

Инструкция по установке экземпляра FeedBackTalk серверное решение

Оглавление

| | |
|---|----|
| Введение | 2 |
| Требования к системе | 2 |
| Установка стороннего программного обеспечения | 3 |
| Установка Docker | 3 |
| Установка Git | 3 |
| Установка PostgreSQL | 4 |
| Установка ClickHouse | 5 |
| Установка Minio | 6 |
| Установка Apache Kafka | 7 |
| Установка ПО FeedBackTalk | 9 |
| Установка Backend (private API) | 9 |
| Установка Public Services (public API) | 9 |
| Установка Frontend (портал) | 11 |
| Публикация сервисов | 11 |
| Пример конфигурации Nginx | 12 |
| Результат установки и доступ | 16 |
| Сводная таблица по развернутым сервисам | 17 |
| Контактная информация | 17 |

Введение

Данное руководство рассматривает вариант установки серверного решения FeedBackTalk. Серверное решение FeedBackTalk может быть развернуто как на физических серверах, так и полностью поддерживается работа в виртуальном окружении.

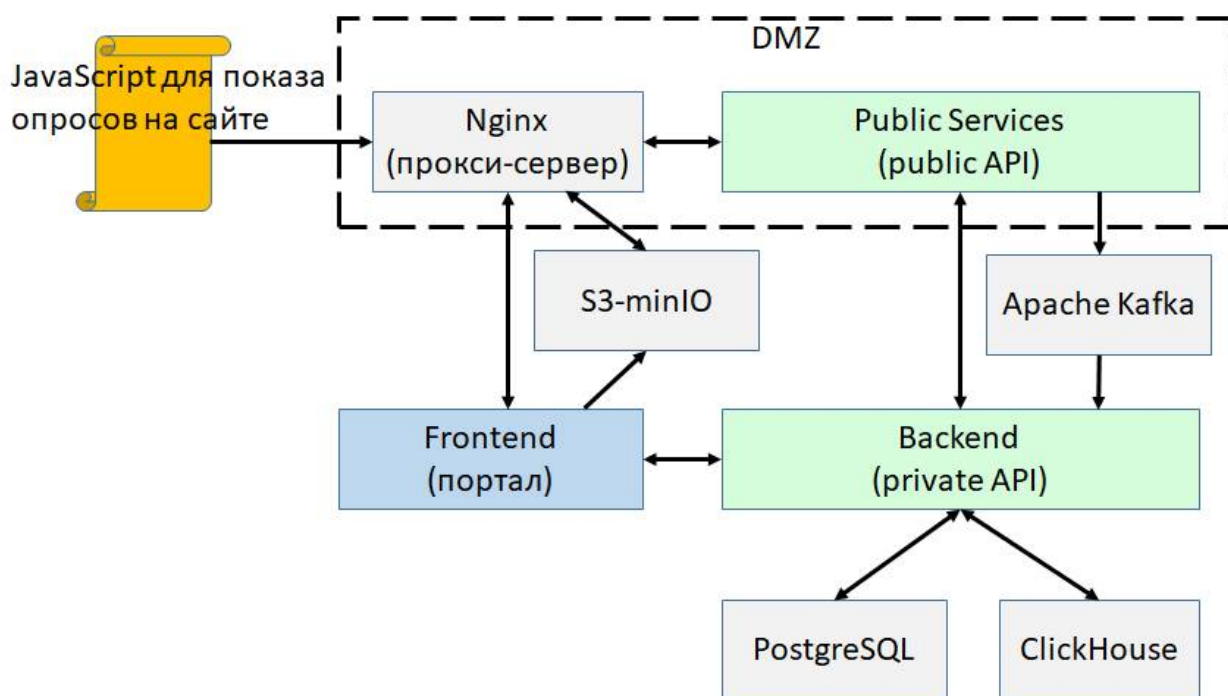
Подробную информацию вы можете получить по электронной почте support@tech-intel.ru и по телефону +7(925) 788-68-44.

