

ЦЕНТРАЛЬНЫЙ ПУЛЬТ



Руководство для
администрирования

Содержание

1. Введение	6
1.1. Описание документа	6
1.2. Назначение системы	6
2. Системные требования	6
2.1. Системные требования к серверу	7
2.2. Системные требования к агенту	7
2.3. Сетевые доступы	8
3. Задачи администратора	11
3.1. Настройка параметров работы системы	11
3.1.1. Установка агента	11
3.1.1.1. Автоинсталляция	11
3.1.1.2. Linux	12
3.1.1.3. macOS	14
3.1.1.4. Windows	15
3.1.1.5. Wiren Board 6	16
3.1.2. Конфигурация агента	17
3.1.2.1. Подключение к Kafka Cluster	19
3.1.3. Конфигурация сервера	20
3.1.3.1. Agent_stat	20
3.1.3.2. Cache	21
3.1.3.3. Custom_scripts	22
3.1.3.4. DB	22
3.1.3.5. Elasticsearch	22
3.1.3.6. Extensions	23
3.1.3.7. Kafka	23
3.1.3.8. Keycloak	24
3.1.3.9. LDAP	24
3.1.3.10. Monitoring	24
3.1.3.11. MQTT	25
3.1.3.12. OpenTSDB	25
3.1.3.13. Push_notification	26
3.1.3.14. Resource_server	26
3.1.3.15. Rest_server	26
3.1.3.16. Server	27
3.1.3.17. SNMP	28

3.1.3.18. Wetty	29
3.1.3.19. Zabbix	30
3.1.4. Настройка элементов web-интерфейса	34
3.1.4.1. Authentication	35
3.1.4.2. Comet	36
3.1.4.3. DefaultClassImage	37
3.1.4.4. DefaultEmbeddableLink	37
3.1.4.5. Documentation	38
3.1.4.6. GeoMap	39
3.1.4.7. Grid	39
3.1.4.8. GridView	40
3.1.4.9. History	41
3.1.4.10. Incidents	42
3.1.4.11. Languages	42
3.1.4.12. Monitoring	43
3.1.4.13. Tree	44
3.1.4.14. Прочие параметры	44
3.1.5. Служба "saymon-server"	49
3.1.6. Просмотр websocket-нотификаций	50
3.1.7. Увеличение количества обработчиков SNMP-Trap	50
3.1.8. Сброс системы к заводским настройкам	51
3.1.9. Включение функционала генерации аварий	52
3.1.10. Передача аварий в сторонние сервисы	53
3.1.11. Работа со свойствами аварий	53
3.1.12. Перевод SNMP OID в текстовый формат	55
3.1.13. Выполнение скриптов при создании и удалении объектов	55
3.1.14. Автоматическое обнаружение объектов	56
3.2. Модуль аналитики	57
3.2.1. Подключение модуля аналитики к Центральному Пульту	58
3.2.2. Мониторинг работы модуля аналитики	59
3.3. Управление сенсорами мониторинга	59
3.3.1. Скрытие неиспользуемых сенсоров	59
3.3.2. Настройка пользовательских сенсоров	60
3.4. Настройка уведомлений	62
3.4.1. Активация функционала отправки SMS и голосовых вызовов	62
3.4.2. Отправка почтовых уведомлений	63
3.4.3. Настройка уведомлений в Telegram	64

