




VISIONLABS LP5 ISO QUALITY CHECK

Инструкция по установке

ООО «ВижнЛабс»

123458, г. Москва, ул. Твардовского д. 8, стр. 1

 +7 (499) 399 3361

 info@visionlabs.ru

 www.visionlabs.ru

Оглавление

Глоссарий	3
Введение	4
Требования ПО	6
Системные требования	6
Стандартные порты для сервисов	7
Названия сервисов в Configurator	8
Общая информация	9
Перед установкой	12
Распаковка дистрибутива	12
Создание символической ссылки	12
SELinux и Firewall	12
Активация лицензионного ключа	13
Лицензирование	13
Установка Docker	15
Установка Docker Compose	16
Авторизация в registry	16
Запуск платформы	17
Конфигурация контейнера User Interface 4	17
Запуск сервисов	17
Команды Docker	18
Показать контейнеры	18
Копировать файлы в контейнер	18
Ввод контейнера	18
Имена образов	18
Удаление образа	18
Остановка контейнера	19
Удаление контейнера	19
Логи сервисов	19
Отображение логов сервисов	19

Глоссарий

Термин	Определение
LUNA PLATFORM 5	Система распознавания лиц. ПО VisionLabs LUNA PLATFORM 5 включено в Единый реестр российских программ для электронных вычислительных машин и баз данных Запись в реестре №12557 от 14.01.2022 произведена на основании поручения Министерства цифрового развития, связи и массовых коммуникаций Российской Федерации от 14.01.2022 по протоколу заседания экспертного совета от 10.01.2022 №П10-12пр

Сокращенное название	Расшифровка
VL-LP5-QC	VISIONLABS LP5 ISO QUALITY CHECK
LP	LUNA PLATFORM 5
DB	База данных
UI	Пользовательский интерфейс
SW	Программное обеспечение
GC	Сбор мусора
API	LUNA PLATFORM 5 API
Faces	LUNA PLATFORM 5 Faces
Image Store	LUNA PLATFORM 5 Image Store
Matcher	LUNA PLATFORM 5 Matcher
Events	LUNA PLATFORM 5 Events
Sender	LUNA PLATFORM 5 Sender
Handlers	LUNA PLATFORM 5 Handlers
Python Matcher	LUNA PLATFORM 5 Python Matcher
Python Matcher Proxy	LUNA PLATFORM 5 Python Matcher Proxy
Backport 3	LUNA PLATFORM 5 Backport 3
Backport 4	LUNA PLATFORM 5 Backport 4
Admin	LUNA PLATFORM 5 Admin
Configurator	LUNA PLATFORM 5 Configurator
Tasks	LUNA PLATFORM 5 Tasks
Licenses	LUNA PLATFORM 5 Licenses
User Interface 3	LUNA PLATFORM 5 User Interface 3
User Interface 4	LUNA PLATFORM 5 User Interface 4
LPE	LUNA PLATFORM 5 Enterprise

Введение

VisionLabs LP5 ISO Quality Check является дополнительным функциональным модулем продукта VisionLabs LUNA PLATFORM 5. Для установки и использования VisionLabs LP5 ISO Quality Check требуется установить и настроить LUNA PLATFORM 5.

Данный документ описывает установку и использование Docker Compose для развертывания LUNA PLATFORM 5.

Docker позволяет создать контейнер, в котором уже имеется требуемый сервис, программная среда для него и минимальный набор необходимых инструментов.

У каждого сервиса LUNA PLATFORM 5 есть собственный образ. Образы Docker являются основой для контейнеров. Каждый контейнер включает в себя библиотеки, необходимые для функционирования сервисов и параметров, используемых во время работы контейнера.

Docker Compose используется для автоматического развертывания контейнеров. Сценарий Docker Compose из данного дистрибутива используется для развертывания сервисов LUNA PLATFORM 5 на одном сервере.

Следует выполнить дополнительные действия для активации лицензионного ключа и установки требуемого ПО перед запуском скрипта Compose. Эти действия описаны в данном руководстве.

Считается, что установка выполняется на сервере, где LP не была установлена.

Образы Docker для контейнеров LP необходимы для установки. Требуется интернет-соединение на сервере для загрузки образов Docker или они должны быть загружены на любом другом устройстве и перенесены на сервер. Требуется вручную задать логин и пароль для загрузки образов Docker.

Администратор должен вручную настроить Firewall и SELinux на сервере. В данном документе не описывается их настройка.

