Process Mining от Сбера

version 2.8.1

Руководство по установке продуктов "Process Mining от Сбера"

February 27, 2024

Содержание

Установка

1. Системные требования

1.1 Сценарии установки продуктов

Возможны различные сценарии установки продуктов:

- Установка продуктов «Process mining от Сбера» и «Task Mining от Сбера»
    - Устанавливаются компоненты: FE, BI, BE, ML, ETL, TM, Auth, PG и CH
- Установка продукта «Process Mining от Сбера»
    - Устанавливаются компоненты: FE, BI, BE, ML, ETL, Auth, PG и CH.
- Установка продукта «Task Mining от Сбера»
    - Устанавливаются компоненты: FE, TM, PG и CH.

Для развертывания компонентов продукта «Process Mining от Сбера» потребуется минимально один сервер, но рекомендуется выделить отдельно сервер под Приложение и отдельные серверы под каждую СУБД.

Для развертывания используется управляющий сервер (CI/CD), в роли которого выступает отдельный сервер, либо совмещенный с сервером Приложений (СП). При необходимости масштабирования предусмотрена возможность выделить под каждый компонент системы отдельный СП.

На сервере СУБД Postgresql будет расположен 1 экземпляр СУБД PostrgeSQL (PG), на сервере СУБД Clichhouse будет расположен 1 экземпляр СУБД ClickHouse (CH).

PG – компонент СУБД PosrgteSQL, содержит:

- БД sberprocessmining
    - Схема processmining (сервис pm-core)
    - Схема analytics (сервис pm-bi)
    - Схема auth (сервис pm-auth)
    - Схема etl (сервис pm-etl)
    - Схема taskmining (сервис pm-tm)

CH – компонент СУБД ClickHouse, содержит БД с пользовательскими данными для работы сервисов BE, BI, ML, ETL, TM.

- _00UserData_spm
- _01RawData_spm
- _02PreparedData_spm
- _03ResearchData_spm
- _01DictionaryData
- TaskMining

1.2. Требования к серверам, мощностям, ПО

1.2.1. Минимальные требования к конфигурации сервисов

	CPU	RAM	HDD
Сервер приложений (СП)	8	16	60
Сервер СУБД PostgreSQL	2	4	100

	CPU	RAM	HDD
Сервер СУБД ClickHouse	8	32	100
Управляющий узел ansible (CI/CD)	2	4	60

1.2.2. Требования к программному обеспечению

	Требуемое СПО	Версия СПО	download
ОС	CentOS, RHLE	7.X	https://www.centos.org/download/, https://access.redhat.com/downloads
	Debian	10.X	https://www.debian.org/distrib/
Сервер приложения (СП)	Docker	20.X и выше	https://docs.docker.com/engine/install/debian/
Сервер СУБД	PostgreSQL server	10.X - 13.X	Поставляются дистрибутивно
	ClickHouse server	22.X	Поставляются дистрибутивно
Сервис кеширования redis	Redis	7.0.9	Поставляются дистрибутивно
Управляющий ansible (CI/CD)	Ansible	2.9 и выше	https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html

1.2.3. Требования к сетевым доступам

Источник	Цель	Порты
Рабочая станция пользователя	Сервер приложений	tcp/80, tcp/443
Сервер приложений	Сервер СУБД PostgreSQL	tcp/5432
Сервер приложений	Сервер СУБД ClickHouse	tcp/8123, tcp/9000
Рабочая станция администратора	Сервер, Сервер СУБД	tcp/22

Сетевая схема описана в П10 данной инструкции.

1.2.4. Требование к DNS записи

DNS
Запись типа «A» указывающая на сервер приложения (пример: spm.company.ru)

1.2.5. Прикладное ПО состоит из сервисов

Имя сервиса	Описание
pm-ui	Компонент FrontEnd (FE)
pm-bi	Компонент для исследования и визуализации данных (BI)
pm-core	Компонент BackEnd (BE)
pm-ml	Компонент Machine Learning для анализа пользовательских путей (ML)

Имя сервиса	Описание
pm-etl	Компонент ETL для извлечения, анализа и загрузки данных (ETL)
pm-auth	Компонент авторизации пользователей (Auth)
pm-tm	Компонент сервиса логирования task mining (TM)
redis	Компонент кеширования данных

2. Установка программного обеспечения

2.1. Состав поставки

Поставка представляет собой набор архивов для каждого сервиса и СУБД, скрипты развертывания.

