



**VisionLabs**  
MACHINES CAN SEE

# **LUNA Index Module**

**Example of deployment using Docker Compose**

**v.5.67.0**

## Contents

<b>Default ports for services</b>	<b>3</b>
<b>Configuration names for services</b>	<b>4</b>
<b>Introduction</b>	<b>5</b>
<b>1 Before launch</b>	<b>6</b>
1.1 Distribution unpacking . . . . .	7
1.2 Symbolic link creation . . . . .	7
1.3 Create directory to store indexes . . . . .	7
1.4 SELinux and Firewall . . . . .	8
1.5 Choose logging method . . . . .	8
1.5.1 Logging to stdout . . . . .	8
1.5.2 Logging to file . . . . .	8
1.6 Docker Compose installation . . . . .	9
1.7 Login to registry . . . . .	9
<b>2 Services launch</b>	<b>10</b>
<b>3 Additional information</b>	<b>11</b>
3.1 Steps to perform descriptors matching . . . . .	12
3.2 Docker commands . . . . .	13
3.2.1 Show containers . . . . .	13
3.2.2 Copy files to container . . . . .	13
3.2.3 Enter container . . . . .	13
3.2.4 Images names . . . . .	13
3.2.5 Delete image . . . . .	13
3.2.6 Stop container . . . . .	14
3.2.7 Delete container . . . . .	14
3.3 Logging to server . . . . .	16
3.3.1 Create logs directory . . . . .	16
3.3.2 Logging activation . . . . .	16

## Default ports for services

Service name	Port
LUNA PLATFORM Python Matcher Proxy	5110
LUNA Index Module Indexer	5180
LUNA Index Module Index Manager	5190
LUNA Index Module Indexed Matcher	5200

## Configuration names for services

The table below includes the LIM service names in the Configurator service. Use these parameters to configure your services.

Service	Service name in Configurator
Python Matcher Proxy	luna-matcher-proxy
Index Manager	lim-manager
Indexer	lim-indexer
Indexed Matcher	lim-matcher

## Introduction

This document describes the installation and usage of Docker Compose for LUNA Index Module (LIM) deployment.

**LIM is an substantive module, so it should be integrated into LUNA PLATFORM 5 of a similar version already deployed. It is necessary to update the LP if the versions differ.**

Docker Compose is used for automated containers deployment. The Docker Compose scenario from this distribution is used for deploying the LIM services on a single server.

It is considered that installation is performed on the server with CentOS OS, where LIM was not installed.

Firewall and SELinux should be manually configured on the server by the administrator. Their configuration is not described in this document.

No data backup is implemented for LIM data in this installation.

LIM is deployed using the minimum power operating for demonstration purposes and cannot be used for the production system.

For a successful launch, you need to perform the actions from the sections “[Before launch](#)” and “[Services launch](#)”. The section “[Additional information](#)” provides useful information on the description of Docker commands, further steps for performing descriptor matching, etc.

See the “docker-compose.yml” file and other files in the “example-docker” directory for the information about launched services and performed actions.

Additional notes about Docker Compose script. The script:

- Is tested using the default services configurations.
- Is not intended to be used for LIM scaling:
  - It is not used for the deployment of LIM services on several servers.
  - It is not used for deployment and balancing of several LIM services on a single service.
- Does not provide the possibility to use external databases already installed on the server.

You can write your scenario that deploys and configures all the required services. This document does not include information about scenario creation or tutorial for Docker usage. Please refer to the Docker documentation to find more information about Docker and Docker Compose:

<https://docs.docker.com>

All the provided commands should be executed using the Bash shell (when you launch commands directly on the server) or Putty (when you remotely connect to the server). The provided commands were tested with these utilities only. The use of other shells or emulators can lead to errors when executing commands.

## 1 Before launch

Make sure that you are the **root** user before launch!

