Process Mining от Сбера

version 2.10.19

Руководство по установке продуктов "Process Mining от Сбера"

April 29, 2026

Содержание

Установка

1. Системные требования

1.1. Минимальные требования к конфигурации серверов

Название сервера	CPU	RAM	HDD
Сервер приложений (СП)	8	16	60
Сервер СУБД PostgreSQL	2	4	100
Сервер СУБД ClickHouse	8	32	100
Управляющий узел ansible (CI / CD)	2	4	60

1.2. Требования к программному обеспечению

	Требуемое СПО	Версия СПО	download
ОС	CentOS, RHEL	7.X и выше	https://www.centos.org/download/ и https://access.redhat.com/downloads
	Debian	10.X и выше	https://www.debian.org/distrib/
	Ubuntu	20.X и выше	https://ubuntu.com/#download-ubuntu
	Astra Linux	1.8 и выше	https://astralinux.ru/os/server-astra/
	РЕД ОС	7.X и выше	https://redos.red-soft.ru/product/downloads/
Сервер приложений (СП)	Docker	27.1.1 и выше	https://docs.docker.com/engine/install/debian/
Сервера СУБД	PostgreSQL server	13.X	Поставляется в дистрибутиве
	ClickHouse server	23.X	Поставляется в дистрибутиве
Сервис кеширования redis	Redis	7.0.12	Поставляется в дистрибутиве
Управляющий узел ansible (CI / CD)	Ansible	2.9 и выше	https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html

1.3. Требование к сетевым доступам

Источник	Назначение	Порты
Рабочая станция пользователя	Сервер приложений	tcp/80, tcp/443

Источник	Назначение	Порты
Сервер приложений	Сервер СУБД Postgresql	tcp/5432
Сервер приложений	Сервер СУБД ClickHouse	tcp/8123, tcp/9000
Управляющий сервер	Сервер приложений, Сервер СУБД	tcp/22
Рабочая станция администратора	Сервер приложений, Сервер СУБД, Управляющий сервер	tcp/22
Управляющий сервер	Сервер приложений	tcp/80, tcp/443, tcp/5432, tcp/8123, tcp/6379, tcp/8010, tcp/8020, tcp/8030, tcp/8040, tcp/8060, tcp/8070, tcp/8090, tcp/8000, tcp/9030

Сетевая схема описана в Приложении 1 к данной инструкции.

1.4. Требование к DNS записи

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

1.5. Прикладное ПО состоит из сервисов

Имя сервиса	Описание
pm-ui	Компонент FrontEnd (IAM)
pm-bi	Компонент для исследования и визуализации данных (BI)
pm-ml	Компонент Machine Learning для анализа пользовательских путей (ML)
pm-etl	Компонент ETL для извлечения, анализа и загрузки данных (ETL), Airflow webserver
pm-flow	Планировщик, Airflow scheduler
pm-auth	Компонент авторизации пользователей (Auth)
pm-tm	Компонент сервиса логирования task mining (TM)
redis	Компонент кеширования данных
pm-docs	Компонент документации (DOCS)

2. Установка программного обеспечения

2.1. Состав поставки

Поставка представляет собой набор архивов для каждого сервиса и СУБД, скрипты развертывания. Имена архивов формируются по правилам.

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

2.2. Подготовка окружения

2.2.1. Настройка серверов СУБД и сервера приложений:

На серверах СУБД и сервере приложений должен быть установлен и настроен Docker версии 27.X и выше. Инструкция по возможному способу установки Docker представлена ниже, также допускается установка иными способами, например, через менеджеры пакетов ОС формата rpm и deb. Если Docker требуемой версии уже установлен, то необходимо проверить соответствие пункту “2.2.1.3” инструкции ниже.

2.2.1.1. Добавить репо docker (подробное описание есть в официальной инструкции, ссылка: https://docs.docker.com/

2.2.1.2. Выполнить установку пакетов (docker и docker-compose) командой:

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

2.2.1.3. Если установка производится не под пользователем root, то перед началом необходимо добавить этого пользователя в группу docker, выполнив команду:

`usermod -aG docker <user_name>`, где <user_name> имя пользователя, под которым будет выполнена установка. Критерием успешного выполнения шага является возможность пользователя <user_name> выполнять команды docker без sudo, например, команда `docker ps` должна успешно выполниться от имени пользователя <user_name> без инструкции sudo.

2.2.2. Настройка управляющего сервера (CI/CD):

2.2.2.1. Должен быть установлен Ansible версии 2.9 или выше. Ниже приведен пример установки Ansible с помощью пакетного менеджера pip, также допускается выполнить установку используя менеджеры пакетов ОС формата rpm и deb:

- Создать виртуальное окружение python venv, выполнив команду: `python -m venv venv`
- Активировать venv, выполнив команду: `source venv/bin/activate`
- Установить пакет ansible через пакетный менеджер python (pip), выполнив команду: `pip install ansible`

2.2.2.2. С управляющего сервера должен быть настроен доступ по протоколу ssh к серверу приложений и серверам СУБД. Настройка подключения возможна как по паролю, так и по ssh ключу. Рекомендуемый способ – по ключу.