Имена архивов формируются по следующим правилам:

1. Приложения:
   - <Имя_сервиса>-<Версия_поставки>.tgz`
2. СУБД:
   - postgres-server-<Версия_postgres>.tgz
   - clickhouse-server-<Версия_clickhouse>.tgz
3. Сервис кеширования данных:
   - redis-<Версия_redis>.tgz
4. Скрипты развертывания:
   - deploy-<Версия_поставки>.tar

2.2. Установка

2.2.1. Подготовка окружения

1. Выполнить на серверах Приложения и СУБД установку СПО Docker версии 20.X и выше:
   - 1.1. Добавить репо docker (подробное описание есть в официальной инструкции
   - 1.2. Выполнить установку пакетов (docker и docker-compose):

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

     Есть возможность подключаться по протоколу ssh с управляющего сервера CI/CD.

     Рекомендуется настроить подключение на серверы не по паролю, а по ssh-ключу.

     Настройка подключения к серверу по ssh-ключу:

     - 1.2.1. Команда создаст 2048-битную пару RSA ключей, приватный и публичный в домашней директории пользователя на сервере:

       ```
       ssh-keygen –b 2048 ~/.ssh/<имя сформированного ключа>.pub
       <пользователь>@<ip-адрес сервера к которому будет идти подключение>
       ```

     - 1.2.2. Далее тиражируем публичный ключ на сервер, к которому будет идти подключение командой:

       ```
       ssh-copy-id -i ~/.ssh/<имя сформированного ключа>.pub
       <пользователь>@<ip-адрес сервера к которому будет идти подключение>
       ```

- 1.2.3. После выполнения команды на хосте в файле

  `/home/<пользователь>/.ssh/authorized_keys`

  будет прописан публичный ключ для подключения пользователем по ssh-ключу.

2. Получить файлы лицензии по запросу.
3. Скачать дистрибутивы по ссылке из поставки.

2.2.2. Развертывание управляющего сервера CI/CD

1. На управляющем сервере (CI/CD) выполнить установку ansible, используя менеджер пакетов python. В данной инструкции описан пример установки через **pip**, также предусмотрена возможность выполнить установку используя менеджеры пакетов ОС формата **rpm и deb**.

   Установка ansible:

   - 1.1. Создать виртуальное окружение python venv:

     ```
     python -m venv venv
     ```

   - 1.2. Активировать venv:

     ```
     source venv/bin/activate
     ```

   - 1.3. Установить пакет ansible через пакетный менеджер python (pip):

     ```
     pip install ansible
     ```

2. Распаковать архив deploy-<Версия_поставки>.tar на управляющем сервере (CI/CD).

   `/opt/deploy`

3. Скопировать файл `deploy-<Версия_поставки>/config.yml` в папку `/opt/config.yml`
4. Заполнить файл окружения, `/opt/config.yml` указав параметры:

   > **Внимание!**
   >
   > Копируется и заполняется при первой установке

   - 4.1. ip-адреса серверов для развертывания компонентов:
     - `app_host`: <ip-адрес для хоста компонентов Приложения>`
     - `db_host`: <ip-адрес для хоста компонентов СУБД Postgresql, метаданные>
     - `dwh_host`: <ip-адрес для хоста компонентов СУБД ClickHouse, аналитические данные>
   - 4.2. Точка входа для пользователей:
     - `ui_url`: <точка входа для пользователей, указать FQDN>
   - 4.3. Сценарий установки:
     - features: <значение зависит от того, какие продукты устанавливаем>
       - указываем значение `ent` если устанавливаем продукты process mining и task mining по сценарию 1 пункта 1.1 инструкции
       - указываем значение `pm` если устанавливаем только process mining по сценарию 2 пункта 1.1 инструкции
       - указываем значение `tm` если устанавливаем только task mining по сценарию 2 пункта 1.1 инструкции

- Логин и пароль для подключения по протоколу ssh к указанным серверам и способ подключения (значение параметра ssh_password опционально в случае добавления возможности входа по ключу):
  - ssh_user: «<имя пользователя>»
  - ssh_password: «<пароль>»
  - use_local_deploy: «yes/no»
- В случае использования сервера приложений в качестве управляющего узла ansible, т.е. когда все компоненты находятся на одном сервере и отсутствует управляющий сервер CI/CD устанавливаем значение параметра равным yes.

