



VisionLabs
MACHINES CAN SEE

VISIONLABS LUNA PLATFORM THERMO

Инструкция по установке

ООО «ВижнЛабс»

123458, г. Москва, ул. Твардовского д. 8, стр. 1

☎ +7 (499) 399 3361

✉ info@visionlabs.ru

🌐 www.visionlabs.ru

Содержание

Глоссарий.....	3
Введение	4
1. Системные требования	5
2. Установка и настройка в Docker.....	6
2.1. Установка сервиса	6
2.1.1. Подготовка и распаковка дистрибутива.....	6
2.2. Настройка сервиса.....	7
2.2.1. Запуск сервиса	8
2.2.2. Добавление аккаунта администратора	8
3. Установка и настройка на сервер	9
3.1. Установка сервиса	9
3.1.1. Подготовка и распаковка дистрибутива.....	9
3.1.2. Настройка сервиса авторизации.....	10
3.1.3. Настройка системы управления базами данных	11
3.1.4. Установка зависимостей для clementine-auth	12
3.1.5. Добавление аккаунта администратора	12
3.1.6. Сервисные файлы	13
3.1.7. Запуск сервиса	13
3.2. Настройка сервиса для работы с отделениями clementine-departments	13
3.2.1. Установка зависимостей для clementine-departments.....	14
3.2.2. Сервисные файлы	15
3.2.3. Запуск сервиса	15
3.3. Настройка FaceStream и установка UI.....	15
3.3.1. Настройка FaceStream	15
3.3.2. Установка и настройка UI.....	16
4. Удаление сервиса	20
5. Логирование	22

Глоссарий

Термин	Определение
User Interface (UI)	Графический интерфейс пользователя
База данных (БД)	Совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных
Биометрический шаблон	Набор данных в закрытом двоичном формате, подготавливаемый системой на основе анализируемой характеристики. Представляет из себя составной вектор признаков фотоизображения лица человека
Детекция	Область на изображении, содержащая обнаруженный объект (лицо)
Идентификация	Поиск наиболее подходящего биометрического шаблона лица путем сравнения векторов признаков фотоизображения лица с перечнем аналогичных биометрических шаблонов в базе (один ко многим)
Обработчик	Набор правил (политик) для обработки изображения
Программное обеспечение (ПО)	Программа или множество программ, используемых для управления компьютером
Распознавание лиц	Совокупность методов сбора, обработки и хранения данных графического изображения лица человека для распознавания личности или подтверждения заявленной личности с использованием математических методов
Система управления базами данных	Совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных
Событие	Зафиксированный системой факт детекции с извлечением атрибутов обработчиком
Список	Множество лиц, объединенных автоматически или вручную по определенному признаку

Введение

Настоящий документ описывает процесс установки и настройки «VisionLabs LUNA PLATFORM THERMO» (далее — Сервис), а также содержит аппаратные и программные требования к ПО. Исключительные права на ПО «VisionLabs LUNA PLATFORM THERMO» принадлежат ООО «ВижнЛабс» (далее — Компания).

Настоящий документ подлежит размещению на официальном сайте Компании в сети Интернет по адресу: <https://visionlabs.ai/ru/>.

VisionLabs LUNA PLATFORM THERMO — веб-сервис, предоставляющий возможности взаимодействия пользователей с коммерческими биометрическими системами в части работы с событиями и списками, и обеспечивающий обработку данных, полученных с тепловизионной камеры.

Перед установкой VisionLabs LUNA PLATFORM THERMO необходимо установить и настроить коммерческую биометрическую систему.

В документе рассмотрен процесс установки VisionLabs LUNA PLATFORM THERMO на примере работы с ПО для обработки видеопотоков VisionLabs FaceStream и с биометрической системой VisionLabs LUNA PLATFORM 5.

Установка и настройка указанного ПО выходит за рамки данного документа.

1. Системные требования

Для установки VisionLabs LUNA PLATFORM THERMO необходимо, чтобы выполнялись следующие требования к аппаратным ресурсам.

