Установка компонентов Системы бизнес-анализа BI

1. Установка операционной системы

2. Установка СУБД PostgresSQL

3. Установка Apache Airflow

4. Установка Apache Superset

В состав Системы входят модули, приведенные в Таблице 3.

Таблица 3. Модули системы

№ п/п	Подсистема	Описание	Предназначение
1.	Подсистема интеграции	Предоставляет собой встроенные средства ППО Apache Airflow	• интеграция со смежными подсистемами
2.	Подсистема администрирования	Реализована средствами Active Directory Domain Services, Apache Airflow, Apache Superset	• ведения списка пользователей, управления правами доступа
3.	Подсистема подготовки подмножества данных (ETL)	Предоставляет собой сервер тестирования процессов извлечения, преобразования и загрузки данных. - Ubuntu - Apache Airflow	• автоматизация процессов загрузки и трансформации данных в модуль хранения данных; • идентификация и приведение к единому формату текстовых и числовых данных систем-источников; • расчет и передачу показателей в модуль хранения данных в разрезах; • предоставление данных во внешние системы; • формирование отчетов в соответствии с требованиями.
4.	Подсистема представления автоматизированной отчетности	Представляет собой сервер с установленными: - Ubuntu - Apache Superset	• представление автоматизированной отчетности с возможностью формирования, настройки и построения графиков и диаграмм в веб-интерфейсе для бизнес-пользователей
5.	Подсистема единого хранилища данных.	Представляет сервер с установленными: - Ubuntu - PostgreSQL	• хранение информации • сжатие сегментов • ускорение работы SQL-запросов • разграничение прав доступа к данным • объединение серверов в кластер • хранение исторической информации с минимальной нагрузкой на базу данных

Система бизнес-анализа BI развернуто на базе облачной инфраструктуры SberCloud. Все компоненты разворачиваются внутри облака SberCloud. Для начала работ по установке, необходимо получить доступ к Учетной записи компании в облаке.

1. Средствами инструментария предоставляемого SberCloud создается виртуальная машина (ВМ) у предустановленной операционной системой Ubuntu Server 14 (Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-99-generic x86_64) рекомендуемые параметры сервера 16Gb RAM, 500Gb, 4 core CPU (2.2 GHz). Архитектура облачного сервиса, предполагает изменение параметров сервера без изменения компонентов ЕХД, в зависимости от возникших требований к производительности системы.

2. На ранее созданной ВМ, в облачном сервисе, создается экземпляр СУБД PostgresSQL 14. При установке параметров экземпляра, создаем новую БД RA_DHW_DEV. Суперпользователя оставляем postgres/postgres.

После окончания установки — Необходимо заменить пароль по умолчанию на безопастный.

Далее, необходимо развернуть из дампа структуру БД ЕХД, для этого необходимо зайти ssh клиентом на сервер БД, в домашнем каталоге создать папку для размещения дампа, например /pg_dupm

Затем клонируем данные дампа с репозитория ssh:

//git@172.20.207.11:2224/uitsber/sc_dwh_db.git

после получения кода дампа, штатными средствами СУБД PosetgreSQL — создаем структуру БД.

pg_dump RA_DWH_DEV > db.sql

За более подробной информацией по настройке СУБД PostgreSQL рекомендуется обратиться к официальной документации https://www.postgresql.org/docs/

База данных из дампа создается без данных!

3. Для сервера ETL создается новая ВМ, аналогичная по параметрам из п.1

Компоненты ETL создаются с использованием технологии контейнеризации. Контейнеризация реализуется с помощью ПО Docker

переходим по ssh протоколу на вновь созданный сервер ETL, производим обновление пакетов ОС.

sudo apt update

sudo apt full-upgrade

затем необходимо установить ПО docker и плагин к нему docker-compose.

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin

проверку корректности установки ПО docker можно сделать, выполнив команду:

sudo docker run hello-world

Для более тонкой настройки ПО docker рекомендуется обратиться к официальной документации https://docs.docker.com

4. Установка Apache Airflow

Создаём на сервере ETL пользователя airflow. Добавляем его в группу docker

sudo usermod -aG docker airflow

1) *Создаем каталог для размещения компонентов airflow(/opt/airflow)*

2) *Копируем с git репозитория файл с конфигурацией для контейнеров входящих в состав airflow(docker-compose.yml)*

При необходимости, необходимо внести изменения в параметры docker-compose.

