Аннотация

Настоящее руководство содержит информацию, необходимую для администрирования программного комплекса выявления и блокирования несанкционированного доступа к информационным системам на основе анализа конфликта полномочий с использованием технологий искусственного интеллекта (ПК SoD).

Используемые в документе термины и определения и сокращения описаны в соответствующих разделах.

Оглавление

1. Установка, настройка и обслуживание ПК SoD

Установка и настройка ПК SoD и связанного с ней программного обеспечения описана в документе Руководство по установке и настройке ПК SoD.

Обслуживание ПК SoD выполняют сотрудники, обязанности которых распределены по следующим административным ролям:

1. Администратор инфраструктурных сервисов.
2. Администратор серверов контейнеризации.
3. Администраторы подсистем.

Администратор инфраструктурных сервисов

Администратор инфраструктурных сервисов реализует следующие функции:

- инсталляция и настройка сервера приложений и инфраструктурных сервисов;
- настройка системных компонентов сервера приложений;
- первичные настройки системных компонентов выполняются в процессе инсталляции, в соответствии с действиями, описанными для каждого компонента в документе Руководство по установке и настройке ПК SoD;
- обновление компонентов программного обеспечения;
- контроль работоспособности серверов и сервисов, восстановление работы после технических отказов;
- контроль функционирования серверов и сервисов, устранение неполадок.

Администратор серверов контейнеризации

Администратор серверов контейнеризации реализует следующие функции:

- инсталляция и настройка серверов контейнеризации;
- контроль работоспособности контейнеризации, восстановление работы после технических отказов. Восстановление работоспособности выполняется автоматически. В случае невозможности выполнения восстановления работоспособности средствами системы администратору следует обратиться к разработчикам;
- контроль функционирования контейнеризации, устранение неполадок. Устранение неполадок выполняется либо администратором, либо сотрудниками компании-разработчика после соответствующего обращения.

Администратор подсистемы ETL

Администратор подсистемы ETL реализует следующие функции:

- настройка выполнения потоков загрузки данных средствами планировщика ETL-потоков;
- выполнение настроек подключения к смежным системам;
- создание новых/корректировка существующих ETL-потоков: изменение и доработка сценариев ETL-потоков (чтение данных из источника, выполнение преобразований, трансформаций, маршрутизации, загрузка в потребитель данных);
- просмотр системных журналов;
- мониторинг выполнения ETL-потоков и передаваемых сообщений: статистика, статусы выполнения потоков, возникающие ошибки;
- мониторинг производительности подсистемы ETL.

1.1. Инсталляционный комплект

Инсталляционный комплект представляет собой набор файлов, определяемый условиями реализации ИС в рамках конкретного проекта. Состав набора определяется условиями технического задания проекта (включая требования к ИС и к инфраструктуре заказчика).

Инсталляционный комплект используется для установки и первичной настройки ИС сотрудниками компании-разработчика.

1.2. Объекты «Планета»

Для реализации заложенной в ПК SoD функциональности необходимо использование следующих программных продуктов «Планета»:

- «Планета. НСИ» — реализует функциональность управления справочниками и реестрами;
- «Планета. Доступ» — реализует функциональность управления авторизации, аутентификации и управления ролевой моделью;
- «Планета. Аналитика» — реализует функциональность конфигурирования, формирования и распространения отчетности и информационных панелей;
- «Планета. Процессы» — реализует функциональность конфигурирования и последующего выполнения процессов внесения изменений в основные данные;
- «Планета. Интеграция» — реализует функциональность поддержки процессов загрузки, консолидации и обработки больших объёмов данных в наборы данных для последующего анализа (опционально, в зависимости от конкретного проекта).

2. Стороннее программное обеспечение

Для установки, развёртывания и применения ПК SoD необходимо использовать программное обеспечение сторонних производителей.

Состав стороннего ПО и реализуемая им функциональность зависят от конкретного проекта.

Пример набора стороннего ПО, обеспечивающего работу ПК SoD в составе ИС на основе «Планета. НСИ»:

- средство контейнеризации и оркестрации — Kubernetes (Deckhouse);
- приложение балансировки нагрузки — Kubernetes;
- файловый сервер — распределенная файловая система NFS;
- сервер облачного хранилища — MinIO;
- система управления базами данных — Postgres;
- платформа управления репозиториями — Nexus.

Сведения об установке и настройке стороннего ПО содержатся в документе Руководство по установке и настройке ПК SoD.

2.1. Обновление компонентов стороннего ПО

Обновление операционной системы и стороннего ПО, обеспечивающих функционирование ИС выполняется в соответствии с политиками производителей конкретных ОС и стороннего ПО.

3. Обновление объектов «Планета»

3.1. Для минорных версий, работающих в Docker Compose

Для обновления минорных версий (в рамках одной мажорной версии), работающих в Docker Compose, необходимо выполнить следующие действия:

1. Сохранить файл настроек и выполнить бэкап образов и конфигурации.
2. Выполнить бэкап всех томов и файла .env.
3. Получить обновленные файлы docker-compose.yml и .env.
4. Заменить на новые следующие файлы:
   - один или несколько docker-compose.yml;
   - файлу .env требуется ручная синхронизация. Синхронизировать существующий .env-файл с обновленным. Если набор переменных в новой версии .env изменился, то необходимо задать значения для новых переменных.
5. Обновить образы. Для этого необходимо:
   - авторизоваться в docker registry:
   `docker login --user $USER --password $PASS $URL`
   - выполнить команду:
   `docker-compose pull`
6. Выполнить команду:
   `docker-compose up d --remove-orphans`
   (ключ `-remove-orphans` удалит неиспользуемые контейнеры старых версий).

3.2. Для мажорных версий, работающих в Docker Compose

Для обновления мажорных версий, работающих в Docker Compose, необходимо выполнить следующие действия:

1. Перейти в домашнюю директорию проекта «Планета» и остановить все контейнеры командой:
   `docker-compose down`
2. Сделать резервную копию БД «Планета. Аналитика»:
   - если БД работает в контейнере, то необходимо сделать копию **docker volume**;
   - если используется внешняя БД, то необходимо создать копию БД средствами СУБД.
3. Создать бэкап всех томов и файла .env.
4. Обновить скрипт *docker-compose* (и dc) простой заменой.
5. Выполнить синхронизацию настроек в файлах docker-compose.yml и .env**.
   Если в версии 1.9 переименовывались/изменялись тома (docker volumes),

синхронизировать настойки **docker volumes** между старым и новым файлами docker-compose.yml и .env**.