Необходимый ресурс	Рекомендовано
Процессор (CPU)	64-битный процессор Intel или AMD с поддержкой SSE2 и AVX2 инструкциями, с 4 ядрами и более, с тактовой частотой 2,0 ГГц и выше.
Оперативная память (RAM)	8 Гб и выше
Свободное место на диске (HDD/SSD)	20 Гб и выше
Операционная система (OS)	CentOS 7.8.2003 x86_64

2. Установка и настройка в Docker

2.1. Установка сервиса

Необходимо выполнить следующие действия:

1. Установить [Docker Engine](#) и [Docker Compose](#).

Установка Docker Engine и Docker Compose в данном руководстве не описывается.

2. Обновить Docker Compose до последней версии.

```
curl -L "https://github.com/docker/compose/releases/download/1.29.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

3. Изменить права.

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

4. Запустить Docker.

```
systemctl start docker  
systemctl enable docker
```

2.1.1. Подготовка и распаковка дистрибутива

Дистрибутив представляет собой архив вида «clementine-X.Y.ZZ», где «X.Y.ZZ» — численный идентификатор, обозначающий версию продукта.

Архив содержит все компоненты, необходимые для установки и эксплуатации системы.

Архив не включает зависимости, которые входят в стандартную поставку репозитория CentOS 7.4 x86_64 и могут быть загружены из открытых источников.

Перед процессом установки необходимо поместить файлы дистрибутива в директорию на сервере. Например, в директорию `/root/`. В данной директории не должно быть других файлов дистрибутива кроме целевых, используемых для установки конечного продукта.

Необходимо выполнить следующие действия:

1. Переключиться на пользователя `root`.

```
sudo su
```

2. Создать директорию для распаковки дистрибутива.

```
mkdir -p /var/lib/clementine
```

3. Переместить дистрибутив в созданную директорию.

```
mv /root/clementine* /var/lib/clementine
```

4. Установить архиватор `unzip`, если он не установлен.

```
yum install -y unzip
```

5. Перейти в папку с дистрибутивом.

```
cd /var/lib/clementine
```

6. Разархивировать файлы.

```
unzip clementine*.zip
```

7. Создать символическую ссылку, указав вместо «X.Y.ZZ» версию продукта. Ссылка указывает, что именно текущая версия дистрибутива используется для запуска.

```
ln -sf clementine-X.Y.ZZ current
```

Если отсутствует ссылка `/var/lib/clementine/current`, скрипты запуска не смогут определить расположение бинарных файлов текущего релиза, а также будет невозможна дальнейшая установка.

2.2. Настройка сервиса

Необходимо настроить Selinux и Firewall, чтобы они не блокировали работу сервиса.

Настройка Selinux и Firewall в данном руководстве не описывается.

Необходимо выполнить следующие действия:

1. Перейти в каталог `docker`:

```
cd /var/lib/clementine/current/extras/docker
```

2. В конфигурационный файл `docker-compose.yml` внести необходимые настройки, представленные в Таблице 1.

Таблица 1. Конфигурация Docker

#	Параметр	Описание
1	DOMAIN=	Домен приложения, используется для генерации ссылок в Email. В зависимости от конфигурации nginx, это может быть именной домен, например, вида <code>clementine.local</code> или IP адрес сервиса, доступный в сети
2	LUNA_ACCOUNT_ID=a0000000-0000-4000-a000-000000000000	Идентификатор аккаунта пользователя в VisionLabs LUNA PLATFORM 5

Параметры конфигурации SMTP представлены в Таблице 2.

Таблица 2. Конфигурация SMTP

#	Параметр	Описание
1	SMTP_SERVER=	Адрес сервера SMTP
2	SMTP_PORT=587	Порт сервера SMTP. Доступные значения: <ul style="list-style-type: none"> • 25 — без шифрования; • 465 — SSL; • 587 — TLS
3	SMTP_SENDER=	Email отправителя для отправки уведомлений пользователям, например, robot@clementine.local
4	SMTP_LOGIN=	Логин для подключения к SMTP серверу
5	SMTP_PASSWORD=	Пароль для подключения к SMTP серверу

Параметры конфигурации FaceStream представлены в Таблице 3.

Таблица 3. Конфигурация FaceStream

#	Параметр	Описание
1	FACESTREAM_HOST=127.0.0.1	Адрес сервера FaceStream
2	FACESTREAM_PORT=34569	Порт FaceStream

2.2.1. Запуск сервиса

Необходимо запустить сервис:

```
docker-compose up -d --build
```

2.2.2. Добавление аккаунта администратора

1. После того, как сервис запустится, необходимо запустить создание нового аккаунта и следовать указаниям.

