



ООО “ТРИДИВИ”

Система интеллектуальной обработки, анализа и распознавания изображений с прикладным сценарием удаленной идентификации человека “3DiVi Biometric Anti-Fraud”

Инструкция по установке

2023 г.

Содержание

Содержание	2
Аннотация	3
Глоссарий	3
Обзор	5
Системные требования	6
Установка 3DiVi BAF	7
Удаление 3DiVi BAF	14

Аннотация

Настоящий документ описывает процесс установки экземпляра программного обеспечения 3DiVi BAF.

Глоссарий

Bbox (Bounding box) - ограничивающий прямоугольник вокруг задетектированного лица. Координаты bbox рассчитываются относительно координат исходного изображения.

Liveness – проверка на принадлежность лица реальному человеку.

Авторизация – предоставление допуска для получения услуг, предоставляемых Заказчиком, при повторном заходе в систему после успешной регистрации.

Аппликант – запрос на верификацию от пользователя, включающий в себя количество произведенных попыток верификации и ее текущий статус, а также другие сопутствующие верификации данные.

Биометрический шаблон лица - уникальный набор биометрических признаков, извлеченных из изображения лица. Шаблоны позволяют сравнить два изображения лица и определить степень их схожести.

Верификация – многоэтапный процесс, задача которого заключается в сравнении предоставленного документа и селфи с полученными при регистрации вариантами документа и лица, и подтверждении их принадлежности одному человеку. Технология применяется для подтверждения личности пользователя.

Идентификация – проверка на совпадение полученных селфи и информации с фото документа с информацией и изображением из базы данных. Таким образом алгоритмы выполняют поиск пользователя по базе и идентифицируют его.

Предъявитель - пользователь системы, участвующий в процессах авторизации, регистрации и верификации.

Регистрация – создание записи о пользователе в базе данных для возможности дальнейшего получения услуг, предоставляемых Заказчиком.

Селфи – автопортретная фотография, снятая на веб-камеру или камеру мобильного/портативного устройства без применения таймера или штатива.

Черный список – базовый механизм контроля доступа к ресурсам Заказчика исключительно для верифицированных пользователей.

Обзор

Система интеллектуальной обработки, анализа и распознавания изображений с прикладным сценарием удаленной идентификации человека 3DiVi Biometric Anti-Fraud (Далее 3DiVi BAF) - система, состоящая из набора функциональных подсистем, встраиваемых компонентов захвата изображения и программного интерфейса (API), которые обеспечивают возможность реализации взаимодействия информационных систем для решения задач по регистрации, авторизации и верификации пользователей на основе изображений в разрезе следующих функциональных процедур:

- распознавание лиц на изображениях
- распознавание документов на изображениях
- анализ качества изображений
- работа с биометрическими шаблонами
- проверка на соответствие изображения лица в документе и предъявляемого селфи пользователя
- проверка лиц на Liveness и документов на подлог
- журналирование сессии и данных об окружении пользователя
- проверка по спискам пользователей

Запросы в Систему направляются через встраиваемые в приложения Web компоненты и через API Системы.

В зависимости от проверок, выбранных в настройках Системы, и применимых к входящим изображениям, могут использоваться подсистемы распознавания лиц и распознавания документов, как совместно, так и по отдельности.

Системные требования

Минимальные системные требования для развертывания комплекса 3DiVi BAF.

Для развертывания сервера комплекса необходимо подготовить машину со следующими характеристиками:

- ЦП:
 - 4 ядра (8 потоков)
 - Частота - 3.5 ГГц
 - 15 МБ кеша
 - Расширения набора команд: AVX/AVX2

Пример подходящего ЦП - Intel® Xeon® E5-2637 v4

- ОЗУ - 32 ГБ
- Накопитель SSD:
 - Свободное пространство от 128 ГБ
 - Не менее 20% свободного места в файловой системе

В случае развертывания отдельного сервера БД (по умолчанию сервер базы данных запускается на машине, где развернута подсистема) потребуется машина с характеристиками ниже:

- ЦП: 2 ядра
- 8 ГБ ОЗУ
- PostgreSQL >= 11.2
- SSD/HDD 100 ГБ

Убедитесь, что на машине установлено следующее ПО:

- ОС Ubuntu 20.04.4
- Docker версии - 20.10.17
- Kubernetes версии - 1.23.8
- Helm версии - 3.10.3

Для установки Docker, Kubernetes и Helm можно воспользоваться скриптом `on_premise/setup/install-packages.sh`, поставляемым вместе с дистрибутивом (требуется подключение к интернету).