Критерием корректной настройки является возможность с управляющего сервера подключиться к серверу приложений и серверам СУБД, выполнив команду:

`ssh <user_name>@<server_ip>`, где <user_name> пользователь от имени которого идет подключение, <server_ip> IP адрес сервера к которому идет подключение.

Описание процесса настройки подключения к серверу по ssh протоколу:

- Сгенерировать 2048-битную пару RSA ключей, выполнив команду: `ssh-keygen –b 2048`. Пара ключей (публичный и приватный) по умолчанию попадают в каталог ~/.ssh/ и называются id_rsa.pub и id_rsa соответственно.
- Далее необходимо скопировать публичный ключ id_rsa.pub на сервер приложений и сервера СУБД. Это можно сделать, вручную скопировав публичный ключ в файл ~/.ssh/ authorized_keys, либо выполнив команду: `ssh-copy-id -i ~/.ssh/id_rsa.pub <user_name>@<server_ip>`, где <user_name> пользователь от имени которого идет подключение, <server_ip> IP адрес сервера к которому идет подключение.
- На управляющем сервере должен быть установлен пакет sshpass. Установка выполняется командой: `yum install sshpass` или `apt install sshpass`, выбор команды зависит от пакетного менеджера используемой ОС.

2.2.2.3. Необходимо получить файлы лицензий license.json и license.json.sig.

2.2.2.4. Необходимо получить архив с дистрибутивом, после чего скопировать его и распаковать на управляющем сервере.

2.3. Установка приложения

2.2.3.1. На управляющем сервере скопировать файлы образов с расширением “.tgz” в удобный каталог, по умолчанию предполагается использование каталога /opt/images. Каталог задаётся параметром source_images_storage_folder в файле config.yml (описание параметров представлено в таблице 2 Описание параметров файла config.yml).

2.2.3.2. На управляющем сервере среди файлов из распакованного ранее дистрибутива найти файл deploy-<version>.tar, где <version> - это версия поставки, например, 2.9.6, после чего распаковать этот файл, выполнив команду: tar –xvf deploy-<version>.tar. После распаковки файла deploy-<version>.tar появится каталог deploy-<version> с файлами деплоя приложения.

2.2.3.3. В каталоге deploy-<version> найти файл config.yml и скопировать его в каталог /opt, выполнив команду: cp config.yml /opt/config.yml. **Этот файл является шаблоном постоянного конфигурационного файла, заполняется единожды при первой установке и в дальнейшем используется для установки обновлений, поэтому важно не потерять его.**

2.2.3.4. Заполнить постоянный конфигурационный файл /opt/config.yml. Описание конфигурационных параметров представлено в таблице 2 Описание параметров файла config.yml и дополнительно в самом файле config.yml в виде комментариев. При заполнении паролей и логинов **важно не использовать спец символы из таблицы 1 Спец символы**, т.к. bash может воспринять их как часть инструкций, что в свою очередь приведет к ошибкам при установке. В качестве спец символа рекомендуем использовать восклицательный знак “!”, с ним проблем не возникнет:

Таблица 1. Спец-символы

~	]	(	„>“
`	{	)	/
„#“	}	\	?
$	;	|	@
&	„	[	“
„*“	<	^	„+“

Описание параметров в файле config.yml:

Таблица 2. Описание параметров файла config.yml

№	Параметр	Значение	Описание
1	app_host	“<IP>”	ip-адрес или hostname хоста компонентов Приложения
2	db_host	“<IP>”	ip-адрес или hostname хоста компонентов СУБД Postgresql, используется для хранения метаданных
3	dwh_host	“<IP>”	ip-адрес хоста компонентов СУБД ClickHouse, используется для хранения аналитических данных
4	ui_url	“<FQDN>”	точка входа для пользователей, указать FQDN
5	ssh_user	“<username>”	Имя пользователя под которым происходит подключение к серверам для автоматизированной установки компонент SPM
6	ssh_password	“<password>”	Пароль для подключение по ssh к серверам для автоматизированной установки компонент SPM (не требуется заполнять при подключении через ssh ключ)

