АВТОМАТИЗИРОВАННАЯ СИСТЕМА

СБЕР ТАРГЕТ

Инструкция по установке

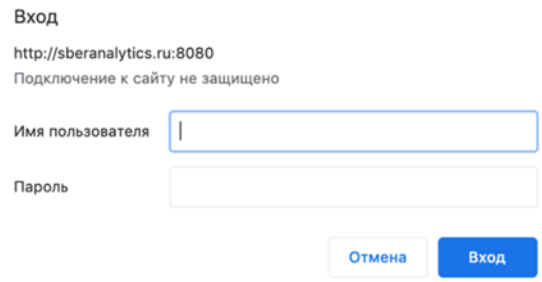
ВЕРСИЯ ДОКУМЕНТА 1.0
ДАТА 20.09.2021
ПАО СБЕРБАНК

Оглавление

1. Инструкция по локальному развёртыванию АС СберТаргет

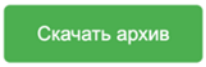
Для скачивания дистрибутива автоматизированной системы «Сбертаргет» необходимо выполнить следующие действия:

1. Зайти на сайт http://sberanalytics.ru:8080

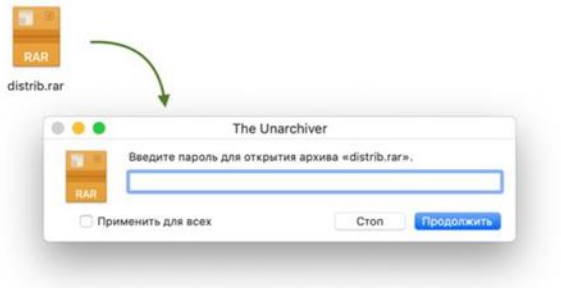


Используйте имя пользователя и пароль из комплекта документов по АС Сбертаргет. Если у Вас нет пользователя на данном сайте, его можно запросить, написав на email sberanalytics-support@sberbank.ru

2. Скачать архив, выложенный на сайте, используя кнопку «Скачать архив» (кнопка появляется после входа на сайт):



3. Распаковать архив, используя пароль от архива.



Если у Вас нет пароля от архива, его можно запросить, написав на email sberanalytics-support@sberbank.ru

Для установки требуется выполнить следующую последовательность действий:

1. Запустить pipeline установки. Выбрать следующие плейбуки:

- WAS_STOP_CLUSTER,
- WAS_FPI_COMMON,
- WAS_FPI_INSTALL,
- WAS_FPI_DEPLOY,
- WAS_START_CLUSTER,
- IMPORT_SM_PARAMS,
- IMPORT_SECURITY_PARAMS,

- IMPORT_ERRORS_PARAMS,
- IMPORT_DICTIONARY_PARAMS,
- IMPORT_LOGGER_PARAMS,
- IMPORT_FILE_TRANSFER_PARAMS,
- IMPORT_AUDIT_PARAMS,
- IMPORT_SUP2_PARAMS,
- IMPORT_SECURITY_DICTIONARIES_PARAMS,
- IMPORT_LOGGER_FILTERS_PARAMS

2. Донастроить/изменить средозависимые параметры:

Описание (что нужно указать)	**Правило заполнения (Пример)**
sbtg.server	Сервер шлюза, который направляет вызовы Пример: https://tvls-efs0004.sigma.sbrf.ru:8091
sbtg.redirect.uri	Ссылка на страницу Сбертаргет, для редиректа клиента Пример: https://target_ift.sigma.sbrf.ru
fintech.login.server	URL сервера аутентификации (логин пользователя) СберБизнес Пример: http://sbt-oafs-941.sigma.sbrf.ru:9080
fintech.sberbusiness.id.servers	URL-ы серверов авторизации посредством сервиса SberBusiness ID можно указать несколько через знак «;» Пример: https://int-rest-sbbol-ift.ca.sbrf.ru:9443
fintech.tariffication.servers	URL-ы серверов управления тарифами можно указать несколько через знак «;» Пример: http://str-vat-was1107.vm.mos.cloud.sbrf.ru:9080
fintech.sbtg.client.id	Указывается идентификатор сервиса Пример, 76716
intech.sbtg.client.secret	Указывается секрет Пример, hKUQ79LW
fintech.oauth.scope	Область сведения SSO – группа атрибутов пользователя, разрешенных для передачи внешнему серверу Например, openid+sberAdv
clickhouse.servers	Внутренний эндпоинт приложения поставщика данных Пример: http://ingress-ci02134870-idevgen2-ci02134870-idevgen-clickhous.apps.dev-gen2.ca.sbrf.ru; https://ci02134870-idevgen-clickhous.ingress.apps.dev-gen2.ca.sbrf.ru

3. Установить jndi переменные для приложения
4. Зайти по ссылке www.target.sber.ru.