Возможен более тонкий подбор комплектующих для сервисов под реальную нагрузку и параметры работы пользователя.

Установка 3DiVi BAF

Подготовка

Скачайте и распакуйте дистрибутив подсистемы на машину, где планируется выполнить установку.

Откройте системную консоль, перейдите в директорию `on-premise` внутри дистрибутива и проверьте содержимое папки, используя команду:

```
$ find -maxdepth 1
```

Файлы и папки, содержащиеся в дистрибутиве, будут выведены в консоль. Пример вывода:

```
./deploy
./face_sdk.lic
./ingress-nginx-4.2.0.tgz
./integration_tests
./kube-flannel.yml
./license_server
./nvidia-device-plugin-0.12.2.tgz
./platform_images.tar.gz
./setup
./upload_script
```

Основные элементы дистрибутива:

- **./license_server** - файлы необходимые для запуска лицензионного сервера, с помощью которого осуществляется лицензирование подсистемы;
- **./integration_tests** - скрипты для автоматического тестирования подсистемы после развертывания;
- **./setup/settings.env** - файл конфигурации экземпляра подсистемы.
- **./upload_script** - папка, в которой находится скрипт загрузки изображений из датасета для создания профилей в подсистеме.

Дальнейшие команды выполняются в системной консоли из директории **on_premise**.

Загрузка образов

Загрузите в локальный **registry** образы из архива:

```
$ sudo docker load -i platform_images.tar.gz
```

Загрузка образов может занять около пяти минут.

Ввод переменных окружения

Откройте файл конфигурации `./setup/settings.env`, используя текстовый редактор, и установите значения следующих переменных:

- **MASTER_NODE_IP_ADDRESS** - IP-адрес машины, на которой выполняется развертывание. Вы можете узнать его у Вашего системного администратора. **DOMAIN** - корневое доменное имя. После развертывания доступ к API и веб-интерфейсу подсистемы будет осуществляться по адресу `http://platform.$DOMAIN`. IP-адрес для доменного имени `platform.$DOMAIN` должен быть сконфигурирован на DNS-сервере.
- **RABBIT_USER** и **RABBIT_PASSWORD** - имя пользователя и пароль для доступа к брокеру сообщений, используется для внутреннего взаимодействия сервисов подсистемы. Задайте произвольное имя, состоящее из латинских букв, без пробелов и пароль, состоящий из латинских букв и цифр, без пробелов.
- **BAF_USER**, **BAF_PASSWORD** - имя пользователя и пароль для подсистемы BAF. Пароль должен состоять не менее чем из 6 символов и содержать буквы в нижнем и верхнем регистре, а также цифру и небуквенные символы.
- **POSTGRES_USER**, **POSTGRES_PASSWORD** - параметры подключения к базе данных. При первом развертывании подсистемы задайте произвольное имя пользователя и название базы данных, состоящих из латинских букв, без пробелов и сгенерируйте пароль, состоящий из латинских букв и цифр. База данных будет создана автоматически.
- **SERVICE_KEY** - секретный ключ, необходимый для внутреннего взаимодействия сервисов подсистемы. Сгенерируйте произвольную строку, состоящую из латинских букв и цифр без пробелов.
- **LIC_KEY** - лицензионный ключ. Обычно ключ отправляется в сопроводительном письме с дистрибутивом. При отсутствии ключа обратитесь к вашему менеджеру по продажам.
- **PLATFORM_ADMIN_EMAIL** и **PLATFORM_ADMIN_PASSWORD** - учетные данные, которые будут использоваться для доступа в панель администратора подсистемы. При первом развертывании пользователь с правами администратора будет создан автоматически. Укажите корректный email и сгенерируйте пароль состоящий из латинских букв и цифр длиной не менее 8-ми символов.
- **PLATFORM_DEFAULT_EMAIL**, **PLATFORM_DEFAULT_PASSWORD** - учетные данные пользователя для доступа в веб-интерфейс подсистемы. При первом развертывании пользователь будет создан автоматически. Укажите корректный email и сгенерируйте пароль состоящий из латинских букв и цифр длиной не менее 8-ми символов.