Требования к системе

С технической точки зрения все компоненты программного продукта FeedBackTalk могут быть развёрнуты в контейнерах на одной виртуальной машине. С точки зрения безопасности мы рекомендуем, как минимум, разделять по отдельным виртуальным машинам публичные и внутренние ресурсы. При наличии высоких требований по производительности системы следующие компоненты должны быть установлены на отдельные виртуальные машины:

- Public Services (public API)
- Backend (private API)
- Frontend (портал)
- PostgreSQL (база данных)
- Clickhouse (база данных)
- Apache Kafka (брокер сообщений)
- S3-minIO (файловый сервер)

Схема взаимодействия компонент для настройки сетевого доступа показана на рисунке ниже:



Минимальные системные требования к виртуальным машинам указаны в таблице ниже:

| Назначение виртуальной машины | Процессор, ядро (vCPU) | ОЗУ, ГБ (vRAM) | Объем SSD диска, ГБ |
|-------------------------------|------------------------|----------------|---------------------|
| Public Services (public API) | 2 | 4 | 20 |
| Backend (private API) | 2 | 8 | 20 |

| | | | |
|---------------------------------|---|---|-----|
| Frontend (портал) | 2 | 8 | 20 |
| PostgreSQL (база данных) | 4 | 8 | 100 |
| ClickHouse (база данных) | 2 | 8 | 50 |
| Apache Kafka (брокер сообщений) | 2 | 4 | 50 |
| S3-minIO | 1 | 2 | 20 |

На виртуальных машинах необходимо установить одну из перечисленных ниже операционных систем: Astra Linux актуальной версии, Ubuntu 20.04 LTS и выше, Debian 11 и выше.

Оборудование пользователя должно соответствовать рекомендуемым требованиям для функционирования браузера, через который пользователь использует программное обеспечение. Для использования программного обеспечения производитель рекомендует пользователю использовать следующие браузеры:

- Яндекс.Браузер 18.2 и выше;
- Google Chrome 87.0 и выше;
- Mozilla Firefox 84.0 и выше;
- Safari 14.0 и выше;
- Opera 72.0 и выше.

Установка стороннего программного обеспечения

Установите на виртуальные машины: Git, SSH для подключения к серверу, Docker Engine + Docker Compose для запуска контейнеров.

Получите логин и пароль от git-репозитория в службе технической поддержки ООО «ТехноИнтеллект» (раздел «Контактная информация»).

Установка Docker

Для Astra Linux

Обратитесь к руководству «Установка и администрирование Docker в Astra Linux» по ссылке <https://wiki.astralinux.ru/pages/viewpage.action?pageId=158601444>

Для Debian и Ubuntu

Обратитесь к руководству по установке Docker для вашей операционной системы по ссылке <https://docs.docker.com/engine/install/>

Проверить что docker установлен в системе можно следующими командами

```
docker --version
docker-compose --version
```

Установка Git

Если git не установлен в вашей ОС, то установите его

```
apt install git -y
```

Проверить что git установлен в системе можно следующими командами

```
git --version
```

Установка PostgreSQL

Перед началом работ убедитесь, что виртуальная машина соответствует общим требованиям.

1. Подключитесь к серверу, на котором будет развернута PostgreSQL по SSH.
2. Перейдите в режим суперпользователя (переключиться в root)

```
sudo su -
```

3. Далее создайте каталог, в котором будут развернуты PostgreSQL сервисы, с помощью docker.

```
mkdir -p /docker-app
```

4. Перейдите в созданный каталог

```
cd /docker-app
```

5. Далее скачайте на сервер файлы (конфиги, docker-compose, db-файлы), необходимые для работы предварительно настроенной базы данных PostgreSQL, из git-репозитория.

```
git clone https://techintel.gitlab.yandexcloud.net/fbt/distribution.git
```

Пример результата выполнения команды git clone

```
Cloning into 'distribution'...
```

```
Username for 'https://techintel.gitlab.yandexcloud.net': login
```

```
Password for 'https://login@techintel.gitlab.yandexcloud.net': password
```

```
Unpacking objects: 100%.
```