- 4.4. Задать логины и пароли автоматически создаваемого пользователя и админа безопасности Приложения
  - spm_admin_username: «<имя пользователя>»
  - spm_admin_password: «<пароль>»
  - auth_admin_username: «<имя пользователя>»
  - auth_admin_password: «<пароль>»

  Примечание: при установке по сценарию 3 (пункта 1.1 инструкции) данные поля не заполняются.
- 4.5. Для СУБД PostgreSQL и ClickHouse задать пароли:
  - postgres_db_password: «<пароль>»
  - clickhouse_db_password: «<пароль>»

  Если необходимо развернуть с отдельно развернутыми СУБД (не из дистрибутива) необходимо в конфигурационном файле указать данные для подключения, имя пользователя и пароль и предсоздать БД (п.1).
- 4.6. Задать пароль технического пользователя интеграции компонентов
  - analytics_integration_password: «<пароль>»

  Примечание: при установке по сценарию №3 пункта 1.1 инструкции) данные поля не заполняются.
- 4.7. Данные для подключения к сервису кеширования redis
  - redis_integration_password: «<пароль>»

  Примечание: при установке по сценарию №3 (пункта 1.1 инструкции) данные поля не заполняются.
- 4.8. Пароль генерируемого сертификата для BE допускается оставить заданным по умолчанию.
  - ssl_certificate_password: «changeit»
- 4.9. Параметр для выбора расположения директории хранения образов на сервере Приложений:
  - source_image_on_app_server: «no» (Значение допускается оставить по умолчанию)
- 4.10. Директории хранения файлов образов и конфигурационных файлов сервисов:
  - source_images_storage_folder: «/opt/images»
  - app_config_folder: «/opt/spm»

  На указанную директорию должны быть предоставлены права на создание папок и файлов для пользователя, подключающегося с управляющего сервера.
- 4.11. Скопировать вручную в каталог <app_config_folder>/license на сервере Приложения файлы лицензии:
  - license.json
  - license.json.sig

5. При необходимости шифрования паролей использовать ansible-vault. Пример шифрования пароля с использованием ansible-vault:

   ```
   ansible-vault encrypt_string '<пароль, который нужно зашифровать>' – name '<имя параметра>'
   ```

6. Файл deploy-<Версия_поставки>/inventories/local/group_vars/all/env.yml содержит служебную информацию о портах, на которых развернуты сервисы, именах контейнеров, для которых предусмотрена возможность изменений в соответствии с требованиями некоторых окружений, но в общем случае дополнительной настройки не требуется.

2.2.3. Развертывание приложений и СУБД

После завершения этапа заполнения файла окружения выполнить запуск развертывания сервисов.

Команды запуска развертывания сервисов не зависят от выбора сценария установки.

1. В первую очередь запустить развертывание сервисов СУБД PostgreSQL и ClickHouse, выполнив playbook setup.yml командой:

   ```
   ansible-playbook -i inventories/local setup.yml -e "@../config.yml"
   ```

2. После успешного выполнения развертывания сервисов СУБД запустить установку компонентов приложения, запустив playbook deploy.yml командой:

   ```
   ansible-playbook -i inventories/local deploy.yml -e "@../config.yml"
   ```

3. В случае если используется шифрование паролей при помощи ansible-vault добавить к команде запуска развертывания параметр:

   ```
   --ask-vault-pass
   ```

3. Обновление

1. Скачать дистрибутивы по ссылке из поставки.
2. Распаковать архив на управляющем сервере (CI/CD) deploy-<Версия_поставки>.tar
3. Запустить установку компонентов приложения, запустив playbook deploy.yml командой:

   ```
   ansible-playbook -i inventories/local deploy.yml -e "@../config.yml"
   ```

4. В случае обновления до версии HotFix, когда в состав поставки входит только 1 компонент запустить playbook развертывания обновляемого компонента поставки:

   ```
   ansible-playbook -i inventories/local <имя_компонента>.yml -e "@../config.yml"
   ```

