопотоков;
- Частота и разрешение кадров видеопотоков;
- Параметры настройки FaceStream. Настройки по умолчанию являются наиболее универсальными. В зависимости от условий эксплуатации приложения с помощью их значений можно повлиять на качество или производительность.

Следует подбирать аппаратное обеспечение на основе вышеперечисленных факторов.

FaceStream также может работать в режиме ускорения вычислений за счет:

Использование ресурсов видеокарты
Вычисления с использованием видеокарты поддерживаются только для детектора FaceDetV3. См. параметр “defaultDetectorType” в “faceengine.conf”.
Требуется минимум 6Гб оперативной или выделенной видеопамяти. Рекомендуется 8 Гб VRAM или более.
Поддерживаются архитектуры Pascal, Volta, Turing.
Требуется Compute Capability 6.1 или выше.

CUDA версии 11.2.1 уже установлена в Docker контейнере FaceStream. Рекомендуемые драйверы NVIDIA - r450, r455.
В данный момент для одного экземпляра FaceStream поддерживается только одна видеокарта.

Использование AVX2 инструкций

Требуется CPU с поддержкой AVX2.
Система автоматически определяет наличие инструкций и запускается в оптимальном режиме.

Программные требования

Для установки Face Stream должны выполняться следующие программные требования:

- Для запуска Face Stream может использоваться RedOS (РЕД ОС) версии 7.3 и выше, CentOS версии 7.8 и выше.

Введение

Данный документ описывает процесс установки, настройки и запуска **VisionLabs Face Stream**, а также содержит аппаратные и программные требования к ПО. Описание параметров конфигурационных файлов приведено в руководстве администратора.

1 Подготовка к запуску

1.1 Распаковка архива

Рекомендуется переместить архив в предварительно созданную директорию для FaceStream и распаковать архив в этой директории.

> Указанные команды следует выполнять под пользователем root.

Создайте директорию для Face Stream.

```
mkdir -p /var/lib/fs
```

Переместите архив в созданную директорию. Предполагается, что архив сохранён на сервере в директории “/root”.

```
mv /root/facestream_docker_v.5.0.6.zip /var/lib/fs/
```

Перейдите в директорию.

```
cd /var/lib/fs/
```

Установите утилиту unzip, если она ещё не установлена.

```
yum install unzip
```

Распакуйте архив.

```
unzip facestream_docker_v.5.0.6.zip
```

Перед запуском Face Stream потребуется выполнить его настройку.

В распакованном архиве находятся конфигурационные файлы, которые необходимы для запуска Face Stream. Описание параметров из данных файлов приведены далее в документе.

1.2 Создание символической ссылки

Создайте символическую ссылку. Символическая ссылка указывает на директорию, в которой хранятся файлы для запуска нужной версии программного продукта.

```
ln -s facestream_docker_v.5.0.6 fs-current
```

1.3 Создание директории для сохранения логов

Следует создать директорию для сохранения логов Face Stream на сервере перед запуском Face Stream.

```
mkdir -p /var/lib/fs/fs-current/logs/
```

1.4 Установка Docker

Docker требуется для запуска контейнера Face Stream.

Установка Docker описана в официальной документации: https://docs.docker.com/engine/install/centos/.

> Если на сервере уже установлен Docker последней версии, то выполнять повторную установку не требуется.

Ниже перечислены команды для быстрой установки:

> При возникновении проблем с установкой обратитесь к официальной документации Docker.

Установите зависимости.

```
yum install -y yum-utils device-mapper-persistent-data lvm2
```

Добавьте репозиторий.

```
yum-config-manager --add-repo https://download.docker.com/linux/centos/docker-ce.repo
```

Установите Docker.

```
yum -y install docker-ce docker-ce-cli containerd.io
```

Запустите Docker.

```
systemctl start docker

systemctl enable docker
```

Проверьте, запущен ли Docker.

```
systemctl status docker
```

1.5 Установка зависимостей для GPU

Пропустите данный раздел если не собираетесь использовать Face Stream с GPU.

Для использования GPU с Docker контейнерами необходимо установить NVIDIA Container Toolkit.