2. Инструкция по началу использования сервиса СберТаргет

2.1. Системные требования

СберТаргет – это сервис, который предоставляется как услуга (SaaS). Все необходимые для работы модули сервиса развертываются внутри облачной инфраструктуры, предоставляемой поставщиком услуги, что позволяет избежать локальных установок на устройство конечного потребителя.

Для работы с сервисом необходимо наличие персонального компьютера или смартфона с выходом в сеть интернет и установленный на устройство вэб-браузер.

Сервис предоставляется для операционных систем Windows, MacOS, Linux, iOS, Android.

2.2. Первичная настройка системы

Для первичной настройки системы и формирования бизнес-аккаунта клиента необходимо перейти на страницу регистрации сервиса с помощью устройства, поддерживающего выход в интернет.

3. Настройка баз данных

3.1. Настройка основной БД

При первой настройке необходимо внутри скрипта createSchemas.sql переименовать aps на нужную БД, после чего выполнить скрипт createSchemas.sql из-под суперпользователя со всеми привилегиями (лежит в дистрибутиве в папке scripts).

Этот скрипт выполняет следующие действия:

1. **Создает схемы: app** - личный кабинет пользователя
2. **Создает пользователей: appuser** - личный кабинет пользователя
3. **Прописывает гранты:** для **appuser** – доступ со всеми привилегиями к схеме app

Данный скрипт необходимо запускать вручную. Перед запуском нужно изменить пароли пользователей, приведенные внутри скрипта.

3.2. Настройка БД аудита

Для настройки аудита необходимо внутри скрипта create_audit_schema.sql переименовать aps на нужную БД, после чего запустить этот скрипт под суперпользователем со всеми привилегиями (лежит в дистрибутиве в папке scripts).

3.3. Для информации

Скрипт создает:

- схему "audit" для аудита в базе данных Postgres;
- пользователя "audituser" с правами на insert и select в схеме "audit";
- структуру базы данных.

Данный скрипт необходимо запускать вручную. Перед запуском нужно изменить пароли пользователей, приведенные внутри скрипта.

4. Настройка среды окружения для развертывания личного кабинета пользователя

Для взаимодействия с сервисами FintechAPI, требуется:

- Положить в дисковое пространство файлы с хранилищами сертификатов clientcert_psw_testtest.jks и clienttrust_psw_testtest.jks, в которых содержаться сертификаты Root CA, Sberbank Issuing CA и сертификат с закрытым ключом.

- Далее нужно будет прописать путь до файлов в соответствующие JNDI переменные (java:/fintech.store.key.file.path и java:/fintech.store.trust.file.path см. таблицу в п. 4. Настройка сервера WildFly для развертывания личного кабинета пользователя и администратора).

5. Настройка сервера WildFly

5.1. Первичная настройка сервера

1. Запускаем WildFly.
2. Заходим в консоль WildFly http://<host>:9990/console/index.html
3. Переходим на вкладку Deployments
4. Нажимаем «+» (ADD), далее на Upload deployment, выбираем файл драйвера postgresql - postgresql-42.2.4.jar (лежит в дистрибутиве в папке scripts). Далее нажимаем Next, Finish, Close.
5. Переходим на вкладку Configuration, выбираем пункт Subsystems, внутри subsistems выбираем пункт Datasources & Drivers
6. Нажимаем «+» , Add datasource
7. Выбираем PostgreSQL, Next
8. Вводим имя «APS_PostgresDS», JNDI имя «java:/APS_PostgresDS», Next
9. Выбираем установленный ранее драйвер postgresql-42.2.4.jar, Next
10. Вводим путь к базе данных jdbc:postgresql://<host>:5432/iftapm – должен содержать имя нужной БД
    - Имя пользователя appuser
    - Пароль из скрипта createSchemas.sql для appuser

Нажимаем Next и Test Connections. Убеждаемся, что соединение установлено. Далее - Next, Close

11. Жмём View на созданном datasource.
12. На вкладке Attributes жмем edit, очищаем поле Datasource Class, Save
13. На вкладке Connection жмем edit:
    - в поле Connection Properties пишем currentSchema=app, Enter, под полем должно появиться добавленное свойство.
    - Так же добавляем свойство autosave ⇒ conservative
    - Жмём Save.
14. На вкладке Timeouts, edit устанавливаем
    - Blocking Timeout Wait Millis = 30000
    - Save
15. Внутри subsistems выбираем пункт Naming, жмем View -> binding
16. Добавляем параметры jndi, следующим образом: нажимаем add, вводим имя jndi, раскрываем Optional Fields, заполняем поле value значением, жмем Add. Таким образом добавляем пары jndi переменных, из таблицы ниже.
17. Далее переходим в раздел Configuration⇒System Properties ⇒View нажимаем Add.

