

# **VisionLabs LUNA Index Module**

**Руководство по установке**

**v.5.96.0**

## Содержание

<b>Порты сервисов по умолчанию</b>	<b>4</b>
<b>Названия сервисов в Configurator</b>	<b>5</b>
<b>Введение</b>	<b>6</b>
<b>1 Подготовка к запуску</b>	<b>7</b>
1.1 Распаковка дистрибутива . . . . .	8
1.2 Создание символической ссылки . . . . .	8
1.3 Создание директории для индексов . . . . .	8
1.4 SELinux и Firewall . . . . .	9
1.5 Авторизация в registry . . . . .	9
1.6 Выбор способа записи логов . . . . .	9
1.6.1 Запись логов в stdout . . . . .	10
1.6.2 Запись логов в файл . . . . .	10
1.7 Загрузка настроек LIM в Configurator . . . . .	10
1.8 Обновление настроек в Configurator . . . . .	11
<b>2 Запуск сервисов</b>	<b>12</b>
2.1 Запуск контейнера Python Matcher Proxy . . . . .	13
2.2 Запуск контейнера Indexer . . . . .	14
2.3 Запуск контейнера Index Manager . . . . .	15
2.4 Запуск контейнера Indexed Matcher . . . . .	16
<b>3 Дополнительная информация</b>	<b>17</b>
3.1 Шаги для выполнения сравнения БШ . . . . .	18
3.2 Команды Docker . . . . .	19
3.2.1 Показать контейнеры . . . . .	19
3.2.2 Копировать файлы в контейнер . . . . .	19
3.2.3 Вход в контейнер . . . . .	19
3.2.4 Имена образов . . . . .	19
3.2.5 Удаление образа . . . . .	19
3.2.6 Остановка контейнера . . . . .	20
3.2.7 Удаление контейнера . . . . .	20
3.2.7.1 Проверка логов сервисов . . . . .	21
3.3 Описание параметров запуска . . . . .	22
3.3.1 Параметры запуска сервисов . . . . .	22
3.3.1.1 Аргументы сервисов . . . . .	23
3.3.2 Параметры создания баз данных . . . . .	25

3.4	Запись логов на сервер . . . . .	27
3.4.1	Создание директории логов . . . . .	27
3.4.2	Активация записи логов . . . . .	27
3.4.3	Монтирование директорий с логами при старте сервисов . . . . .	28

## Порты сервисов по умолчанию

Имя сервиса	Порт
LUNA PLATFORM Python Matcher Proxy	5110
LUNA Index Module Indexer	5180
LUNA Index Module Index Manager	5190
LUNA Index Module Indexed Matcher	5200

## Названия сервисов в Configurator

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

Сервис	Название сервиса в Configurator
Python Matcher Proxy	luna-matcher-proxy
Index Manager	lim-manager
Indexer	lim-indexer
Indexed Matcher	lim-matcher

## Введение

Данный документ описывает общий подход для развертывания LUNA Index Module (LIM) в Docker-контейнерах.

**LIM — это самостоятельный модуль, поэтому его следует интегрировать в уже развернутую LUNA PLATFORM 5 аналогичной версии. Необходимо обновить LP если версии отличаются.**

Каждый сервис LIM имеет собственный образ. Образы Docker являются основой для контейнеров. Каждый контейнер включает в себя библиотеки, необходимые для запуска сервисов, и рабочие параметры для использования в период запуска контейнера.

Считается, что установка выполняется на сервере с Almalinux 8 OS, где LIM не был установлен.

Администратор должен вручную сконфигурировать Firewall и SELinux на сервере. Их конфигурация не описывается в данном документе.

В данной установке не предполагается резервное копирование каких-либо данных LIM.

Для успешного запуска нужно выполнить действия из разделов «Подготовка к запуску» и «Запуск сервисов». В разделе «Дополнительная информация» приводится полезная информация по описанию командах Docker, дальнейшим шагам для выполнения сравнения биометрических шаблонов и др.

