



VisionLabs
MACHINES CAN SEE

LUNA Index Module

Upgrade manual

v.5.84.0

Contents

Default ports for services	3
Configuration names for services	4
Introduction	5
1 Before upgrade	6
1.1 Move index storage	7
1.2 Delete symbolic link	7
1.3 Distribution unpacking	7
1.4 Symbolic link creation	8
1.5 SELinux and Firewall	8
1.6 Migrate LIM settings	8
1.7 Update settings in the Configurator	8
1.8 Remove old containers	9
1.9 Augment built indexes with descriptor version information	9
2 Services launch	11
2.1 Python Matcher proxy container launch	12
2.2 Indexer container launch	13
2.3 Index Manager container launch	14
2.4 Indexed Matcher container launch	15
3 Additional information	16
3.1 Docker commands	17
3.1.1 Show containers	17
3.1.2 Copy files to container	17
3.1.3 Enter container	17
3.1.4 Images names	17
3.1.5 Delete image	17
3.1.6 Stop container	18
3.1.7 Delete container	18
3.2 Launching parameters description	20
3.2.1 Launching services parameters	20
3.2.2 Creating DB parameters	23
3.3 Logging to server	25
3.3.1 Create logs directory	25
3.3.2 Logging activation	25
3.3.3 Mounting directories with logs when starting services	26

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 gives an example of the steps for upgrading from previous LIM build to a new build.

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.

This manual is designed with an assumption that:

- You already have a previous minor version of LIM installed and the required environment is up and running at your server(s).
- LP 5 and LIM is installed according to installation manuals, and the default paths are used. Otherwise, you should consider your manual changes during the update.

For a successful upgrade, you need to perform the actions from the sections [“Before upgrade”](#) and [“Services launch”](#). The section [“Additional information”](#) provides useful information on the description of Docker commands and activation of logging to a file.

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 may lead to errors when executing commands.

This document includes an example of LIM deployment. It implements LIM minimum power operating for demonstration purposes and cannot be used for the production system.

1 Before upgrade

Make sure you are the **root** user before upgrade!

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

- [Move the index store from the old distribution](#) if it was located there.
- [Delete old symbolic link](#).
- [Unpack the distribution of the new version of LIM](#).
- [Create new symbolic link](#).
- [Configure SELinux and Firewall](#) if not previously configured.
- [Migrate old LIM settings](#).
- [Update LIM settings in Configurator](#).
- [Remove old containers](#).
- [Augment built indexes with descriptor version information](#).

1.1 Move index storage

By default, the index storage is located in the “/var/lib/luna/lim_storage” directory, without reference to the LIM version. You need to move the index storage to the “/var/lib/luna” directory if the directory was located elsewhere in previous versions.

1.2 Delete symbolic link

Delete the symbolic link to the previous version directory using the following command:

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

1.3 Distribution unpacking

The distribution package is an archive **lim_v.5.84.0**, where **v.5.84.0** 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.5.84.0.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.5.84.0.zip
```

1.4 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.5.84.0 lim-current
```

1.5 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.

1.6 Migrate LIM settings

Run the following command to migrate the LIM settings in the LUNA Configurator service.

```
docker run \
-v /etc/localtime:/etc/localtime:ro \
--rm \
--entrypoint=' ' \
--network=host \
dockerhub.visionlabs.ru/luna/lim-configs:v.0.6.0 python3 -m configs.migrate
  head --config_db_url postgres://luna:luna@127.0.0.1:5432/
  luna_configurator
```

Where `--config_db_url postgres://luna:luna@127.0.0.1:5432/luna_configurator` is the flag for specifying the `luna_configurator` database address.

1.7 Update settings in the Configurator

Next, you need to enable the use of the Python Matcher Proxy service and add the matching plugin to the list of plugins used by the Python Matcher Proxy service.

Copy the file with the necessary settings to the Configurator container.

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


Update the settings in the Configurator service.

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

As a result, the following settings will be updated in the Configurator service:

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

1.8 Remove old containers

Before launching the containers of the current minor version, stop all LIM related containers of the previous minor version.

For example, to remove LIM containers only use the following command:

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

To see the containers names or IDs, use the following command:

```
docker ps -a
```

It is also recommended to delete old images of the containers to free space. You can use the following command to delete all unused images.

If there is enough space on the server it is recommended to perform this action only after new version of LIM is successfully launched.

The command deletes all the unused images, not only the images related to LIM.

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

1.9 Augment built indexes with descriptor version information

Note: Follow these steps only if you are upgrading from version **5.32.0**.

