



VisionLabs
MACHINES CAN SEE

VisionLabs LUNA Access

Инструкция по установке

Содержание

Глоссарий	3
Введение	4
Системные требования	5
Требования к программному обеспечению	5
Требования к аппаратному обеспечению рабочей станции	5
1 Лицензирование	6
2 Развертывание Сервиса	7
2.1 Установка Docker и Docker Compose	7
2.2 Подготовка и распаковка дистрибутива	7
3 Управление учетными записями	12
3.1 Добавление учетной записи	12
3.2 Просмотр списка учетных записей	12
3.3 Удаление учетной записи	12
4 Обновление Сервиса	14
5 Удаление Сервиса	15
6 Мониторинг внутренних задач с помощью Flower	16

Глоссарий

Термин	Определение
Docker	Платформа для разработки, доставки и запуска контейнерных приложений. Позволяет создавать контейнеры, автоматизировать их запуск и развертывание, управляет жизненным циклом. Позволяет запускать множество контейнеров на одной хост-машине
Docker Compose	Позволяет разворачивать и настраивать несколько контейнеров одновременно
База данных (БД)	Совокупность данных, хранимых в соответствии со схемой данных, манипулирование которыми выполняют в соответствии с правилами средств моделирования данных
Идентификация	Поиск наиболее подходящего биометрического шаблона лица путем сравнения векторов признаков фотоизображения лица с перечнем аналогичных биометрических шаблонов в базе (один ко многим)
Контейнер	Способ упаковать приложение и все его зависимости в единый образ. Этот образ запускается в изолированной среде, не влияющей на основную операционную систему. Контейнеры позволяют отделить приложение от инфраструктуры
Образ	Неизменяемый образ, из которого разворачивается контейнер. Набор файлов, необходимых для запуска и работы приложения на другом хосте
Программное обеспечение (ПО)	Программа или множество программ, используемых для управления компьютером
Система контроля управления доступом (СКУД)	Совокупность программно-аппаратных технических средств, направленных на контроль входа и выхода в помещение с целью обеспечения безопасности и регулирования посещения определенного объекта. Например, турникеты на входе в банки/офисные здания
Событие	Зафиксированный системой факт детекции с извлечением атрибутов обработчиком

Введение

Настоящий документ описывает процесс установки и настройки сервиса VisionLabs Luna Access (далее — Сервис или Access), а также содержит аппаратные и программные требования к ПО.

Процесс настройки и установки необходимо выполнять под учетной записью суперпользователя (с root правами).

Количество подключенных камер, терминалов и турникетов может быть любым и зависит от требований к разворачиваемой системе, но ограничено лицензией на продукты VisionLabs и возможностью СКУД.

В зависимости от выбранного решения контроль доступа может применяться с применением Face ID или магнитной карты-пропуска.

Системные требования

Требования к программному обеспечению

Для работы Access необходимо, чтобы выполнялись требования к программному обеспечению, приведенные в Таблице 1.

Таблица 1. Требования к программному обеспечению

Необходимый ресурс	Рекомендовано
Операционная система (ОС)	РЕД ОС 7.3
Docker	v.20.10 и новее
Docker-compose	v.1.29 и новее

Требования к аппаратному обеспечению рабочей станции

Для установки Access необходимо, чтобы выполнялись требования к аппаратным ресурсам, приведенные в Таблице 2.

Таблица 2. Требования к аппаратному обеспечению

Ресурс	Рекомендовано
Процессор (CPU)	64-битный процессор Intel или AMD, с 2 ядрами и более, с тактовой частотой 2,0 ГГц и выше
Оперативная память (RAM)	4 ГБ и выше
Свободное место на диске (HDD/SSD)	20 ГБ и выше

1 Лицензирование

Для работы сервиса Access не требуется лицензия.

Лицензированию могут быть подвержены внешние системы и сервисы, которые используются в интеграции. Лицензия в этом случае приобретается отдельно у правообладателя.

2 Развертывание Сервиса

Данный раздел описывает установку и использование Docker Compose для развертывания Сервиса.

Docker и Docker Compose не входят в дистрибутив Сервиса.

