Платформа Радар

Руководство администратора

Версия 3.1.0

ООО Пангео Радар

Оглавление

1. Интерфейс администратора

По умолчанию интерфейс пользователя доступен по URL http://<адрес сервера>:8080.

Процедура входа и общее описание интерфейса подробно описано в «Руководстве пользователя».

При наличии дополнительных прав пользователю доступен раздел «Администрирование», который содержит следующий пункты (Рисунок 1):

- **Пользователи и права** - управление пользователями;
- **Кластер** - управление Платформой;
- **Источники** - управление подключением источников событий;
- **Мониторинг** - просмотр метрик работоспособности компонентов Платформы;
- **Репутационные списки** - управление списками индикаторов компрометации;
- **База знаний** - управление базой знаний по типам инцидентов и правилам корреляции.

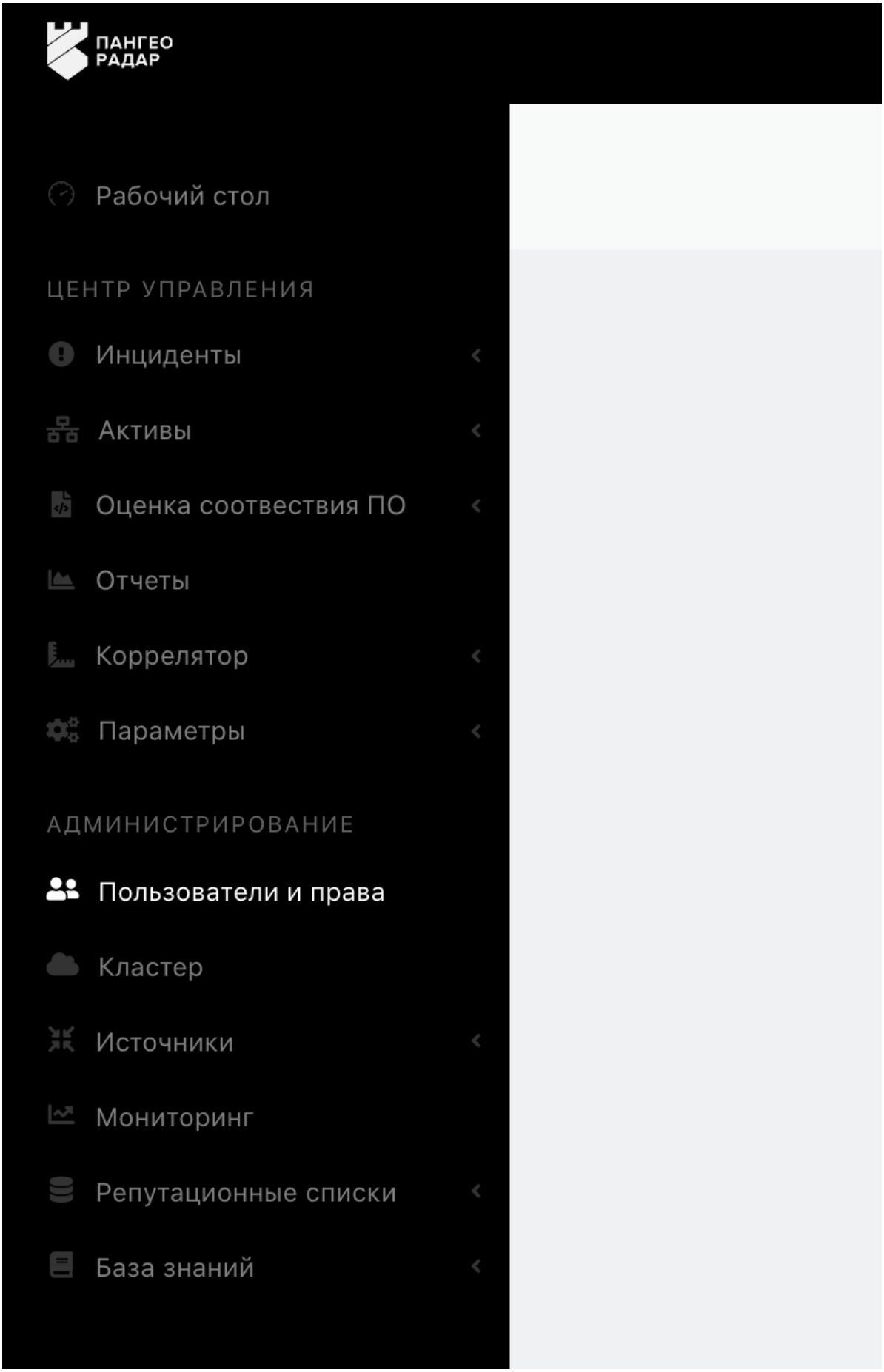

Рисунок 1 - Пункты меню раздела «Администрирование»

2. Управление пользователями

2.1. Общее описание

В Платформе Радар предусмотрена возможность многопользовательской работы. Каждый пользователь работает под своими учетными данными. Для построения рабочего процесса по управлению инцидентами пользователи включаются в группы согласно выполняемым функциям.

Основной раздел интерфейса «**Пользователи и группы**» предназначен для выполнения следующих функций:

- управление учетными записями пользователей;
- контроль состояния активности пользователя;
- управления ролями и группами пользователей.

Раздел содержит следующие вкладки:

- "**Пользователи**" — вкладка предназначена для управления учётными записями пользователей.
- "**Группы**" — вкладка предназначена для управления группами пользователей.
- "**Роли**" — вкладка предназначена для управления ролями, назначаемыми пользователям.
- "**Аудит действий**" — вкладка предназначена для просмотра действий пользователей.

2.2. Управление учетными записями пользователей

2.2.1. Вкладка "Пользователи". Общее описание

Вкладка "**Пользователи**" содержит (см. Рисунок 2):

- Текущий список зарегистрированных на Платформе пользователей в виде табличного списка.
- Форму для создания нового пользователя на Платформе — форма "Создать нового".
- Функцию назначения пользователю роли.
- Функцию добавления пользователя в группу.

На вкладке "**Пользователи**" доступны следующие опции по управлению учетными записями пользователей:

- Создание нового пользователя;
- Редактирование существующего;
- Удаление пользователя;
- Переключение статуса активности пользователя;
- Назначение пользователю групп и ролей;
- Исключение пользователя из группы;
- Снятие с пользователя определенной роли;
- Смена пароля пользователю.

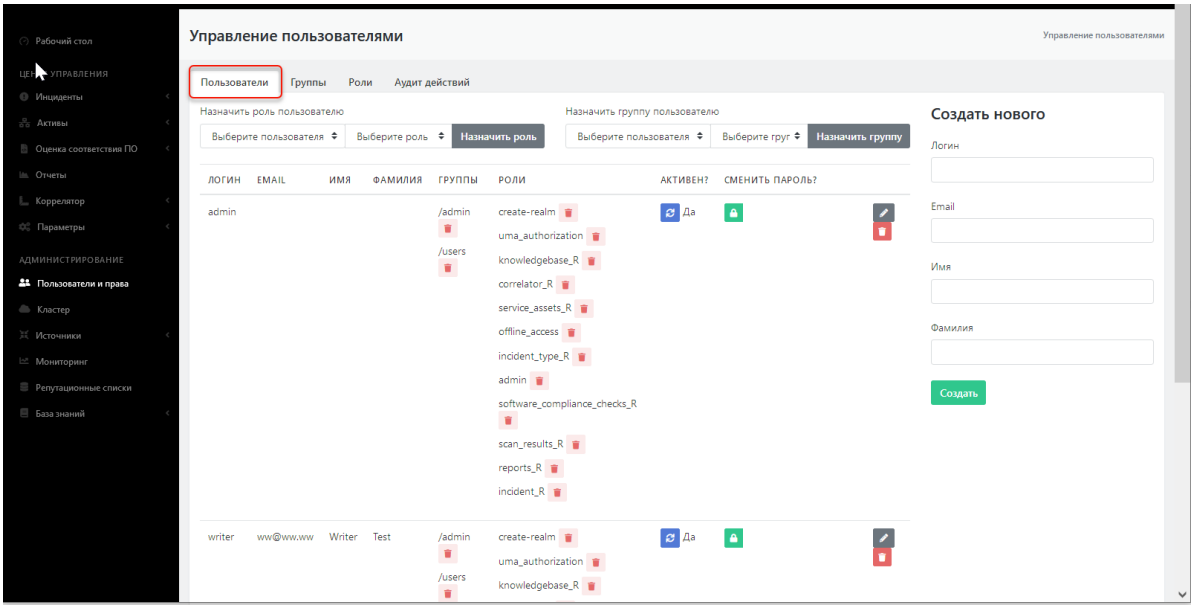

Рисунок 2 - Экран управления учетными записями пользователя (вкладка "Пользователи

2.2.2. Создание нового пользователя

Для создания нового пользователя в Платформе необходимо выполнить следующие действия:

1. Заполнить форму "Создать нового" (см. Рисунок 4). При создании формы обязательны к заполнению все поля формы.
2. Сохранить введенные данные нажав на кнопку "Создать".

   В таблице пользователей должна появится строка с новым пользователем. Пользователь автоматически создается с ролью user и включен в соответствующую группу.
3. Сгенерировать пароль для нового пользователя нажав на пиктограмму  ;
4. При необходимости провести настройку групп и ролей пользователя (см. раздел "Присвоение роли и группы пользователю").

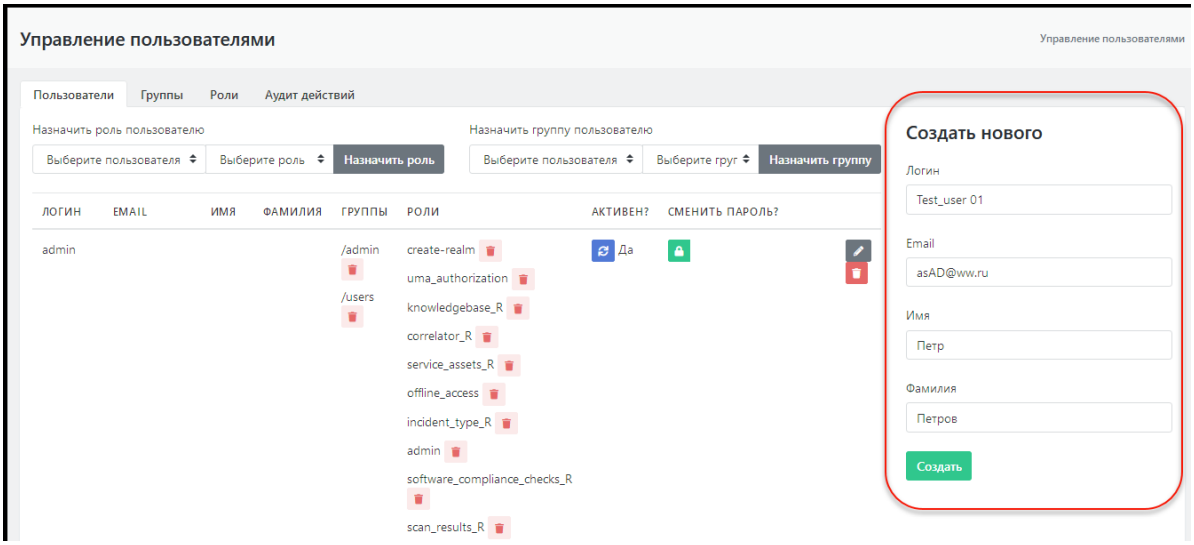

Рисунок 4 - Форма создания нового пользователя

2.2.3. Редактирование данных пользователя

Для внесения изменений в данные пользователя необходимо:

1. Нажать на пиктограмму  в строке интересующего пользователя. Откроется форма "Изменение" с данными пользователя доступными для редактирования (см. Рисунок 5).
2. Внести необходимые изменения и нажать кнопку «Обновить» для сохранения изменений или кнопку «Отменить» (для отмены).

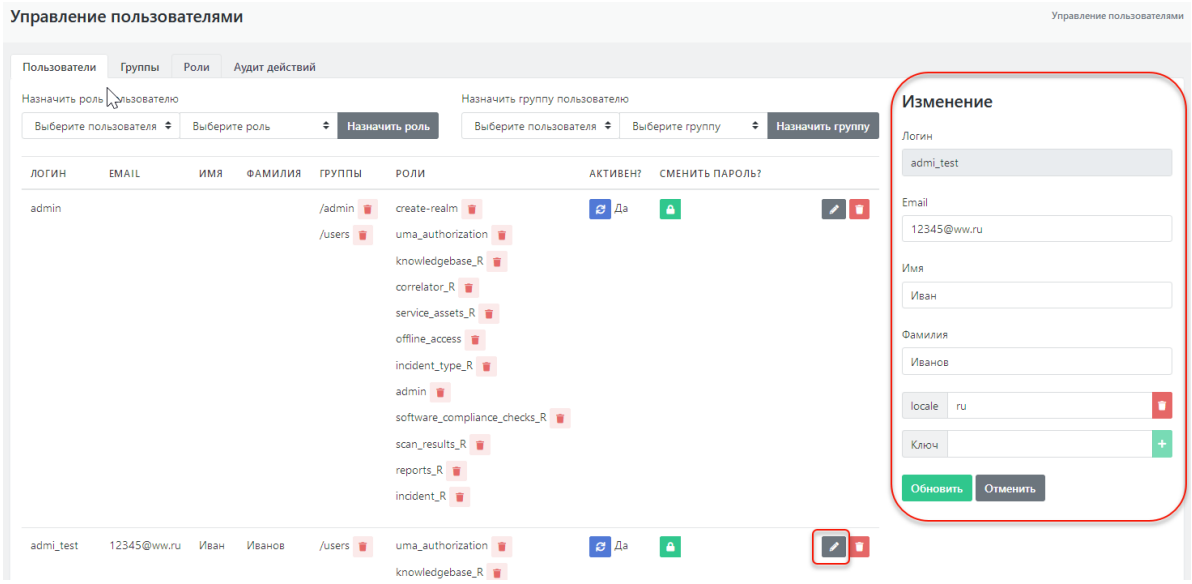

Рисунок 5 - Форма редактирования данных пользователя

2.2.4. Изменение пароля

Для изменения пароля пользователя необходимо нажать на пиктограмму  , после чего будет сгенерирован новый пароль в появившейся форме рядом (см. Рисунок 6).

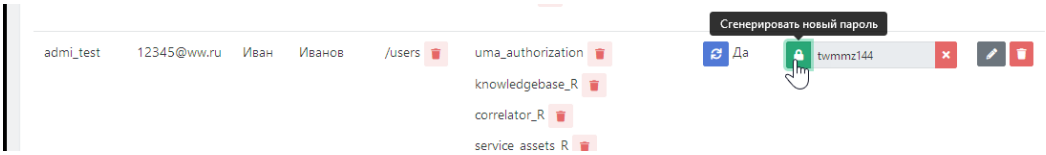

Рисунок 6 - Генерация нового пароля пользователя

2.2.5. Удаление пользователя

Для удаления пользователя с Платформы необходимо:

1. Нажать на пиктограмму  .
2. В открывшемся окне подтвердить удаление пользователя (см. Рисунок 7).

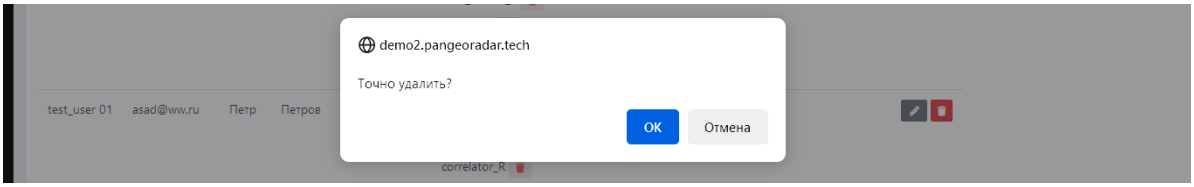

Рисунок 7 - Удаление пользователя

2.2.6. Присвоение роли пользователю. Отключение роли

Для присвоения пользователю новой роли необходимо (см. Рисунок 8):

1. В области "Назначить роль пользователю" выбрать в раскрывающемся списке пользователей интересующего пользователя.
2. Выбрать в раскрывающемся списке ролей необходимую роль.
3. Нажать на кнопку "Назначить роль".

Указанная роль должна появится в списке ролей пользователя (колонка таблицы "Роли").

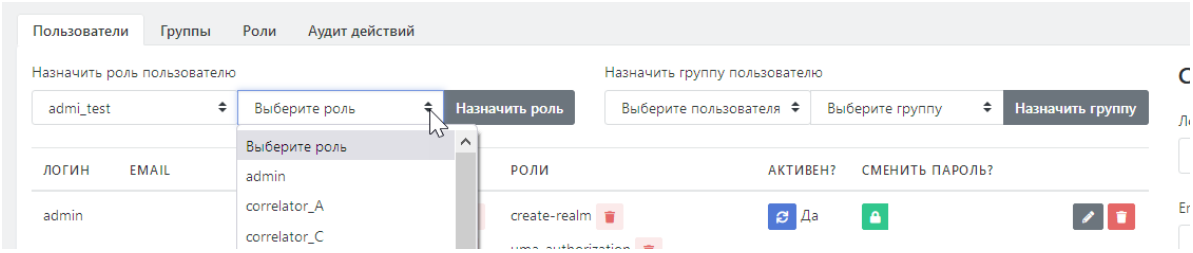

Рисунок 8 - Присвоение пользователю новой роли

Для отключения роли от пользователя необходимо выбрать в списке нужного пользователя и в колонке "Роли" нажать на пиктограмму  рядом с названием той роли, которую необходимо отключить от данного пользователя. Указанная роль будет удалена из строки пользователя.

Таким образом можно удалить только отдельно добавленную пользователю роль. Роль, привязанную к группе и добавленную пользователю при включении его в группу удалить таким образом невозможно. Роли, ассоциированные с группой, удаляются у пользователя, только тогда, когда его исключают из этой группы.

2.2.6.1. Включение пользователя в группу. Удаление из группы

Для включения пользователя в новую группу:

1. В области "Назначить группу пользователю" выбрать в раскрывающемся списке пользователей интересующего пользователя.
2. Выбрать в раскрывающемся списке групп необходимую группу.
3. Нажать на кнопку "Назначить группу".

Указанная группа должна появится в списке групп пользователя (колонка таблицы "Группы").

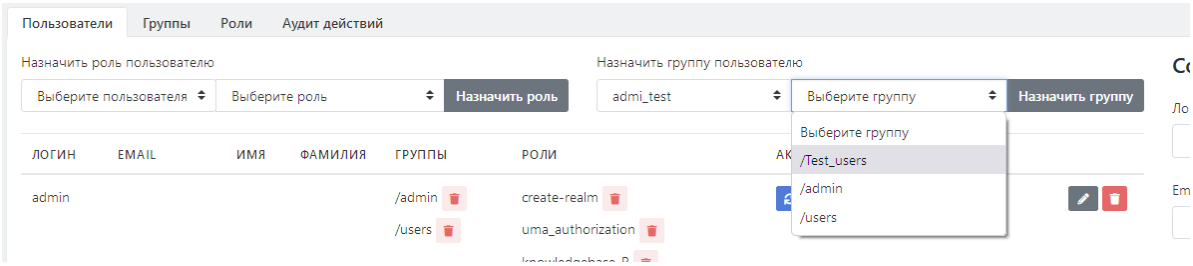

Рисунок 9 - Включение пользователя в группу

Для исключения пользователя из группы выберете в списке нужного пользователя и в колонке "Группы" нажать на пиктограмму  рядом с именем той группы, из которой необходимо удалить данного пользователя. Указанная группа будет удалена из строки пользователя.

2.2.6.2. Изменение атрибутов пользователя

Пользователю могут быть назначены разного рода атрибуты, влияющие на поведение Платформы или содержащие информационный характер.

1. Для добавления нового атрибута пользователю необходимо открыть форму редактирования данных пользователя (см. раздел "Редактирование данных пользователя")(см. Рисунок 10).
2. Ввести в поле "Ключ" название нового атрибута и нажать на пиктограмму +.

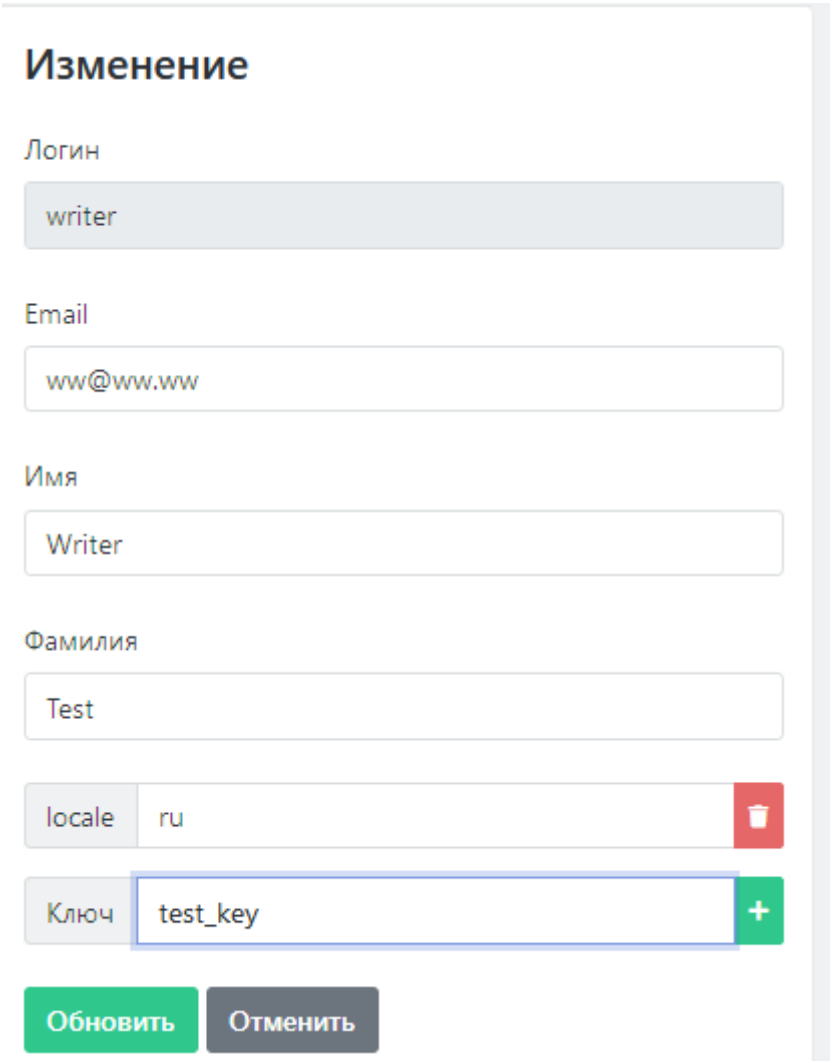

Рисунок 10 - Создание нового атрибута пользователя

В форме редактирования добавится поле нового атрибута (см. Рисунок 11). В данное поле можно внести соответствующее значение.

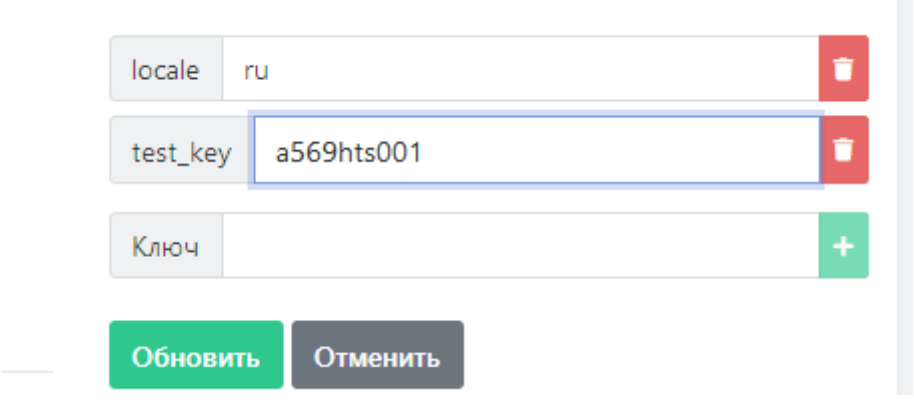

Рисунок 11 - Изменение значения нового атрибута пользователя

По завершению внесения изменений их необходимо сохранить, нажав на кнопку «Обновить» (см. Рисунок 10).
Для удаления атрибута из профиля пользователя необходимо нажать на соответствующую атрибуту пиктограмму  .

Ниже в Таблице 1 приведен список используемых системных атрибутов.

Таблица 1. Список системных атрибутов:

Название	Описание
tz	Строковое значение отвечающее за конвертацию временных меток в интерфейсе в нужную таймзону пользователю. По умолчанию — Europe/Moscow. Список доступных таймзон представлен в приложении №1
is_system_notification	Строковое значение с любым содержимым отвечает за доставку системных уведомлений от Платформы пользователю на E-mail

2.2.7. Управление группами пользователей

2.2.7.1. Вкладка "Группы". Общее описание

Вкладка "**Группы**" содержит (см. Рисунок 12):

- Текущий список зарегистрированных на Платформе групп пользователей в виде табличного списка.
- Поле для создания новой группы пользователей на Платформе — форма "Создать".
- Функцию ассоциации роли с группой.

На вкладке "**Группы**" доступны следующие опции по управлению группами (см. Рисунок 12) :

- Просмотр существующих групп пользователей;
- Создание новой группы пользователей;
- Изменений существующей группы пользователей;
- Удаление группы.
- Назначение пользовательских ролей группам;
- Отключение пользовательских ролей от группы.

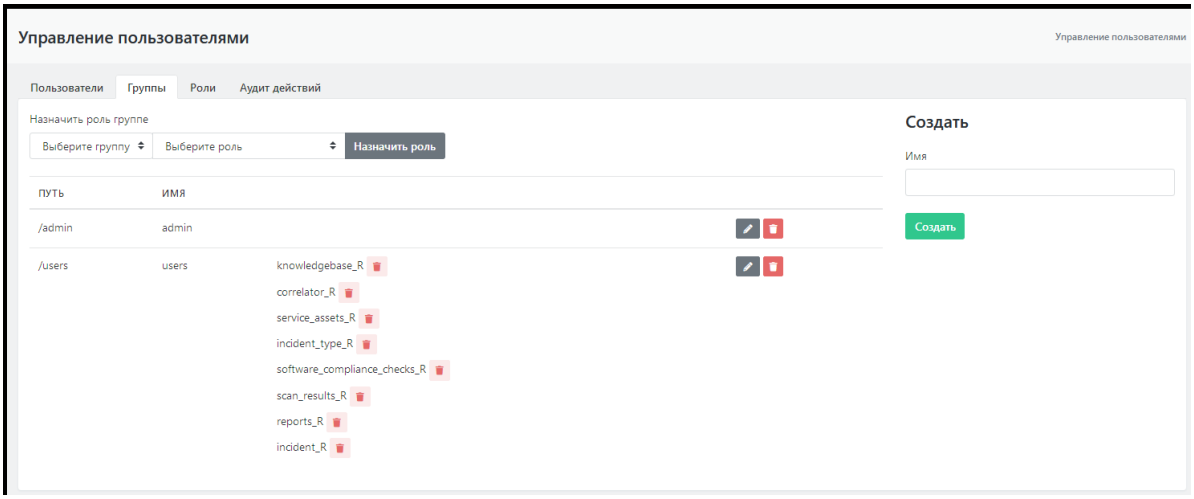

Рисунок 12 - Экран управления пользовательскими группами (вкладка "Группы

2.2.7.2. Создание группы пользователей

Для создания новой группы необходимо ввести имя группы в форму "Создать" и нажать на кнопку "Создать" (см. Рисунок 13). Новая группа отобразится в списке пользовательских групп Платформы.

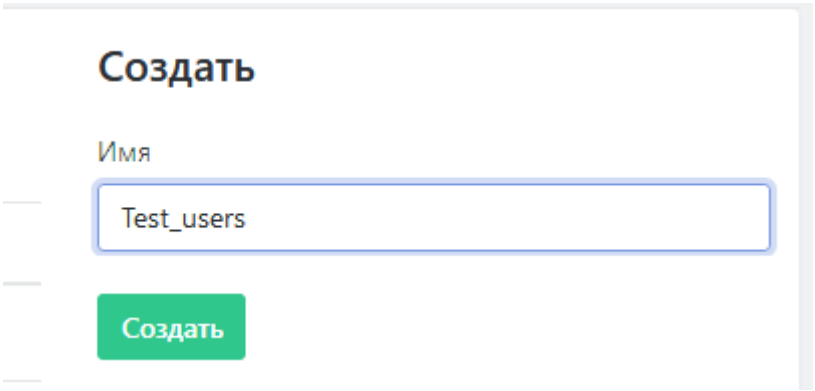

Рисунок 13 - Форма создания новой группы

2.2.7.3. Редактирование данных группы пользователей

Для внесения изменений в данные пользователя необходимо:

1. Нажать на пиктограмму  в строке интересующей группы. Откроется форма "Изменение" с данными группы, доступными для редактирования (см. Рисунок 14).
2. Внести необходимые изменения и нажать кнопку «Обновить» для сохранения изменений или кнопку «Отменить» (для отмены).

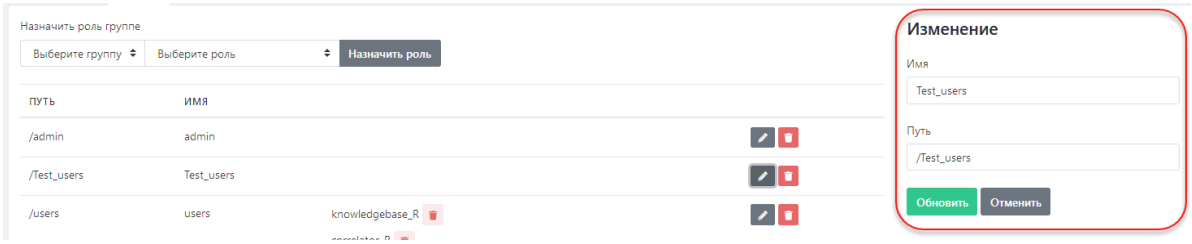

Рисунок 14 - Форма редактирования группы

2.2.7.4. Назначение группы пользователю. Исключение пользователя из группы

Назначение группы пользователю и исключение пользователя из группы выполняется на вкладке "Пользователи" и подробно приведено в разделе "Управление пользователями" (см. пункт "Включение пользователя в группу. Удаление из группы").

(см. пункт "Присвоение роли пользователю. Отключение роли").

2.2.7.5. Присвоение роли группе пользователей. Отключение роли от группы

Для присвоения группе пользователей новой роли необходимо (см. Рисунок 15):

1. В области "Назначить роль группе" выбрать в раскрывающемся списке групп интересующую группу пользователей.
2. Выбрать в раскрывающемся списке ролей необходимую роль.
3. Нажать на кнопку "Назначить роль".

Указанная роль должна появится в списке ролей выбранной группы (колонка таблицы "Роли").

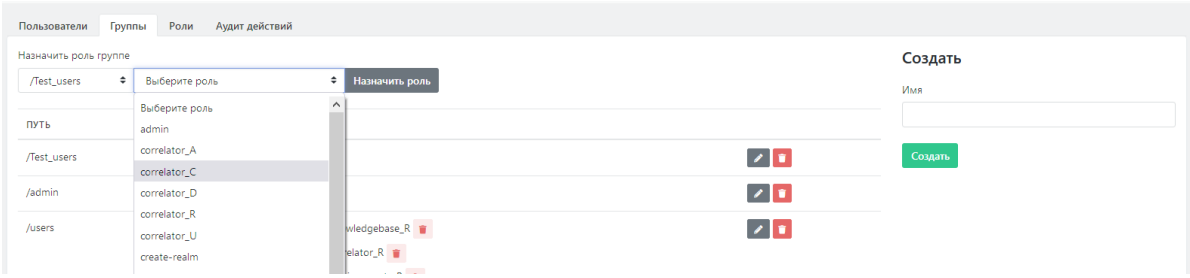

Рисунок 15 - Присвоение пользователю новой роли

Для отключения пользовательской роли от группы необходимо выбрать в списке нужную группу и в колонке "Роли" нажать на пиктограмму  рядом с названием той роли, которую необходимо отключить от данной пользовательской группы. Указанная роль будет удалена из строки группы.

2.2.7.6. Предустановленный список групп пользователей

В Платформе Радар используются группы пользователей по умолчанию представленные в Таблице 2.

Таблица 2 - Предустановленные группы пользователей

Название группы	Включенные роли
admin	
users	- knowledgebase_R - correlator_R - service_assets_R - incident_type_R - software_compliance_checks_R -scan_results_R - reports_R - incident_R

Внимание! Группа «Users» добавляется всем пользователям по умолчанию. Её наличие необходимо для корректной работы пользователя с Платформой.

2.2.8. Управление ролями пользователей

2.2.8.1. Вкладка "Роли". Общее описание

Вкладка "**Роли**" содержит (см. Рисунок 16):

- Текущий список созданных на Платформе ролей пользователей в виде табличного списка.
- Поле для создания новой роли пользователя на Платформе - форма "Создать".

На вкладке "**Роли**" доступны следующие опции по управлению ролями (см. Рисунок 16) :

- Просмотр существующих ролей пользователей;
- Создание новой роли пользователей;
- Изменений существующей роли пользователей;
- Удаление роли.

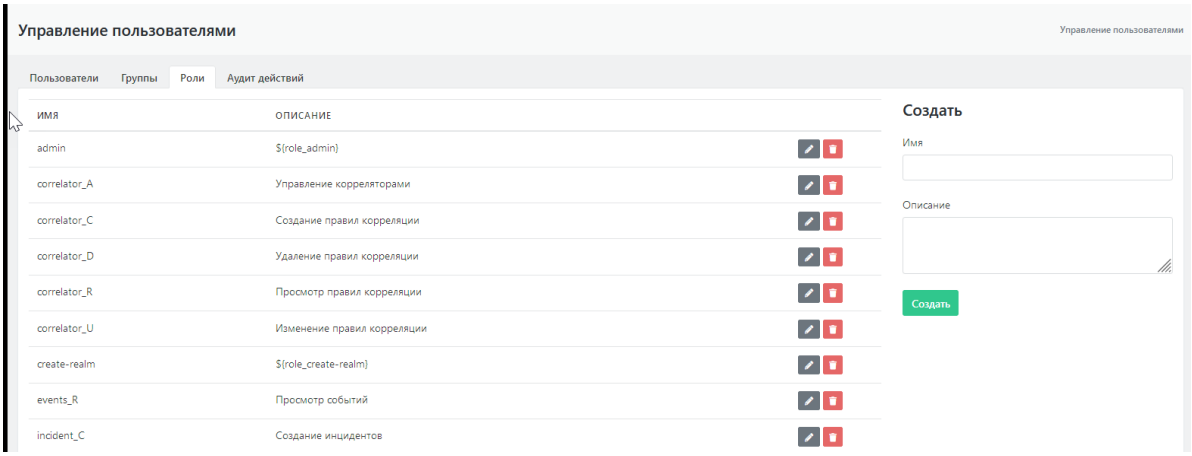

Рисунок 16 - Экран управления ролями пользователей (вкладка "Роли

2.2.8.2. Создание пользовательской роли

Для создания новой роли необходимо ввести имя и описание роли в форму "Создать" и нажать на кнопку "Создать" (см. Рисунок 17). Новая роль отобразится в списке ролей Платформы.

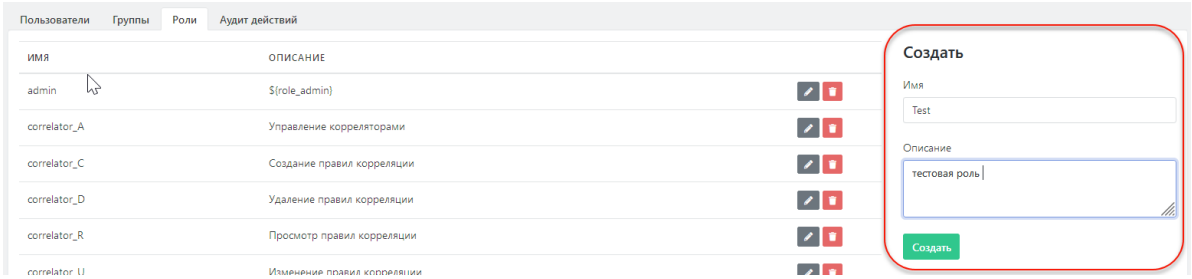

Рисунок 17 - Форма создания новой пользовательской роли

2.2.8.3. Редактирование данных пользовательской роли

Для внесения изменений в данные роли необходимо:

1. Нажать на пиктограмму  в строке интересующей роли. Откроется форма "Изменение" с параметрами роли, доступными для редактирования (см. Рисунок 18).
2. Внести необходимые изменения и нажать кнопку «Обновить» для сохранения изменений или кнопку «Отменить» (для отмены).

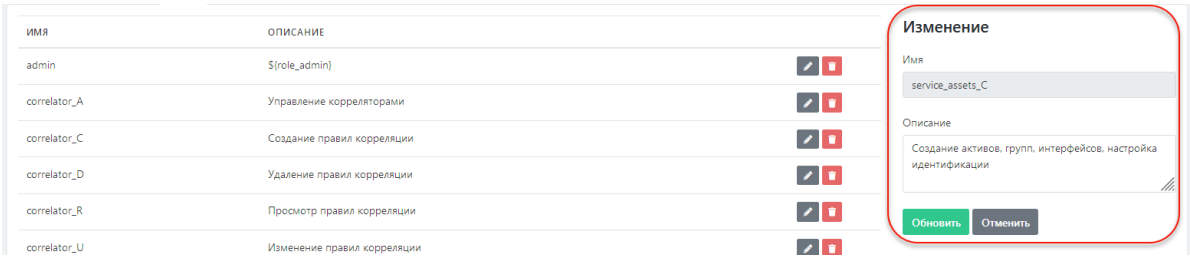

Рисунок 18 - Форма редактирования параметров пользовательской роли

2.2.8.4. Назначение роли пользователю. Отключение роли от пользователя

Назначение роли пользователю и отключение роли от пользователя выполняется на вкладке "**Пользователи**" и подробно приведено в разделе "Управление пользователями" (см. п."Присвоение роли пользователю. Отключение роли").

2.2.8.5. Назначение роли группе пользователей. Отключение от группы

Назначение роли группе пользователей и отключение роли от группы выполняется на вкладке "**Группы**" и подробно приведено в разделе "Управление группами пользователей" (см. п. "Присвоение роли группе пользователей. Отключение роли от группы").

2.2.8.6. Предустановленный список ролей пользователей

В системе есть предустановленный набор привилегий (ролей), которые могут быть назначены пользователям.

Таблица 3 - Предустановленные пользовательские роли

Название роли	Описание роли
admin	Администратор системы
correlator_A	Управление корреляторами
correlator_C	Создание правил корреляции
correlator_D	Удаление правил корреляции
correlator_R	Просмотр правил корреляции
correlator_U	Изменение правил корреляции
events_R	Просмотр событий
incident_C	Создание инцидентов
incident_D	Удаление инцидентов
incident_mass_U	Массовые действия с инцидентами
incident_R	Просмотр инцидентов
incident_status_U	Изменять статус инцидента
incident_type_C	Создание типов инцидентов

Название роли	Описание роли
incident_type_D	Удаление типов инцидентов
incident_type_R	Просмотр типов инцидентов
incident_type_U	Изменение типов инцидентов
incident_U	Изменение инцидентов
incident_users_U	Назначения пользователей для инцидента
knowledgebase_C	Создание записей базы знаний
knowledgebase_D	Удаление записей базы знаний
knowledgebase_R	Просмотр записей базы знаний
knowledgebase_U	Изменение записей базы знаний
reports_C	Создание отчетов
reports_D	Удаление отчетов
reports_R	Просмотр отчетов
reports_U	Изменение отчетов
scan_results_C	Загрузка результатов сканирования
scan_results_D	Удаление результатов сканирования
scan_results_R	Просмотр результатов сканирования
scan_results_U	Изменение результатов сканирования
service_assets_C	Создание активов, групп, интерфейсов, настройка идентификации
service_assets_D	Удаление активов, групп, интерфейсов, настройка идентификации
service_assets_R	Просмотр активов, групп, интерфейсов, настройка идентификации
service_assets_U	Изменение активов, групп, интерфейсов, настройка идентификации
software_compliance_checks_C	Создание правил и наборов правил оценки соответствия ПО
software_compliance_checks_D	Удаление правил и наборов правил оценки соответствия ПО
software_compliance_checks_R	Просмотр всех сущностей оценки соответствия ПО
software_compliance_checks_U	Изменение правил и наборов правил оценки соответствия ПО

2.2.9. Аудит действий пользователя

2.2.9.1. Общее описание

На вкладке «**Аудит действий**» администратору Платформы представляются функции просмотра всех совершаемых пользователями действий по внесению изменений в Платформу.

Для настройки просмотра используются следующие фильтры:

- По временному окну.
- По пользователю (по всем пользователям или по выбранному).
- По сервису (по всем сервисам или по выбранному).
- По сущности (фильтр доступен при выборе отдельного сервиса).

Для осуществления поиска действий необходимо выбрать временное окно и нажать кнопку «Поиск» (см. Рисунок 19). По умолчанию поиск действий будет произведен по всем пользователям и по всем сервисам.

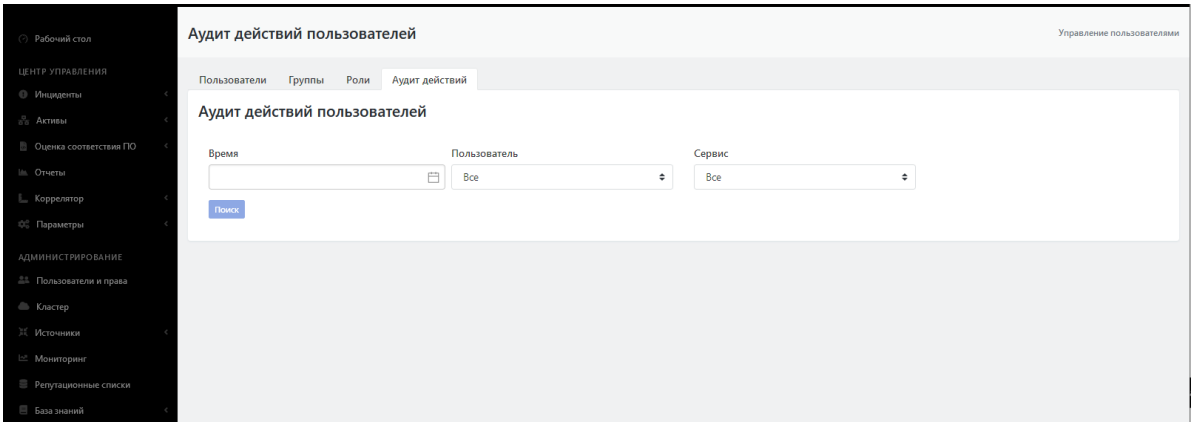

Рисунок 19 - Экран просмотра действий пользователя до ввода данных

2.2.9.2. Настройка списка действий пользователя

Для проведения поиска по действиям конкретного пользователя необходимо выбрать данного пользователя в раскрывающемся списке "Пользователь".

Для проведения поиска по действиям, связанным с конкретным сервисом, необходимо выбрать интересующий сервис в раскрывающемся списке "Сервис".

Результаты поиска представляют собой список действий пользователя (всех пользователей) на выбранном сервисе (или всех сервисах) за указанный период времени (см. Рисунок 20).

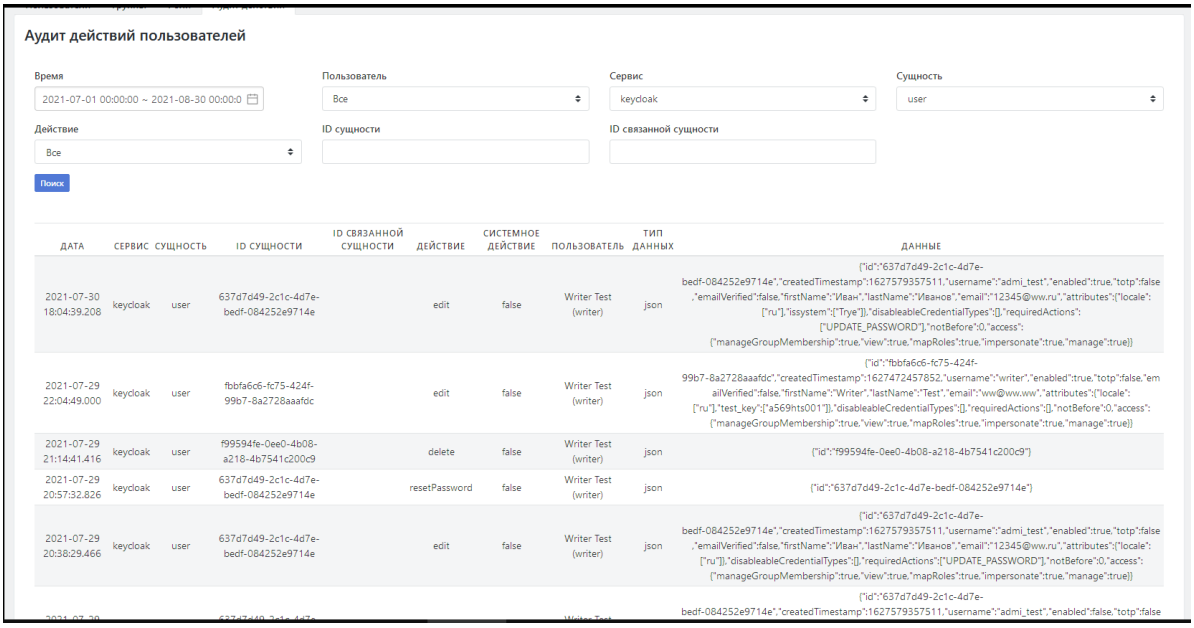

Рисунок 20 - Список действий пользователя

При выборе поиска по отдельному сервису добавляется фильтр по сущности, ассоциированной с сервисом. При выборе какой-либо сущности на экран добавляются возможности фильтрации по действиям сущности, ID сущности и ID связанной сущности (см. Рисунок 20).