- **EMAIL_HOST, EMAIL_PORT, EMAIL_HOST_USER, EMAIL_HOST_PASSWORD** - параметры доступа к SMTP-серверу. SMTP-сервер используется для отправки писем, например при сбросе пароля или для оповещений. Чтобы отключить отправку писем, оставьте данные поля пустыми. Для получения параметров доступа к SMTP-серверу обратитесь к администратору вашей сети.
- **EMAIL_USE_SSL** - включение/отключение SSL-протокола, укажите значения true/false, соответственно. Если не используется SMTP-сервер укажите значение false.
- **EMAIL_FROM** - значение, которое будет отправляться в заголовке FROM и отображаться в качестве отправителя письма. Требования к формату данного поля могут зависеть от SMTP-сервера. Пример значения поля FROM - "Bob Example" bob@example.org.
- **QUERY_LIMIT** - ограничение количества возвращаемых элементов в API запросах для получения сущностей системы. Увеличение данного лимита не рекомендуется, т.к. время выполнения API запроса может увеличиться в несколько раз. Также, обратите внимание, что увеличение лимитов приведет к ухудшению работы системы.
- **ENABLE_PROFILE_AUTOGENERATION** - авто-создание профилей для приходящих активностей с агента. Необходимо учитывать, что при включении данной опции будет увеличенный расход ресурсов лицензии (размер базы данных). Если функция не требуется, то нужно оставить поле пустым, в противном случае установить значение 1.
- **USE_CUDA** - отвечает за использование CUDA ядер в сервисах обработки изображений. 0 - отключить GPU, 1 - включить GPU для сервиса **template_extractor**.

Сохраните изменения в файле.

Расширенная конфигурация системы

В комплекте поставки находится конфигурационный файл **./deploy/values.yaml**. В разделе **env** присутствует возможность изменить конфигурацию элементов системы.

Значения для каждого из параметров заданы по умолчанию. Изменять значения параметров рекомендуется только в случае явной необходимости.

Параметры конфигурационного файла values.yaml

- **ACTIVITY_TTL**: Время хранения активностей в базе данных. Значение задается в секундах. Например, 2592000 секунд (30 дней)
- **SAMPLE_TTL**: Время хранения сэмплов в базе данных. Значение задается в секундах. Например, 2592000 секунд (30 дней)

Установка и настройка кластера

3DiVi BAF Инструкция по установке

Запустите команду для создания и настройки кластера.

```
$ ./setup/init-cluster.sh
```

Эта команда выполняет следующие действия:

- Инициализация узла для развертывания кластера
- Создание секретов
- Создание необходимых папок
- Установка ingress-controller
- Установка nvidia-device-plugin, если включено использование видеокарты

Проверка работоспособности кластера

После инициализации главного узла убедитесь, что все узлы готовы к работе и имеют статус Ready. Для проверки выполните следующую команду:

```
$ kubectl get nodes
```

В результате в терминале будет отображен следующий вывод:

NAME	STATUS	ROLES	AGE	VERSION
master-node	Ready	control-plane,master	11d	v1.23.8

Для проверки всех элементов кластера запустите следующую команду:

```
$ kubectl get all --all-namespaces
```

Использование локальной СУБД

По умолчанию используется локальная СУБД.

Для использования локальной СУБД, убедитесь, что на сервере, на котором будет развернута база данных, создана директория **/kv/pgdata**, в противном случае, создайте ее, выполнив следующие команды:

```
$ sudo mkdir -p /kv/pgdata  
$ sudo chmod -R 777 /kv/pgdata
```

Настройка лицензирования

Установка сервера лицензий

Перед установкой откройте файл **./setup/settings.env** и задайте IP адрес машины, на которой будет установлен сервер лицензий, в переменной **LIC_SERVER_IP_ADDRESS**.