В данной установке не предполагается резервное копирование или копирование баз данных для данных LP.

Данный документ включает в себя пример развертывания LUNA PLATFORM 5 с помощью скрипта Compose. Он использует минимальные ресурсы, необходимые для демонстрации работы LUNA PLATFORM 5 и не может использоваться для реальных рабочих целей.

См. файл “docker-compose.yml” и другие файлы в директории “example-docker” для информации о запущенных сервисах и выполненных действиях.

Рекомендуется использовать сервисы оркестрации для коммерческого использования LP. Их использование не описано в данном руководстве.

Примечания о скрипте Docker Compose. Скрипт:

- тестируется с использованием настроек сервисов по умолчанию.
- не поддерживает использование GPU для вычислений LP.

- не предназначен для использования в целях масштабирования LP:
 - Не используется для развертывания сервисов LP на нескольких серверах.
 - Не используется для развертывания и балансирования нескольких сервисов LP на одном сервере.
- запускает базы данных по умолчанию и не включает в себя встроенную возможность изменять используемые базы данных.
- не обеспечивает возможность использования внешних баз данных, уже установленных на сервере.
- не выполняет миграции из предыдущих версий LP и обновления предыдущих сборок LP.

Можно написать собственный сценарий, который разворачивает и конфигурирует все необходимые сервисы. Данный документ не включает информацию о создании сценариев и не обучает использованию Docker. Обратитесь к документации Docker для получения подробной информации о Docker и Docker Compose:

<https://docs.docker.com>

Все представленные команды необходимо исполнять с использованием Bash-оболочки (когда команды запускаются напрямую на сервере) или Putty (при удалённом подключении к серверу). Представленные команды тестировались только с помощью этих утилит. Использование других оболочек или эмуляторов может привести к ошибкам при исполнении команд.

Требования ПО

Для запуска LUNA PLATFORM 5 с Модулем VL-LP5-QC может использоваться RedOS (РЕД ОС) версии 7.3 и выше, CentOS версии 7.8 и выше.

Системные требования

Следующие минимальные системные требования необходимы для установки программного пакета LUNA PLATFORM 5 с Модулем VL-LP5-QC:

- CPU Intel, минимум 4 физических ядра с тактовой частотой 2.0 GHz или выше. Требуется поддержка набора инструкций AVX2 для CPU;
- RAM DDR3 (рекомендуется DDR4), 8 Гб или выше.
- Свободное место на диске - минимум 80 Гб.

Рекомендуется использование SSD для баз данных и хранилищ Image Store.

- Доступ к Интернету (для контейнеров и дополнительных загрузок ПО).

Примечание Приведенная выше конфигурация обеспечит минимальную мощность для работы ПО, но для использования системы в продуктивном контуре этого недостаточно. Требования для использования системы в продуктивном контуре рассчитываются в зависимости от предполагаемой нагрузки.

GPU

Для ускорения GPU необходим NVIDIA GPU. Поддерживаются следующие архитектуры:

- Pascal или более новые.

Требуется минимум 6Гб оперативной или выделенной видеопамяти. Рекомендуется 8 Гб VRAM или более.

На сервере должно быть установлено ПО CUDA версии 11.2. Рекомендуемые драйверы NVIDIA - r450, r455.

ПО CUDA версии 11.2 уже установлено в Docker контейнере в сервисе Handlers.

Стандартные порты для сервисов

Таблица 1. Стандартные порты для сервисов

Название сервиса	Порт
LUNA PLATFORM 5 API	5000
LUNA PLATFORM 5 Admin	5010
LUNA PLATFORM 5 Image Store	5020
LUNA PLATFORM 5 Faces	5030
LUNA PLATFORM 5 Events	5040
LUNA PLATFORM 5 Tasks	5050
LUNA PLATFORM 5 Tasks Worker	5051
LUNA PLATFORM 5 Configurator	5070
LUNA PLATFORM 5 Sender	5080
LUNA PLATFORM 5 Handlers	5090
LUNA PLATFORM 5 Python Matcher	5100
LUNA PLATFORM 5 Licenses	5120
LUNA PLATFORM 5 Backport 4	5130
LUNA PLATFORM 5 Backport 3	5140
LUNA PLATFORM 3 User Interface	4100
LUNA PLATFORM 4 User Interface	4200
Oracle DB	1521
PostgreSQL	5432
Redis DB	6379
InfluxDB	8086

Схема взаимодействия сервисов приведена в документе с описанием функциональных характеристик программного обеспечения.

