

VisionLabs LP5 DeepFake detection

Инструкция по установке

Содержание

Глоссарий	3
Стандартные порты для сервисов	4
Названия сервисов в Configurator	5
Системные требования	6
Процессоры	6
1. Введение	7
2. Подготовка к запуску	9
2.1 Распаковка дистрибутива	9
2.2 Создание символической ссылки	10
2.3 SELinux и Firewall	10
2.5 Установка Docker	10
2.6 Установка Docker Compose	11
2.7 Выбор способа записи логов	11
2.8 Вычисления с помощью GPU	12
2.9 Авторизация в registry	14
3. Запуск LUNA PLATFORM	15
3.1 Запуск сервисов	15
3.2 Создание аккаунта с помощью сервиса API	16

Глоссарий

Термин	Определение
VisionLabs 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
Сокращённое название	Расшифровка
LP	VisionLabs LUNA PLATFORM 5
API	VisionLabs LUNA PLATFORM 5 API
VL-LP5-DFD	VisionLabs LP5 DeepFake detection

Стандартные порты для сервисов

Стандартные порты для сервисов

Название сервиса	Порт
LUNA PLATFORM API	5000
LUNA PLATFORM Admin	5010
LUNA PLATFORM Image Store	5020
LUNA PLATFORM Faces	5030
LUNA PLATFORM Events	5040
LUNA PLATFORM Tasks	5050
LUNA PLATFORM Tasks Worker	5051
LUNA PLATFORM Configurator	5070
LUNA PLATFORM Sender	5080
LUNA PLATFORM Handlers	5090
LUNA PLATFORM Python Matcher	5100
LUNA PLATFORM Licenses	5120
LUNA PLATFORM Backport 4	5130
LUNA PLATFORM Backport 3	5140
LUNA PLATFORM Accounts	5170
LUNA PLATFORM Lambda	5210
LUNA PLATFORM Remote SDK	5220
LUNA PLATFORM 3 User Interface	4100
LUNA PLATFORM 4 User Interface	4200
Oracle DB	1521
PostgreSQL	5432
Redis DB	6379
InfluxDB	8086
Grafana	3000

Названия сервисов в Configurator

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

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
Accounts	luna-accounts
Tasks	luna-tasks
Events	luna-events
Sender	luna-sender
Admin	luna-admin
Remote SDK	luna-remote-sdk
Handlers	luna-handlers
Lambda	luna-lambda
Python Matcher	luna-python-matcher
Backport 3	luna-backport3
Backport 4	luna-backport4

Настройки для сервиса Configurator устанавливаются в его конфигурационном файле.

Системные требования

LUNA PLATFORM поставляется в Docker-контейнерах и может запускаться на CPU и GPU. Для установки необходимы образы Docker-контейнеров LP. Для загрузки образов Docker на сервере требуется подключение к сети Интернет, либо образы следует загрузить на любое другое устройство и перенести на сервер. Требуется вручную указать логин и пароль для загрузки образов Docker.

LUNA PLATFORM можно запустить с помощью скрипта Docker Compose.

Рекомендуется использовать следующие версии Docker и Docker Compose для запуска LP:

- Docker: 20.10.8 (для ручного запуска контейнеров)
- Docker Compose: 1.29.2 (для автоматического запуска контейнеров)

Процессоры

Приведенная ниже 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.

CPU

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

- 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 хранилища Image Store.

GPU

Для ускорения GPU необходим NVIDIA GPU. Поддерживаются следующие архитектуры:

- Pascal или более новые.

Требуется Compute Capability 6.1 или выше.

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

CUDA версии 11.4 должна быть установлена на сервере сервиса Remote SDK. Рекомендуемый драйвер NVIDIA - r470.

1. Введение

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

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

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

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

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

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

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

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

Данный документ также содержит инструкцию по автоматическому запуску LUNA Dashboards (Grafana) и Loki (см. раздел “[Включение Grafana и Loki](#)”).

Для успешного запуска нужно выполнить действия из разделов “[Подготовка к запуску](#)” и “[Запуск платформы](#)”.

Примечания о скрипте Docker Compose. Скрипт:

- тестируется с использованием настроек сервисов по умолчанию;
- 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;
- поддерживает использование GPU для вычислений LP;
- не обеспечивает возможность использования внешних баз данных, уже установленных на сервере;

VisionLabs LP5 DeepFake detection: Инструкция по установке

- не выполняет миграции из предыдущих версий LP и обновления предыдущих сборок LP;
- не запускает сервисы Backport 3, Backport 4, User Interface 3, User Interface 4.

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

<https://docs.docker.com>

Все описываемые команды необходимо исполнять в оболочке Bash (когда команды запускаются напрямую на сервере) или в программе для работы с сетевыми протоколами (в случае удаленного подключения к серверу), например, Putty.

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.

2 Подготовка к запуску

Убедитесь в том, что вы являетесь **root**-пользователем перед тем, как начать запуск!

Перед запуском LUNA Index Module необходимо выполнить следующие действия:

1. [Распаковать дистрибутив LUNA PLATFORM](#)
2. [Создать символическую ссылку](#)
3. [Настроить SELinux и Firewall](#)
4. [Выполнить установку Docker](#)
5. [Выполнить установку Docker Compose](#)
6. [Выбрать способ записи логов](#)
7. [Авторизоваться в registry VisionLabs](#)
8. [Настроить вычисления с помощью GPU](#), если планируется использовать GPU

2.1 Распаковка дистрибутива

Дистрибутив представляет собой архив **luna_v.5.51.6**, где **v.5.51.6** это числовой идентификатор, обозначающий версию LUNA PLATFORM.