Если сервер лицензий запускается на той же машине, где развернут кластер, то **LIC_SERVER_IP_ADDRESS** будет совпадать с **MASTER_NODE_IP_ADDRESS**.

Запустите команду для установки и запуска сервера лицензий:

```
$ ./setup/install-lic-server.sh
```

Проверьте, что сервер лицензий находится в статусе `running`, выполнив следующую команду:

```
$ ./setup/status-license-server.sh
```

Пример вывода в консоль:

```
floatingserver.service - Floating License Server
   Loaded: loaded (/etc/systemd/system/floatingserver.service; enabled; vendor preset: enabled)
   Active: active (running) since Tue 2022-12-20 12:25:54 +05; 1min 48s ago
```

Убедитесь, что сервер лицензий доступен. Для этого перейдите по адресу **http://<LIC_SERVER_IP_ADDRESS>:8090** в вашем браузере, на странице должна отобразиться форма входа.

Офлайн активация лицензии

Перед активацией убедитесь, что в переменной **LIC_KEY** (в файле `./setup/settings.env`) записан лицензионный ключ.

Запустите команду для генерации офлайн запроса на лицензию:

```
$ ./setup/activate-lic-server.sh --generate-offline
```

В результате выполнения команды в директории **on_premise** должен появиться файл **request-offline.license**.

Отправьте сгенерированный файл запроса **request-offline.license** на почту **support-platform@3divi.com**. В ответном письме будет отправлен файл лицензии.

Поместите полученный лицензионный файл в папку **on_premise**.

Откройте файл конфигурации `./setup/settings.env`, используя текстовый редактор, и заполните значение переменной **OFFLINE_LICENSE_FILE** именем файла лицензии и его расширением, если присутствует, через точку.

3DiVi BAF Инструкция по установке

Запустите команду для активации полученной лицензии:

```
$ ./setup/activate-lic-server.sh --activate-offline
```

Пример вывода в консоль при успешной активации лицензии:

```
[2022-09-08 01:30:36+05:00] INF Offline activating license key...  
[2022-09-08 01:30:36+05:00] INF License activated successfully!
```

Развертывание подсистемы распознавания лиц

Запуск развертывания

Запустите скрипт для развертывания подсистемы в кластере:

```
$ ./setup/deploy.sh
```

Для отслеживания процесса развертывания откройте ещё одну вкладку терминала и введите следующую команду:

```
$ watch 'kubectl get pods'
```

Наличие у всех pods статуса *Running* означает, что подсистема запущена.

Настройка DNS

Для обеспечения доступа к подсистеме DNS сервер вашей сети должен содержать запись о том, что домен **platform.<DOMAIN>** доступен по адресу **<MASTER_NODE_IP_ADDRESS>**. Значения переменных можно получить из файла **./setup/settings.env**. Обратитесь к администратору вашей сети, чтобы выполнить данную конфигурацию.

Для целей тестирования можно указать IP-адрес и домен в файле **/etc/hosts** на Linux.

Для этого добавьте в конец данного файла новую строку вида **<MASTER_NODE_IP_ADDRESS> platform.<DOMAIN>**, подставив значения соответствующих переменных, и сохраните файл. Обратите внимание, что для редактирования файла **hosts** необходимо обладать правами администратора.

Для использования подсистемы с той же машины, где выполнено развертывание, можно воспользоваться скриптом. Он автоматически добавит необходимую запись в файл **/etc/hosts**.

3DiVi BAF Инструкция по установке

```
$ ./setup/add-dns.sh
```

Описание элементов развернутой системы

Для получения статуса сервисов подсистемы выполните команду:

```
$ kubectl get pods
```

В консоль будет выведен список сервисов, их статус, количество перезапусков и время с момента создания сервиса. Пример вывода:

NAME	READY	STATUS	RESTARTS	AGE
activity-matcher-dep-6fdc8bfb5-pl8ql	1/1	Running	2 (24h ago)	24h
age-estimator-dep-544cdf7c-4khhh	1/1	Running	0	24h
agent-sync-dep-854474ddc9-ntd2w	1/1	Running	2 (24h ago)	24h
backend-dep-74c5d86d77-wldcn	1/1	Running	2 (24h ago)	24h
body-detector-dep-788dc59547-5jqtf	1/1	Running	0	24h
broker-dep-f6dfdf55b-kl76k	1/1	Running	0	24h
cache-dep-7dbc644bcf-6qpzb	db- 1/1	Running	0	24h
dep-cf96d8d4c-btmbf	1/1	Running	0	24h
emotion-estimator-dep-764c8d8669-wzssn	1/1	Running	0	24h
face-detector-face-fitter-dep-8585d54d67-j	1/1	Running	0	24h
face-detector-liveness-estimator-dep-66c8789ddb-x4h95	2/2	Running	0	24h
face-detector-template-extractor-dep-6b844fd9-tjprf	1/1	Running	0	24h
gateway-dep-67c7d6f4c7-5lpsb	gender- 1/1	Running	0	24h
estimator-dep-7b7d859c6f-n9f76	image- 1/1	Running	0	24h
api-dep-6dc7f868f6-gz56v	licensing- 1/1	Running	0	24h
dep-967cc7b65-wg6jq	mask- 1/1	Running	0	24h
estimator-dep-7db6779bc5-nnwt5	matcher-dep- 1/1	Running	0	24h
696d66b65b-fqn9z	processing- 1/1	Running	2 (24h ago)	24h
dep-f7d7867f6-25tjl	quality- 1/1	Running	0	24h
assessment-estimator-dep-76fcdf6cf-zjldq	quality- 1/1	Running	0	24h
dep-86cc5488d9-22tkn	redis- 1/1	Running	0	24h
dep-5d8cd4d657-7vw8c	verify- 1/1	Running	0	24h
matcher-dep-85ddfd4f-7t7br	1/1	Running	0	24h

Ниже приведено краткое описание сервисов:

- activity-matcher-dep - сервис используется для поиска людей по активностям;
- age-estimator-dep - сервис используется для оценки возраста человека по изображению лица;
- agent-sync-dep - сервис отвечает за синхронизацию данных о профилях с агентами;
- backend-dep - основной контейнер подсистемы, отвечает за работу большей части API;
- body-detector-dep - сервис предназначен для детекции силуэтов на изображении;

- broker-dep - сервис RabbitMQ, используется для работы асинхронной очереди задач;
- cache-dep - сервис Memcached, используется для кэширования данных;
- db-dep - экземпляр СУБД PostgreSQL, хранит всю информацию подсистемы;
- emotion-estimator-dep - сервис используется для оценки эмоций человека по изображению лица;
- face-detector-face-fitter-dep - сервис используется для детекции лиц и определения антропометрических точек лица и углов наклона/поворота головы;
- face-detector-liveness-estimator-dep - сервис используется для детекции лиц и оценки принадлежности лица на изображении реальному человеку;
- face-detector-template-extractor-dep - сервис предназначен для детекции лиц и извлечения биометрического шаблона лица;
- gateway-dep - сервис nginx, отвечает за доступ к подсистеме и за работу веб-интерфейса подсистемы;
- gender-estimator-dep - сервис используется для оценки пола человека по изображению лица;
- image-api-dep - сервис ImageAPI, доступный по URL /image-api/ (является устаревшим, используется для обратной совместимости);
- licensing-dep - сервис, ограничивающий работу подсистемы согласно параметрам лицензии;
- mask-estimator-dep - сервис позволяет определить наличие/отсутствие медицинской маски на лице человека;
- matcher-dep - сервис используется для поиска людей по профилям;
- processing-dep - сервис аккумулирует результаты работы сервисов-обработчиков (age-estimator-dep, emotion-estimator-dep, gender-estimator-dep, face-detector-face-fitter-dep, mask-estimator-dep);
- face-detector-liveness-estimator-dep);
- quality-assessment-estimator-dep - сервис предназначен для оценки качества изображения лица;
- quality-dep - сервис отвечает за расчет качества изображения (является устаревшим, используется для обратной совместимости);
- redis-dep - сервис Redis, используется для работы WebSockets;
- verify-matcher-dep - сервис отвечает за сравнение двух лиц с изображений.

Удаление 3DiVi BAF

Для удаления системы выполните следующую команду:

```
$ ./cli/deployment/delete.sh
```