Названия сервисов в Configurator

Таблица ниже включает в себя названия сервисов в сервисе Configurator. Данные параметры используются для конфигурации сервисов.

Таблица 2. Названия сервисов в сервисе Configurator в поле “Service name”

Сервис	Название сервиса в Configurator
API	luna-api
Licenses	luna-licenses
Faces	luna-faces
Image Store	luna-image-store
Tasks	luna-tasks
Events	luna-events
Sender	luna-sender
Admin	luna-admin
Handlers	luna-handlers
Python Matcher	luna-python-matcher
Backport 3	luna-backport3
Backport 4	luna-backport4

Настройки для сервиса Configurator устанавливаются в его конфигурационном файле.

Общая информация

Рекомендуется внимательно ознакомиться с данным документом. Это поможет разобраться с тем, из каких компонентов состоит LUNA PLATFORM 5 и какие задачи они решают.

Развертывание следует выполнять в порядке, указанном в данном документе.

Для получения справки обратитесь к документации по Docker compose:

<https://docs.docker.com/compose/install/>

Файл лицензии требуется для активации LUNA PLATFORM 5. Этот файл предоставляется компанией VisionLabs исключительно по запросу.

Все действия, описанные в данном руководстве, должны выполняться пользователем **root**. В данном документе не описывается создание пользователя с привилегиями администратора и последующая установка, выполняемая этим пользователем.

Все сервисы можно разделить на основные и дополнительные. Запустить и использовать LP без основных сервисов нельзя, тогда как дополнительные сервисы не обязательны для запуска LP, но предоставляют больше возможностей. У большинства сервисов имеется собственная база данных или файл для хранения данных.

Таблица 3. Основные сервисы

Сервис	Описание	База данных
API	Основной интерфейс доступа для работы с LP. Получает запросы, распределяет задачи между другими сервисами LP.	
Handlers	Распознает лица на изображениях, получает параметры лиц и создает биометрические образцы. Получает из образцов дескрипторы. Получает базовые атрибуты изображений. Создает и хранит обработчики	PostgreSQL/ Oracle
Python Matcher	Выполняет операции сравнения дескрипторов	
Faces	Создает лица, списки и атрибуты. Сохраняет эти объекты в базе данных. Позволяет другим сервисам получать требуемые данные из базы данных	PostgreSQL/ Oracle, Redis

Image Store	Хранит: биометрические образцы, отчеты о долгом выполнении задач, создаваемые кластеры и дополнительные метаданные	Local storage/ Amazon S3
Events	Хранит события в базе данных. Этот сервис можно отключить, но рекомендуется его использовать, в случае если вы планируете сохранять события	Vertica/ PostgreSQL
Licenses	Проверяет данные лицензии и возвращает информацию о них.	
Admin	Позволяет выполнять общие административные процедуры	PostgreSQL/ Oracle
Configurator	Хранит конфигурации всех сервисов в одном месте	PostgreSQL/ Oracle
Tasks	Выполняет длительные задачи, такие как сбор мусора, получение дескрипторов с помощью новой версии нейросети, кластеризация	PostgreSQL/ Oracle

Таблица 2. Дополнительные сервисы

Сервис	Описание	База данных
Sender	Отправляет уведомления о создаваемых событиях через веб-сокеты	Redis
Backport 3	Данный сервис используется для обработки запросов LUNA PLATFORM 3 Enterprise с использованием LUNA PLATFORM 5.	PostgreSQL/ Oracle
Backport 4	Данный сервис используется для обработки запросов LUNA PLATFORM 4 Enterprise с использованием LUNA PLATFORM 5.	

User Interface 3	Пользовательский интерфейс, используемый для визуального представления возможностей, предоставляемых сервисом Backport 3. Он не включает в себя весь функционал LPE 3.
User Interface 4	Пользовательский интерфейс, используемый для визуального представления возможностей, предоставляемых сервисом Backport 4. Он не включает в себя весь функционал LPE 4.

Дополнительные очереди сообщений и базы данных требуются для LUNA PLATFORM 5. Основная информация об этом ПО содержится в данном документе.

Следующие сторонние службы используются по умолчанию с LUNA PLATFORM 5.