> CUDA версии 11.2.1 установлена в Docker контейнере Face Stream.

Пример установки приведен ниже.

```
distribution=$(. /etc/os-release;echo $ID$VERSION_ID)
```

```
curl -s -L https://nvidia.github.io/nvidia-docker/$distribution/nvidia-docker.repo
| tee /etc/yum.repos.d/nvidia-docker.repo

yum install -y nvidia-container-toolkit

systemctl restart docker
```

Проверьте работу NVIDIA Container toolkit, запустив базовый контейнер CUDA (он не входит в дистрибутив Face Stream, его необходимо загрузить из Интернета):

```
docker run --rm --gpus all nvidia/cuda:11.2.1-base nvidia-smi
```

Для дополнительной информации см. следующую документацию:

https://github.com/NVIDIA/nvidia-docker#centos-7x8x-docker-ce-rhel-7x8x-docker-ce-amazon-linux-12.

Извлечение атрибутов на GPU разработано для максимальной пропускной способности. Выполняется пакетная обработка входящих изображений. Это снижает затраты на вычисления для изображения, но не обеспечивает минимальную задержку для каждого изображения.
GPU-ускорение разработано для приложений с высокой нагрузкой, где количество запросов в секунду достигает тысяч. Нецелесообразно использовать ускорениse GPU в сценариях с небольшой нагрузкой, когда задержка начала обработки имеет значение.

1.6 Вход в registry

При запуске контейнера необходимо указать ссылку на образ, необходимый для запуска контейнера. Этот образ загружается из VisionLabs registry. Перед этим необходима авторизация.

```
 docker login dockerhub.visionlabs.ru --username <username> --password <password>
```

Введите логин <username> и пароль <password>. Получить их можно из VisionLabs.

1.7 Активация лицензии

Данный раздел описывает активацию лицензии.

Активация лицензии выполняется после того, как комплект поставки Face Stream был разархивирован. Следует изменять конфигурационный файл “license.conf” в распакованном комплекте поставки Face Stream, на который указывает символическая ссылка “fs-current”.

Полученная лицензия может быть использована только на том устройстве, где она была активирована.
Помните, что неправильная конфигурация может привести к возникновению ошибок. Обратите внимание на то, что вы настраиваете и как.

1.7.1 Онлайн активация лицензии

Онлайн активация выполняется если на устройстве, где будет установлен Face Stream, есть доступ к Интернету.

Настройки для получения лицензии задаются в файле “license.conf”. Это XML файл с определённым форматированием.

Файл “license.conf” находится в директории “license” в поставке.

Данный файл необходим для получения лицензии. Следует заполнить его актуальными значениями перед запуском Face Stream.

Неправильно введённые данные могут привести к проблемам с активацией лицензии на устройстве.

1.7.1.1 Процесс активации лицензии онлайн

Для активации лицензии выполните следующие шаги:

- Запросите значения **Server**, **EID** и **ProductID** у VisionLabs
- Откройте файл “license.conf”
- Внесите в него полученные параметры
- Сохраните изменения в файле “license.conf”

Для добавления конфигурационного файла в контейнер используется следующая строка:

```
-v /var/lib/fs/fs-current/license/license.conf:/srv/facestream/data/license.conf \
```

Необходимый конфигурационный файл будет добавлен в контейнер Face Stream при его запуске. См. описание запуска в разделе “Запуск Face Stream”.

После активации лицензии в контейнере будет создан файл лицензии с именем, указанным в параметре “Filename” в license.conf.

1.7.2 Офлайн активация лицензии

Офлайн активация выполняется если на устройстве, где будет установлен Face Stream, нет доступа к Интернету.

В этом случае следует создать отпечаток устройства и использовать его для получения лицензии на другом устройстве, на котором есть доступ к интернету.

1.7.2.1 Процесс офлайн активации

Для активации лицензии выполните следующие шаги:

- Запросите адрес сайта для активации лицензии, EID и ProductID у VisionLabs.