In the LUNA Index Module v.5.34.0, accounting for descriptor versions of index set in the setting “DEFAULT_FACE_DESCRIPTOR_VERSION” of the Index Manager service has been added.

After starting the Index Manager service, it will automatically start rebuilding all indexes that lack information about descriptors, i.e. all created in the version 5.32.0. Rebuilding the index can take a long time, depending on the number of faces on the lists.

In order to avoid the lengthy process of rebuilding the index, you need to add the “descriptor_version” field with the corresponding version of the descriptor to the “meta.json” files of all previously created indexes before starting the Index Manager service.

2 Services launch

This section gives examples for launching LIM containers.

The sequence of launching LIM services is as follows:

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

To launch LIM services, the LP 5 must be deployed.

It is recommended to launch containers one by one and wait for the container status to become “up” (use the `docker ps` command).

When launching each service, certain parameters are used, for example, `--detach`, `--network`, etc. See the section [“Launching parameters description”](#) for more detailed information about all launch parameters of LUNA PLATFORM services and databases.

See the [“Docker commands”](#) section for details about working with containers.

2.1 Python Matcher proxy container launch

Use the following command to launch the service:

```
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.1.11.1
```

2.2 Indexer container launch

Use the following command to launch the service:

```
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.0.6.0
```

The deployment of the Indexer service should be done on a separate server, because building an index takes a lot of resources for a long time. One Indexer instance can only build one index at a time, so it is recommended to run multiple indexer instances. The indexer must be also configured with storage, which must be large enough.

2.3 Index Manager container launch

Use the following command to launch the service:

```
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.0.6.0
```

It is recommended to run at least two manager instances for redundancy purposes. Since task management is carried out through the Redis, if one manager is down, the second one will be able to continue its work from the instant step.

2.4 Indexed Matcher container launch

Use the following command to launch the service:

```
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.0.6.0
```

Indexed Matcher does not communicate with other LIM services. It only monitors the storage, and when indices appear it loads them into memory. Since matching requests processing is carried out through the Redis streams, any number of matcher instances could be run without any system config updates. The number of Indexed Matcher instances should be determined by performance requirements.

For each instance of the Indexed Matcher service, you can set an environment variable `VL_LIM_MATCHER_HOST` indicating its IP address. This can be useful for separating monitoring data into different instances. For example, you can determine how many instances are in work, which instance caused the error, which indexes are in operation, etc.

3 Additional information

This section provides the following additional information:

- [Useful commands for working with Docker.](#)
- [Description of the parameters for launching LIM services and creating databases.](#)
- [Actions to enable saving LIM service logs to files.](#)

3.1 Docker commands

3.1.1 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
```

3.1.2 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>
```

3.1.3 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
```

3.1.4 Images names

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

```
docker images
```

3.1.5 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)
```

3.1.6 Stop container

You can stop the container using the command:

```
docker stop <container_name>
```

Stop all the containers:

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

3.1.7 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)
```

3.1.7.1 Check service logs

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

```
docker logs <container_name>
```

3.2 Launching parameters description

When launching a Docker container for a LUNA Index Module service you should specify additional parameters required for the service launching.

The parameters specific for a particular container are described in the section about this container launching.

All the parameters given in the service launching example are required for proper service launching and utilization.

3.2.1 Launching services parameters

Example command of launching LP services containers:

```
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>
```

The following parameters are used when launching LP services containers:

- `docker run` — Command for running the selected image as a new container.
- `dockerhub.visionlabs.ru/luna/<service-name>:<version>` — Sets the image required for the container launching.

Links to download the container images you need are available in the description of the corresponding container launching.

- `--network=host` — Sets that a network is not simulated and the server network is used. If you need to change the port for third-party containers, you should change this string to `-p 5440:5432`. Where the first port 5440 is the local port and 5432 is the port used inside the container. The example is given for PostgreSQL.

- `--env=` — Sets the environment variables required to run the container (see the “[Service arguments](#)” section).
- `--name=<service_container_name>` — Sets the name of the launched container. The name must be unique. If there is a container with the same name, an error will occur.
- `--restart=always` — Sets a restart policy. The daemon will always restart the container regardless of the exit status.
- `--detach=true` — Run the container in the background mode.
- `-v` — Enables you to mount the content of a server folder into a volume in the container. Thus their contents will synchronize. The following general data is mounted:
- `/etc/localtime:/etc/localtime:ro` — Sets the current time zone used by the system in the container.
- `/tmp/logs/<service>:/srv/logs/` — Enables copying of the folder with service logs to your server `/tmp/logs/<service>` directory. You can change the directory where the logs will be saved according to your needs.

3.2.1.1 Service arguments

Each service in LUNA PLATFORM has its own launch arguments. These arguments can be passed through:

- Setting a flag for the launch script (`run.py`) of the corresponding service.
- Setting environment variables (`--env`) on the Docker command line.

For example, using the `--help` flag you can get a list of all available arguments. An example of passing an argument to an API service:

