

Модуль индексирования LUNA

Руководство по обновлению

v.5.57.0

Содержание

Порты сервисов LIM по умолчанию	4
Названия сервисов LIM в Configurator	5
Системные требования	6
Процессор	6
Введение	7
1 Подготовка к обновлению	8
1.1 Перенос хранилища индексов	9
1.2 Удаление символической ссылки	9
1.3 Распаковка дистрибутива	9
1.4 Создание символической ссылки	10
1.5 SELinux и Firewall	10
1.6 Авторизация в registry	10
1.7 Миграция настроек LIM	10
1.8 Обновление настроек в Configurator	11
1.9 Удаление старых контейнеров	11
1.10 Дополнить построенные индексы информацией о версии БШ	12
2 Запуск сервисов	13
2.1 Запуск контейнера Python Matcher Proxy	14
2.2 Запуск контейнера Indexer	15
2.3 Запуск контейнера Index Manager	16
2.4 Запуск контейнера Indexed Matcher	17
3 Дополнительная информация	18
3.1 Команды Docker	19
3.1.1 Показать контейнеры	19
3.1.2 Копировать файлы в контейнер	19
3.1.3 Вход в контейнер	19
3.1.4 Имена образов	19
3.1.5 Удаление образа	19
3.1.6 Остановка контейнера	20
3.1.7 Удаление контейнера	20
3.2 Описание параметров запуска	22
3.2.1 Параметры запуска сервисов	22
3.2.2 Параметры создания баз данных	25

3.3	Запись логов на сервер	27
3.3.1	Создание директории логов	27
3.3.2	Активация записи логов	27
3.3.3	Монтирование директорий с логами при старте сервисов	28

Порты сервисов LIM 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 LIM 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

Системные требования

Модуль индексирования для LUNA PLATFORM поставляется в Docker-контейнерах. Для запуска LIM необходимо наличие Docker и образов Docker-контейнеров LIM. 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.

Запуск контейнеров LIM официально поддерживается на CentOS 7/8. Корректная работа на других системах не гарантируется. Все процедуры в руководстве по установке описаны для CentOS 7.

В сервисах LIM используются операционная система CentOS Linux 8.3.2011.

LIM лицензируется с помощью отдельного параметра в ключе LUNA PLATFORM 5.

Процессор

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.

Следующие минимальные системные требования необходимы для установки программного пакета LIM:

- 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.

Введение

В данном документе приводится пример шагов, необходимых для обновления с предыдущей сборки LIM на новую сборку.

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

Данное руководство написано с предположением, что:

- предыдущая минорная версия LIM уже установлена, и требуемое окружение на сервере готово к работе.
- LP 5 и LIM установлены в соответствии с руководствами по установке, и используются пути по умолчанию. В противном случае следует внести изменения вручную в процессе обновления.

Для успешного обновления нужно выполнить действия из разделов [«Подготовка к обновлению»](#) и [«Запуск сервисов»](#). В разделе [«Дополнительная информация»](#) приводится полезная информация по описанию команд Docker и включению логирования в файл.

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.

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 обновлению

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

Перед обновлением LUNA Index Module необходимо выполнить следующие действия:

- Перенести хранилище индексов из старого дистрибутива, если оно там располагалось
- Удалить старую символическую ссылку
- Распаковать дистрибутив LIM
- Создать новую символическую ссылку
- Настроить SELinux и Firewall, если ранее не были настроены
- Авторизоваться в registry VisonLabs
- Выполнить миграцию старых настроек LIM
- Обновить настройки LIM в Configurator
- Удалить старые контейнеры
- Дополнить построенные индексы информацией о версии БШ

1.1 Перенос хранилища индексов

По умолчанию хранилище индексов располагается в директории «/var/lib/luna/lim_storage» без привязки к версии LIM. Необходимо переместить хранилище индексов в директорию «/var/lib/luna» если в предыдущих версиях директория располагалась в ином месте.

1.2 Удаление символической ссылки

Удалите символическую ссылку в директорию предыдущей версии с помощью следующей команды:

```
rm -f /var/lib/luna/lim-current
```

1.3 Распаковка дистрибутива

Дистрибутив представляет собой архив **lim_v.5.57.0**, где **v.5.57.0** 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.5.57.0.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.5.57.0.zip
```

1.4 Создание символической ссылки

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

```
ln -s lim_v.5.57.0 lim-current
```

1.5 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.

1.6 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.

1.7 Миграция настроек LIM

Выполните следующую команду для миграции настроек LIM в сервисе LUNA Configurator.