Выполните следующие шаги на устройстве, на котором следует активировать Face Stream:

- Перейдите в директорию “license” комплекта поставки;
- Откройте файл “license.conf”.
- Введите полученный EID и ProductID;
- Укажите “http://localhost” в поле “Server”;
- Сохраните изменения в файле “license.conf”;
- Запустите утилиту “FingerprintViewer” из контейнера Face Stream для создания отпечатка вашего устройства:

```
docker run \
--rm \
--network=host \
-w /srv/facestream \
-v /var/lib/fs/fs-current/license/license.conf:/srv/facestream/data/license.co
nf \
--entrypoint /srv/facestream/FingerprintViewer \
dockerhub.visionlabs.ru/luna/facestream:v.5.0.6
```

- Отпечаток будет выведен в консоль. Скопируйте и сохраните его.

Выполните следующие шаги на устройстве, на котором есть доступ в Интернет:

- Перейдите на веб-сайт для получения лицензии (адрес веб-сайта был получен на первом шаге данной инструкции);
- Введите свой EID для входа на веб-сайт и, используя отпечаток своего устройства, активируйте лицензию;
- Скачайте сертификат лицензии. Обратите внимание, что по умолчанию файл называется “licenseFile.v2c”. Если полученный файл имеет другое имя и формат, следует переименовать его в “licenseFile.v2c”. Данное имя файла задано по умолчанию в “license.conf”.

Выполните следующий шаг на устройстве, на котором следует активировать лицензию:

- Переместите файл “licenseFile.v2c” в директорию “license” поставки Face Stream. Он должен быть расположен рядом с файлом “license.conf”.

Для добавления конфигурационных файлов в контейнер используются следующие строки:

```
-v /var/lib/fs/fs-current/license/license.conf:/srv/facestream/data/license.conf \
-v /var/lib/fs/fs-current/license/licenseFile.v2c:/srv/facestream/data/licenseFile
.v2c \
```

Необходимые файлы должны быть добавлены в контейнер Face Stream при его запуске. См. описание запуска в разделе “Запуск Face Stream”.

1.7.3 Описание параметров лицензионного файла

Для активации лицензии и её последующего использования следует заполнить следующие параметры.

Параметр	Описание	Тип	Значение по умолчанию
Server	URL сервера активации	“Value::String”	(пусто)
EID	Entitlement ID	“Value::String”	(пусто)
ProductID	ID продукта	“Value::String”	(пусто)
Filename	Имя создаваемого файла лицензии	“Value::String”	“licenseFile.v2c”
ContainerMode	Технический параметр	“Value::String”	0
ConnectionTimeout	Время ожидания запроса (в секундах)	“Value::Int1”	120

Server, **EID** и **ProductID** - данная информация должна быть получена у VisionLabs и записана в файл.

При офлайн активации следует указать “http://localhost” в качестве значения для **Server**.

Filename - имя файла, в который будет сохранена сгенерированная лицензия. Максимально допустимое количество символов - 64. Не изменяйте это имя!

ConnectionTimeout устанавливает максимальное время в секундах, в течение которого можно подключиться к сервису лицензирования. С помощью данного параметра можно настроить максимальное время ожидания в соответствии с различными условиями работы, т.к. операция разрешения имени сервера и соединения с ним может занять значительное время, которое зависит от конкретной сети. Время ожидания равное 0 означает отсутствие ограничений на время подключения к сервису лицензирования. `ConnectionTimeout` не может иметь отрицательное значение или значение, превышающее максимальное, т.е. `300` секунд.

ContainerMode является техническим параметром. Не следует изменять его без консультации с VisionLabs.

Пример конфигурационного файла “license.conf” приведён ниже.

```
<section name="Licensing::Settings">
    <param name="Server" type="Value::String" text=""/>
    <param name="EID" type="Value::String" text=""/>
    <param name="ProductID" type="Value::String" text=""/>
    <param name="Filename" type="Value::String" text="licenseFile.v2c"/>
    <param name="ContainerMode" type="Value::Int1" x="0"/>
    <param name="ConnectionTimeout" type="Value::Int1" x="120"/>
</section>
```

2 Запуск Face Stream

Запуск Face Stream возможен одним из следующих способов:

- в обычном режиме
- в серверном режиме с использованием конфигурационных файлов

Примеры запуска Face Stream приведены в разделах ниже.

