



**VisionLabs**  
MACHINES CAN SEE

# VisionLabs LP5 Smoke&Fire Detection

Инструкция по установке

## Оглавление

Глоссарий .....	3
Введение .....	5
Аппаратные и программные требования .....	7
Минимальные аппаратные требования .....	7
Программные требования.....	7
Общая информация.....	9
Перед установкой.....	10
Распаковка дистрибутива.....	10
SELinux и Firewall .....	10
Установка Docker.....	11
Установка Docker Compose.....	12
Авторизация в registry .....	12
Запуск LUNA PLATFORM.....	13
Запуск сервисов.....	13
Создание аккаунта с помощью сервиса API.....	13
Команды Docker.....	15
Показать контейнеры.....	15
Копировать файлы в контейнер.....	15
Ввод контейнера.....	15
Имена образов.....	15
Удаление образа.....	15
Остановка контейнера.....	16
Удаление контейнера.....	16
Логи сервисов.....	16
Отображение логов сервисов.....	17

## Глоссарий

Термин	Определение
LUNA PLATFORM 5	<p>Система распознавания лиц.</p> <p>ПО VisionLabs LUNA PLATFORM 5 включено в Единый реестр российских программ для электронных вычислительных машин и баз данных</p> <p>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</p>

## VisionLabs LP5 Smoke&Fire Detection. Инструкция по установке

Сокращённое название	Расшифровка
LP	LUNA PLATFORM 5
VL-LP5-SFD	VisionLabs LP5 Smoke&Fire Detection
LPE	LUNA PLATFORM 5 Enterprise
API	LUNA PLATFORM API
DB	База данных
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
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

## Введение

VisionLabs LP5 Smoke&Fire Detection (далее – Модуль) – это Модуль, предназначенный для обнаружения огня и дыма на кадрах видеопотока.

Для установки и использования VisionLabs LP5 Smoke&Fire Detection 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.

## VisionLabs LP5 Smoke&Fire Detection. Инструкция по установке

- 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.*

## Аппаратные и программные требования

### Минимальные аппаратные требования

Для работы Модуля необходим запущенный экземпляр LUNA PLATFORM 5.

Следующие минимальные системные требования необходимы для установки программного пакета LUNA PLATFORM 5:

- 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.*

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

- CPU с частотой 2 ГГц и выше;
- 4 Гб оперативной памяти и выше;
- 400 Мб свободного места на жестком диске.

### Программные требования

Для запуска LUNA PLATFORM может использоваться RedOS (РЕД ОС) версии 7.3 и выше.

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
LUNA PLATFORM 5 Remote SDK	5220
Oracle DB	1521
PostgreSQL	5432
Redis DB	6379
InfluxDB	8086

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



## 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/>

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.

## Перед установкой

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

### Распаковка дистрибутива

Дистрибутив представляет собой архив **luna\_v.5.51.0**, где **5.51.0** 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.5.51.0.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.5.51.0.zip
```

Создание символической ссылки

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

```
ln -s luna_v.5.51.0 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.

## Установка Docker

Установка Docker описана в [официальной документации](#)

Если на сервере уже установлен Docker версии 20.10.8, то выполнять повторную установку не требуется. Не гарантируется работа с более высокими версиями Docker.

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/1.29.2/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
```

### 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.

Логин и пароль можно запросить у представителя VisionLabs.

Введите логин <username>.

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

После выполнения команды будет запрошен ввод пароля. Введите пароль.

В команде `docker login` можно вводить логин и пароль одновременно, однако это не гарантирует безопасность, т.к. пароль можно будет увидеть в истории команд.

## Запуск LUNA PLATFORM

Запуск Docker Compose осуществляется с помощью скрипта “start\_platform.sh”, расположенного в директории “example-docker”.

При необходимости можно модифицировать скрипт запуска Docker Compose под пользовательские нужды. Модификация скрипта предназначена только для опытных пользователей.

Во время запуска скрипта будет создан стандартный аккаунт типа **user** с логином `user@mail.com` и паролем `password`. Инструкция по созданию собственного аккаунта приведена ниже.

См. подробную информацию об аккаунтах в разделе “Аккаунты, токены и способы авторизации” руководства администратора.

### Запуск сервисов

Откройте директорию 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.

Проверьте статус всех запущенных Docker-контейнеров.

```
docker ps
```

### Создание аккаунта с помощью сервиса API

**Примечание.** При запуске скрипта Docker Compose автоматически создается аккаунт типа “user” с логином “user@mail.com” и паролем “password”. Инструкция по созданию аккаунта со своими аутентификационными данными приведена ниже.

Аккаунт создается с помощью HTTP-запроса к ресурсу [“create account”](#).

## VisionLabs LP5 Smoke&Fire Detection. Инструкция по установке

Аккаунт также можно создать с помощью сервиса Admin. Данный способ требует наличия существующих логина и пароль (или логина и пароля по умолчанию) и позволяет создать аккаунты типа “admin”. См. подробную информацию в разделе “Сервис Admin” руководства администратора.

Для создания аккаунта с помощью запроса к сервису API необходимо указать следующие обязательные данные:

- login — электронный адрес
- password — пароль
- account\_type — тип аккаунта (“user” или “advanced\_user”)

Создайте аккаунт, используя свои аутентификационные данные.

Пример CURL-запроса к ресурсу [“create account”](#):

```
curl --location --request POST 'http://127.0.0.1:5000/6/accounts' \
--header 'Content-Type: application/json' \
--data '{
  "login": "user@mail.com",
  "password": "password",
  "account_type": "user",
  "description": "description"
}'
```

Необходимо заменить аутентификационные данные из примера на свои.

См. подробную информацию об аккаунтах в разделе “Аккаунты и типы авторизации” руководства администратора.

Для работы с токенами необходимо наличие аккаунта.

## 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
```

## VisionLabs LP5 Smoke&Fire Detection. Инструкция по установке

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>
```