3) *Клонируем текущую версия DAG файлов и вспомогательных файлов из репозитория*

ssh://git@172.20.207.11:2224/uitsber/sc_dwh.git

Дополнительно, к стандартной поставке airflow, необходимо добавить следующие библиотеки:

openpyxl, xlrd, dbt-postgres

Для этого необходимо создать файл Dockerfile

Внести в него нижеследующие строки

FROM apache/airflow:2.4.0

RUN pip3 install --user --upgrade pip

RUN pip3 install --upgrade pandas

RUN pip3 install openpyxl

RUN pip3 install xlrd

RUN pip3 install dbt-postgres

и запустить пересборку docker образа:

docker build .

Заупскаем контейнеры с компонетами airflow:

docker compose up

Корректность работы сервиса ETL можно проверить, перейдя по ссылке:

http://[ELT_SERVER_IP]:8080

В случае успешной установки экземпляра airflow, будет открыта страница с авторизацией airflow.

Контактное лицо для связи:
Морозов Максим Владимирович (Руководитель проекта)
Телефон: 89195479659.

Реализация процедур ETL реализована с использованием ПО Apache Airflow v.2.4.

Пример развернутого ПО:

Apache Airflow развернут на машине *172.20.200.41* вход на сервер осуществляется через TTY консоль, SSH протокол. Пользователь airflow.
Для работы с контейнерами Airflow необходимо перейти в папку установка ПО
cd /opt/airflow

```
airflow@ecs-dwh-vm1:/opt/airflow$ ls -l
total 100
-rw-r--r--  1 root root 49634 Oct  3 21:56 airflow.cfg
drwxrwxr-x  7 root root  4096 Jan 22 12:21 dags
-rw-rw-r--  1 root root 10739 Jan 18 23:41 docker-compose.yaml
-rw-r--r--  1 root root 10732 Jan 11 10:07 docker-compose.yaml.bak
-rw-r--r--  1 root root   150 Jan 18 23:39 Dockerfile
drwxr-xr-x  2 root root  4096 Oct 27 22:49 libs
drwxrwxr-x 20 root root  4096 Jan 22 12:38 logs
drwxrwxr-x  2 root root  4096 Sep 25 23:07 plugins
drwxr-xr-x  2 root root  4096 Jan 30 03:00 tmp_files
airflow@ecs-dwh-vm1:/opt/airflow$
```

Проверка статуса сервисов.