Docker позволяет создать контейнер, в котором уже имеется требуемый Сервис, программная среда для него и минимальный набор необходимых инструментов. Docker Compose используется для автоматического развертывания контейнеров.

2.1 Установка Docker и Docker Compose

Используйте официальную инструкцию для установки [Docker Compose](#) для РЕД ОС.

Проверьте Docker:

```
systemctl status docker
```

Ответ должен содержать статус Active (running).

2.2 Подготовка и распаковка дистрибутива

Дистрибутив представляет собой архив вида «vl-access-2-v2.9.0».

Ссылку для скачивания дистрибутива необходимо запросить у представителя VisionLabs.

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

Архив не включает зависимости, которые входят в стандартную поставку репозитория РЕД ОС и могут быть загружены из открытых источников.

Запуск Сервиса осуществляется из Docker образа.

Для запуска необходимо выполнить следующие действия:

Установите архиватор unzip, если он не установлен. В команде указана версия 6.0-21, в работе более поздних версий замечены ошибки:

```
yum install unzip-6.0-21.el7
```

Распакуйте файлы дистрибутива:

```
unzip vl-access-2-v2.9.0.zip -d vl-access
```

После распаковки появится директория, содержащая:

- README_FOR_ENGINEERS.md файл с описанием быстрого запуска;
- .env файл конфигурации.
- конфигурацию docker-compose.yml;
- архив изменений CHANGELOG.md;
- архив образа vl_access_2_v2.9.0.tar.gz.

В случае изменения расположения программно-аппаратной части БД в docker-compose, выполните команду для доступа из контейнера к хранению данных PostgreSQL (выполнить единоразово для образа):

```
docker volume create --name=db_data
```

Произведите настройку с помощью файла .env:

```
nano .env
```

Параметр	Описание	Значения по умолчанию
Параметры FastAPI		
DEBUG	Режим отладки Сервиса - вывод в логах ОС и в интерфейсе информации типа Debug о работе Сервиса	0
	1 – отладка.	
	0 – без отладки.	
VL_ACCESS_TAG	Тэг сервиса, берется из внутренних настроек. Не рекомендуется изменять этот параметр.	2.9.0
VL_ACCESS_PORT	Порт backend сервиса.	9091
DB_USER	Логин пользователь базы данных.	postgres
DB_HOST	Хост на котором развернута БД. По умолчанию БД разворачивается там же, где и сервис.	postgres

Параметр	Описание	Значения по умолчанию
DB_PORT	Порт на котором развернута БД	5432
DB_NAME	Имя БД	postgrese_base
DB_PASSWORD	Пароль пользователя базы данных.	postgres_password
LOG_DB_USER	Логин к БД для сбора логов.	postgres
LOG_DB_HOST	Имя хоста для сбора логов БД.	postgres
LOG_DB_PORT	Порт на котором развернута БД.	5432
LOG_DB_NAME	Имя БД.	postgrese_base
LOG_DB_PASSWORD	Пароль для подключения к логам БД.	postgres_password
C_FORCE_ROOT	Принудительный запуск Celery от имени root пользователя.	true
MAX_CONCURRENCY	Максимальное кол-во процессов у компонента Worker, которые могут обрабатываться параллельно. Задается исходя от нагрузки на систему.	20
Параметры БД PostgreSQL		
POSTGRES_USER	Логин пользователя базы данных.	postgres
POSTGRES_PASSWORD	Пароль пользователя базы данных.	postgres_password
POSTGRES_HOST	Хост на котором развернута БД. По умолчанию БД разворачивается там же, где и сервис.	postgres
Параметры Redis		
REDIS_DSN	Расположения сервиса Redis. Не рекомендуется менять путь.	redis://redis/0
Параметры Rabbit		

Параметр	Описание	Значения по умолчанию
RABBITMQ	Название брокера очередей сообщений. В данной версии поддерживается только Rabbit MQ.	rabbitmq
RABBITMQ_USER	Логин пользователя для подключения к Rabbit MQ	guest
RABBITMQ_PASSWORD	Пароль для подключения к Rabbit MQ	guest
RABBITMQ_PROTOCOL	Тип протокола RabbitMQ. Поддерживается только AMQP.	amqp
Параметры подключения FrontEnd		
BACKEND_HOST	Название контейнера backend сервиса.	fastapi
BACKEND_PORT	Порт для подключения к backend сервиса.	9091
Параметры Celery		
CELERY_BROKER_URL	Параметры менеджера очередей сообщений Celery	<pre> \${ RABBITMQ_PROTOCOL }:// \${ RABBITMQ_USER}:\$ { RABBITMQ_PASSWORD }@ \${RABBITMQ }:5672/ </pre>

Импортируйте образ:

```
docker load < vl-access-2-images-v2.9.0.tar.gz
```

Запустите проект:

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

Проверьте доступность Access по адресу: http://<IP_address>:9092/.