- PostgreSQL используется в качестве базы данных по умолчанию для сервисов Faces, Configurator, Events, Handlers, Tasks, Admin и Backport3.
 - Также возможно использование базы данных Oracle DB вместо PostgreSQL,
 - Для сервиса Events можно использовать Vertica DB вместо PostgreSQL;
- Для сервисов Faces и Sender используется Redis DB;
- Для мониторинга используется InfluxDB.

Следующие версии баз данных рекомендованы к использованию с LP:

- PostgreSQL: 12.2
- Oracle: 11.2
- Vertica: 9.0.0

Установка и конфигурация Oracle и Vertica не описываются в данном руководстве.

Balancers и другие программы могут использоваться при масштабировании системы для обеспечения отказоустойчивости. Их конфигурация не описывается в данном руководстве.

Перед установкой

Следует переключиться на пользователя **root** перед началом установки!

Распаковка дистрибутива

Дистрибутив представляет собой архив **luna_v.5.18.0**, где **5.18.0** это числовой идентификатор, обозначающий версию LUNA PLATFORM 5.

Архив включает в себя все компоненты, требуемые для установки и использования. Он не включает в себя зависимости из стандартного репозитория ОС. Они доступны в интернете.

Переместите установочный файл и файл лицензии в директорию на вашем сервере перед установкой. Например, переместите файлы в директорию `/root/`. В ней не должно быть никакого другого дистрибутива или файлов лицензии кроме целевых.

Создайте директорию для распаковки файла дистрибутива.

```
mkdir -p /var/lib/luna
```

Переместите дистрибутив в созданную директорию.

```
mv /root/luna_v.5.18.0.zip /var/lib/luna
```

Установите приложение для распаковки архива при необходимости

```
yum install -y unzip
```

Откройте папку с дистрибутивом

```
cd /var/lib/luna
```

Распакуйте файлы

```
unzip luna_v.5.18.0.zip
```

Создание символической ссылки

Создайте символическую ссылку. Она показывает, что актуальная версия файла дистрибутива используется для запуска программного пакета.

```
ln -s luna_v.5.18.0 current
```

SELinux и Firewall

SELinux и Firewall необходимо настроить так, чтобы они не блокировали сервисы LUNA PLATFORM 5.

Конфигурация SELinux и Firewall не описываются в данном руководстве.

Если SELinux и Firewall не настроены, установка невозможна.

Активация лицензионного ключа

Сервис HASP используется для лицензирования LUNA PLATFORM 5. Без лицензии невозможно запускать и использовать сервисы LUNA.

Лицензионные ключи предоставляются компанией VisionLabs по запросу отдельно от поставки.

Сетевая лицензия требуется для использования LUNA PLATFORM 5 в Docker контейнерах.

Лицензионный ключ создается с помощью отпечатка системы. Этот отпечаток создается на базе информации об аппаратных характеристиках сервера. Таким образом, полученный лицензионный ключ будет работать только на том же сервере, с которого был получен отпечаток системы.

Существует вероятность, что потребуется новый лицензионный ключ при внесении каких-либо изменений на сервере лицензии.

Последовательность действий:

- Установите на сервер утилиту HASP. Обычно утилита HASP устанавливается на отдельный сервер;
- Запустите утилиту HASP;
- Создайте отпечаток системы для вашего сервера и отправьте его в VisionLabs;
- Активируйте свой ключ, полученный от VisionLabs;
- Укажите адрес вашего сервера HASP в специальном файле.

Лицензирование

LP использует утилиту HASP определённой версии. Если на сервере установлена более старая версия утилиты, её следует удалить перед установкой новой версии. См. раздел [“Удаление утилиты LP HASP”](#).

Установка утилиты для LP

Перейдите в директорию.

```
cd /var/lib/luna/current/extras/hasp/
```

Установите утилиту на сервер.

```
yum -y install /var/lib/luna/current/extras/hasp/aksusbd-*.rpm
```

Запустите утилиту.

```
systemctl daemon-reload
systemctl start aksusbd
systemctl enable aksusbd
```

```
systemctl status aksusbd
```

Конфигурация утилиты

Осуществить конфигурацию утилиты HASP можно с помощью файла “/etc/hasplm/hasplm.ini”.

Примечание! Не выполняйте это действие, если INI файл для утилиты HASP уже сконфигурирован.