Before launching the LUNA Index Module, you must perform the following actions:

- [Unpack the LIM distribution.](#)
- [Create symbolic link.](#)
- [Create directory to store indexes.](#)
- [Configure SELinux and Firewall.](#)
- [Choose logging method.](#)
- [Install Docker Compose.](#)
- [Login to VisionLabs registry.](#)

## 1.1 Distribution unpacking

The distribution package is an archive **lim\_v.5.67.0**, where **v.5.67.0** is a numerical identifier, describing the current LUNA Index Module version.

The archive includes configuration files, required for installation and exploitation. It does not include Docker images for the services. They should be downloaded from the Internet.

Move the distribution package to the directory on your server before the installation. For example, move the files to `/root/` directory. The directory should not contain any other distribution or license files except the target ones.

Move the distribution to the `/var/lib/luna/` directory.

```
mv /root/lim_v.5.67.0.zip /var/lib/luna
```

Install the unzip archiver if it is necessary.

```
yum install -y unzip
```

Go to the folder with distribution.

```
cd /var/lib/luna
```

Unzip files.

```
unzip lim_v.5.67.0.zip
```

## 1.2 Symbolic link creation

Create a symbolic link. The link indicates that the current version of the distribution file is used to run LIM.

```
ln -s lim_v.5.67.0 lim-current
```

## 1.3 Create directory to store indexes

To store indexes and interact with them on the server, you need to create an appropriate directory. This directory will be mounted when the LIM services are launched.

Create folder to store indexes.

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

Set appropriate permissions to read/write this folder.

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

## 1.4 SELinux and Firewall

You must configure SELinux and Firewall so that they do not block LUNA PLATFORM services.

SELinux and Firewall configurations are not described in this guide.

**If SELinux and Firewall are not configured, the installation cannot be performed.**

## 1.5 Choose logging method

There are two methods to output logs in LUNA PLATFORM:

- Standard log output (stdout).
- Log output to a file.

Log output settings are set in the settings of each service in the <SERVICE\_NAME>\_LOGGER section.

If necessary, you can use both methods of displaying logs.

For more information about the LUNA PLATFORM logging system, see the “Logging” section in the administrator manual.

### 1.5.1 Logging to stdout

This method is used by default and requires no further action.

### 1.5.2 Logging to file

**Note:** When you enable saving logs to a file, you should remember that logs occupy a certain place in the storage, and the process of logging to a file negatively affects system performance.

To use this method, you need to perform the following additional actions:

- **Before launching the services:** Create directories for logs on the server.
- **After launching the services:** Activate log recording and set the location of log storage inside LP service containers.
- **During the launch of services:** Configure synchronization of log directories in the container with logs on the server using the `volume` argument at the start of each container.



Synchronization of log directories is already configured in the Docker Compose script, you only need to create directories and activate logging.

See the instructions for enabling logging to files in the [“Logging to server”](#) section.

## 1.6 Docker Compose installation

**Note:** Docker Compose version 2.24.6 was used when testing this instruction. It is not guaranteed to work with higher versions of Docker Compose.

Download Docker Compose binary:

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

Set execution permissions:

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

Create symbolic link:

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

## 1.7 Login to registry

When launching containers, you should specify a link to the image required for the container launching. This image will be downloaded from the VisionLabs registry. Before that, you should login to the registry.

Login and password can be requested from the VisionLabs representative.

Enter login <username>.

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

After running the command, you will be prompted for a password. Enter password.

In the `docker login` command, you can enter the login and password at the same time, but this does not guarantee security because the password can be seen in the command history.

## 2 Services launch

Go to the Docker Compose folder.

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

Make sure that all required LP containers are launched (see LIM system requirements).

Launch Docker Compose:

You must be logged in the VisionLabs registry (see section [“Login to registry”](#))

```
./start_lim.sh
```

The script loads the LIM settings into the Configurator container with the standard name `luna-configurator`. If the name of the Configurator container for some reason has a different name, then it must be changed in the `LUNA_CONFIGURATOR_NAME` variable in the `.env` file.

Deploying containers takes some time. You have to wait until all the services are running before working with LUNA Index Module.