6. Распакуйте tgz-архив скачанный с помощью команды git clone

```
cd /docker-app/fbt-postgres
```

```
tar -xvzf fbt-postgres.tgz
```

7. Пример структуры каталога /docker-app/fbt-postgres после выполнения команды git clone и распаковки tgz-архива

```
.
├── data
│   ├── base
│   ├── global
│   ├── pg_commit_ts
│   ├── pg_dynshmem
│   ├── pg_hba.conf
│   ├── pg_ident.conf
│   ├── pg_logical
│   ├── pg_multixact
│   ├── pg_notify
│   ├── pg_replslot
│   ├── pg_serial
│   ├── pg_snapshots
│   ├── pg_stat
│   ├── pg_stat_tmp
│   ├── pg_subtrans
│   ├── pg_tblspc
│   ├── pg_twophase
│   ├── PG_VERSION
│   └── pg_wal
```

```
├── pg_hact
├── postgresql.auto.conf
├── postgresql.conf
├── postmaster.opts
└── postmaster.pid
```

8. Авторизуйтесь в docker registry

```
docker login https://docker.gqdev.ru
```

Результат:

Username: login

Password: password

...

Login Succeeded

9. Отредактируйте раздел extra_hosts в docker-compose файле, который отвечает за сопоставления имен хостов и IP-адресов для контейнера PostgreSQL.

```
nano /docker-app/fbt-postgres/docker-compose.yml
```

10. Запустите контейнер с PostgreSQL

```
cd /docker-app/fbt-postgres
```

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

Результат:

Creating fbt-postgres... done

root@fbt-postgres:/docker-app/fbt-postgres#

Посмотреть реквизиты подключения к PostgreSQL вы можете в docker-compose.yml - /docker-app/fbt-postgres/docker-compose.yml

Установка ClickHouse

Перед началом работ убедитесь, что виртуальная машина соответствует общим требованиям.

1. Подключитесь к серверу, на котором будет развернута Clickhouse по SSH.
2. Перейдите в режим суперпользователя (переключиться в root)

```
sudo su -
```

3. Далее создайте каталог, в котором будут развернуты Clickhouse сервисы, с помощью docker.

```
mkdir -p /docker-app
```

4. Перейдите в созданный каталог

```
cd /docker-app
```

5. Далее скачайте на сервер файлы (конфиги, docker-compose, db-файлы), необходимые для работы предварительно настроенной базы данных Clickhouse, из git-репозитория.

```
git clone https://techintel.gitlab.yandexcloud.net/fbt/distribution.git
```

Результат выполнения команды git clone

```
Cloning into 'distribution'...
```

Username for 'https://techintel.gitlab.yandexcloud.net ': login

Password for 'https://login@techintel.gitlab.yandexcloud.net': password

```
Unpacking objects: 100%.
```

6. Пример структуры каталога /docker-app/fbt-clickhouse после выполнения команды git clone и распаковки tgz-архива fbt-clickhouse/fbt-clickhouse.tgz

```

.
├── conf
├── data
│   ├── access
│   ├── data
│   ├── dictionaries_lib
│   ├── flags
│   ├── format_schemas
│   ├── metadata
│   ├── metadata_dropped
│   ├── preprocessed_configs
│   ├── status
│   ├── store
│   ├── tmp
│   ├── user_files
│   ├── user_scripts
│   └── uuid
└── logs
    ├── clickhouse-server.err.log
    ├── clickhouse-server.log
    └── clickhouse-server.log.0.gz

```

7. Авторизуйтесь в docker registry

```
docker login https://docker.gqdev.ru
```

Результат:

Username: login

Password: password

...

Login Succeeded

8. Отредактируйте раздел extra_hosts в docker-compose файле, который отвечает за сопоставления имен хостов и IP-адресов для контейнера Clickhouse.

```
nano /docker-app/fbt-clickhouse/docker-compose.yml
```

9. Запустите контейнер с Clickhouse

```
cd /docker-app/fbt-clickhouse
```

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

Результат:

Creating fbt-clickhouse ... done

Установка Minio

Перед началом работ убедитесь, что виртуальная машина соответствует общим требованиям.