18. Вводим следующие значения в соответствующих полях:

- Name=io.undertow.legacy.cookie.ALLOW_HTTP_SEPARATORS_IN_V0
- Value=true

19. Нажимаем Add

5.2. Настройка аудита

Для работы приложения в Wildfly необходимо создать datasource "java:/APS_Audit_PostgresDS".

Создается по аналогии с java:/APS_PostgresDS (см. 4. Настройка сервера WildFly для развертывания личного кабинета пользователя и администратора)

5.3. Дополнительно

- Url для подключения к бд jdbc:postgresql://10.53.90.113:5432/iftapm (ИФТ стенд).
- Имя пользователя указываем "audituser". Пароль из скрипта.
- В Connection Properties добавляем currentSchema=audit.
- Остальное как при настройке java:/APS_PostgresDS.

5.4. Настройка хранилища для шифрования паролей

1. Перейти в раздел /usr/WF/WF_ИМЯ_AC/bin/
2. Создать хранилище для Vault.

$ keytool -genseckey -alias vault(короткое ИМЯ_AC) -storetype jceks -keyalg AES keysize 128 -storepass взять из teampass -keypass взять из teampass -keystore keystore /usr/WF/WF_ИМЯ_AC/vault.keystore
(для ПРОМ серверов файл назвать PROM.keystore, для ТЕСТОВЫХ файл назвать TEST.keystore)

Пример
$ keytool -genseckey -alias STRELA -storetype jceks -keyalg AES -keysize 128 storepass взять из teampass -keypass взять из teampass - keystore /usr/WF/WF_ИМЯ_AC/TEST.keystore
пароли -storepass ПАРОЛЬ -keypass ПАРОЛЬ должны быть одинаковы

3. Добавить необходимый пароль в Vault.

$ vault.sh --keystore KEYSTORE_URL --keystore-password KEYSTORE_PASSWORD -alias KEYSTORE_ALIAS --vault-block VAULT_BLOCK -- attribute ATTRIBUTE --sec-attr SEC-ATTR --enc-dir ENC_FILE_DIR --iteration ITERATION_COUNT --salt взять из teampass

Пример

./vault.sh --keystore /usr/WF/WF_ИМЯ_AC/TEST.keystore --keystore-password ПАРОЛЬ --alias STRELA --vault-block BIND --attribute PWD3 --sec- attr Шифруемый пароль (от учетной записи bind) --enc-dir /usr/WF/WF_ИМЯ_AC/ --iteration 50 -salt взять из teampass

4. Добавить в файле

/usr/WF/WF_ИМЯ_AC/standalone/configuration/standalone.xml внутри тега server:

<vault>

<vault-option name="KEYSTORE_URL" value="ПУТЬ ДО keystore ИЗ п.2"/> <vault-option name="KEYSTORE_PASSWORD" value=" маска сформируется при шаге_2 в ответе – начинается на MASK"/>

<vault-option name="KEYSTORE_ALIAS" value="APM"/>

<vault-option name="SALT" value="Должен совпадать с введенным в шаге 3"/>

<vult-option name="ITERATION_COUNT" value="50"/>

<vault-option name="ENC_FILE_DIR" value="/usr/WF/ WF_ИМЯ_AC/"/> </vault>

Например:

<vault>

<vault-option name="KEYSTORE_URL" value="/usr/WF/WF_EFS/TEST.keystore"/>

<vault-option name="KEYSTORE_PASSWORD" value="MASK-FIfXn10v5vO"/>

<vault-option name="KEYSTORE_ALIAS" value="APM"/> <vault-option name="SALT" value="12345678"/>

<vault-option name="ITERATION_COUNT" value="50"/>

<vault-option name="ENC_FILE_DIR" value="/usr/WF/WF_EFS/"/> </vault>

5. Для нужных параметров jndi заменить значение поля на то, которое было получено в ответе пункта 3, в консоли администратора WF, либо в standalone.xml, например:

<simple name="java:/aps.admin.ldap.password" value="${VAULT::BIND::PWDLDAP::1}"/>

Важно

Пункты 3 и 5 нужно выполнить для параметров jndi:

- java:/aps.admin.ldap.password
- java:/fintech.auth.clientSecret
- java:/aps.admin.jwtSecret
- java:/fintech.store.key.pass
- java:/fintech.store.key.private.pass

- java:/fintech.store.trust.pass

Также для паролей Datasources:

- java:/APS_PostgresDS
- java:/APS_Audit_PostgresDS

5.5. Настройка логирования

1. Заходим в конфигурацию логов:

Configuration⇒Subsystems⇒Logging⇒Configuration – View

2. Заходим в FORMATTER - Pattern Formatter. В списке должна быть запись

PATTERN %d{yyyy-MM-dd HH:mm:ss,SSS} %-5p [%c] (%t) %s%e%n

Если записи нет, ее необходимо добавить.