2.2.9.3. Расширенный лог действий, связанных с авторизацией

Также есть расширенный лог действий связанный с авторизацией. Он доступен по адресу <http://<адрес сервера>: 8180/auth/admin/master/console/#/realms/master/events

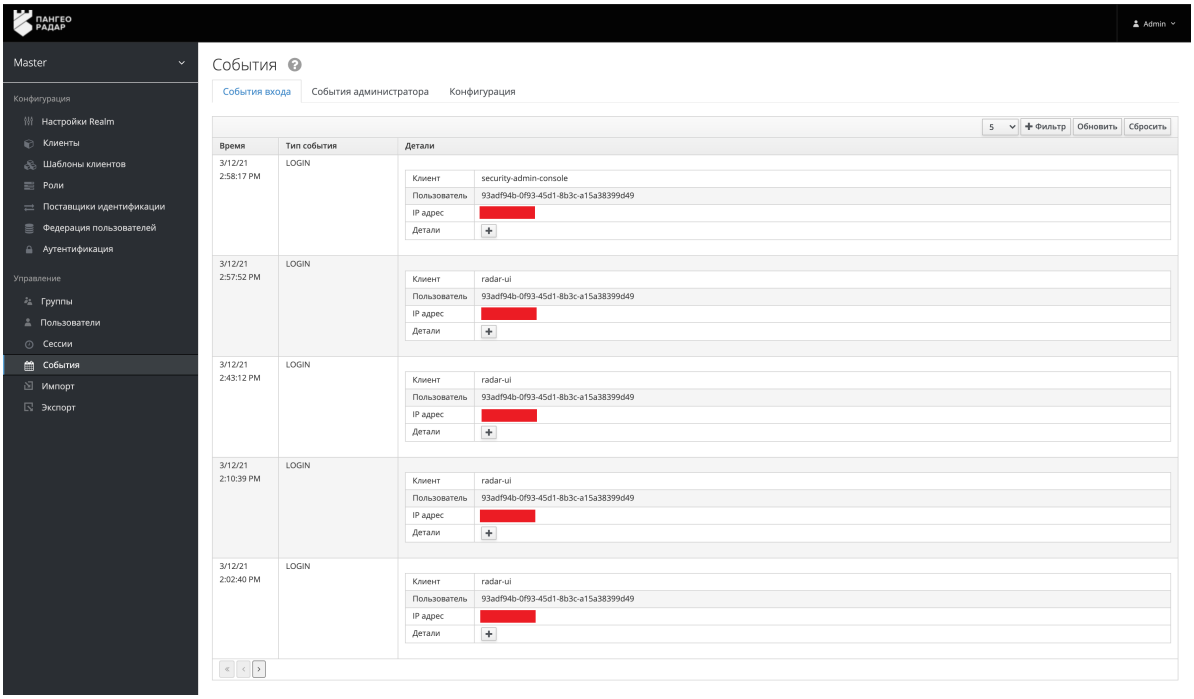

Рисунок 21 - Расширенный интерфейс управления пользователями, просмотр событий входа

Данный интерфейс является частью службы централизованной аутентификации Платформы и не предназначен для ручного администрирования без необходимости.

Для возврата в интерфейс Платформы воспользуйтесь переходом из пользовательского меню (Рисунок 20)

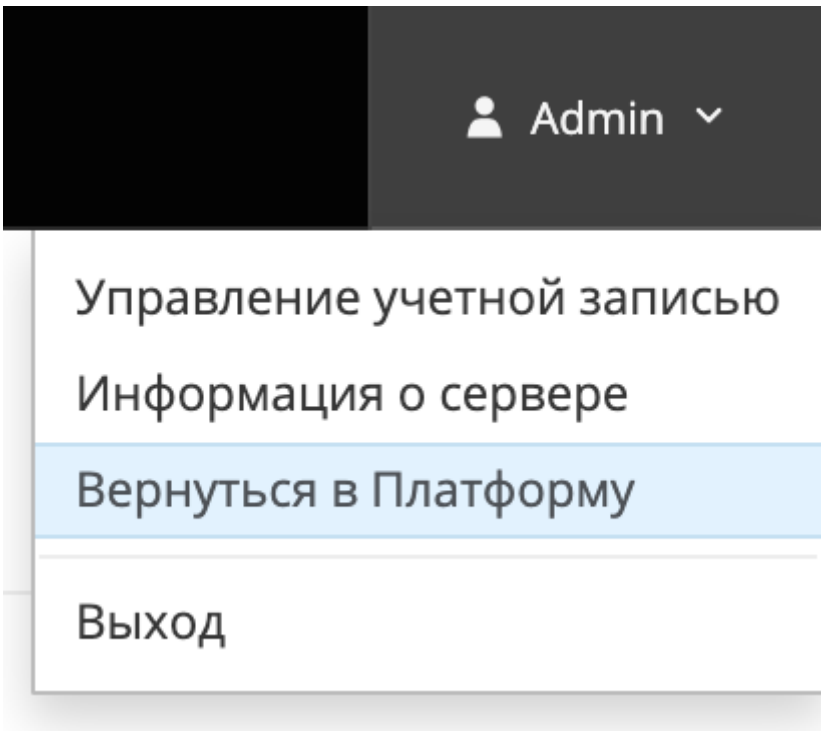

Рисунок 22 - пользовательское меню службы централизованной аутентификации.

3. Управление кластером Платформы

3.1. Управление кластером Платформы

Раздел «Кластер» предоставляет инструментарий для упрощенного управления Платформой без необходимости подключения к серверам через терминальные соединения.

Раздел предоставляет следующие возможности (см. Рисунок 23):

- управление узлами кластера и контроль состояния узлов;
- управление скриптами установки и конфигурационные файлами сервисов;
- управление учетными записями для сбора данных, протоколами и API ключами.

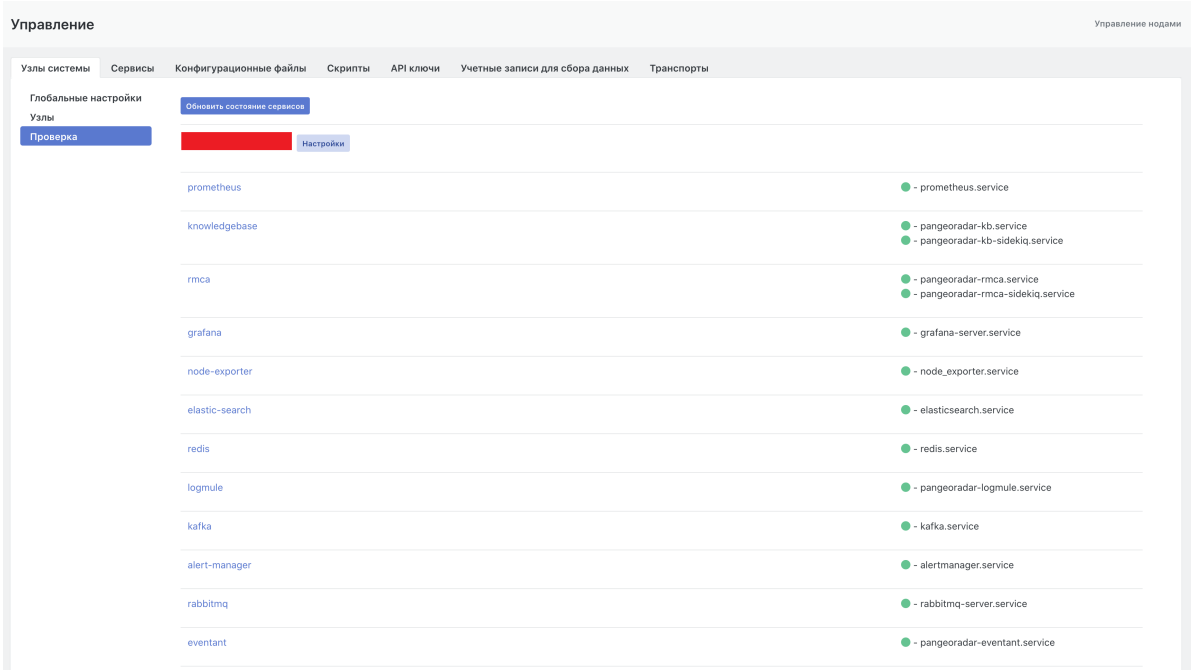

Рисунок 23 - Главный экран раздела «Кластер»

3.1.1. Концепция кластера Платформы Радар

Кластер Платформы Радар состоит из трех составляющих:

- Агент управления узлом кластера;
- Менеджер управления агентами;
- Интерфейс управления менеджером.

На каждый узел кластера необходимо установить Агент управления, через который будет осуществляться управление и контроль состояния узла.

Интерфейс и Менеджер управления агентами должны находиться на одном сервере, по умолчанию они находятся на сервере с ролью MASTER (подробнее в документе "Руководство по установке").

3.1.2. Добавление узла кластера

Для добавления нового узла должно быть соблюдено несколько подготовительных условий:

- Узел развернут и готов принимать внешние соединения.
- На узле установлена ОС - Debian 9 x64.
- На узле поднят ssh сервер.
- Узел разрешает соединения под привилегированным пользователем root с паролем.

После установки Агента, сервер может быть закрыт для подключения по ssh с паролем. При установке Агента на сервер прописывается доверенный сертификат мастер-сервера для подключения без ввода пароля. Пароль в системе не сохраняется.

Для добавления узла и начала установки Агента на узел необходимо выполнить следующие действия :

1. На вкладке "Узлы системы" выбрать подраздел "Узлы" (см. Рисунок 24).
2. Заполнить форму "Добавление нового узла". Все поля формы обязательны к заполнению.
3. Для создания кластера нажать на кнопку "Добавить".
4. Присвоить новому кластеру одну или несколько ролей, указав их в области "Добавление ролей к узлам". Для добавления роли необходимо выбрать нужную роль из списка и нажать

либо на кнопку «+», либо на кнопку «Добавить все» (см. Рисунок 24).

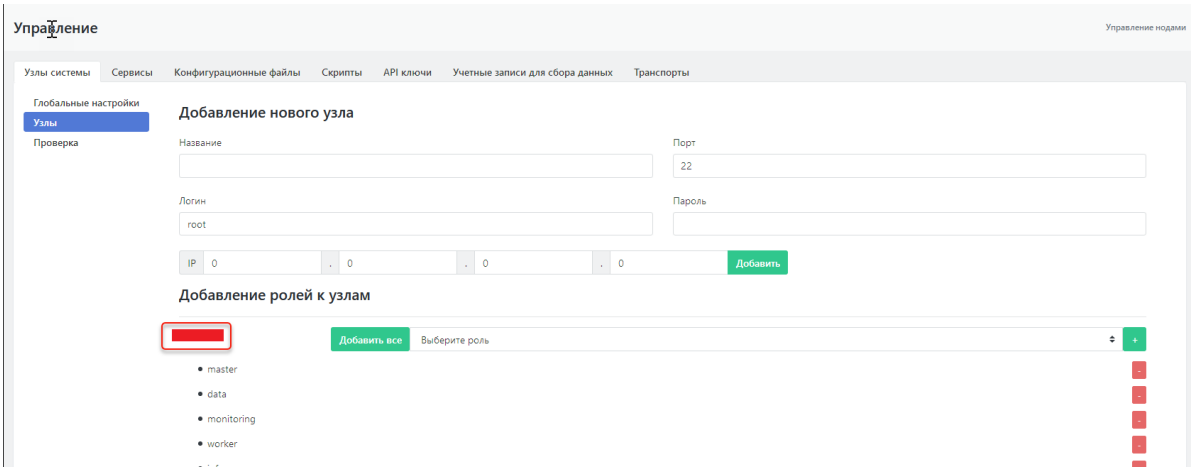

Рисунок 24 - Экран добавления узла Платформы

По умолчанию все необходимое программное обеспечение для узла регулируется ролью закрепленной за узлом (подробнее см. документ "Руководство по установке").

После добавления узла, назначение ему роли и установки Агента станет доступно управление узлом кластера, расположенное на специальном экране "Управление хостом <адрес хоста>" - см. раздел "Управление узлом кластера". Данный экран содержит функции редактирования роли кластера.

3.1.3. Управление узлом кластера

3.1.3.1. Экран управления узлом, общее описание

Управление узлом кластера осуществляется на отдельном экране интерфейса, перейти на который возможно следующим способом:

- из подраздела "Узлы", область "Добавление ролей к узлам" при нажатии на название узла (IP-адреса, см. Рисунок 24) ;
- из подраздела «Проверка» - нажать на кнопку «Настройка» рядом с названием узла (см. Рисунок 25).

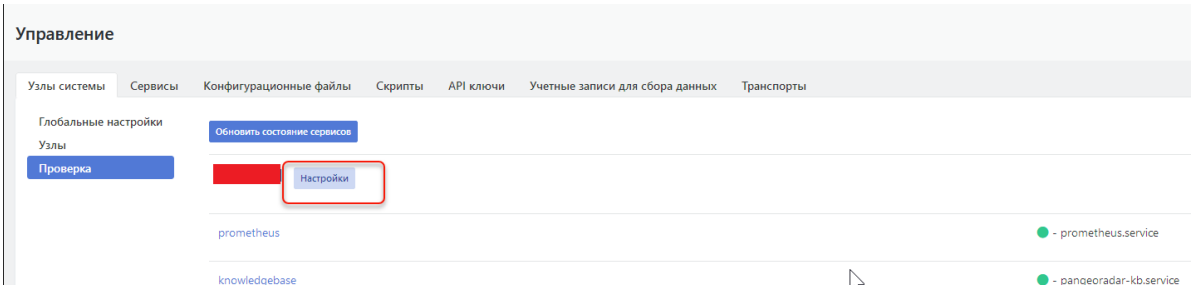

Рисунок 25 - Переход на экран управления узлом кластера из подраздела "Проверка"

Экран управления узлом предоставляет следующие возможности (см. Рисунок 26):

- управление серверными ролям узла;
- управление сервисами узла;
- контроль состояния работы сервисов;
- контроль работы узла;
- выполнение скриптов на узле.

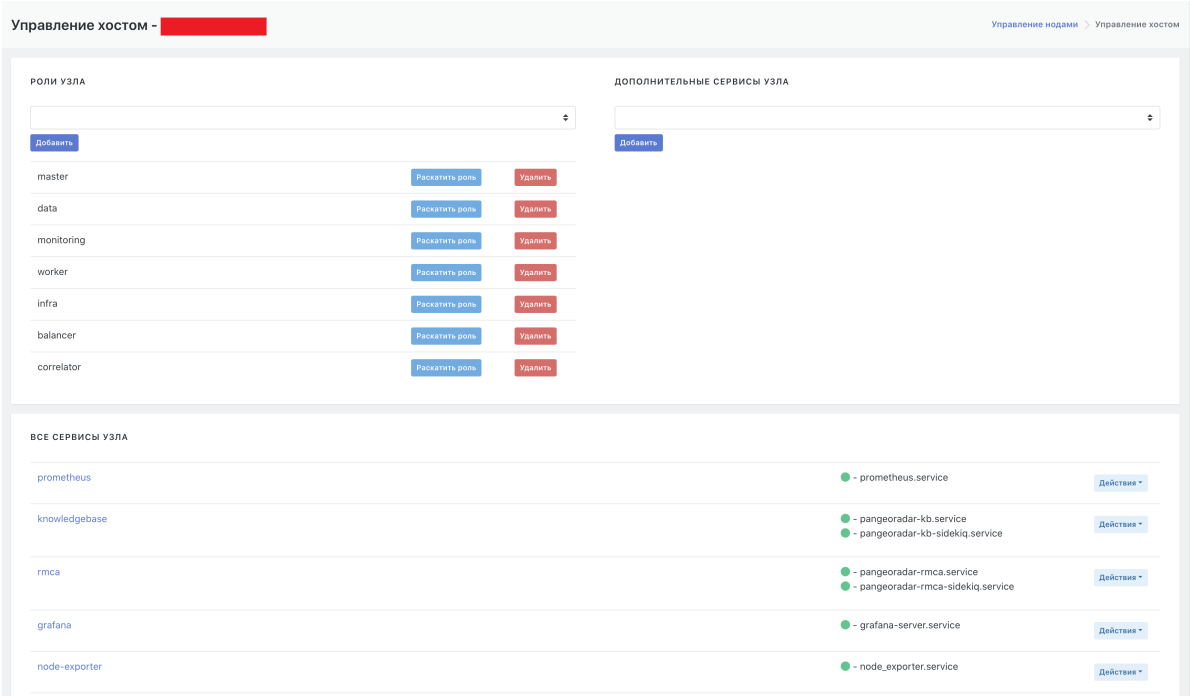

Рисунок 26 - Экран управления узлом

3.1.3.2. Управление сервисами узла кластера

Функции управления сервисами расположены в области "Все сервисы узла" (см. Рисунок 26). Каждый сервис узла кластера может предоставить информацию о своем статусе и последних логах. Для получения информации необходимо нажать на кнопку «Действия» и выбрать соответствующий пункт меню (см. Рисунок 27). Так же через данное меню доступны:

- переустановка сервиса;
- обновление конфигурационных файлов сервиса;
- перезапуск сервиса.

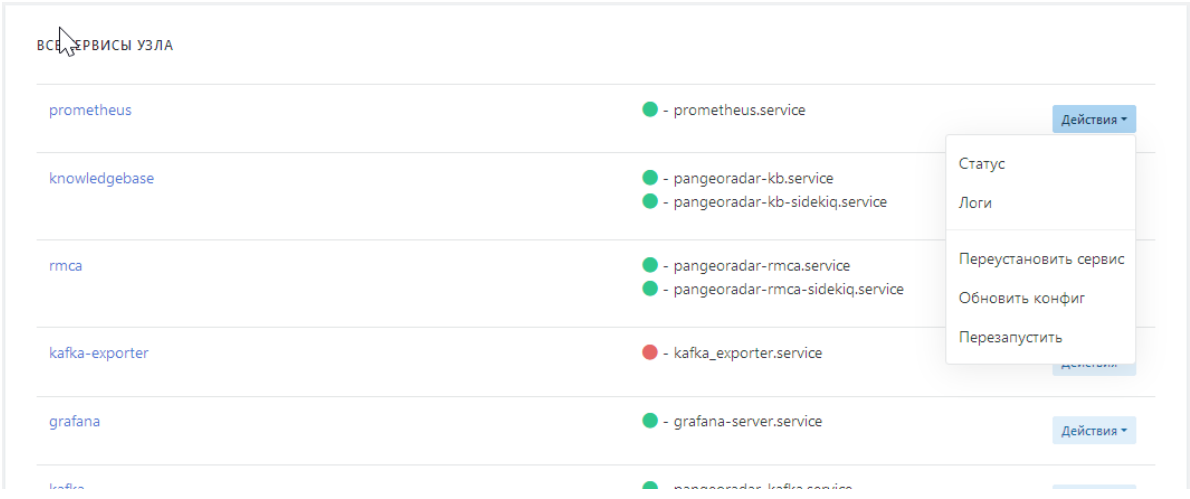

Рисунок 27 - Информация о статусе и последних логах выбранного узла кластера

3.1.3.3. Установка сервиса на узел кластера

Для установки на узел отдельно выбранного сервиса (не рекомендуется так делать) на экране управления узлом кластера необходимо выполнить следующие действий:

1. В области "Дополнительные сервисы узла" выбрать в раскрывающемся списке дополнительных сервисов необходимый сервис (см. Рисунок 28).

2. Нажать на кнопку «Добавить». Сервис добавится в общий список сервисов на узле в области "Все сервисы узла" .
3. В строке нового сервиса нажать на кнопку «Действия» и в раскрывшемся меню выбрать функцию «Переустановить сервис» (см. Рисунок 27). Дождаться завершения процесса.
4. В строке нового сервиса нажать на кнопку «Действия» и в раскрывшемся меню выбрать функцию «Обновить конфиг» (см. Рисунок 27). Дождаться завершения процесса.

В процессе переустановки сервиса будет показан лог действий установщика.

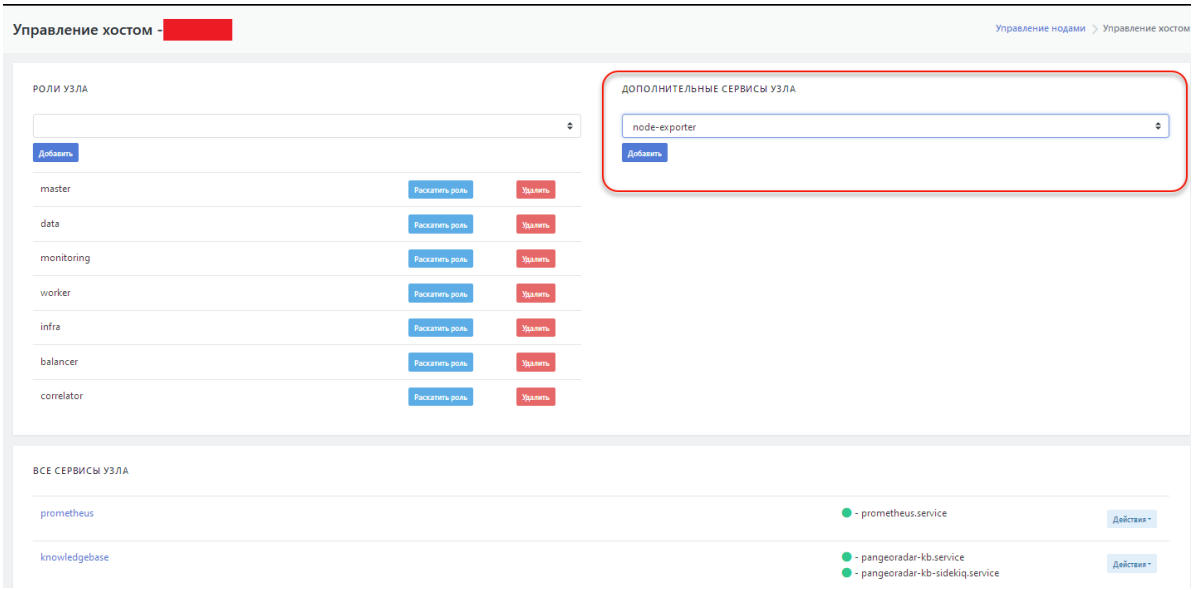

Рисунок 28 - Выбор дополнительного сервиса

3.1.3.4. Установка серверной роли на узел кластера

Рекомендуется использовать серверные роли как абстракцию установки программного обеспечения на узел кластера.

На экране управления узлом кластера для присвоение новой роли узлу необходимо выполнить следующие действия (см. Рисунок 28):

1. В области "Роли узла" в раскрывающемся списке выбрать нужную роль.
2. Нажать на кнопку «Добавить». Выбранная роль будет добавлена к списку ниже.
3. В строке добавленной роли нажать на кнопку «Раскатить роль» для запуска процесса установки на узел кластера ПО, соответствующего роли.

Процесс установки представлен в виде обновляемого лога установки.

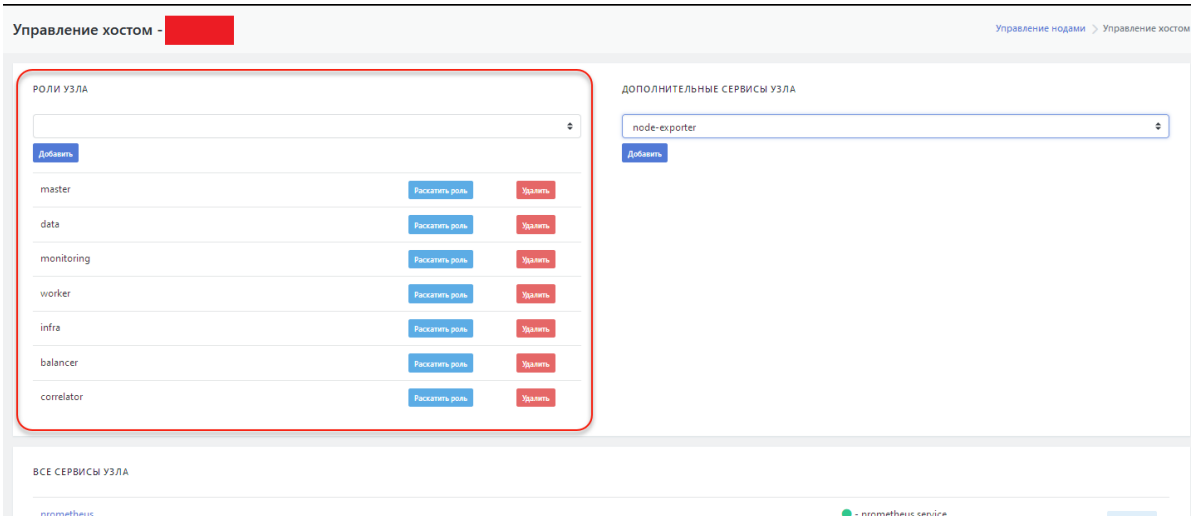

Рисунок 29 - Установка серверной роли на узел кластера

3.1.4. Управление сервисами

3.1.4.1. Набор сервисов, добавление/удаление сервисов

В интерфейсе Платформы Радар набор доступных сервисов платформы отображается в разделе "Управление кластером" на вкладке "Сервисы" в виде таблицы. Область справа от таблицы позволяет добавлять новые сервисы (см. Рисунок 30).
Можно удалить существующий сервис из таблицы нажав на кнопку "Удалить" в строке соответствующего сервиса.

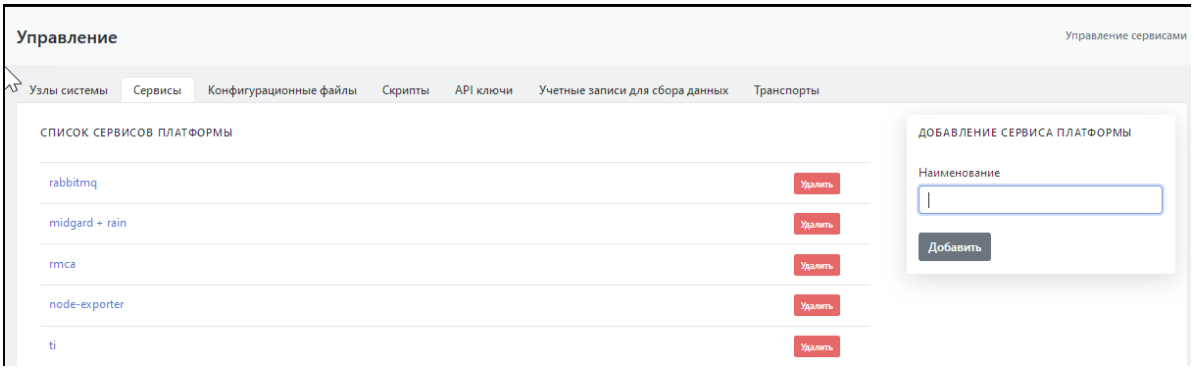

Рисунок 30 - Набор сервисов Платформы

3.1.4.2. Экран управления сервисами

При выборе в таблице интересующего сервиса открывается экран управления данным сервисом, который содержит следующие области задач (см. Рисунок 31):

- область "**Ассоциирован с ролями**" - обеспечивает настройку списка ролей, с которыми ассоциирован данный сервис;
- область "**Ассоциированные конфиги**" - обеспечивает настройку списка конфигурационных файлов, ассоциированных с данным сервисом;
- область "**Доступные скрипты**" - содержит перечень скриптов, доступных для данного сервиса;
- область "**Ассоциирован с узлами**" - содержит список названий (ролей) кластерного узла с которыми ассоциирован данный сервис.

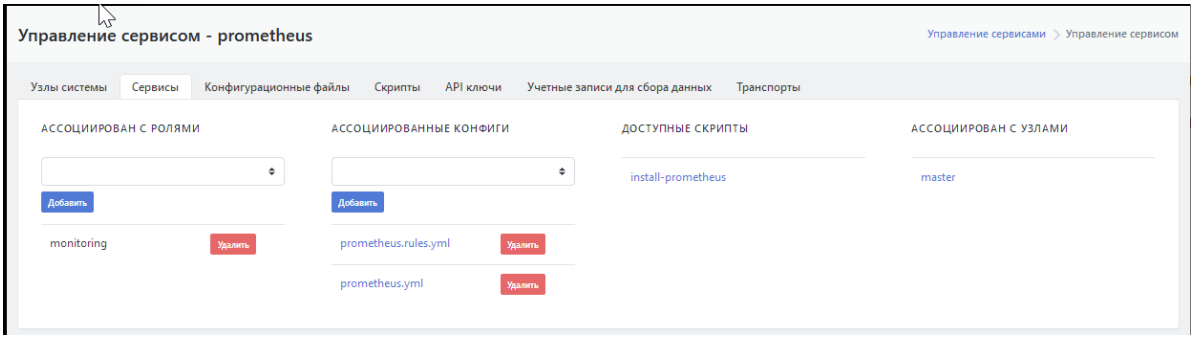

Рисунок 31 - Экран управления сервисами

3.1.4.3. Настройка списка ролей, с которыми ассоциирован сервис

Для проведения ассоциации сервиса с новой ролью необходимо:

1. Перейти на вкладку «Сервисы».
2. Выбрать нужный сервис в таблице. Откроется форма настроек сервиса (см. Рисунок 32).

3. В области "**Ассоциирован с ролями**" и выбрать нужную роль в раскрывающемся списке.
4. Нажать кнопку «Добавить».

Указанная роль отобразится в списке ролей, с которыми ассоциирован выбранный сервис.

Для удалении роли из списка нажать на кнопку "Удалить" в строке данной роли.

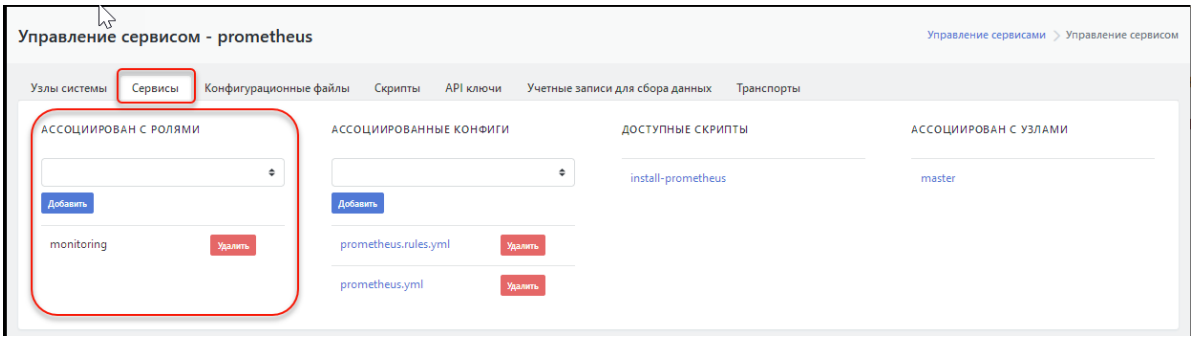

Рисунок 32 - Область настройки ролей, с которыми ассоциирован сервис

3.1.4.4. Настройка списка конфигурационных файлов, ассоциированных с сервисом

Для ассоциации конфигурационного файла с сервисом, необходимо:

1. Перейти на вкладку «Сервисы».
2. Выбрать нужный сервис в таблице. Откроется форма настроек сервиса (см. Рисунок 33).
3. Выбрать настройку «**Ассоциированные конфиги**» и выбрать нужный конфигурационный файл в раскрывающемся списке.
4. Нажать кнопку «Добавить».

Выбранный конфигурационный файл отобразится в списке конфигурационных файлов, ассоциированных с сервисом.

Для удаления конфигурационного файла из списка нажать на кнопку "Удалить" в строке данного файла.

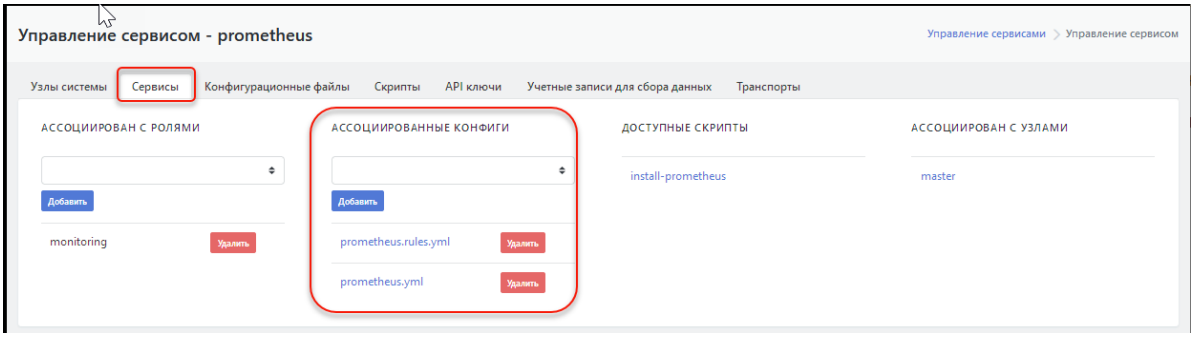

Рисунок 33 - Область настройки конфигурационных файлов ассоциированных с сервисом

3.1.5. Управление конфигурационными файлами кластера

3.1.5.1. Набор конфигурационных файлов, добавление/удаление файлов

Внимание! Не рекомендуется вносить изменения в конфигурационные файлы без консультации с разработчиками.

Все конфигурационные файлы сервисов Платформы лежат по адресу: /opt/pangeoradar/configs/.

В интерфейсе Платформы Радар данный набор конфигурационных файлов отображается в разделе "Управление кластером" на вкладке "Конфигурационные файлы" в виде таблицы. Область справа от таблицы позволяет переопределить конфигурационные файлы по умолчанию из этой или вложенных директорий путем создания одноименных файлов (см. Рисунок 34).

В названии файла допускается использовать относительный путь от директории хранения конфигурационных файлов, например - termite/test.conf.

Для удаления конфигурационного файла из списка нажать на кнопку "Удалить" в строке интересующего файла.

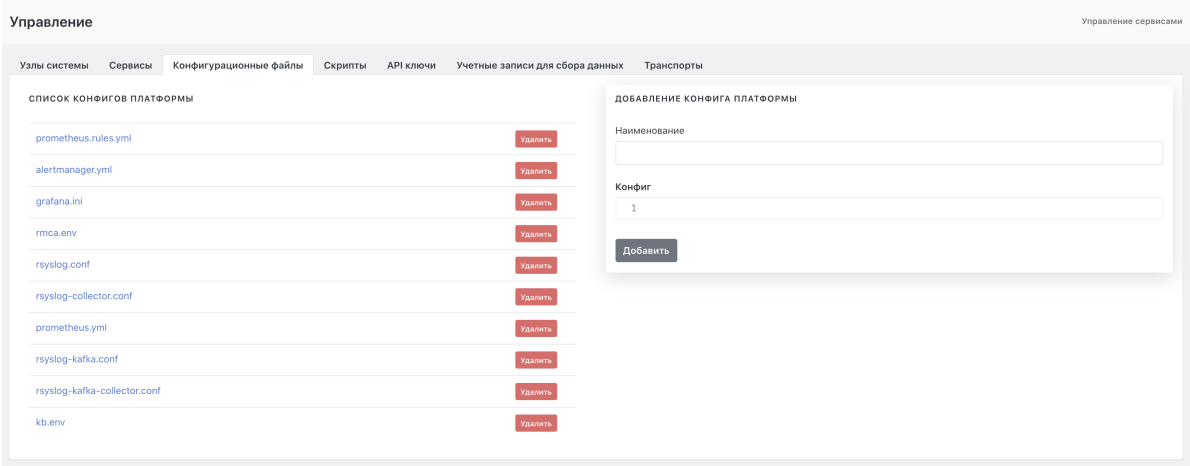

Рисунок 34 - Набор конфигурационных файлов Платформы

3.1.5.2. Экран редактирования конфигурационного файла

Внимание! Не рекомендуется вносить изменения в конфигурационные файлы без консультации с разработчиками.

При необходимости текст конфигурационного файла можно отредактировать. Для это надо надо выбрать интересующий файл в списке конфигурационных файлов. На экране откроется текст выбранного файла, доступный для редактирования (см. Рисунок 35). Введенные изменения сохраняются при нажатии на кнопку "Изменить".

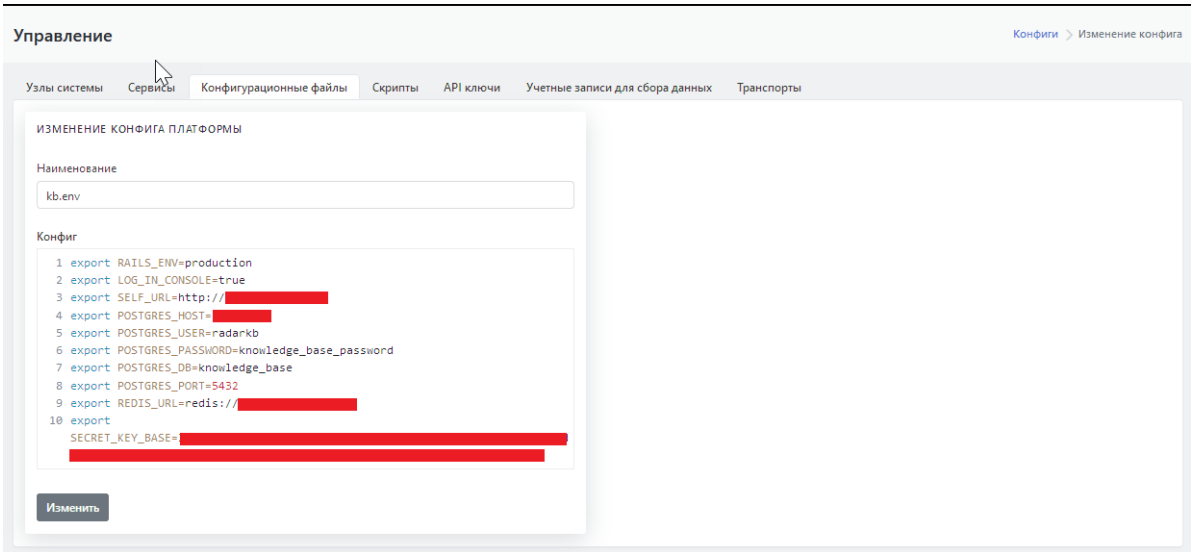

Рисунок 35 - Редактирование конфигурационного файла Платформы

3.1.5.3. Ассоциация конфигурационного файла с сервисом

См. раздел "Настройка списка конфигурационных файлов, ассоциированных с сервисом".

3.1.6. Управление инсталляционными скриптами кластера

3.1.6.1. Набор скриптов, добавление/удаление скриптов

Внимание! Не рекомендуется вносить в скрипты изменения без консультации с разработчиками.

Инсталляционные скрипты необходимы при установке или переустановке сервисов.

Языком описания скрипта является bash.

В интерфейсе Платформы Радар набор скриптов отображается в разделе "Управление кластером" на вкладке "Скрипты" в виде таблицы. Область справа от таблицы позволяет переопределить скрипты по умолчанию (см. Рисунок 36).

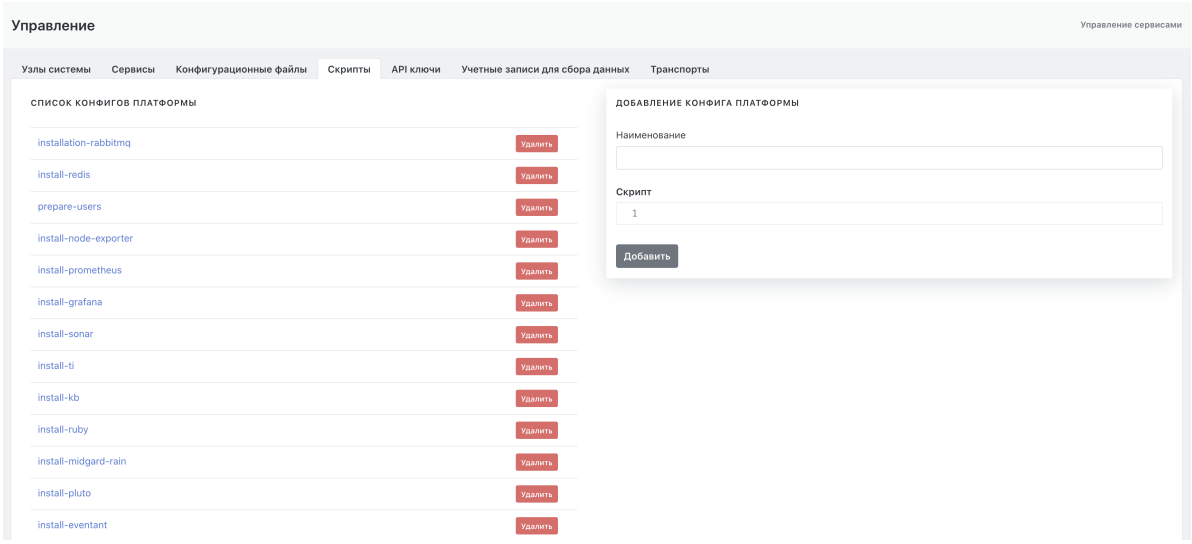

Рисунок 36 - Набор скриптов Платформы

3.1.6.2. Экран редактирования скрипта

При выборе в таблице интересующего скрипта открывается экран управления данным скриптом, который содержит следующие области задач (см. Рисунок 37):

- слева расположен текст выбранного скрипта, доступный для редактирования.;
- область "**Ассоциация скрипта с ролью**" - обеспечивает ассоциацию скрипта с ролью, выбранной из раскрывающегося списка;
- область "**Ассоциация скрипта с сервисом**"- содержит список сервисов, с которыми ассоциирован данный скрипт на текущий момент, и раскрывающийся список сервисов, с которыми можно провести ассоциацию данного скрипта.

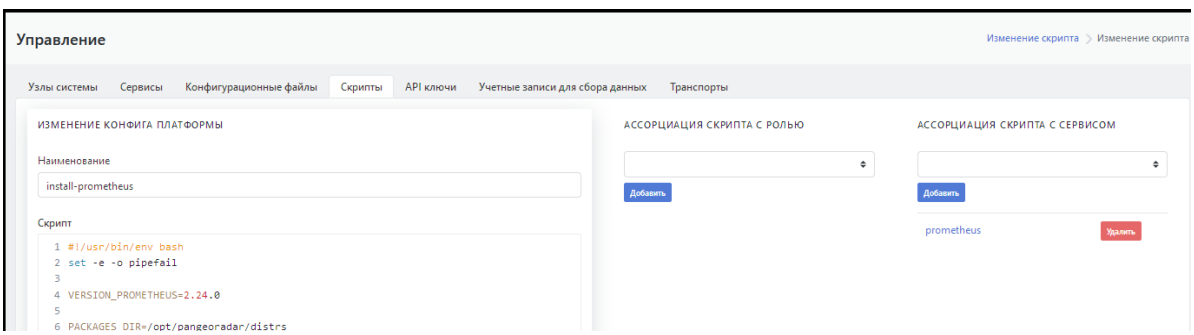

Рисунок 37 - Редактирование скрипта Платформы

3.1.7. Управление API ключами кластера

Доверенные ключи API используются для меж-сервисного взаимодействия. Управление ключами API реализовано на вкладке "API ключи". Экран содержит (см. Рисунок 38):

- текущий список API ключей;
- область добавления новых ключей.

Не рекомендуется удалять ключ global_api_key во избежание потери работоспособности Платформы.