1. Подключитесь к серверу, на котором будет развернута Minio по SSH.
2. Перейдите в режим суперпользователя (переключиться в root)

```
sudo su -
```

3. Далее создайте каталог, в котором будет развернут Minio сервис, с помощью docker.

```
mkdir -p /docker-app
```

4. Перейдите в созданный каталог

```
cd /docker-app
```

5. Далее скачайте на сервер файлы (конфиги, docker-compose), необходимые для работы предварительно настроенного s3 сервера minio, из git-репозитория.

```
git clone https://techintel.gitlab.yandexcloud.net/fbt/distribution.git
```

Результат выполнения команды git clone

```
Cloning into 'distribution'...
```

```
Username for 'https://techintel.gitlab.yandexcloud.net': login
```

```
Password for 'https://login@techintel.gitlab.yandexcloud.net': password
```

```
Unpacking objects: 100%.
```

6. Пример структуры каталога /docker-app/fbt-minio после выполнения команды git clone:

```
.
├── data
│   └── fbt
└── docker-compose.yml
```

7. Авторизуйтесь в docker registry

```
docker login https://docker.gqdev.ru
```

Результат:

```
Username: login
```

```
Password: password
```

```
...
```

```
Login Succeeded
```

8. Отредактируйте раздел extra_hosts в docker-compose файле, который отвечает за сопоставления имен хостов и IP-адресов для контейнера minio.

```
nano /docker-app/fbt-minio/docker-compose.yml
```

9. Запустите контейнер с minio

```
cd /docker-app/fbt-minio
```

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

Результат:

```
Creating fbt-minio... done
```

Установка Apache Kafka

Перед началом работ убедитесь, что виртуальная машина соответствует общим требованиям.

1. Подключитесь к серверу, на котором будет развернута Kafka по SSH.
2. Перейдите в режим суперпользователя (переключиться в root)

```
sudo su -
```

3. Далее создайте каталог, в котором будут развернуты Kafka сервисы, с помощью docker.

```
mkdir -p /docker-app
```

4. Перейдите в созданный каталог

```
cd /docker-app
```

5. Далее скачайте на сервер файлы (конфиги, docker-compose), необходимые для работы предварительно настроенного брокера очередей Kafka, из git-репозитория.

```
git clone https://techintel.gitlab.yandexcloud.net/fbt/distribution.git
```

Пример выполнения команды git clone

```
Cloning into 'distribution'...
Username for 'https://techintel.gitlab.yandexcloud.net': login
Password for 'https://login@techintel.gitlab.yandexcloud.net': password
Unpacking objects: 100%.
```

6. Пример структуры /docker-app/fbt-kafka после выполнения команды git clone

```
├─ config
│   ├── connect-console-sink.properties
│   ├── connect-console-source.properties
│   ├── connect-distributed.properties
│   ├── connect-file-sink.properties
│   ├── connect-file-source.properties
│   ├── connect-log4j.properties
│   ├── connect-mirror-maker.properties
│   ├── connect-standalone.properties
│   ├── consumer.properties
│   ├── log4j.properties
│   ├── producer.properties
│   ├── server.properties
│   ├── tools-log4j.properties
│   ├── trogdor.conf
│   └─ zookeeper.properties
├─ data
│   ├── hspcrdata_root
│   ├── kafka-logs
│   └─ zookeeper
└─ docker-compose.yml
```

7. Авторизуйтесь в docker registry

```
docker login https://docker.gqdev.ru
Результат:
Username: login
Password: password
...
Login Succeeded
```

8. Отредактируйте раздел extra_hosts в docker-compose файле, который отвечает за сопоставления имен хостов и IP-адресов для контейнера Kafka.

```
nano /docker-app/fbt-kafka/docker-compose.yml
```

9. Запустите контейнер с Kafka

```
cd /docker-app/fbt-kafka
docker-compose up -d

Результат:
Creating fbt-kafka ... done
Creating fbt-kafka-ui ... done
```


Установка ПО FeedBackTalk

Установка Backend (private API)

Перед началом работ убедитесь, что виртуальная машина соответствует общим требованиям.