4. Удаление

Для удаления продукта необходимо выполнить команды для остановки, удаления данных и контейнеров, проверки удаления:

```
docker stop $(docker ps -a -q) && docker

rm $(docker ps -a -q) && docker volume rm clickdata pgdata && docker ps -a
```

5. Проверка работоспособности

1. Проверить что все компоненты развернуты и запущены в docker контейнерах:

   ```
   docker ps --a
   ```

   В выводе должны быть все компоненты в статусе Up:

```
CONTAINER ID   IMAGE                          COMMAND                  CREATED      STATUS       PORTS                                                NAMES
a857efcb46fc   pm-auth:2.8.1                  "/opt/bitnami/script…"   2 days ago   Up 2 days    0.0.0.0:8090->8090/tcp, 0.0.0.0:9090->9090/tcp   pm-auth
ecd95b4c9bb4   pm-ml:2.8.1                    "/docker-entrypoint.…"   2 days ago   Up 2 days                                                         pm-ml
e64ae4ced97b   pm-core:2.8.1                  "/bin/bash -o pipefa…"   2 days ago   Up 2 hours                                                        pm-core
ffaa5b7a2333   pm-etl:2.8.1                   "/docker-entrypoint.…"   2 days ago   Up 2 days    0.0.0.0:8020->8020/tcp                               pm-etl
f690a641220f   pm-bi:2.8.1                    "/docker-entrypoint.…"   2 days ago   Up 2 days                                                         pm-bi
c98bfe102b43   pm-ui:2.8.1                    "/docker-entrypoint.…"   2 days ago   Up 2 days                                                         pm-ui
153044fcda85   redis:7.0.9                    "/opt/bitnami/script…"   2 days ago   Up 2 days                                                         redis
391688dbea4c   clickhouse-server:22.10.4      "/entrypoint.sh"         2 days ago   Up 2 days                                                         clickhouse-server
a8163caa507a   postgres:13.5                  "docker-entrypoint.s…"   2 days ago   Up 2 days                                                         postgres-server
```

В случае если был выбран сценарий №1 или сценарий №3 установки продуктов:

```
c29023c852bb       pm-tm:2.6.1     "dotnet ProcessMinin…"       5 hours ago  Up 5 hours                pm-tm
```

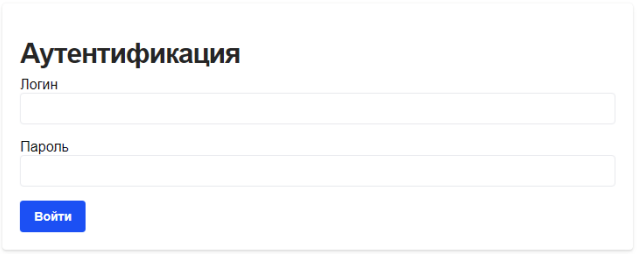
2. Проверить логи работы компонентов сервиса в случае, если контейнер с компонентом не запустился. Для получения лога конкретного компонента выполнить команду:

   ```
   docker logs \--tail \<количество строк лога\> \<имя контейнера\>
   ```

   Пример:

   ```
   docker logs \--tail 1000 pm-ui
   ```

3. Проверка портов в случае возникновения проблем, например, telnet по портам 80 и 443 для FE.
4. Для сценария №1 и сценария №2 установки:
   - Проверка входа на UI по ссылке `ui_url` пользователем `spm_admin_username`.
   - Данные для `ui_url` и `spm_admin_username` были заданы в окружении развертывания.
   - В качестве первоначальной проверки: успешность логина пользователем `spm_admin_username` через компонент авторизации pm-ui по ссылке `ui_url`.
5. Для сценария №3 установки:
   - Перейти по ссылке: `<ui_url>/events`
   - Пользователь увидит форму ввода логина и пароля:



6. Установка TLS сертификата, выпущенного ЦС организации

6.1. Создание запроса на сертификат и выпуск сертификата

Важно

В случае необходимости выпуска собственного серверного сертификата, подписанного Центром сертификации организации, требуется выполнить установку TLS сертификата.

Установка выполняется после успешной установки компонентов Sber Process Mining.