docker container ls

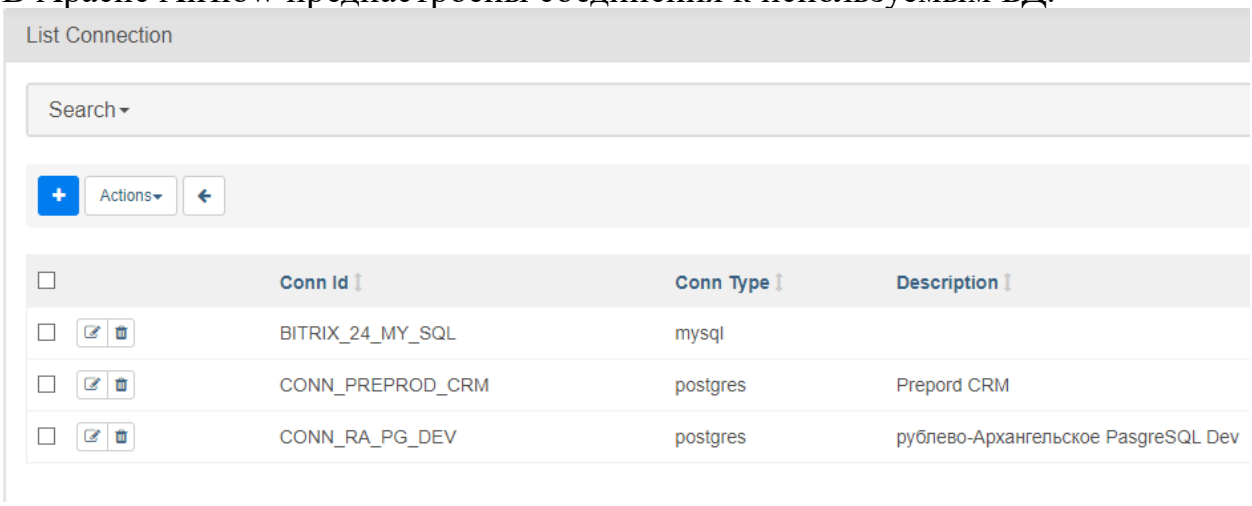
```
airflow@ecs-dwh-vm1:/opt/airflow$ docker container ls
CONTAINER ID   IMAGE                      COMMAND                  CREATED      STATUS                    PORTS                                       NAMES
e672e2e29299   apache/ra_airflow:1.0.1    "/usr/bin/dumb-init …"   6 days ago   Up 6 days (unhealthy)     8080/tcp                                    airflow-airflow-triggerer-1
0b7c1f072ef8   apache/ra_airflow:1.0.1    "/usr/bin/dumb-init …"   6 days ago   Up 6 days (healthy)       0.0.0.0:8080->8080/tcp, :::8080->8080/tcp   airflow-airflow-webserver-1
89b662567f06   apache/ra_airflow:1.0.1    "/usr/bin/dumb-init …"   6 days ago   Up 6 days (unhealthy)     8080/tcp                                    airflow-airflow-worker-1
e8fb68ad3f6a   apache/ra_airflow:1.0.1    "/usr/bin/dumb-init …"   6 days ago   Up 6 days (unhealthy)     8080/tcp                                    airflow-airflow-scheduler-1
116c742bf287   postgres:13                "docker-entrypoint.s…"   6 days ago   Up 6 days (healthy)       5432/tcp                                    airflow-postgres-1
1e73124342f5   redis:latest               "docker-entrypoint.s…"   6 days ago   Up 6 days (healthy)       6379/tcp                                    airflow-redis-1
```