№	Параметр	Значение	Описание
7	use_local_deploy	“<no>”	Значение по умолчанию “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 “yes”
8	source_images_on_app_server	“<no>”	Значение по умолчанию “no”. В случае расположения файлов образов *.tgz на сервере приложений устанавливаем значение параметра “yes”. По умолчанию предполагается расположения файлов образов на управляющем сервере
9	source_images_storage_folder	“/opt/images”	Абсолютный путь до каталога с файлами образов *.tgz. По умолчанию предполагается каталог /opt/images. На указанную директорию должны быть предоставлены права на создание папок и файлов для пользователя, подключающегося с управляющего сервера
10	app_config_folder	“/opt/spm”	Абсолютный путь до каталога в котором будут хранится конфигурационные файлы сервисов. Каталог должен быть как на сервере приложений, так и на серверах баз данных. По умолчанию предполагается каталог /opt/spm. На указанную директорию должны быть предоставлены права на создание папок и файлов для пользователя, подключающегося с управляющего сервера
11	features	“pm”	Параметр задает сценарий установки. Допустимые значения: pm, tm, ent. Значение по умолчанию “pm”. Указываем значение ent если устанавливаем продукты process mining и task mining по сценарию №1; pm — только process mining сценарий №2; tm — только task mining сценарий №3
12	spm_admin_username	“<username>”	Имя бизнес пользователя, который будет автоматически создан в системе. Под ним будет возможность войти в систему сразу после установки. Примечание: при установке по сценарию №3 данное поле не заполняется
13	spm_admin_password	“<password>”	Пароль бизнес пользователя для входа в систему. Требования: минимум 1 заглавная буква, 1 цифра, 1 спецсимвол (без символов из Таблицы 1), длина ≥8. Пример: Admin123! Примечание: при установке по сценарию №3 — не заполняется
14	auth_admin_username	“<username>”	Имя пользователя, который будет администратором Keycloak. Под ним можно создавать пользователей, управлять доступом, сбрасывать пароли. Примечание: при установке по сценарию №3 — не заполняется

№	Параметр	Значение	Описание
15	auth_admin_password	“<password>”	Пароль администратора Keycloak. Требования: минимум 1 заглавная буква, 1 цифра, 1 спецсимвол, длина ≥12. Пример: Admin123!spm! Примечание: при установке по сценарию №3 — не заполняется
16	infra_admin_username	“<username>”	Имя пользователя под которым будет доступ к мониторингу
17	infra_admin_password	“<password>”	Пароль пользователя для входа в систему мониторинга. Требования: как у spm_admin_password. Пример: Admin123!
18	postgres_db_password	“<password>”	Пароль для подключения к БД PostgreSQL. При использовании внешних СУБД — указать в конфиге и предварительно создать БД (см. п.1)
19	clickhouse_db_password	“<password>”	Пароль для подключения к БД ClickHouse. При использовании внешних СУБД — указать в конфиге и предварительно создать БД
20	analytics_integration_password	“<password>”	Пароль технического пользователя интеграции компонентов. Примечание: при установке по сценарию №3 — не заполняется
21	redis_integration_password	“<password>”	Данные для подключения к сервису кеширования Redis. Примечание: при установке по сценарию №3 — не заполняется
22	ssl_certificate_password	“<password>”	Пароль генерируемого сертификата для Auth
23	use_external_db	“no”	Параметр регулирует использование внешних баз данных. По умолчанию “no”. Если используются собственные СУБД — установить “yes”
24	ml_host	“<IP>”	IP-адрес хоста компонента pm-ml. По умолчанию совпадает с app_host. При необходимости вынести ML на отдельный сервер — указать его IP
25	pg_port	“5432”	Порт для подключения к PostgreSQL. По умолчанию 5432, может быть изменён
26	ch_port_tcp	“9000”	Порт для подключения к ClickHouse. По умолчанию 9000, может быть изменён
27	title	“”	Сообщение на странице входа. По умолчанию “On-Premise”
28	timezone	"Europe/Moscow"	Временная зона, по умолчанию «Europe/Moscow»
29	calendar	“Ru”	Календарь: “Ru”, “By”, “Kg”. По умолчанию “Ru”
30	use_preview	“no”	Включает/выключает preview дашбордов. Возможные значения: “no”, “yes”. По умолчанию “no”
31	create_pg_db	“yes”	Автоматическое создание БД и схем в PostgreSQL. По умолчанию “yes”. Не менять при использовании СУБД из дистрибутива
32	processmining_db	“”	Заполняется только при use_external_db: “yes”. База данных для схемы processmining