По умолчанию для вычислений используется CPU. Имеется возможность запустить Face Stream вышеперечисленными способами с использованием GPU. В разделе “Команда запуска контейнера с использованием GPU” приведен пример запуска контейнера в серверном режиме с Configurator с использованием GPU. Для остальных режимов запуск контейнера с использованием GPU аналогичен.
Можно создавать новые файлы конфигураций на основе существующих файлов конфигураций (fs3Config.conf, input.json) и указывать их при запуске Face Stream (в ключах запуска). Таким образом вы можете создать несколько файлов конфигураций для разных целей не изменяя файлы по умолчанию.

Никогда не используйте конфигурационные файлы из предыдущих версий Face Stream! Face Stream не сможет запуститься, или результаты обработки будут отличаться от ожидаемых.

2.1 Запуск Face Stream в обычном режиме

2.1.1 Настройка Face Stream перед запуском

Запуск Face Stream в обычном режиме осуществляется с использованием следующих конфигурационных файлов:

- input.json (см. описание настроек в разделе “Настройка источников видеопотока” руководства администратора)
- fs3Config.conf (см. описание настроек в разделе “Настройка Face Stream” руководства администратора)
- trackengine.conf (см. описание настроек в разделе “Настройка Trackengine” руководства администратора)
- faceengine.conf

Следует предварительно задать все необходимые параметры в данных файлах перед запуском Face Stream.

Face Stream можно запустить в обычном режиме с использованием GPU. В разделе “Команда запуска контейнера с использованием GPU” приведен пример запуска контейнера в серверном режиме с Configurator с использованием GPU.

2.1.2 Команда запуска контейнера в обычном режиме

Примечание. Если была произведена **офлайн** активация лицензии, то во время запуска контейнера необходимо дополнительно указать файл `licenseFile.v2c` с помощью параметра `-v /var/lib/fs/fs-current/license/licenseFile.v2c:/srv/FaceStream/data/licenseFile.v2c`.

Запуск контейнера осуществляется следующим образом:

```
docker run \
-v /var/lib/fs/fs-current/license/license.conf:/srv/facestream/data/license.conf \
-v /var/lib/fs/fs-current/conf/configs/input.json:/srv/facestream/data/input.json
\
-v /var/lib/fs/fs-current/conf/configs/fs3Config.conf:/srv/facestream/data/fs3Conf
ig.conf \
-v /var/lib/fs/fs-current/conf/configs/faceengine.conf:/srv/facestream/data/faceen
gine.conf \
-v /var/lib/fs/fs-current/conf/configs/trackengine.conf:/srv/facestream/data/track
engine.conf \
-v /var/lib/fs/fs-current/logs/:/srv/logs/ \
-v /etc/localtime:/etc/localtime:ro \
--name=facestream \
--network=host \
--restart=always \
--detach=true \
--entrypoint /srv/facestream/FaceStreamStatic \
dockerhub.visionlabs.ru/luna/facestream:v.5.0.6 \
--source-path /srv/facestream/data/input.json \
--config-path /srv/facestream/data/fs3Config.conf \
```

```
--data-dir /srv/facestream/data \
--log-dir /srv/facestream/logs
```

2.1.2.1 Ключи запуска в обычном режиме

Для запуска Face Stream в обычном режиме внутри контейнера используется следующая команда, которая позволяет указать корректные пути до директорий внутри контейнера.

```
--entrypoint /srv/facestream/FaceStreamStatic \
dockerhub.visionlabs.ru/luna/facestream:v.5.0.6 \
--source-path /srv/facestream/data/input.json \
--config-path /srv/facestream/data/fs3Config.conf \
--data-dir /srv/facestream/data \
--log-dir /srv/facestream/logs
```

Доступны следующие ключи:

- `--source-path` – путь к файлу конфигурации источников видеопотока. Следует указать путь к файлу “input.json”.
- `--config-path` – полный путь к файлу настроек приложения “fs3Config.conf”. Если определен данный параметр, то при поиске конфигурационного файла игнорируется путь к данным.
- `--data-dir` – путь к директории с данными детекторов и настройками.
- `--log-dir` – директория для записи файлов логирования.