6. Синхронизировать значения переменных, содержащие данные УЗ для подключения к почте, к БД и к logtsash и другим внешним сервисам.

7. Задать значения новых переменных в файлах docker-compose.yml и .env:
   - если БД работает в контейнере, то должны выполняться следующие условия:
   а). Значения переменных **ACCESS_PG_DB_** должны быть установлены равными **ANALYTICS_PG_DB_****
   б). Значения переменных **ACCESS_OAUTH_DB_** должны быть установлены равными **ANALYTICS_CONTENT_DB_****
   в). **ACCESS_PG_DOCKER_VOLUME_NAME** должен указывать на клон **ANALYTICS_PG_DOCKER_VOLUME_NAME**, описанный на **шаге 2**.
   - если используется внешняя БД, то следует установить такие значения переменных **ACCESS_OAUTH_DB_***, чтобы они соответствовали БД-клону из **шага 2**.

8. Перейти в домашнюю директорию проекта «Планета» и остановить все контейнеры командой:
   `docker-compose down`

9. Сделать копию БД «Планета. Аналитика»:
   - если БД работает в контейнере, то необходимо сделать копию **docker volume**;
   - если используется внешняя БД, то необходимо создать новую БД и загрузить дамп.

10. Создать бэкап всех томов и файла .env.

11. Обновить скрипт *docker-compose* (и dc) простой заменой.

12. Применить скрипт миграции:
    - скрипт миграции запустится автоматически, для этого нужно выполнить:
    `PLANETA_№version_MIGRATION=yes dc up -d`
    - дождаться окончания и убедиться, что миграция завершилась успешно. Для этого нужно получить логи командой:
    `dc logs access-pg-ready`
    и убедиться, что после строки `№version migration started` отсутствуют сообщения об ошибке;
    - остановить все контейнеры, выполнив команду
    `docker-compose down`

13. Запустить планету в штатном режиме. Для этого следует выполнить команду (без переменной **PLANETA_№version_MIGRATION=yes**):
    `docker-compose up -d`

3.3. Для версий, работающих в Kubernetes

Для обновления минорных или мажорных версий, работающих в Kubernetes, необходимо выполнить следующие действия:

1. Обновить файлы манифестов Kubernetes (если в них выполнялись правки). Если поставка выполнялась с помощью **Helm chart**, то изменение образов осуществляется в файле values-*.yaml.
2. Выполнить бэкап образов контейнеров (images).
3. Выполнить бэкап всех томов (PVCs).
4. Получить обновленные файлы манифестов Kubernetes, а при наличии, и файлы Helm chart.
5. Обновить полученные файлы манифестов Kubernetes (а при наличии, и файлы Helm chart), в соответствии с добавленными к репозиторию инструкциями.

4. Базы данных

Используемые для работы базы данных перечислены в таблице далее.

Объект «Планета»	Назначение БД
Планета. НСИ	Используется для хранения метаинформации, структуры, данных справочников
Планета. Доступ	Используется для хранения идентификационных данных (учетных записей, групп, ролей), настроек системы разграничения прав доступа
Планета. Процессы	Используется для хранения метаинформации и контекста исполнения бизнес процессов
Планета. Аналитика	Используется для хранения метаинформации для визуализации аналитических панелей
	Используется для хранения витринного слоя данных для аналитических панелей

4.1. Конфигурация СУБД

Конфигурация СУБД использует настройки по умолчанию.

4.2. Размещение СУБД

Пути размещения всех объектов СУБД на сервере (рабочего каталога СУБД, табличных пространств БД, конфигурационных файлов СУБД, логов транзакций СУБД, архива логов транзакций СУБД, локальных копий СУБД, диагностических журналов СУБД, исполняемых файлов СУБД) указываются по умолчанию.

4.3. Резервное копирование и восстановление СУБД

Резервное копирование и восстановление предусмотрено для следующих видов данных:

- базы данных (PostgreSQL);
- содержимое файлового хранилища (DRBD). Восстановление обеспечивается дублированием сервера компонента;
- конфигурации Deckhouse (Deckhouse).

Методика восстановления СУБД подразумевает выполнение действий, обеспечивающихся штатной функциональностью ПО PostgreSQL.

4.4. Мониторинг работы СУБД

Мониторинг работоспособности БД в системе обеспечивается функциональностью ПО Grafana и ПО Prometheus.

Сведения об этом ПО содержатся в документации компаний-разработчиков:

Prometheus — https://prometheus.io/docs/visualization/grafana/

Grafana — https://grafana.com/docs/grafana/latest/#guides

5. Управление контейнерами

Управление контейнерами выполняет сотрудник с административной ролью «Администратор серверов контейнеризации».

Информация о функционировании контейнеров отображается на информационных панелях.

Доступ к информационным панелям осуществляется через веб-интерфейс.

Установка и настройка инструмента контейнеризации и связанного с ней ПО описана в документе Руководство по установке и настройке ПК SoD.

6. Управление профилями пользователей

Управление профилями пользователей осуществляется сотрудниками с соответствующими административными ролями.

В процессе развёртывания системы используются скрипты(-ы) для создания «первичного» пользователя.

Пример скрипта

Для добавления пользователя следует использовать скрипт *add-user.sql*, добавляющий роль **SYS_ADMIN**. Если нужна другая роль — следует указать требующуюся.

Необходимые параметры скрипта:

`:username` — логин пользователя;

`:password` — пароль. Указывается в обычной форме, без шифрования (при первом входе система потребует изменить пароль и сохранит его в зашифрованном виде);

`:realm_id` — пространство в котором создается пользователь.

Для добавления клиента следует использовать скрипт *add-new-client.sql*.

Необходимые параметры скрипта:

`:client_id` — текстовый идентификатор клиента;

`:client_secret` — секрет клиента;

`:realm_id` — идентификатор пространства.