Для доступа к функционалу Сервиса создайте администратора (см. раздел [Управление учетными записями](#)).

3 Управление учетными записями

3.1 Добавление учетной записи

В Access данной версии поддерживается создание учетной записи только с ролью Администратора. Более подробно о ролях и доступах см. в Руководстве пользователя раздел Роли в сервисе.

1. Запустите скрипт создания администратора

```
docker-compose exec fastapi python backend/manage.py createadmin
```

2. Следуйте указаниям консоли.

```
(venv) [root@localhost vl-access-2]# docker-compose exec fastapi python
  backend/manage.py createadmin
Admin creation:
Enter your login: admin
Enter your password:
Confirm your password
Admin user created successfully
```

3. Проверьте корректность создания администратора – войдите в созданную учетную запись `http://<IP_address>:9092/`.

3.2 Просмотр списка учетных записей

1. Выполните команду просмотра списка созданных администраторов:

```
docker-compose exec fastapi python backend/manage.py listadmin
```

В ответе на запрос консоль выдаст список созданных учетных записей:

```
(venv) [root@localhost vl-access-2]# docker-compose exec fastapi python
  backend/manage.py listadmin
admin
admin2
```

3.3 Удаление учетной записи

Получите информацию о логине необходимо аккаунта для удаления с помощью команды [просмотра](#).

1. Выполните команду для удаления администратора:

```
docker-compose exec fastapi python backend/manage.py deleteadmin
```

2. Введите логин, который необходимо удалить.

В ответе на запрос консоль выдаст сообщение об успешном удалении:

```
(venv) [root@localhost vl-access-2]# docker-compose exec fastapi python
  backend/manage.py deleteadmin
Admin deletion:
Enter your login: admin2
Admin was deleted successfully.
```

4 Обновление Сервиса

Для обновления Сервиса необходимо выполнить повторную установку и запуск Сервиса с новой версией архива, аналогично Разделу [Развертывание Сервиса](#).

5 Удаление Сервиса

Удаление Access выполняется путем остановки и удаления запущенных контейнеров Docker. Выполните следующие действия.

1. Перейдите в директорию с файлом `docker-compose.yml`. Остановите запущенные контейнеры:

```
docker-compose down
```

2. Удалите директорию установленного Сервиса:

```
rm -rf vl-access-2-v2.9.0
```

6 Мониторинг внутренних задач с помощью Flower

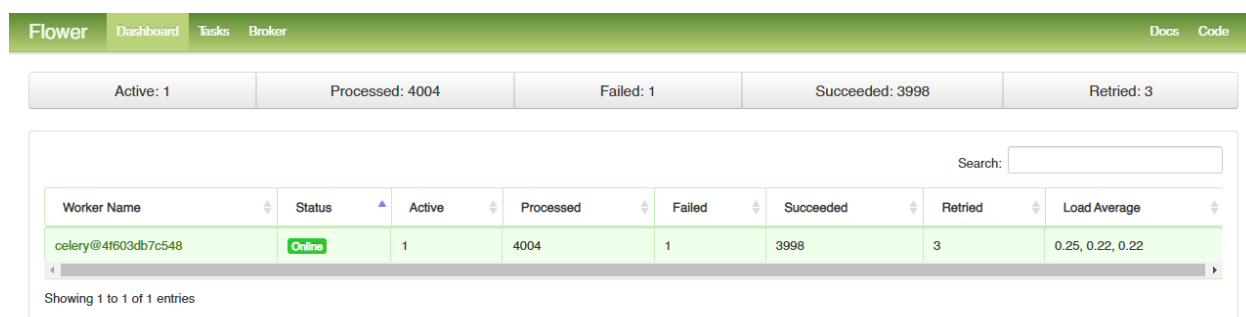
Сервис Flower позволяет отслеживать список задач и их статус, выполняемый компонентом Worker.