```
docker-compose exec clementine_auth python3 cli.py createaccount
```

Для последующего входа в систему в качестве логина необходимо использовать указанный Email.

После создания аккаунта будет получен идентификатор аккаунта.

2. Необходимо запустить создание новой авторизации для аккаунта и следовать указаниям.

```
docker-compose exec clementine_auth python3 cli.py createauth
```

3. Необходимо запустить смену роли аккаунта и следовать указаниям.

```
docker-compose exec clementine_auth python3 cli.py setaccountrole
```


3. Установка и настройка на сервер

3.1. Установка сервиса

3.1.1. Подготовка и распаковка дистрибутива

Дистрибутив представляет собой архив вида «clementine-X.Y.ZZ», где «X.Y.ZZ» — численный идентификатор, обозначающий версию продукта.

Архив содержит все компоненты, необходимые для установки и эксплуатации системы.

Архив не включает зависимости, которые входят в стандартную поставку репозитория CentOS 7.4 x86_64 и могут быть загружены из открытых источников.

Перед процессом установки необходимо поместить файлы дистрибутива в директорию на сервере. Например, в директорию `/root/`. В данной директории не должно быть других файлов дистрибутива кроме целевых, используемых для установки конечного продукта.

Необходимо выполнить следующие действия:

1. Переключиться на пользователя `root`.

```
sudo su
```

2. Создать директорию для распаковки дистрибутива.

```
mkdir -p /var/lib/clementine
```

3. Переместить дистрибутив в созданную директорию.

```
mv /root/clementine* /var/lib/clementine
```

4. Установить архиватор `unzip`, если он не установлен.

```
yum install -y unzip
```

5. Перейти в папку с дистрибутивом.

```
cd /var/lib/clementine
```

6. Разархивировать файлы.

```
unzip clementine*.zip
```

7. Создать символическую ссылку, указав вместо «X.Y.ZZ» версию продукта. Ссылка указывает, что именно текущая версия дистрибутива используется для запуска.

```
ln -sf clementine-X.Y.ZZ current
```

Если отсутствует ссылка `/var/lib/clementine/current`, скрипты запуска не смогут определить расположение бинарных файлов текущего релиза, а также будет невозможна дальнейшая установка.

3.1.2. Настройка сервиса авторизации

Необходимо настроить Selinux и Firewall, чтобы они не блокировали работу сервиса.

Настройка Selinux и Firewall в данном руководстве не описывается.

Необходимо создать файл конфигурации:

1. Перейти в каталог auth.

```
cd /var/lib/clementine/current/clementine-auth
```

2. Скопировать базовый файл конфигурации в текущий.