3.5. Интеграция со сторонними системами	66
3.5.1. Интеграция с Zabbix.	66
3.5.2. Интеграция с Grafana.	69
3.5.3. Интеграция с Keycloak.	73
3.5.3.1. Настройка сервера	73
3.5.3.2. Настройка web-интерфейса.	79
3.6. Управление логированием	84
3.6.1. Конфигурация log-файлов агента.	84
3.6.1.1. Базовое логирование	84
3.6.1.2. Активация debug-режима.	85
3.6.2. Конфигурация ротации log-файлов сервера	86
3.6.3. Просмотр информации о Журнале событий	86
3.6.4. Назначение ответственного за событие	87
3.6.5. Установка ограничения для логирования	87
3.6.6. Удаление всех SNMP-Трап'ов из Журнала событий	88
3.6.7. Удаление логов.	89
3.7. Управление учётными записями пользователей.	90
3.7.1. Авторизация пользователей через LDAP.	90
3.7.2. Авторизация пользователей через Keycloak.	92
3.7.3. Создание учётной записи.	92
3.7.4. Назначение пользователям прав доступа.	94
3.7.5. Назначение парольных политик	96
3.7.6. Изменение пароля от учётной записи.	97
3.7.7. Настройка доступа к объектам.	98
3.7.8. Блокировка учётной записи	99
3.7.9. Удаление учётной записи	100
3.7.9.1. Просмотр журнала действий пользователей	101
3.8. Работа с объектами и связями.	102
3.8.1. Создание объекта	103
3.8.2. Клонирование объекта	104
3.8.3. Удаление объекта	104
3.8.4. Создание ссылки на объект	105
3.8.5. Создание связи	106
3.8.6. Удаление связи	108
3.9. Настройка интерфейса	108
3.9.1. Выравнивание/расстановка объектов в стандартном виде.	108
3.9.2. Настройка заголовка web-интерфейса.	109

3.9.3. Перемещение/отключение фоновой иконки объекта	109
3.9.4. Вертикальное отображение имени объекта	110
3.9.5. Редактирование стилей состояний	110
3.10. Настройка мониторинга	111
3.10.1. Мониторинг основных параметров ПК	111
3.10.2. Мониторинг процесса памяти	112
3.10.3. Мониторинг изменения файлов и папок	112
3.10.4. Проверка доступности web-ресурса	112
3.10.5. Безагентный мониторинг web-сервера	113
3.11. Страницы для внедрения в IFrame	114
3.11.1. Страница с авариями	114
3.11.1.1. Параметры элементов управления	114
3.11.1.2. Параметры фильтра	116
3.11.2. Страница с графиками	117
3.11.3. Страница с журналом сессий	123
3.12. Открытие терминала удалённого доступа	124
3.13. Переход к web-интерфейсу устройства	125
3.14. Резервное копирование и восстановление данных	126
3.15. Серверные расширения	129
4. Проблемы в работе системы и способы их решения	132
4.1. Недостаточно места на виртуальной машине с сервером	132
4.2. Отсутствие подключения агента к серверу	133
4.3. Проверка работы MongoDB	134
4.4. Проверка работы Redis	134
4.5. 500 Internal Server Error и отсутствие графиков	135
4.6. Ошибка работы HTTP-проверки	136
Приложение А (обязательное)	136
Приложение Б (обязательное)	142

1. Введение

Центральный Пульт — это платформа для визуализации и мониторинга работы сети, оборудования, приложений и служб.

1.1. Описание документа

Настоящий документ является руководством для администрирования автоматизированной системы "Центральный Пульт" и предназначен для конкретизации задач и функций должностных лиц организации (предприятия, фирмы), планирующих и осуществляющих сбор, хранение, передачу и анализ данных по объектам мониторинга с применением системы.

В документе приведены основные функции администратора, архитектура системы и её модулей, алгоритм создания учётных записей, порядок установки прав доступа пользователей и другие сведения, необходимые для управления АС "Центральный Пульт".

1.2. Назначение системы

Система предназначена для визуализации и мониторинга различных объектов. Центральный Пульт нацелен на упрощение сбора данных, ускорение их анализа, визуализации результатов и непрерывного хранения.

Автоматизации подвергаются следующие функциональные возможности процесса мониторинга:

- процесс обработки данных;
- хранение оригинальных значений;
- обеспечение анализа информации;
- управление объектами мониторинга;
- уведомление пользователей о состояниях объектов;
- исправление аварийных ситуаций;
- преобразование данных в компактный вид;
- экспорт данных;
- удаление устаревших данных.

2. Системные требования

Система может быть установлена на выделенных аппаратных или виртуальных мощностях.

Для стабильной работы клиента системы необходим web-браузер Google Chrome версии не

ниже 58.0.

2.1. Системные требования к серверу

Для работы сервера системы требуется следующая конфигурация:

- 64-bit OS;
- CPU - 4 cores;
- RAM - 8 GB;
- HDD - 72 GB.