Данный документ не включает руководство по использованию Docker. Пожалуйста, обратитесь к документации Docker для более подробной информации:

<https://docs.docker.com>

Данный документ включает в себя пример развертывания LIM. LIM разворачивается в минимальной рабочей конфигурации для использования в демонстрационных целях. Данная конфигурация не является достаточной для реальной эксплуатации системы в продуктивном контуре.

Для коммерческого использования LIM рекомендуется оркестрация сервисов. Их использование не описано в данном документе.

Все описываемые команды необходимо исполнять в оболочке Bash (когда команды запускаются напрямую на сервере) или Putty (в случае удаленного подключения к серверу). Описываемые команды тестировались только с помощью этих средств. Использование других оболочек или эмуляторов может привести к ошибкам при выполнении команд.

## 1 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:

- [Распаковать дистрибутив LIM](#)
- [Создать символическую ссылку](#)
- [Создать директорию для хранения индексов](#)
- [Настроить SELinux и Firewall](#)
- [Авторизоваться в registry VisonLabs](#)
- [Выбрать способ записи логов](#)
- [Загрузить настройки LIM в Configurator](#)
- [Обновить настройки LIM в Configurator](#)

## 1.1 Распаковка дистрибутива

Дистрибутив представляет собой архив **lim\_v.5.96.0**, где **v.5.96.0** это числовой идентификатор, обозначающий версию LIM.

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/`. В ней не должно быть никакого другого дистрибутива или файлов лицензии кроме целевых.

Переместите дистрибутив в директорию `/var/lib/luna/`.

```
mv /root/lim_v.5.96.0.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 lim_v.5.96.0.zip
```

## 1.2 Создание символической ссылки

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

```
ln -s lim_v.5.96.0 lim-current
```

## 1.3 Создание директории для индексов

Для хранения индексов и взаимодействия с ними на сервере нужно создать соответствующую директорию. Эта директория будет смонтирована при запуске сервисов LIM.

Создайте папку для хранения индексов:

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

Присвойте соответствующие права для возможности чтения/записи данной папки:

```
chown -R 1001:0 /var/lib/luna/lim_storage
```

## 1.4 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.**

## 1.5 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.

## 1.6 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.

### 1.6.1 Запись логов в stdout

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

### 1.6.2 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.

Примеры команд запуска контейнеров в данной документации содержат аргументы для синхронизации директорий логов.

Обратите внимание, что вышеперечисленные действия должны выполняться перед, во время и после запуска сервисов. Запись логов в файл не будет работать если выполнять все действия после запуска контейнеров.

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

## 1.7 Загрузка настроек LIM в Configurator

Для использования сервисов LIM с LP 5 необходимо загрузить их настройки в сервис Configurator, используя механизм миграции конфигураций.

```
docker run \  
-v /etc/localtime:/etc/localtime:ro \  
--rm \  
--entrypoint=' ' \  
--network=host \  

```

```
dockerhub.visionlabs.ru/luna/lim-configs:v.0.7.0 python3 -m configs.migrate
head --config_db_url postgres://luna:luna@127.0.0.1:5432/
luna_configurator
```

--config\_db\_url postgres://luna:luna@127.0.0.1:5432/luna\_configurator — флажок указания адреса БД luna\_configurator

## 1.8 Обновление настроек в Configurator

Далее необходимо включить использование сервиса Python Matcher Proxy и добавить плагин сравнения в список плагинов, используемых сервисом Python Matcher Proxy.