Для добавления списка ролей следует использовать скрипт *add-sys-roles.sql*.

Необходимые параметры скрипта:

`:username` — пользователь, которому выдаются эти системные роли;

`:realm_id` — идентификатор пространства, которому принадлежит пользователь;

`:role_names` — наименования (коды) ролей. В параметрах задаётся как список (`'test1'`, `'test2'`).

Интерфейс

В процессе эксплуатации – работа с пользователями, их группами, ролями и т. п. выполняется через интерфейс «Планета. Доступ».

7. Мониторинг и журналирование

В качестве инструмента мониторинга системы используется ПО Prometheus.

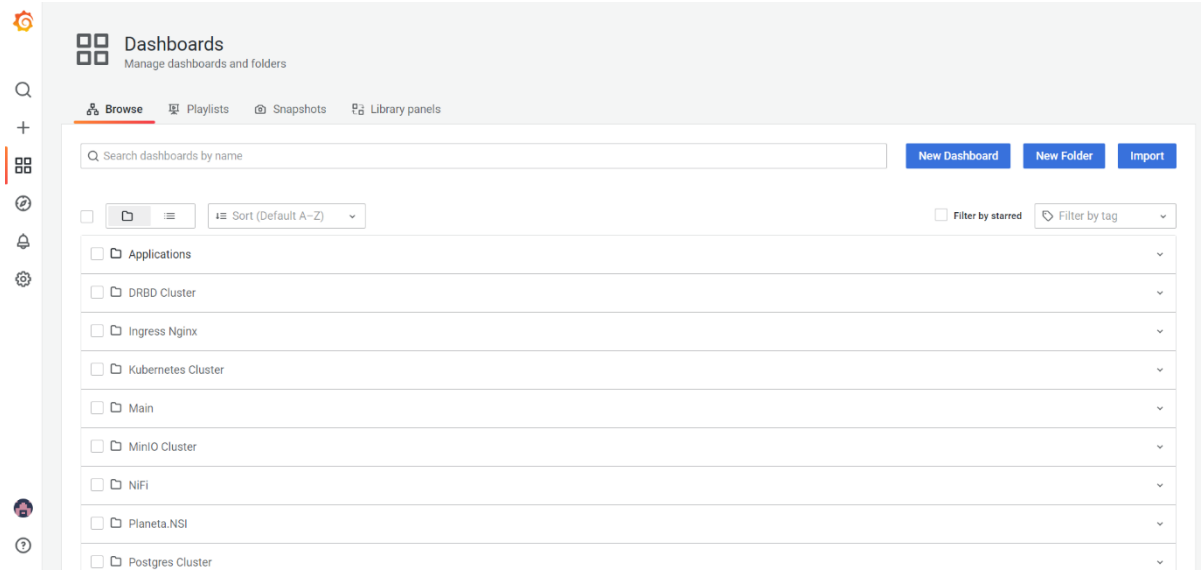
В качестве инструмента журналирования используется ПО Logstash.

7.1. Мониторинг

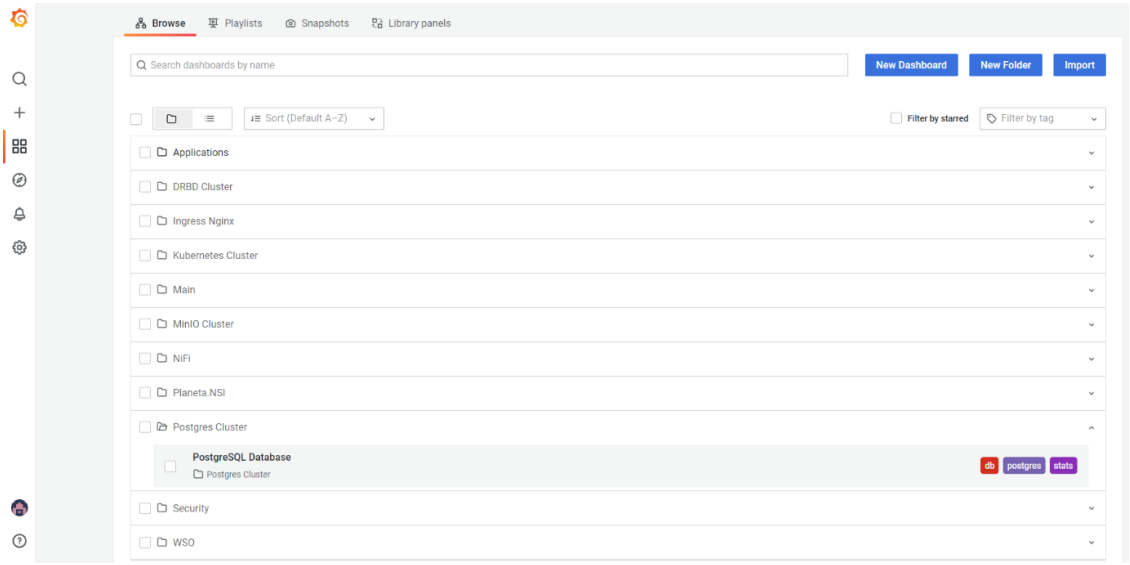
В качестве средства мониторинга системы применяется ПО Prometheus.

Для отображения сведений о состоянии системы используется ПО Grafana с предварительно настроенным набором элементов (панелей) визуализации определённых показателей — информационной панелью.