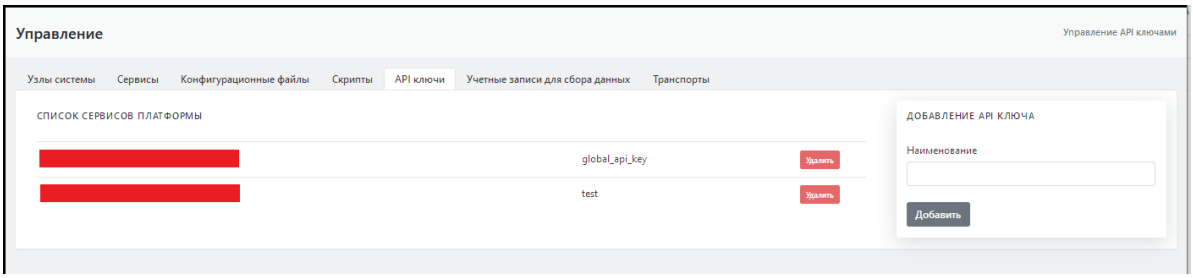

Рисунок 38 - Управление API ключами

3.1.8. Управление учетными записями для сбора данных

На вкладке "Учетные записи для сбора данных" реализовано управление авторизационными данными для сборщика данных с хостов при процедуре инвентаризации (подробнее см. документ "Руководство пользователя"). Вкладка содержит (см. Рисунок 39):

- область "**Список учетных записей**" - текущий список учётных записей;
- область **"Добавление учетной записи"** - содержит форму для введения параметров новой учетной записи.

Изменение созданной записи не предусмотренно. Возможно только удаление и повторное создание.

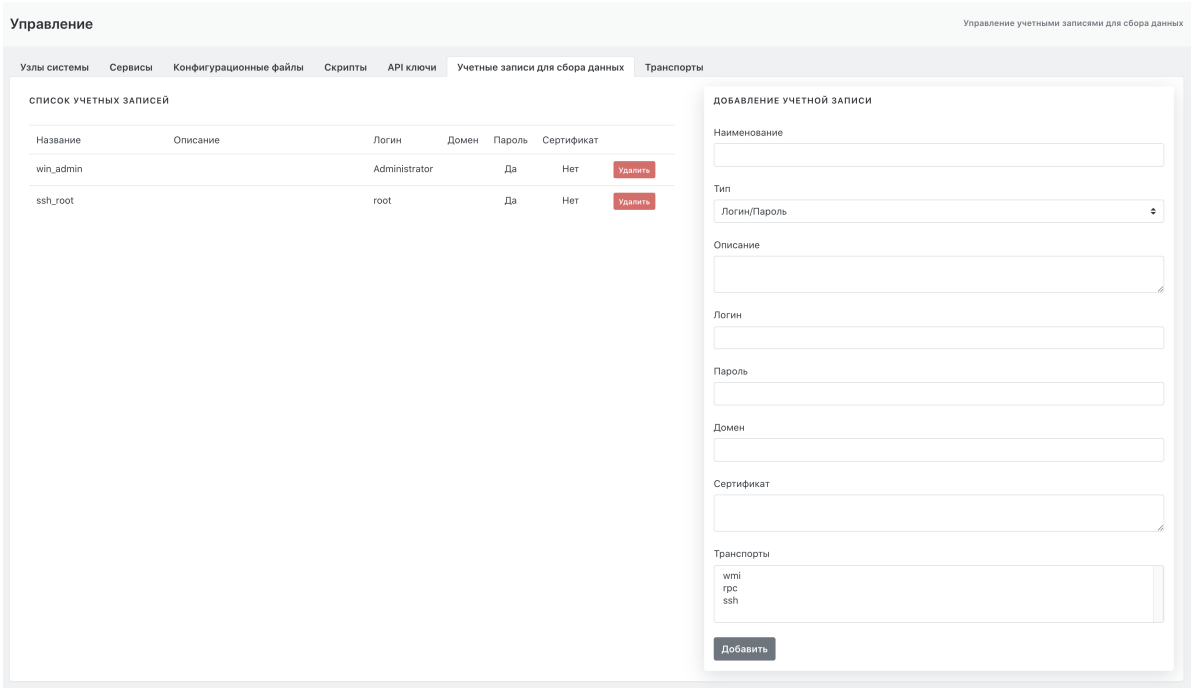

Рисунок 39 - Управление списком учетных записей для сбора данных

3.1.9. Управление транспортами сбора данных

Внимание! Не рекомендуется вносить изменения в набор транспортов без консультации с разработчиками.

На вкладке "Транспорт" реализовано управление словарем для формы добавления учетных записей для сбора данных (см. Рисунок 39). Вкладка содержит (см. Рисунок 40):

- область "**Список транспортов платформы**" - текущий список транспортов с возможностью удаления транспорта из списка;
- область **"Добавление транспорта"** - содержит форму для добавления нового транспорта .

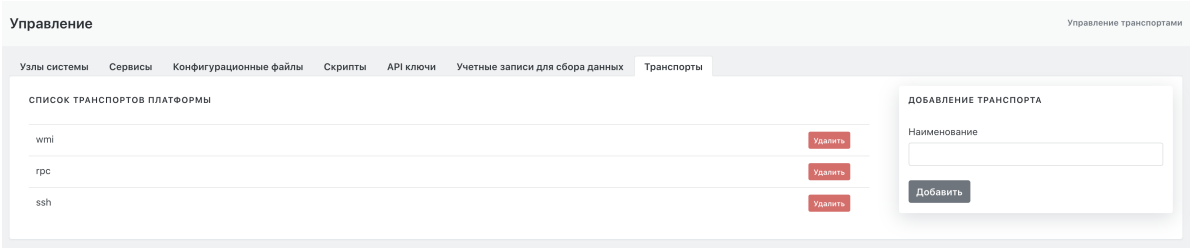

Рисунок 40 - Управление транспортами сбора данных

4. Управление источниками событий

4.1. Управление источниками событий

4.1.1. Общее описание

Раздел "Источники" отвечает за управление подключением источников событий к Платформе.

Содержит следующие вкладки:

- "**Источники**" - ;
- "**Шаблоны**" - ;
- "**Нормализаторы**" - ;
- "**Парсеры**" - ;;
- "**Grok паттерны**" - .

4.1.2. Управление источниками

Вкладка "Источники" обеспечивает выполнение следующих действий(см. Рисунок 41):

- Добавление нового источника;
- Включение отключение существующего;
- Изменение параметров существующего источника;
- Синхронизация конфигурации со всеми сервисами Платформы;
- Управление файрволом на основе данных по подключенным источникам;
- Удаление подключенных источников.

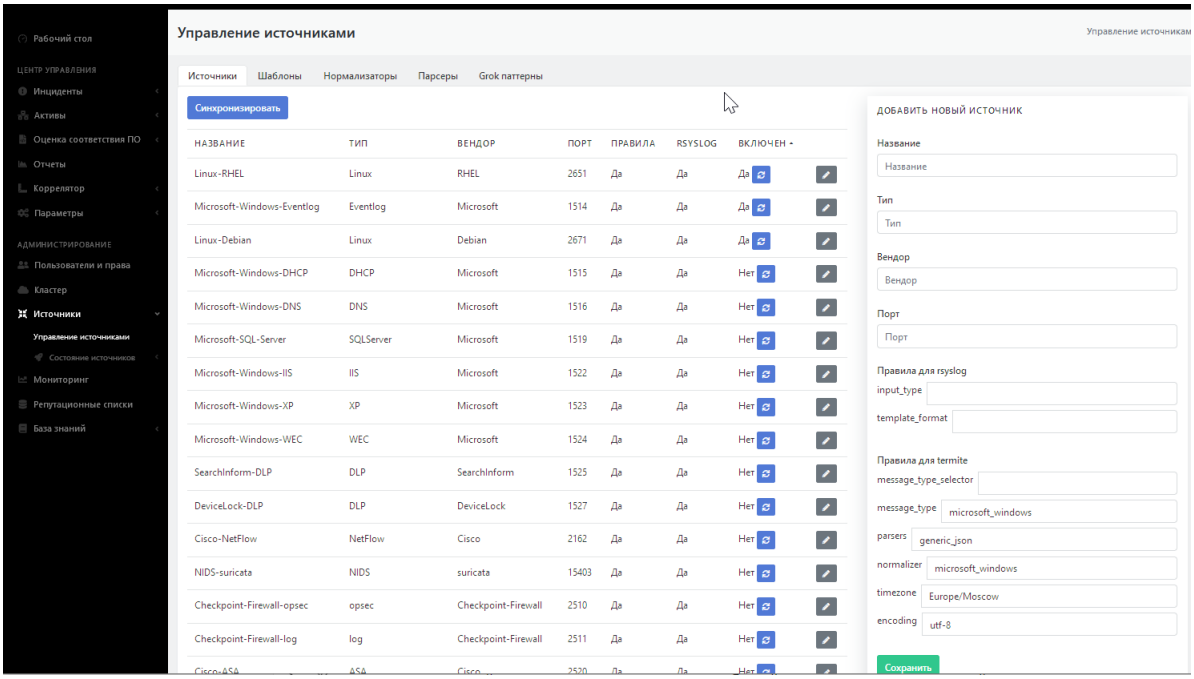

Рисунок 41 - Вкладка "Источники"

Подробная инструкция по управлению источниками приведена в отдельном документе "Руководство по подключению источников".

4.1.3. Контроль состояния источников

Подраздел «Состояние источников» в разделе «Источники» отвечает за мониторинг состояния источников событий и создание
уведомлений (как внутрисистемных, так и на электронную почту) в случае остановки потока событий на источнике в указанный диапазон времени (см. Рисунок 42).

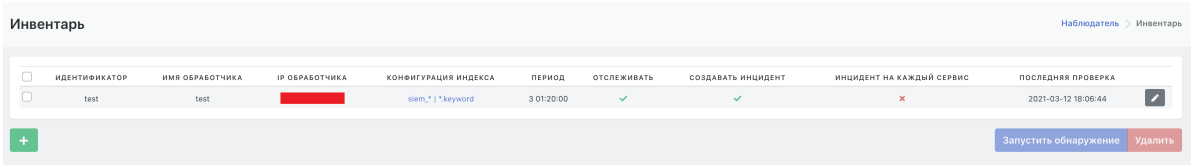

Рисунок 42 - Данные по состоянию источников

Подробная инструкция по Контролю состояния источников приведена в отдельном документе "Руководство по подключению источников".

5. Мониторинг работы Платформы

5.1. Общее описание

Раздел интерфейса **"Мониторинг"** содержит наборы интегрированных в интерфейс Платформы Радар приборных панелей (дашбордов) Grafana. В Платформе организованы следующие наборы приборных панелей (см. Рисунок 43):

- **Общий мониторинг** — мониторинг основных параметров Платформы;
- **Поток событий** — мониторинг параметров потока событий;
- **MongoDB** — мониторинг параметров СУБД MongoD;

- **RabbitMQ** — мониторинг параметров брокера сообщений RabbitMQ;
- **Kafka** — мониторинг параметров системы обмена сообщениями Kafka;
- **ElasticSearch** — мониторинг параметров поисковой системы ElasticSearch.

Необходимый набор приборных панелей выбирается из раскрывающегося списка (см. Рисунок 43).

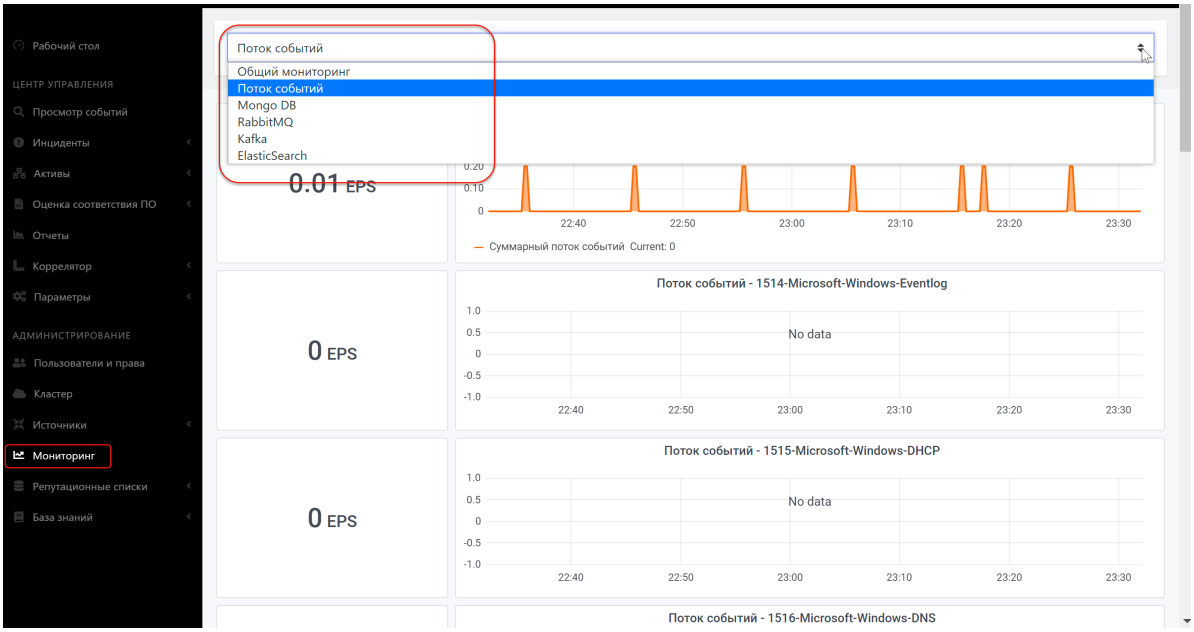

Рисунок 43 - Выбор набора приборных панелей

В разделе **"Мониторинг"** доступно переключение между встроенными в систему приборными панелями без необходимости заходить в интерфейс Grafana.

> Grafana относится к свободно распространяемому ПО. Подробную информацию о продукте можно посмотреть на сайте https://grafana.com/.

5.2. Набор приборных панелей «Общий мониторинг»

Набор приборных панелей **Общий мониторинг** предназначен для мониторинга основных параметров работы Платформы, таких как (см. Рисунок 44):

- мониторинг метрик потребления памяти — виджеты *Ram Used* (текущее потребление памяти), *Memory Basic* (график потребления памяти).
- мониторинг метрик загрузки процессора — виджеты *CPU Busy* (текущая загрузка процессора), *CPU Basic* (график загрузки процессора).
- мониторинг метрик состояния дискового пространства — виджеты *Root FS Used* (текущее состояние дискового пространства), *Disk Space Used Basic* (график загрузки дискового пространства).

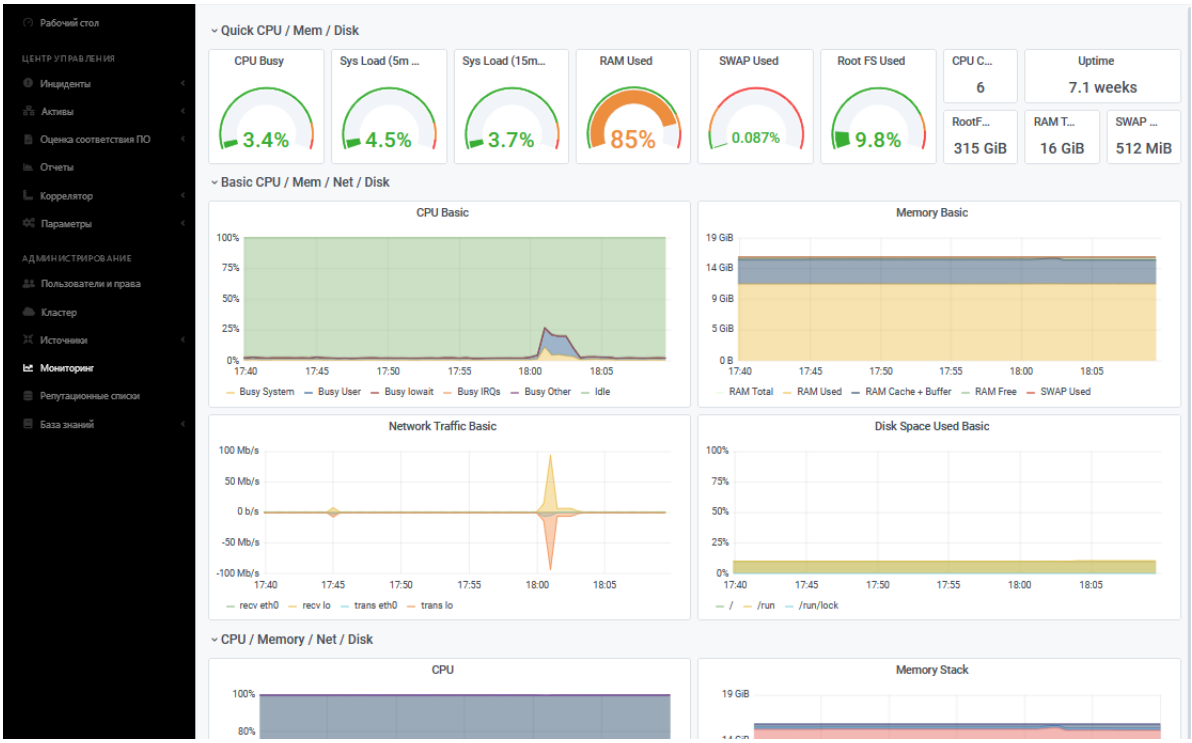

Рисунок 44 - Приборные панели из набора панелей "Общий мониторинг"

Предустановленный список приборных панелей в наборе **"Основной мониторинг"** приведен на Рисунке 3. При щелчке по названию приборной панели можно открыть/скрыть набор графиков/диаграмм, входящих в приборную панель.

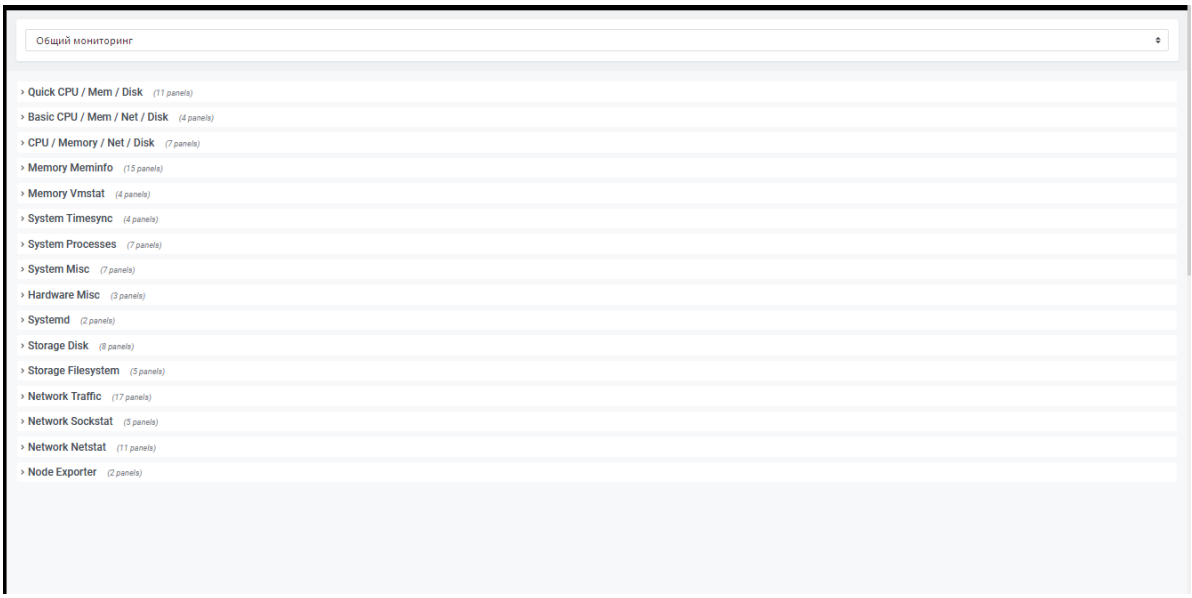

Рисунок 45 - Предустановленный список приборных панелей Платформы Радар

5.3. Приборная панель «Поток событий»

Приборная панель **Поток событий** предназначена для мониторинга метрик обрабатываемых событий в секунду — EPS — и содержит два типа виджетов (см. Рисунок 46):

- виджет с отображением информации о текущем потоке событий (слева);
- виджет по потоку событий в виде линейных графиков, построенных на основе исторических данных (справа).

Первыми отображаются метрики текущего EPS в системе — **Суммарный поток событий** (см. Рисунок 46). Далее следуют виджеты, где предоставляется информация по потокам от каждого из подключенных источников событий.

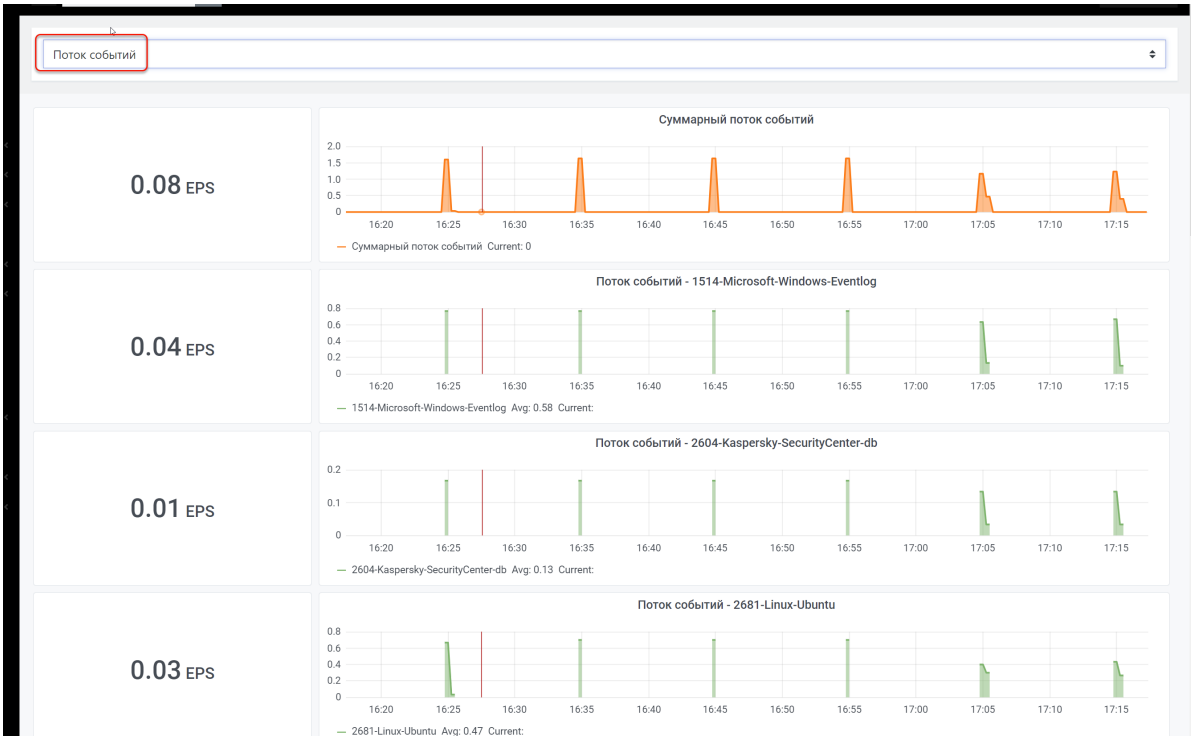

Рисунок 46 - Приборные панели из набора панелей "Суммарный поток событий"

5.4. Работа с графиками и диаграммами приборных панелей

При щелчке справа от названия графика/диаграммы открывается меню (см. Рисунок 47):

- "**View**" — раскрытие графика/диаграммы на весь экран Платформы.
- "**Share**" — поделиться панелью в виде прямой ссылки, снимка или встроенной ссылки.
- "**Inspect**" — корректировка запросов и устранение неполадок.
- "**More**" (toggle legend) — отображение/скрытие на графике легенды.

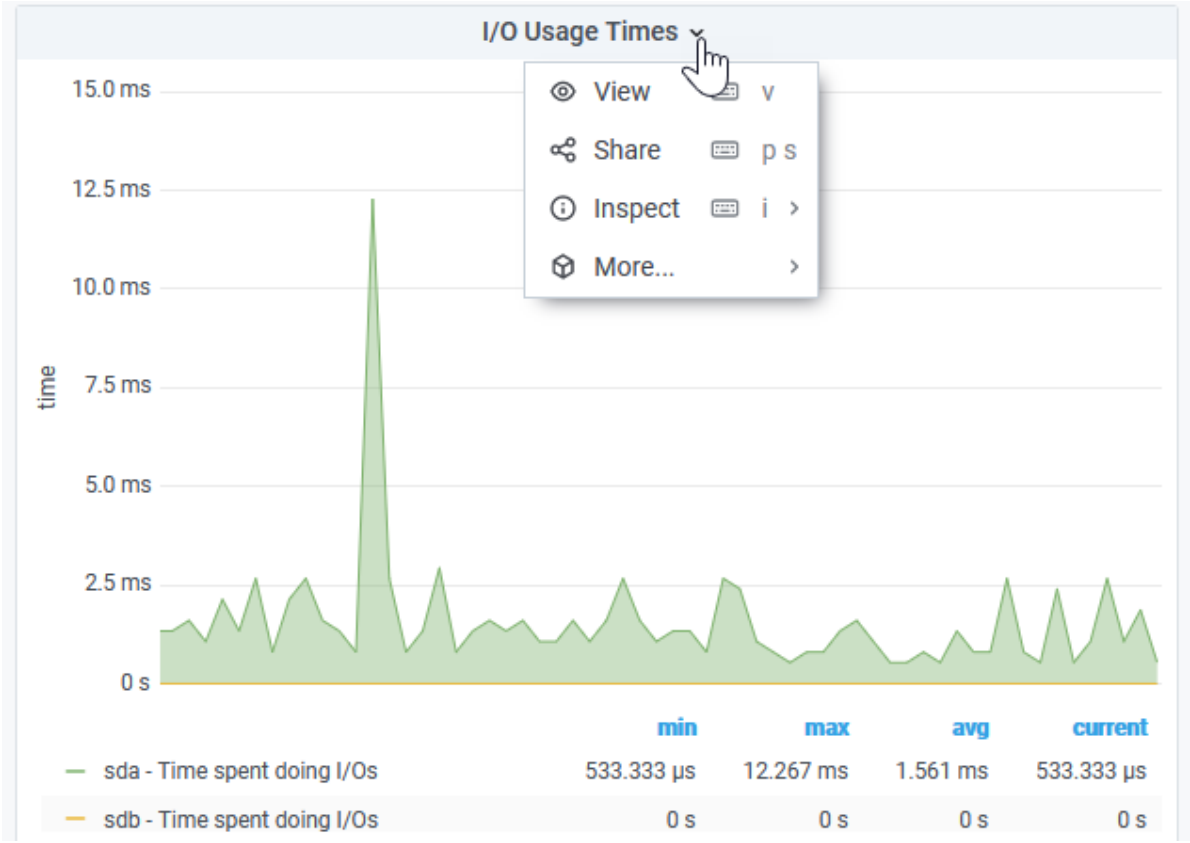

Рисунок 47 - Работа с графиком/ диаграммой

При наведении курсора на график открывается окно с данными точки, указанной курсором (см. Рисунок 48).

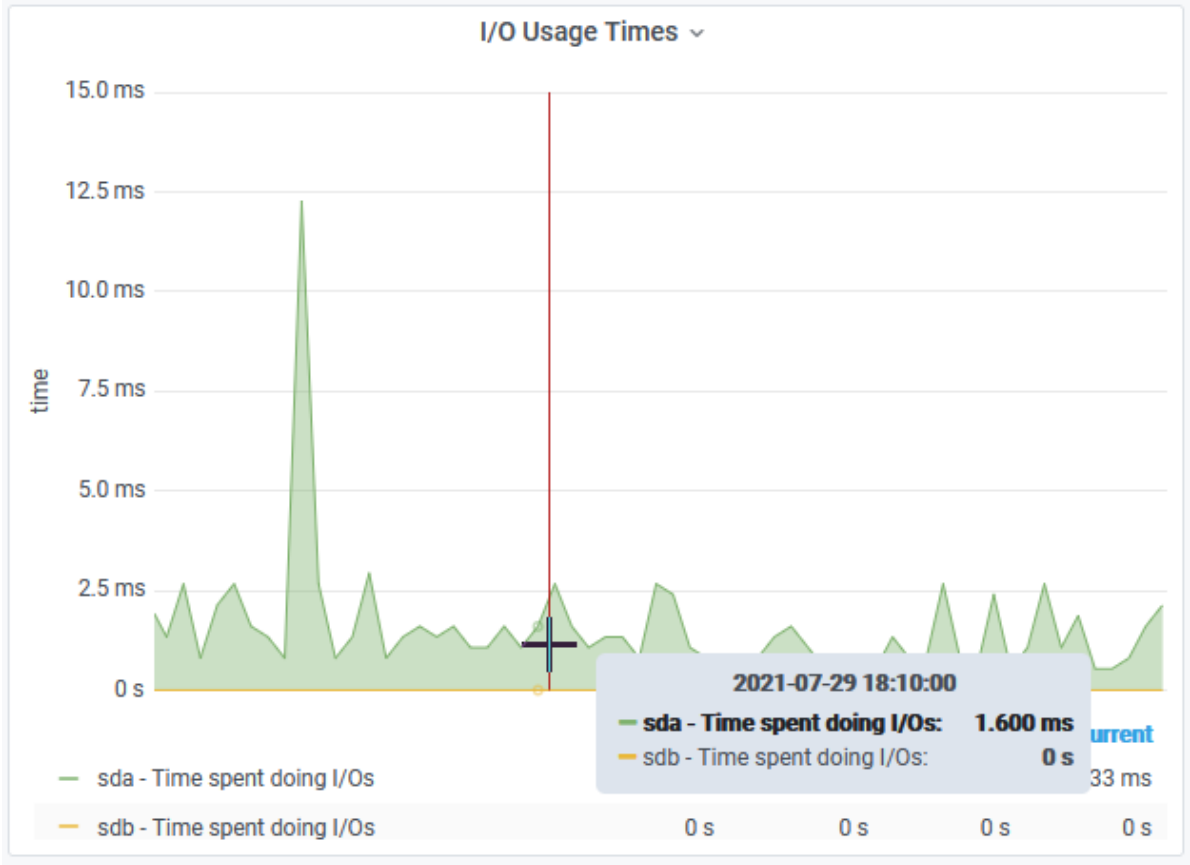

Рисунок 48 - Просмотр данных на графиках

5.5. Передача метрик производительности во внешние системы мониторинга

В Платформе предусмотрена возможность передачи метрик производительности во внешние системы мониторинга.

Платформа обеспечивает многострочный вывод метрик производительности в формате строки *Prometheus* (ключ, значение), что позволяет экспортировать метрики в систему Zabbix (https://www.zabbix.com/documentation/current/ru/manual/config/items/itemtypes/prometheus).

6. Репутационная база

6.1. Назначение репутационной базы

Данный модуль предназначен для обогащения событий данными, полученными из различных репутационных списков.

6.2. Состав репутационной базы

Репутационная база представлена службой `pangeoradar-ti-updater.service`

6.3. Работа с репутационными списками из UI

6.3.1. Репутационные списки

Для просмотра и управления репутационными списками необходимо перейти в раздел "Репутационные списки", "Репутационные списки".

На рисунке 49 изображено окно просмотра и управления репутационными списками.

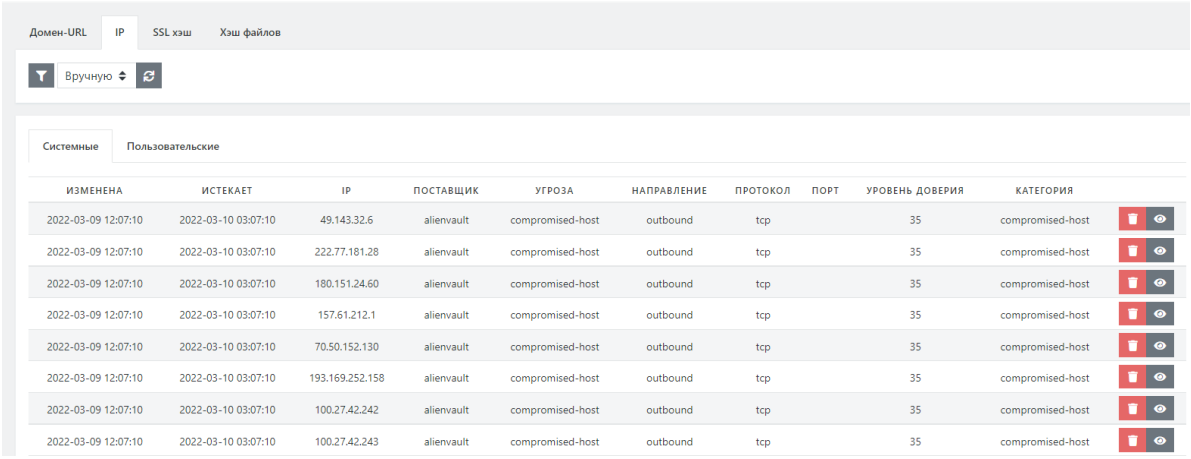

Рисунок 49 - "Репутационные списки"

В рассматриваемом интерфейсе присутствуют следующие функциональные возможности:

- Просмотр полного списка индикаторов компрометации;
- Фильтрация записей в репутационных списках;

  Для этого необходимо нажать на пиктограмму фильтра, выбрать поле, по которому необходимо отфильтровать записи, а также значение этого поля (полностью или частично), как изображено на рисунке 50.

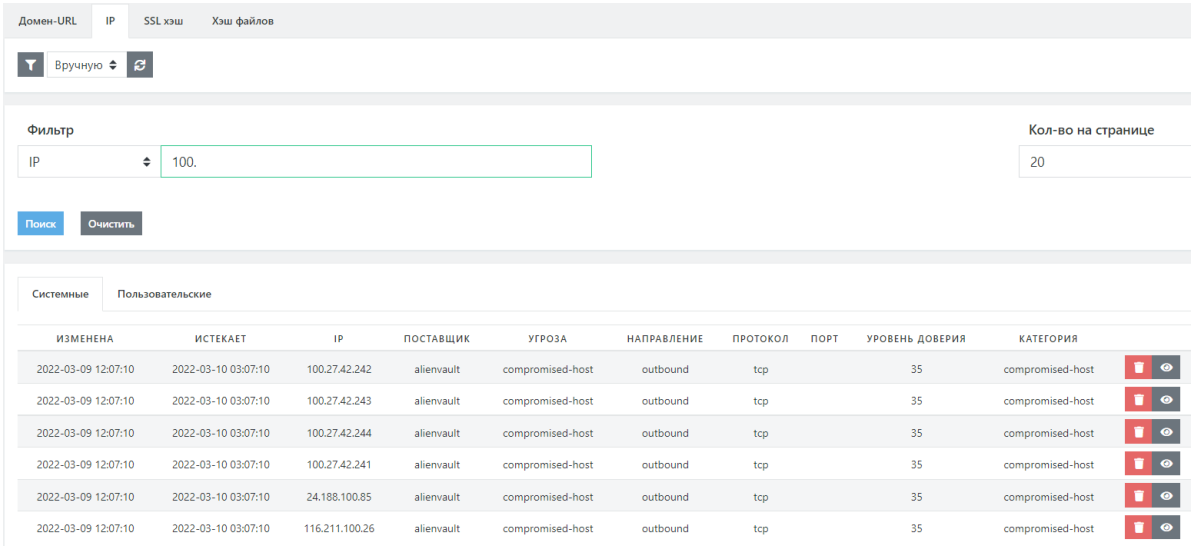

Рисунок 50 - "Фильтрация репутационнных записей"

- Создание пользовательских индикаторов компрометации;

  Для создания необходимо перейти во вкладку "Пользовательские", после чего нажать на кнопку "+".
  В открывшемся окне заполнить значения полей и нажать "Сохранить"

  Пример пользовательского индикатора компрометации представлен на рисунке 51

Рисунок 51 - "Создание пользовательского индикатора компрометации"

- Удаление индикаторов компрометации;

  Для удаления индикаторов компрометации необходимо нажать на пиктограмму урны справа от индикатора компрометации, который необходимо удалить.

6.3.2. Источники IOC

Для просмотра и управления источниками идентификаторов компрометации необходимо перейти в раздел "Репутационные списки", "Источники IOC".

На рисунке 52 изображено окно просмотра и управления источниками идентификаторов компрометации.

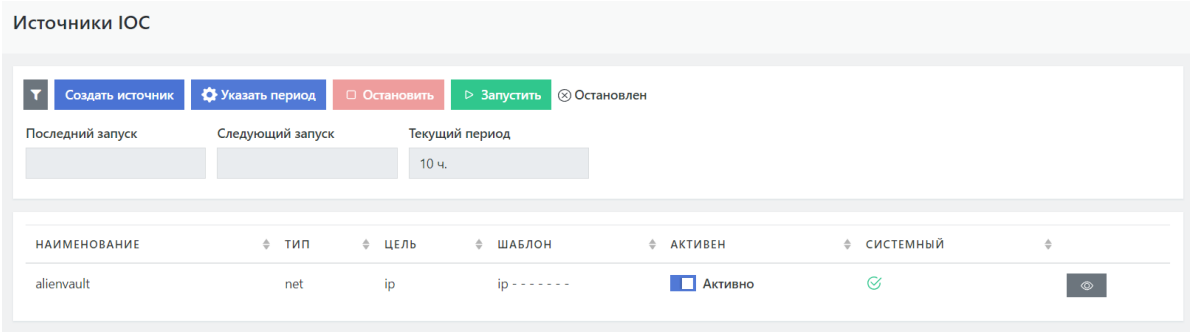

Рисунок 52 - "Источники индикаторов компрометации"

В рассматриваемом интерфейсе присутствуют следующие функциональные возможности:

- Включение встроенных источников идентификаторов компрометации;

  Для этого необходимо нажать на пиктограмму фильтра, перевести "Активность" в статус "Не важно".
  После, напротив нужных источников, нажать на кнопку-тумблер для перевода статуса в "Активно".
  Пример включения источника представлен на рисунке 53.

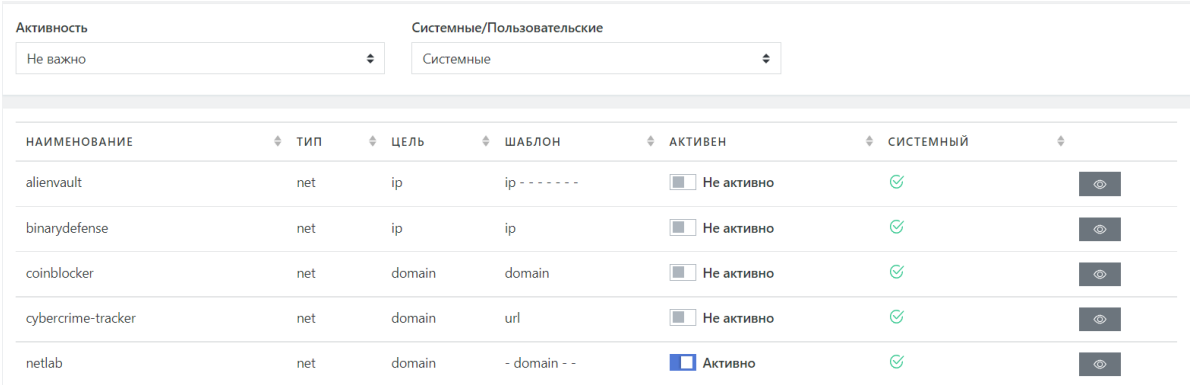

Рисунок 53 - "Включение источника индикаторов компрометации"

- Указание периода получения идентификаторов компрометации из активных источников;

  Для указания периода, нужно нажать на соответствующую кнопку и установить нужное количество часов (от 1 до 24)

- Ручной запуск и остановка сбора идентификаторов компрометации из активных источников;

  Для этого необходимо нажать на соответствующие кнопки в верхней части страницы, как изображено на рисунке 54.

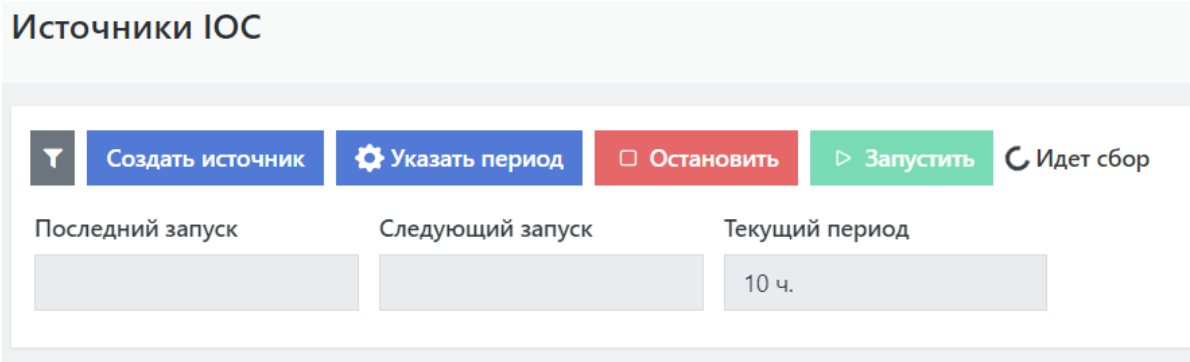

Рисунок 54 - "Включение\отключение сбора индикаторов компрометации"

- Создание пользовательского источника идентификаторов компрометации.

Для создания пользовательского источника нужно нажать на кнопку "Создать источник", заполнить поля и нажать на "Сохранить".

7. Табличные списки

7.1. Работа с табличными списками из UI

Система использует для обогащения событий безопасности, а также при работе правил корреляции, различные массивы информации. Для хранения статических данных используются хранилища значений, которые входят в пакет поставки, а также могут заполняться пользователем. Для хранения динамических данных используются табличные списки. Табличные списки могут быть пополнены как пользователем, так и правилами корреляции при работе. Правила корреляции могут как вносить новые записи в табличные списки, так и получать и удалять существующие. Также для записей в табличном списке может быть определен срок жизни записи (TTL), по истечение которого запись будет автоматически удалена.

Для управления табличными списками необходимо перейти в раздел “Коррелятор” → “Табличные списки”.

На рисунке 55 изображено окно управления табличными списками.

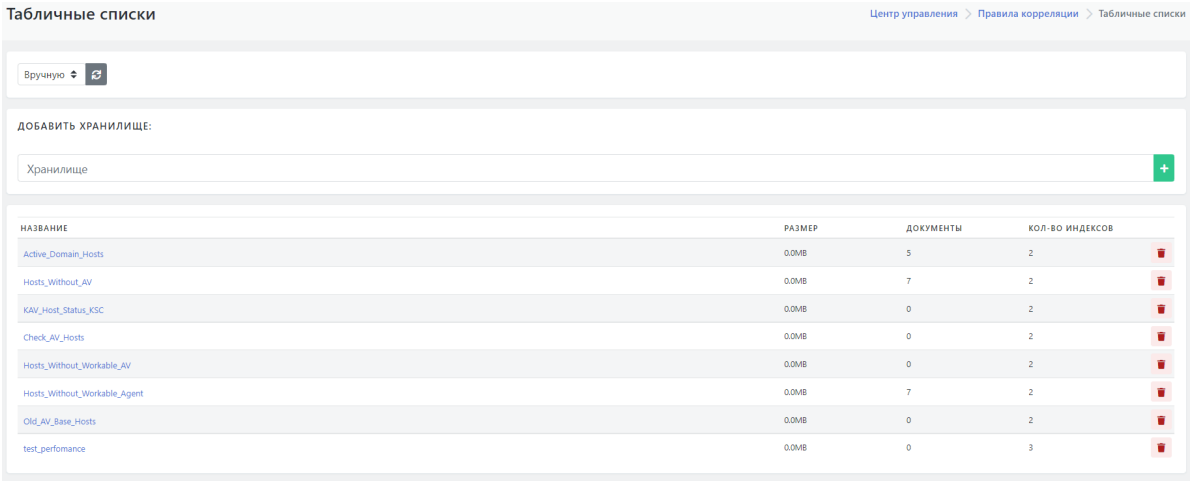

Рисунок 55 - "Табличные списки"

В рассматриваемом интерфейсе присутствуют следующие функциональные возможности:

- Создание табличных списков;

  Для этого необходимо ввести название создаваемого списка и нажать на “+”.
  В результате должен быть создан табличный список, как изображено на рисунке 56.

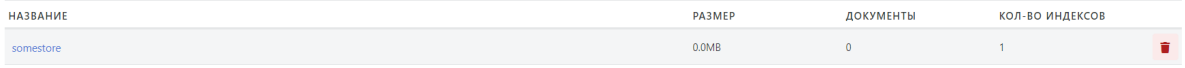

Рисунок 56 - "Результат создания табличного списка"

- Удаление табличного списка

  Для этого необходимо выбрать список, который необходимо удалить и нажать на пиктограмму урны. После чего список будет удалён.

Для того что бы перейти во внутрь списка, необходимо нажать на его название, после чего откроется его содержимое, представленное на рисунке 57.