1. Подключитесь к серверу, на котором будет развернут Backend по SSH.
2. Перейдите в режим суперпользователя (переключиться в root)

```
sudo su -
```

3. Далее создайте каталог, в котором будет развернут сервис Backend, с помощью docker.

```
mkdir -p /docker-app
```

4. Перейдите в созданный каталог

```
cd /docker-app
```

5. Далее скачайте на сервер файлы (конфигурационные файлы, docker-compose), необходимые для работы Backend, из git-репозитория.

```
git clone https://techintel.gitlab.yandexcloud.net/fbt/distribution.git
```

Пример выполнения команды git clone

```
Cloning into 'distribution'...
```

```
Username for 'https://techintel.gitlab.yandexcloud.net': login
```

```
Password for 'https://login@techintel.gitlab.yandexcloud.net': password
```

```
Unpacking objects: 100%.
```

6. Пример структуры /docker-app/fbt-backend после выполнения команды git clone

```
├── .env
```

```
└── docker-compose.yml
```

7. Авторизуйтесь в docker registry

```
docker login https://docker.gqdev.ru
```

Результат:

```
Username: login
```

```
Password: password
```

```
...
```

```
Login Succeeded
```

8. Отредактируйте .env файл в котором нужно указать данные для подключения к базам данных.

```
nano /docker-app/fbt-backend/.env
```

9. Запустите контейнер с Backend

```
cd /docker-app/fbt-backend
```

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

Результат:

```
Creating fbt-backend ... done
```

Установка Public Services (public API)

Перед началом работ убедитесь, что виртуальная машина соответствует общим требованиям.

1. Подключитесь к серверу, на котором будет развернут Backend по SSH.
2. Перейдите в режим суперпользователя (переключиться в root)

```
sudo su -
```

3. Далее создайте каталог, в котором будут развернуты Public Services, с помощью docker.

```
mkdir -p /docker-app
```

4. Перейдите в созданный каталог

```
cd /docker-app
```

5. Далее скачайте на сервер файлы (конфигурационные файлы, docker-compose), необходимые для работы Public Services, из git-репозитория.

```
git clone https://techintel.gitlab.yandexcloud.net/fbt/distribution.git
```

Пример выполнения команды git clone

```
Cloning into 'distribution'...
```

```
Username for 'https://techintel.gitlab.yandexcloud.net ': login
```

```
Password for 'https://login@techintel.gitlab.yandexcloud.net': password
```

```
Unpacking objects: 100%.
```

6. Пример структуры /docker-app/fbt-survey-generator/ после выполнения команды git clone

```
├── .env
└── docker-compose.yml
```

7. Авторизуйтесь в docker registry

```
docker login https://docker.gqdev.ru
```

Результат:

```
Username: login
```

```
Password: password
```

```
...
```

```
Login Succeeded
```

8. Отредактируйте .env файл в котором нужно указать данные для подключения к Backend сервисам.

```
nano /docker-app/fbt-survey-generator/.env
```

```
nano /docker-app/fbt-data-import-API/.env
```

```
nano /docker-app/fbt-connect2gounisender/.env
```

9. Запустите контейнер с Public Services

```
cd /docker-app/fbt-survey-generator
```

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

Результат:

```
Creating fbt-survey-generator ... done
```

```
cd /docker-app/fbt-data-import-API
```

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

Результат:

```
Creating fbt-data-import-API ... done
```

```
cd /docker-app/fbt-connect2gounisender
```

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

Результат:

```
Creating fbt-connect2gounisender ... done
```

Установка Frontend (портал)

Перед началом работ убедитесь, что виртуальная машина соответствует общим требованиям.