```
cp .env.example .env
```

3. В конфигурационный файл `.env` внести необходимые настройки, соответствующие актуальным параметрам VisionLabs LUNA PLATFORM 5.

Параметры конфигурации clementine-auth представлены в Таблице 4.

Таблица 4. Параметры конфигурации clementine-auth

#	Параметр	Описание
1	PORT=9000	Порт, на котором будет запущен сервис
2	DOMAIN=	Домен приложения, используется для генерации ссылок в Email. В зависимости от конфигурации nginx, это может быть именной домен, например, <code>clementine.local</code> или IP адрес сервиса, доступный в сети
3	PROTOCOL=HTTP	Протокол, для работы в web-среде. Зависит от текущих настроек инфраструктуры и требований по безопасности. Возможные значения: <ul style="list-style-type: none"> • HTTP — протокол без шифрования; • HTTPS — протокол с SSL-шифрованием
4	LOG_LEVEL=0	Уровень логирования: <ul style="list-style-type: none"> • 0 — запись только ошибок; • 1 — запись ошибок и информационных сообщений; • 2 — запись ошибок, информационных сообщений и отладочной информации
5	LOG_FILE=	Файл для записи лога. Если файл не указывается, то пишется в <code>stderr/stdout</code>
6	CLEMTINE_AUTH_DSN=	Строка подключения к БД clementine_auth. Например, <code>postgresql://clemetine:clemetine@127.0.0.1/clementine_auth</code>

#	Параметр	Описание
7	LUNA_ACCOUNT_ID= a0000000-0000-4000-a000- 000000000000	Идентификатор аккаунта пользователя в VisionLabs LUNA PLATFORM 5
8	LUNA_ADMIN_DSN=	Строка подключения к БД luna_admin. Если подключение указано, будет происходить синхронизация контактов между clementine auth и luna admin. Например, <code>postgres://luna:luna@127.0.0.1/luna_admin</code>

Параметры конфигурации SMTP представлены в Таблице 5.

Таблица 5. Параметры конфигурации SMTP

#	Параметр	Описание
1	SMTP_SERVER=	Адрес сервера SMTP
2	SMTP_PORT=587	Порт сервера SMTP. Доступные значения: <ul style="list-style-type: none"> • 25 — без шифрования; • 465 — SSL; • 587 — TLS
3	SMTP_SENDER=	Email отправителя для отправки уведомлений пользователям, например <code>robot@clementine.local</code>
4	SMTP_LOGIN=	Логин для подключения к SMTP серверу
5	SMTP_PASSWORD=	Пароль для подключения к SMTP серверу

3.1.3. Настройка системы управления базами данных

Для настройки системы управления базами данных необходимо выполнить следующие действия:

1. Войти в интерфейс БД.

```
psql -U postgres
```

2. Создать пользователя БД.

```
create role clementine;
```

3. Назначить для пользователя пароль.

```
ALTER USER clementine WITH PASSWORD 'clementine';
```

4. Создать БД.

```
CREATE DATABASE clementine_auth;
```

5. Назначить права для пользователя, созданного ранее.

```
GRANT ALL PRIVILEGES ON DATABASE clementine_auth TO clementine;
```

6. Дать возможность пользователю авторизоваться в БД.

```
ALTER ROLE clementine WITH LOGIN;
```

7. Выйти.

```
\q
```

3.1.4. Установка зависимостей для clementine-auth

Необходимо перейти в директорию clementine-auth.

Для корректной установки зависимостей сервиса в системе должны быть установлены пакеты gcc, python3-devel.

Необходимо установить системные пакеты, если они еще не установлены.

```
yum install -y gcc python3-devel
```

Необходимо выполнить следующие действия:

1. Перейти в директорию модуля.

```
cd /var/lib/clementine/current/clementine-auth
```

2. Создать виртуальное окружение.

```
python3.6 -m venv venv
```

3. Активировать виртуальное окружение.

```
source venv/bin/activate
```

4. Запустить установку зависимостей Python.

```
pip3.6 install -r requirements.txt
```

5. Запустить наполнение БД структурой.

```
python3.6 ./base_scripts/dbs_migrate.py
```

6. Деактивировать виртуальную среду.

```
deactivate
```

3.1.5. Добавление аккаунта администратора

Для добавления аккаунта администратора необходимо выполнить следующие действия:

1. Перейти в директорию модуля.

```
cd /var/lib/clementine/current/clementine-auth
```

2. Активировать виртуальное окружение.

```
source venv/bin/activate
```

3. Запустить создание нового аккаунта и следовать указаниям.

```
python3.6 ./cli.py createaccount
```

Для последующего входа в систему в качестве логина необходимо использовать указанный Email.

После создания аккаунта будет получен идентификатор аккаунта.

4. Запустить создание новой авторизации для аккаунта и следовать указаниям.

```
python3.6 ./cli.py createauth
```

5. Запустить смену роли аккаунта и следовать указаниям.

```
python3.6 ./cli.py setaccountrole
```

6. Деактивировать виртуальную среду.

```
deactivate
```

3.1.6. Сервисные файлы

Для сервиса в поставке присутствует unit-файл.

1. Необходимо создать символическую ссылку unit-файла в папку `system` для запуска сервиса после установки.

Следует скопировать файлы сервисов из комплекта поставки в системную директорию.