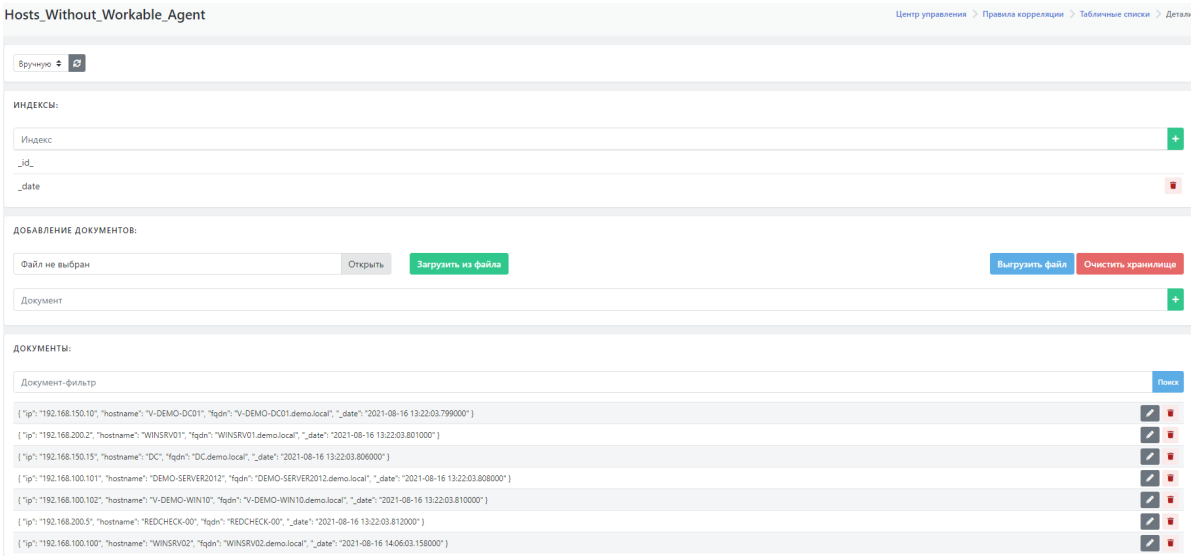

Рисунок 57 - "Содержимое списка Hosts_Without_Workable_Agent"

В интерфейсе управления конкретным табличным списком присутствуют следующие функциональные возможности:

- Создание индексов (сущность, позволяющая упорядочить документы табличного списка по определенному полю, что существенно ускоряет поиск);

  Для создания индекса, необходимо ввести название поля, по которому необходимо проиндексировать документы в табличном списке и нажать на “+”. Результат создания индекса изображен на рисунке 58.

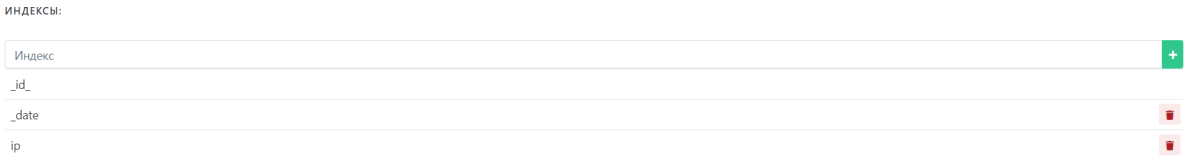

Рисунок 58 - "Результат создания индекса ip"

- Добавление документов из файла;

  Для этого необходимо нажать на кнопку “Открыть”, выбрать .json файл, содержащий документ и нажать на кнопку “Загрузить из файла”.

  Пример .json файла содержащего документ:

```
{
"documents": [
  {
    "ip": "192.168.110.2",
    "first_seen": 1628816008.011093,
    "first_seen_timestamp": "2021-08-13T00:53:28.0110938Z"
  }
 ]
}
```

  В результате приведенных выше действий, в табличный список будет добавлен новый документ.

- Выгрузка всех документов списка в файл;

Для этого необходимо нажать на кнопку “Выгрузить файл”, после чего автоматически будет скачан файл, содержащий информацию об актуальных документах табличных списков. Пример содержимого файла выгрузки документов из списка Hosts_Without_Workable_Agent (рассмотренного выше):


```
{
"documents": [
  {
    "ip": "192.168.150.10",
    "hostname": "V-DEMO-DC01",
    "fqdn": "V-DEMO-DC01.demo.local",
    "_id": "611914fb1f7eb2aed3fc74d1",
    "_date": "2021-08-16 13:22:03.799000"
  },
  {
    "ip": "192.168.200.2",
    "hostname": "WINSRV01",
    "fqdn": "WINSRV01.demo.local",
    "_id": "611914fb1f7eb2aed3fc74d2",
    "_date": "2021-08-16 13:22:03.801000"
  },
  {
    "ip": "192.168.150.15",
    "hostname": "DC",
    "fqdn": "DC.demo.local",
    "_id": "611914fb1f7eb2aed3fc74d3",
    "_date": "2021-08-16 13:22:03.806000"
  },
  {
    "ip": "192.168.100.101",
    "hostname": "DEMO-SERVER2012",
    "fqdn": "DEMO-SERVER2012.demo.local",
    "_id": "611914fb1f7eb2aed3fc74d4",
    "_date": "2021-08-16 13:22:03.808000"
  },
  {
    "ip": "192.168.100.102",
    "hostname": "V-DEMO-WIN10",
    "fqdn": "V-DEMO-WIN10.demo.local",
    "_id": "611914fb1f7eb2aed3fc74d5",
    "_date": "2021-08-16 13:22:03.810000"
  },
  {
    "ip": "192.168.200.5",
    "hostname": "REDCHECK-00",
    "fqdn": "REDCHECK-00",
    "_id": "611914fb1f7eb2aed3fc74d6",
    "_date": "2021-08-16 13:22:03.812000"
  },
  {
    "ip": "192.168.100.100",
    "hostname": "WINSRV02",
    "fqdn": "WINSRV02.demo.local",
    "_id": "61191f4b1f7eb2aed3fc74d7",
    "_date": "2021-08-16 14:06:03.158000"
```

```
    }
  ]
}
```

- Очистка табличных списков;

  Для очистки списков от всех документов необходимо нажать на кнопку “Очистить список”, в результате чего все документы из табличного списка будут удалены (при этом сам список и созданные индексы останутся).

- Редактирование документов;

  Для редактирования документов необходимо нажать на пиктограмму карандаша возле одного из документов, после чего откроется возможность текстового редактирования документа, как изображено на рисунке 59.

Рисунок 59 - "Изменение документа"

```
После внесения изменений необходимо нажать на галочку, после чего изменения в документе должны сохраниться.
```

- Удаление документов;

  Для удаления документа необходимо нажать на пиктограмму урны возле одного из документов, после чего данный документ будет удален из хранилища.

- Поиск по документам табличных списков;

  Выборку по документам в списке можно проводить по определенным фильтрам.

  1. Поиск по конкретному значению определенного поля;

  Пример представлен на рисунке 60

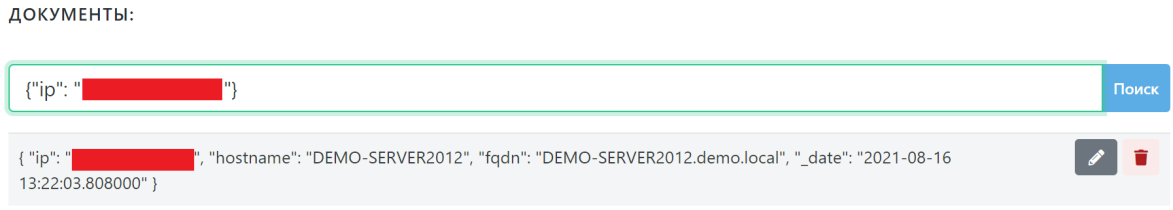

Рисунок 60 - "Фильтр по конкретному значению"

2. Поиск по регулярному выражению.

Пример представлен на рисунке 61

Рисунок 61 - "Фильтр по регулярному выражению"

8. Настройка контроля установленного ПО

8.1. Настройка контроля установленного ПО

8.1.1. Добавление правила контроля ПО

Для создания правила контроля ПО необходимо:

1. Перейти в раздел основного меню «**Оценка соответствия ПО**» и выбрать подраздел «Правила». Откроется текущий список правил (см. Рисунок 62).

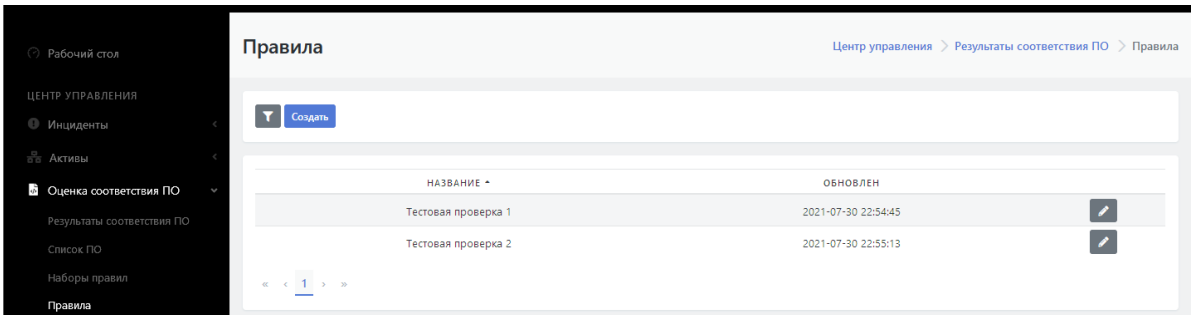

Рисунок 62 - Список правил.

2. Нажать на кнопку "Создать". На экране откроется форма для создания нового правила (см. Рисунок 62).
3. Заполнить форму согласно подсказкам.
4. Нажать на кнопку «Сохранить».

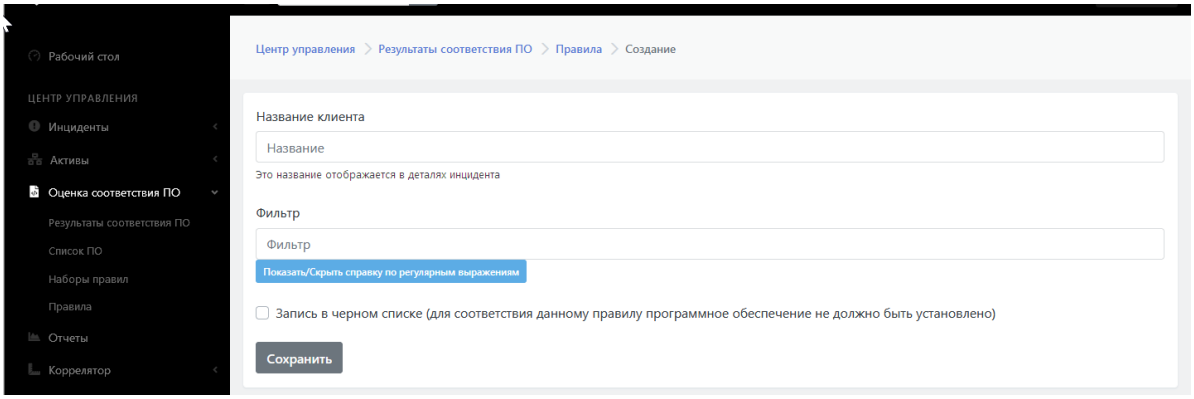

Рисунок 63 - Форма создания нового правила

В список правил добавится новое правило.

8.1.2. Редактирование правила контроля ПО. Удаление правила

Для редактирования правила контроля ПО необходимо:

1. Перейти в раздел основного меню «**Оценка соответствия ПО**» и выбрать подраздел «**Правила**». Откроется текущий список правил (см. Рисунок 62).
2. Выбрать нужную строку с правилом и нажать на пиктограмму  . Откроется форма редактирования правила.
3. Внести необходимые изменения.
4. Нажать на кнопку «Сохранить»

5. Нажать на кнопку «Удалить» если необходимо удаление правила.

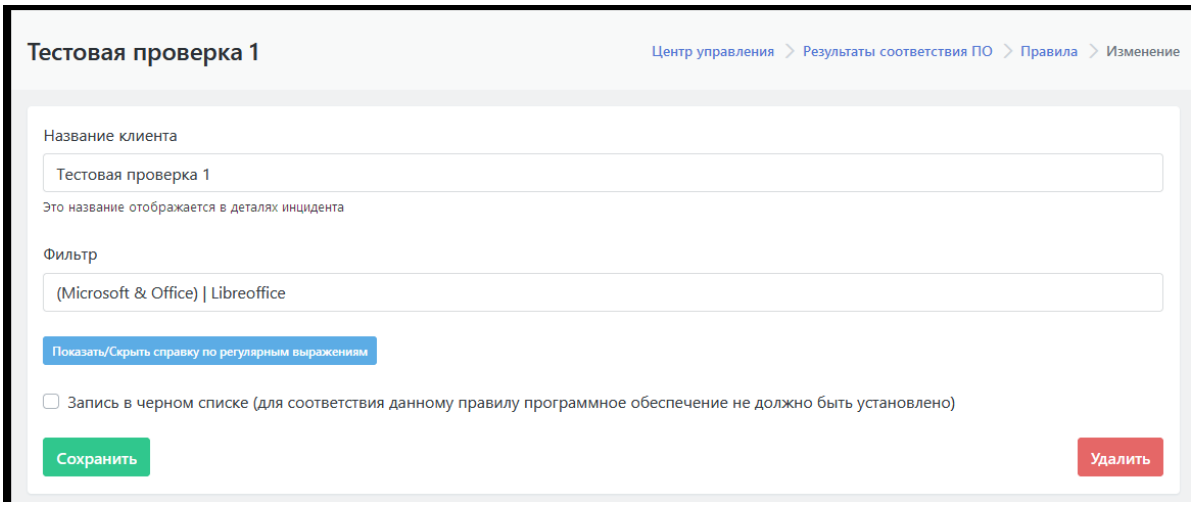

Рисунок 64

9. Параметры

9.1. Параметры

9.1.1. Общее описание подраздела "Параметры"

Основное меню Центра управления Платформы Радар содержит раздел "Параметры", включающий два подраздела (см. Рисунок 65):

- "**Параметры**";
- "**Черный список ID-плагинов**".

Подраздел "**Параметры**" предназначен для выполнения следующих функций:

- Вкладка "**Параметры**" - предназначена для обновления параметров уведомления.
- Вкладка "**Обработка уязвимостей**" - предназначена для настройки параметров автоматического переоткрытия инцидентов.
- Вкладка "**Синхронизация с Базой данных**"- предназначена для проведения синхронизации с Базой данных типов инцидентов и коррелятора.

9.1.2. Обновления параметров уведомления

Для обновления параметров уведомления необходимо зайти в раздел «Параметры», подраздел «Параметры». По умолчанию откроется вкладка «Общие», содержащая форму для ввода параметров уведомления (см. Рисунок 65).

Для обновления параметров уведомления необходимо:

1. Заполнить поле "**Название клиента**".
2. Заполнить поле "**Расположение**" (по умолчанию указана Москва).
3. Из выпадающего списка "**Группа пользователей по-умолчанию для инцидентов, связанных с активами, без определенного "ответственного пользователя"**" выбрать нужную группу пользователей.

4. Из выпадающего списка "**Стратегия идентификации активов по-умолчанию**" выбрать нужную стратегию:
    - IP;
    - FQDN;
    - MAC.
5. При необходимости включить совпадение по имени хоста (PQDN), где выбрана стратегия идентификации FQDN.
6. Нажать на кнопку "Сохранить".

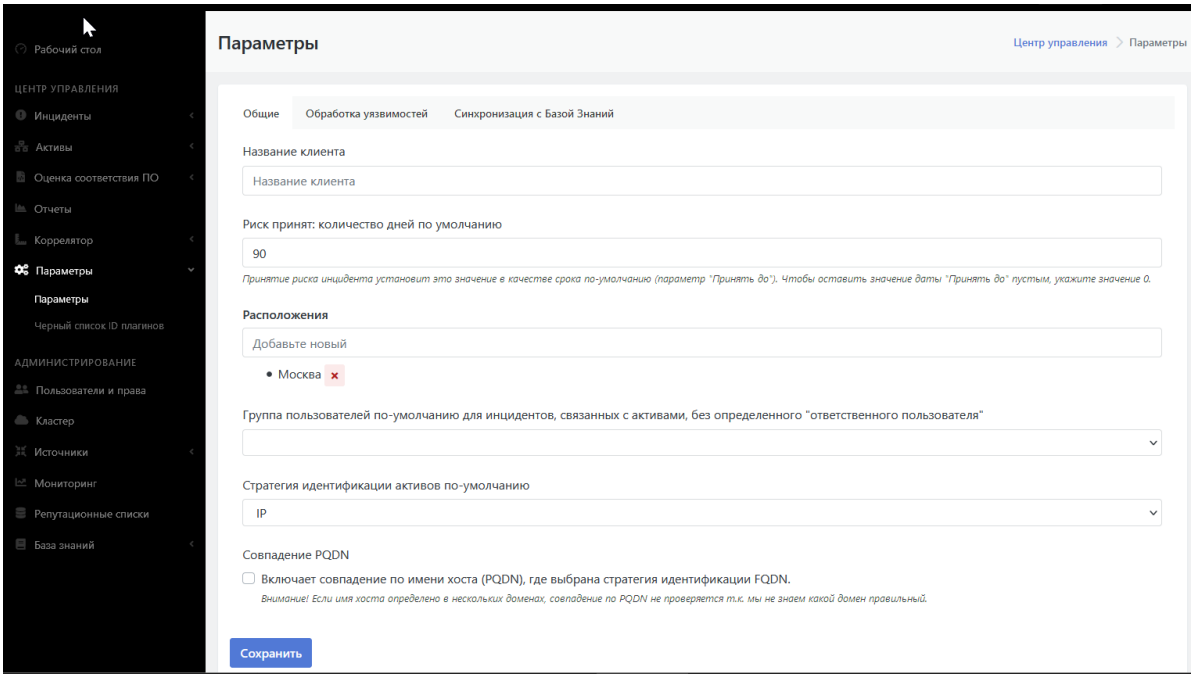

Рисунок 65 - Раздел "Параметры", вкладка "Общие"

9.1.3. Настройка автоматического переоткрытия инцидентов

Функционал по открытию инцидентов, находящихся в статусе «Закрыт», имеет ряд настроек. Для изменения настроек
переоткрытия инцидентов необходимо зайти в раздел «Параметры», подраздел «Параметры» и открыть вкладку "Обработка уязвимостей" (см. Рисунок 66).

Вкладка содержит следующие настройки переоткрытия инцидента:

- "**Минимальный уровень риска для повторного открытия инцидентов**" - риски ниже установленного в данном поле уровня не переоткрываются автоматически. Из раскрывающегося списка можно установить следующие уровни:
    - Высокий;
    - Средний;
    - Низкий;
    - Нет.
- "**Статус повторно открытых инцидентов**" -- переоткрываемые инциденты будут автоматически переводиться в указанный в данном поле статус. Из раскрывающегося списка можно установить следующие статусы:
    - Новый;
    - Назначена.
- Нажать кнопку «Сохранить».

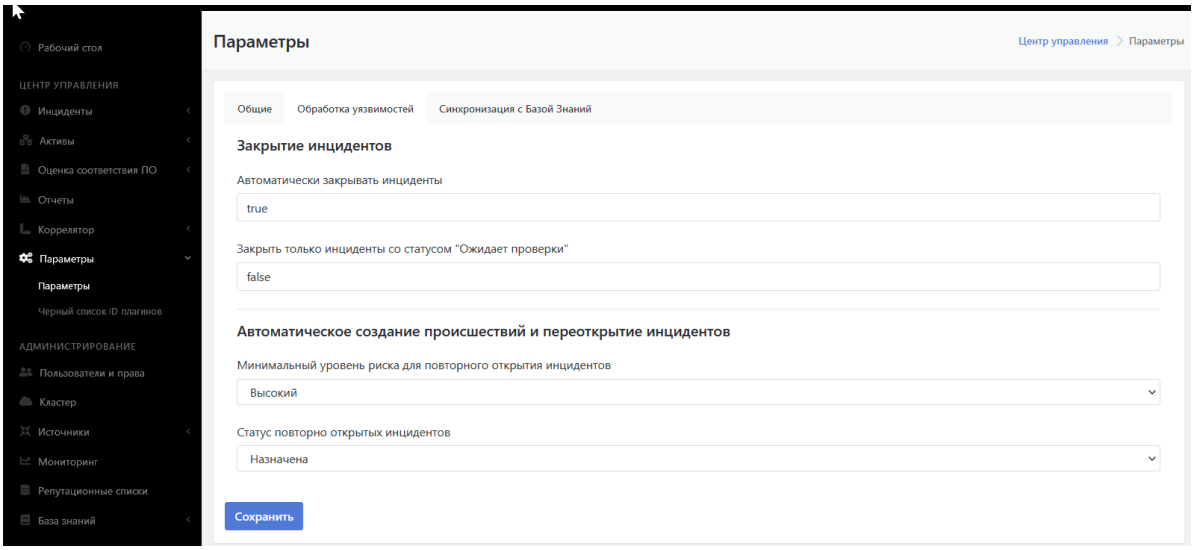

Рисунок 66 - Раздел "Параметры", вкладка "Обработка уязвимостей"

9.1.4. Синхронизация с базой знаний

Для синхронизации с Базой знаний необходимо зайти в раздел "Параметры", подраздел "Параметры" и открыть вкладку "Синхронизация с Базой знаний" (см. Рисунок 67).

Нажать последовательно кнопки «**Синхронизация типов инцидентов**» и «**Синхронизация коррелятора**».

Внимание! Данная процедура требуется только при установке и обновлении Платформы.

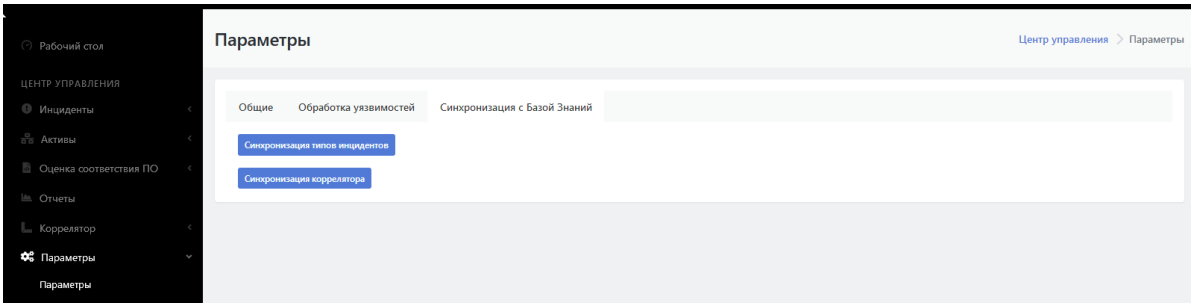

Рисунок 67 - Раздел "Параметры", вкладка "Синхронизация с Базой данных"

10. Пример настройки службы синхронизации времени в ОС Debian

Для настройки службы синхронизации времени необходимо выполнить следующие настройки:

!!! note "Все команды выполняются под привилегированным пользователем"

1. Добавить адрес NTP сервера в файл конфигурации службы:

```
echo 'NTP=<адрес NTP сервера>'>> /etc/systemd/timesyncd.conf
```

1. Перезапустить службу:

```
systemctl restart systemd-timesyncd.service
```

1. Проверка синхронизации:

```
timedatectl status
```

1. Проверка состояния службы:

```
systemctl status systemd-timesyncd.service
```

1. Добавление службы в атозапуск:

```
systemctl enable --now systemd-timesyncd.service
```

11. Аудит действий пользователей

Для настройки аудита действий пользователей необходимо выполнить следующие действия:

1. Перейти в настройки службы авторизации KeyCloak: https://<Ip-адрес-Платформы>:8180, как изображено на рисунке 68;

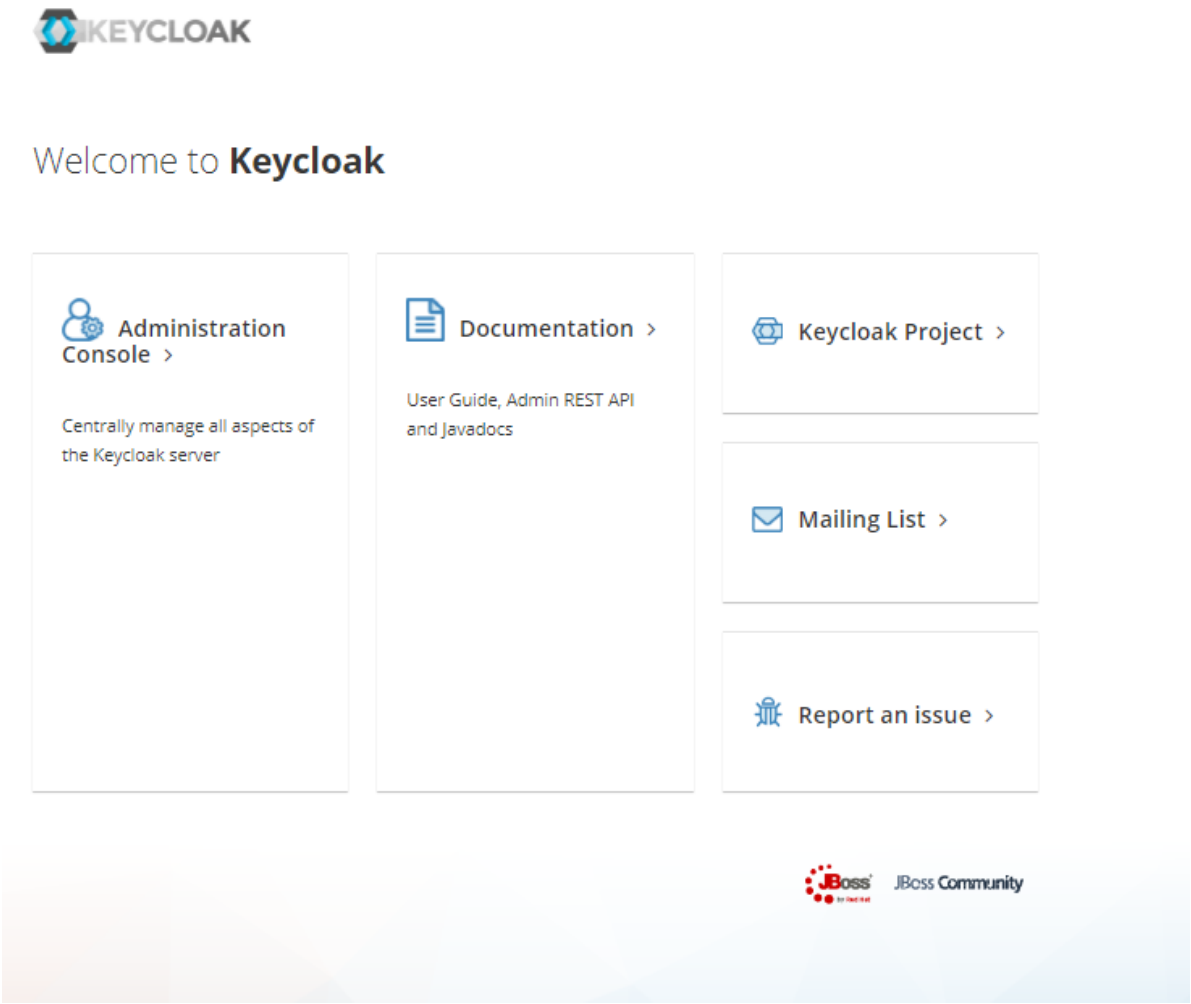

Рисунок 68 - "Страница управления службой KeyCloak"

2. Нажать на вкладку 'Administration Console' для того чтобы перейти в настройку службы авторизации;
3. Перейти в раздел "Управление", "События";
4. После чего, перейти во вкладку "Конфигурация", как изображено на рисунке 69;

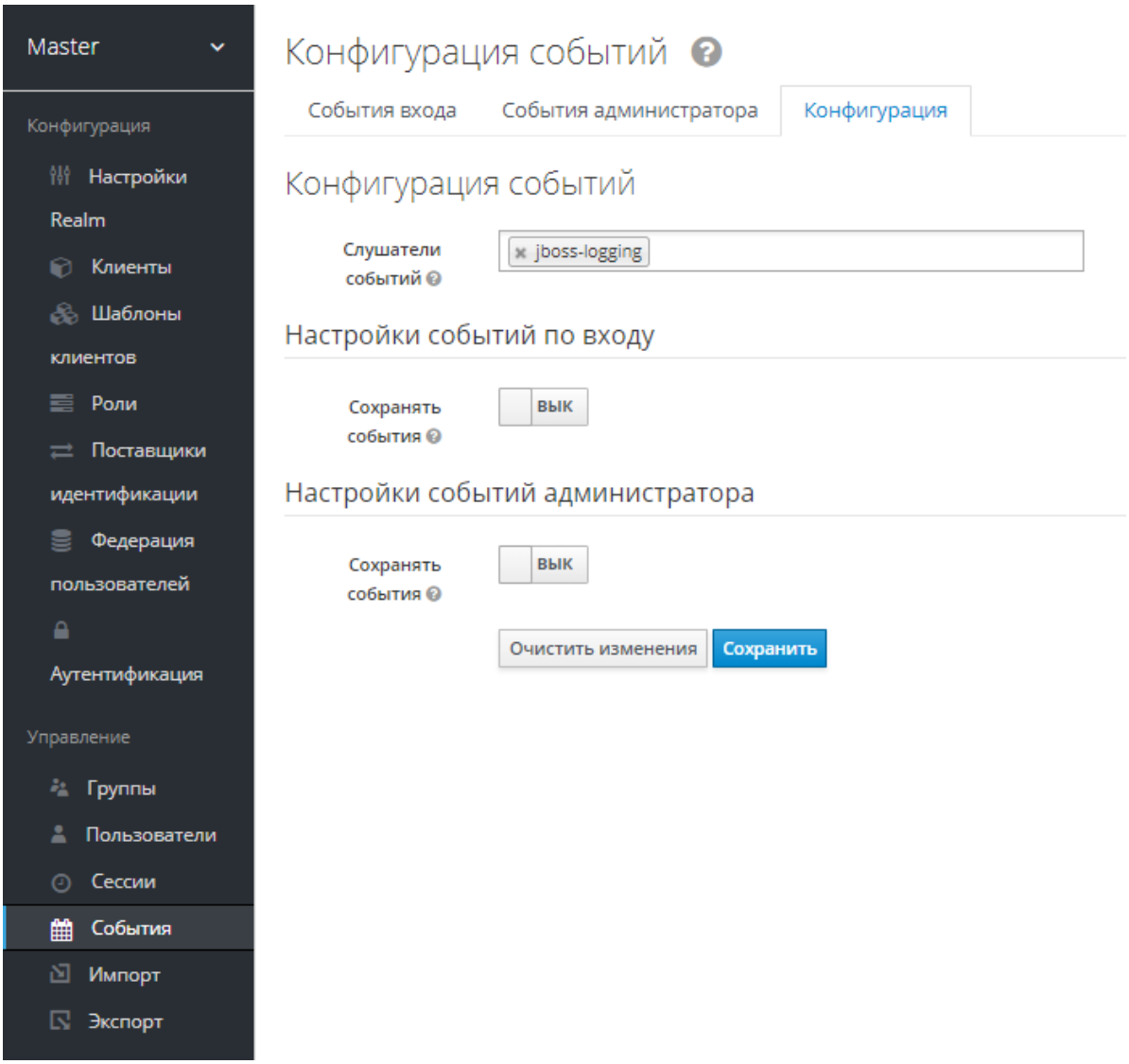

Рисунок 69 - "Страница управления службой KeyCloak"

5. Для включения логирования журналов аудита необходимо переключить кнопку-тумблер "Сохранять события", в каждом из блоков настроек событий, как изображено на рисунке 70;

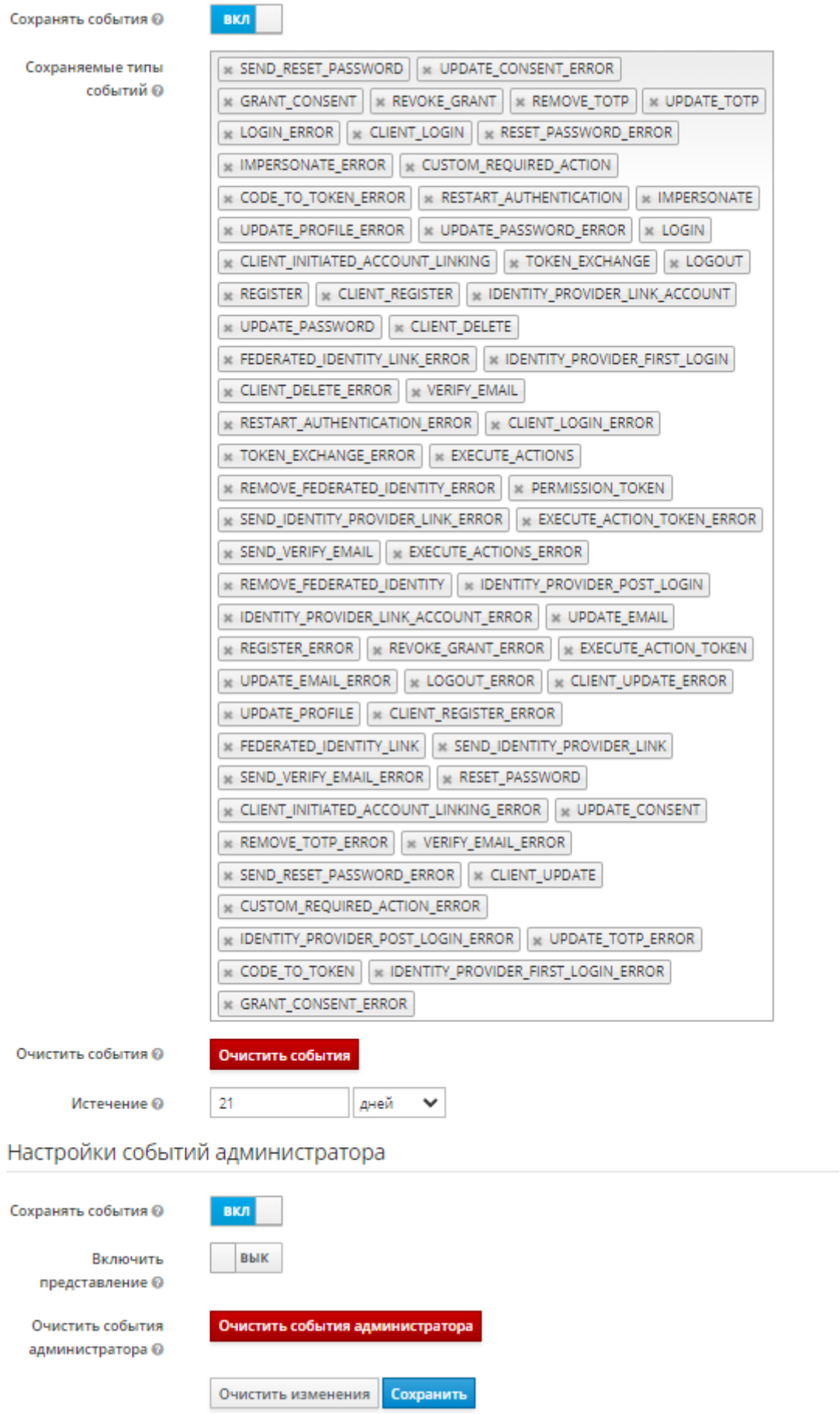

Рисунок 70 - "Включение логирования событий аудита"

6. После этого, необходимо настроить период хранения событий в поле "Истечение", указав количество минут\часов\дней;
7. Для завершения настроек логирования событий аудита нужно сохранить изменения, которые были внесены. Для этого необходимо нажать на кнопку "Сохранить".

После чего, настройку логирования аудита действий можно считать завершенной.

События аудита действий и события входа находятся в разделе "Пользователи и права" во вкладках "Аудит действий" и "События входа" соответственно.

На рисунке 71 изображена вкладка "Аудит действий"

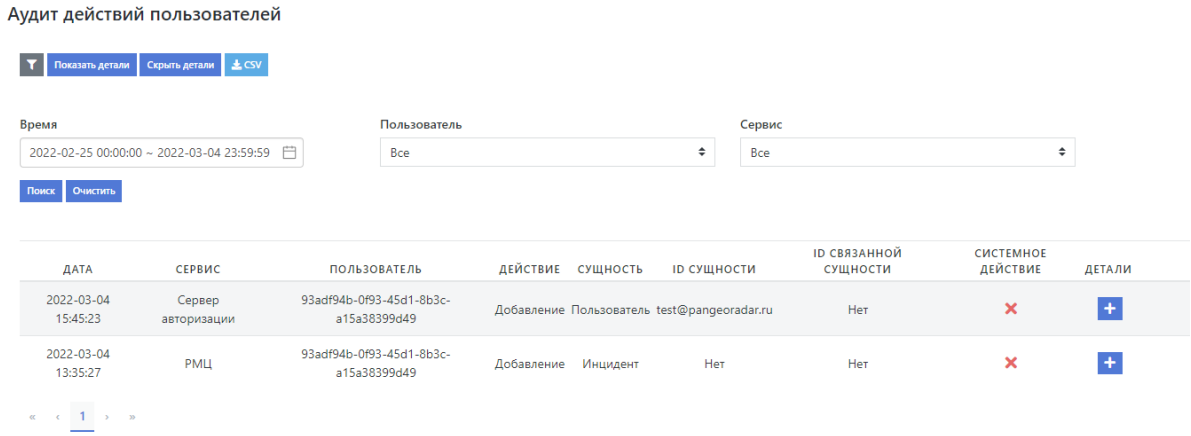

Рисунок 71 - "Аудит действий"

12. Интеграционный слой

Внутри поставки имеется гибкий инструмент для организации интеграции между Платформой Радар и любой другой системой с которой можно коммуниициировать через Json API.

12.1. Концепция интеграционного слоя

Основной логикой является наблюдение за изменением данных в структуре хранения информации Платформы PostgreSQL и запуск некоторых действий при наступлении того или иного изменения.

12.1.1. Наблюдение за изменениями

Основной для реагирования на изменение является секция `radar-tables-trigger`. По сути является перечнем наблюдателей реагирующие на изменения в базе данных.

```
radar-tables-trigger: &radar-tables-trigger
  - name: create_incident
    table: service_asset_findings
    fields: [ ]
    kind: insert
    sql: "SELECT ..."
    sql_vars:
      id: float64toInt64
    outputs:
      - *rvision_insert
```

Доступные триггеры, поле `kind`:

- insert
- update
- delete

При срабатывании тригера на выбранной таблице указанной в секции `table` запускается `sql` скрипт собирающий основной объект для передачи запуска сборки объекта отправляемого в интегрируемую системы. Если `sql` скрипт не указан будет взята строка из наблюдаемой таблицы.

Также есть поле `fields` заполнив которое можно указать те поля при изменении которых должен запускать триггер.

поле `sql_vars` - необходимо для приведения типов:

- bool
- int
- int32
- int64
- string
- float32
- float64
- float64toInt64

поле `outputs` указывает необходимые каналы для дальнейшей интеграции с внешними системами.

12.1.2. Отправка изменений

Для организации канала отправки объекта соответствия сформированного после срабатывания триггера, необходимо создать секцию в которую направить вывод в поле `outputs`

Секций имеет следующую структуру:

```
rvision_update: &rvision_update
  type: http
  url: "https://IP/api/v2/incidents"
  method: POST
  content_type: "application/json"
  headers:
    PgrApiKey: "test"
  prepend:
    - type: http
      url: "https://IP/api/v2/incidents"
      method: GET
      content_type: "application/json"
      query:
        token: "SECRET"
        fields: "identifier"
        filter: '[{"operator":"=","value":%d,"property":"id_siem"}]'
      query_vars_from_trigger:
        filter:
          field: "id"
          type: "float64toInt64"
      append_to_mapping:
        identifier: "data.result.0.identifier"
  mappging: *rvision_mapping
```

`type` - транспорт, на данны момент поддерживается только `http`
`url` - адрес ендпоинта для отправки запроса
`method` - тип запроса (GET, POST, PUT)
`content_type` - значение заголовка content-type
`headers` - дополнительные заголовки в формате `ключ: значение`
`mapping` - объект соответствия
`prepend` - дополнительная секция, которая будет выполненна перед отправкой основного запроса, полезна если нужно сделать запрос на получение доп. информации и обогащения объекта соответствия перед отправкой

В секции `prepend` доступны теже основные поля, что и в основной. В дополнении к ней доступны опции:
`query` - строка запроса
`query_vars_from_trigger` - шаблонизация для строки запроса
`append_to_mapping` - обогащение объекта соответветствия по ключу и значению из ответа

12.1.3. Объект соответствия

Объект соответвия - это объект который будет собран после срабатывания триггера и передан в канал на отправку.

Структура объекта:

```yaml
rvision_mapping: &rvision_mapping
  token:
    type: "manual"
    value: "SECRET_KEY"
  status_siem:
    type: "map"
    value: "status"
  STATUS:
    type: "active_map"
    value: "status"
    map:
      new: "Создан"
      risk_accepted: "Зарегистрирован"
      assigned_customer: "Назначен"
      working_customer: "Обработка"
      feedback_required: "Раследование"
      closed: "Закрыт"
```

`type` - тип соответствия, доступны:

- `manual` - заданное ручное значение в ключе `value`
- `map` - ключу объекта будет соответствовать полю указанного в значение ключа `value` из триггера изменений
- `active_map` - ключу объекта, будет соответствовать значение из объекта `map` найденного при совпадении ключа и значению поля из триггера изменений, указанного в ключе `value`

12.2. Пример интеграции с SOAR RVision

Ниже представлен конфигурационный файл `/opt/pangeoradar/configs/pangeoradar-pgr-wal-listener.yaml` настроенный на обмен информацией об инцидентах с SOAR RVision.

```yaml
---

global:
  force_replica_identity: true
  log_level: warning

# Mappings
rvision_mapping: &rvision_mapping
  token:
    type: "manual"
    value: "SECRET_KEY"
  category:
    type: "manual"
    value: "Инцидент из Пагео Радар"
  info_source:
    type: "manual"
    value: "SIEM Пангео"
  type:
    type: "manual"
    value: "Инцидент полученный из Пангео Радар"
  id_siem:
    type: "map"
    value: "id"
  DESCRIPTION:
    type: "map"
    value: "DESCRIPTION"
  risk_impact:
    type: "map"
    value: "risk_impact"
  solution:
    type: "map"
    value: "solution"
  mitigation:
    type: "map"
    value: "mitigation"
  status_siem:
    type: "map"
    value: "status"
  STATUS:
    type: "active_map"
    value: "status"
    map:
      new: "Создан"
      risk_accepted: "Зарегистрирован"
      assigned_customer: "Назначен"
      working_customer: "Обработка"
      feedback_required: "Раследование"
      closed: "Закрыт"
  risklevel:
    type: "map"
    value: "risklevel"
  service_asset_id:
    type: "map"
    value: "service_asset_id"
  DETECTION_DATE:
```

```
    type: "map"
    value: "created_at"
  UPDATE:
    type: "map"
    value: "updated_at"
  finding_id:
    type: "map"
    value: "finding_id"
  analysis_output:
    type: "map"
    value: "analysis_output"
  synopsis:
    type: "map"
    value: "synopsis"
  title:
    type: "map"
    value: "title"
  risk:
    type: "map"
    value: "risk"
  OCCUR_DATE:
    type: "map"
    value: "acknowledged_at"
  alert_type:
    type: "map"
    value: "alert_type"
  client_note:
    type: "map"
    value: "client_note"
  internal_note:
    type: "map"
    value: "internal_note"
  external:
    type: "map"
    value: "external"
  immediate_action_score:
    type: "map"
    value: "immediate_action_score"
  throughput_period:
    type: "map"
    value: "throughput_period"
  throughput_period_change:
    type: "map"
    value: "throughput_period_change"
  customer_created:
    type: "map"
    value: "customer_created"
  c_visible_since:
    type: "map"
    value: "c_visible_since"
  c_visible_since_in_days:
    type: "map"
    value: "c_visible_since_in_days"
  c_reopened_count:
    type: "map"
```

```
    value: "c_reopened_count"
  c_last_customer_status_change:
    type: "map"
    value: "c_last_customer_status_change"
  c_customer_retention_time:
    type: "map"
    value: "c_customer_retention_time"
  logmule_identifier:
    type: "map"
    value: "logmule_identifier"
  c_remote_exploitable:
    type: "map"
    value: "c_remote_exploitable"
  c_occurrence_count:
    type: "map"
    value: "c_occurrence_count"
  last_occurrence_id:
    type: "map"
    value: "last_occurrence_id"
  itsm_last_synced_at:
    type: "map"
    value: "itsm_last_synced_at"
  itsm_sync_status:
    type: "map"
    value: "itsm_sync_status"
  external_id:
    type: "map"
    value: "external_id"
  itsm_sync_error:
    type: "map"
    value: "itsm_sync_error"
  user_id:
    type: "map"
    value: "user_id"
  updated_by:
    type: "map"
    value: "updated_by"
  group_id:
    type: "map"
    value: "group_id"
  acknowledged_by:
    type: "map"
    value: "acknowledged_by"
  created_by_customer:
    type: "map"
    value: "created_by_customer"
  edited_by:
    type: "map"
    value: "edited_by"
  active_name:
    type: "map"
    value: "active_name"
  IP:
    type: "map_from_json"
    value: "ip"
```

```
    from: "inet"
  fqdn:
    type: "map_form_json"
    value: "fqdn"
    from: "varchar_array"


# Outputs
rvision_insert: &rvision_insert
  type: http
  url: "https://IP/api/v2/incidents"
  method: POST
  content_type: "application/json"
  headers:
    PgrApiKey: "test"
  mappging: *rvision_mapping

rvision_update: &rvision_update
  type: http
  url: "https://IP/api/v2/incidents"
  method: POST
  content_type: "application/json"
  headers:
    PgrApiKey: "test"
  prepend:
    - type: http
      url: "https://IP/api/v2/incidents"
      method: GET
      content_type: "application/json"
      query:
        token: "SECRET"
        fields: "identifier"
        filter: '[{"operator":"=","value":%d,"property":"id_siem"}]'
      query_vars_from_trigger:
        filter:
          field: "id"
          type: "float64toInt64"
      append_to_mapping:
        identifier: "data.result.0.identifier"
  mappging: *rvision_mapping


radar-tables-trigger: &radar-tables-trigger
  - name: create_incident
    table: service_asset_findings
    fields: [ ]
    kind: insert
```

```
    sql: "SELECT t.id, CASE WHEN t.description ISNULL THEN f.description ELSE
t.description END, CASE WHEN t.risk_impact ISNULL THEN f.risk_impact ELSE
t.risk_impact END, CASE WHEN t.solution ISNULL THEN f.solution ELSE t.solution
END, t.mitigation, CASE WHEN t.solution ISNULL THEN f.solution ELSE t.solution
END, t.status, t.risklevel, t.service_asset_id, t.created_at, t.updated_at,
t.finding_id, t.analysis_output, t.synopsis, CASE WHEN t.synopsis ISNULL THEN
f.synopsis ELSE t.synopsis END, t.title, CASE WHEN t.title ISNULL THEN f.title
ELSE t.title END, t.risk, t.acknowledged_at, t.alert_type, t.client_note,
t.internal_note, t.external, t.immediate_action_score, t.throughput_period,
t.throughput_period_change, t.customer_created, t.c_visible_since,
t.c_visible_since_in_days, t.c_reopened_count, t.c_last_customer_status_change,
t.c_customer_retention_time, t.logmule_identifier, t.c_remote_exploitable,
t.c_occurrence_count, t.last_occurrence_id, t.itsm_last_synced_at,
t.itsm_sync_status, t.external_id, t.itsm_sync_error, t.user_id, t.updated_by,
t.group_id, t.acknowledged_by, t.created_by_customer, t.edited_by, s.name as
active_name, ni.ip as ip, ni.fqdn as fqdn FROM public.service_asset_findings t
left join findings f on t.finding_id = f.id left join service_assets s on
t.service_asset_id = s.id left join network_interfaces ni on s.id =
ni.service_asset_id where t.id = $1"
    sql_vars:
      id: float64toInt64
    outputs:
      - *rvision_insert

  - name: update_incident
    table: service_asset_findings
    fields: [ ]
    kind: update
    sql: "SELECT t.id, CASE WHEN t.description ISNULL THEN f.description ELSE
t.description END, CASE WHEN t.risk_impact ISNULL THEN f.risk_impact ELSE
t.risk_impact END, CASE WHEN t.solution ISNULL THEN f.solution ELSE t.solution
END, t.mitigation, CASE WHEN t.solution ISNULL THEN f.solution ELSE t.solution
END, t.status, t.risklevel, t.service_asset_id, t.created_at, t.updated_at,
t.finding_id, t.analysis_output, t.synopsis, CASE WHEN t.synopsis ISNULL THEN
f.synopsis ELSE t.synopsis END, t.title, CASE WHEN t.title ISNULL THEN f.title
ELSE t.title END, t.risk, t.acknowledged_at, t.alert_type, t.client_note,
t.internal_note, t.external, t.immediate_action_score, t.throughput_period,
t.throughput_period_change, t.customer_created, t.c_visible_since,
t.c_visible_since_in_days, t.c_reopened_count, t.c_last_customer_status_change,
t.c_customer_retention_time, t.logmule_identifier, t.c_remote_exploitable,
t.c_occurrence_count, t.last_occurrence_id, t.itsm_last_synced_at,
t.itsm_sync_status, t.external_id, t.itsm_sync_error, t.user_id, t.updated_by,
t.group_id, t.acknowledged_by, t.created_by_customer, t.edited_by, s.name as
active_name, ni.ip as ip, ni.fqdn as fqdn FROM public.service_asset_findings t
left join findings f on t.finding_id = f.id left join service_assets s on
t.service_asset_id = s.id left join network_interfaces ni on s.id =
ni.service_asset_id where t.id = $1"
    sql_vars:
      id: float64toInt64
    outputs:
      - *rvision_update


connections:
  - database: radar
```

```
    username: user
    password: secret
    host: 127.0.0.1
    port: 5432
    triggers: *radar-tables-trigger
```

13. Подготовка дисковой подсистемы для реализации роли DATA

Действия, описанные в данной инструкции, выполняются на хосте, который планируется ввести в состав кластера elasticsearch с ролью data (см. Раздел "Подготовка дисковой системы" (Install.md)).