1. Подключитесь к серверу, на котором будет развернут Frontend по SSH.
2. Перейдите в режим суперпользователя (переключиться в root)

```
sudo su -
```

3. Далее создайте каталог, в котором будет развернут сервис Frontend, с помощью docker.

```
mkdir -p /docker-app
```

4. Перейдите в созданный каталог

```
cd /docker-app
```

5. Далее скачайте на сервер файлы (конфигурационные файлы, docker-compose), необходимые для работы Frontend, из git-репозитория.

```
git clone https://techintel.gitlab.yandexcloud.net/fbt/distribution.git
```

Пример выполнения команды git clone

```
Cloning into 'distribution'...
```

```
Username for 'https://techintel.gitlab.yandexcloud.net': login
```

```
Password for 'https://login@techintel.gitlab.yandexcloud.net': password
```

```
Unpacking objects: 100%.
```

6. Пример структуры /docker-app/fbt-frontend после выполнения команды git clone

```
└─ .env
```

```
└─ docker-compose.yml
```

7. Авторизуйтесь в docker registry

```
docker login https://docker.gqdev.ru
```

Результат:

```
Username: login
```

```
Password: password
```

```
...
```

```
Login Succeeded
```

8. Отредактируйте .env файл в котором нужно указать данные для подключения к Backend.

```
nano /docker-app/fbt-frontend/.env
```

9. Запустите контейнер с Frontend

```
cd /docker-app/fbt-frontend
```

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

Результат:

```
Creating fbt-frontend... done
```

Публикация сервисов

Для публикации FeedBackTalk рекомендуется использовать Nginx. В зависимости от решения, которое используется для публикации сервисов в корпоративной сети, функционал обратного прокси-сервера может быть совмещён со функционалом интернет-шлюза.

Публичными сервисами программного обеспечения FeedBackTalk являются:

- FBT-survey-generator;
- FBT-data-import-API;
- FBT-connect2gounisender.

Рекомендуется использовать https при публикации сервисов.

Приватными сервисами программного обеспечения FeedBackTalk являются:

- FBT-backend;
- FBT-frontend.

Доступ к приватным сервисам ограничен корпоративной сетью заказчика. Виртуальные машины с базами данных и Kafka НЕ должны быть доступны из публичной сети интернет.

Для примера будут использованы следующие значения:

| Внешний URL | Описание | Внутренний IP-адрес:Порт |
|-------------------------|-------------------------|------------------------------|
| api.feedbacktalk.ru/js | FBT-survey-generator | fbt-survey-generator:8001 |
| api.feedbacktalk.ru/di | FBT-data-import-API | fbt-data-import-API:8002 |
| api.feedbacktalk.ru/cu | FBT-connect2gounisender | fbt-connect2gounisender:8003 |
| be.feedbacktalk.ru | FBT-backend | fbt-backend:8000 |
| lk.feedbacktalk.ru | FBT-frontend | fbt-frontend:3001 |
| content.feedbacktalk.ru | FBT-minio | fbt-minio:9000 |

Примеры конфигурационных файлов Nginx приведены ниже. Не забудьте заменить значение для «Внешний URL» и «Внутренний IP» на ваши актуальные адреса.