1. Создание ключа и запроса на сертификат одним из двух способов:
   - 1.1. прописав параметры напрямую в команде

```
openssl req -newkey rsa:2048 -nodes -keyout servercert.key -out
cervercert.csr -subj "/CN=FQDN/OU=Organization Unit/O=Organization/C=RU"
```

Заполнить Параметры:

- CN=FQDN
- OU=Organization Unit
- O=Organization
- C=RU

В результате выполнения команды будут созданы файлы ключа и запроса на сертификат:

- servercert.key
- servercert.csr
- 1.2. с использованием файла конфигурации (предварительно создать файл конфигурации)

  ```
  openssl req -out servercert1.csr -newkey rsa:2048 -nodes -keyout
  servercert.key -config certificate.conf
  ```

  Пример содержимого файла конфигурации certificate.conf:

  ```
  [req]
  prompt = no
  default_bits = 2048
  distinguished_name = dn
  req_extensions = req_ext
  [dn]
  countryName = RU
  organizationName = <Имя организации>
  commonName = <CN сертификата>
  organizationalUnitName = <Наименование организации>
  [req_ext]
  subjectAltName =@alt_names
  [alt_names]
  DNS.1 = <FQDN1>
  DNS.2 = <FQDN2>
  ```

  В результате выполнения команды будут созданы файлы ключа и запроса на сертификат:

  - servercert.key
  - servercert.csr

  > **Внимание!**
  >
  > Ключ –nodes не шифрует содержимое файла закрытого ключа.

2. Подписание, выпуск серверного сертификата:

   Выпуск сертификата выполняется Удостоверяющим Центром (далее - УЦ) с использованием ключ и сертификат УЦ.

   Команда для выпуска сертификата с использованием ключа и сертификата УЦ:

   ```
   openssl x509 -req -CA issuer.cer -CAkey issuer.key -extensions server_cert
   -in <hostname>.csr -out <hostname>.cer -days 2048 -CAcreateserial
   ```

   - issuer.cer - сертификат-подпись УЦ
   - issuer.key - ключ УЦ,

- <hostname>.csr запрос на сертификат сервера (п.1).

3. Проверить выпуск сертификата:

```
openssl x509 -noout -text –in <hostname>.cer
```

6.2. Конвертация приватного ключа и сертификата в требуемый формат base64

В случае если при выпуске исходный файл сертификата был сформирован в формате pfx (хранилище p12), то требуется преобразовать его в отдельный файл сертификата и отдельный файл закрытого ключа. И далее сконвертировать полученные ключ и сертификат в формат base64.

1. Посмотреть содержимое хранилища сертификата:

```
openssl pkcs12 -in <имя_файла_сертификата>.pfx -nodes -passin pass:<пароль>
```

Извлечение ключа:

```
openssl pkcs12 –in <имя_файла_хранилища>.pfx -nocerts -nodes -out <имя_файла_ключа1>.key
```

Вводим пароль от хранилища и получаем файл ключа.

Конвертация ключа из PKCS#8 в base64:

```
openssl pkey -inform PEM –in <имя_файла_ключа1>.key -outform PEM –out <имя_файла_ключа2>.key
```

Извлечение сертификата:

```
openssl pkcs12 -in <имя_файла_хранилища>.pfx -clcerts -nokeys –out <имя_файла_сертификата1>.pem
```

Вводим пароль от хранилища и получаем файл ключа.

Конвертация сертификата из PKCS#8 в base64:

```
openssl x509 -inform PEM -in <имя_файла_сертификата1>.pem -outform PEM -out <имя_файла_сертификата2>.pem
```

2. В случае если ключ и сертификат выпущены в формате DER - конвертировать в base64:
Конвертация ключа:

```
openssl pkey -inform der -outform pem -in
<имя_файла_ключа_в_формате_DER>.key -out
<имя_файла_ключа_в_формате_BASE64>.key
```

Конвертация сертификата:

```
openssl x509 -inform der -outform pem -in
<имя_файла_сертификата_в_формате_DER>.cer –out
<имя_файла_сертификата_в_формате_BASE64>.pem
```

6.3. Переименование файлов приватного ключа и сертификата по шаблону именования

Файл серверного сертификата должен называться: `<URL-входа-в-систему>.crt`

Файл ключа должен называться: `<URL-входа-в-систему>.key`

Название должно быть таким же как значение переменной `ui_url` файла конфигурации `config.yml`