При развертывании операционной системы необходимо выполнить следующие действия по разметке дисковой подсистемы:

1. Разметить основной диск, на который будет устанавливаться операционная система и дополнительное программное обеспечение для функционирования компонентов Платформы. Размеры разделов для основного диска выбираются исходя из объема жесткого диска. Основная рекомендация по разметке диска для ОС - размер раздела /var должен быть не менее 75 Gb, т.к. в нем будут размещать все необходимые для работы модуля данные .
2. Разметить дополнительные физические диски.

На Рисунке 1Б показан вариант разметки основного диска и три не размеченных физических диска.

Чтобы провести разметку дисковой системы необходимо выполнить следующие действия:

1. Выбрать диск и нажать кнопку «Continue» (см. Рисунок 72).

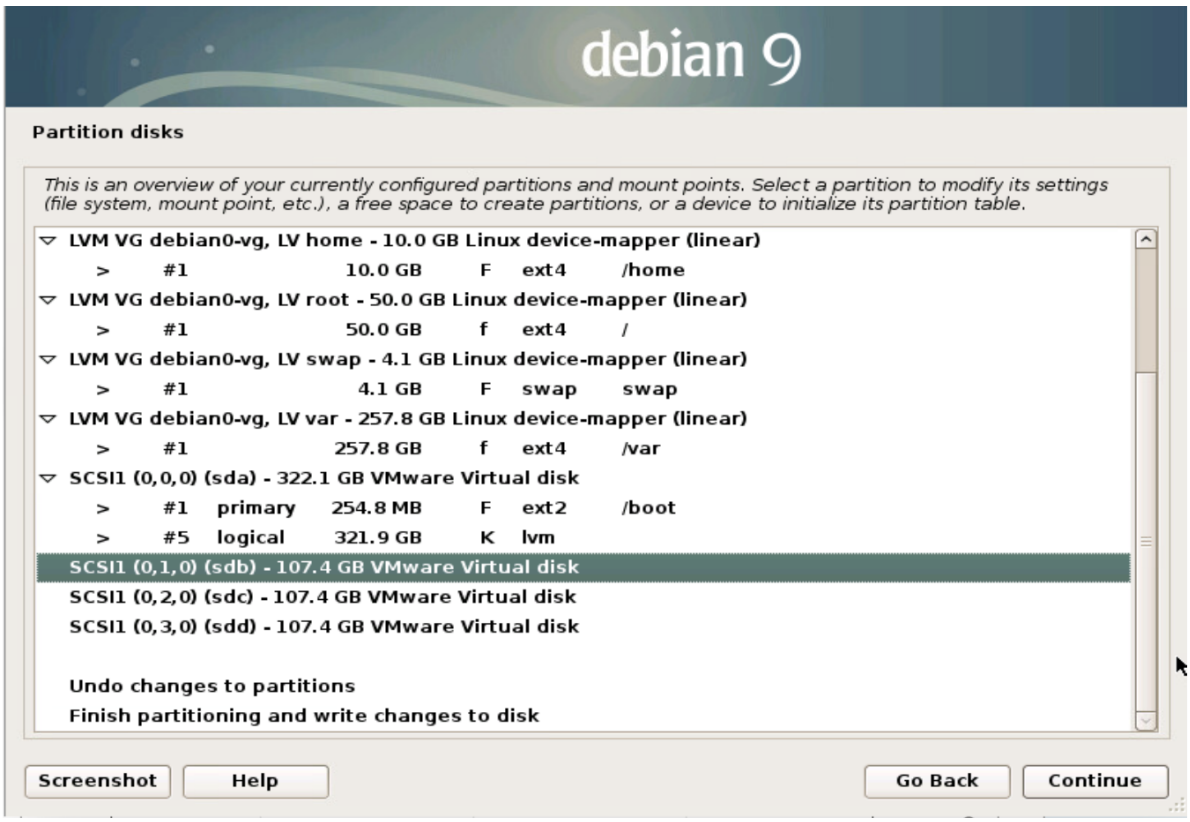

Рисунок 72 - Вариант разметки основного диска

1. Выбирать пункт «Yes» и нажать кнопку «Continue» (см. Рисунок 73).

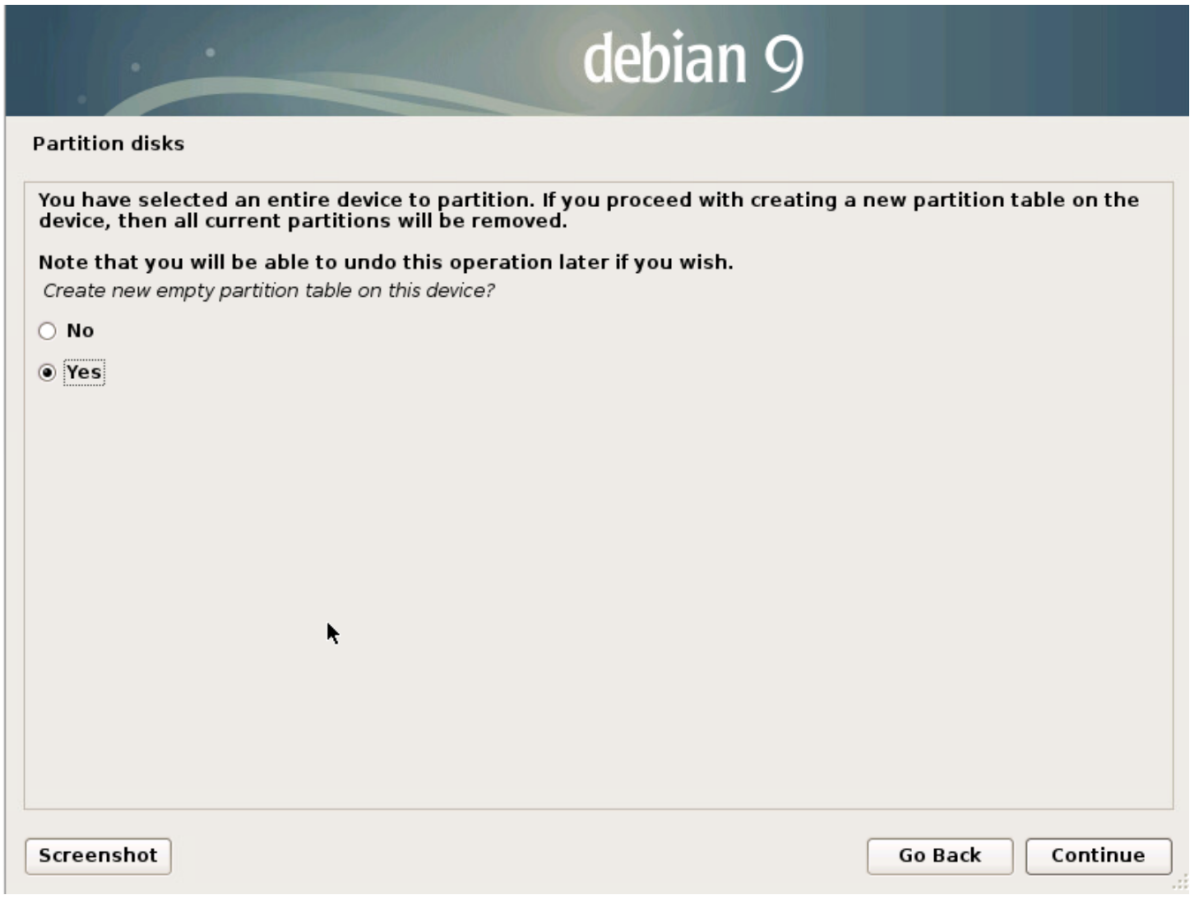

Рисунок 73 - Создание таблицы разделов

1. Выбирать вновь созданный пустой раздел на дополнительном диске (см. Рисунок 74).

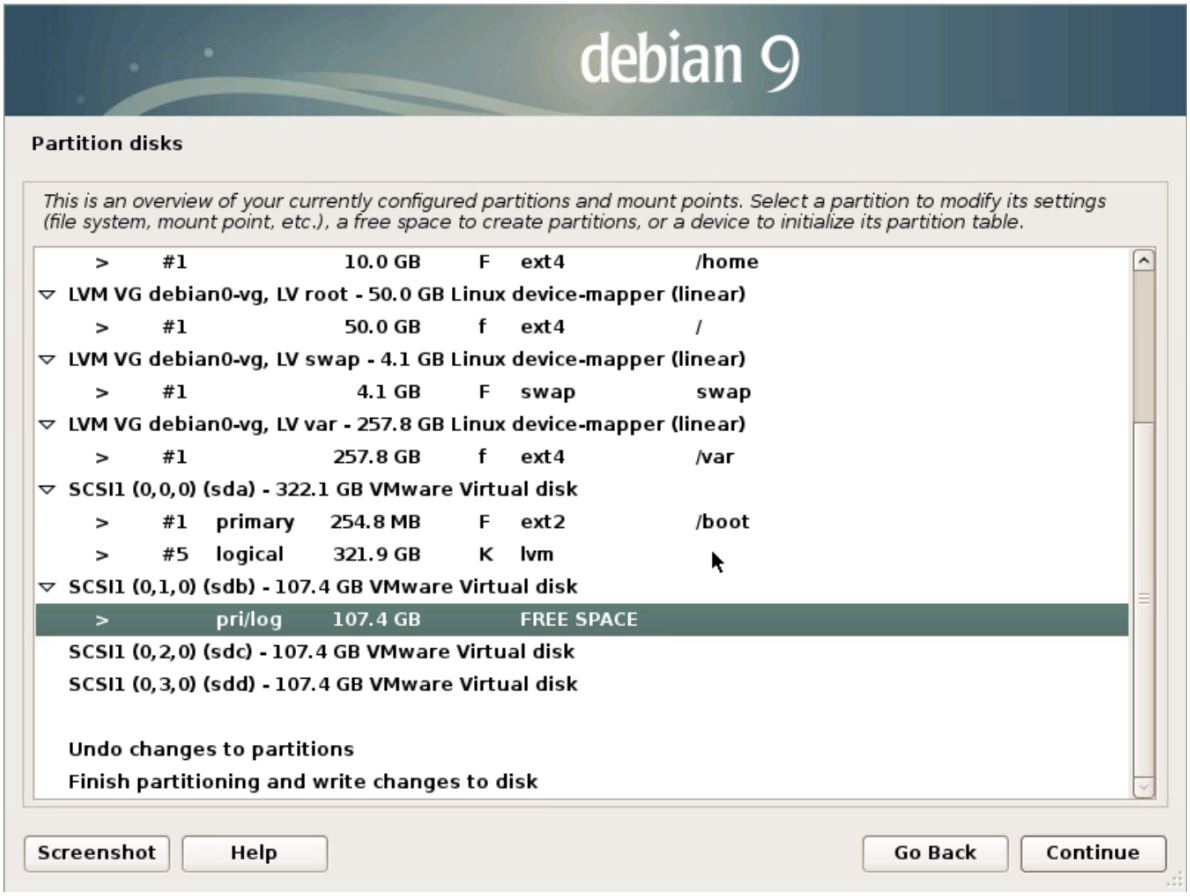

Рисунок 74 - Пустой раздел

1. Выбрать пункт «Create a new partition» и нажать кнопку «Continue» (см. Рисунок 75).

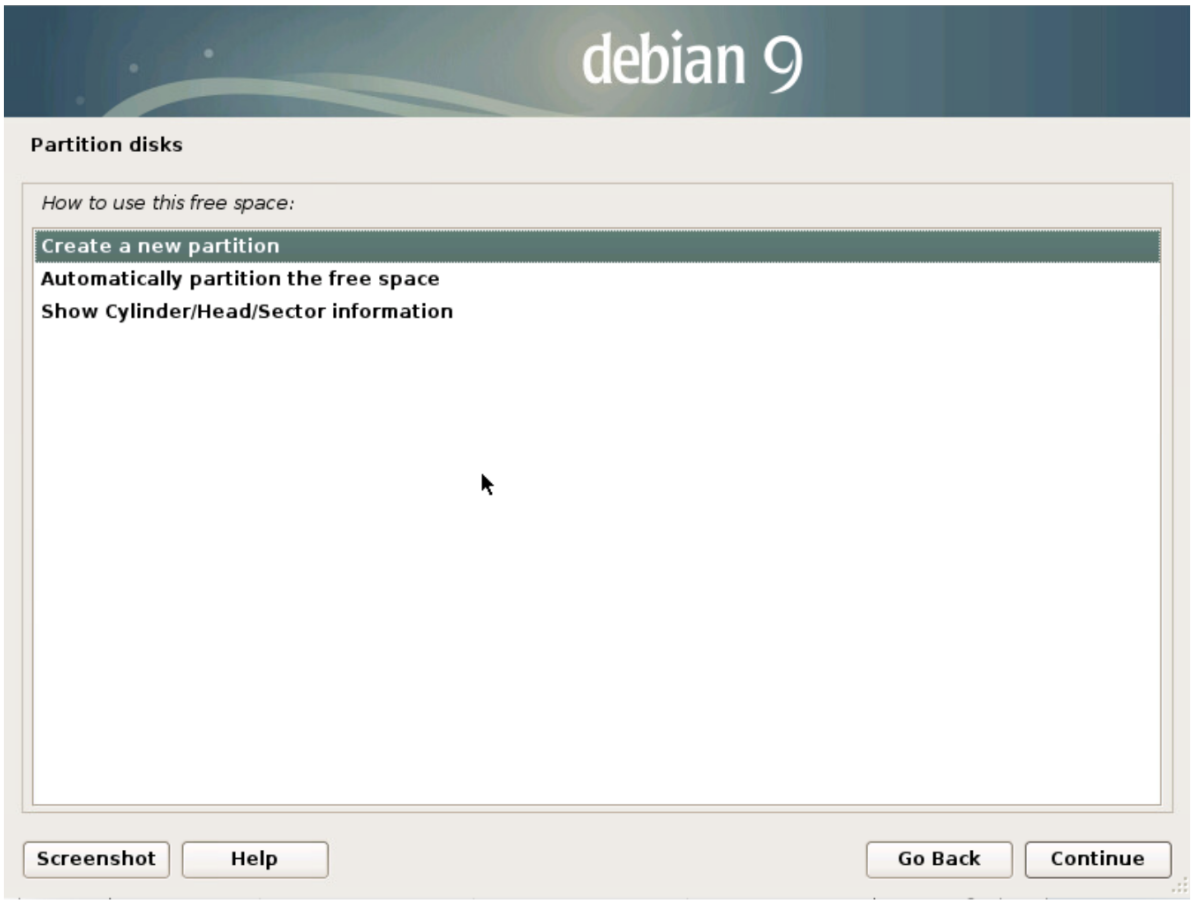

Рисунок 75 - Выбор действий

1. Указать объем раздела и нажать кнопку «Continue» (см. Рисунок 76).

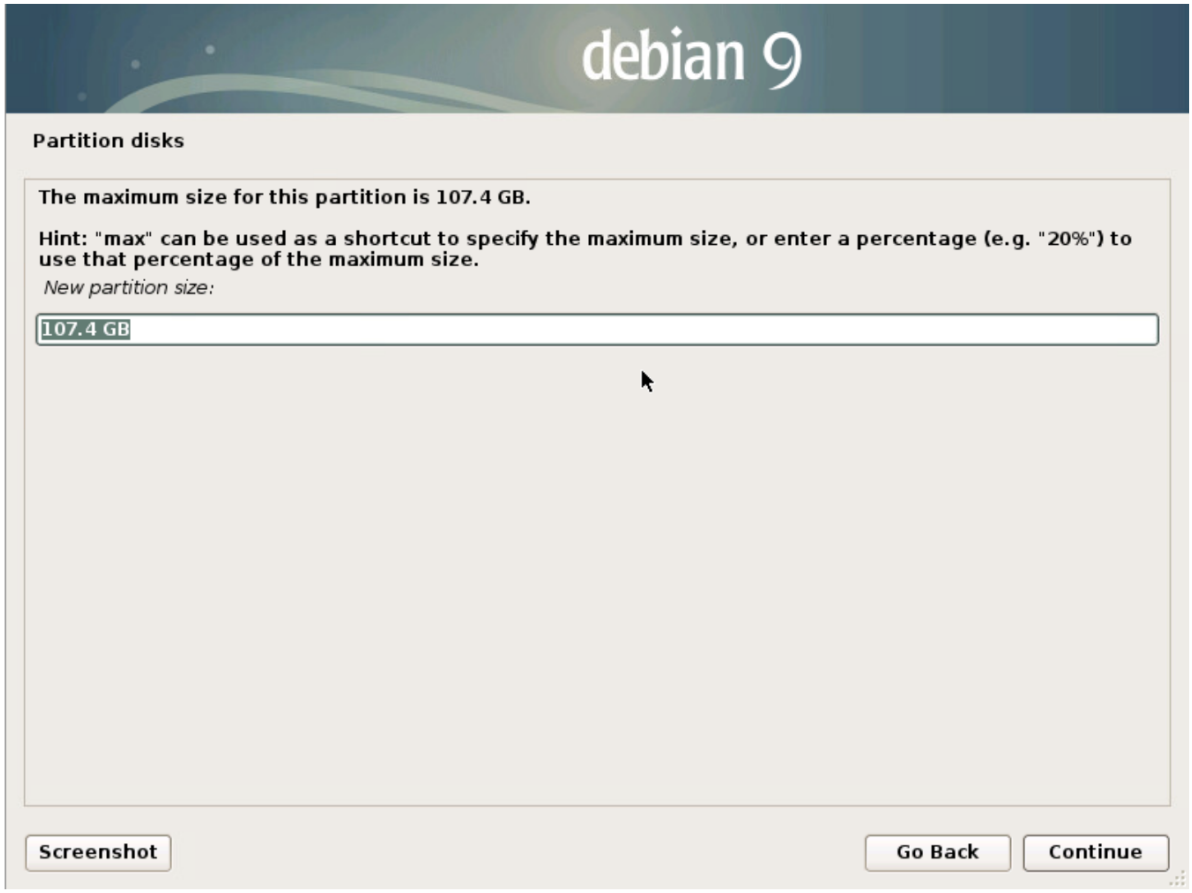

Рисунок 76 - Определение объема раздела

1. Выбрать тип раздела «Primary» и нажать кнопку «Continue» (см. Рисунок 77).

Рисунок 77 - Определение типа раздела

1. Далее необходимо настроить параметры раздела (см. Рисунок 78). Основные параметры:

- «Use as» - установить «Ext4 journaling file system»;
- «Mount point» - ввести «/dataN», где N - порядковый номер дополнительного физического диска, начиная с 1.
- Выбрать пункт «Done setting up the partition» и нажать кнопку «Continue».

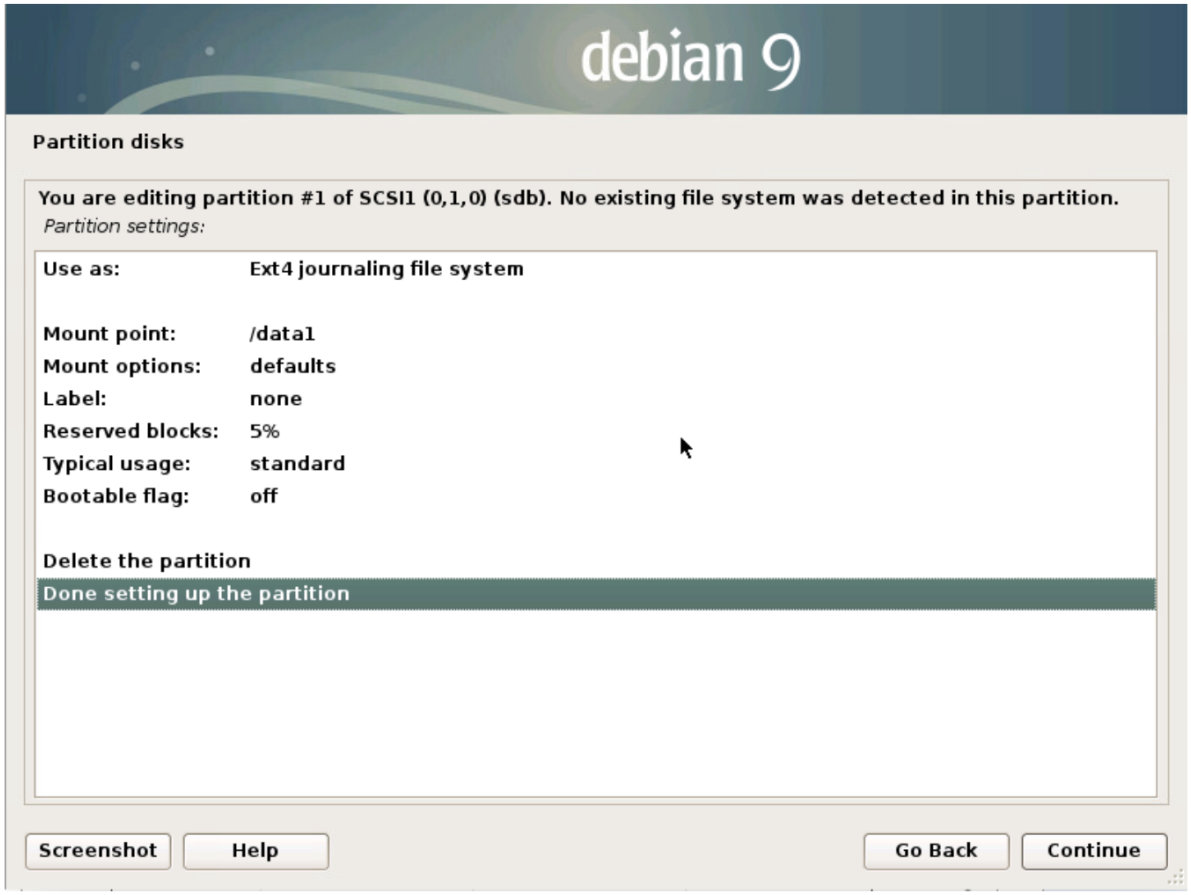

Рисунок 78 - Определение параметров раздела

1. Если дополнительных дисков более одного, то необходимо повторить все выше перечисленные действия для всех дополнительных физических дисков (см. Рисунок 79).

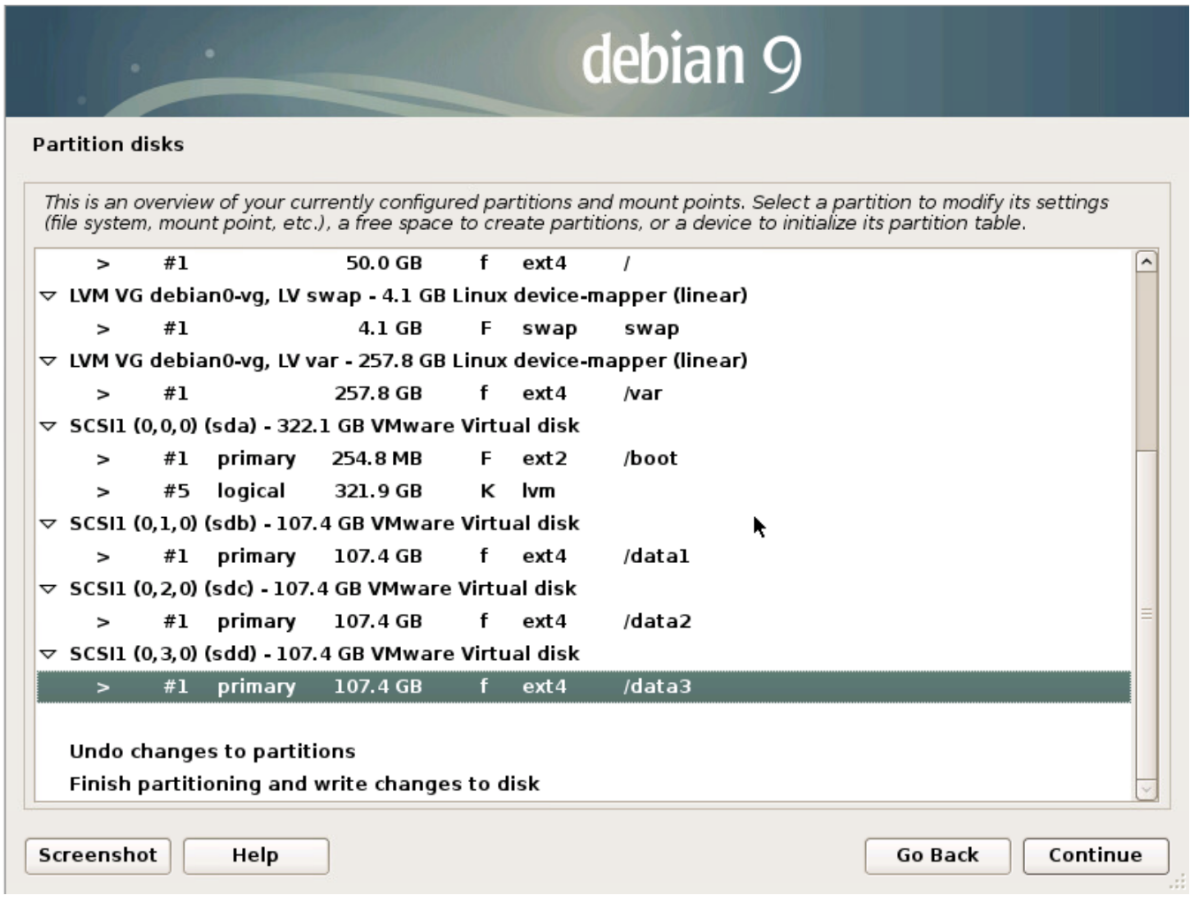

Рисунок 79 - Финальный вариант разметки

14. Сетевое взаимодействие

14.1. Централизованная установка Платформы

Ниже приведены необходимые сетевые настройки при централизованной установке Платформы Радар:

Исходящий	Входящий	Порты	Описание
Log-Collector	Master	1500-5000, 9997-9999, 15403, 15404TCP/UDP	Передача данных от сборщика в балансировщик и менеджер очередей
Log-Collector	Источники событий	21, 22, 135, 445, 1443, 3306, 5432, 5601	Активный сбор событий
Log-Collector	Log-Collector	4813	Взаимодействие между двумя сборщиками
Log-Collector	Master	9000	Запрос конфигурационных данных

Исходящий	Входящий	Порты	Описание
Master	Log-Collector	4805 4806	Мониторинг и сбор статистики. Управление сборщиком событий API
Источники событий	Log-Collector	162 SNMP trap 4807 UDP receiver 4808 TCP receiver 4809 TCP receiver SSL/TLS 4810 HTTP receiver 4811 HTTPS receiver 4812 NetFlow reciever	Пассивный сбор событий
Пользователи Платформы	Master	8080 9000 6676 6677	Доступ к интерфейсу платформы, проверка API ключей

14.2. Распределенная установка Платформы

Ниже приведены необходимые сетевые настройки при распределенной установке Платформы Радар (независимо от вариантов распределенной установки):

Исходящий	Входящий	Порты	Описание
Correlator	Master	5672	Чтение нормализованных событий из очереди для корреляции
Correlator	Master	8086	Передача результатов корреляции
Log-Collector	Balancer	1500-5000, 9997-9999, 15403, 15404TCP/UDP	Передача данных от сборщика в балансировщик и менеджер очередей
Log-Collector	Источники событий	21, 22, 135, 445, 1443, 3306, 5432, 5601	Активный сбор событий
Log-Collector	Log-Collector	4813	Взаимодействие между двумя сборщиками
Log-Collector	Master	9000	Запрос конфигурационных данных
Master	Data	9200	Работа с сырыми событиями
Master	Correlator	2092	Управление правилами корреляции

Исходящий	Входящий	Порты	Описание
Master	Log-Collector	4805 4806	Мониторинг и сбор статистики. Управление сборщиком событий API
Master	Balancer Correlator Data Worker	6676	Управление агентами кластера
Master	Balancer Correlator Data Worker	9100	Мониторинг и сбор статистики
Master	Balancer	9292, 9308	Мониторинг и сбор статистики
Master	Data	9114	Мониторинг и сбор статистики
Worker	Balancer	9092	Чтения событий из менеджера очередей для их обработки
Worker	Master	5672	Передача нормализованных событий в очередь на корреляцию
Worker	Data	9200	Передача событий на хранение
Источники событий	Log-Collector	162 SNMP trap; 4807 UDP receiver; 4808 TCP receiver; 4809 TCP receiver SSL/TLS; 4810 HTTP receiver; 4811 HTTPS receiver; 4812 NetFlow reciever	Пассивный сбор событий
Пользователи Платформы	Master	8080 9000 6676 6677	Доступ к интерфейсу платформы, проверка API ключей
Data (с ролью Master)	Data (с ролью Data)	9300	Взаимодействие между нодами в кластере хранилища данных

15. Список доступных таймзон

Africa/Abidjan	America/Indiana/Winamac	Asia/Ho_Chi_Minh
Africa/Accra	America/Indianapolis	Asia/Hong_Kong

Africa/Abidjan	America/Indiana/Winamac	Asia/Ho_Chi_Minh
Africa/Addis_Ababa	America/Inuvik	Asia/Hovd
Africa/Algiers	America/Iqaluit	Asia/Irkutsk
Africa/Asmara	America/Jamaica	Asia/Istanbul
Africa/Asmera	America/Jujuy	Asia/Jakarta
Africa/Bamako	America/Juneau	Asia/Jayapura
Africa/Bangui	America/Kentucky/Louisville	Asia/Jerusalem
Africa/Banjul	America/Kentucky/Monticello	Asia/Kabul
Africa/Bissau	America/Knox_IN	Asia/Kamchatka
Africa/Blantyre	America/Kralendijk	Asia/Karachi
Africa/Brazzaville	America/La_Paz	Asia/Kashgar
Africa/Bujumbura	America/Lima	Asia/Kathmandu
Africa/Cairo	America/Los_Angeles	Asia/Katmandu
Africa/Casablanca	America/Louisville	Asia/Khandyga
Africa/Ceuta	America/Lower_Princes	Asia/Kolkata
Africa/Conakry	America/Maceio	Asia/Krasnoyarsk
Africa/Dakar	America/Managua	Asia/Kuala_Lumpur
Africa/Dar_es_Salaam	America/Manaus	Asia/Kuching
Africa/Djibouti	America/Marigot	Asia/Kuwait
Africa/Douala	America/Martinique	Asia/Macao
Africa/El_Aaiun	America/Matamoros	Asia/Macau
Africa/Freetown	America/Mazatlan	Asia/Magadan
Africa/Gaborone	America/Mendoza	Asia/Makassar
Africa/Harare	America/Menominee	Asia/Manila
Africa/Johannesburg	America/Merida	Asia/Muscat
Africa/Juba	America/Metlakatla	Asia/Nicosia
Africa/Kampala	America/Mexico_City	Asia/Novokuznetsk
Africa/Khartoum	America/Miquelon	Asia/Novosibirsk
Africa/Kigali	America/Moncton	Asia/Omsk
Africa/Kinshasa	America/Monterrey	Asia/Oral
Africa/Lagos	America/Montevideo	Asia/Phnom_Penh
Africa/Libreville	America/Montreal	Asia/Pontianak
Africa/Lome	America/Montserrat	Asia/Pyongyang
Africa/Luanda	America/Nassau	Asia/Qatar
Africa/Lubumbashi	America/New_York	Asia/Qyzylorda
Africa/Lusaka	America/Nipigon	Asia/Rangoon
Africa/Malabo	America/Nome	Asia/Riyadh
Africa/Maputo	America/Noronha	Asia/Saigon
Africa/Maseru	America/North_Dakota/Beulah	Asia/Sakhalin
Africa/Mbabane	America/North_Dakota/Center	Asia/Samarkand

Africa/Abidjan	America/Indiana/Winamac	Asia/Ho_Chi_Minh
Africa/Mogadishu	America/North_Dakota/New_Salem	Asia/Seoul
Africa/Monrovia	America/Ojinaga	Asia/Shanghai
Africa/Nairobi	America/Panama	Asia/Singapore
Africa/Ndjamena	America/Pangnirtung	Asia/Srednekolymsk
Africa/Niamey	America/Paramaribo	Asia/Taipei
Africa/Nouakchott	America/Phoenix	Asia/Tashkent
Africa/Ouagadougou	America/Port	au
Africa/Porto	Novo	America/Port_of_Spain
Africa/Sao_Tome	America/Porto_Acre	Asia/Tel_Aviv
Africa/Timbuktu	America/Porto_Velho	Asia/Thimbu
Africa/Tripoli	America/Puerto_Rico	Asia/Thimphu
Africa/Tunis	America/Punta_Arenas	Asia/Tokyo
Africa/Windhoek	America/Rainy_River	Asia/Tomsk
America/Adak	America/Rankin_Inlet	Asia/Ujung_Pandang
America/Anchorage	America/Recife	Asia/Ulaanbaatar
America/Anguilla	America/Regina	Asia/Ulan_Bator
America/Antigua	America/Resolute	Asia/Urumqi
America/Araguaina	America/Rio_Branco	Asia/Ust
America/Argentina/Buenos_Aires	America/Rosario	Asia/Vientiane
America/Argentina/Catamarca	America/Santa_Isabel	Asia/Vladivostok
America/Argentina/ComodRivadavia	America/Santarem	Asia/Yakutsk
America/Argentina/Cordoba	America/Santiago	Asia/Yangon
America/Argentina/Jujuy	America/Santo_Domingo	Asia/Yekaterinburg
America/Argentina/La_Rioja	America/Sao_Paulo	Asia/Yerevan
America/Argentina/Mendoza	America/Scoresbysund	Atlantic/Azores
America/Argentina/Rio_Gallegos	America/Shiprock	Atlantic/Bermuda
America/Argentina/Salta	America/Sitka	Atlantic/Canary
America/Argentina/San_Juan	America/St_Barthelemy	Atlantic/Cape_Verde
America/Argentina/San_Luis	America/St_Johns	Atlantic/Faeroe
America/Argentina/Tucuman	America/St_Kitts	Atlantic/Faroe
America/Argentina/Ushuaia	America/St_Lucia	Atlantic/Jan_Mayen
America/Aruba	America/St_Thomas	Atlantic/Madeira
America/Asuncion	America/St_Vincent	Atlantic/Reykjavik
America/Atikokan	America/Swift_Current	Atlantic/South_Georgia
America/Atka	America/Tegucigalpa	Atlantic/St_Helena
America/Bahia	America/Thule	Atlantic/Stanley
America/Bahia_Banderas	America/Thunder_Bay	Australia/ACT
America/Barbados	America/Tijuana	Australia/Adelaide
America/Belem	America/Toronto	Australia/Brisbane

Africa/Abidjan	America/Indiana/Winamac	Asia/Ho_Chi_Minh
America/Belize	America/Tortola	Australia/Broken_Hill
America/Blanc	Sablon	America/Vancouver
America/Boa_Vista	America/Virgin	Australia/Currie
America/Bogota	America/Whitehorse	Australia/Darwin
America/Boise	America/Winnipeg	Australia/Eucla
America/Buenos_Aires	America/Yakutat	Australia/Hobart
America/Cambridge_Bay	America/Yellowknife	Australia/LHI
America/Campo_Grande	Antarctica/Casey	Australia/Lindeman
America/Cancun	Antarctica/Davis	Australia/Lord_Howe
America/Caracas	Antarctica/DumontDUrville	Australia/Melbourne
America/Catamarca	Antarctica/Macquarie	Australia/NSW
America/Cayenne	Antarctica/Mawson	Australia/North
America/Cayman	Antarctica/McMurdo	Australia/Perth
America/Chicago	Antarctica/Palmer	Australia/Queensland
America/Chihuahua	Antarctica/Rothera	Australia/South
America/Coral_Harbour	Antarctica/South_Pole	Australia/Sydney
America/Cordoba	Antarctica/Syowa	Australia/Tasmania
America/Costa_Rica	Antarctica/Troll	Australia/Victoria
America/Creston	Antarctica/Vostok	Australia/West
America/Cuiaba	Arctic/Longyearbyen	Australia/Yancowinna
America/Curacao	Asia/Aden	Brazil/Acre
America/Danmarkshavn	Asia/Almaty	Brazil/DeNoronha
America/Dawson	Asia/Amman	Brazil/East
America/Dawson_Creek	Asia/Anadyr	Brazil/West
America/Denver	Asia/Aqtau	CET
America/Detroit	Asia/Aqtobe	CST6CDT
America/Dominica	Asia/Ashgabat	Canada/Atlantic
America/Edmonton	Asia/Ashkhabad	Canada/Central
America/Eirunepe	Asia/Atyrau	Canada/Eastern
America/El_Salvador	Asia/Baghdad	Canada/Mountain
America/Ensenada	Asia/Bahrain	Canada/Newfoundland
America/Fort_Nelson	Asia/Baku	Canada/Pacific
America/Fort_Wayne	Asia/Bangkok	Canada/Saskatchewan
America/Fortaleza	Asia/Barnaul	Canada/Yukon
America/Glace_Bay	Asia/Beirut	Chile/Continental
America/Godthab	Asia/Bishkek	Chile/EasterIsland
America/Goose_Bay	Asia/Brunei	Cuba
America/Grand_Turk	Asia/Calcutta	EET
America/Grenada	Asia/Chita	EST

Africa/Abidjan	America/Indiana/Winamac	Asia/Ho_Chi_Minh
America/Guadeloupe	Asia/Choibalsan	EST5EDT
America/Guatemala	Asia/Chongqing	Egypt
America/Guayaquil	Asia/Chungking	Eire
America/Guyana	Asia/Colombo	Etc/GMT
America/Halifax	Asia/Dacca	Etc/GMT+0
America/Havana	Asia/Damascus	Etc/GMT+1
America/Hermosillo	Asia/Dhaka	Etc/GMT+10
America/Indiana/Indianapolis	Asia/Dili	Etc/GMT+11
America/Indiana/Knox	Asia/Dubai	Etc/GMT+12
America/Indiana/Marengo	Asia/Dushanbe	Etc/GMT+2
America/Indiana/Petersburg	Asia/Famagusta	Etc/GMT+3
America/Indiana/Tell_City	Asia/Gaza	Etc/GMT+4
America/Indiana/Vevay	Asia/Harbin	Etc/GMT+5
America/Indiana/Vincennes	Asia/Hebron	Etc/GMT+6
Europe/Amsterdam	GB	Etc/GMT+7
Europe/Andorra	GB-Eire	Etc/GMT+8
Europe/Astrakhan	GMT	Etc/GMT+9
Europe/Athens	GMT+0	Etc/GMT-0
Europe/Belfast	GMT-0	Etc/GMT-1
Europe/Belgrade	GMT0	Etc/GMT-10
Europe/Berlin	Greenwich	Etc/GMT-11
Europe/Bratislava	HST	Etc/GMT-12
Europe/Brussels	Hongkong	Etc/GMT-13
Europe/Bucharest	Iceland	Etc/GMT-14
Europe/Budapest	Indian/Antananarivo	Etc/GMT-2
Europe/Busingen	Indian/Chagos	Etc/GMT-3
Europe/Chisinau	Indian/Christmas	Etc/GMT-4
Europe/Copenhagen	Indian/Cocos	Etc/GMT-5
Europe/Dublin	Indian/Comoro	Etc/GMT-6
Europe/Gibraltar	Indian/Kerguelen	Etc/GMT-7
Europe/Guernsey	Indian/Mahe	Etc/GMT-8
Europe/Helsinki	Indian/Maldives	Etc/GMT-9
Europe/Isle_of_Man	Indian/Mauritius	Etc/GMT0
Europe/Istanbul	Indian/Mayotte	Etc/Greenwich
Europe/Jersey	Indian/Reunion	Etc/UCT
Europe/Kaliningrad	Iran	Etc/UTC
Europe/Kiev	Israel	Etc/Universal
Europe/Kirov	Jamaica	Etc/Zulu
Europe/Lisbon	Japan	Pacific/Norfolk

Africa/Abidjan	America/Indiana/Winamac	Asia/Ho_Chi_Minh
Europe/Ljubljana	Kwajalein	Pacific/Noumea
Europe/London	Libya	Pacific/Pago_Pago
Europe/Luxembourg	MET	Pacific/Palau
Europe/Madrid	MST	Pacific/Pitcairn
Europe/Malta	MST7MDT	Pacific/Pohnpei
Europe/Mariehamn	Mexico/BajaNorte	Pacific/Ponape
Europe/Minsk	Mexico/BajaSur	Pacific/Port_Moresby
Europe/Monaco	Mexico/General	Pacific/Rarotonga
Europe/Moscow	NZ	Pacific/Saipan
Europe/Nicosia	NZ	CHAT
Europe/Oslo	Navajo	Pacific/Tahiti
Europe/Paris	PRC	Pacific/Tarawa
Europe/Podgorica	PST8PDT	Pacific/Tongatapu
Europe/Prague	Pacific/Apia	Pacific/Truk
Europe/Riga	Pacific/Auckland	Pacific/Wake
Europe/Rome	Pacific/Bougainville	Pacific/Wallis
Europe/Samara	Pacific/Chatham	Pacific/Yap
Europe/San_Marino	Pacific/Chuuk	Poland
Europe/Sarajevo	Pacific/Easter	Portugal
Europe/Saratov	Pacific/Efate	ROC
Europe/Simferopol	Pacific/Enderbury	ROK
Europe/Skopje	Pacific/Fakaofo	Singapore
Europe/Sofia	Pacific/Fiji	Turkey
Europe/Stockholm	Pacific/Funafuti	UCT
Europe/Tallinn	Pacific/Galapagos	US/Alaska
Europe/Tirane	Pacific/Gambier	US/Aleutian
Europe/Tiraspol	Pacific/Guadalcanal	US/Arizona
Europe/Ulyanovsk	Pacific/Guam	US/Central
Europe/Uzhgorod	Pacific/Honolulu	US/East
Europe/Vaduz	Pacific/Johnston	US/Eastern
Europe/Vatican	Pacific/Kiritimati	US/Hawaii
Europe/Vienna	Pacific/Kosrae	US/Indiana
Europe/Vilnius	Pacific/Kwajalein	US/Michigan
Europe/Volgograd	Pacific/Majuro	US/Mountain
Europe/Warsaw	Pacific/Marquesas	US/Pacific
Europe/Zagreb	Pacific/Midway	US/Pacific
Europe/Zaporozhye	Pacific/Nauru	US/Samoa
Europe/Zurich	Pacific/Niue	UTC
		Universal

Africa/Abidjan	America/Indiana/Winamac	Asia/Ho_Chi_Minh
		W-SU
		WET
		Zulu

16. Включение режима распределенной корреляции

Для включения режима распределенной корреляции необходимо добавить узел в кластер платформы и назначить этому узлу роль **Correlator**. После этого выполнить настройку всех экземпляров коррелятора.

