

# Модуль индексирования LUNA

Пример развертывания с использованием Docker Compose

v.5.67.0

## Содержание

<b>Порты сервисов по умолчанию</b>	<b>3</b>
<b>Названия сервисов в Configurator</b>	<b>4</b>
<b>Введение</b>	<b>5</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 Выбор способа записи логов . . . . .	9
1.5.1 Запись логов в stdout . . . . .	9
1.5.2 Запись логов в файл . . . . .	9
1.6 Установка Docker Compose . . . . .	10
1.7 Авторизация в registry . . . . .	10
<b>2 Запуск сервисов</b>	<b>12</b>
<b>3 Дополнительная информация</b>	<b>13</b>
3.1 Шаги для выполнения сравнения БШ . . . . .	14
3.2 Команды Docker . . . . .	15
3.2.1 Показать контейнеры . . . . .	15
3.2.2 Копировать файлы в контейнер . . . . .	15
3.2.3 Вход в контейнер . . . . .	15
3.2.4 Имена образов . . . . .	15
3.2.5 Удаление образа . . . . .	15
3.2.6 Остановка контейнера . . . . .	16
3.2.7 Удаление контейнера . . . . .	16
3.3 Запись логов на сервер . . . . .	18
3.3.1 Создание директории логов . . . . .	18
3.3.2 Активация записи логов . . . . .	18

## 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 установку и использование Docker Compose для развертывания LUNA Index Manager (LIM).

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

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

Считается, что установка выполняется на сервере с операционной системой CentOS, где LIM не был установлен.

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.

LIM разворачивается с использованием минимальных ресурсов, необходимые для демонстрации работы LIM и не может использоваться для реальных рабочих целей.

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

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

Примечания о скрипте Docker Compose. Скрипт:

- тестируется с использованием настроек сервисов по умолчанию.
- не предназначен для использования в целях масштабирования LIM:
  - Не используется для развертывания сервисов LIM на нескольких серверах.
  - Не используется для развертывания и балансирования нескольких сервисов LIM на одном сервере.
- не обеспечивает возможность использования внешних баз данных, уже установленных на сервере.

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 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](#)
- [Выбрать способ записи логов](#)
- [Выполнить установку Docker Compose](#)
- [Авторизоваться в registry VisonLabs](#)

## 1.1 Распаковка дистрибутива

Дистрибутив представляет собой архив **lim\_v.5.67.0**, где **v.5.67.0** 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.5.67.0.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.5.67.0.zip
```

## 1.2 Создание символической ссылки

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

```
ln -s lim_v.5.67.0 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 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.

### 1.5.1 Запись логов в stdout

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

### 1.5.2 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](#)».

## 1.6 Установка Docker Compose

**Примечание.** При тестировании данной инструкции использовался Docker Compose версии 2.24.6. Не гарантируется работа с более высокими версиями Docker Compose.

Загрузите бинарный файл Docker Compose:

```
curl -L "https://github.com/docker/compose/releases/download/v2.24.6/docker-  
compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

Установите разрешения на выполнение:

```
chmod +x /usr/local/bin/docker-compose
```

Создайте символическую ссылку:

```
ln -s /usr/local/bin/docker-compose /usr/bin/docker-compose
```

## 1.7 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.

## 2 Запуск сервисов

Откройте директорию Docker Compose:

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

Убедитесь в том, что запущены все необходимые контейнеры LP (см. системные требования LIM).

Запуск Docker Compose:

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

```
./start_lim.sh
```

Скрипт загружает настройки LIM в контейнер Configurator со стандартным названием `luna-configurator`. Если имя контейнера Configurator по каким-либо причинам имеет другое название, то его нужно сменить в переменной `LUNA_CONFIGURATOR_NAME` в файле `.env`.

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

Проверьте статус всех запущенных Docker-контейнеров.

```
docker ps
```

### 3 Дополнительная информация

В данном разделе приводится следующая дополнительная информация:

- [Дальнейшие шаги для выполнения сравнения БШ](#)
- [Полезные команды для работы с Docker](#)
- [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 Запись логов на сервер

Чтобы включить сохранение логов на сервере, 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.

В скрипте Docker Compose уже настроена синхронизация директорий с папками, создаваемыми в разделе ниже.

#### 3.3.1 Создание директории логов

Необходимо создать следующие директории для хранения логов и присвоить им соответствующие права.

```
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 после запуска сервиса Configurator:

```
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.