```
docker run --rm dockerhub.visionlabs.ru/luna/luna-api:v.6.39.0 python3 /srv/luna_api/run.py --help
```

List of main arguments:

Launch flag	Environment variable	Description
<code>--port</code>	PORT	Port on which the service will listen for connections.
<code>--workers</code>	WORKER_COUNT	Number of workers for the service.
<code>--log_suffix</code> <code>--log_suffix</code>	LOG_SUFFIX LOG_SUFFIX	Suffix added to log file names (with the option to write logs to a file enabled).

<code>--config-reload</code>	RELOAD_CONFIG	Enable automatic configuration reload. See “Automatic configurations reload” in the LUNA PLATFORM 5 administrator manual.
<code>--pulling-time</code>	RELOAD_CONFIG_INTERVAL	Configuration checking period (default 10 seconds). See “Automatic configurations reload” in the LUNA PLATFORM 5 administrator manual.
<code>--luna-config</code> <code>--luna-config</code>	CONFIGURATOR_HOST, CONFIGURATOR_PORT	Address of the Configurator service for downloading settings. For <code>--luna-config</code> it is sent in the format <code>http://localhost:5070/1</code> . For environment variables, the host and port are set explicitly. If the argument is not given, the default configuration file will be used.
<code>--config</code>	None	Path to the file with service configurations.
<code>--<config_name></code>	<code>--EXTEND_CMD=<config_name></code>	Tag of the specified configuration in the Configurator. When setting this configuration, the value of the tagged configuration will be used. Example: <code>--LUNA_MONITORING TAG_1</code> Note: You must pre-tag the appropriate configuration in. Configurator. Note: Only works with the <code>--luna-config</code> flag.
<code>--tls_cert</code>	None	Path to the SSL certificate for launching the service using the HTTPS protocol.
<code>--tls_key</code>	None	Path to the SSL private key for launching the service using the HTTPS protocol.
<code>--tls_key_pass</code>	None	Password for the SSL private key for launching the service using the HTTPS protocol.

The list of arguments may vary depending on the service.

It is also possible to override the settings of services at their start using environment variables.

The VL_SETTINGS prefix is used to redefine the settings. Examples:

- `--env=VL_SETTINGS.LUNA_MONITORING.SEND_DATA_FOR_MONITORING=0`. Using the environment variable from this example will set the “SEND_DATA_FOR_MONITORING” setting for the LUNA_MONITORING section to “0”.
- `--env=VL_SETTINGS.OTHER.STORAGE_TIME=LOCAL`. For non-compound settings (settings that are located in the “OTHER” section in the configuration file), you must specify the “OTHER” prefix. Using the environment variable from this example will set the value of the “STORAGE_TIME” setting (if the service uses this setting) to “LOCAL”.

Passing flags using environment variable

Flags for which an environment variable is not explicitly allocated can be passed using the environment variable EXTEND_CMD.

For example, you can pass the configurations tag in the following way:

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

- `/var/lib/luna/lim_storage:/srv/local_storage` — enables you to mount a directory for storing indexes in local storage. The location and name of the directory for storing indexes inside LIM containers is set in the “INDEX_STORAGE_LOCAL” settings of the LIM services. Note that the directory must be the same for all three services. The local directory can be changed according to your needs.

3.2.2 Creating DB parameters

Example command of launching containers for database migration or database creation:

```
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
```

The following parameters are used when launching containers for database migration or database creation:

Here:

- `--rm` — Sets if the container is deleted after all the specified scripts finish processing.

- `python3 ./base_scripts/db_create.py` — Sets Python version and a script `db_create.py` launched in the container. The script is used for the database structure creation.
- `--luna-config http://localhost:5070/1` — Sets where the launched script should receive configurations. By default, the service requests configurations from the Configurator service.

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.

3.3.1 Create logs directory

Below are examples of commands for creating directories for saving logs and assigning rights to them for all LUNA PLATFORM services.

```
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:

```
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 a file.

3.3.3 Mounting directories with logs when starting services

The log directory is mounted with the following argument when starting the container:

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

where `<server_logs_folder>` is the directory created in the [create logs directory](#) step, and `<container_logs_folder>` is the directory created in the [activate logging](#) step.

Example of command to launch the Index Manager service with mounting a directory with logs:

```
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.0.6.0
```

The example container launch commands in this documentation contain these arguments.