16.1. Настройка экземпляров коррелятора

Включение функции распределенной корреляции осуществляется на всех экземплярах коррелятора в конфигурационном файле `/opt/pangeoradar/configs/logmule/conf.yaml`.

1. Зайдите в раздел «**Администрирование**» — «**Кластер**» — «**Узлы**», найдите узлы с ролями **Correlator** и узнайте их IP-адреса.
2. Подключитесь к узлу по SSH и добавьте параметр **shared_instance** в конфигурационный файл узла с помощью команды:

   `nano /opt/pangeoradar/configs/logmule/conf.yaml`
3. Для включения в конфигурационный файл нужно добавить следующую строку: `shared_instance: true`. Сохраните изменения.
4. Если необходимо, подключитесь ко следующему узлу аналогично пункту 2.
5. Тогда для первого узла коррелятора выставите значение `shared_instance: false`. Для второго узла — `shared_instance: true`.
6. Сохраните конфигурационные файлы и перезапустите службу с помощью команды:

   `service pangeoradar-logmule restart`

16.2. Настройка правила для работы с несколькими корреляторами

Для переключения правила в режим распределенной корреляции нужно сделать следующее:

1. В веб-интерфейсе перейдите в раздел «**Коррелятор**» — «**Правила**».
2. Нажмите на пиктограмму карандаша у правила, которое необходимо переключить в режим распределенной корреляции.
3. В режиме редактирования правила создайте новое хранилище значений с названием **shared_memory** (переменные можно оставить пустыми).
4. Добавьте хранилище **shared_memory** к правилу как изображено на Рисунке 1.
5. Сохраните изменения.

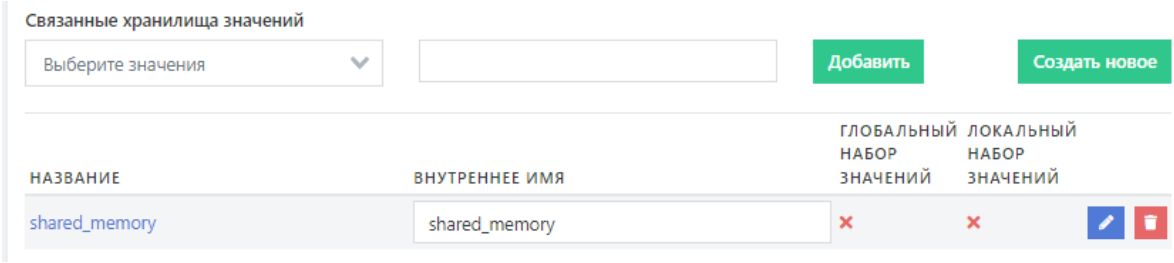

Рисунок 80 - Добавление хранилища shared_memory

Для проверки работы режима распределенной корреляции, необходимо перейти в раздел «**Коррелятор**» — «**Правила**».

При переключении между экземплярами коррелятора правило, в которое было добавлено хранилище **shared_memory**, будет помечено как активное.

Для переключения между экземплярами коррелятора необходимо нажать на выпадающее меню с названием экземпляра, как изображено на Рисунке 2.

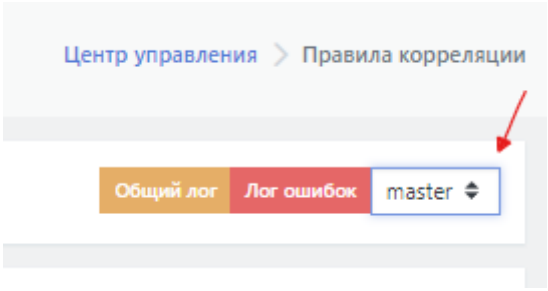

Рисунок 81 - Переключение между экземплярами коррелятора

16.3. Проверка работы правила

Для проверки работы правила при распределенной коммутации необходимо сгенерировать ситуацию, для которой выбранное для проверки правило будет срабатывать. Например, для выбранного для проверки правила **Account_added_and_removed_from_a_group_in_short_period_of_time** необходимо выполнить следующие действия:

1. Подключиться по RDP к лог-коллектору.
2. Запустить командную строку от имени Администратора.
3. Выполнить попытку добавление нового пользователя в Платформу, выполнив команду:

   `C:\Log-collector\user_add_to_group.cmd`

После завершения выполнения команды перейти в веб-интерфейс Платформы в раздел «**Инциденты**». При правильной отработке правила в данном разделе в списке инцидентов отобразится инцидент с названием «**MS-WIN – Пользователь добавлен в локальную группу или удален из нее**» и IP-адресом лог-коллектора.

17. Настройка интеграции со службой Active Directory

В платформе предусмотрена возможность использования доменных учетных записей по средствам интеграции с Active Directory.

Для настройки интеграции необходимо:

- указать адрес LDAP сервера,
- указать аккаунт и пароль для поиска по LDAP в настройках KeyCloak.

Если на контроллере(ах) домена LDAP ранее не настраивался, то необходимо установить **Microsoft Identity Management for UNIX Role Service** (см. Рисунок 82).

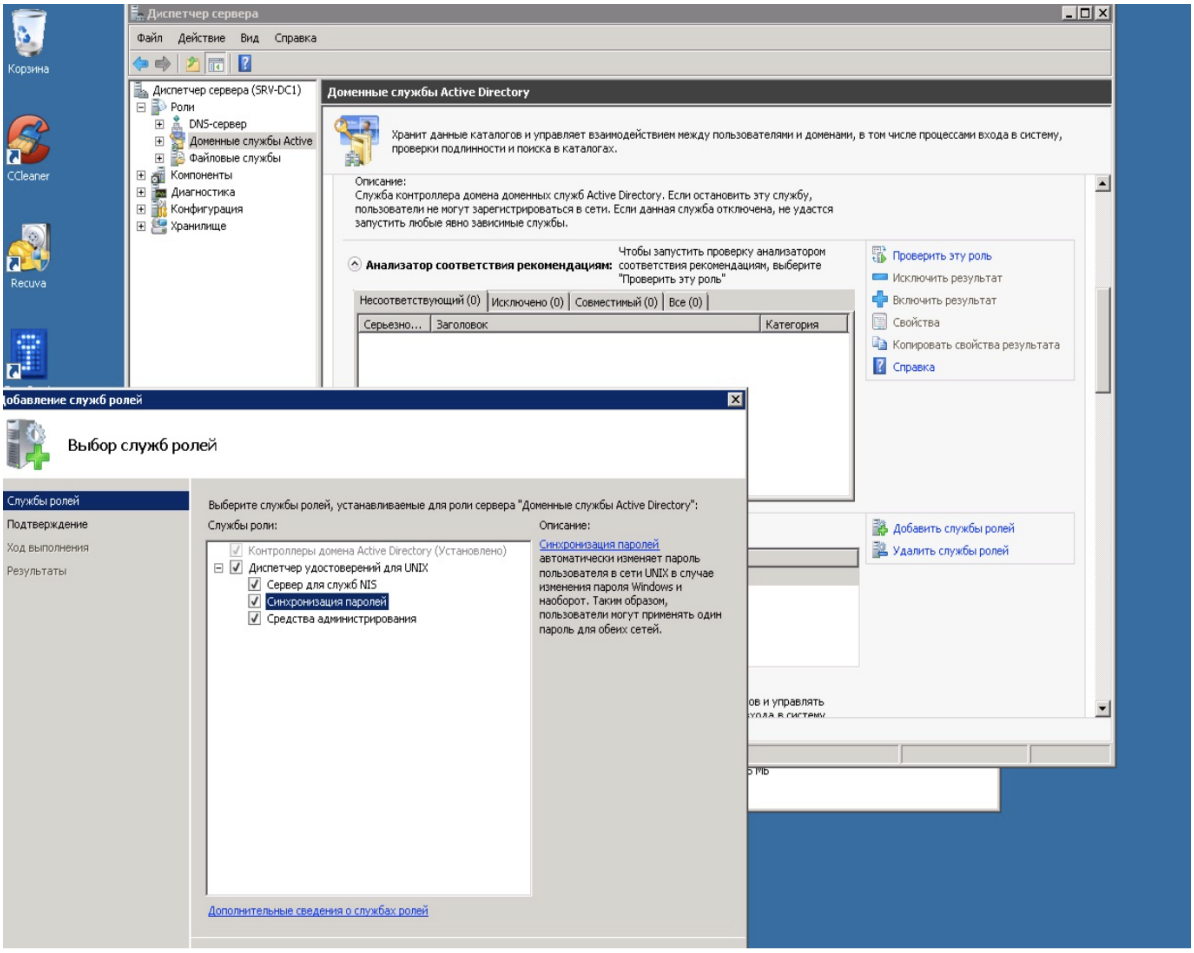

Рисунок 82 - Выбор служб ролей в Microsoft Identity Management for UNIX Role Service

17.1. Настройка LDAP

После установки службы перейдите в KeyCloak и начните настройку LDAP, выполнив следующие действия:

1. Откройте консоль администрирования **KeyCloak** (<адрес Платформы>:8180) и перейдите в пункт меню "**User Federation**" (см. Рисунок 83).

2. Откройте список "**Add Provider**" (см. Рисунок 83).

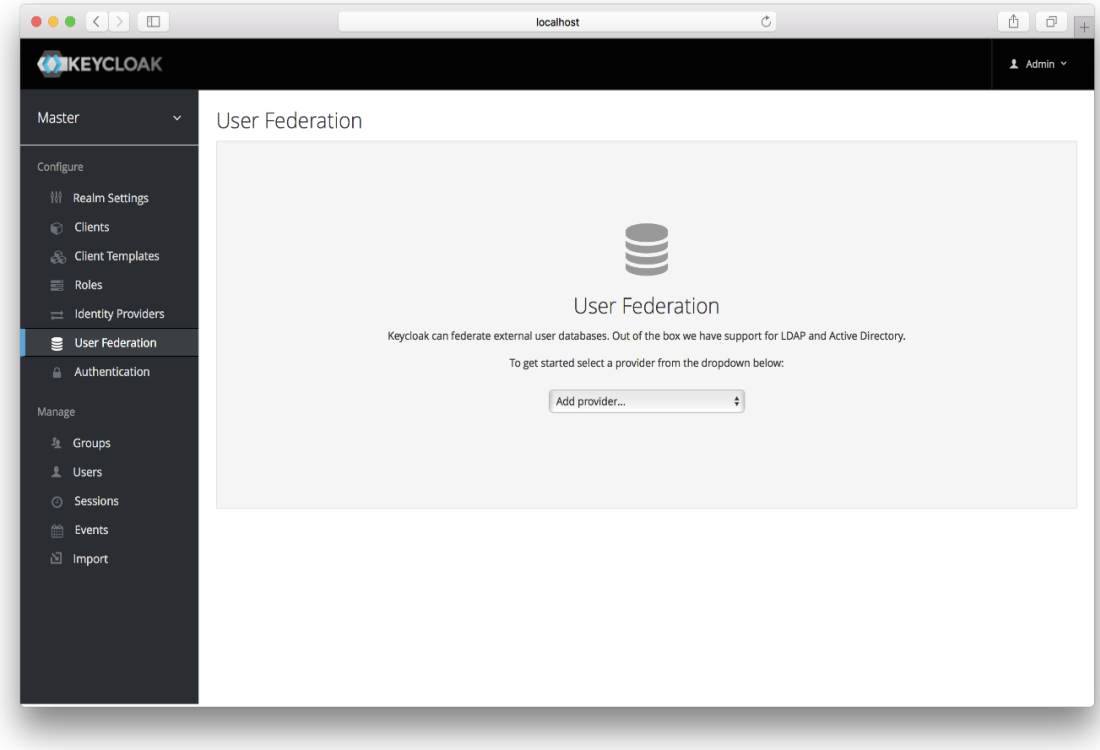

Рисунок 83 - Консоль администрирования KeyCloak, раздел меню "User Federation", список "Add Provider"

3. В открывшемся списке "**Add Provider**" выберите раздел "**LDAP**" и заполните на вкладке "Settings" предоставленные программой поля (см. Рисунок 84).
   Следующие поля обязательны для заполнения:

- `Edit Mode` - значение READ_ONLY (устанавливается по умолчанию);
- `Vendor` - указать Active Directory;
- `Username LDAP attribute` - указать sAMAccountName;
- `RDN LDAP attribute` - значение cn (установлено по умолчанию);
- `UUID LDAP attribute` - значение objectGUID (установлено по умолчанию);
- `User Object Classes` - значения person, organizationPerson, user (установлены по умолчанию);
- `Connection URL` - указать IP-адрес сервера Active Directory, например - ldap://srv-dc2.youdomain.local ;
- `Users DN` - в соответствии с примером `DC=youdomain,DC=local` ;
- `Authentication Type` - выбрать Simple;
- `Bind DN` - указать системный аккаунт в Active Director для чтения данных из LDAP (например, ldap-ro-user@youdoman.local) ;
- `Bind Credential` - пароль системного аккаунта;
- `Search Scope` - выберите Subtree.

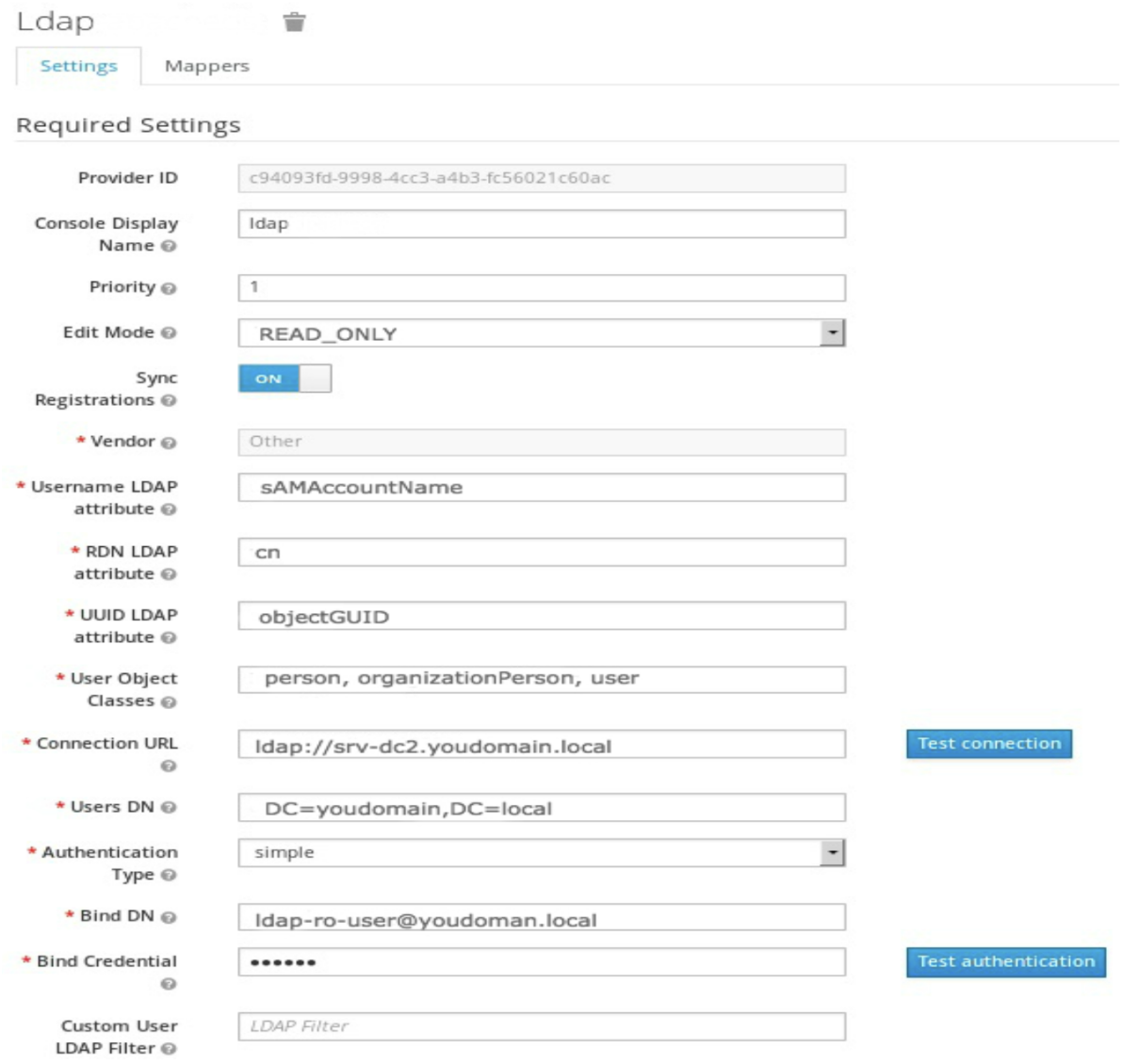

Рисунок 84 - Заполнение данных по LDAP

4. При необходимости можно протестировать введенные параметры LDAP, нажав последовательно кнопки "**Test connection**" и "**Test authentication**" (см. Рисунок 84).
5. Для сохранения введённых настроек LDAP нажмите кнопку **Save**, расположенную в самом низу экрана.
6. Перейдите к процедуре настройки маппинга, который необходим для подмены атрибутов LDAP. Для этого перейдите на вкладку "Mappers" и выбрать пункт **Username** (см. Рисунок 85).
7. Укажите следующие характеристики:

- `LDAP attribute` - указать sAMAccountName;

8. Сохраните введенные настройки маппинга, нажав **Save** (см. Рисунок 85).

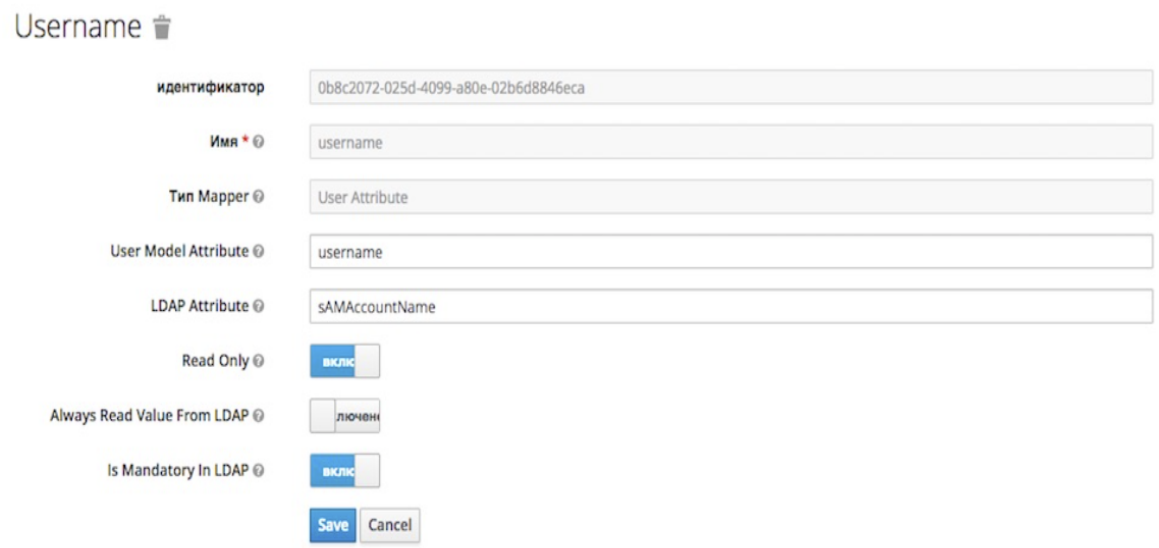

Рисунок 85 - Настройка маппинга

17.2. Синхронизация доменных пользователей

После настройки LDAP необходимо провести синхронизацию доменных пользователей. Для этого выполните следующие действия:

1. Вернитесь на основную страницу раздела "User Federation" и выберите созданный LDAP.
2. Для импорта пользователей из Active Directory нажмите кнопку **Synchronize all users** (см. Рисунок 86).

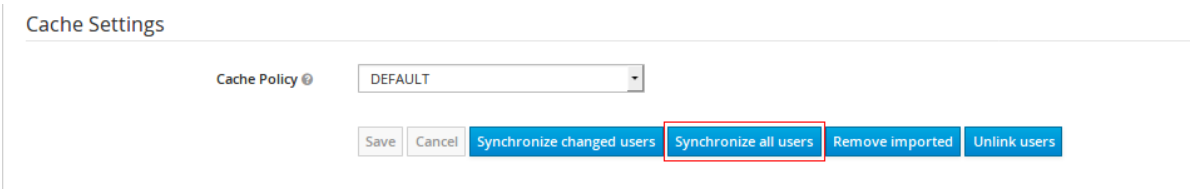

Рисунок 86 - Синхронизация доменных пользователей в разделе "User Federation"

Для проверки проведённого импорта пользователей из Active Directory перейдите в раздел основного меню "Users" и нажмите кнопку **View all users**. На экране должен открыться список синхронизированных пользователей.

17.3. Определение возможных причин сбоя при синхронизации

Если синхронизация пользователей не произошла, то для определения причины сбоя в первую очередь надо смотреть лог плагина /opt/wildfly/standalone/log/keycloak.log. В логе следует просмотреть события, зафиксированные в момент нажатия тестовых кнопок или кнопок синхронизации пользователей.

18. Служба уведомлений Toller

18.1. Назначение ПО

Данный программный модуль предназначен для формирования уведомлений от Платформы Радар и пересылки сформированных уведомлений пользователям и администраторам.

18.2. Конфигурационный файл Toller

Расположение конфигурационного файла: `/opt/pangeoradar/configs/pangeoradar-toller.yaml`

Пример конфигурационного файла:

```
UI: "https://<IP-адрес-Платформы>"
RMCA: "http://127.0.0.1:8086"
host: ""
port: 6699
PGAddr: "<IP-адрес-Платформы>:5432"
PGUser: "<Логин от учетной записи базы данных>"
PGPass: "<Пароль от учетной записи базы данных>"
PGDB: "<Имя экземпляра базы данных службы>"
AuthHost: "https://<IP-адрес-Платформы>:8180"
InstanceID: "<ID экземпляра Платформы>"
Debug: false

SmtpEnable: true
SmtpIdentity: ""
SmtpUsername: "<Логин от учетной записи SMTP>"
SmtpPassword: "<Пароль от записи SMTP>"
SmtpAddress: "<Адрес SMTP-сервера>"
SmtpPort: "<Порт SMTP-сервера>"
SmtpFrom: "<Адрес отправителя уведомлений>"
SmtpDefaultTo: "<Адрес пересылки для всех уведомлений>"

SlackEnable: false
SlackDefaultWebHook: "<WebHook для уведомлений Slack>"

UseTLS: true
SkipTLSVerify: true
PgCert: "<Путь до сертификата>"
PgKey: "<Путь до ключа>"
RootCrt: "<Путь до корневого сертификата>"
```

18.3. Настройка пользователей

Для настройки получения уведомлений от Платформы Радар конкретными пользователями необходимо выполнить следующие шаги:

1. Зайти в интерфейс Платформы Радар с правами администратора;
2. Перейти в раздел "Администрирование", "Пользователи и права", как изображено на рисунке 87;

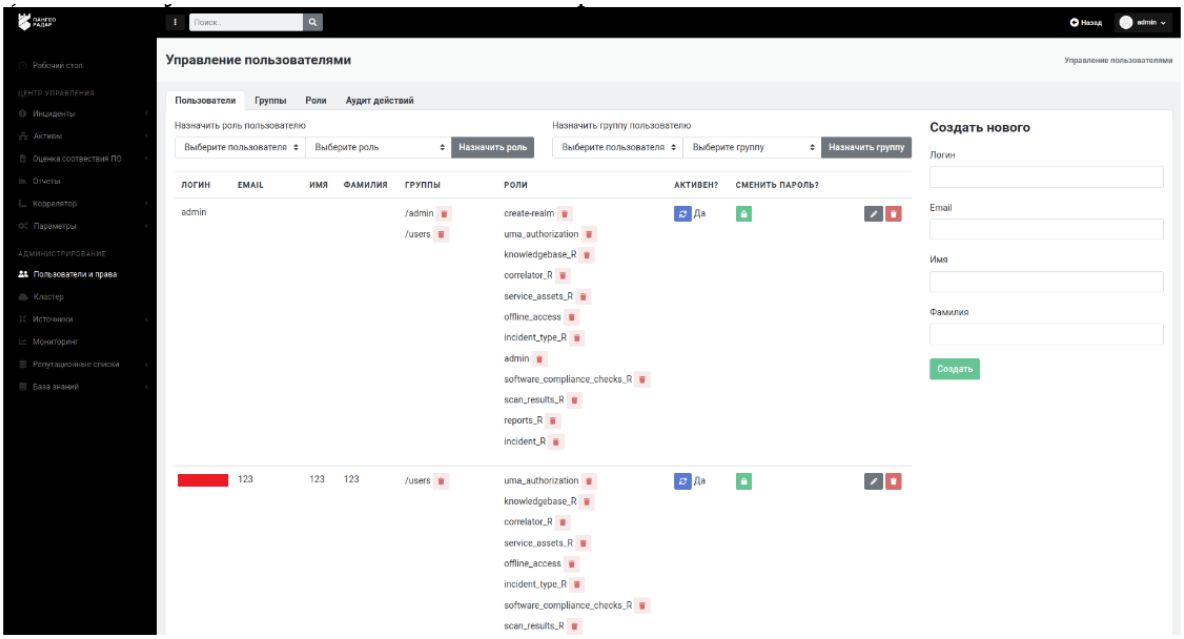

Рисунок 87 - "Пользователи и права"

3. Найти или создать нужного пользователя и нажать на кнопку редактирования  в строке данного пользователя;

   В правой части страницы откроется форма редактирования параметров пользователя, как изображено на рисунке 88;

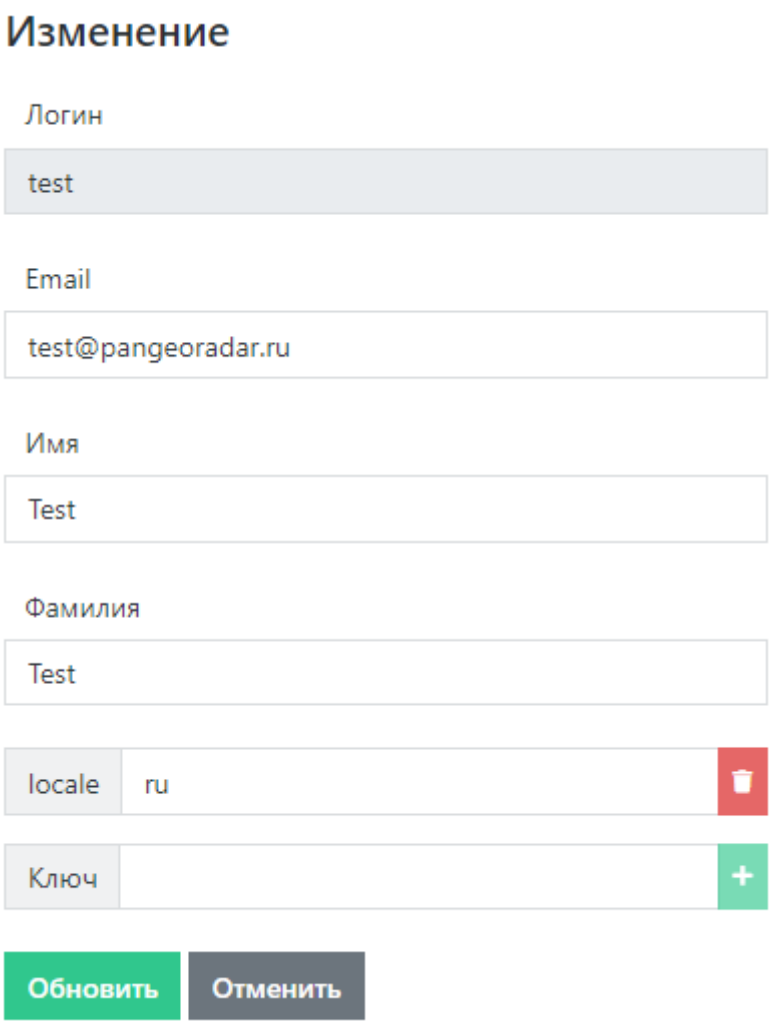

Рисунок 88 - "Параметры пользователя"

4. Указать актуальный "Email" пользователя, на который предусмотрена отправка уведомлений;
5. Нажать кнопку "Обновить" для сохранения введённых настроек;
6. Авторизоваться под только что созданным\отредактированным пользователем;
7. Перейти в настройки профиля данного пользователя, нажав на имя пользователя в правом верхнем углу интерфейса и нажав на кнопку "Профиль";
8. Произвести необходимые настройки оповещений, как изображено на рисунке 89;

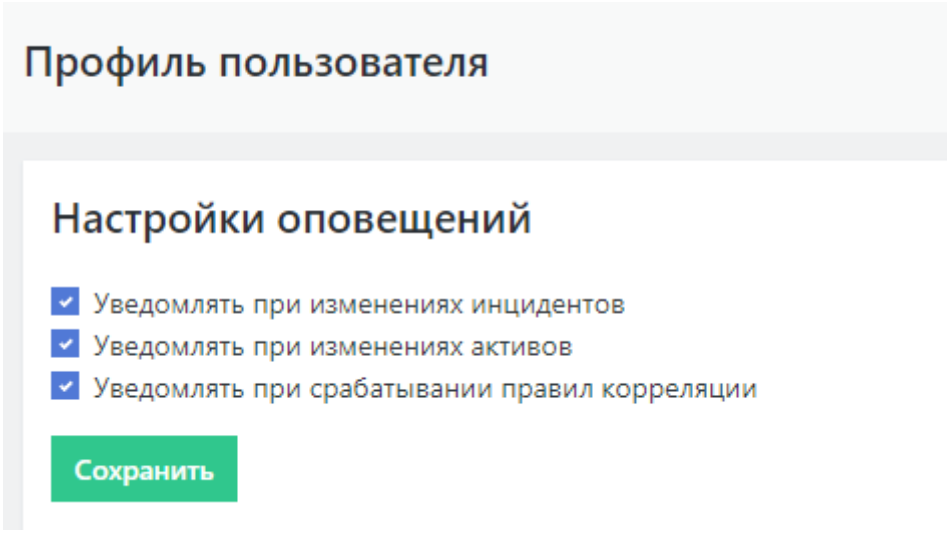

Рисунок 89 - "Настройки оповещений"

9. Нажать кнопку "Сохранить" для сохранения настроек.

При корректном выполнении вышеописанных действий данный пользователь будет получать уведомления Платформы Радар на указанный в параметрах учетной записи почтовый адрес.

18.4. Настройка оповещений о работе сервисов

Для настройки оповещений о работе сервисов необходимо сделать следующее:

1. Произвести настройку службы node_exporter;

   Расположение конфигурационного файла: `/etc/systemd/system/node_exporter.service`

   В конец строки ExecStart добавить `--collector.systemd`

   После чего конфигурационный файл должен выглядеть следующим образом:

```
[Unit]
Description=Node Exporter
Wants=network-online.target
After=network-online.target

[Service]
User=node_exporter
Group=node_exporter
Type=simple
ExecStart=/opt/pangeoradar/node_exporter/node_exporter --web.listen-
address=":9101" --collector.systemd

[Install]
WantedBy=multi-user.target
```

2. Далее необходимо выполнить команду `systemctl daemon-reload`
3. После чего, перезапустить службу node_exporter командой `service node_exporter restart`

!!! note "Оповещения будут отправляться на адрес, указанный в параметре SmtpDefaultTo" конфигурационного файла `/opt/pangeoradar/configs/pangeoradar-toller.yaml`"

19. Резервное копирование

19.1. Утилиты для снятия резервной копии ElasticSearch

Ниже представлен один из способов работы с снятием резервных копий индексов ElasticSearch путем архивирования индексов. Важно помнить, что для корректной работы потребуется curator версии старше 5.0.

19.1.1. Архивирование индексов

В файле `/etc/elasticsearch/elasticsearch.yml`, прописан путь до файлового репозитория:

```
path.repo: /opt/elasticsearch/snapshots
```

Если такой строки нет, необходимо прописать и перезагрузить elasticsearch.

Далее необходимо создать репозиторий, в котором будут размещены снапшоты:

```
mkdir -p /opt/elasticsearch/snapshots/repository
curl -XPUT 'http://localhost:9200/_snapshot/repository' -H 'Content-Type:
application/json' -d '{
    "type": "fs",
    "settings": {
        "location": "repository",
        "compress": true
    }
}'
```

Также необходимо создать ещё один репозиторий с именем «recovery», который понадобится для восстановления индексов:

```
mkdir -p /opt/elasticsearch/snapshots/recovery
curl -XPUT 'http://localhost:9200/_snapshot/recovery' -H 'Content-Type:
application/json' -d '{
    "type": "fs",
    "settings": {
        "location": "recovery",
        "compress": true
    }
}'
```

Далее представл пример скрипта для архивирования индексов.
Логика работы скрипта описана в комментариях. Не забудьте поправить значения переменных, если ваши настройки будут отличаться от дефолтных.

```
#!/bin/bash

DAYS=31 #Количество дней, от текущей даты, старше которого индексы будут
архивироваться
SNAPSHOT_DIRECTORY="/opt/elasticsearch/snapshots"
BACKUP_DIR="/opt/elasticsearch/elasticsearch_backup"
REPOSITORY="repository"
LOG="/var/log/elasticsearch/elasticsearch_backup.log"
DATE=`date`

#Проверим существование папки для архивов и если нет, создадим её
if ! [ -d $BACKUP_DIR ]; then
  mkdir -p $BACKUP_DIR
fi

#Получаем массив индексов, которые старше $DAYS
INDICES=`curator_cli --config /etc/elasticsearch/curator-config.yml --host
localhost --port 9200 show_indices --filter_list "
[{\"filtertype\":\"age\",\"source\":\"creation_date\",\"direction\":\"older\",\"
unit\":\"days\",\"unit_count\":\"$DAYS\"},
{\"filtertype\":\"kibana\",\"exclude\":\"True\"},
{\"filtertype\":\"pattern\",\"kind\":\"regex\",\"value\":\"elastalert_status\",\
"exclude\":\"True\"}]"`

#Проверим, не пустой ли список
TEST_INDICES=`echo $INDICES | grep -q -i "error" && echo 1 || echo 0`
```

```
if [ $TEST_INDICES == 1 ]
then
  echo "$DATE Не найдено индексов для обработки" >> $LOG
  exit
else
# Составляем цикл для каждого индекса в массиве $INDICES
for i in $INDICES
  do
    # Создаём снапшот для индекса $i
    curator_cli --config /etc/elasticsearch/curator-config.yml --timeout 600 --
host localhost --port 9200 snapshot --repository $REPOSITORY --filter_list "
{\"filtertype\":\"pattern\",\"kind\":\"regex\",\"value\":\"$i\"}"

    # Заносим в переменную имя снапшота для индекса $i
    SNAPSHOT=`curator_cli --config /etc/elasticsearch/curator-config.yml --host
localhost --port 9200 show_snapshots --repository $REPOSITORY`

    # Архивируем папку репозитория и складываем архив в хранилище
    cd $SNAPSHOT_DIRECTORY/$REPOSITORY && tar cjf $BACKUP_DIR"/"$i".tar.bz" ./*

    # Удаляем snapshot
    curator_cli --config /etc/elasticsearch/curator-config.yml --host localhost
--port 9200 delete_snapshots --repository $REPOSITORY --filter_list "
{\"filtertype\":\"pattern\",\"kind\":\"regex\",\"value\":\"$SNAPSHOT\"}"

    # Удаляем индекс
    curator_cli --config /etc/elasticsearch/curator-config.yml --host localhost
--port 9200 delete_indices --filter_list "
{\"filtertype\":\"pattern\",\"kind\":\"regex\",\"value\":\"$i\"}"

    # Очищаем папку репозитория
    rm -rf $SNAPSHOT_DIRECTORY/$REPOSITORY/*
  done
fi
```

19.1.2. Удаление устаревших архивов

Ниже представлен скрипт для удаления устаревших архивов индексов.

```
#!/bin/bash

# Удаление бекапов старше $DAYS дней
# ВАЖНО! В имени файла архива может быть только один знак "-" перед датой. Дата
должна быть в формате "yyyy.mm.dd".
# Например: aaa_bbb.ccc-yyyy.mm.dd.tar.bz

DAYS=91
BACKUP_DIR="/opt/elasticsearch/elasticsearch_backup"

#Определяем пороговую дату для удаления архивов
THRESHOLD=$(date -d "$DAYS days ago" +%Y%m%d)

#echo "THRESHOLD=$THRESHOLD"

FILES=`ls -1 $BACKUP_DIR`
```

```
TODELETE=`for i in $FILES; do echo $i | awk -F- '{printf "%s\n",$2 ;}' | awk -F.
'{printf "%s%s%s \n",$1,$2,$3 ;}' | sed "s/$/$i/"; done`

echo -e "$TODELETE" |\
while read DATE FILE
  do
     [[ $DATE -le $THRESHOLD ]] && rm -rf $BACKUP_DIR/$FILE
  done
```

Как правило, удалять устаревшие копии необходимо регулярно, и обычно это делается в периоды наименьшей нагрузки на сервер. Вы можете задать команду на запуск выше представленного скрипта как задачу планировщика (cron).

19.1.3. Восстановление индексов из архива

Ниже представлен скрипт для восстановления индекса из архива. Скрипт принимает первым аргументом путь до архива.

```
#!/bin/bash

#Зададим переменные
ARCHIVE=$1
BACKUP_DIR="/opt/elasticsearch/elasticsearch_backup"
RECOVERY_DIR="/opt/elasticsearch/snapshots/recovery/"

# На всякий случай очищаем папку репозитория
rm -rf $RECOVERY_DIR/*

# Разархивируем индекс в папку репозитория
tar xjf $BACKUP_DIR/$ARCHIVE -C $RECOVERY_DIR

# Заносим в переменную $SNAPSHOT имя снапшота в репозитории
SNAPSHOT=`curl -s -XGET "localhost:9200/_snapshot/recovery/_all?pretty" | jq
'.snapshots[0].snapshot' | sed 's/\"//g'`

# Восстанавливаем индекс из снапшота
curl -XPOST "localhost:9200/_snapshot/recovery/$SNAPSHOT/_restore?pretty"

# Нужно выставить небольшую задержку, чтобы Elasticsearch не ругался на удаление
восстанавливаемого снапшота
sleep 30

# Удалим снапшот из репозитория
curl -XDELETE "localhost:9200/_snapshot/recovery/$SNAPSHOT?pretty"

# Очистим папку репозитория
rm -rf $RECOVERY_DIR/*
```

19.2. Утилиты для снятия резервной копии MongoDB

MongoDB использует для хранения информации форматы JSON и BSON (двоичный JSON). JSON - это удобный для прочтения человеком формат, идеально подходящий для экспорта и импорта данных. Вы сможете управлять экспортированными данными в этом формате с помощью любого инструмента, поддерживающего JSON, включая простой текстовый редактор.

Для создания резервных копий и восстановления, лучше использовать двоичный формат BSON.

Согласованность информации может представлять проблему, если у вас загруженный сервер MongoDB, где информация может изменяться при экспорте или резервном копировании базы данных. Одно из возможных решений этой проблемы — репликация, и вы можете использовать его, когда лучше освоитесь с MongoDB.

Чтобы создать резервную копию данных, вам следует использовать команду `mongodump`. Для восстановления используйте команду `mongorestore`.

19.2.1. Утилита mongodump

создайте каталог /var/backups/mongobackups:

```
sudo mkdir /var/backups/mongobackups
```

Затем запустите команду `mongodump`:

```
mongodump --db newdb --out /var/backups/mongobackups/`date +"%m-%d-%y"`
```

Результат должен будет выглядеть следующим образом:

```
Output
2020-10-29T19:22:36.886+0000    writing newdb.restaurants to
2020-10-29T19:22:36.969+0000    done dumping newdb.restaurants (25359 documents)
```

Теперь у вас имеется полная резервная копия базы данных newdb в каталоге /var/backups/mongobackups/10-29-20/newdb/.

Как правило, резервное копирование выполняется регулярно, и обычно это делается в периоды наименьшей нагрузки на сервер. Вы можете задать команду mongodump как задачу планировщика (cron)

Используя аргумент `--drop` мы обеспечим предварительное отбрасывание целевой базы данных так, чтобы резервная копия была восстановлена в чистой базе данных.

```
sudo mongorestore --db newdb --drop /var/backups/mongobackups/10-29-20/newdb/
```

Результат должен будет выглядеть следующим образом:

```
Output
2020-10-29T19:25:45.825+0000    the --db and --collection args should only be
used when restoring from a BSON file. Other uses are deprecated and will not
exist in the future; use --nsInclude instead
2020-10-29T19:25:45.826+0000    building a list of collections to restore from
/var/backups/mongobackups/10-29-20/newdb dir
2020-10-29T19:25:45.829+0000    reading metadata for newdb.restaurants from
/var/backups/mongobackups/10-29-20/newdb/restaurants.metadata.json
2020-10-29T19:25:45.834+0000    restoring newdb.restaurants from
/var/backups/mongobackups/10-29-20/newdb/restaurants.bson
2020-10-29T19:25:46.130+0000    no indexes to restore
2020-10-29T19:25:46.130+0000    finished restoring newdb.restaurants (25359
documents)
2020-10-29T19:25:46.130+0000    done
```

19.2.2. Утилита mongorestore

При восстановлении базы данных MongoDB из предыдущей резервной копии вы получите точную копию информации MongoDB на определенный момент времени, включая все индексы и типы данных.

19.3. Утилиты для снятия резервной копии PostgreSQL

19.3.1. Утилита pg_dumpall

Утилита *pg_dumpall* реализует резервное копирование всего экземпляра (кластера или инстанса) базы данных без указания конкретной базы данных на инстансе. По принципу схожа с *pg_dump*. Добавим, что только утилиты *pg_dump* и *pg_dumpall* предоставляют возможность создания логической копии данных, остальные утилиты, рассматриваемые в этой статье, позволяют создавать только бинарные копии.

```
# pg_dumpall > /tmp/instance.bak
```

Чтобы сразу сжать резервную копию экземпляра базы данных, нужно передать вывод на архиватор *gzip*:

```
# pg_dumpall | gzip > /tmp/instance.tar.gz
```

Ниже приведены параметры, с которыми может вызываться утилита *pg_dumpall*.

-d <имя*бд*>, —*dbname=имя*бд — имя базы данных.

-h <сервер>, —host=сервер — имя сервера.

-p <порт>, —port=порт — TCP-порт, на который принимаются подключения.

-U <пользователь>, —username=пользователь — имя пользователя для подключения.

-w, —no-password — деактивация требования ввода пароля.

-W, —password — активация требования ввода пароля.

—role=<имя роли> — роль, от имени которой генерируется резервная копия.

-a, —data-only — создание резервной копии без схемы данных.

-c, —clean — добавление операторов DROP перед операторами CREATE.

