Инструкция по внедрению платформы в контур клиента

Данный файл является инструкцией по полному внедрению BI-системы Fastboard на ресурсе клиента. Для проверки системы без разворачивания контейнеров на мощностях конечного пользователя предоставлен файл “Инструкция по запуску платформы”

Перед внедрением необходимо запросить у сотрудника технической поддержки предоставляемый дистрибутив: докеры, docker-compose.yml, fastboard-back.tar и fastboard-front.tar

Подготовка к развертыванию

Сохраните все файлы, включая docker-compose.yml, fastboard-back.tar, fastboard-front.tar, в одной папке.

Откройте консоль, перейдите в папку с файлами:

```
cd {Расположение папки}
```

1. Для разворачивания контейнеров создаем директории

```
mkdir resources - директория для баз данных
```

2. Переходим в директорию баз данных и создаем директорию для postgres

```
cd resources/
mkdir postgres
```

3. Переходим в директорию postgres, копируем в нее файл docker-compose.yml из предоставленного дистрибутива. Создаем сеть для контейнеров с базами данных и запускаем контейнер с postgres

```
cd postgres/
docker network create resources
docker-compose up -d
```

После запуска, скачается необходимый образ и запустится контейнер postgres. Логин по умолчанию postgres пароль postgres, изменить можно в файле docker-compose.yml перед запуском контейнера

4. Создаем базу данных postgres, например с именем fastboard_back, создать можно разными способами, например подключиться к контейнеру при помощи PgAdmin или psql-16

```
sudo sh -c 'echo "deb http://apt.postgresql.org/pub/repos/apt $(lsb_release -cs)-pgdg main" > /etc/apt/sources.list.d/pgdg.list'
curl -fsSL https://www.postgresql.org/media/keys/ACCC4CF8.asc | sudo gpg --dearmor -o /etc/apt/trusted.gpg.d/postgresql.gpg
apt update
apt install postgresql-client-16
psql -h 127.0.0.1 -U postgres
                \l
                CREATE DATABASE fastboard_back;
                \l
                \q
```

5. Переходим на директорию выше, т.е. в директорию resources

```
cd ..
```

Создаем директории для Clickhouse переходим в нее

```
mkdir clickhouse
cd  clickhouse/
```

6. Копируем в директорию clickhouse файл docker-compose.yml и директории с конфигами (etc, custom_config) из предоставленного дистрибутива Clickhouse. Меняем параметры выделенной памяти для контейнера в файле docker-compose.yml (по умолчанию выставлено от 2 до 8 Гб ОЗУ) и запускаем контейнер с clickhouse

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

После запуска, скачается необходимый образ и запустится контейнер с clickhouse. Логин по умолчанию admin пароль Passw0rd, изменить можно в файле etc/users.xml перед запуском контейнера

7. Переходим на директорию выше, т.е. в директорию resources

```
cd ..
```

8. Создаем директории для Redis и переходим в нее

```
mkdir redis
cd  redis/
```

9. Создаем директории для баз и логов redis

```
mkdir -p data/{bases,log}
```

10. Копируем в директорию redis файл docker-compose.yml и директорию с конфигами etc из предоставленного дистрибутива Redis. Запускаем контейнер с Redis

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

После запуска, скачается необходимый образ и запустится контейнер с redis. Пароль по умолчанию Passw0rd, изменить можно в файле etc/redis.conf перед запуском контейнера

11. Создаем директории для RabbitMQ и переходим в нее

```
mkdir rabbitmq
cd rabbitmq/
```

Копируем в директорию RabbitMQ файл docker-compose.yml из предоставленного дистрибутива RabbitMQ. Запускаем контейнер с RabbitMQ

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

После запуска, скачается необходимый образ и запустится контейнер с RabbitMQ. Логин и пароль по умолчанию fb_rabbit, изменить можно в файле docker-compose.yml перед запуском контейнера

Теперь нужно создать VHOST для контейнера бэк:

```
docker exec -it rabbitmq_1 bash
rabbitmqctl add_vhost fb1
rabbitmqctl list_vhosts
rabbitmqctl set_permissions -p "fb1" "fb_rabbit" ".*" ".*" ".*"
```

12. Создаем директорию для FastBoard, переходим в нее, копируем docker-compose.yml, license_rsa.pub, директорию static-files и образы docker из архива fastboard.tar.gz в рабочую директорию проекта

```
cd ..
mkdir fastboard
cd fastboard
```

Проверяем и устанавливаем данные для авторизации в базах данных и параметры подключения к контейнерам с базами в файле docker-compose.yml, в данный момент там установлены данные для подключения по умолчанию.

Далее импортируем образы контейнеров fastboard:back и fastboard:front, для этого выполняем команды:

```
docker load -i fastboard-back.tar
docker load -i fastboard-front.tar
```

После загрузки образов, запускаем контейнеры бэк и фронт, а также проверяем логи их запуска

```
docker-compose up -d
docker-compose logs -f
```

или

```
docker-compose up -d
docker-compose ps
docker logs -f container_name
```

В момент запуска конейнера с бэкенд приложения, пройдут миграции и создастся база данных с логином и паролем по умолчанию. Логин admin пароль YYYY-MM-DD(дата первого запуска контейнеров)

Если контейнеры запустились без проблем, то проверяем работу пройдя по ссылке для контейнера фронт: http://front.example.ru и введя логин и пароль по умолчанию (admin, YYYY-MM-DD).