```
cp /var/lib/clementine/current/extras/systemd/clementine_auth.service  
/etc/systemd/system/clementine_auth.service
```

2. Необходимо перезагрузить системные сервисы.

```
systemctl daemon-reload
```

3.1.7. Запуск сервиса

Необходимо запустить сервис `clementine-auth` и добавить его в автозагрузку.

```
systemctl start clementine_auth.service  
systemctl enable clementine_auth.service
```

3.2. Настройка сервиса для работы с отделениями `clementine-departments`

Необходимо настроить `Selinux` и `Firewall`, чтобы они не блокировали работу сервиса.

Настройка `Selinux` и `Firewall` в данном руководстве не описывается.

Необходимо выполнить следующие действия:

1. Создать файл конфигурации. Перейти в каталог departments.

```
cd /var/lib/clementine/current/clementine-departments
```

2. Скопировать базовый файл конфигурации в текущий.

```
cp .env.example .env
```

3. В конфигурационный файл .env внести необходимые настройки, соответствующие актуальным параметрам VisionLabs LUNA PLATFORM 5.

Конфигурация clementine-departments представлена в Таблице 6.

Таблица 6. Конфигурация clementine-departments

#	Параметр	Описание
1	PORT=9100	Порт, на котором будет запущен сервис
2	LOG_LEVEL=0	Уровень логирования: <ul style="list-style-type: none"> • 0 — запись только ошибок; • 1 — запись ошибок и информационных сообщений; • 2 — запись ошибок, информационных сообщений и отладочной информации
3	CLEMENTINE_DEPARTMENTS_DSN=	Строка подключения к БД clementine-departments. Например, postgresql://clementine:clementine@127.0.0.1/departments

3.2.1. Установка зависимостей для clementine-departments

Необходимо перейти в директорию clementine-departments.

Для корректной установки зависимостей сервиса в системе должны быть установлены пакеты gcc, python3-devel

Необходимо выполнить следующие действия:

1. Перейти в директорию модуля.

```
cd /var/lib/clementine/current/clementine-departments
```

2. Создать виртуальное окружение.

```
python3.6 -m venv venv
```

3. Активировать виртуальное окружение.

```
source venv/bin/activate
```

4. Запустить установку зависимостей Python.

```
pip3.6 install -r requirements.txt
```

5. Запустить наполнение БД структурой.

```
python3.6 ./base_scripts/dbs_migrate.py
```

6. Деактивировать виртуальную среду.

```
deactivate
```

3.2.2. Сервисные файлы

Для сервиса в поставке присутствует unit-файл.

1. Необходимо создать символическую ссылку unit-файла в папку `system` для запуска сервиса после установки.

Следует скопировать файлы сервисов из комплекта поставки в системную директорию.

```
cp /var/lib/clementine/current/extras/systemd/  
clementine_departments.service/etc/systemd/system/clementine_departments.service
```

2. Необходимо перезагрузить системные сервисы.

```
systemctl daemon-reload
```

3.2.3. Запуск сервиса

Необходимо запустить сервис `clementine-departments` и добавить его в автозагрузку.

```
systemctl start clementine_departments.service  
systemctl enable clementine_departments.service
```

3.3. Настройка FaceStream и установка UI

3.3.1. Настройка FaceStream

Для корректной работы VisionLabs LUNA PLATFORM THERMO сервис FaceStream необходимо запускать в серверном режиме.

Процесс установки, первичной настройки и запуска FaceStream в серверном режиме описан в руководстве пользователя FaceStream.

В конфигураторе сервиса FaceStream необходимо использовать аккаунт `Luna-Account-id`, указанный в настройках сервиса `clementine-auth` (п. 3.1.2)¹, чтобы у сервиса VisionLabs LUNA PLATFORM THERMO была возможность просмотра сгенерированных событий.

¹ — По умолчанию в сервисе VisionLabs LUNA PLATFORM THERMO используется `Luna-Account-id` «a0000000-0000-4000-a000-000000000000». Если установка сервиса производится на систему VisionLabs LUNA PLATFORM 5, в которой уже имеются данные, необходимо указать корректный `Luna-Account-id` для совместного использования данных между сервисами.

3.3.2. Установка и настройка UI

Для работы LUNA PLATFORM THERMO UI необходим веб-сервер nginx.

Необходимо выполнить следующие действия:

1. Установить nginx из репозитория CentOS.