Запуск Apache Airflow

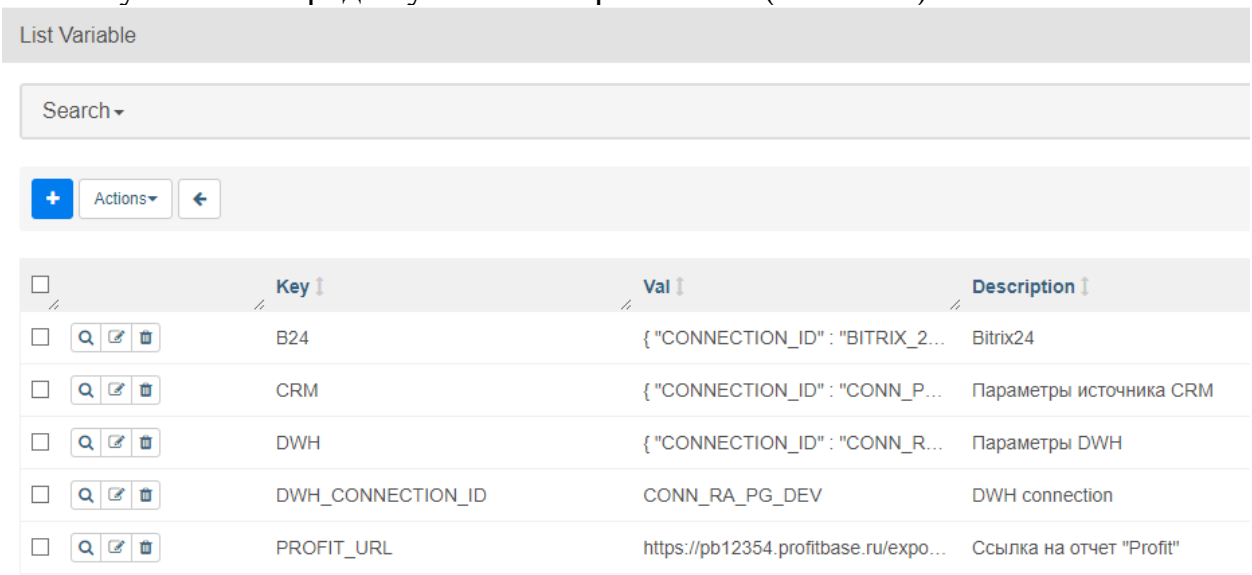
Docker compose up

Остановка Apache Airflow

Docker compose down
WEB интерфейс Apache Airflow доступен через браузер, по адресу http://172.20.200.41:8080/home
В Apache Airflow преднастроены соединения к используемым БД.



Так же установлен ряд служебных переменных (Variables).



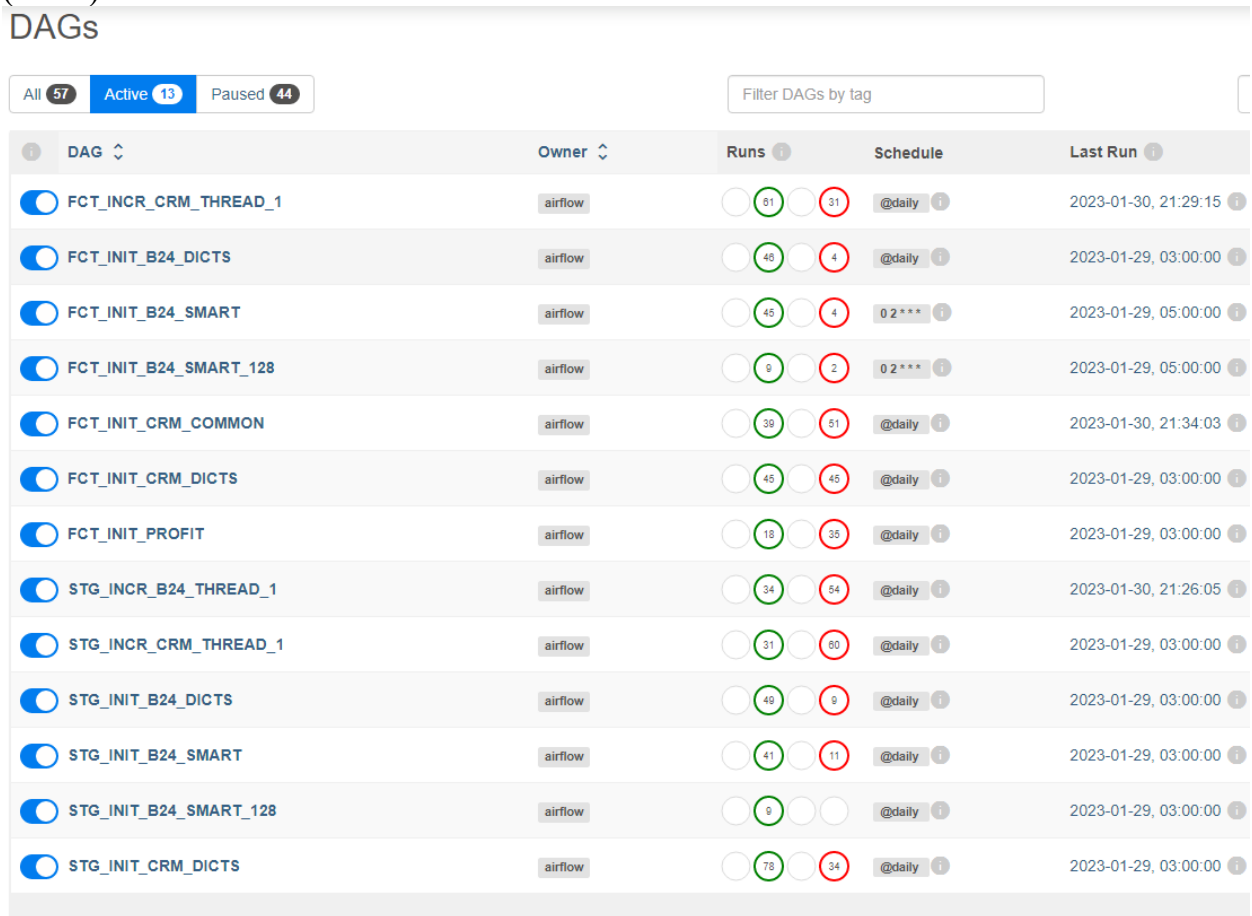
Для загрузки данных написан ряд Airflow DAGs, код DAGs расположен в /opt/airflow/dags.