Удалите старый файл настроек, если необходимо.

```
rm -rf /etc/hasplm/hasplm.ini
```

Скопируйте INI файл с конфигурациями. Параметры не описаны в данном документе.

```
cp /var/lib/luna/current/extras/hasp/hasplm.ini /etc/hasplm/
```

Добавление библиотеки LP

Скопируйте библиотеку LP (x32 and x64). Она требуется для использования лицензионного ключа LP.

```
cp /var/lib/luna/current/extras/hasp/haspvlib_111186.so /var/hasplm/
```

```
cp /var/lib/luna/current/extras/hasp/haspvlib_x86_64_111186.so /var/hasplm/
```

Перезапустите утилиту

```
systemctl restart aksusbd
```

Создание отпечатка системы для LUNA PLATFORM

Перейдите в директорию.

```
cd /var/lib/luna/current/extras/hasp/
```

Добавьте разрешения для скрипта.

```
chmod +x LicenseAssist
```

Запустите скрипт.

```
./LicenseAssist fingerprint fingerprint_111186.c2v
```

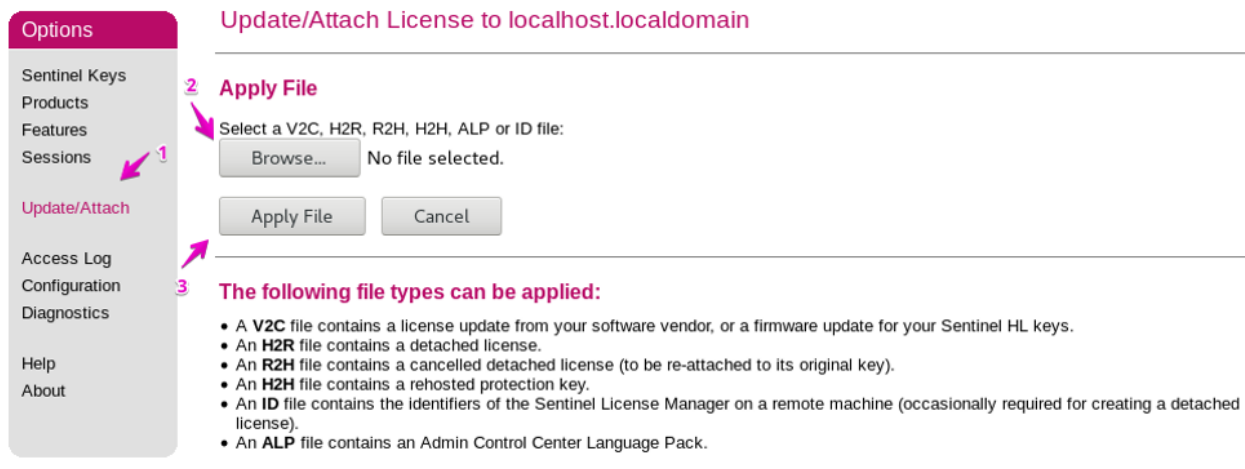
Отпечаток системы сохраняется в файл “fingerprint_111186.c2v”.

Отправьте файл в VisionLabs. Ваш лицензионный ключ будет создан с использованием данного отпечатка.

Добавление файла с лицензией вручную с помощью пользовательского интерфейса

- Перейдите: <host_address>:1947 (если доступ запрещен, проверьте настройки Firewall/ SELinux (данная процедура не описана в этом документе));
- Выберите **Update/Attach** в левой панели;

- Нажмите “Browse” и выберите файл(ы) лицензии в появившемся окне;
- Нажмите “Apply file”.



Лицензионный файл добавляется вручную

Задание адреса лицензированного сервера для LP

Укажите IP адрес сервера с лицензией в конфигурационном файле в следующей директории:

```
/var/lib/luna/current/example-docker/hasp_redirect/
```

Поменяйте адрес сервера HASP в следующих документах:

```
vi /var/lib/luna/current/example-docker/hasp_redirect/hasp_111186.ini
```

Поменяйте адрес сервера в файле “hasp_111186.ini”.

```
serveraddr = <HASP_server_address>
```

Файл “hasp_111186.ini” используется сервисом Licenses при запуске его контейнера. Требуется перезапустить уже запущенный контейнер при изменении сервера.

HASP_server_address - IP адрес сервера с вашим ключом HASP. Необходимо использовать IP адрес, а не имя сервера.

Установка Docker

Установка Docker описана в официальной документации:

<https://docs.docker.com/engine/install/>

Устанавливать Docker не требуется, если у вас на сервере уже установлен Docker последней версии.