№	Параметр	Значение	Описание
33	db_processmining_username	“”	Имя пользователя PostgreSQL для схемы processmining (только при use_external_db: “yes”)
34	processmining_db_passwd	“”	Пароль пользователя для схемы processmining (только при use_external_db: “yes”)
35	analytics_db	“”	База данных для схемы analytics (при use_external_db: “yes”)
36	db_analytics_username	“”	Имя пользователя для схемы analytics (при use_external_db: “yes”)
37	analytics_db_passwd	“”	Пароль для схемы analytics (при use_external_db: “yes”)
38	etl_db	“”	База данных для схемы etl (при use_external_db: “yes”)
39	db_etl_username	“”	Имя пользователя для схемы etl (при use_external_db: “yes”)
40	etl_db_passwd	“”	Пароль для схемы etl (при use_external_db: “yes”)
41	ml_db	“”	База данных для схемы ml (при use_external_db: “yes”)
42	db_ml_username	“”	Имя пользователя для схемы ml (при use_external_db: “yes”)
43	ml_db_passwd	“”	Пароль для схемы ml (при use_external_db: “yes”)
44	ai_db	“”	База данных для схемы ai (при use_external_db: “yes”)
45	db_ai_username	“”	Имя пользователя для схемы ai (при use_external_db: “yes”)
46	ai_db_passwd	“”	Пароль для схемы ai (при use_external_db: “yes”)
47	auth_db	“”	База данных для схемы auth (при use_external_db: “yes”)
48	db_auth_username	“”	Имя пользователя для схемы auth (при use_external_db: “yes”)
49	auth_db_passwd	“”	Пароль для схемы auth (при use_external_db: “yes”)
50	taskmining_db	“”	База данных для схемы taskmining (при use_external_db: “yes”)
51	db_taskmining_username	“”	Имя пользователя для схемы taskmining (при use_external_db: “yes”)
52	taskmining_db_passwd	“”	Пароль для схемы taskmining (при use_external_db: “yes”)
53	pre_existing_db_name	“”	Имя существующей БД в PostgreSQL, используемой приложением (при use_external_db: “yes”)
54	external_db_username	“”	Пользователь PostgreSQL с правами на инициализацию БД (при use_external_db: “yes”)

№	Параметр	Значение	Описание
55	use_vault	“no”	Использование Vault для хранения секретов. Поддерживается только AppRole. Возможные значения: “yes”, “no”
56	vault_url	«https://example.com:8200»	Актуальная ссылка на Vault
57	vault_secrets_path	"secret_name"	Название секрета в Vault, где хранятся пароли и чувствительные данные
58	vault_role_id	“”	Требуется указать актуальный role_id
59	vault_secret_id	“”	Требуется указать актуальный secret_id
60	vault_storage_path	"storage/path"	Путь до нужного секрета в Vault
61	ansible_hashi_vault_namespace	“”	Vault namespace. Если используется стандартный — оставить пустым

2.2.3.5. На сервере Приложений создать каталог <app_config_folder>/license и скопировать в него файлы лицензии: license.json и license.json.sig

2.2.3.6. При необходимости шифрования паролей использовать ansible-vault. Пример шифрования пароля с использованием ansible-vault: `ansible-vault encrypt_string '<password>' –-name '<name>'`, где <password> это пароль, который нужно зашифровать, <name> это имя параметра.

2.2.3.7. Файл `deploy-<version>/inventories/local/group_vars/all/env.yml`, где <version> - это версия поставки, например, 2.9.6, 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.

2.2.3.8. В первую очередь запустить развертывание сервисов СУБД PostgreSQL и ClickHouse, выполнив playbook setup.yml командой (запуск команды предполагается из каталога deploy-<version>):
`ansible-playbook -i inventories/local setup.yml -e "@/opt/config.yml"`

2.2.3.9. После успешного выполнения развертывания сервисов СУБД запустить установку компонентов приложения, запустив playbook deploy.yml командой (запуск команды предполагается из каталога deploy-<version>):
`ansible-playbook -i inventories/local deploy.yml -e "@/opt/config.yml"`

2.2.3.10. В случае, если используется шифрование паролей при помощи ansible-vault добавить к команде запуска развертывания параметр: `--ask-vault-pass`

3. Обновление

3.1. Скачать дистрибутив по ссылке из поставки, скопировать их на управляющий сервер и распаковать.

3.2. Найти и распаковать архив deploy-<version>.tar на управляющем сервере (CI/CD) командой: `tar –xvf deploy-<version>.tar`, где <version> - это версия поставки, например, 2.9.6.

3.3. Запустить установку компонентов приложения, запустив playbook deploy.yml командой (запуск команды предполагается из каталога deploy-<version>):
`ansible-playbook -i inventories/local deploy.yml -e "@/opt/config.yml"`

3.4. В случае обновления до версии HotFix, когда в состав поставки входит только 1 компонент запустить playbook развертывания обновляемого компонента поставки (запуск команды предполагается из каталога deploy-<version>):
`ansible-playbook -i inventories/local <имя_компонента>.yml -e "@/opt/config.yml»`, где <имя_компонента> это имя контейнера, например, pm-ui, pm-bi и пр.

4. Удаление

4.1. Удаление компонентов продукта с сервера приложений выполняется командой:

```
docker stop $(docker ps -a -q) || true && \
docker rm $(docker ps -a -q) || true && \
docker volume rm $(docker volume ls -q) || true && \
docker rmi $(docker images -aq) || true
```

> **Внимание!**
>
> Команда выше остановит и удалит все запушенные контейнеры, удалит все docker volumes и все образы на сервере приложений! Если на сервере приложений существуют контейнеры помимо компонент продукта, то данную команду запускать не следует, в противном случае они также будут удалены. В этом случае следует:

4.1.1. заменить инструкцию $(docker ps -a -q) на список контейнеров, например, pm-ui pm-bi pm-etl и т.д. (список контейнеров можно получить, выполнив команду: `docker ps -a`);

4.1.2. инструкцию $(docker volume ls -q) заменить на список конкретных вольюмов, например, etldata, authdata и т.д. (список волюмов можно получить, выполнив команду: `docker volume ls`);

4.1.3. инструкцию $(docker images -aq) на список конкретных образов, например, pm-ui:2.9.6 pm-bi:2.9.6 и т.д. (список образов можно получить, выполнив команду: `docker images`).