Пример: `spm.company.crt` и `spm.company.key`

6.4. Копирование файлов приватного ключа и сертификата в целевое расположение

Скопировать файлы ключа и сертификата в папку: `/opt/spm/certs`

6.5. Копирование файлов корневых и промежуточных сертификатов в целевое расположение

- скопировать корневой сертификат root ca в формате base64 в файл trusted_root_ca.crt в папку: `/opt/spm/certs`
- скопировать промежуточный сертификат sub ca в формате base64 в файл trusted_sub_ca.crt в папку: `/opt/spm/certs`

6.6. Запуск установки/обновления сертификата

Сертификаты обновляются вне зависимости от установки релиза. Если работы по обновлению сертификатов выполняются вместе с установкой или обновлением релиза, то запуск обновления сертификатов необходимо провести после установки/обновления релиза и проверки работоспособности запуском playbook:

```
ansible-playbook -i inventories/local tasks.yml -e "@../config.yml" -e "cert_renew=yes"
```

где `./config.yml` путь до файла конфигурации с которым выполняется деплой или обновление системы.

7. Откат

Откат до предыдущей версии не предусмотрен ввиду отсутствия обратной совместимости после обновления СУБД. Для исправления ошибок выпускается и устанавливается релиз HotFix.

8. Проблемы и пути их устранения (FAQ)

В случае возникновения ошибки развертывания через ansible необходимо проанализировать текст ошибки на предмет причины возникновения.

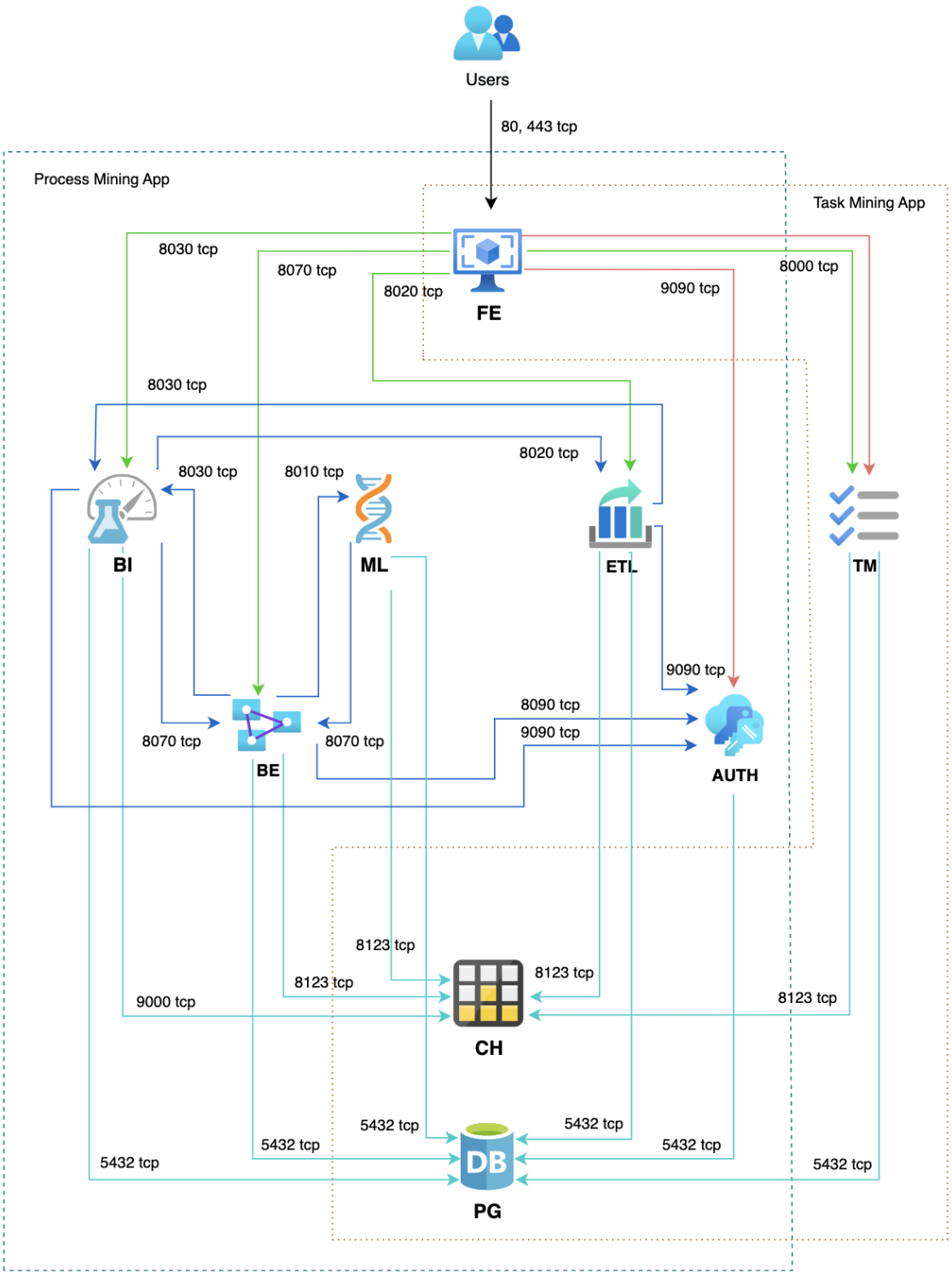
В случае инфраструктурных проблем выполнить поиск данной ошибки по общедоступным источникам, официальной документации на платформу ОС. Если решение найти не удалось - выписать название ansible TASK, текст ошибки и передать сопровождению.