Скопируйте файл с необходимыми настройками в контейнер Configurator:

```
docker cp /var/lib/luna/lim-current/example-docker/configs/lim_settings.json
luna-configurator:/srv/lim_settings.json
```

Обновите настройки в сервисе Configurator:

```
docker exec -it luna-configurator python3 ./base_scripts/db_create.py --dump
-file /srv/lim_settings.json
```

В результате в сервисе Configurator будут обновлены следующие настройки:

```
LUNA_MATCHER_PROXY_ACTIVE_PLUGINS = ["indexed_matcher"]
ADDITIONAL_SERVICES_USAGE = "luna_matcher_proxy":true
```

## 2 Запуск сервисов

В данном разделе приведены примеры команд для запуска контейнеров LIM.

Последовательность запуска сервисов LIM такова:

- [Python Matcher Proxy](#)
- [Indexer](#)
- [Index Manager](#)
- [Indexed Matcher](#)

Для запуска сервисов LIM должна быть развернута LP 5.

Рекомендуется запускать контейнеры один за другим и ожидать отображения статуса контейнера «up» (команда `docker ps`).

При запуске каждого сервиса используются определенные параметры, например, `--detach`, `--network` и др. См. раздел «[Описание параметров запуска](#)» для получения более подробной информации о всех параметрах запуска сервисов LUNA PLATFORM и баз данных.

См. раздел «[Команды Docker](#)» для получения более подробной информации о работе с контейнерами.

## 2.1 Запуск контейнера Python Matcher Proxy

Используйте следующую команду для запуска сервиса:

```
docker run \  
--env=CONFIGURATOR_HOST=127.0.0.1 \  
--env=CONFIGURATOR_PORT=5070 \  
--env=PORT=5110 \  
--env=WORKER_COUNT=1 \  
--env=RELOAD_CONFIG=1 \  
--env=RELOAD_CONFIG_INTERVAL=10 \  
--env=SERVICE_TYPE="proxy" \  
-v /etc/localtime:/etc/localtime:ro \  
-v /tmp/logs/python-matcher-proxy:/srv/logs \  
--name=luna-python-matcher-proxy \  
--restart=always \  
--detach=true \  
--network=host \  
dockerhub.visionlabs.ru/luna/luna-python-matcher:v.1.14.1
```

## 2.2 Запуск контейнера Indexer

Используйте следующую команду для запуска сервиса:

```
docker run \
--env=CONFIGURATOR_HOST=127.0.0.1 \
--env=CONFIGURATOR_PORT=5070 \
--env=PORT=5180 \
--env=WORKER_COUNT=1 \
--env=RELOAD_CONFIG=1 \
--env=RELOAD_CONFIG_INTERVAL=10 \
-v /etc/localtime:/etc/localtime:ro \
-v /tmp/logs/lim-indexer:/srv/logs \
-v /var/lib/luna/lim_storage:/srv/local_storage \
--name=lim-indexer \
--restart=always \
--detach=true \
--network=host \
dockerhub.visionlabs.ru/luna/lim-indexer:v.0.7.0
```

Развертывание сервиса Indexer должно выполняться на отдельном сервере, поскольку создание индекса занимает много ресурсов в течение длительного времени. Один экземпляр сервиса Indexer может одновременно создавать только один индекс, поэтому рекомендуется запускать несколько экземпляров сервиса. Сервис также должен быть настроен с хранилищем, которое должно быть достаточно большим.

## 2.3 Запуск контейнера Index Manager

Используйте следующую команду для запуска сервиса:

```
docker run \
--env=CONFIGURATOR_HOST=127.0.0.1 \
--env=CONFIGURATOR_PORT=5070 \
--env=PORT=5190 \
--env=WORKER_COUNT=1 \
--env=RELOAD_CONFIG=1 \
--env=RELOAD_CONFIG_INTERVAL=10 \
-v /etc/localtime:/etc/localtime:ro \
-v /tmp/logs/lim-manager:/srv/logs \
-v /var/lib/luna/lim_storage:/srv/local_storage \
--name=lim-manager \
--restart=always \
--detach=true \
--network=host \
dockerhub.visionlabs.ru/luna/lim-manager:v.0.7.0
```

Рекомендуется запускать как минимум два экземпляра сервиса Index Manager в целях резервирования. Так как управление задачами осуществляется через Redis, то если один сервис Index Manager не работает, второй сможет продолжить свою работу с шага, на котором остановился предыдущий.