Гарантируется корректная работа инсталляционного скрипта сервера на следующих ОС:

- Ubuntu Linux 20.04;
- Ubuntu Linux 18.04.

Установка системы также возможна на операционные системы:

- РЕД ОС,
- Red Hat Enterprise Linux,
- Astra Linux,
- Debian.

"Центральный Пульт" может быть запущен и на других дистрибутивах Linux. Свяжитесь с нами welcome@saymon.info, если вы не нашли нужную вам операционную систему среди перечисленных.

Серверная часть системы может быть поставлена в виде готового образа виртуальной машины или Docker-контейнера. Ссылки можно получить в службе технической поддержки по адресу welcome@saymon.info.

2.2. Системные требования к агенту

Агенты обладают кросс-платформенной совместимостью и могут быть установлены на различные операционные системы, например:

- Ubuntu Linux;
- Red Hat Enterprise Linux / CentOS Linux;
- Raspberry Pi;
- macOS;
- Windows.

Требованием к операционным системам является поддержка Java SE 6, 7 и 8.

Рекомендуемая конфигурация для работы агентов системы:

- OS with Java 6/7/8 support;
- CPU – 2 GHz single core;
- RAM – 1 GB;
- HDD – OS + 2 GB

2.3. Сетевые доступы

В таблице ниже перечислены порты по умолчанию, необходимые для корректной работы Центрального Пульта:

Порт	Протокол	Сервис	Инициатор соединения	Место назначения	Комментарий
22	TCP	SSH	Сотрудники технической поддержки	Все серверы платформы	Удалённый доступ к серверам для настройки и оказания технической поддержки.
80	TCP	HTTP	Пользователи Центрального Пульта	Серверы с сервисами Центрального Пульта	Доступ к web-интерфейсу платформы. Рекомендуется перенаправление на 443 порт.
162	UDP	SNMP Trap	Оборудование, отправляющее SNMP Trap	Серверы с агентами	Получение SNMP Trap от оборудования. По умолчанию на каждом сервере с сервисами Центрального Пульта установлен агент, который получает SNMP Trap'ы.

Порт	Протокол	Сервис	Инициатор соединения	Место назначения	Комментарий
443	TCP	HTTPS	Пользователи Центрального Пульта	Серверы с сервисами Центрального Пульта	Доступ к web-интерфейсу Центрального Пульта с шифрованием трафика.
3001	TCP	Wetty	Пользователи Центрального Пульта	Оборудование в инфраструктуре заказчика	Доступ к терминалам устройств через SSH и Telnet.
4242	TCP	OpenTSD B	Серверы с сервисами Центрального Пульта	Серверы с OpenTSDB	Доступ к базе данных хранения временных рядов для построения графиков. Необходим при инсталляции OpenTSDB на отдельных серверах.
6379	TCP	Redis	Серверы с сервисами Центрального Пульта	Серверы с Redis	Доступ к кэшу данных. Необходим при инсталляции Redis на отдельных серверах.
8090	TCP	SAYMON REST API	Пользователи, интеграционные модули и расширения	Серверы с сервисами Центрального Пульта	Взаимодействие с системой и данными через REST API.

Порт	Протокол	Сервис	Инициатор соединения	Место назначения	Комментарий
8091	TCP	SAYMON Notification Actor	Пользователи Центрального Пульта	Серверы с сервисами Центрального Пульта	Необходим для быстрого обновления данных в web-интерфейсе и мобильных клиентах через WebSocket.
8092	TCP	SAYMON Resource Server	Пользователи, интеграционные модули и расширения	Серверы с сервисами Центрального Пульта	Отображение в web-интерфейсе приложенных к объектам документов и фоновых изображений.
9092	TCP	Kafka	Серверы с сервисами Центрального Пульта и агентами	Серверы с Kafka	Доступ к шине данных. Необходим при инсталляции системы на нескольких серверах и при использовании агента для сбора данных.
27017	TCP	MongoDB	Серверы с сервисами Центрального Пульта	Серверы с MongoDB	Доступ к базе данных основной информации об объектах и пользователях. Необходим при инсталляции MongoDB на отдельных серверах.