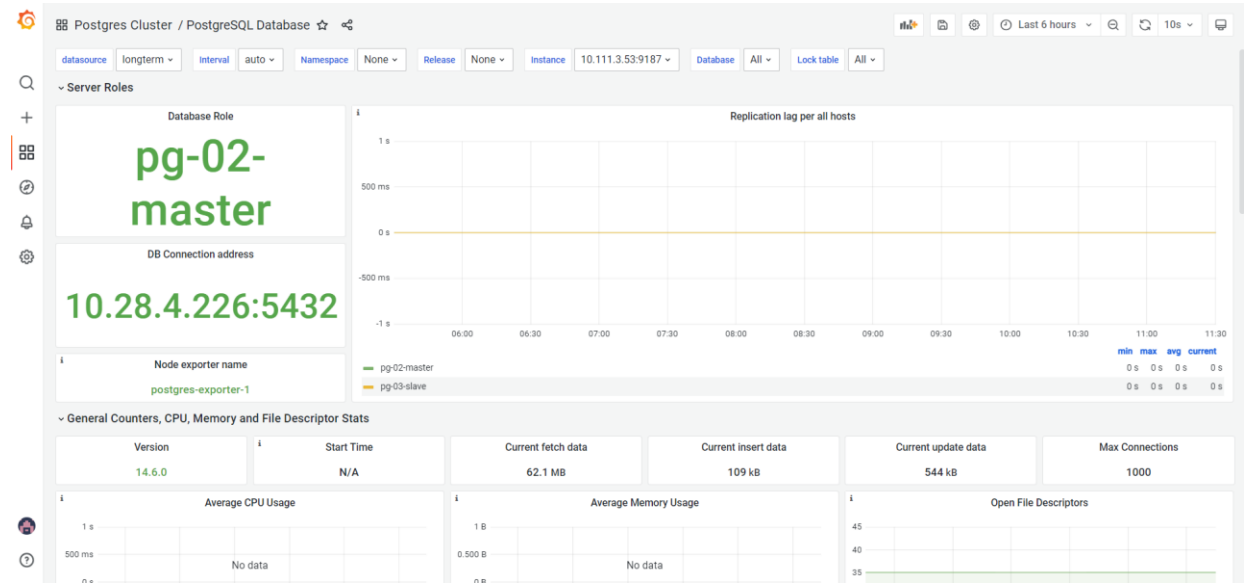
Для перехода в раздел со списком компонентов следует выбрать в меню пункты **Dashboards** – **Browse**:



Выбрав необходимый раздел списка следует указать конкретный элемент, например: **Postgres Cluster**:



После выбора элемента будет отображена информационная панель, содержащая сведения о состоянии компонента системы:



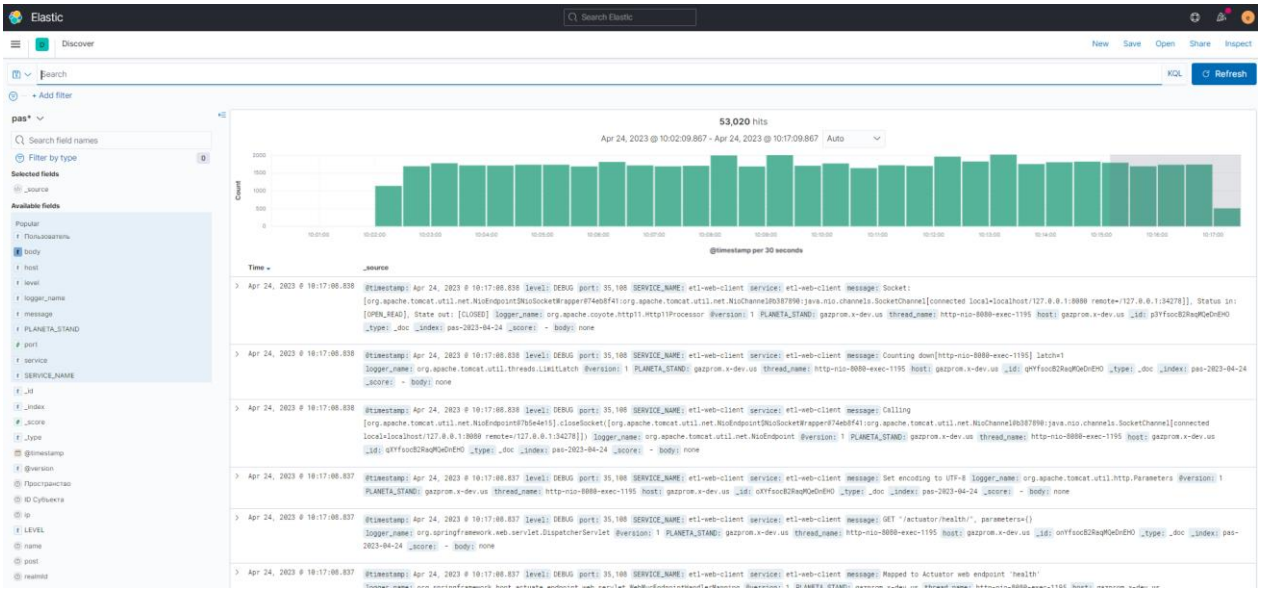
Мониторинг осуществляют сотрудники заказчика со следующими административными ролями:

- администратор инфраструктурных сервисов;
- администратор подсистемы «Интеграционная шина»;
- администратор подсистемы ETL.