## 2.4 Запуск контейнера Indexed Matcher

Используйте следующую команду для запуска сервиса:

```
docker run \
--env=CONFIGURATOR_HOST=127.0.0.1 \
--env=CONFIGURATOR_PORT=5070 \
--env=PORT=5200 \
--env=WORKER_COUNT=1 \
--env=RELOAD_CONFIG=1 \
--env=RELOAD_CONFIG_INTERVAL=10 \
-v /etc/localtime:/etc/localtime:ro \
-v /tmp/logs/lim-matcher:/srv/logs \
-v /var/lib/luna/lim_storage:/srv/local_storage \
--name=lim-matcher \
--restart=always \
--detach=true \
--network=host \
dockerhub.visionlabs.ru/luna/lim-matcher:v.0.7.0
```

Сервис Indexed Matcher не взаимодействует с другими сервисами LIM. Он только следит за хранилищем и при появлении индексов загружает их в память. Поскольку обработка запросов на сравнение выполняется через потоки Redis, любое количество экземпляров сервиса Indexed Matcher может быть запущено без каких-либо обновлений конфигурации системы. Количество экземпляров сервиса Indexed Matcher должно определяться требованиями к производительности.

Для каждого экземпляра сервиса Indexed Matcher можно задать переменную окружения `VL_LIM_MATCHER_HOST`, указывающую его IP-адрес. Это может быть полезно для разделения данных мониторинга по разным экземплярам. Например, можно определить сколько экземпляров находится в работе, из-за какого экземпляра произошла ошибка, какие индексы в работе и т.д.

### 3 Дополнительная информация

В данном разделе приводится следующая дополнительная информация:

- [Дальнейшие шаги для выполнения сравнения БШ](#)
- [Полезные команды для работы с Docker](#)
- [Описание параметров запуска сервисов LIM и создания баз данных](#)
- [Действия по включению сохранения логов сервисов LIM в файлы](#)

### 3.1 Шаги для выполнения сравнения БШ

Для начала процедуры сравнения БШ с помощью сервисов LIM необходимо выполнить следующие действия:

- выбрать предпочтительный режим, который будет использоваться для сравнения БШ — **разовый** или **автоматический**;
- **разовый режим**: указать в теле запроса к ресурсу «create task» сервиса Index Manager требуемый «list\_id» и выполнить запрос;
- **автоматический режим для работы с конкретными списками**: указать требуемые «list\_id» в настройке «indexing\_lists» секции «LIM\_MANAGER\_INDEXING» сервиса Configurator;
- **автоматический режим для работы со всеми существующими списками**:
  - указать в настройке «indexing\_lists» секции «LIM\_MANAGER\_INDEXING» сервиса Configurator значение «dynamic»;
  - указать минимальное количество лиц в списке в настройке «min\_indexing\_list\_size» секции «LIM\_MANAGER\_INDEXING» сервиса Configurator.
- отправить запрос на сравнение БШ из сервиса API.

Подробную информацию о взаимодействии сервисов LIM см. в разделе «Взаимодействие сервисов» руководства администратора LIM.

## 3.2 Команды Docker

### 3.2.1 Показать контейнеры

Чтобы показать список запущенных Docker-контейнеров, используйте команду:

```
docker ps
```

Чтобы показать все имеющиеся Docker-контейнеры, используйте команду:

```
docker ps -a
```

### 3.2.2 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>
```

### 3.2.3 Вход в контейнер

Можно входить в отдельные контейнеры с помощью следующей команды:

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

Для выхода из контейнера используйте следующую команду:

```
exit
```

### 3.2.4 Имена образов

Можно увидеть все имена образов с помощью команды