Переиспользуемый в DAGs код вынесен в отдельные python модули /opt/airflow/dags/lib.
Конфигурационные параметры для загрузки данных расположены в yaml файлах.
/opt/airflow/dags/config
Конфигурация содержит данные об источнике, целевой таблице, названия целевой схемы, [перечень целевых столбцов], флаг необходимости очистки таблицы перед вставкой, словарь для подстановок source-target (преобразования типов, применение функций и т.д.).

Пример конфигурации для ETL

```
b24_deal:
  source: DWH
  src_table: stage.b24_deal
  dest_schema: facts
  dest_fields:
  dest_truncate_yn: n
  substs:
    ddu_payment_method: ddu_payment_method::int
```

Процессы загрузки запускаются автоматически по расписанию начиная с 0:00 (UTC)



Результаты работы DAGs доступны через веб-интерфейс. Доступны как общая информация о статусе загрузки, так и детальная информация (logs)

Установка Apache Superset

Средствами инструментария предоставляемого SberCloud создается виртуальная машина (ВМ) у предустановленной операционной системой Ubuntu Server 14 (Ubuntu 20.04.5 LTS (GNU/Linux 5.4.0-99-generic x86_64) рекомендуемые параметры сервера 16Gb RAM, 500Gb, 4 core CPU (2.2 GHz).

Скачиваем образ docker-контейнера с Superset:

$ sudo docker pull apache/superset

Проверяем корректность установки:

$ sudo docker images

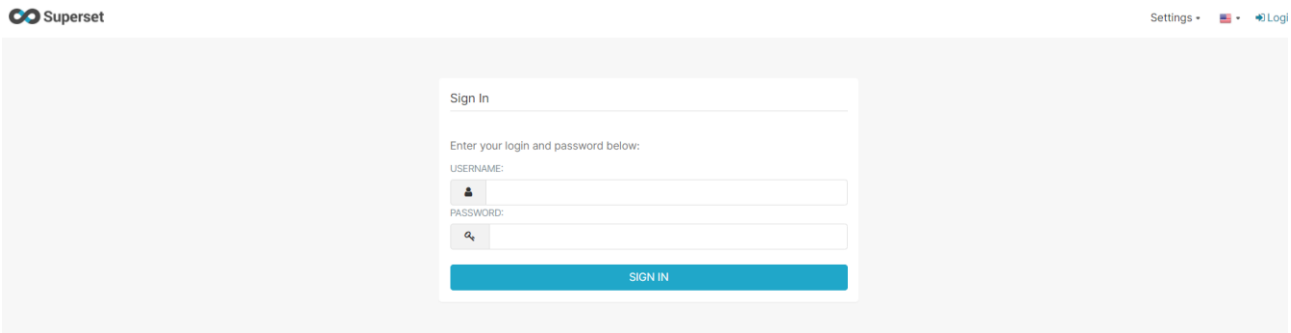
Запускаем docker-контейнер с Superset:

$ sudo docker run -d -p 8080:8088 --name superset apache/superset;

Устанавливаем утилиту для мониторинга сети:

$ sudo apt-get install net-tools

Если порт 8080 занят процессом docker-proxy и прослушивается, а также присутствует активный контейнер с названием apache/superset, то вводим IP адрес своей машины и открываем в браузере наш порт с Superset.



Далее создаем и настраиваем пользователей:

$ sudo docker exec -it superset superset fab create-admin \

--username admin \

--firstname Superset \

--lastname Admin \

--email admin@admin.com \

--password postgres123; \

sudo docker exec -it superset superset db upgrade; \

sudo docker exec -it superset superset load_examples; \

sudo docker exec -it superset superset init.

Заходим в Apache Superset и подключаемся к БД ЕХД через Postgres.