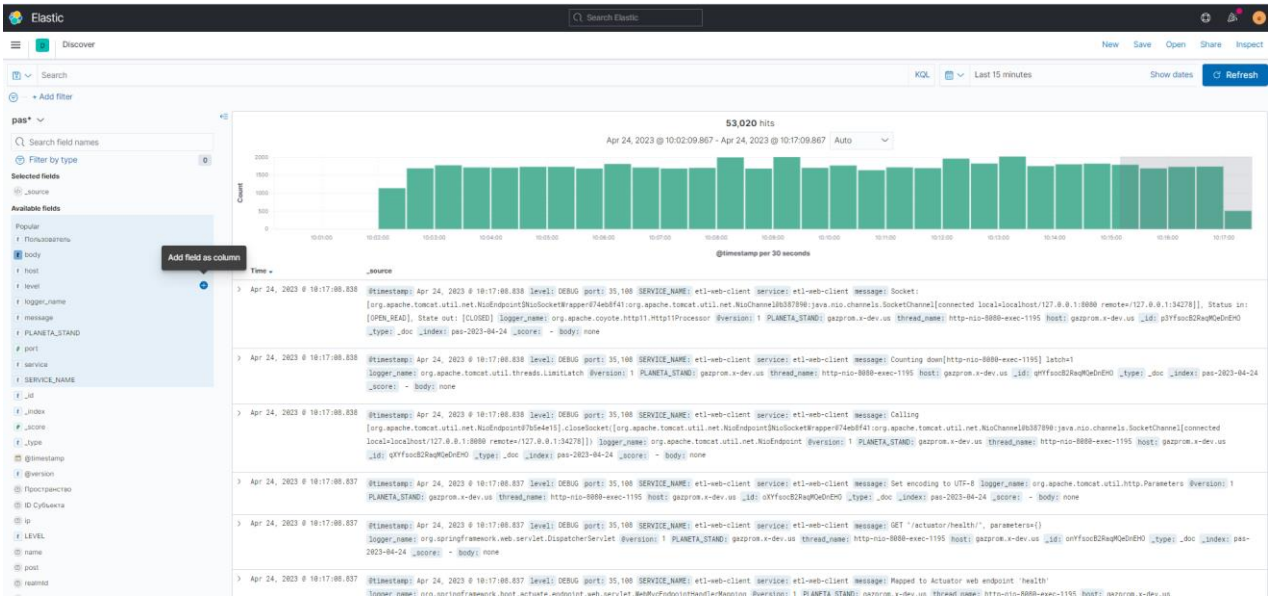
7.2. Журналирование

Для настройки журнала операций выполните следующие действия:

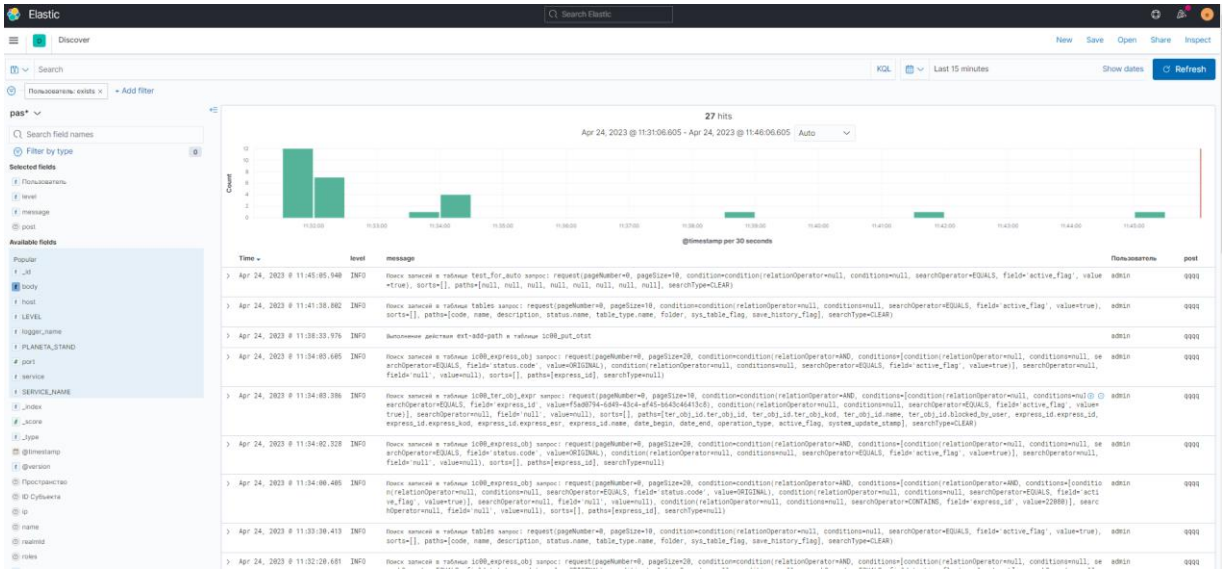
1. Откройте в браузере страницу Kibana (адрес страницы запросите у администратора системы).
2. В диалоговом окне регистрации введите логин и пароль пользователя Kibana (может отличаться от логина и пароля пользователя системы) и нажмите на кнопку **Log In**. В результате загрузится страница с таблицей событий:



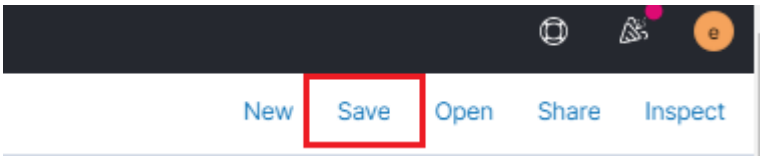
3. С левой стороны в списке полей выберите **level** и нажмите на пиктограмму +:



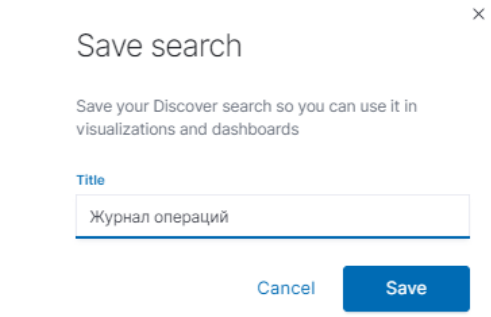
4. Добавьте следующие поля в таблицу событий:
   - **Message** — текст события;
   - **Пользователь** — логин пользователя, инициатора события;
   - **Post** — должность пользователя.



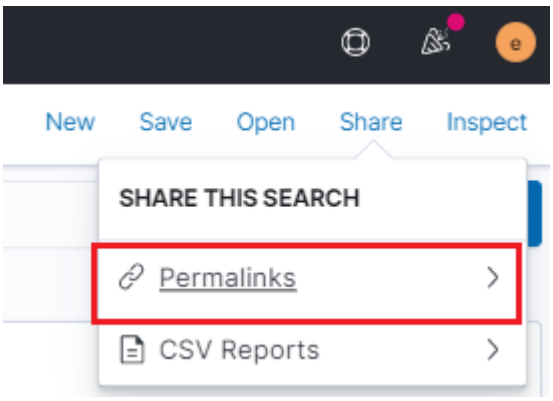
5. Нажмите на кнопку **Save** в правом верхнем углу страницы:



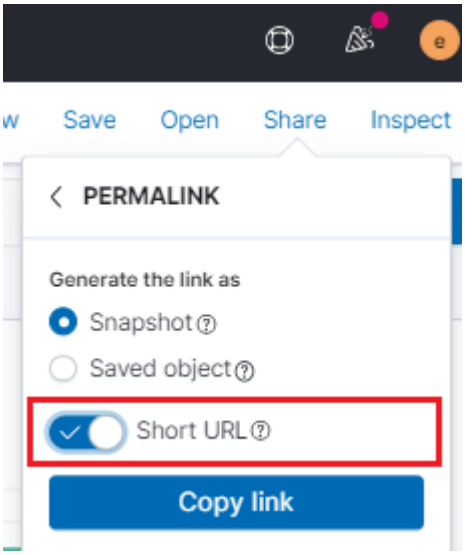
6. В диалоговом окне сохранения настроек в поле **title** введите: **Журнал операций** и нажмите на кнопку **Save**:



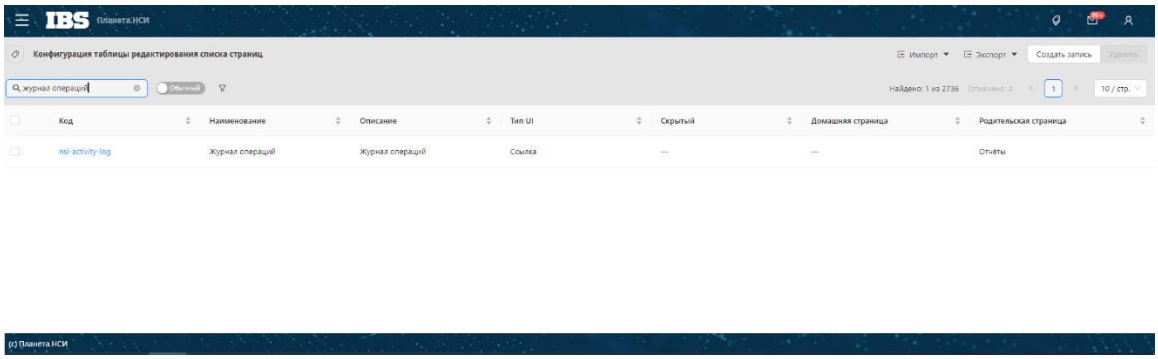
7. Для получения короткой ссылки нажмите на кнопку **Share** в правом верхнем углу страницы, выберите пункт меню **Permalinks**:



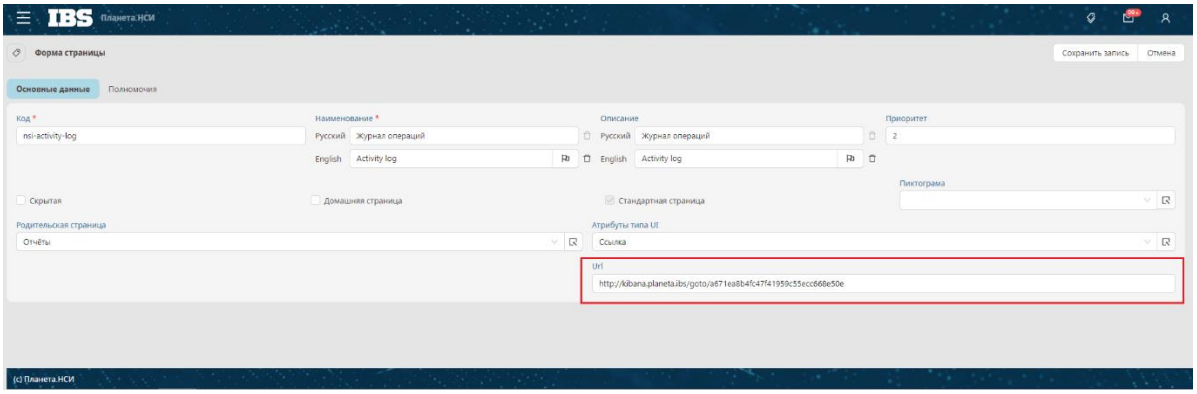
8. В пункте **Short URL** передвиньте переключатель вправо и скопируйте ссылку, нажав на кнопку **Copy link**:



Откройте «Планета. НСИ» и выберите раздел с настройками меню навигации. В поле простого поиска введите: **Журнал операций**:



9. Откройте на редактирование форму ввода журнала операций и скопируйте в поле URL-ссылку, полученную на шаге **8**:



10. Сохраните изменения.

В результате в меню навигации появится пункт **Журнал операций**, при вызове которого будет происходить переход к журналу операция в Kibana.

7.3. Оповещения

Оповещения о состоянии системы формируются при помощи ПО Awesome Prometheus.

Оповещения о состоянии системы и/или об инцидентах направляются посредством электронных писем на адреса электронной почты соответствующих сотрудников заказчика.

Адресатом оповещений в контексте работы системы является Администратор инфраструктурных сервисов.

Оповещения о состоянии системы имеют стандартную структуру:

Значение	Описание
- alert:	Название оповещения
expr:	Условие создания оповещения
for:	Настроенное время срабатывания оповещения
labels: severity:	Принадлежность оповещения к определённой группе Уровен важности оповещения
annotations: summary:	Раздел описания оповещения Описание оповещения
description:	Заголовок оповещения

7.4. Управление логированием

Управление логированием используется при работе со всеми backend-сервисами.

Параметры подключения к Logstash (хранилищу логов, из которого они переносятся в Elasticsearch):

`SPRING_PROFILES_ACTIVE= logstash` — активирует передачу логов в Logstash

`LOGSTASH_HOST` — хост для подключения к Logstash

`LOGSTASH_PORT` — хост для подключения к Logstash

`PLANETA_STAND` — метка в логах для указания с какого стенда пришли логи. Актуально при подключении нескольких стендов к одному ELK.

Уровни логирования

OFF — никакие логи не записываются;

FATAL — ошибка, после которой приложение уже не сможет работать и будет остановлено;

ERROR — уровень на котором выводятся ошибки приложения, не останавливающие работу системы в целом;

WARN — логи, которые содержат предостережение. Неожиданные действия, непрерывающие выполнение запросов, но на которые необходимо обратить внимание;

INFO — логи, обозначающие важные события в системе, а также ожидаемые действия пользователя;

DEBUG — логи, необходимые для отладки приложения;

TRACE — логи отладки, с наименьшим уровнем логирования;

ALL — уровень, при котором будут записаны все логи из системы.