-f <имя*файла*>, —*file=имя*файла — активация направления вывода в указанный файл.

-g, —globals-only — выгрузка глобальных объектов без баз данных.

-o, —oids — выгрузка идентификаторов объектов (OIDs) вместе с данными таблиц.

-O, —no-owner — деактивация генерации команд, устанавливающих принадлежность объектов, как в исходной базе данных.

-r, —roles-only — выгрузка только ролей без баз данных и табличных пространств.

-s, —schema-only — выгрузка только схемы без самих данных.

-S <имя*пользователя*>, —*superuser=имя*пользователя — привилегированный пользователь, используемый для отключения триггеров.

-t, —tablespaces-only — выгрузка табличных пространства без баз данных и ролей.

-v, —verbose — режим подробного логирования.

-V (—version — вывод версии утилиты pg_dumpall.

19.3.2. Утилита pg_restore

Утилита позволяет восстанавливать данные из резервных копий. Например, чтобы восстановить только определенную БД (в нашем примере zabbix), нужно запустить эту утилиту с параметром *-d*:

```
# pg_restore -d zabbix /tmp/zabbix.bak
```

Чтобы этой же утилитой восстановить определенную таблицу, нужно использовать ее с параметром *-t*:

```
# pg_restore -a -t history /tmp/zabbix.bak
```

Также утилитой pg_restore можно восстановить данные из бинарного или архивного файла. Соответственно:

```
# pg_restore -Fc zabbix.bak
# pg_restore -Ft zabbix.tar
```

При восстановлении можно одновременно создать новую базу:

```
# pg_restore -Ft -C zabbix.tar
```

Восстановить данные из дампа также возможно при помощи *psql*:

```
# psql zabbix < /tmp/zabbix.dump
```

Если для подключения нужно авторизоваться, вводим следующую команду:

```
# psql -U zabbix -W zabbix < /tmp/zabbix.dump
```

Ниже приведен синтаксис утилиты *pg_restore*.

-h <сервер>, —host=сервер — имя сервера, на котором работает база данных.

-p <порт>, —port=порт — TCP-порт, через база данных принимает подключения.

-U <пользователь>, —username=пользователь — имя пользователя для подключения..

-w, —no-password — деактивация требования ввода пароля.

-W, —password — активация требования ввода пароля.

—role=имя роли — роль, от имени которой выполняется восстановление резервная копия.

<имя_файла> — расположение восстанавливаемых данных.

-a, —data-only — восстановление данных без схемы.

-c, —clean — добавление операторов DROP перед операторами CREATE.

-C, —create — создание базы данных перед запуском процесса восстановления.

-d <имя*бд*>, —*dbname*=*имя*бд — имя целевой базы данных.

-e, —exit-on-error — завершение работы в случае возникновения ошибки при выполнении SQL-команд.

-f <имя*файла*>, —*file*=*имя*файла — файл для вывода сгенерированного скрипта.

-F <формат>, —format=формат — формат резервной копии. Допустимые форматы:

- p, plain — формирует текстовый SQL-скрипт;
- c, custom — формирует резервную копию в архивном формате;
- d, directory — формирует копию в directory-формате;
- t, tar — формирует копию в формате tar.

-I <индекс>, —index=индекс — восстановление только заданного индекса.

-j <число-заданий>, —jobs=число-заданий — запуск самых длительных операций в нескольких параллельных потоках.

-l, —list) — активация вывода содержимого архива.

-L <файл-список>, —use-list=файл-список — восстановление из архива элементов, перечисленных в файле-списке в соответствующем порядке.

-n <пространство_имен>, —schema=схема — восстановление объектов в указанной схеме.

-O, —no-owner — деактивация генерации команд, устанавливающих владение объектами по образцу исходной базы данных.

-P <имя-функции(тип-аргумента[, ...])>, —function=имя-функции(тип-аргумента[, ...]) — восстановление только указанной функции.

-s, —schema-only — восстановление только схемы без самих данных.

-S <пользователь>, —superuser=пользователь — учетная запись привилегированного пользователя, используемая для отключения триггеров.

-t <таблица>, —table=таблица — восстановление определенной таблицы.

-T <триггер>, —trigger=триггер — восстановление конкретного триггера.

-v, —verbose — режим подробного логирования.

-V, —version — вывод версии утилиты pg_restore.

19.3.3. Утилита pg_basebackup

Утилитой *pg_basebackup* можно выполнять резервное копирования работающего кластера баз данных PostgreSQL. Результирующий бинарный файл можно использовать для репликации или восстановления на определенный момент в прошлом. Утилита создает резервную копию всего экземпляра базы данных и не дает возможности создавать слепки данных отдельных сущностей. Подключение pg_basebackup к PostgreSQL выполняется при помощи протокола репликации с полномочиями суперпользователя или с правом REPLICATION.

Для выполнения резервного копирования локальной базы данных достаточно передать утилите *pg_basebackup* параметр *-D*, обозначающий директорию, в которой будет сохранена резервная копия:

```
# pg_basebackup -D /tmp
```

Чтобы создать сжатые файлы из табличных пространств, добавим параметры *-Ft* и *-z*:

```
# pg_basebackup -D /tmp -Ft -z
```

То же самое, но со сжатием bzip2 и для экземпляра базы с общим табличным пространством:

```
# pg_basebackup -D /tmp -Ft | bzip2 > backup.tar.bz2
```

Ниже приведен синтаксис утилиты *pg_basebackup.*

-d <строка*подключения*>, —*dbname*=*строка*подключения — определение базы данных в виде строки для подключения.

-h <сервер>, —host=сервер — имя сервера с базой данных.

-p <порт>, —port=порт — TCP-порт, через база данных принимает подключения.

-s <интервал>, —status-interval=интервал — количество секунд между отправками статусных пакетов.

-U <пользователь>, —username=пользователь — установка имени пользователя для подключения.

-w, —no-password — отключение запроса на ввод пароля.

-W, —password — принудительный запрос пароля.

-V, —version — вывод версии утилиты *pg_basebackup*.

-?, —help — вывод справки по утилите pg_basebackup.

-D каталог, —pgdata=каталог — директория записи данных.

-F <формат>, —format=формат — формат вывода. Допустимые варианты:

- **p, plain** — значение для записи выводимых данных в текстовые файлы;

- **t, tar** — значение, указывающее на необходимость записи в целевую директорию в формате tar.

-r <скорость*передачи*>, —*max-rate*=*скорость*передачи — предельная скорость передачи данных в Кб/с.

-R, —write-recovery-conf — записать минимальный файл *recovery.conf* в директорию вывода.

-S <имя*слота*>, —*slot*=*имя*слота — задание слота репликации при использовании WAL в режиме потоковой передачи.

-T <каталог*1*=*каталог*2>, —tablespace-mapping=каталог*1*=*каталог*2 — активация миграции табличного пространства из одного каталога в другой каталог при копировании.

—xlogdir=каталог_xlog — директория хранения журналов транзакций.

-X <метод>, —xlog-method=метод — активация вывода файлов журналов транзакций WAL в резервную копию на основе следующих методов:

- **f, fetch** — включение режима сбора файлов журналов транзакций при окончании процесса копирования;
- **s, stream** — включение передачи журнала транзакций в процессе создания резервной копии.

-z, —gzip — активация gzip-сжатия результирующего tar-файла.

-Z <уровень>, —compress=уровень — определение уровня сжатия механизмом gzip.

-c , —checkpoint=fast|spread — активация режима реперных точек.

-l <метка>, —label=метка — установка метки резервной копии.

-P, —progress — активация в вывод отчета о прогрессе.

-v, —verbose — режим подробного логирования.

20. Настройка сессий пользователя

Перейдите в административный раздел управления сервисом авторизации

https://auth.domain.ltd

http://ip:8180

Там перейдите в раздел - Настройки Realm / Токены

Есть две настройки:

1. Таймаут сессии SSO
   По умолчанию 30 минут

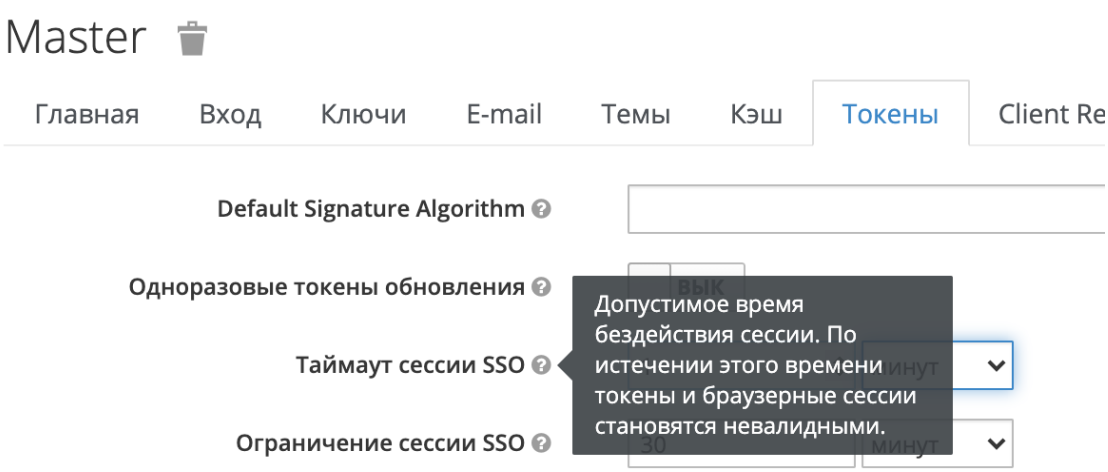

Рисунок 90

2. Ограничение сессии SSO
   По умолчанию 10 часов

21. Миграция индексов базы Elasticsearch

На рисунке 91 представлена схема миграции индексов базы Elasticsearch.

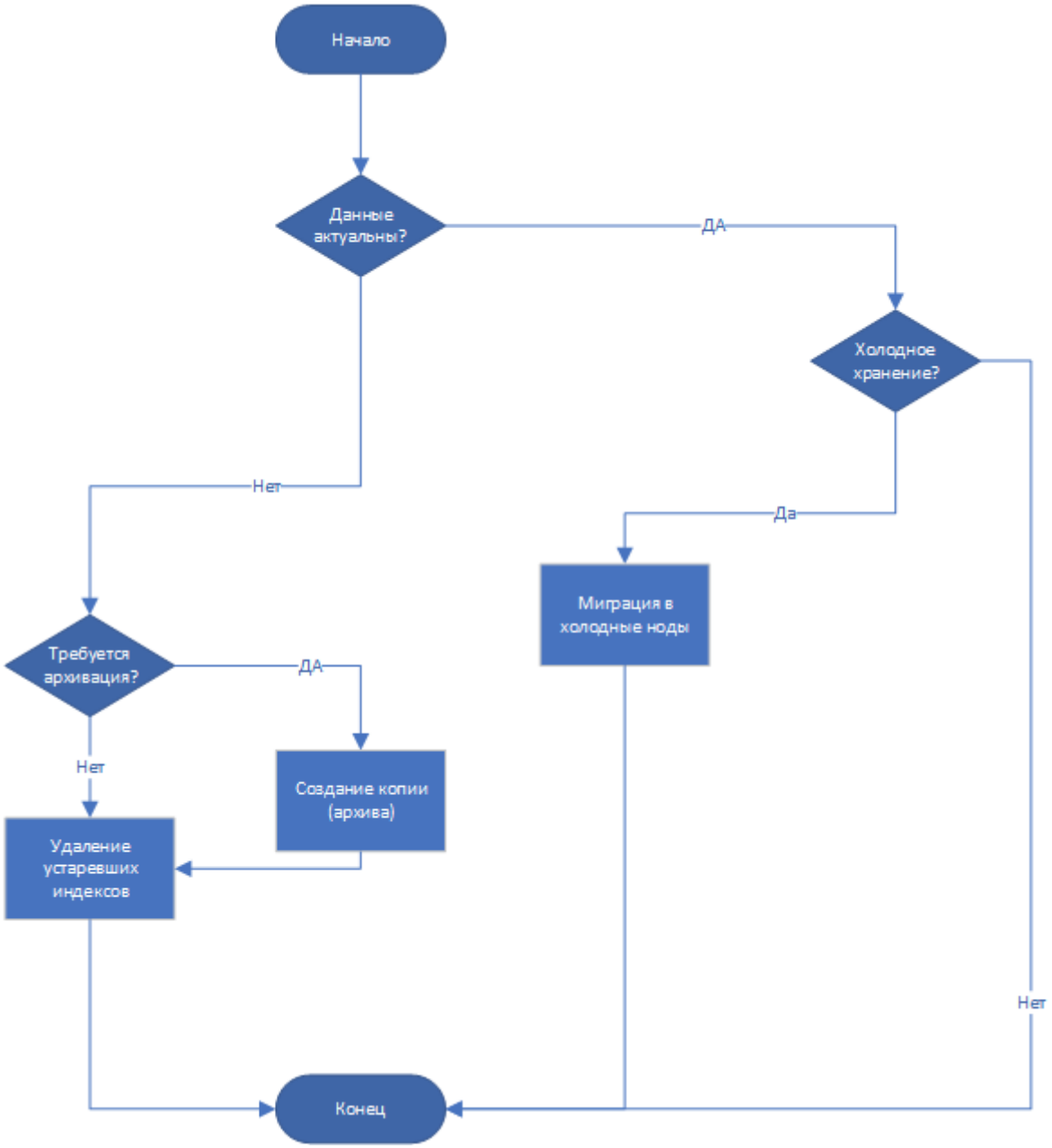

Рисунок 91 - "Схема миграции"

21.1. Настройка миграции

Настройка миграции производится на одной из нод кластера Elasticsearch.

Для работы скрипта миграции необходимо установить программный компонент `elasticdump`, для этого нужно выполнить команду: `bash /opt/pangeoradar/support_tools/elasticdump/install.sh`

После установки нужно произвести настройку ноды, выполнив команду: `bash /opt/pangeoradar/support_tools/elastic/es_config.sh`

На рисунке 92 изображен этап включения дополнительных параметров, рекомендуется использовать все.

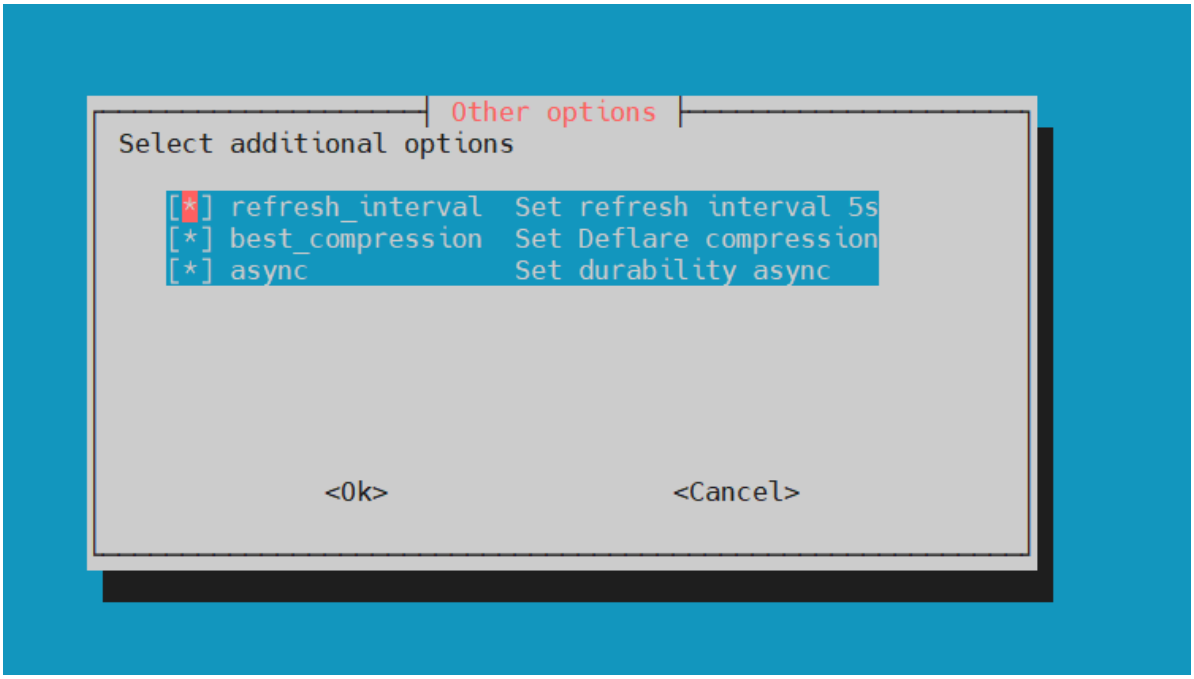

Рисунок 92 - "Настройка дополнительных параметров"

Далее необходимо произвести настройку самой миграции в конфигурационном файле: `/opt/pangeoradar/support_tools/elastic/indices_route.sh`

Основные настройки и их описания представлены ниже:

```
SNAPSHOT_DIRECTORY="/data/archive"  # путь сохранения архивированных индексов

es_proto="https"                    # протокол по которому осуществляется
подключение к базе
es_host="127.0.0.1"                 # IP адрес сервера ES
es_port=9200                        # порт подключения к ES
hot_cold=1                          # включить\отключить функционал миграции в
"холодное" хранилище
archive=1                           # включить\отключить создание архива
индексов
archive_error=1                     # включить\отключить архивацию индексов
ошибок
cold_day=1                          # количество дней, после которых индексы
перемещаются в "холодное" хранилище
delete_error_day=1                  # количество дней, после которых индексы
ошибок удаляются
delete_day=2                        # количество дней, после которых индексы
удаляются
archive_day=90                      # количество дней хранения архива индексов,
после которых они будут удалены
```

После чего необходимо добавить задачу на ежедневное выполнение. Для этого нужно выполнить команду: `crontab -e`

Удалить из планировщика задач строку:

```
0 4 * * * /usr/bin/curator --config /etc/curator/config.yml
/etc/curator/action.yml
```

Добавить следующую строку:

```
0 4 * * * /bin/bash /opt/pangeoradar/support_tools/elastic/indices_route.sh
```

После чего настройку миграции можно считать завершенной.

21.2. Восстановление индексов из архива

Для восстановления индексов из архива необходимо выполнить команду: `bash /opt/pangeoradar/support_tools/elastic/restore.sh`

После выполнения, должно появиться окно с фильтрацией индексов, изображенное на рисунке 93.

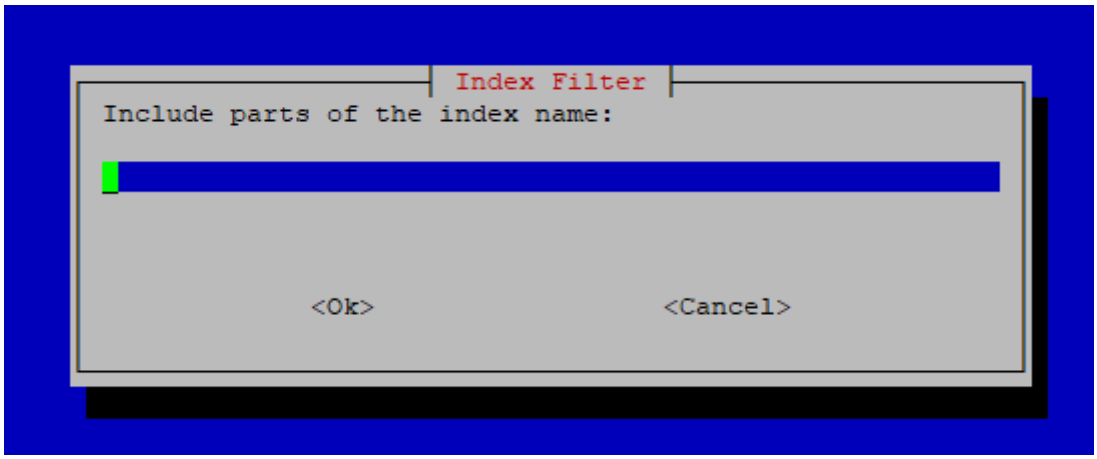

Рисунок 93 - "Фильтрация архивированных индексов"

В качестве аргументов можно использовать часть имени индекса. Примеры:

- 'firewall' - все архивные индексы межсетевого экрана;
- '2021.11.23' - все архивные индексы за 23 ноября 2021 года;
- '2021.12' - все архивные индексы за декабрь 2021 года.

> Если не указывать аргументы - будут отображены все архивированные индексы.

После чего, необходимо выбрать индексы, которые необходимо разархивировать, как изображено на рисунке 94.

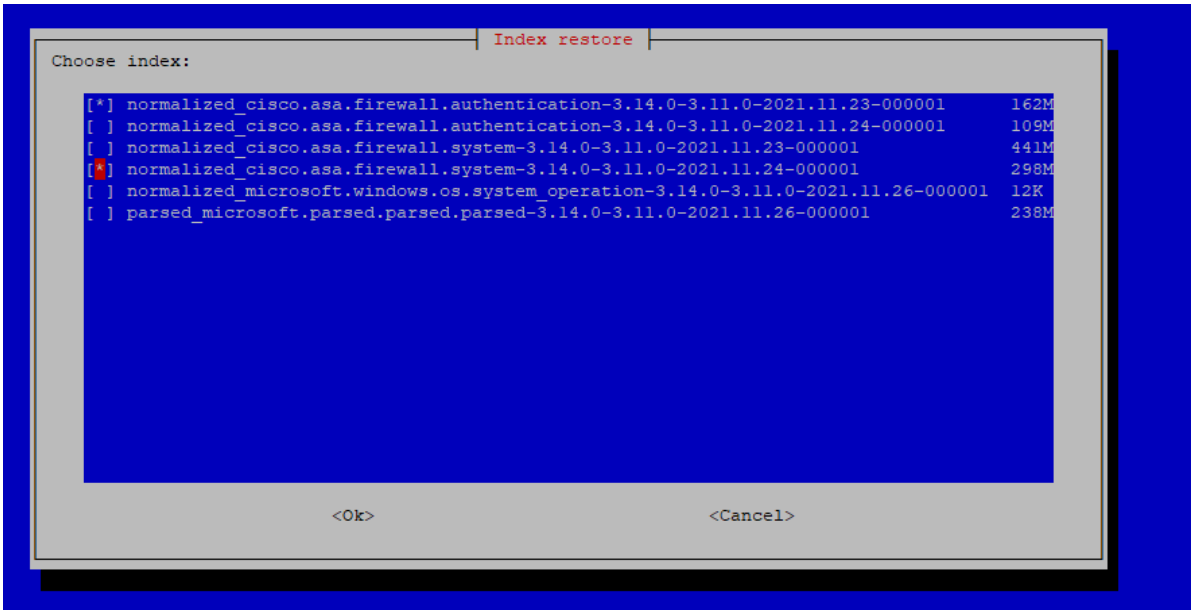

Рисунок 94 - "Выбор индексов для разархивации"

> Возможно восстановление нескольких индексов

Процесс разархивации индексов фиксируется в журнале: `/var/log/restore.log`

22. Исходные ("сырые") события

22.1. Включение\выключение исходных ("сырых") событий

Общий алгоритм для включения\выключения исходных ("сырых") событий.

1. Подключитесь по SSH к узлу обработки событий Платформы (Worker).
2. Выберите какие исходные ("сырые") события вы хотите включить: для всех источников или для определенного источника.
3. Перейдите в конфигурационный файл (подробнее в следующих подразделах) и внесите изменения.
4. Сохраните конфигурационный файл и перезапустите сервис.

22.1.1. Для всех источников

Включение\выключение исходных ("сырых") событий **для всех источников** осуществляется в файле `/opt/pangeoradar/configs/termite/conf.yaml`

```
output:
  no_raw: false
```

> Перевод данного параметра в состояние 'false' приведет к добавлению "сырой" части событий для всех входящих событий.

22.1.2. Для определенного источника

Включение\выключение исходных ("сырых") событий **для определенного источника** осуществляется в файле `/opt/pangeoradar/configs/termite/inputs-kafka.yaml`

Путем добавления строки `no_raw: true` в блоке с источником, для которого требуется включение\выключение "сырой" части события.

Пример изменения в конфигурационном файле для источника **Microsoft-Windows-Eventlog**:

```
1514-Microsoft-Windows-Eventlog:
  input: kafka
  encoding: utf-8
  kafka-config:
    enable.ssl.certificate.verification: false
    security.protocol: SSL
    ssl.ca.location: /opt/pangeoradar/certs/pgr.crt
  message-type: microsoft_windows
  servers: ['<IP-адрес-Kafka>:9992']
  topics: ['1514-Microsoft-Windows-Eventlog']
  timezone: Europe/Moscow
  no_raw: true
```

> **Важно!** При проведении "Синхронизации" источников, включение\выключение "сырой" части события для определенного источника необходимо производить повторно.

22.2. Просмотр сохраненных исходных ("сырых") событий

Для просмотра сохраненных исходных (сырых) событий необходимо выполнить следующие действия:

1. В веб-интерфейс Платформы зайдите в раздел **"Просмотр событий"**.
2. Задайте временной интервал в поле **Время**.
3. При необходимости введите или выберите в раскрывающемся списке нужный индекс в поле **Индекс**.
4. Обновите данные на экране согласно заданным параметрам, нажав .
5. В левой части экрана в области "**Доступные поля**" найдите поле **raw** и нажмите .

Поле **raw** добавится в область "**Выбранные поля**". В правой части экрана под графиком отобразятся сырые события в формате JSON (см. Рисунок 95).

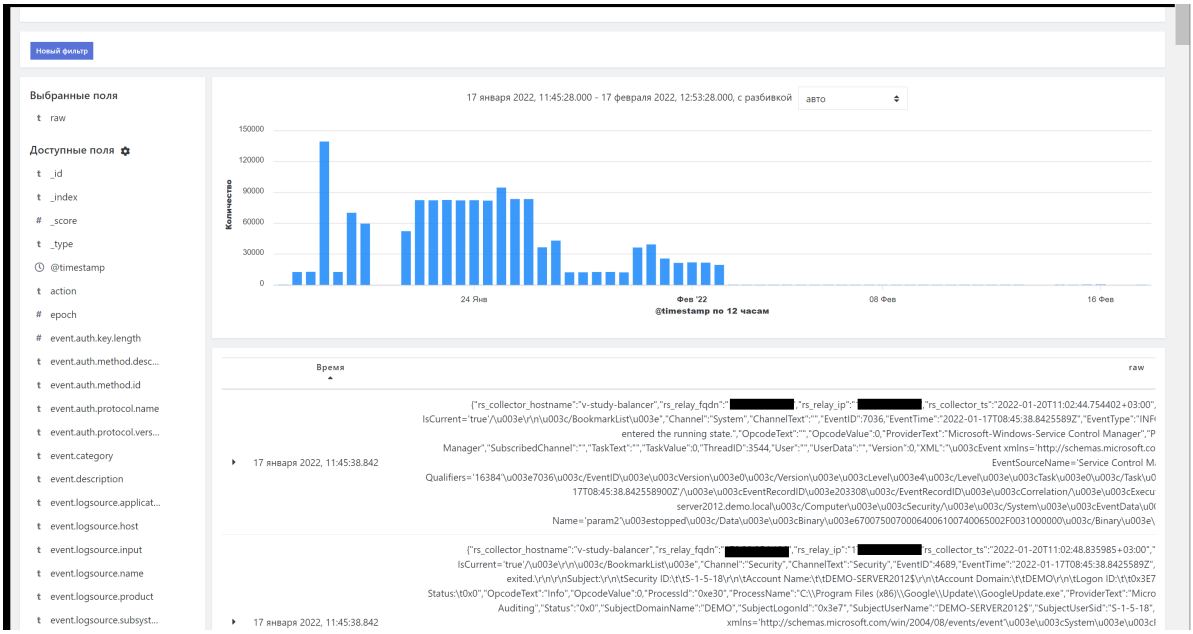

Рисунок 95 - Отображение исходных (сырых) событий в разделе "Просмотр событий"

23. Корректировка времени источника

Корректировка времени источника осуществляется в файле `/opt/pangeoradar/configs/termite/time-fix.yaml`

```
#test-type:
#  hosts:
#    - address: 1.1.1.1
#      lookup: 'initiator.host.ip'
#      timedelta: 1000
#    - address: 2.2.2.2
#      lookup: 'initiator.host.ip'
#      timedelta: -1000
```

в поле `timedelta` укажите число секунд сдвига времени, значение может быть как положительным, так и отрицательным.

в поле `lookup` необходимо указать поле нормализации из которого будет происходить сравнение с полем `address`

> Также при подключении источника возможно указать его таймзону, что приведет к автоматической конвертации её в указанную из нормализованного поля `@timestamp`.

24. Настройка режима мультиарендности

Система может обеспечивать передачу данных между несколькими экземплярами Платформы.

Платформа Радар обеспечивает возможность работы системы в режиме мультиарендности.

24.1. Настройка режима мультиарендности

Для организации режима мультиарендности необходимо выполнить подключение удаленного экземпляра Платформы:

1. Подключитесь по SSH к терминалу сервера с ролью **Master**.
2. Выполните команду для подключения удаленного экземпляра:
   `/opt/pangeoradar/bin/pangeoradar-karaken create-instance --name=second --url=<https://........ .3:9000> --order=0 --conf=/opt/pangeoradar/configs/`
3. Внесите изменения в следующие конфигурационные файлы на удаленном экземпляре:

- в конфигурационный файл /opt/pangeoradar/configs/pangeoradar-cerberus.yaml:
  `KEYCLOAK_URL: <IP-адрес>`
- в конфигурационный файл /opt/pangeoradar/configs/pangeoradar-manager.yaml:
  `AuthHost: <IP-адрес>`
- в конфигурационный файл /opt/pangeoradar/configs/pangeoradar-agent.yaml:
  `AuthHost: <IP-адрес>`
- в конфигурационный файл /opt/pangeoradar/configs/pangeoradar-toller.yaml:
  `AuthHost: <IP-адрес>`

4. Перейдите в веб-интерфейс Платформы и убедитесь, что появилась возможность переключения между экземплярами (см. Рисунок 96).

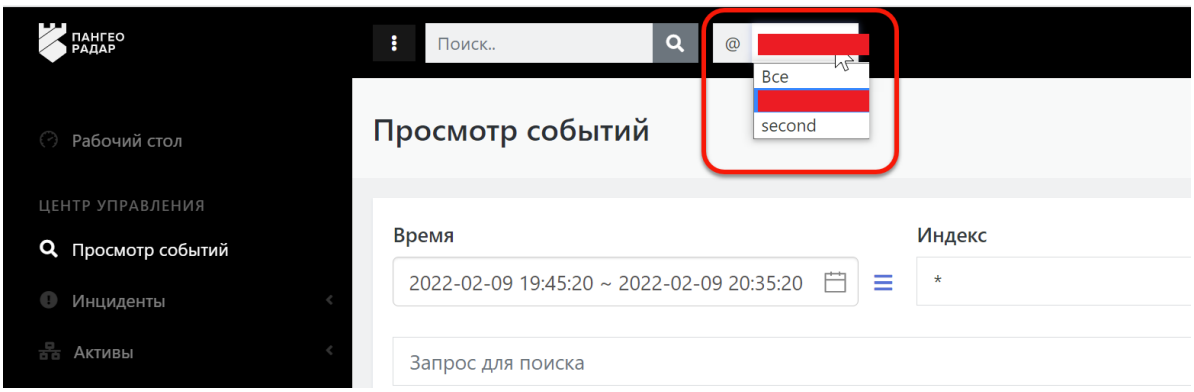

Рисунок 96 - Подключение удаленного экземпляра Платформы

Для настройки пользовательского профиля для работы в режиме мультиарендности необходимо зайти в веб-интерфейс Платформы с правами администратора и выполнить следующие действия:

1. Перейдите в раздел «**Администрирование**» — «**Пользователи и права**» — «**Пользователи**» и создать пользователя **fpc_user**, указав **Логин**, **email**, **Имя** и **Фамилию** (см. Рисунок 97).

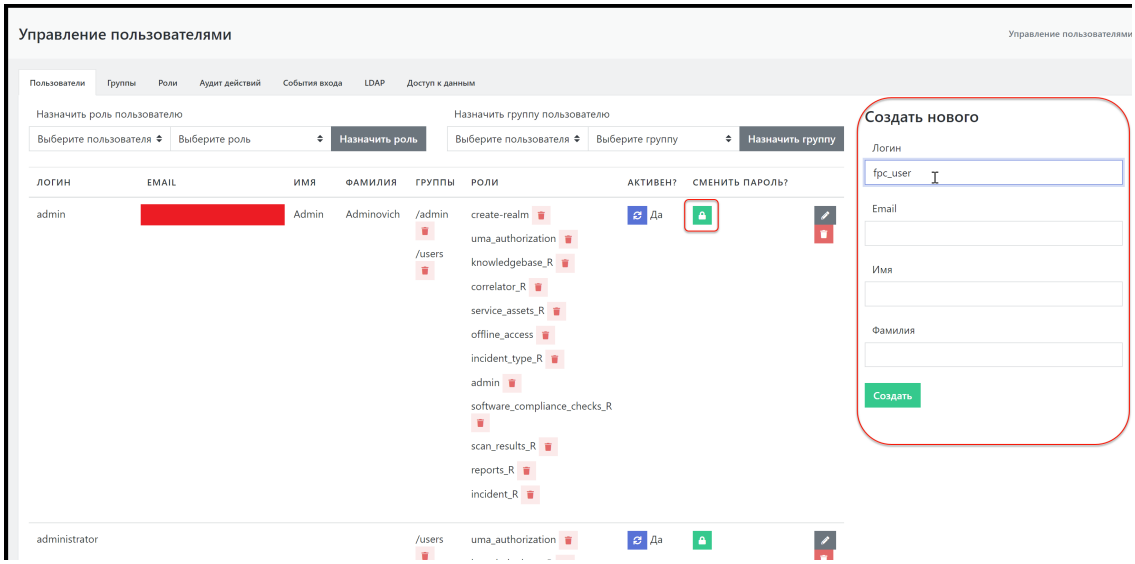

Рисунок 97 - Создание нового пользователя

2. После создания пользователя скопируйте одноразовый пароль нажав на кнопку справа от учетных данных пользователя (см. Рисунок 97).
3. Перейдите на вкладку "Доступ к данным" (см. Рисунок 98).

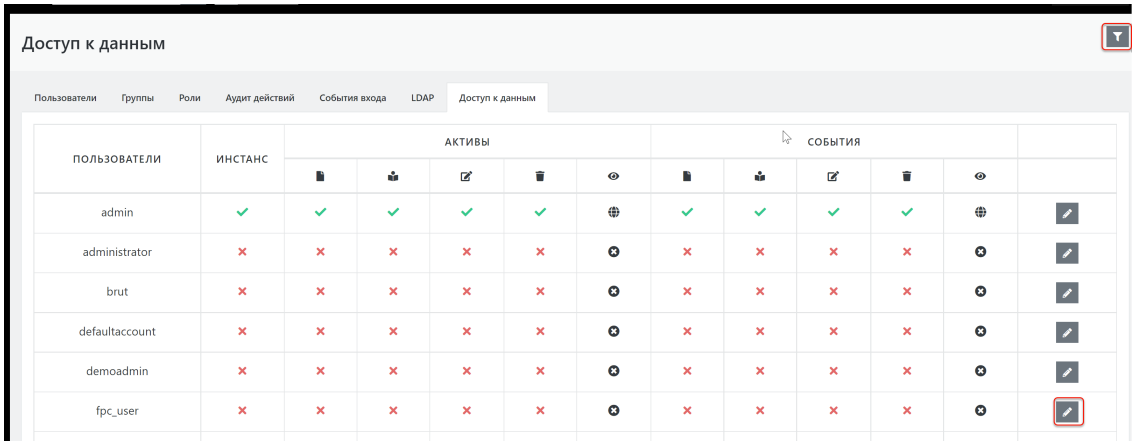

Рисунок 98 - Список пользователей платформы с параметрами доступа

4. Нажмите на пиктограмму в правом верхнем углу вкладки (см. Рисунок 98) и в раскрывающемся списке выберите подчиненный экземпляр Платформы.
5. Откройте для редактирования пользователя **fpc_user**, нажав пиктограмму в строке с учетной записью данного пользователя (см. Рисунок 98).
6. В меню редактирования пользователя: предоставьте пользователю доступ к экземпляру, дайте права на просмотр активов, проставив соответствующие галочки в чек-боксах напротив имени учетной записи и сохраните изменения, нажав на соответствующую кнопку

Сохранить (см. Рисунок 99).

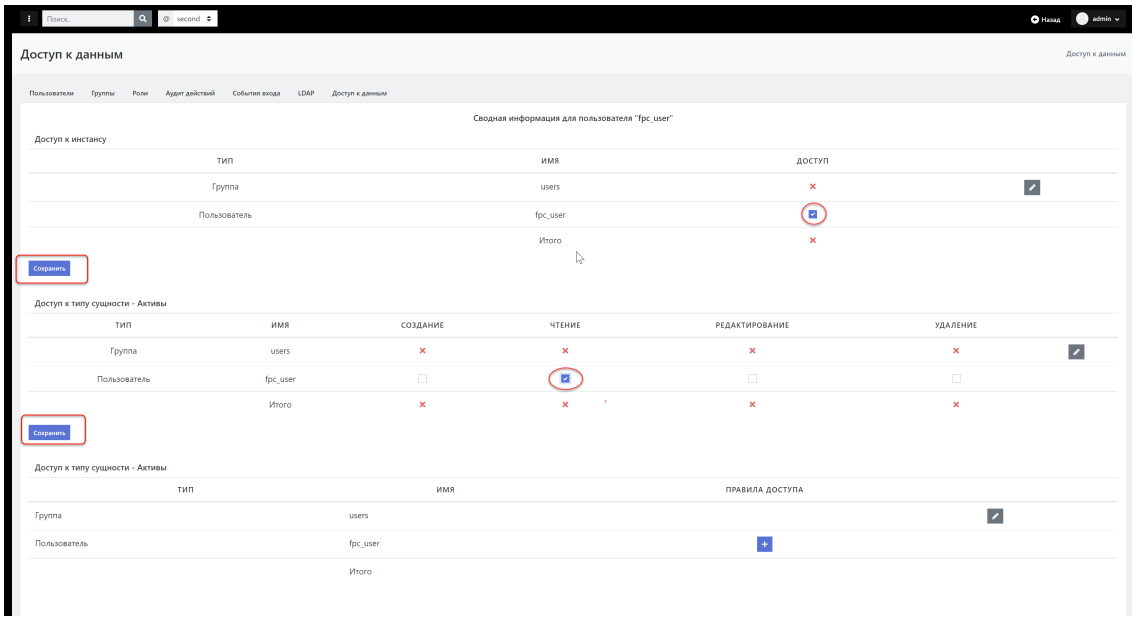

Рисунок 99 - Настройка прав доступа пользователя

При возврате на вкладку **Доступ к данным** в строке пользователя отобразятся заданные настройки доступа (см. Рисунок 100).

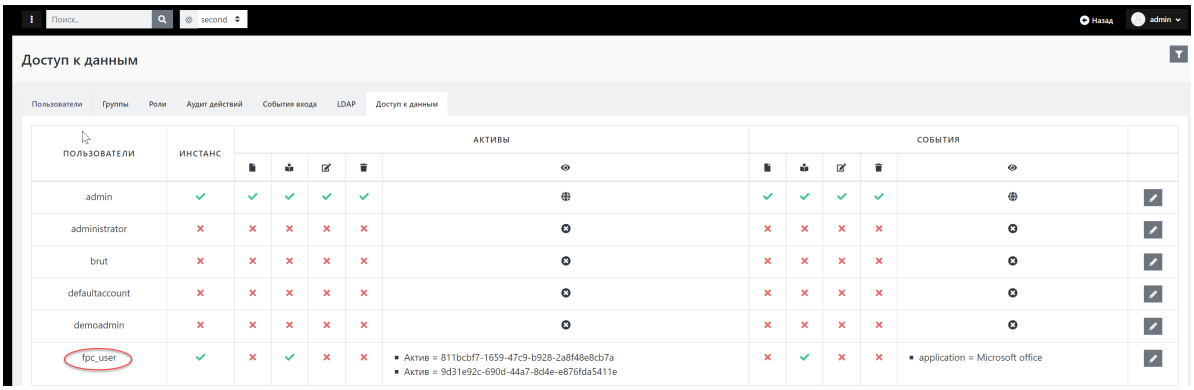

Рисунок 100 - Пример настроек доступа для пользователя

24.2. Проверка режима мультиарендности

Для проверки работы платформы в режиме мультиарендности выполните следующие действия:

1. Откройте новое окно браузера в режиме инкогнито.
2. Введите в адресной строке браузера адрес веб-интерфейса удаленного экземпляра Платформы.
3. Зайдите в Платформу под пользователем **fpc_user**, используя одноразовый пароль, скопированный при создании пользователя (см. пункт 2 из инструкции по настройке режима мультиарендности настоящего раздела). Система предложит сменить пароль.
4. Установите новый пароль и войдите в систему.
5. Перейдите в раздел «**Активы**» — «**Активы**».

В разделе активов отобразится только тот список активов, к которым данному пользователю был предоставлен доступ при настройке профиля пользователя для удаленного экземпляра Платформы. Доступные действия с активами — только чтение согласно проведенным настройкам.

25. Настройка архивации событий

25.1. Проверка текущих настроек политики архивации устаревших событий

В Платформе предусмотрена возможность архивации устаревших событий.

Для проверки текущего состояния политики архивации выполните следующие действия:

1. Зайдите по SSH на сервер архивации событий (узел платформы с ролью **DATA**).
2. Откройте конфигурационный файл /opt/pangeoradar/scripts/indices_route.sh командой: `nano/opt/pangeoradar/scripts/indices_route.sh`
3. Посмотрите значения параметров **cold_day** и **delete_day**. По умолчанию они должны иметь значение 27 (27 дней оперативного хранения).

Для проверки работы политики архивации выполните следующие действия:

1. В веб-интерфейсе Платформы перейдите в раздел «**Просмотр событий**».
2. Для формирования отчета выставите в области задания временного интервала промежуток времени длиннее чем заданный промежуток в политиках архивации, например, 30 дней считая от сегодняшнего.

На экране в статистике по событиям не должны отображаться события старше 27 дней.

25.2. Изменение политики архивации устаревших событий

Для изменения политики архивации выполните следующие действия:

1. Зайдите по SSH на сервер архивации событий (узел платформы с ролью **DATA**).
2. Откройте конфигурационный файл узла /opt/pangeoradar/scripts/indices_route.sh командой: `nano /opt/pangeoradar/scripts/indices_route.sh`
3. Установите для параметров **cold_day** и **delete_day** новое значение оперативного хранения данных.
4. Принудительно запустите архивацию, выполнив команду: `bash /usr/bin/bash /opt/pangeoradar/scripts/indices_route.sh`
5. Дождитесь окончания выполнения скрипта.

Для проверки введённых изменений выполните следующие действия:

1. В веб-интерфейсе Платформы перейдите в раздел «**Просмотр событий**».
2. Проверьте, что нет индексов старше 20 дней (см. алгоритм проверки описан в предыдущем подразделе).
3. Вернитесь в терминал сервера архивации событий (узел **DATA**).
4. Перейдите в директорию **/data/archive**.
5. Выведите листинг директории командой:

`ls -lah`

6. Убедитесь в появлении новых архивов.
7. Для просмотра запланированных заданий выполните команду:

   `crontab -l`

8. Убедитесь в наличии запланированного задания по ротированию и архивации событий.

В результате проведенных действий в веб-интерфейсе платформы должны отсутствовать записи об индексах и событиях старше заданного количества дней в политике архивации.

Должны быть созданы новые архивы с названиями индексов, экспортированных из системы для архивации и долгосрочного хранения.

25.3. Восстановление данных из архива и обращения к восстановленным событиям

В Платформе предусмотрена возможность обращения к устаревшим событиям, находящимся на архивном хранении.

Для того, чтобы получить доступ к архивным данным, необходимо сначала выполнить восстановление данных из архива:

1. Зайдите по SSH на сервер архивации событий (узел платформы с ролью **DATA**).
2. Запустите скрипт восстановления данных командой:

   `bash /opt/pangeoradar/scripts/elastic_restore.sh`

3. В появившемся окне укажите фильтр ***** и нажмите **ОК**.
4. Выберите интересующий индекс из списка, выделите напротив него чекбокс (запомните имя восстанавливаемого индекса) и нажмите **ОК**.
5. Дождитесь окончания восстановления (восстановление осуществляется в фоновом режиме).

Для просмотра восстановленных данных необходимо:

1. Перейдите в веб-интерфейс Платформы в раздел «Просмотр событий».
2. В поле **Время** укажите временной диапазон восстанавливаемого индекса (см. Рисунок 101).
3. В поле **Индекс** укажите имя восстанавливаемого индекса (см. Рисунок 101).
4. Нажмите кнопку **Поиск**.

На экран должен быть выведен список событий (включая диаграмму), относящийся к восстанавливаемому индексу и указанному временному периоду (см. Рисунок 101).