3. Заходим в Handler - Periodic Size Handler. Здесь должна быть запись

FILE_ROTATE ALL jboss.server.log.dir/server.log

Имя может быть другим. Выделяем запись, нажимаем Edit. Должны быть установлены следующие параметры:

Append	true
Autoflush	true
Enabled	true
Encoding	
File / Path	server.log
File / Relative To	jboss.server.log.dir
Filter Spec	
Formatter	%d{HH:mm:ss,SSS} %-5p [%c] (%t) %s%e%n
Level	ALL
Max Backup Index	99
Named Formatter	PATTERN
Rotate Size	5m
Rotate On Boot	false
Suffix	yyyy-MM-dd

Сохраняем.

4. Root Logger – Edit. В Handlers нужно отредактировать чтобы были только FILE_ROTATE(Handler из п.3) и CONSOLE

5. Заходим в Categories. Жмём Add. Добавляем записи ниже (указываем только Name и Level).

   - org.hibernate.SQL INFO
   - org.springframework.web.filter.CommonsRequestLoggingFilter

     DEBUG
   - org.springframework.web.client.RestTemplate DEBUG
   - ru.sbrf.aps DEBUG

5.6. Развертывание приложений личного кабинета пользователя и редактора контента

1. В консоли WF на вкладке Deployment нажать +(Add) - Upload Deployment.
2. Выбрать или перетащить в область файл bh/application-<version>.war .
3. Нажать next.
4. Нажать finish.

Те же действия выполнить с файлом bh/admin-<version>.war .

- admin-6.0.0.war
- application-6.0.0.war

Указанные файлы находятся в дистрибутиве в папке bh.

6. Настройка NGINX

6.1. Настройка nginx в сегменте sigma (для части приложения редактора контента)

Обновить статику nginx(/opt/nginx/html) содержимым из папки fe-contenteditor дистрибутива.

Конфигурация сервера NGINX (nginx.conf) располагается в дистрибутиве в папке nginx_configuration/content_editor/

Необходимые изменения в разделе location ~ /aps-admin:

В proxy_pass указать адрес приложения ВН части редактора контента

Необходимые изменения в разделе server (раздел начинается с 51-й строки):

В server_name указать host сервера nginx для приложения редактор контента (сегменте сигма)

В ssl_certificate указать путь в системе до сертификата.

В ssl_certificate_key указать путь в системе до секретного ключа.

После изменений в файле nginx.conf необходимо обновить соответствующий файл (в /opt/nginx/conf) на стенде.

6.2. Настройка nginx в сегменте internet (для клиентской части приложения)

Обновить статику nginx(/opt/nginx/html) содержимым из папки fe-client дистрибутива.

Конфигурация сервера NGINX (nginx.conf) располагается в дистрибутиве в папке nginx_configuration/client/

Необходимые изменения в разделе location ~ /aps:

В proxy_pass указать адрес приложения ВН редактора контента

Для боевого сервера добавить параметры (5 штук, описаны ниже):

#указать путь в системе до файла с сертификатом в формате PEM для аутентификации на проксируемом сервере.

proxy_ssl_certificate /opt/nginx/ssl/00CA0001CAPMngx99usr.pem;

#указать путь в системе до файла с секретным ключем в формате PEM для аутентификации на проксируемом сервере.

proxy_ssl_certificate_key /opt/nginx/ssl/00CA0001CAPMngx99usr.key;

указать путь в системе до файла с доверенными сертификатами CA в формате PEM, используемыми при проверке сертификата проксируемого сервера.

proxy_ssl_trusted_certificate /opt/nginx/ssl/CA_chain.pem; proxy_ssl_verify off; proxy_ssl_verify_depth 2;

Необходимые изменения в разделе server (раздел начинается с 55-й строки):

В server_name указать host сервера nginx для клиентской части приложения (сегменте internet)

В ssl_certificate указать путь в системе до сертификата nginx сервера.

В ssl_certificate_key указать путь в системе до секретного ключа.

После изменений в файле nginx.conf необходимо обновить соответствующий файл (в /opt/nginx/conf) на стенде.

6.3. Дополнительно

Перегрузить сервер приложений.