```
yum -y --nogpgcheck install nginx
```

2. Запустить сервис nginx и добавить его в автозагрузку.

```
sudo systemctl start nginx  
sudo systemctl enable nginx
```

3. Скопировать файл конфигурации LUNA PLATFORM THERMO UI для nginx.

```
cp /var/lib/clementine/current/extras/nginx/clementine.conf  
/etc/nginx/conf.d/clementine.conf
```

4. Внести необходимые правки в файл конфигурации при необходимости.

```
vim /etc/nginx/conf.d/clementine.conf
```

5. Перезагрузить сервис nginx для применения файла конфигурации.

```
sudo systemctl reload nginx
```

Файл конфигурации:

```
upstream facestreamapi {  
    # УКАЗАТЬ АКТУАЛЬНЫЙ АДРЕС FaceStream сервера  
    server 127.0.0.1:34569;  
}  
upstream lp5api {  
    # УКАЗАТЬ АКТУАЛЬНЫЙ АДРЕС LUNA PLATFORM 5 API  
    server 127.0.0.1:5000;  
}  
upstream senderapi {  
    # УКАЗАТЬ АКТУАЛЬНЫЙ АДРЕС LUNA Sender API  
    server 127.0.0.1:5080;  
}  
upstream departmentsapi {  
    # УКАЗАТЬ АКТУАЛЬНЫЙ АДРЕС clementine-departments API  
    server 127.0.0.1:9100;  
}  
upstream authapi {  
    # УКАЗАТЬ АКТУАЛЬНЫЙ АДРЕС clementine-auth API  
    server 127.0.0.1:9000;  
}  
  
server {  
    client_max_body_size 20M;  
    server_name _;  
    # ВАЖНО!!! Если указывается имя сервера, оно также должно быть указано в
```


конфигурации сервиса clementine-auth в параметре DOMAIN=. Если имя сервера не указывается, в Clementine Auth в параметре DOMAIN= следует указать IP-адрес сервера, на котором запущен NGINX.

```
listen 80 default_server;
listen [::]:80 default_server;

location ~* ^.+\.(\?:css|js|ttf|eot|woff|woff2|svg)$ {
    access_log off;
    expires 1y;
    add_header Cache-Control public;
}

root /var/lib/clementine/current/clementine-ui/build;

location / {
    try_files $uri /index.html =404;
    add_header Cache-Control "no-store, no-cache, must-revalidate";
}

location /api/lp5/ {
    auth_request /auth;
    auth_request_set $luna_account_id $upstream_http_luna_account_id;
    auth_request_set $original_luna_account_id
$upstream_http_x_original_luna_account_id;
    proxy_set_header Luna-Account-id $luna_account_id;
    proxy_set_header X-Original-Luna-Account-id $luna_account_id;

    rewrite /api/lp5/(.*) /$1 break;
    proxy_set_header Host $http_host;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_redirect off;
    proxy_buffering off;
    proxy_http_version 1.1;
    proxy_set_header Connection "";
    proxy_pass http://lp5api;
}

location /api/fs4/ {
    auth_request /auth;
    auth_request_set $luna_account_id $upstream_http_luna_account_id;
    auth_request_set $original_luna_account_id
$upstream_http_x_original_luna_account_id;
    proxy_set_header Luna-Account-id $luna_account_id;
    proxy_set_header X-Original-Luna-Account-id $luna_account_id;

    rewrite /api/fs4/(.*) /$1 break;
    proxy_set_header Host $http_host;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_redirect off;
    proxy_buffering off;
    proxy_http_version 1.1;
    proxy_set_header Connection "";
```