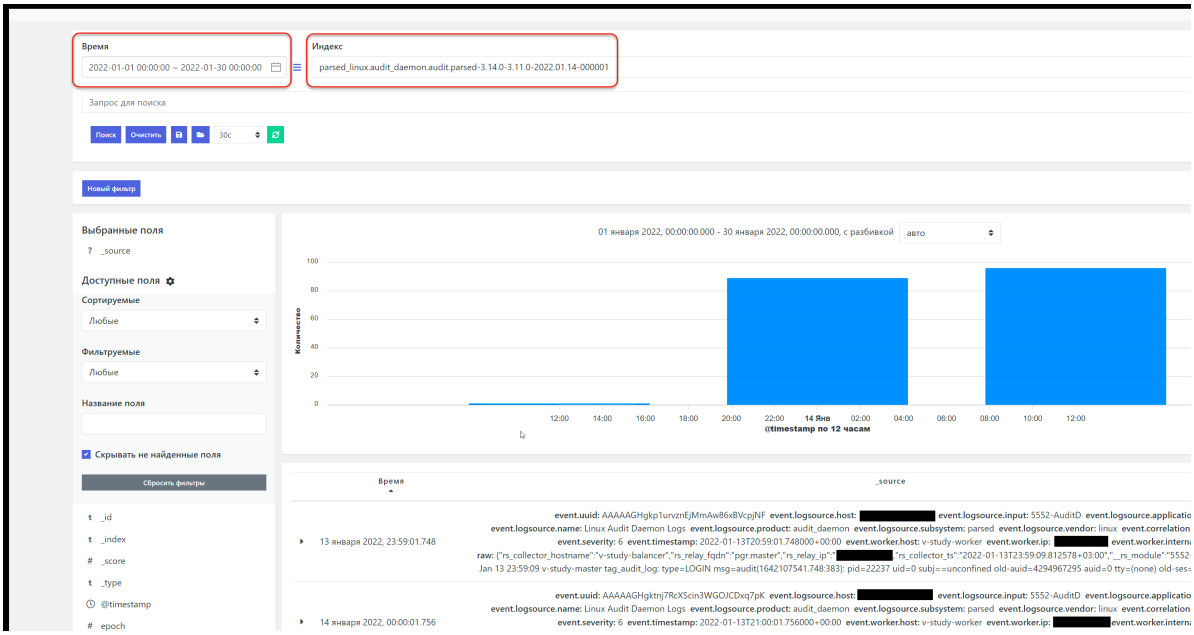

Рисунок 101 - Просмотр данных, восстановленных из архива

26. Настройка и проверка интеграции через API

В Платформе Радар реализована интеграция посредством API с IRP-системами - R-Vision и Security Vision.

26.1. Настройка и проверка передачи через API информации об инциденте во внешнюю систему

Для настройки интеграции с внешними системами через API необходимой выполнить следующие действия:

1. Подключитесь по SSH к узлу платформы с ролью **Master**.
2. Внесите следующие изменения в конфигурационный файл узла **/opt/pangeoradar/configs/pangeoradar-pgr-wal-listener.yaml**:

- Добавьте реквизиты интегрируемой системы (R-Vision) — ключ доступа к API R-Vision и IP-адрес R-Vision;
- Измените схему соответствия полей согласно требованиям интеграции.

3. Запустите сервис **pangeoradar-pgr-wal-listener**:

   `service pangeordar-pgr-wal-listener start`

Для проверки проведенного подключения выполнить следующие действия:

1. Зайдите в веб-интерфейс Платформы (с правами администратора).
2. Зайдите в раздел «**Инциденты**» — «**Инциденты**».
3. Создайте инцидент вручную, нажав кнопку **Создать инцидент**.

Подробное описание создания инцидента вручную приведено в документе «*Руководство оператора*», раздел «*Работа с инцидентами*».

При настроенном API новый инцидент передается во внешнюю систему в автоматическом режиме в процессе создания. Созданный инцидент автоматически создан в IRP.

26.2. Генерация ключа для доступа к API. Использование ключа

Для работы по API необходимо сгенерировать ключ для доступа к API. Для этого выполните следующие действия:

1. Перейдите в веб-интерфейс Платформы в подраздел "Кластер"->"API ключи" (см. Рисунок 102).

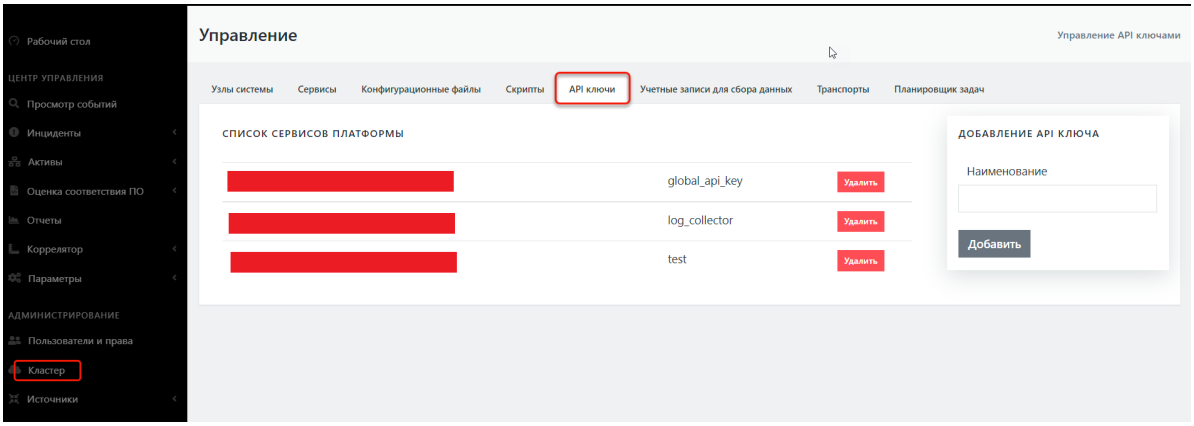

Рисунок 102 - Настройка ключей для работы через API

2. Добавьте ключ, введя значение параметра в поле **Наименование** ключа (например, integration) и нажав кнопку **Добавить**.
3. Подключитесь по SSH к узлу платформы с ролью **Master**.
4. Выполните с использованием ключа **integration**, который был сгенерирован на этапе предварительных действий в данной проверке, следующую команду:

```
curl -k -H "PgrApiKey:<ключ, сгенерированный на шаге 2 >"
"https://10.170.9.22:9000/cruddy/public/api/v1/incidents?
page=1&per_page=1&order=id%20DESC"
```

На экран будут выведена запись по одному инциденту в формате JSON.

27. Настройка политики противодействия попыткам подбора пароля

Платформа обладает встроенными механизмами противодействия попыткам подбора пароля (BruteForce атаки) на базе открытого ПО **Keycloak** (идентификационный брокер).

Для настройки политики противодействия попыткам подбора пароля выполните следующие действия:

1. С правами администратора войдите в специализированный веб-интерфейс **Keycloak** платформы `https://<адрес платформы>:8180` (см. Рисунок 103).

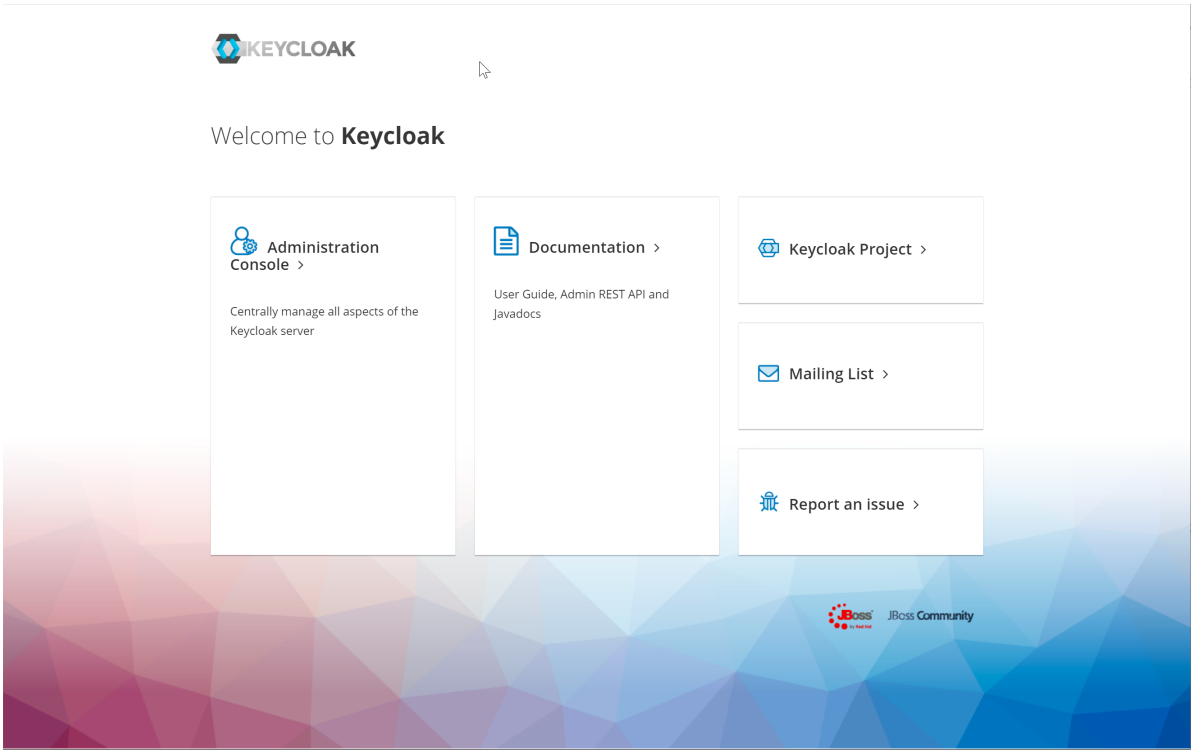

Рисунок 103 - Интерфейс "идентификационного брокера" **Keycloak**

2. Перейдите в раздел "Administration Console" -> "Защита безопасности" -> "Определение Brute Force" (см. Рисунок 104).

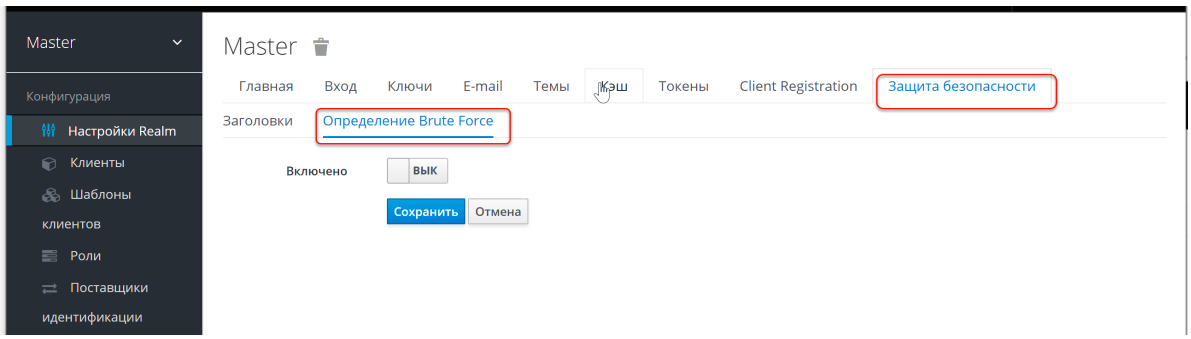

Рисунок 104 - Раздел "Определение Brute Force" при отключенных политиках.

3. Включите политику Определение Brute Force, установив переключатель в поле **Включено** в положение **вкл**. Откроются параметры настройки политики (см. Рисунок 105).

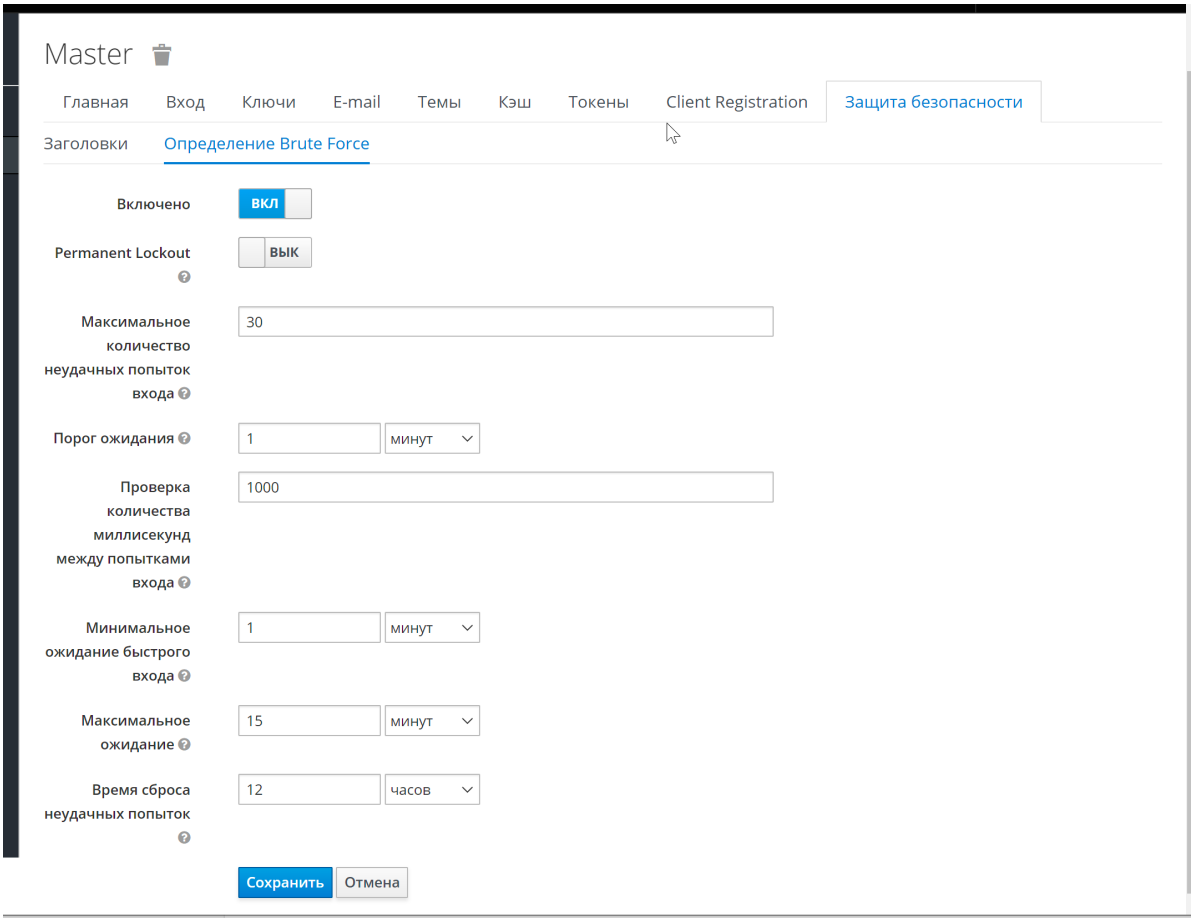

Рисунок 105 - Параметры настройки политики

4. При необходимости установите следующие параметры:
    - максимальное количество неудачных попыток входа (основная настройка) — количество неудачных попыток входа до блокировки пользователя;
    - порог ожидания (основная настройка) — если порог ошибок превышен, сколько времени пользователь будет заблокирован;
    - проверка количества миллисекунд между попытками входа — если попытки аутентификации происходят слишком часто, то пользователя необходимо заблокировать;
    - минимальное ожидание быстрого входа — как долго ждать после неудачной попытки быстрого входа;
    - максимальное ожидание — максимальное время, на которое пользователь будет заблокирован;
    - время сброса неудачных попыток — через какое время счетчик неудачных попыток будет сброшен.
5. Сохраните настройки, нажав кнопку **Сохранить**.

28. Процедура обновления

Процедура обновления отличается у разных версий Платформы, а также зависит от конкретной архитектуры при распределённой установке. Инструкции по обновлению входят в комплект пакетов обновления.

Обновление Платформы не приводит к потере накопленной информации из баз данных. При обновлении сохраняются собранные события, инциденты, база активов и база знаний со всеми пользовательскими изменениями.

Пакеты обновлений могут быть доставлены на серверы Платформы как на съёмных носителях информации (оптические диски, флеш-карты, переносные HDD/SSD накопители), так и с помощью сетевого хранилища при наличии сетевого доступа с серверов Платформы.

Обновления базы знаний с пополнением правил корреляции, правил разбора и нормализации без обновления основных пакетов Платформы могут быть предоставлены отдельно по запросу Заказчика.

29. Проведение централизованного обновления конфигурации и перезапуска сервисов компонентов Платформы

Рассмотрим проведение централизованного (через веб-интерфейс Платформы) обновления конфигурации и перезапуска сервисов компонентов платформы на примере сервиса **rsyslog**, функционирующего в составе узла Платформы с ролью **Balancer** (обновление конфигурации и перезапуск других сервисов Платформы проводится аналогичным образом).

Для проведения централизованного обновления конфигурации и перезапуска сервиса **rsyslog** выполните следующие действия:

1. Зайдите в веб-интерфейс Платформы (с правами администратора).
2. Зайдите в раздел "Кластер"->"Узлы системы"->"Карта кластера" (или "Узлы").
3. Выберите узел с ролью **Balancer** и зафиксировать его IP-адрес (см. Рисунок 106).

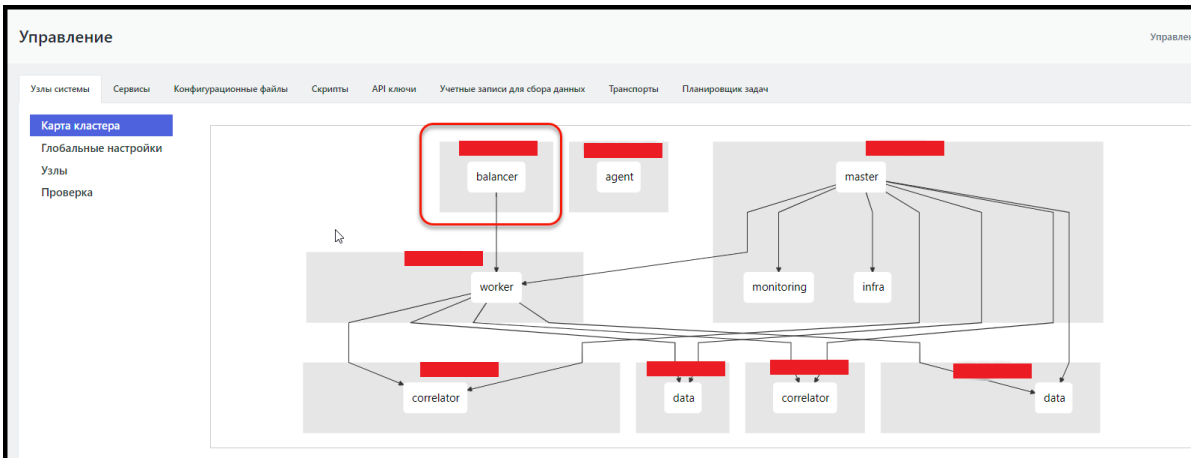

Рисунок 106 - Фиксация IP-адреса узла с ролью Balancer через "Карту кластера"

4. Перейдите в подраздел "Проверка", в списке узлов найти по IP-адресу узел **Balancer** и нажмите кнопку **Настройки** рядом с адресом узла (см. Рисунок 107).

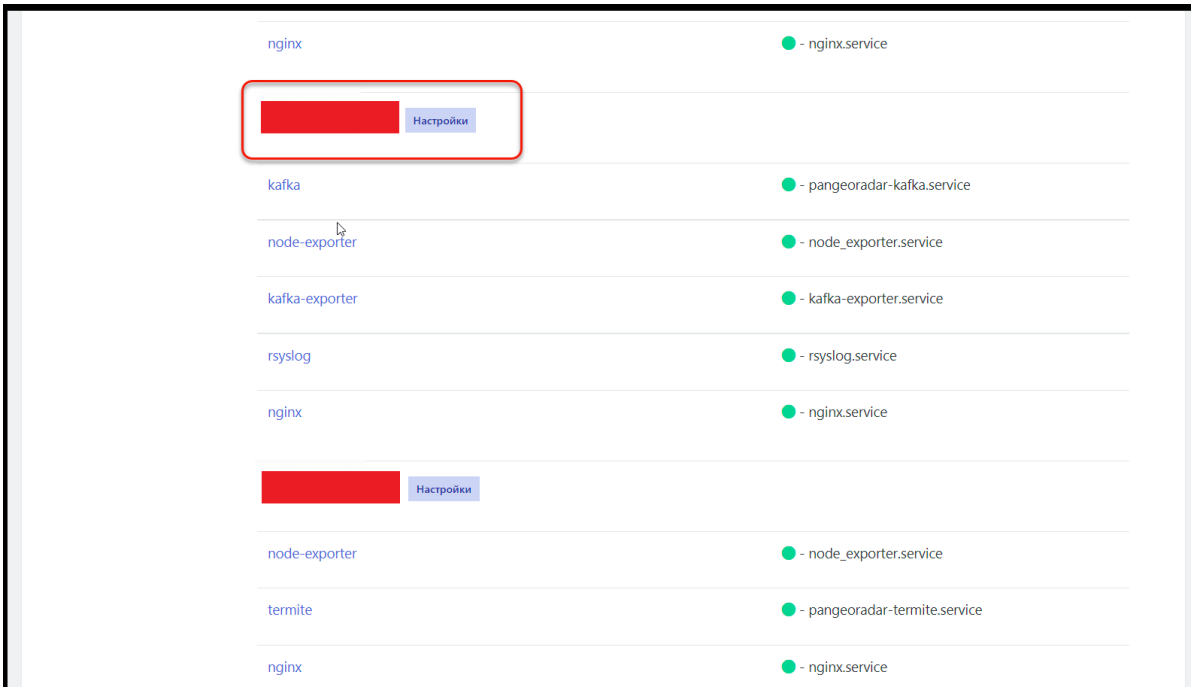

Рисунок 107 - Функция настройки узла Balancer

5. На открывшейся странице настроек узла выберите сервис **rsyslog** и проверьте его текущее состояние — посмотрите статус (см. Рисунок 108) и журнал логов сервиса. Сервис должен находится в статусе **Active**.

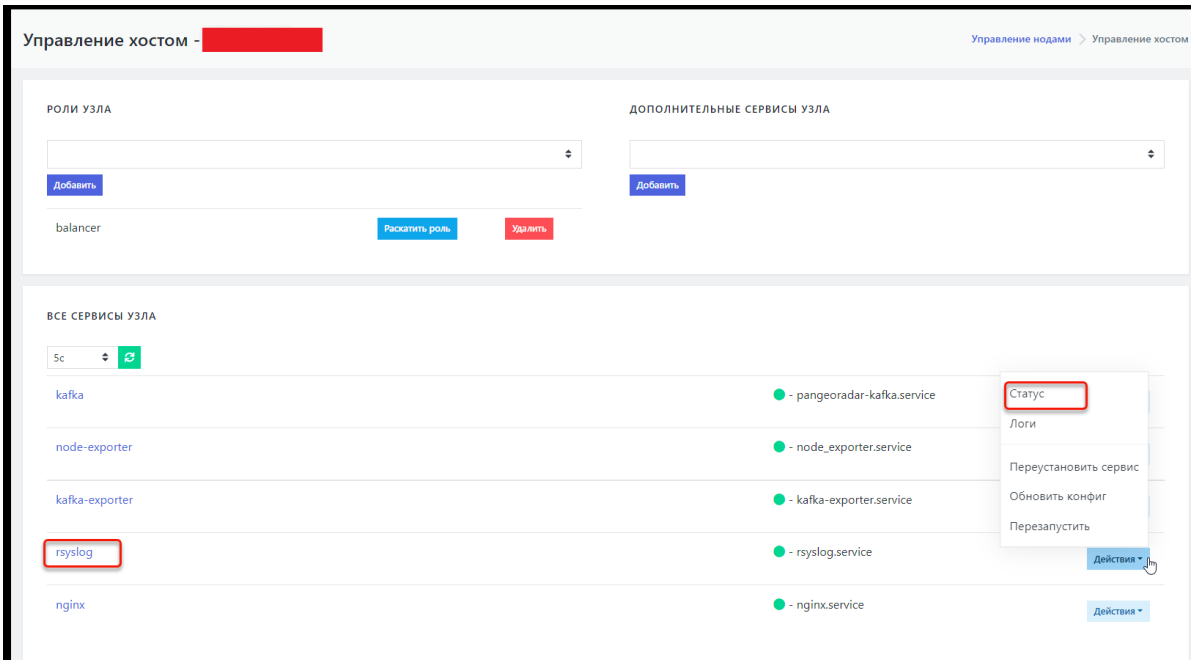

Рисунок 108 - Проверка статуса сервиса rsyslog

6. Перейдите на вкладку "Кластер"->"Конфигурационные файлы".
7. В списке конфигурационных файлов найдите для сервиса **rsyslog** конфигурационный файл **rsyslog-kafka.conf** и щелкните по названию файла. Откроется текст конфигурационного файла (см. Рисунок 109).

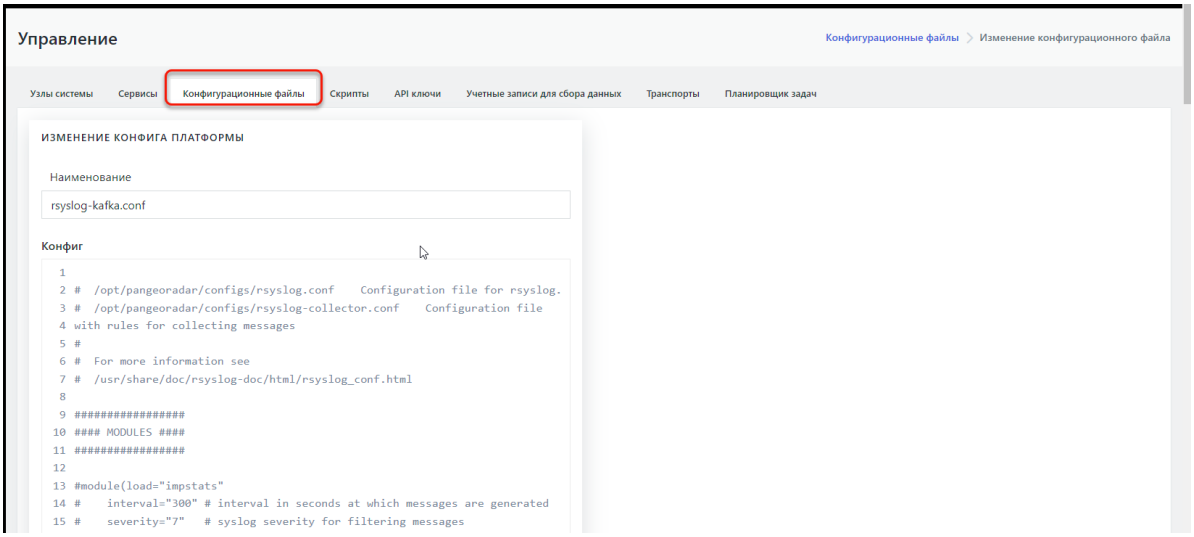

Рисунок 109 - Открытие текста конфигурационного файла для редактирования

8. Внесите в конфигурационный файл необходимые изменения. Например, в конце конфигурационного файла добавьте запись:
   `auth,authpriv.* @10.170.9.21:2671`
9. Нажмите кнопку **Изменить.**
10. Вернитесь в подраздел "Кластер"->"Узлы системы"->"Проверка" и откройте настройки соответствующего узла **Balancer** (см. шаги 4 и 5 данного алгоритма).
11. Для сервиса **rsyslog** в раскрывающемся меню "Действия" выберите пункт **Обновить конфиг**. Появится модальное окно с сообщением об успешном обновлении конфигурационного файла (см. Рисунок 110).

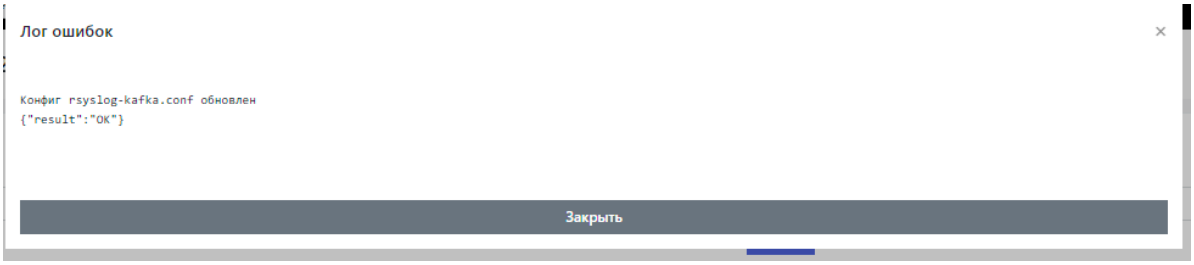

Рисунок 110 - Системное сообщение об успешном обновлении конфигурационного файла

12. Для сервиса **rsyslog** в раскрывающемся меню "Действия" выберите пункт **Перезапустить**. Дождитесь сообщения об успешном перезапуске сервиса (см. Рисунок 111).

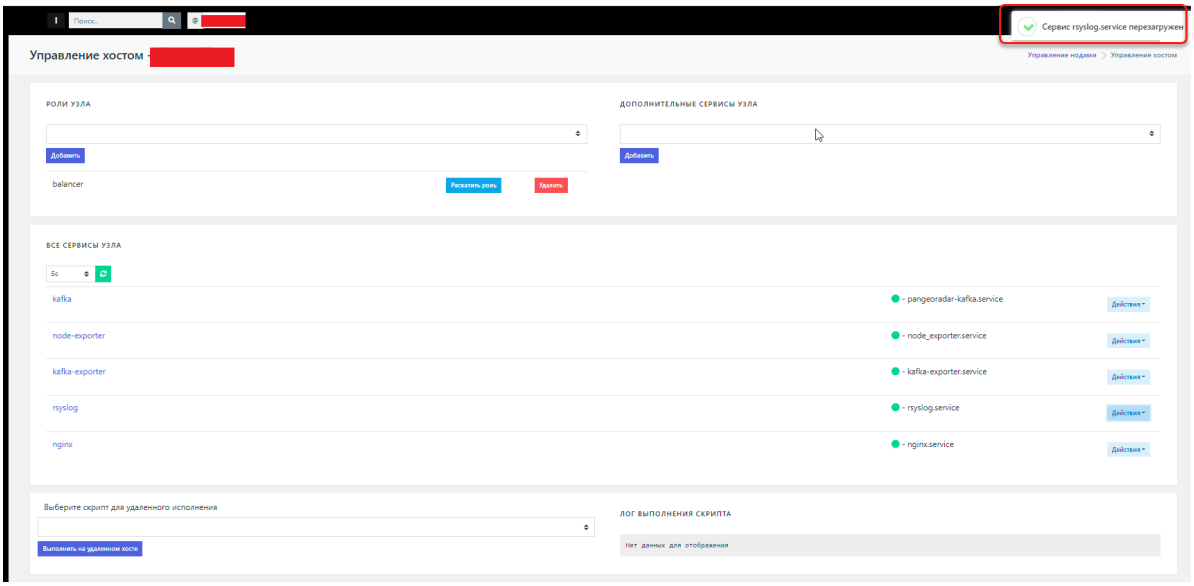

Рисунок 111 - сообщение об успешной перезагрузке сервиса

После проведения перезапуска сервиса при необходимости можно проверить статус сервиса (см. шаги 4 и 5). Также можно подключиться к узлу **Balancer** по SSH и выполнить для проверки изменений, следующую команду:
`cat /opt/pangeoradar/configs/rsyslog-kafka.conf`

---

30. Интеграционный слой

30.1. Описание

В Платформе Радар существует две механики интеграции с внешними системами.

1. Через API Платформы Радар получать необходимую информацию и интегрировать её на стороне подключаемой системы.
2. С помощью API интегрируемой системы передавать данные со стороны Платформы Радар

В первом случае необходимо воспользоваться документацией по API Платформы Радар, которую предоставляется по запросу.
Во втором же случае необходимо настроить специальный сервис и соблюсти ряд ограничений.

30.2. Ограничения

На данный момент поддерживаются только JSON REST Api с авторизацией через HTTP заголовок или GET параметр.

30.3. Описание работы сервиса интеграции

Сервис pgr-wal-listener следит за всеми изменениями в базе данных postgresql и при наступлении интересующего нас изменения, может выполнить sql запрос для формирования объекта пересылаемых данных и инициировать отправку объекта в подключаемую систему. Также предусмотрена механика формирования требуемого объекта данных для запроса в исполняемой системе из полученных данных при sql запросе.

Концептуально конфигурацию сервиса можно поделить на три секции:

1. Описание пересылаемого объекта
2. Описание вызова API методов подключаемой системы
3. Описание триггера вызова sql запроса

30.4. Конфигурационный файл

Конфигурационный файл находится по адресу - `/opt/pangeoradar/configs/pangeoradar-pgr-wal-listener.yaml`

После изменения конфигурационного файла необходимо перезапустить сервис - `service pangeoradar-pgr-wal-listener restart`

30.4.1. Описание секций конфигурационного файла

30.4.1.1. Пересылаемый объект

При формировании пересылаемого объекта мы описываем структуру json объекта следующим образом:

```
Наименование мэппинга:
  Наименование json поля объекта:
    type: тип значения поля объекта (manual, map, active_map, map_form_json)
    value: строковое значение присвоиваемое значению поля объекта в случае
выбора типа manual, либо указание поле в sql объекте для осуществления мэпинга в
случае выбора типа map, active_map, map_form_json
    from: откуда брать список для приведения соответсвия в случае выбранного
типа map_form_json
    map: # указывается при выборе типа active_map
      ключ: значение
```

Для типа map_for_json поддерживаемые специальные обработчики from

- varchar_array
- inet

30.4.1.2. API методы

При написании API метода для вызова мы описываем структуру следующим образом:

```
Наименование метода:
  type: http
  url: url api метода
  method: http метод POST|GET|PUT|DELETE
  content_type: "application/json"
  headers: # заполняется, если требуется передать дополнительные заголовки
(например авторизацию)
    ExampleHeader: "test"
  prepend: # дополнительный запрос, который нужно выполнить перед основным
    - type: http
      url: url другого api метода
      method: GET
      content_type: "application/json"
      query: # описание заполнения get параметров
        ключ: значение
        filter: '[{"operator":"=","value":%d,"property":"id_siem"}]' # можно
указать с учетом наличия внутри синтаксиса sprintf
      query_vars_from_trigger: # при указании sprintf синтаксиса выше, нужно
сформировать данные для подстановки
        filter: # для какого ключа в query запросе будет применено формирование
данных для подстановки
          field: "id" # поле которое возьмется из sql запроса
          type: "float64toInt64" # конвертация типов sql данных к sprintf типу
      append_to_mapping: # в какое поле добавить результат ответа
предварительного запроса
```

```
        identifier: "data.result.0.identifier" #секция ключ: значение, для
значения поддерживается доступ к json объекту ответа через обращения с помощью
точки
  mappging: *rvision_mapping # выбор мэппинга для формирования запроса
```

Поддерживаемая конвертация типов:

- bool
- int
- int32
- int64
- string
- float64
- float64toInt64
- float32

30.4.1.3. Триггеры

При формировании триггеров вызова мы оперируем следующими параметрами объектов:

```
Наименование группы тригеров: &radar-tables-trigger
  - name: название триггера
    table: наименовае наблюдаемой таблицы
    fields: [ ] # список наблюдаемых полей в таблице
    kind: insert # тип наблюдаемого запроса (insert, update, delete)
    sql: # sql запрос для формирования объекта измененных данных, обязательно
должно быть условие выборки по идентификатору, где $1 подставляемое значение
идентификатора из измененной строки в таблице - "t.id = $1"
    sql_vars:
      id: float64toInt64 # приведение типов wal журнала к требуемым типам данным
sql объекта
    outputs: # в какие API методы требуется отправить результат формирования sql
объектрап
      - *rvision_insert
```

30.4.1.4. Соединение с СУБД

Финальным шагом является описание подключения к СУБД и инициализация группы триггеров:

```
connections:
  - database: rmca
    username: radar
    password: ****
    host: 127.0.0.1
    port: 5432
    triggers: *Наименование группы тригеров
    UseTLS: false
    SkipTLSVerify: false
    PgCert: ""
    PgKey: ""
    RootCrt: ""
```

30.4.2. Пример конфигурации

Рассмотрим пример конфигурации с системой R-Vision

```
---

global:
  force_replica_identity: true
  log_level: warning

# Mappings (Описание пересылаемого объекта)
rvision_mapping: &rvision_mapping
  token:
    type: "manual"
    value: "******"
  category:
    type: "manual"
    value: "Инцидент из Пангео Радар"
  info_source:
    type: "manual"
    value: "SIEM Пангео "
  type:
    type: "manual"
    value: "Инцидент полученный из Пангео Радар"
  id_siem:
    type: "map"
    value: "id"
  DESCRIPTION:
    type: "map"
    value: "DESCRIPTION"
  risk_impact:
    type: "map"
    value: "risk_impact"
  solution:
    type: "map"
    value: "solution"
  mitigation:
    type: "map"
    value: "mitigation"
  status_siem:
    type: "map"
    value: "status"
  STATUS:
    type: "active_map"
    value: "status"
    map:
      new: "Создан"
      risk_accepted: "Зарегистрирован"
      assigned_customer: "Назначен"
      working_customer: "Обработка"
      feedback_required: "Раследование"
      closed: "Закрыт"
  risklevel:
    type: "map"
    value: "risklevel"
```

```
  service_asset_id:
    type: "map"
    value: "service_asset_id"
  DETECTION_DATE:
    type: "map"
    value: "created_at"
  UPDATE:
    type: "map"
    value: "updated_at"
  finding_id:
    type: "map"
    value: "finding_id"
  analysis_output:
    type: "map"
    value: "analysis_output"
  synopsis:
    type: "map"
    value: "synopsis"
  title:
    type: "map"
    value: "title"
  risk:
    type: "map"
    value: "risk"
  OCCUR_DATE:
    type: "map"
    value: "acknowledged_at"
  alert_type:
    type: "map"
    value: "alert_type"
  client_note:
    type: "map"
    value: "client_note"
  internal_note:
    type: "map"
    value: "internal_note"
  external:
    type: "map"
    value: "external"
  immediate_action_score:
    type: "map"
    value: "immediate_action_score"
  throughput_period:
    type: "map"
    value: "throughput_period"
  throughput_period_change:
    type: "map"
    value: "throughput_period_change"
  customer_created:
    type: "map"
    value: "customer_created"
  c_visible_since:
    type: "map"
    value: "c_visible_since"
  c_visible_since_in_days:
```

```
    type: "map"
    value: "c_visible_since_in_days"
  c_reopened_count:
    type: "map"
    value: "c_reopened_count"
  c_last_customer_status_change:
    type: "map"
    value: "c_last_customer_status_change"
  c_customer_retention_time:
    type: "map"
    value: "c_customer_retention_time"
  logmule_identifier:
    type: "map"
    value: "logmule_identifier"
  c_remote_exploitable:
    type: "map"
    value: "c_remote_exploitable"
  c_occurrence_count:
    type: "map"
    value: "c_occurrence_count"
  last_occurrence_id:
    type: "map"
    value: "last_occurrence_id"
  itsm_last_synced_at:
    type: "map"
    value: "itsm_last_synced_at"
  itsm_sync_status:
    type: "map"
    value: "itsm_sync_status"
  external_id:
    type: "map"
    value: "external_id"
  itsm_sync_error:
    type: "map"
    value: "itsm_sync_error"
  user_id:
    type: "map"
    value: "user_id"
  updated_by:
    type: "map"
    value: "updated_by"
  group_id:
    type: "map"
    value: "group_id"
  acknowledged_by:
    type: "map"
    value: "acknowledged_by"
  created_by_customer:
    type: "map"
    value: "created_by_customer"
  edited_by:
    type: "map"
    value: "edited_by"
  active_name:
    type: "map"
```

```
    value: "active_name"
  IP:
    type: "map_from_json"
    value: "ip"
    from: "inet"
  fqdn:
    type: "map_form_json"
    value: "fqdn"
    from: "varchar_array"



# Outputs (Описание вызова API методов подключаемой системы)
rvision_insert: &rvision_insert
  type: http
  url: "https://192.168.10.0/api/v2/incidents"
  method: POST
  content_type: "application/json"
  headers:
    PgrApiKey: "test"
  mappging: *rvision_mapping

rvision_update: &rvision_update
  type: http
  url: "https://192.168.10.0/api/v2/incidents"
  method: POST
  content_type: "application/json"
  headers:
    PgrApiKey: "test"
  prepend:
    - type: http
      url: "https://192.168.10.0/api/v2/incidents"
      method: GET
      content_type: "application/json"
      query:
        token:
"61e1cce4c8b77de22574131da651d822a159225dd4d5f9781360c471046e1530"
        fields: "identifier"
        filter: '[{"operator":"=","value":%d,"property":"id_siem"}]'
      query_vars_from_trigger:
        filter:
          field: "id"
          type: "float64toInt64"
      append_to_mapping:
        identifier: "data.result.0.identifier"
  mappging: *rvision_mapping

# Inputs (Описание триггера вызова sql запроса)
radar-tables-trigger: &radar-tables-trigger
  - name: create_incident
    table: service_asset_findings
    fields: [ ]
    kind: insert
```

```
    sql: "SELECT t.id, CASE WHEN t.description ISNULL THEN f.description ELSE
t.description END, CASE WHEN t.risk_impact ISNULL THEN f.risk_impact ELSE
t.risk_impact END, CASE WHEN t.solution ISNULL THEN f.solution ELSE t.solution
END, t.mitigation, CASE WHEN t.solution ISNULL THEN f.solution ELSE t.solution
END, t.status, t.risklevel, t.service_asset_id, t.created_at, t.updated_at,
t.finding_id, t.analysis_output, t.synopsis, CASE WHEN t.synopsis ISNULL THEN
f.synopsis ELSE t.synopsis END, t.title, CASE WHEN t.title ISNULL THEN f.title
ELSE t.title END, t.risk, t.acknowledged_at, t.alert_type, t.client_note,
t.internal_note, t.external, t.immediate_action_score, t.throughput_period,
t.throughput_period_change, t.customer_created, t.c_visible_since,
t.c_visible_since_in_days, t.c_reopened_count, t.c_last_customer_status_change,
t.c_customer_retention_time, t.logmule_identifier, t.c_remote_exploitable,
t.c_occurrence_count, t.last_occurrence_id, t.itsm_last_synced_at,
t.itsm_sync_status, t.external_id, t.itsm_sync_error, t.user_id, t.updated_by,
t.group_id, t.acknowledged_by, t.created_by_customer, t.edited_by, s.name as
active_name, ni.ip as ip, ni.fqdn as fqdn FROM public.service_asset_findings t
left join findings f on t.finding_id = f.id left join service_assets s on
t.service_asset_id = s.id left join network_interfaces ni on s.id =
ni.service_asset_id where t.id = $1"
    sql_vars:
      id: float64toInt64
    outputs:
      - *rvision_insert

  - name: update_incident
    table: service_asset_findings
    fields: [ ]
    kind: update
    sql: "SELECT t.id, CASE WHEN t.description ISNULL THEN f.description ELSE
t.description END, CASE WHEN t.risk_impact ISNULL THEN f.risk_impact ELSE
t.risk_impact END, CASE WHEN t.solution ISNULL THEN f.solution ELSE t.solution
END, t.mitigation, CASE WHEN t.solution ISNULL THEN f.solution ELSE t.solution
END, t.status, t.risklevel, t.service_asset_id, t.created_at, t.updated_at,
t.finding_id, t.analysis_output, t.synopsis, CASE WHEN t.synopsis ISNULL THEN
f.synopsis ELSE t.synopsis END, t.title, CASE WHEN t.title ISNULL THEN f.title
ELSE t.title END, t.risk, t.acknowledged_at, t.alert_type, t.client_note,
t.internal_note, t.external, t.immediate_action_score, t.throughput_period,
t.throughput_period_change, t.customer_created, t.c_visible_since,
t.c_visible_since_in_days, t.c_reopened_count, t.c_last_customer_status_change,
t.c_customer_retention_time, t.logmule_identifier, t.c_remote_exploitable,
t.c_occurrence_count, t.last_occurrence_id, t.itsm_last_synced_at,
t.itsm_sync_status, t.external_id, t.itsm_sync_error, t.user_id, t.updated_by,
t.group_id, t.acknowledged_by, t.created_by_customer, t.edited_by, s.name as
active_name, ni.ip as ip, ni.fqdn as fqdn FROM public.service_asset_findings t
left join findings f on t.finding_id = f.id left join service_assets s on
t.service_asset_id = s.id left join network_interfaces ni on s.id =
ni.service_asset_id where t.id = $1"
    sql_vars:
      id: float64toInt64
    outputs:
      - *rvision_update


connections:
  - database: rmca
```

```
    username: radar
    password: *****
    host: 127.0.0.1
    port: 5432
    triggers: *radar-tables-trigger
    UseTLS: false
    SkipTLSVerify: false
    PgCert: ""
    PgKey: ""
    RootCrt: ""
```