2.1.2.2 Расшифровка параметров запуска контейнера

`docker run` - команда для запуска выбранного образа в качестве нового контейнера.

`-v` - параметр volume позволяет загружать содержимое серверной папки в объем контейнера. Таким образом содержимое синхронизируется.
Файлы настроек находятся в комплекте поставки Face Stream в директории “conf/configs/” и добавляются в контейнер при запуске следующими командами:

```
-v /var/lib/fs/fs-current/conf/configs/input.json:/srv/facestream/data/input.json
\
-v /var/lib/fs/fs-current/conf/configs/fs3Config.conf:/srv/facestream/data/fs3Conf
ig.conf \
-v /var/lib/fs/fs-current/conf/configs/faceengine.conf:/srv/facestream/data/faceen
gine.conf \
-v /var/lib/fs/fs-current/conf/configs/trackengine.conf:/srv/facestream/data/track
engine.conf \
```

`-v /var/lib/fs/fs-current/license/license.conf:/srv/facestream/data/license.conf \` - позволяет добавить файл для активации лицензии в контейнер Face Stream.

`-v /var/lib/fs/fs-current/logs/:/srv/logs/ \` - позволяет копировать папку с записями (логами) Face Stream на сервер в директорию `/var/lib/fs/fs-current/logs/`. Директорию для хранения логов можно изменить при желании.

`--network=host` - этот параметр указывает, что отсутствует симуляция сети и используется серверная сеть. При необходимости изменить порт для сторонних контейнеров следует заменить эту строку на `-p 5440:5432`. Здесь первый порт `5440` - это локальный порт, а `5432` - это порт, используемый в контейнере.
`/etc/localtime:/etc/localtime:ro` - задает текущий часовой пояс, используемый системой контейнера.
`--name=facestream` - этот параметр задает имя запускаемого контейнера. Имя должно быть уникальным. Если уже существует контейнер с таким же именем, произойдет ошибка.
`--restart=always` - этот параметр определяет политику перезагрузки. Daemon всегда перезагружает контейнер вне зависимости от кода завершения.
`--detach=true` - запуск контейнера в фоновом режиме.

2.2 Запуск Face Stream в серверном режиме с конфигурационными файлами

2.2.1 Настройка Face Stream перед запуском

Запуск Face Stream в серверном режиме с настройками из конфигурационных файлов осуществляется с использованием следующих конфигурационных файлов:

- fs3Config.conf (см. описание настроек в разделе “Настройка Face Stream” руководства пользователя)
- trackengine.conf (см. описание настроек в разделе “Конфигурация Trackengine” руководства пользователя)
- faceengine.conf

Конфигурационный файл input.json не используется при запуске Face Stream. Видеопотоки создаются с помощью API Face Stream после его запуска (см. FaceStreamApi.html).

Следует предварительно задать все необходимые параметры в данных файлах перед запуском Face Stream.

Face Stream можно запустить в серверном режиме с использованием GPU. В разделе “Команда запуска контейнера с использованием GPU” приведен пример запуска контейнера в серверном режиме с Configurator с использованием GPU.

2.2.2 Команда запуска контейнера в серверном режиме с конфигурационными файлами

Примечание. Если была произведена **офлайн** активация лицензии, то во время запуска контейнера необходимо дополнительно указать файл `licenseFile.v2c` с помощью параметра `-v /var/lib/fs/fs-current/license/licenseFile.v2c:/srv/facestream/data/licenseFile.v2c`.