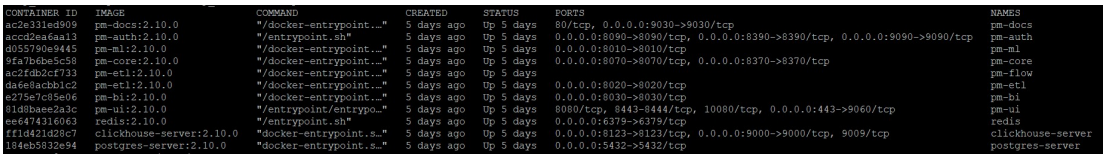
4.2. Удаление компонентов продукта с серверов баз данных производится аналогично удалению компонентов с сервера приложений, описанному в предыдущем пункте 4.1.

> **Внимание!**
>
> Отчистка вольюмов баз данных clickdata и pgdata приведет к безвозвратной потери данных!

5. Проверка работоспособности

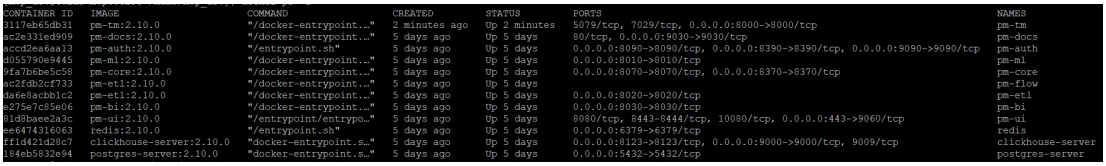
5.1. Проверить что все компоненты развернуты и запущены в docker контейнерах: docker ps –a

В выводе должны быть все компоненты в статусе Up:



```
CONTAINER ID   IMAGE                          COMMAND                  CREATED       STATUS       PORTS                                                                                  NAMES
ac2e331ed909   pm-docs:2.10.0                 "/docker-entrypoint.…"   5 days ago    Up 5 days    80/tcp, 0.0.0.0:9030->9030/tcp                                                         pm-docs
accd2ea6aa13   pm-auth:2.10.0                 "/entrypoint.sh"         5 days ago    Up 5 days    0.0.0.0:8090->8090/tcp, 0.0.0.0:8390->8390/tcp, 0.0.0.0:9090->9090/tcp                 pm-auth
d055790e9445   pm-ml:2.10.0                   "/docker-entrypoint.…"   5 days ago    Up 5 days    0.0.0.0:8010->8010/tcp                                                                 pm-ml
9fa7b6be5c58   pm-core:2.10.0                 "/docker-entrypoint.…"   5 days ago    Up 5 days    0.0.0.0:8070->8070/tcp, 0.0.0.0:8370->8370/tcp                                         pm-core
ac2fdb2cf733   pm-etl:2.10.0                  "/docker-entrypoint.…"   5 days ago    Up 5 days                                                                                           pm-flow
da6e8acbb1c2   pm-etl:2.10.0                  "/docker-entrypoint.…"   5 days ago    Up 5 days    0.0.0.0:8020->8020/tcp                                                                 pm-etl
e275e7c85e06   pm-bi:2.10.0                   "/docker-entrypoint.…"   5 days ago    Up 5 days    0.0.0.0:8030->8030/tcp                                                                 pm-bi
81d8baee2a3c   pm-ui:2.10.0                   "/entrypoint/entrypo…"   5 days ago    Up 5 days    8080/tcp, 8443-8444/tcp, 10080/tcp, 0.0.0.0:443->9060/tcp                              pm-ui
ee6474316063   redis:2.10.0                   "/entrypoint.sh"         5 days ago    Up 5 days    0.0.0.0:6379->6379/tcp                                                                 redis
ff1d421d28c7   clickhouse-server:2.10.0       "docker-entrypoint.s…"   5 days ago    Up 5 days    0.0.0.0:8123->8123/tcp, 0.0.0.0:9000->9000/tcp, 9009/tcp                               clickhouse-server
184eb5832e94   postgres-server:2.10.0         "docker-entrypoint.s…"   5 days ago    Up 5 days    0.0.0.0:5432->5432/tcp                                                                 postgres-server
```