Пример конфигурации Nginx

```
server {
    listen    443 ssl;
    listen    [::]:443 ssl;
    server_name be.feedbacktalk.ru;

    access_log /var/log/nginx/host.access.log main;
    include /etc/nginx/conf.d/ssl/ssl.conf;
    location / {
        proxy_hide_header Access-Control-Allow-Origin;
        proxy_hide_header Access-Control-Allow-Credentials;
        add_header 'Access-Control-Allow-Origin' "$http_origin" always;
        add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT';

        add_header 'Access-Control-Allow-Headers' 'DNT,Authorization,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,Content-Type,Range';
        if ($request_method = OPTIONS) {
            return 204;
        }
    }
    # X-Real-IP
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-Host $host;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

    proxy_headers_hash_max_size 1024;
    proxy_headers_hash_bucket_size 128;
```

```

    proxy_pass http://fbt-backend:8000/;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
## Timeout
    proxy_connect_timeout 90s;
    proxy_send_timeout 90s;
    proxy_read_timeout 90s;
    send_timeout 90s;

#
}
}

server {
    listen 443 ssl;
    listen [::]:443 ssl;
    server_name lk.feedbacktalk.ru;

    access_log /var/log/nginx/host.access.log main;
    include /etc/nginx/conf.d/ssl/ssl.conf;
    location / {

        proxy_hide_header Access-Control-Allow-Origin;
        proxy_hide_header Access-Control-Allow-Credentials;
        add_header 'Access-Control-Allow-Origin' "$http_origin" always;
        add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT';

        add_header 'Access-Control-Allow-Headers' 'DNT,Authorization,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,Content-Type,Range';
        if ($request_method = OPTIONS) {
            return 204;
        }
    }
# X-Real-IP
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-Host $host;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

    proxy_headers_hash_max_size 1024;
    proxy_headers_hash_bucket_size 128;
    proxy_pass http://fbt-frontend:3001/;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
## Timeout
    proxy_connect_timeout 90s;
    proxy_send_timeout 90s;
    proxy_read_timeout 90s;
    send_timeout 90s;

#
}
}

server {
    listen 443 ssl;
    listen [::]:443 ssl;
    server_name content.feedbacktalk.ru;

    access_log /var/log/nginx/host.access.log main;

```

```

include /etc/nginx/conf.d/ssl/ssl.conf;
location / {

    proxy_hide_header Access-Control-Allow-Origin;
    proxy_hide_header Access-Control-Allow-Credentials;
    add_header 'Access-Control-Allow-Origin' "$http_origin" always;
    add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT';

    add_header 'Access-Control-Allow-Headers' 'DNT,Authorization,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,Content-Type,Range';
    if ($request_method = OPTIONS) {
        return 204;
    }
    # X-Real-IP
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-Host $host;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

    proxy_headers_hash_max_size 1024;
    proxy_headers_hash_bucket_size 128;
    proxy_pass http://fbt-minio:9000/;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
    ## Timeout
    proxy_connect_timeout 90s;
    proxy_send_timeout 90s;
    proxy_read_timeout 90s;
    send_timeout 90s;
    #
    }
}

server {
    listen 443 ssl;
    listen [::]:443 ssl;
    server_name api.feedbacktalk.ru;

    access_log /var/log/nginx/host.access.log main;
    include /etc/nginx/conf.d/ssl/ssl.conf;

    location /js/ {

        proxy_hide_header Access-Control-Allow-Origin;
        proxy_hide_header Access-Control-Allow-Credentials;
        add_header 'Access-Control-Allow-Origin' "$http_origin" always;
        add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT';

        add_header 'Access-Control-Allow-Headers' 'DNT,Authorization,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,Content-Type,Range';
        if ($request_method = OPTIONS) {
            return 204;
        }
        # X-Real-IP

```

```

    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-Host $host;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

    proxy_headers_hash_max_size 1024;
    proxy_headers_hash_bucket_size 128;
    proxy_pass http://fbt-survey-generator:8001/;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
## Timeout
    proxy_connect_timeout 90s;
    proxy_send_timeout 90s;
    proxy_read_timeout 90s;
    send_timeout 90s;
#
}
location /di/ {

    proxy_hide_header Access-Control-Allow-Origin;
    proxy_hide_header Access-Control-Allow-Credentials;
    add_header 'Access-Control-Allow-Origin' "$http_origin" always;
    add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT';

    add_header 'Access-Control-Allow-Headers' 'DNT,Authorization,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,Content-Type,Range';
    if ($request_method = OPTIONS) {
        return 204;
    }
# X-Real-IP
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-Host $host;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

    proxy_headers_hash_max_size 1024;
    proxy_headers_hash_bucket_size 128;
    proxy_pass http://fbt-data-import-api:8002/;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
## Timeout
    proxy_connect_timeout 90s;
    proxy_send_timeout 90s;
    proxy_read_timeout 90s;
    send_timeout 90s;
#
}

location /cu/ {

    proxy_hide_header Access-Control-Allow-Origin;
    proxy_hide_header Access-Control-Allow-Credentials;
    add_header 'Access-Control-Allow-Origin' "$http_origin" always;
    add_header 'Access-Control-Allow-Methods' 'GET, POST, OPTIONS, PUT';

    add_header 'Access-Control-Allow-Headers' 'DNT,Authorization,User-Agent,X-Requested-With,If-Modified-Since,Cache-Control,Content-Type,Range';
    if ($request_method = OPTIONS) {

