



ООО «ПОЛЕТ»

ИНН 7730582675 КПП 773301001 ОГРН 1087746610896

125373, Москва, Походный проезд д. 14, цок.эт, пом.І, ком .3, тел: 8 (967) 104-01-09

Retail Suite.Business Intelligence

Инструкция по установке.

1. Инструкция по установке программы для ЭВМ Retail Suite.Business Intelligence.

Установка ПО Retail Suite.Business Intelligence на сервер предполагается с использованием контейнеризации Docker. Для установки требуется дистрибутив продукта, доступный для скачивания в личном кабинете клиента по адресу lk.supsoft.ru после покупки продукта.

Минимальные требования к стенду

CPU - 8

RAM - 16 Гб

DISK - 50 Гб

ОС – Ред ОС, Астра linux, Debian linux.

СУБД – PostgresPRO, PostgreSQL.

Перед установкой продукта необходимо установить ПО Docker по официальной инструкции к используемому дистрибутиву <https://docs.docker.com/engine/install/>

2. Авторизоваться в корпоративном docker-registry

```
echo "${REGISTRY_PASSWORD}" | docker login "${REGISTRY_HOST}" -u  
"${REGISTRY_USER}" --password-stdin
```

3. Загрузить на сервер, где планируется установить программное обеспечение Retail Suite.Business Intelligence, архив с необходимой версией релиза,

предварительно скачанный в личном кабинете клиента по адресу lk.supsoft.ru.

Архив дистрибутива имеет вид distribution-`{RS.BI_VERSION}`.zip

4. Переместить полученный архив в /opt, распаковать его и переименовать директорию, полученную в результате распаковки архива

```
mv distribution-{RS.BI_VERSION} /opt && \  
cd /opt && \  
unzip distribution-{RS.BI_VERSION} && \  
mv distribution rs.bi-micro
```

5. Переименовать .env_template в .env

```
cd /opt/rs.bi-micro && mv .env_template .env
```

6. Внести соответствующие изменения в .env-файл. Ниже перечислены переменные, значения которых необходимо установить

POSTGRES_PASSWORD - пароль пользователя postgres. Все микросервисы используют пользователя postgres, при работе с базой данных

KEYCLOAK_ADMIN_PASSWORD - пароль admin-пользователя для доступа в консоль keycloak

KEYCLOAK_HOSTNAME - хостнейм или ip-адрес хоста, который будет ассоциирован с консолью keycloak.

API_GATEWAY_EXTERNAL_URL - URL по которому будет доступен портал.