Команды для быстрой установки приведены ниже.

Проверьте официальную документацию на наличие обновлений при возникновении каких-либо проблем с установкой.

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 статус Docker.

```
systemctl status docker
```

Установка Docker Compose

Установите Docker Compose.

```
curl -L "https://github.com/docker/compose/releases/download/1.24.1/docker-  
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose  
  
chmod +x /usr/local/bin/docker-compose  
  
ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
```

Для более подробной информации обратитесь к официальной документации:

<https://docs.docker.com/compose/install/>

Авторизация в registry

При запуске контейнеров 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>
```

Введите логин <username>. После ввода команды система запросит пароль. Логин и пароль можно получить у VisionLabs.

Запуск платформы

Конфигурация контейнера User Interface 4

Необходима конфигурация поля `BP4_EXTERNAL_IP:` в файле `"/var/lib/luna/current/example-docker/.env"` перед запуском скрипта Compose, если требуется использовать сервис User Interface 4 с сервисом Backport 4.

```
BP4_EXTERNAL_IP=<Backport_4_external_IP>
```

Необходимо задать внешний IP сервера через Backport 4. В противном случае User Interface 4 не сможет отправлять запросы в сервис Backport 4.

Запуск сервисов

Откройте папку Docker Compose:

```
cd /var/lib/luna/current/example-docker
```

Убедитесь в том, что контейнеры LP не запущены до выполнения скрипта. Попытка запустить контейнер с таким же именем, как существующий контейнер, приведет к ошибке. Если запущен один или несколько контейнеров LP, необходимо остановить их с помощью команды `docker container rm -f <container_name>`. Чтобы остановить все контейнеры, используйте `docker container rm -f $(docker container ls -aq)`.

Запуск Docker Compose:

Необходимо выполнить вход в VisionLabs registry (см. раздел ["Вход в registry"](#))

```
./start_platform.sh
```

Развертывание контейнеров требует времени. Необходимо дождаться того, чтобы все сервисы были запущены перед началом работы с LUNA PLATFORM 5.

Проверьте статус всех запущенных контейнеров Docker.

```
docker ps
```

Команды Docker

Показать контейнеры

Чтобы показать список запущенных контейнеров Docker, используйте команду:

```
docker ps
```

Чтобы показать все имеющиеся контейнеры Docker, используйте команду:

```
docker ps -a
```

Копировать файлы в контейнер

Можно переносить файлы в контейнер. Используйте команду `docker cp` для копирования файла в контейнер.

```
docker cp <file_location> <container_name>:<folder_inside_container>
```

Ввод контейнера

Можно вводить отдельные контейнеры с помощью следующей команды:

```
docker exec -it <container_name> bash
```

Для выхода из контейнера используйте следующую команду:

```
exit
```

Имена образов

Можно увидеть все имена образов с помощью команды

```
docker images
```

Удаление образа

Если требуется удаление образа:

- запустите команду `docker images`
- найдите требуемый образ, например `registry.dev.vlabs/luna-image-store`
- скопируйте соответствующий ID образа из IMAGE ID, например, "61860d036d8c"
- укажите его в команде удаления:

```
docker rmi -f 61860d036d8c
```

Удалите все существующие образы:

```
docker rmi -f $(docker images -q)
```

Остановка контейнера

Контейнер можно остановить с помощью следующей команды:

```
docker stop <container_name>
```

Остановить все контейнеры:

```
docker stop $(docker ps -a -q)
```

Удаление контейнера

Если вам необходимо удалить контейнер:

- запустите команду "docker ps"
- остановите контейнер (см. [Остановка контейнера](#))
- найдите требуемый образ, например: `registry.dev.vlabs/luna-image-store`
- скопируйте соответствующий ID контейнера из столбца CONTAINER ID, например, "23f555be8f3a"
- укажите его в команде удаления:

```
docker container rm -f 23f555be8f3a
```

Удалить все контейнеры:

```
docker container rm -f $(docker container ls -aq)
```

Логи сервисов

Все сервисы пишут логи. Они сохраняются в контейнере каждого сервиса.

Они хранятся в директории "srv/" и имеют следующие имена:

`<service_name>_ERROR.txt`

`<service_name>_INFO.txt`

Отображение логов сервисов

Если Docker запущен как демон, логи отправляются в отдельный файл. Чтобы показать логи сервиса, используйте команду:

```
docker logs <container_name>
```