```
docker images
```

### 3.2.5 Удаление образа

Если требуется удаление образа:

- запустите команду `docker images`
- найдите требуемый образ, например `dockerhub.visionlabs.ru/luna/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)
```

### 3.2.6 Остановка контейнера

Контейнер можно остановить с помощью следующей команды:

```
docker stop <container_name>
```

Остановить все контейнеры:

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

### 3.2.7 Удаление контейнера

Если необходимо удалить контейнер:

- запустите команду «`docker ps`»
- остановите контейнер (см. [Остановка контейнера](#))
- найдите требуемый образ, например: `dockerhub.visionlabs.ru/luna/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)
```

### 3.2.7.1 Проверка логов сервисов

Чтобы показать логи сервиса, используйте команду:

```
docker logs <container_name>
```

### 3.3 Описание параметров запуска

При запуске Docker-контейнера для какого-либо из сервисов LUNA Index Module необходимо задать дополнительные параметры, требуемые для работы этого сервиса.

Параметры, требуемые для конкретного контейнера, описаны в разделе, посвященном запуску этого контейнера.

Все параметры, приведенные в примере запуска сервиса, необходимы для корректного запуска и работы сервиса.

#### 3.3.1 Параметры запуска сервисов

Пример команды запуска контейнеров сервисов LP:

```
docker run \
--env=CONFIGURATOR_HOST=127.0.0.1 \
--env=CONFIGURATOR_PORT=5070 \
--env=PORT=<Port_of_the_launched_service> \
--env=WORKER_COUNT=1 \
--env=RELOAD_CONFIG=1 \
--env=RELOAD_CONFIG_INTERVAL=10 \
-v /etc/localtime:/etc/localtime:ro \
-v /tmp/logs/<service>:/srv/logs/ \
--name=<service_container_name> \
--restart=always \
--detach=true \
--network=host \
dockerhub.visionlabs.ru/luna/<service-name>:<version>
```

Следующие параметры используются при запуске контейнеров сервисов LP:

- `docker run` — команда для запуска выбранного образа в качестве нового контейнера.
- `dockerhub.visionlabs.ru/luna/<service-name>:<version>` — позволяет задать образ, требуемый для запуска контейнера.

Ссылки для загрузки требуемых образов контейнера доступны в описании запуска соответствующего контейнера.

- `--network=host` — 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. Пример приведен для PostgreSQL.

- `--env=` — задает переменные окружения, требуемые для запуска контейнера (см. раздел «Аргументы сервисов»).
- `--name=<service_container_name>` — задает имя запускаемого контейнера. Имя должно быть уникальным. Если уже существует контейнер с таким же именем, произойдет ошибка.
- `--restart=always` — определяет политику перезагрузки. Демон всегда перезагружает контейнер вне зависимости от кода завершения.
- `--detach=true` — позволяет запустить контейнер в фоновом режиме.
- `-v` — позволяет загружать содержимое серверной папки в объем контейнера. Таким образом содержимое синхронизируется. Загружаются следующие общие данные:
- `/etc/localtime:/etc/localtime:ro` — задает текущий часовой пояс, используемый системой контейнера.
- `/tmp/logs/<service>:/srv/logs/` — позволяет копировать папку с записями (логами) сервиса на сервер в директорию `/tmp/logs/<service>`. Директорию для хранения логов можно изменить при желании.

### 3.3.1.1 Аргументы сервисов

Каждый сервис в LUNA PLATFORM имеет свои собственные аргументы запуска. Эти аргументы можно передать через:

- задание флага для скрипта запуска (`run.py`) соответствующего сервиса
- установку отдельных переменных окружения (`--env`) в командной строке Docker

Например, с использованием флага `--help` можно получить список всех доступных аргументов. Пример передачи аргумента для сервиса API может выглядеть следующим образом:

```
docker run --rm dockerhub.visionlabs.ru/luna/luna-api:v.6.47.0 python3 /srv/luna_api/run.py --help
```

Список основных аргументов:

Флаг в строке запуска	Переменная окружения	Описание
<code>--port</code>	PORT	Порт, на котором сервис будет ожидать подключений.
<code>--workers</code>	WORKER_COUNT	Количество «рабочих процессов» для сервиса.
<code>--log_suffix</code>	LOG_SUFFIX	Суффикс, добавляемый к именам файлов логов (при включенном параметре записи логов в файл).

<code>--config-reload</code>	RELOAD_CONFIG	Включение автоматической перезагрузки конфигураций. См. раздел «Автоматическая перезагрузка конфигураций» в руководстве администратора LUNA PLATFORM 5.
<code>--pulling-time</code>	RELOAD_CONFIG_INTERVAL	Период проверки конфигураций (по умолчанию 10 секунд). См. раздел «Автоматическая перезагрузка конфигураций» в руководстве администратора LUNA PLATFORM 5.
<code>--luna-config</code>	CONFIGURATOR_HOST, CONFIGURATOR_PORT	Адрес сервиса Configurator для загрузки настроек. Для <code>--luna-config</code> передается в формате <code>http://localhost:5070/1</code> . Для переменных окружения хост и порт задаются явно. Если аргумент не задан, то будет использован конфигурационный файл по умолчанию.
<code>--config</code>	Нет	Путь до конфигурационного файла с настройками сервиса.
<code>--&lt;config_name&gt;</code>	Нет	Тег указанной настройки в Configurator. При задании данной настройки будет использовано значение тегированной настройки. Пример: <code>--LUNA_MONITORING TAG_1</code> . <b>Примечание.</b> Необходимо заранее присвоить тег соответствующим настройкам в Configurator. <b>Примечание.</b> Работает только с флагом <code>--luna-config</code> .
<code>--tls-cert</code>	Нет	Путь к SSL-сертификату для запуска сервиса с использованием протокола HTTPS.

<code>--tls_key</code>	Нет	Путь к SSL-закрытому ключу для запуска сервиса с использованием протокола HTTPS.
<code>--tls_key_pass</code>	Нет	Пароль для SSL-закрытого ключа для запуска сервиса с использованием протокола HTTPS.

Перечень аргументов может отличаться в зависимости от сервиса.

Также доступна возможность переопределить настройки сервисов при их старте с помощью переменных окружения.

Для переопределения настроек используется префикс `VL_SETTINGS`. Примеры:

- `--env=VL_SETTINGS.LUNA_MONITORING.SEND_DATA_FOR_MONITORING=0`. Использование переменной окружения из данного примера установит значение настройки «`SEND_DATA_FOR_MONITORING`» для секции «`LUNA_MONITORING`» равным «0».
- `--env=VL_SETTINGS.OTHER.STORAGE_TIME=LOCAL`. Для несоставных настроек (настроек, которые расположены в секции «`OTHER`» в конфигурационном файле) необходимо указать префикс «`OTHER`». Использование переменной окружения из данного примера установит значение настройки «`STORAGE_TIME`» (если сервис использует данную настройку) на значение «`LOCAL`».

### Передача флагов с использованием переменной окружения

Флаги, для которых явно не выделена переменная окружения, можно передать с помощью переменной окружения `EXTEND_CMD`.

Например, можно передать тег настроек следующим способом:

```
--env=EXTEND_CMD="--LUNA_MONITORING=TAG_1 --LUNA_EVENTS_DB=TAG_2"
```

- `/var/lib/luna/lim_storage:/srv/local_storage` — позволяет смонтировать директорию для хранения индексов в локальное хранилище. Расположение и название директории для хранения индексов внутри контейнеров LIM задается в настройках «`INDEX_STORAGE_LOCAL`» сервисов LIM. Обратите внимание, что для всех трех сервисов директория должна быть одинаковой. Локальную директорию можно изменить при желании.

### 3.3.2 Параметры создания баз данных

Пример команды запуска контейнеров для миграции баз данных или их создания:

```
docker run \  
-v /etc/localtime:/etc/localtime:ro \  
-v /tmp/logs/<service>:/srv/logs/ \  
--rm \  
--network=host \  
dockerhub.visionlabs.ru/luna/<service-name>:<version> \  
python3 ./base_scripts/db_create.py --luna-config http://localhost:5070/1
```

Следующие параметры используются при запуске контейнеров для миграции баз данных или их создания:

`--rm` — этот параметр указывает, удаляется ли контейнер после завершения обработки всех заданных скриптов.

`python3 ./base_scripts/db_create.py` — этот параметр содержит версию Python и скрипт `db_create.py`, запускаемый в контейнере. Этот скрипт используется для создания структуры базы данных.

`--luna-config http://localhost:5070/1` — этот параметр указывает, откуда запущенный скрипт должен получать конфигурации. По умолчанию конфигурации запрашиваются сервисами от сервиса Configurator.

### 3.4 Запись логов на сервер

Чтобы включить сохранение логов на сервер, необходимо:

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

#### 3.4.1 Создание директории логов

Ниже приведены примеры команд для создания директорий для хранения логов и присвоения им прав для всех сервисов LUNA PLATFORM.

```
mkdir -p /tmp/logs/lim-manager /tmp/logs/lim-indexer /tmp/logs/lim-matcher /  
tmp/logs/python-matcher-proxy
```

```
chown -R 1001:0 /tmp/logs/lim-manager /tmp/logs/lim-indexer /tmp/logs/lim-  
matcher /tmp/logs/python-matcher-proxy
```

#### 3.4.2 Активация записи логов

Для активации записи логов в файл необходимо задать настройки `log_to_file` и `folder_with_logs` в секции `<SERVICE_NAME>_LOGGER` настроек каждого сервиса.

##### Автоматический способ

Для обновления настроек ведения логов можно использовать файл настроек `logging.json`, предоставленный в комплекте поставки.

Выполните следующую команду:

```
docker cp /var/lib/luna/lim-current/example-docker/configs/logging.json luna  
-configurator:/srv/luna_configurator/used_dumps/logging.json
```

Обновите настройки записи логов с помощью скопированного файла.

```
docker exec -it luna-configurator python3 ./base_scripts/db_create.py --dump  
-file /srv/luna_configurator/used_dumps/logging.json
```

## Ручной способ

Перейдите в интерфейс сервиса Configurator (127.0.0.1:5070) и задайте путь расположения логов в контейнере в параметре `folder_with_logs` для всех сервисов, чьи логи необходимо сохранить. Например, можно использовать путь `/srv/logs`.

Установите параметр `log_to_file` как `true` чтобы активировать запись логов в файл.

### 3.4.3 Монтирование директорий с логами при старте сервисов

Директория с логами монтируется с помощью следующего аргумента при старте контейнера:

```
-v <server_logs_folder>:<container_logs_folder> \
```

где `<server_logs_folder>` директория, созданная на этапе [создания директории логов](#), а `<container_logs_folder>` директория, созданная на этапе [активации записи логов](#).

Пример команды запуска сервиса Index Manager с монтированием директории с логами:

```
docker run \
--env=CONFIGURATOR_HOST=127.0.0.1 \
--env=CONFIGURATOR_PORT=5070 \
--env=PORT=5190 \
--env=WORKER_COUNT=1 \
--env=RELOAD_CONFIG=1 \
--env=RELOAD_CONFIG_INTERVAL=10 \
-v /etc/localtime:/etc/localtime:ro \
-v /tmp/logs/lim-manager:/srv/logs \
-v /var/lib/luna/lim_storage:/srv/local_storage \
--name=lim-manager \
--restart=always \
--detach=true \
--network=host \
dockerhub.visionlabs.ru/luna/lim-manager:v.0.7.0
```

Примеры команд запуска контейнеров в данной документации содержат эти аргументы.