```
    proxy_pass http://facestreamapi;
}

location /api/sender/ {
    auth_request /auth;
    auth_request_set $luna_account_id $upstream_http_luna_account_id;
    auth_request_set $original_luna_account_id
$upstream_http_x_original_luna_account_id;
    proxy_set_header Luna-Account-id $luna_account_id;
    proxy_set_header X-Original-Luna-Account-id $luna_account_id;

    rewrite /api/sender/(.*) /$1 break;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_http_version 1.1;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header Host $http_host;
    proxy_pass http://senderapi;
}

location /api/artemis/ {
    auth_request /auth;
    auth_request_set $luna_account_id $upstream_http_luna_account_id;
    auth_request_set $original_luna_account_id
$upstream_http_x_original_luna_account_id;
    proxy_set_header Luna-Account-id $luna_account_id;
    proxy_set_header X-Original-Luna-Account-id $luna_account_id;

    rewrite /api/artemis/(.*) /$1 break;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header Host $http_host;
    proxy_set_header X-Service clementine-ui;
    proxy_pass http://authapi;
}

location /api/artemis/auth/ {
    rewrite /api/artemis/(.*) /$1 break;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    proxy_set_header Host $http_host;
    proxy_set_header X-Service clementine-ui;
    proxy_pass http://authapi;
}

location /api/hermes/ {
    auth_request /auth;
    auth_request_set $luna_account_id $upstream_http_luna_account_id;
    auth_request_set $original_luna_account_id
$upstream_http_x_original_luna_account_id;
    proxy_set_header Luna-Account-id $luna_account_id;
    proxy_set_header X-Original-Luna-Account-id $luna_account_id;

    rewrite /api/hermes/(.*) /$1 break;
```

```
proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
proxy_set_header Host $http_host;
proxy_set_header X-Service clementine-ui;
proxy_pass http://departmentsapi;
}

location = /auth {
    proxy_pass http://authapi/auth/proxy;
    proxy_set_header X-Service clementine-ui;
    proxy_pass_request_body off;
    proxy_set_header Content-Length "";
    proxy_set_header X-Original-URI $request_uri;
}
}
```

По умолчанию UI использует порт 80.

Для перехода в UI необходимо ввести: `http://<correct_address>:80`.

4. Удаление сервиса

Удаление выполняется путем последовательного отключения сервисов и удаления сервисного файла.

При удалении Сервиса необходимо отключить LUNA PLATFORM THERMO UI.

Необходимо выполнить следующие действия:

1. Переключиться на пользователя root.

```
sudo su
```

2. Удалить конфигурационный файл для nginx.

```
rm /etc/nginx/conf.d/clementine.conf
```

3. Перезагрузить сервис nginx.

```
systemctl reload nginx
```

4. Отключить сервис clementine-auth.

Перейти в директорию auth.

```
cd /var/lib/clementine/current/clementine-auth
```

5. Активировать виртуальное окружение.

```
source venv/bin/activate
```

6. Запустить очистку БД.

```
python3.6 ./base_scripts/dbs_downgrade.py
```

7. Деактивировать виртуальную среду.

```
deactivate
```

8. Удалить сервисный файл.

```
rm /etc/systemd/system/clementine_auth.service
```

9. Убрать из автозагрузки сервис clementine_auth.

```
systemctl stop clementine_auth  
systemctl disable clementine_auth
```

10. Отключить сервис clementine-departments.

Перейти в директорию departments.

```
cd /var/lib/clementine/current/clementine-departments
```

11. Активировать виртуальное окружение.

```
source venv/bin/activate
```

12. Запустить очистку БД.

```
python3.6 ./base_scripts/dbs_downgrade.py
```

13. Деактивировать виртуальную среду.

```
deactivate
```

14. Удалить сервисный файл.

```
rm /etc/systemd/system/clementine_departments.service
```

15. Убрать из автозагрузки сервис clementine-departments.

```
systemctl stop clementine_departments  
systemctl disable clementine_departments
```

16. Удалить дистрибутив VisionLabs LUNA PLATFORM THERMO.

```
rm -rf /var/lib/clementine/current
```

5. Логирование

Ошибки загрузки данных записываются в системный лог посредством подсистемы логирования с тремя уровнями логирования:

- 0 — запись только ошибок;
- 1 — запись ошибок и информационных сообщений;
- 2 — запись ошибок, информационных сообщений и отладочной информации.

Внутренние ошибки записываются в системный лог посредством подсистемы логирования:

- с уровнем важности «ошибка» в случае, если дальнейшее выполнение программы невозможно;
- с уровнем важности «предупреждение», если программа сохраняет работоспособность.