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



```
CONTAINER ID   IMAGE                          COMMAND                  CREATED         STATUS         PORTS                                                                                  NAMES
2117eb65db31   pm-tm:2.10.0                   "/docker-entrypoint.…"   2 minutes ago   Up 2 minutes   5079/tcp, 7029/tcp, 0.0.0.0:8000->8000/tcp                                             pm-tm
ac2e331ed909   pm-docs:2.10.0                 "/docker-entrypoint.…"   5 days ago      Up 5 days      80/tcp, 0.0.0.0:9030->9030/tcp                                                         pm-docs
accd2ea6aa13   pm-auth:2.10.0                 "/entrypoint.sh"         5 days ago      Up 5 days      0.0.0.0:8090->8090/tcp, 0.0.0.0:8390->8390/tcp, 0.0.0.0:9090->9090/tcp                 pm-auth
d055790e9445   pm-ml:2.10.0                   "/docker-entrypoint.…"   5 days ago      Up 5 days      0.0.0.0:8010->8010/tcp                                                                 pm-ml
9fa7b6be5c58   pm-core:2.10.0                 "/docker-entrypoint.…"   5 days ago      Up 5 days      0.0.0.0:8070->8070/tcp, 0.0.0.0:8370->8370/tcp                                         pm-core
ac2fdb2cf733   pm-etl:2.10.0                  "/docker-entrypoint.…"   5 days ago      Up 5 days                                                                                             pm-flow
da6e8acbb1c2   pm-etl:2.10.0                  "/docker-entrypoint.…"   5 days ago      Up 5 days      0.0.0.0:8020->8020/tcp                                                                 pm-etl
e275e7c85e06   pm-bi:2.10.0                   "/docker-entrypoint.…"   5 days ago      Up 5 days      0.0.0.0:8030->8030/tcp                                                                 pm-bi
81d8baee2a3c   pm-ui:2.10.0                   "/entrypoint/entrypo…"   5 days ago      Up 5 days      8080/tcp, 8443-8444/tcp, 10080/tcp, 0.0.0.0:443->9060/tcp                              pm-ui
ee6474316063   redis:2.10.0                   "/entrypoint.sh"         5 days ago      Up 5 days      0.0.0.0:6379->6379/tcp                                                                 redis
ff1d421d28c7   clickhouse-server:2.10.0       "docker-entrypoint.s…"   5 days ago      Up 5 days      0.0.0.0:8123->8123/tcp, 0.0.0.0:9000->9000/tcp, 9009/tcp                               clickhouse-server
184eb5832e94   postgres-server:2.10.0         "docker-entrypoint.s…"   5 days ago      Up 5 days      0.0.0.0:5432->5432/tcp                                                                 postgres-server
```

5.2. 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>`. Например, docker logs –tail 1000 pm-ui

5.3. Проверить порт в случае возникновения проблем командой telnet или аналогичной: `telnet <server_ip> <port>`, где <server_ip> IP адрес серврера на котором работает контейнер, а <port> порт контейнера.

Таблица соответствия портов и сервисов:

№	Сервис (контейнер)	Порты
1	IAM (pm-ui)	80, 443
2	BI (pm-bi)	8030
3	ML (pm-ml)	8010
4	ETL (pm-etl)	8020
5	Auth (pm-auth)	9090, 8090
6	Tm (pm-tm)	8000
7	PG (postgres-server)	5432
8	CH (clickhouse-server)	9000, 8123
9	DOCS (pm-docs)	9030

5.4. Для Сценария №1 и Сценария №2 установки:

Проверка входа в UI интерфейс по ссылке https://ui_url под пользователем `spm_admin_username` с паролем `spm_admin_password`.

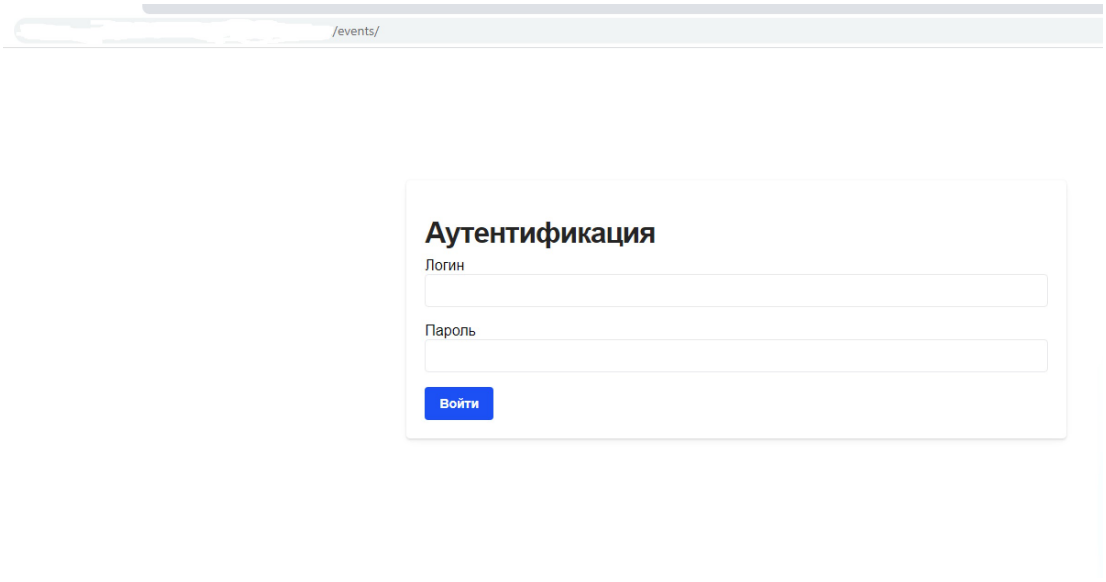
Значения параметров `ui_url`, `spm_admin_username` и `spm_admin_password` были ранее заданы в файле config.yml.