```

```

    return 204;
}
# X-Real-IP
    proxy_set_header X-Real-IP $remote_addr;
    proxy_set_header X-Forwarded-Host $host;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

    proxy_headers_hash_max_size 1024;
    proxy_headers_hash_bucket_size 128;
    proxy_pass http://fbt-connect2gounisender:8003/;
    proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;
## Timeout
    proxy_connect_timeout 90s;
    proxy_send_timeout 90s;
    proxy_read_timeout 90s;
    send_timeout 90s;
#
}

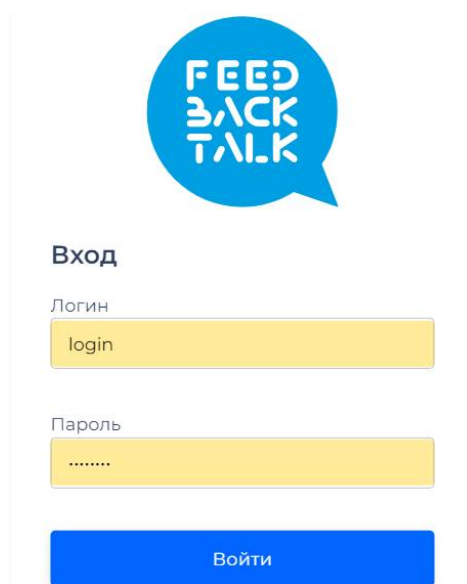
}

```

Результат установки и доступ

После выполнения указанных выше шагов в вашем контуре будет развернуто программное обеспечение FeedBackTalk. URL для входа в систему указан в разделе «Публикация сервисов».

Вход в программу с логином login с паролем password.



The image shows a login interface for FeedBackTalk. At the top is a blue speech bubble logo with the text 'FEED BACK TALK' in white. Below the logo is the heading 'Вход' (Login). There are two input fields: 'Логин' (Login) with the text 'login' and 'Пароль' (Password) with masked characters '.....'. At the bottom is a blue button labeled 'Войти' (Login).

Установка системы закончена. Вы можете зайти на портал и начать работу с системой FeedBackTalk. Документ «Руководство по работе с FeedBackTalk.docx» доступен на сайте производителя системы по ссылке <https://tech-intel.ru/documentation> и на портале в меню «Документация».

Сводная таблица по развернутым сервисам

| Виртуальная машина | Название docker-контейнеров и используемые порты | Примечание |
|------------------------------|--|------------|
| PostgreSQL | fbt-postgres 5432 | |
| Clickhouse | fbt-clickhouse 8123, 9000, 9009 | |
| Apache Kafka | fbt-kafka 2181, 9092, 9093 fbt-kafka-ui 8080 | |
| Backend | fbt-backend 8000 | |
| Frontend | fbt-frontend 3001 | |
| S3-minIO | fbt-minio 9000 | |
| Public Services (public API) | fbt-survey-generator 8001 | |
| | fbt-data-import-api 8002 | |
| | fbt-connect2gounisender 8003 | |

Контактная информация

Связаться со специалистами службы технической поддержки ООО «ТехноИнтеллект» можно одним из следующих способов:

- Сайт: <https://tech-intel.ru>
- Телефон: +7(925) 788-68-44
- Email: support@tech-intel.ru

Фактический адрес размещения службы поддержки: РФ, 125047, Город Москва, вн.тер.г. муниципальный округ Тверской, ул Бутырский Вал, д. 10