9. Проверка установки. Чек-лист

Сокращения

Условие	Результат
Развертывание выполнено без ошибок	Запущенные playbook отработали без ошибок
Все развернутые сервисы запущены	Результат команды docker ps –a возвращает, что все сервисы находятся в статусе Up
В случае установки по сценарию 1 и 2: Вход на UI успешен	Вход на UI по ссылке `ui_url` пользователем `spm_admin_username` успешен
В случае установки по сценарию 3 переход по ссылке `<ui_url>/events`	Переход по ссылке `<ui_url>/events`, появится форма авторизации *

10. Схема сетевого взаимодействия



→ цветом показаны не аутентифицированные запросы.
→ цветом отмечены аутентифицированные запросы.
→ цветом отражены взаимодействия между компонентами внутри системы.
→ выделены обращения к базам данных.

Описание схемы сетевого взаимодействия:

1. Пользователь из браузера отправляет запрос. Запрос попадает во FE и далее в BE где проверяется наличие корректного JWT-токена. В случае отсутствия JWT-токена пользователь перенаправляется на страницу аутентификации компонента Auth для ввода логина и пароля. Если JWT-токен присутствует, то пользователь авторизован и имеет

возможность выполнять запросы к системе. В противном случае пользователь аутентифицируется в сервисе авторизации Auth и получает JWT-токен.

2. FE отправляет и получает данные от BE и BI, Auth.
3. BE по REST API отправляет запросы в BI и ML, получает от них ответ.
4. BE обращается к БД spm_processmining в СУБД PostgreSQL.
5. BI обращается к БД spm_analytics в СУБД PostgreSQL.
6. Auth обращается к БД spm_auth в СУБД PostgreSQL.
7. BI, ML и BE обращаются к базам данных в СУБД ClickHouse для построения аналитики на основе загруженных пользователем данных.
8. Данные от агентов логирования на рабочих станциях пользователей попадают в сервис логирования и в БД TaskMining.

Сокращения и определения

Сокращение	Описание
FE	Компонент FrontEnd (pm-ui)
BI	Компонент для исследования и визуализации данных (pm-bi)
BE	Компонент BackEnd (pm-core)
ML	Компонент Machine Learning для анализа пользовательских путей (pm-ml)
ETL	Компонент извлечения, преобразования и загрузки данных
Auth	Компонент авторизации пользователей (pm-auth)
TM	Компонент логирования task mining
PG	СУБД Postgresql
CH	СУБД Clickhous
СП	Сервер Приложений
ОС	Операционная система

Продукт «Process Mining от Сбера» включает в себя компоненты FE, BI, BE, ML, ETL, Auth, PG и CH.

Продукт «Task Mining от Сбера» включает в себя компоненты FE, TM, PG и CH.