Критерием успешности проверки является авторизация пользователя `spm_admin_username` в системе.

5.5. Для Сценария №3 установки:

Перейти по ссылке: https://<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

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.

6.1.1. Создание ключа и запроса на сертификат одним из двух способов:

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
```

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
```

- Ключ –nodes не шифрует содержимое файла закрытого ключа.

6.2. Подписание, выпуск серверного сертификата

Выпуск сертификата выполняется Удостоверяющим Центром (далее - УЦ) с использованием ключа 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.6.1).

6.3. Проверить выпуск сертификата

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

6.4. Конвертация приватного ключа и сертификата в требуемый формат base 64

6.4.1. 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. Последовательность действий:

1. Посмотреть содержимое хранилища сертификата:

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

2. Извлечение ключа:

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

Вводим пароль от хранилища и получаем файл ключа.

4. Конвертация ключа из PKCS#8 в base64:

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

5. Извлечение сертификата:

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

Вводим пароль от хранилища и получаем файл ключа.

6. Конвертация сертификата из PKCS#8 в base64:

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

6.4.2. В случае если ключ и сертификат выпущены в формате DER - конвертировать в base64:

1. Конвертация ключа:

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

2. Конвертация сертификата:

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

6.5. 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 в файле конфигурации **config.yml**

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

6.6. Копирование файлов приватного ключа и сертификата в целевое расположение

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

6.7. Копирование файлов корневых и промежуточных сертификатов в целевое расположение

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

6.8. 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 (запуск команды предполагается из каталога deploy-<version>):

ansible-playbook -i inventories/local tasks.yml -e "@/opt/config.yml" -e "cert_renew=yes", где ../config.yml путь до файла конфигурации с которым выполняется деплой или обновление системы.

7. Настройка интеграции с Vault

Интеграция с HashiCorp Vault является опциональной. Поддерживается вариант интеграции только через AppRole. 7.1. Список параметров, которые могут быть получены из Vault и должны быть предварительно туда занесены (описание параметров в таблице ниже):

analytics_integration_password
auth_admin_password
clickhouse_db_password
infra_admin_password
postgres_db_password
redis_integration_password
spm_admin_password
ssl_certificate_password

В случае использования Vault данные параметры в файле config.yml не заполняются!

7.2. В файле config.yml необходимо заполнить следующие параметры занесены (описание параметров в таблице ниже):

use_vault
vault_url
vault_secrets_path
vault_role_id
vault_secret_id
vault_storage_path
ansible_hashi_vault_namespace

7.3. Необходимо на сервере приложения добавить актуальную цепочку сертификатов в каталог <app_config_folder>/certs (по умолчанию /opt/spm/certs) для коммуникации с сервером vault. Важно, чтобы расширение файла было .crt. Без сертификатов будет ssl error при попытке соединения с сервером Vault.

7.4. Далее необходимо запустить установку приложения той же командой, что и установку и обновление приложения пункт 2.2.3.9

8. Откат

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

9. Проблемы и пути их устранения (FAQ)