В данном разделе будут описаны только функции необходимые в контексте мониторинга Access, алгоритмы работы с Flower описаны в официальной [справке](#).

Задачи выполняемые Worker:

- проверка состояния компонентов `is_alive`;
- создание компонентов в модулях;

Flower располагается по адресу `http://<ip_address_Access>:9093` (Рисунок 1).



The screenshot shows the Flower Dashboard with a green header. The main content area has a summary bar with five boxes: Active: 1, Processed: 4004, Failed: 1, Succeeded: 3998, and Retried: 3. Below this is a table with columns: Worker Name, Status, Active, Processed, Failed, Succeeded, Retried, and Load Average. A single row is visible for worker 'celery@4f603db7c548' with status 'Online' and 1 failed task.

Worker Name	Status	Active	Processed	Failed	Succeeded	Retried	Load Average
celery@4f603db7c548	Online	1	4004	1	3998	3	0.25, 0.22, 0.22

Рис. 1: Dashboard Flower

На страницу Dashboard необходимо отслеживать количество задач со статусом Failed (Рисунок 2).



This screenshot is identical to the previous one, but the 'Failed: 1' box in the summary bar is highlighted with a red rectangle to draw attention to it.

Рис. 2: Статус Failed

При появлении задач со статус Failed необходимо разобрать каждый конкретный случай.

Для просмотра задач нажмите на Failed, при этом откроется страница Task с фильтром статуса задач Failure (Рисунок 3).

Name	UUID	State	args	kwargs	Result	Received	Started	Runtime	Worker
Worker	869250a1-dfba-4589-989f-fa8292954466	FAILURE	()	{'component_type': 'devices', 'component_id': 'a794d519-392c-4198-9d60-857abd3fbd31'}		2023-02-16 10:17:38.195	2023-02-16 10:18:38.282		celery@4f603db7c548

Рис. 3: Страница Tasks с фильтром Failure

В колонке **kwarg** указано сообщение с описанием компонента, который сломался:

```
{'component_type': 'devices', 'component_id': 'a794d519-392c-4198-9d60-857abd3fbd31'}
```

- component_type - тип компонента: devices, service, controller, pipeline.
- component_id - UUID компонента.

Для просмотра логов выполнения задачи нажмите на идентификатор в колонке UUID, при этом откроется детальная информация по задаче (Рисунок 4).

Basic task options		Advanced task options	
Name	Worker	Received	2023-02-16 10:17:38.195038 UTC
UUID	869250a1-dfba-4589-989f-fa8292954466	Started	2023-02-16 10:18:38.282951 UTC
State	FAILURE	Failed	2023-02-16 10:18:38.408056 UTC
args	()	Retried	2023-02-16 10:17:38.193645 UTC
kwargs	{'component_type': 'devices', 'component_id': 'a794d519-392c-4198-9d60-857abd3fbd31'}	ETA	2023-02-16T07:18:38.190915+00:00
Result	None	Retries	3
		Worker	celery@4f603db7c548
		Exception	MaxRetriesExceededError("Can't retry Worker[869250a1-dfba-4589-989f-fa8292954466] args:() kwargs:{'component_type': 'devices', 'component_id': 'a794d519-392c-4198-9d60-857abd3fbd31'}")
		Timestamp	2023-02-16 10:18:38.408056 UTC
		Traceback	Traceback (most recent call last): File "/usr/local/lib/python3.9/site-packages/urllib3/connection.py", line 174, in _new_conn conn = connection.create_connection(File "/usr/local/lib/python3.9/site-packages/urllib3/util/connection.py", line 95, in create_connection raise err File "/usr/local/lib/python3.9/site-packages/urllib3/uti

Рис. 4: Детальная информация по задаче

Далее необходимо перейти в Access UI в указанный раздел, найти устройство с указанным идентификатором и изучить логи.

При использовании большого количества устройств рекомендуется перейти по ссылке: http://<ip_address_Access>:9092/<component_type>/<component_id>