Архив включает в себя конфигурационные файлы, требуемые для установки и использования. Он не включает в себя Docker образы сервисов, их требуется скачать из Интернета отдельно.

Переместите дистрибутив в директорию на вашем сервере перед установкой. Например, переместите файлы в директорию `/root/`.

Переместите дистрибутив в директорию с LUNA PLATFORM.

```
| mv /root/luna_v.5.51.6.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.5.51.6.zip |
```

2.2 Создание символической ссылки

Создайте символическую ссылку. Она показывает, что актуальная версия файла дистрибутива используется для запуска LUNA PLATFORM.

```
ln -s luna_v.5.51.6 current
```

2.3 SELinux и Firewall

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

Конфигурация SELinux и Firewall не описываются в данном руководстве.

Если SELinux и Firewall не настроены, дальнейшая установка невозможна.

2.5 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
```

2.6 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
```

2.7 Выбор способа записи логов

В LUNA PLATFORM существует два способа вывода логов:

- стандартный вывод логов (stdout);
- вывод логов в файл.

Настройки вывода логов задаются в настройках каждого сервиса в секции `<SERVICE_NAME>_LOGGER`.

При необходимости можно использовать оба способа вывода логов.

Для более подробной информации о системе логирования LUNA PLATFORM см. раздел “Логирование информации” в руководстве администратора.

2.7.1 Запись логов в stdout

Данный способ используется по умолчанию и для него не требуется выполнять дополнительных действий.

Рекомендуется настроить ротацию логов Docker для ограничения их размеров (см. раздел “Настройка ротации логов Docker”).

2.7.2 Запись логов в файл

Примечание. При включении сохранения логов в файле необходимо помнить о том, что логи занимают определенное место в хранилище, а процесс логирования в файл негативно влияет на производительность системы.

Для использования данного способа необходимо выполнить следующие дополнительные действия:

- **перед запуском сервисов:** создать директории для логов на сервере;
- **после запуска сервисов:** активировать запись логов и задать расположение хранения логов внутри контейнеров сервисов LP;
- **во время запуска сервисов:** настроить синхронизацию директорий логов в контейнере с логами на сервере с помощью аргумента `volume` при старте каждого контейнера.

В скрипте Docker Compose уже настроена синхронизация директорий логов, необходимо только создать директории и активировать запись логов.

См. инструкцию по включению записи логов в файлы в разделе [“Запись логов на сервер”](#).

2.8 Вычисления с помощью GPU

Для основных вычислений, выполняемых сервисом Remote SDK, можно использовать GPU.

Пропустите данный раздел, если не собираетесь использовать GPU для вычислений.

Для использования GPU необходимо установить Docker Compose v1.28.0+.

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

После обновления списка пакетов установите пакет `nvidia-docker2` и зависимости:

```
yum clean expire-cache
yum install -y nvidia-docker2
```

VisionLabs LP5 DeepFake detection: Инструкция по установке

```
systemctl restart docker
```

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

```
docker run --rm --gpus all nvidia/cuda:11.4.3-base-centos7 nvidia-smi
```

Далее необходимо дополнительно добавить секцию `deploy` в поле `remote-sdk` в файл `docker-compose.yml`.

```
vi /var/lib/luna/current/example-docker/docker-compose.yml

remote-sdk:
  image: ${DOCKER_URL}/luna-remote-sdk:${LUNA_REMOTE_SDK_VER}
  deploy:
    resources:
      reservations:
        devices:
          - driver: nvidia
            count: all
            capabilities: [gpu]
  restart: always
  ...
```

`driver` - в данном поле указывается драйвер для зарезервированного устройства(устройств);

`count` - в данном поле задается количество графических процессоров, которые должны быть зарезервированы (при условии, что хост содержит такое количество графических процессоров);

`capabilities` - данное поле выражает как общие, так и специфические возможности драйвера. Его необходимо задать, иначе будет возвращена ошибка при развертывании сервиса.

Для дополнительной информации см. следующую документацию:

<https://docs.docker.com/compose/gpu-support/#enabling-gpu-access-to-service-containers>.

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 ускорение GPU в сценариях с небольшой нагрузкой, когда задержка начала обработки имеет значение.

2.9 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.

3 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.

3.1 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
```

3.1.1 Запуск Remote SDK с использованием GPU

VisionLabs LP5 DeepFake detection: Инструкция по установке

Сервис Remote SDK не использует GPU по умолчанию. Если вы собираетесь использовать GPU, то следует включить его использование для сервиса Remote SDK в сервисе Configurator.

Если необходимо использовать GPU сразу для всех эстиматоров и детекторов, то необходимо использовать параметр “global_device_class” в секции “LUNA_REMOTE_SDK_RUNTIME_SETTINGS”. Все эстиматоры и детекторы будут использовать значение данного параметра, если в параметре “device_class” их собственных настроек выставлено значение “global” (по умолчанию).

Если необходимо использовать GPU для определенного эстиматора или детектора, то необходимо использовать параметр “device_class” в секциях вида “LUNA_REMOTE_SDK_<estimator-or-detector-name>_SETTINGS.runtime_settings”.

См. раздел “[Вычисления с помощью GPU](#)” для получения дополнительных требований к использованию GPU.

3.2 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”.

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",
```

VisionLabs LP5 DeepFake detection: Инструкция по установке

```
"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.