```
docker run \
-v /etc/localtime:/etc/localtime:ro \
--rm \
--entrypoint=' ' \
--network=host \
dockerhub.visionlabs.ru/luna/lim-configs:v.0.4.1 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` - флаг 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
```

1.9 Удаление старых контейнеров

Перед запуском контейнеров текущей минорной версии необходимо остановить все контейнеры, относящиеся к предыдущей минорной версии LIM.

Для удаления контейнера используйте следующую команду:

```
docker container rm -f [container_name]
```

где `[container_name]` это имя контейнера сервиса docker или ID.

Например, для удаления контейнеров LP используйте следующую команду:

```
docker container rm -f lim-indexer lim-manager lim-matcher luna-python-
matcher-proxy
```

Чтобы посмотреть имена контейнеров или их ID, используйте следующую команду:

```
docker ps -a
```

Также рекомендуется удалить старые образы контейнеров для освобождения места. Можно использовать следующую команду для удаления всех неиспользуемых образов.

Если на сервере достаточно места, рекомендуется выполнить это действие только после успешного запуска новой версии LIM.

Данная команда удаляет все неиспользуемые образы, а не только образы, относящиеся к LIM.

```
docker image prune -a -f
```

1.10 Дополнить построенные индексы информацией о версии БШ

Примечание. Выполняйте данные действия только если обновляйтесь с версии **5.32.0**.

В версии LIM 5.34.0 добавлен учет версий биометрических шаблонов индексов, задаваемых в настройке «DEFAULT_FACE_DESCRIPTOR_VERSION» сервиса Index Manager.

После запуска сервиса Index Manager, он автоматически начнет перестраивание всех индексов, в которых отсутствует информация о биометрических шаблонах, т.е. всех, созданных в версии LIM 5.32.0. Перестройка индекса может занять длительное время, зависящее от количества лиц в списках.

Для того, чтобы не выполнять длительный процесс перестроения индекса, нужно добавить поле «descriptor_version» с соответствующей версией биометрического шаблона в файлы «meta.json» всех созданных ранее индексов перед запуском сервиса Index Manager.

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.1.8.6
```

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.0.4.1
```

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.0.4.1
```

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.0.4.1
```

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:

- [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 Команды Docker

3.1.1 Показать контейнеры

Чтобы показать список запущенных Docker-контейнеров, используйте команду:

```
docker ps
```

Чтобы показать все имеющиеся Docker-контейнеры, используйте команду:

```
docker ps -a
```

3.1.2 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>
```

3.1.3 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
```

3.1.4 Имена образов

Можно увидеть все имена образов с помощью команды

```
docker images
```

3.1.5 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)
```

3.1.6 Остановка контейнера

Контейнер можно остановить с помощью следующей команды:

```
docker stop <container_name>
```

Остановить все контейнеры:

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

3.1.7 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)
```

3.1.7.1 Проверка логов сервисов

Чтобы показать логи сервиса, используйте команду:

```
docker logs <container_name>
```

3.2 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.

3.2.1 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.

3.2.1.1 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

Некоторые аргументы можно передать только через задание флага. Для сервисов Handlers и Remote SDK доступна возможность использовать переменную окружения «EXTEND_CMD» для явной передачи флагов. См. пример использования переменной «EXTEND_CMD» в разделе «Запуск облегченной версии Remote SDK» руководства по установке LUNA PLATFORM 5.

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.6.24.0 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>--<config_name></code>	Нет	<p>Тег указанной настройки в Configurator. При задании данной настройки будет использовано значение тегированной настройки.</p> <p>Пример:</p> <pre>--INFLUX_MONITORING TAG_1.</pre> <p>Примечание. Необходимо заранее присвоить тег соответствующим настройкам в Configurator.</p> <p>Примечание. Работает только с флагом <code>--luna-config</code>.</p>
------------------------------------	-----	---

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.INFLUX_MONITORING.SEND_DATA_FOR_MONITORING=0`. Использование переменной окружения из данного примера установит значение настройки «`SEND_DATA_FOR_MONITORING`» для секции «`INFLUX_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`».
- `/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.

3.2.2 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.

3.3 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.

3.3.1 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
```

3.3.2 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.

3.3.3 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.0.4.1
```

Примеры команд запуска контейнеров в данной документации содержат эти аргументы.