Уровни указаны в порядке уменьшения важности, при указании одного из уровней будет выводится соответствующий уровень и выше. Так при указании уровня **INFO** будут выведены логи уровней: **INFO**, **WARN**, **ERROR**, **FATAL**.

Стандартный уровень логирования для сервисов «Планета» — **INFO**.

Для управления уровнем логирования используется параметр `logging.level.[путь до класса]`.

Примеры использования:

`logging.level.ru.ibs=INFO` — устанавливает уровень логирования INFO для кода содержащего бизнес-логику сервисов «Планета»;

`logging.level.root=INFO` — устанавливает уровень логирования INFO для всего кода приложения.

Чтобы включить вывод запросов, генерируемых «Планета. НСИ» к базе данных нужно указать параметр `HIBERNATE_SHOW_SQL=true`.

8. Информационная безопасность

Доступ к данным прикладных корпоративных систем производится в соответствии с утвержденной программой взаимодействия подразделениями, осуществляющими разработку и/или сопровождение.

Доступ на изменение данных предоставляется только авторизованным пользователям на основании их служебных полномочий и в соответствии с ролевой моделью.

Доступ на просмотр данных предоставляется всем пользователям сети заказчика, без дополнительной авторизации.

8.1. Использование «Планета. Доступ»

В качестве средства защиты информации для систем, создаваемых на основе программных продуктов Платформы «Планета», и для обеспечения работы технологии единого входа (Single Sign-On) используется продукт «Планета. Доступ».

Продукт обеспечивает разграничение доступа пользователей в зависимости от их ролей и прав на объекты.

Подробная информация о работе с «Планета. Доступ» содержится в документе Руководство пользователя «Планета. Доступ».

9. Устранение сбоев

В случае сбоев, связанных с ОС и драйверами устройств, работоспособность восстанавливается средствами ОС.

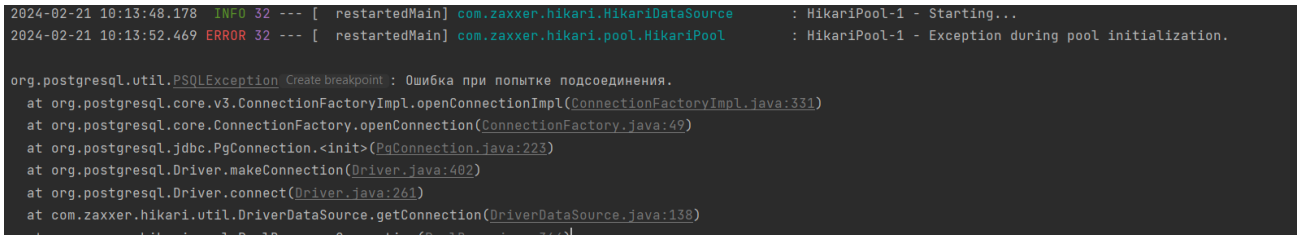
В случае сбоя в работе инструмента контейнеризации восстановление его работоспособности обеспечивается функциональностью оркестратора контейнеризации и выполняется автоматически.

Описание ошибок и способы их устранения:

Код ошибки	Описание, действия администратора
HTTP-code 500	Система недоступна. Необходимо связаться с администратором системы
HTTP-code 503	Сервис временно недоступен. Попробуйте повторить запрос позднее. В случае повторения ошибки, свяжитесь с администратором системы
HTTP-code 401	При обращении указан токен, срок действия которого вышел. Запросите актуальный токен доступа и повторите запрос. В случае повторения ошибки, свяжитесь с администратором системы
HTTP-code 405	Используемый метод не разрешен для вызова API. Уточните по документации API поддерживаемый метод и скорректируйте запрос. В случае повторения ошибки, свяжитесь с администратором системы

9.1. Недоступность базы данных

Пример логов access-oauth:



Ошибка возникает при отсутствии сетевой связанности или указании некорректных настроек при подключении к БД. Необходимо проверить и указать корректные настройки в переменных окружения:

```
ACCESS_OAUTH_DB_HOST=<значение>
ACCESS_OAUTH_DB_NAME=<значение>
ACCESS_OAUTH_DB_PASS=<значение>
ACCESS_OAUTH_DB_PORT=<значение>
ACCESS_OAUTH_DB_USER=<значение>
```

Также изменить переменные для подключения к БД можно в конфигурационном файле. Например, в docker-compose.default.yml. В данном случае указываются переменные из раздела 4. Переменные окружения:

```
access-oauth:
    environment:
        PG_DB_HOST: <значение>
```

```
PG_DB_PORT: <значение>
PG_DB_NAME: <значение>
PG_DB_USERNAME: <значение>
PG_DB_PASS: <значение>
```

9.2. Недоступность Redis

Пример логов access-oauth:

Ошибка возникает при отсутствии сетевой связанности или указании некорректных настроек при подключении к Redis. Необходимо проверить и указать корректные настройки.

Следует внести изменения в файл .env:

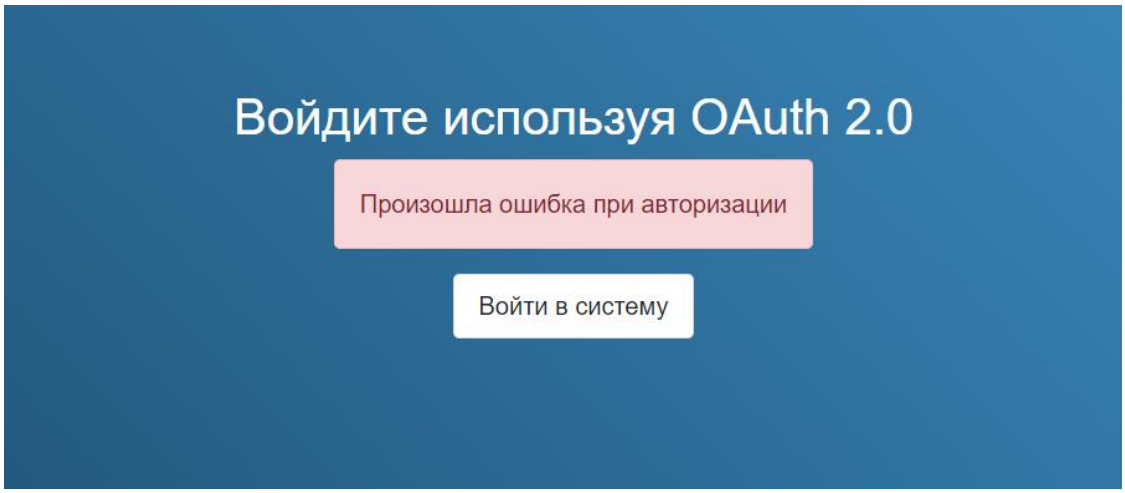
```
UTILS_REDIS_AUX_HOST=<значение>
UTILS_REDIS_AUX_PORT=<значение>
UTILS_REDIS_SERVICE_PORT=<значение>
```