Check the state of launched Docker containers.

```
docker ps
```

### 3 Additional information

This section provides the following additional information:

- [Steps to perform descriptors matching.](#)
- [Useful commands for working with Docker.](#)
- [Actions to enable saving LIM service logs to files.](#)

### 3.1 Steps to perform descriptors matching

To start the descriptors matching procedure using LIM services, you should perform the following steps:

- select the preferred mode that will be used for the descriptors matching — **one-time** or **automatic**.
- **One-time mode:** Specify the required “list\_id” in the request body to the “create task” resource of the Index Manager service and perform the request.
- **Automatic mode for working with specific lists:** Specify the required “list\_id” in the “indexing\_lists” setting of the “LIM\_MANAGER\_INDEXING” of the Configurator service.
- **Automatic mode for working with all existing lists:**
  - Specify the “dynamic” value in the “indexing\_lists” setting of the “LIM\_MANAGER\_INDEXING” of the Configurator service.
  - Specify the minimum number of faces in the list in the “min\_indexing\_list\_size” setting of the “LIM\_MANAGER\_INDEXING” of the Configurator service to index all existing descriptors in the lists.
- Send a request for matching descriptors from the API service.

For details on LIM service interaction, see the “Service interaction” section of the LIM administrator manual.

## 3.2 Docker commands

### 3.2.1 Show containers

To show the list of launched Docker containers use the command:

```
docker ps
```

To show all the existing Docker containers use the command:

```
docker ps -a
```

### 3.2.2 Copy files to container

You can transfer files into the container. Use the `docker cp` command to copy a file into the container.

```
docker cp <file_location> <container_name>:<folder_inside_container>
```

### 3.2.3 Enter container

You can enter individual containers using the following command:

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

To exit the container, use the command:

```
exit
```

### 3.2.4 Images names

You can see all the names of the images using the command:

```
docker images
```

### 3.2.5 Delete image

If you need to delete an image:

- Run the `docker images` command.

- Find the required image, for example [dockerhub.visionlabs.ru/luna/luna-image-store](https://hub.docker.com/r/visionlabs/luna-image-store).
- Copy the corresponding image ID from the IMAGE ID, for example, “61860d036d8c”.
- Specify it in the deletion command:

```
docker rmi -f 61860d036d8c
```

Delete all the existing images.

```
docker rmi -f $(docker images -q)
```

### 3.2.6 Stop container

You can stop the container using the command:

```
docker stop <container_name>
```

Stop all the containers:

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

### 3.2.7 Delete container

If you need to delete a container:

- Run the “docker ps” command.
- Stop the container (see [Stop container](#)).
- Find the required image, for example [dockerhub.visionlabs.ru/luna/luna-image-store](https://hub.docker.com/r/visionlabs/luna-image-store).
- Copy the corresponding container ID from the CONTAINER ID column, for example, “23f555be8f3a”.
- Specify it in the deletion command:

```
docker container rm -f 23f555be8f3a
```

Delete all the containers.

```
docker container rm -f $(docker container ls -aq)
```

#### 3.2.7.1 Check service logs

You can use the following command to show logs for the service:

```
docker logs <container_name>
```

### 3.3 Logging to server

To enable saving logs to the server, you should:

- Create directories for logs on the server.
- Activate log recording and set the location of log storage inside LP service containers.
- Configure synchronization of log directories in the container with logs on the server using the `volume` argument at the start of each container.

Docker Compose script is already configured to synchronize directories with the folders created in the section below.

#### 3.3.1 Create logs directory

You need to create the following directories for storing logs and assign them the appropriate rights.

```
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 Logging activation

To enable logging to file, you need to set the `log_to_file` and `folder_with_logs` settings in the `<SERVICE_NAME>_LOGGER` section of the settings for each service.

##### Automatic method

To update logging settings, you can use the `logging.json` settings file provided with the distribution package.

Run the following command after starting the Configurator service:

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

Update your logging settings with the copied file.

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



### **Manual method**

Go to the Configurator service interface (127.0.0.1:5070) and set the logs path in the container in the `folder_with_logs` parameter for all services whose logs need to be saved. For example, you can use the path `/srv/logs`.

Set the `log_to_file` option to `true` to enable logging to file.