Запуск контейнера осуществляется следующим образом:

```
docker run \
-v /var/lib/fs/fs-current/license/license.conf:/srv/facestream/data/license.conf \
-v /var/lib/fs/fs-current/conf/configs/fs3Config.conf:/srv/facestream/data/fs3Conf
ig.conf \
-v /var/lib/fs/fs-current/conf/configs/faceengine.conf:/srv/facestream/data/faceen
gine.conf \
-v /var/lib/fs/fs-current/conf/configs/trackengine.conf:/srv/facestream/data/track
engine.conf \
-v /etc/localtime:/etc/localtime:ro \
-v /var/lib/fs/fs-current/logs/:/srv/logs/ \
--restart=always \
--detach=true \
--name=facestream \
--network=host \
--env=PORT=34569 \
--entrypoint /srv/facestream/FaceStream \
dockerhub.visionlabs.ru/luna/facestream:v.5.0.6 \
```

```
--config-path /srv/facestream/data/fs3Config.conf \
--data-dir /srv/facestream/data \
--log-dir /srv/facestream/logs \
--http-address http://0.0.0.0:34569
```

2.2.2.1 Ключи запуска в серверном режиме с конфигурационными файлами

Для запуска Face Stream в серверном режиме с конфигурационными файлами внутри контейнера используется следующая команда, которая позволяет указать корректные пути до директорий внутри контейнера.

```
--entrypoint /srv/facestream/FaceStream \
dockerhub.visionlabs.ru/luna/facestream:v.5.0.6 \
--config-path /srv/facestream/data/fs3Config.conf \
--data-dir /srv/facestream/data \
--log-dir /srv/facestream/logs \
--http-address http://0.0.0.0:34569
```

Доступны следующие ключи:

- `--config-path` – полный путь к файлу настроек приложения “fs3Config.cfg”. Если определен данный параметр, то при поиске конфигурационного файла игнорируется путь к данным.
- `--data-dir` – путь к директории с данными детекторов и настройками.
- `--log-dir` – директория для записи файлов логирования.
- `--http-address` - HTTP адрес, который будет прослушивать Face Stream. Задаётся в формате “address:port” (используется только для Face Stream в серверном режиме). На этот адрес пользователь будет отправлять запросы.

  Следует задать внешний IP сервера Face Stream. По умолчанию задано «http://0.0.0.0:34569».

2.2.2.2 Расшифровка параметров запуска контейнера

`docker run` - команда для запуска выбранного образа в качестве нового контейнера.

`-v` - параметр volume позволяет загружать содержимое серверной папки в объем контейнера. Таким образом содержимое синхронизируется.
Файлы настроек находятся в комплекте поставки Face Stream в директории “conf/configs/” и добавляются в контейнер при запуске следующими командами:

```
-v /var/lib/fs/fs-current/conf/configs/fs3Config.conf:/srv/facestream/data/fs3Conf
ig.conf \
-v /var/lib/fs/fs-current/conf/configs/faceengine.conf:/srv/facestream/data/faceen
gine.conf \
-v /var/lib/fs/fs-current/conf/configs/trackengine.conf:/srv/facestream/data/track
engine.conf \
```

`-v /var/lib/fs/fs-current/license/license.conf:/srv/facestream/data/license.conf \`
- позволяет добавить файл для активации лицензии в контейнер Face Stream.

`-v /var/lib/fs/fs-current/logs/:/srv/logs/ \` - позволяет копировать папку с записями (логами) Face Stream на сервер в директорию `/var/lib/fs/fs-current/logs/`. Директорию для хранения логов можно изменить при желании.
`--network=host` - этот параметр указывает, что отсутствует симуляция сети и используется серверная сеть. При необходимости изменить порт для сторонних контейнеров следует заменить эту строку на `-p 5440:5432`. Здесь первый порт `5440` - это локальный порт, а `5432` - это порт, используемый в контейнере.
`/etc/localtime:/etc/localtime:ro` - задает текущий часовой пояс, используемый системой контейнера.
`--name=facestream` - этот параметр задает имя запускаемого контейнера. Имя должно быть уникальным. Если уже существует контейнер с таким же именем, произойдет ошибка.
`--restart=always` - этот параметр определяет политику перезагрузки. Daemon всегда перезагружает контейнер вне зависимости от кода завершения.
`--detach=true` - запуск контейнера в фоновом режиме.