Следует внести изменения в конфигурационный файл (например, docker-compose.default.yml):

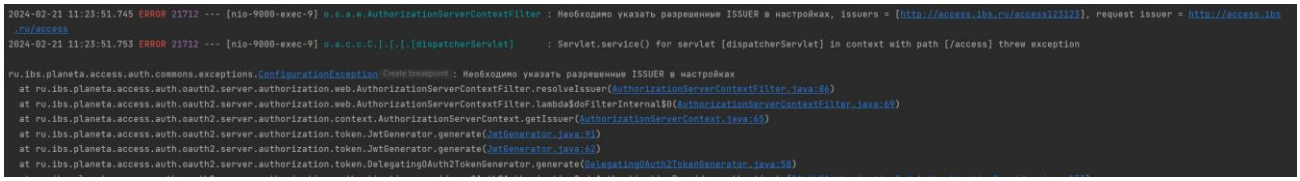
```
access-oauth:
    environment:
       REDIS_HOST: <значение>
       REDIS_PORT: <значение>
```

9.3. Ошибка аутентификации

Пример сообщения об ошибке:



Пример логов access-oauth:



Ошибка возникает при некорректном указании переменной **ISSUER**. Необходимо проверить и указать значение, соответствующее домену, с которого происходит аутентификация.

Следует внести изменения в файл .env:

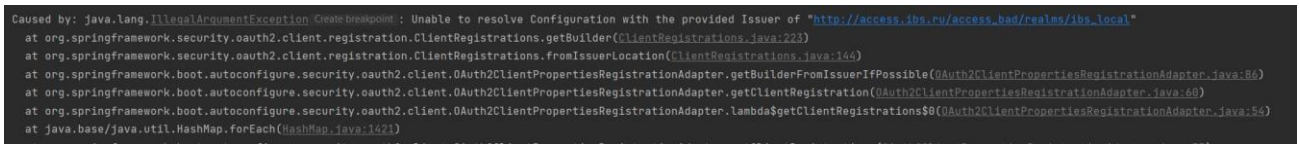
```
ACCESS_OAUTH_OPT_ISSUER=<значение>
```

Следует внести изменения в конфигурационный файл (например, docker-compose.default.yml):

```
access-oauth:
    environment:
       ISSUER: <значение>
```

9.4. Ошибка на старте сервисов при интеграции с «Планета. Доступ»

Пример логов:



Ошибка возникает при отсутствии сетевой связанности между сервисом и «Планета. Доступ», либо при некорректном указании адреса для подключения к «Планета. Доступ»:

Следует внести изменения в файл .env:

PLANETA_ACCESS_BASE_URL=<значение>

Следует внести изменения в конфигурационный файл (например, docker-compose.default.yml):

```
access-oauth:
    environment:
       PLANETA_ACCESS_BASE_URL: <значение>
```

9.5. Недостаток ресурсов

При падении сервиса с ошибкой *OutOfMemoryException* необходимо увеличить значение **Xmx** у JVM, а также увеличить ограничение памяти у контейнера. Значение **Xmx** должно быть равно 70% от ограничения у контейнера.

Следует внести изменения в файл .env:

```
ACCESS_OAUTH_JAVA_XMX=<значение>
ACCESS_OAUTH_DOCKER_LIMIT_CPU=<значение>
ACCESS_OAUTH_DOCKER_LIMIT_MEM=<значение>
```

Следует внести изменения в конфигурационный файл (например, docker-compose.default.yml):

```
access-oauth:
    cpus: <значение>
    environment:
        JAVA_OPTS: <значение>
        mem_limit: <значение>
```

10. Набор документации продукта ПК SoD

Общие сведения о продукте содержатся в документе Описание программного продукта ПК SoD.

Сведения об установке и настройке продукта содержатся в документе Руководство по установке и настройке ПК SoD.

Сведения об использовании продукта содержатся в документе Руководство пользователя ПК SoD.

Термины и определения

Термин	Определение
Администратор	Сотрудник организации, обладающий компетенциями и правами для управления ИС
Информационная панель	Набор элементов интерфейса ИС, обеспечивающий визуализацию данных о состоянии компонентов
Информационная система (ИС)	Совокупность программных продуктов, использование которых позволяет выполнять определённые задачи
Объект (программный продукт)	Часть программного обеспечения ИС, выполняющая определённые задачи
Платформа «Планета»	Набор разработанных ГК ИБС программных продуктов, обеспечивающих решение различных корпоративных задач в области обработки данных и управления данными и процессами компании
Программный продукт	Набор программных компонентов
Справочник	Сущность ИС, содержащая определённые данные и позволяющая их обработку

Сокращения

Сокращение	Значение
БД	База данных
ИС	Информационная система
НСИ	Нормативно-справочная информация
ПО	Программное обеспечение
СУБД	Система управления базами данных
ELK	Elasticsearch, Logstash, Kibana — программные решения для решения задач поиска и аналитики при обработке данных
ETL	Extract, Transform, Load — извлечение, преобразование, загрузка — процессы управления хранилищами данных
HTTP	HyperText Transfer Protocol (протокол передачи гипертекста) — сетевой протокол прикладного уровня
TCP	Transmission Control Protocol (протокол управления передачей) — сетевая модель, описывающая процесс передачи цифровых данных