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

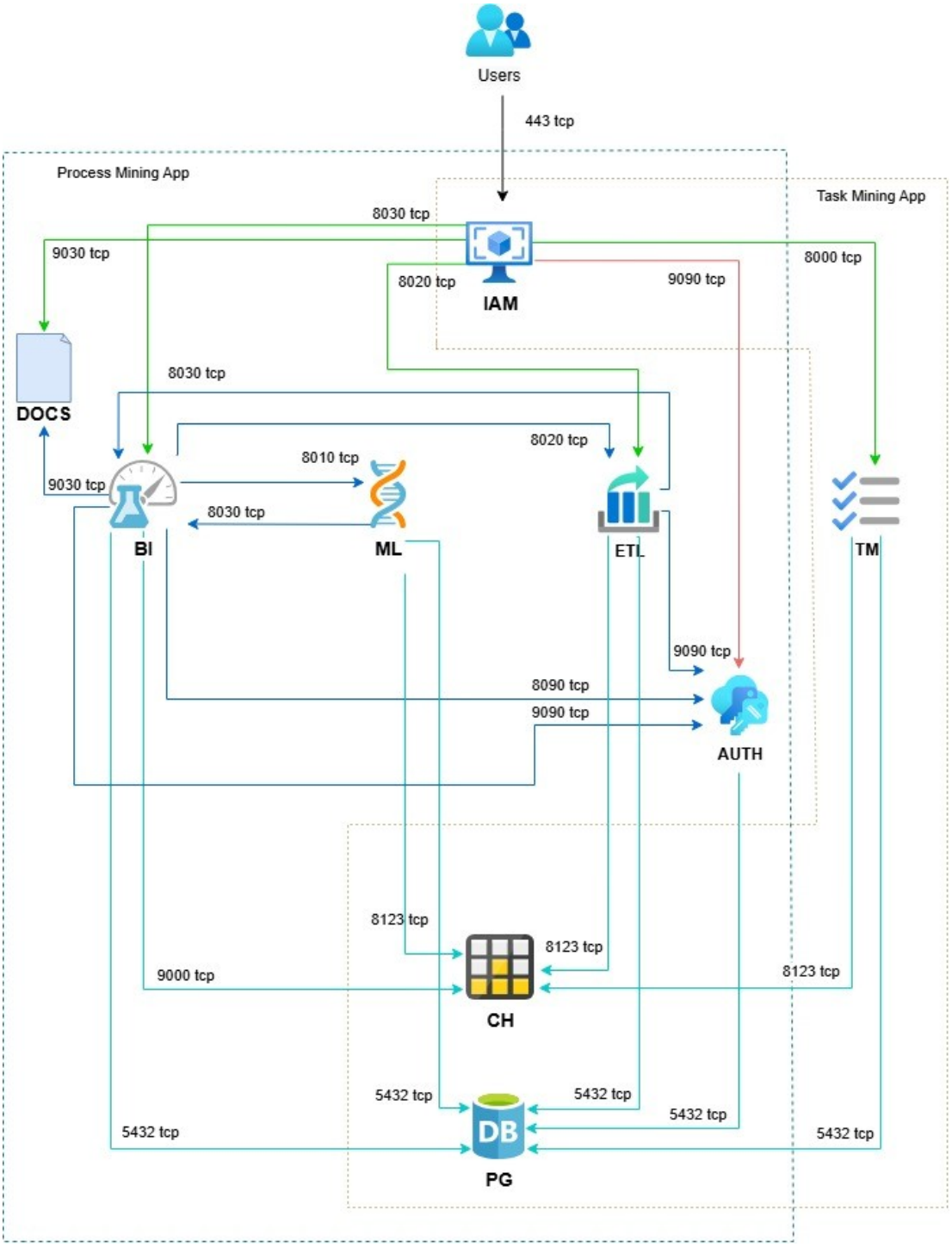
В случае инфраструктурных проблем выполнить поиск данной ошибки по общедоступным источникам, официальной документации на платформу ОС. Если решение найти не удалось – выписать название ansible TASK, текст ошибки и передать сопровождению.

10. Проверка установки. Чек-лист

Критерий проверки	Ожидаемый результат
Развертывание выполнено без ошибок	Запущенные playbook отработали без ошибок

Критерий проверки	Ожидаемый результат
Все развернутые сервисы запущены	Результат команды `docker ps -a` возвращает, что все сервисы находятся в статусе Up
В случае установки по сценарию №1 и сценарию №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 *

Приложение 1:



Легенда:

Описание схемы сетевого взаимодействия:

1. Пользователь из браузера отправляет запрос. Запрос попадает во IAM и 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. IAM отправляет и получает данные от BI и Auth.
3. BI обращается к БД spm_analytics в СУБД PostgreSQL.
4. Auth обращается к БД spm_auth в СУБД PostgreSQL.
5. BI и ML обращаются к базам данных в СУБД ClickHouse для построения аналитики на основе загруженных пользователем данных.
6. Данные от агентов логирования на рабочих станциях пользователей попадают в сервис логирования и в БД TaskMining.

Сокращения и определения

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

Продукт Process Mining включает в себя компоненты IAM, BI, ML, ETL, Auth, DOCS, PG и CH.

Продукт Task Mining включает в себя компоненты IAM, TM, PG и CH.

Возможны различные сценарии установки продуктов:

Сценарий №1: Установка продуктов Process Mining и Task Mining.

Устанавливаются компоненты: IAM, BI, ML, ETL, TM, Auth, DOCS, PG и CH.

Сценарий №2: Установка продукта Process Mining.

Устанавливаются компоненты: IAM, BI, ML, ETL, Auth, DOCS, PG и CH.

Сценарий №3: Установка продукта Task Mining.

Устанавливаются компоненты: IAM, TM, PG и CH.

Для развертывания компонентов продукта Sber Process Mining потребуется минимально четыре сервера, перечисленных в таблице раздела 1.2. 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 PostgreSQL (PG), на сервере СУБД ClickHouse будет расположен 1 экземпляр СУБД ClickHouse (CH).

PG – компонент СУБД PostgreSQL, содержит следующие базы данных и схемы (создаются автоматически):

- 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 следующие БД с пользовательскими данными для работы сервисов BI, ML, ETL, TM (создаются автоматически):

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