SHELFSPACE_ADMIN_EMAIL - email admin-пользователя портала

METATRON_DB_USER - пользователь СУБД postgres

METATRON_DB_PASSWORD - пароль пользователя postgres

DISCOVERY_DATASOURCE_USERNAME - пользователь СУБД postgres

DISCOVERY_DATASOURCE_PASSWORD - пароль пользователя postgres

MDX_DATASOURCE_USERNAME -пользователь СУБД postgres

MDX_DATASOURCE_PASSWORD - пароль пользователя postgres

7. Пример .env-файла, в котором заполнены все обязательные переменные:

```
DOCKER_REGISTRY_HOST=registry.supsoft.ru
```

```
BI_VERSION=97.1.0
```

```
BI_UI_SOURCE=rs
```

```
API_GATEWAY_EXTERNAL_URL=test-docker
```

```
# SSL
```

```
SSL_ENABLED=true
```

```
SSL_TRUST_STORE_PASSWORD=password
```

```
# Portainer
```

```
PORTAINER_EXTERNAL_PORT=9100
```

```
# Postgres
```

```
POSTGRES_EXTERNAL_PORT=5432
```

```
POSTGRES_PASSWORD=password
```

```
# Keycloak
```

```
KEYCLOAK_HOSTNAME=test-docker
```

```
KEYCLOAK_EXTERNAL_PORT=9000
```

```
KEYCLOAK_ADMIN_PASSWORD=admin
```

```
KEYCLOAK_IMPORT_FILE=rs-bi-realm.json
```

```
# Microservices common
```

```
EUREKA_INSTANCE_PREFER_IP_ADDRESS=false
```

```
EUREKA_REGISTRY_EXTERNAL_PORT=8761
```

```
CONFIG_SERVER_EXTERNAL_PORT=8888
```

```
GATEWAY_EXTERNAL_PORT=8765
```

```
# OAuth
```

```
OAUTH_SERVER_URL=http://keycloak:8080
```

```
OAUTH_REALM_NAME=rs-bi
```

```
OAUTH_CLIENT_ID=oauth
```

```
OAUTH_CLIENT_SECRET=PeLRGb4QatijQjKi7DiicJrN6Xb1TWEs
```

```
OAUTH_RESOURCE_ACCESS_RESOURCE_NAME=oauth
```

```
# Discovery Legacy
```

```
DISCOVERY_LEGACY_VERSION=development-latest
```

```
DISCOVERY_LEGACY_SERVER_URL=http://discovery-legacy:8180
```

```
METATRON_JAVA_OPTS=$JAVA_OPTS -Xms2g -Xmx4g -
```

```
XX:MaxMetaspaceSize=512m
```

```
METATRON_ENV_PROFILES=local,postgres-default-db,logging-  
file,scheduling,microservice,managements
```

```
METATRON_CACHE_PATH=cache
```

```
METATRON_SMTP_HOST=<SMTP-узел>
```

```
METATRON_SMTP_PORT=<SMTP-порт>
```

```
METATRON_SMTP_USERNAME=<SMTP-пользователь>
```

```

METATRON_SMTP_PASSWORD=<SMTP-пароль>

METATRON_DB_TYPE=postgres
METATRON_DB_SCHEMA=discovery_legacy
METATRON_DB_URL=jdbc:postgresql://db:5432/rs_bi?currentSchema=$METATRON_DB_SCHEMA
METATRON_DB_USER=postgres
METATRON_DB_PASSWORD=${POSTGRES_PASSWORD}

METATRON_MAIL_BASE_URL=test-docker
HADOOP_HDFS_BASE_URL=hdfs://<HDFS_URL>:8020
HADOOP_SPARK_LIVY_URL=http://<SPARK_LIVY_URL>:8999
HADOOP_SPARK_HISTORY_URL=http://<SPARK_HISTORY_URL>:18081
HADOOP_HIVE_HOSTNAME=<HIVE_URL>
HADOOP_HIVE_PORT=10500
HADOOP_HIVE_USER=hive
HADOOP_HIVE_PASSWORD=hive
HADOOP_THRIFT_HIVE_URL=thrift://<THRIFT_HIVE_URL>:9083

# Discovery API
DISCOVERY_DATASOURCE_URL=jdbc:postgresql://db:5432/rs_bi
DISCOVERY_DATASOURCE_SCHEMA_NAME=discovery_api
DISCOVERY_DATASOURCE_USERNAME=postgres
DISCOVERY_DATASOURCE_PASSWORD=${POSTGRES_PASSWORD}

# MDX API
MDX_DATASOURCE_URL=jdbc:postgresql://db:5432/rs_bi
MDX_DATASOURCE_SCHEMA_NAME=mdx_api
MDX_DATASOURCE_USERNAME=postgres
MDX_DATASOURCE_PASSWORD=${POSTGRES_PASSWORD}

# Logging
LOGGING_LOGSTASH_ENABLED=false
LOGGING_LOGSTASH_DESTINATION=localhost:5000

```

8. После подготовки .env-файла, необходимо запустить все микросервисы.

Запуск выполняется с использованием docker-compose
`cd /opt/rs.bi-micro && docker compose --env-file .env up -d`

9. Статус запущенных контейнеров можно посмотреть через docker

`cd /opt/rs.bi-micro && docker compose ps`

Либо через portainer, доступный по адресу

`http://${SERVER_IP}:9000`

Во время первого доступа в portainer, необходимо будет создать админ-пользователя.

10. Для доступа к порталу удобно использовать nginx в качестве прокси.

Пример виртуального хоста для портала:

```

server {
    listen 80;

```

```

server_name test-docker;
charset utf-8;

set $gateway 'http://127.0.0.1:7070';

proxy_http_version 1.1;
proxy_set_header Host $host;
proxy_redirect .* $host;
proxy_cookie_domain .* $host;

proxy_connect_timeout 3600;
proxy_send_timeout 3600;
proxy_read_timeout 3600;

default_type text/html;
client_max_body_size 0;

error_log /var/log/nginx/test-docker/error.log error;
access_log /var/log/nginx/test-docker/access.log;

location ^~/ {
    proxy_pass $gateway$request_uri;
}

location ^~/stomp/ {
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
    proxy_pass $gateway$request_uri;
}
}

```

11. Просмотр логов контейнера
`cd /opt/rs.bi-micro && docker logs {ID_CONTAINER}`
12. Остановка стека докер контейнеров
`cd /opt/rs.bi-micro && docker compose down --volume`

Обновление стенда на следующий релиз

1. Остановить контейнеры
`docker compose down`
2. В файле `.env` в параметре `BI_VERSION` прописать новое значение версии, и запустить повторно стек контейнеров. Произойдет скачивание новых образов `docker` содержащих более свежую версию `rs.bi`.