3. Задачи администратора

Глава содержит информацию об основных задачах, возникающих в процессе администрирования платформы.

3.1. Настройка параметров работы системы

Раздел настройки – один из самых важных разделов, предназначенный для администратора системы. Этот раздел наполнен советами о том, как настроить Центральный Пульт для мониторинга вашей среды, начиная настройкой сервера для получения необходимой информации и заканчивая просмотром данных, настройкой оповещений и удалённых команд, выполняемых в случае возникновения проблем.

3.1.1. Установка агента

Перед началом работы с платформой "Центральный Пульт" необходимо выполнить следующие действия:

1. Получить актуальную версию агента одним из способов:
 - зайти на сайт платформы на страницу "Загрузки" saymon.info/support_ru/downloads_en/ и скачать подходящую под ОС сборку агента;
 - перенести с CD-ROM, USB-накопителя или другого носителя, на котором поставляется платформа "Центральный Пульт", подходящую под пользовательскую ОС сборку агента на компьютер, сервер или устройство, на котором планируется осуществлять сбор данных.
2. Установить и настроить агента согласно дальнейшим инструкциям.

WARNING

*При установке агента на Linux в системе автоматически создаётся пользователь **saymon**, от имени которого будет работать агент.*

Если в системе уже есть пользователь с таким логином, работа агента будет проходить от его имени.

3.1.1.1. Автоинсталляция

Чтобы приступить к началу работы и настройке мониторинга, необходимо выполнить следующие действия:

1. Открыть web-интерфейс платформы "Центральный Пульт".
2. Ввести логин и пароль учётной записи с правами на управление объектами, свойствами и документами.

NOTE

Назначение прав пользователей описано в подразделе 3.7.4, "Назначение пользователям прав доступа".

3. Создать объект-агент, для этого нажать кнопку **+ Создать объект** на панели инструментов, ввести имя объекта и выбрать класс **Saymon Agent**.
4. Навести курсор на созданный объект и нажать на появившуюся иконку - Настройки агента (Рис. 1):



Рис. 1. Мониторинговый агент

5. В появившемся окне скопировать ссылку из строки "Команда для установки агента" (Рис. 2):

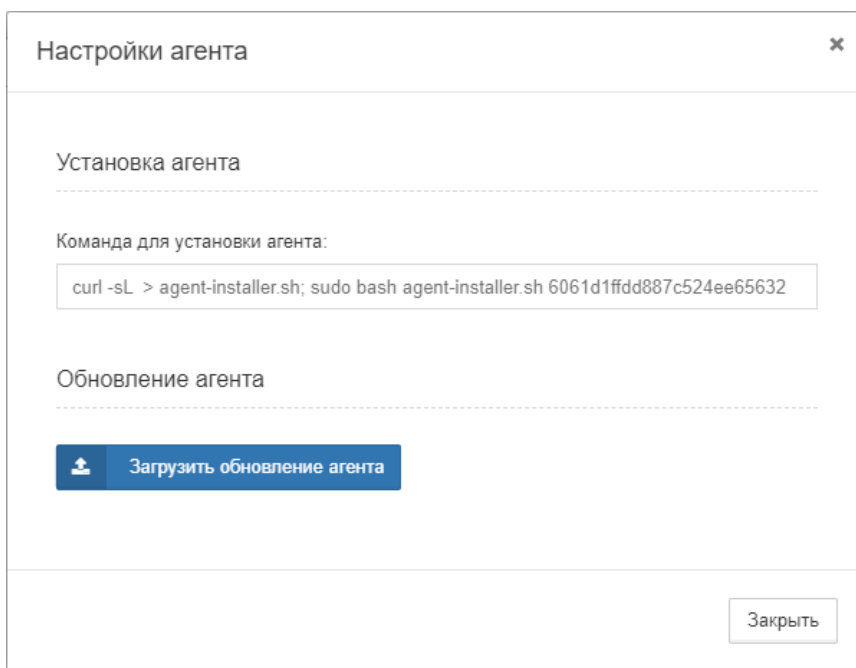


Рис. 2. Команда для установки агента

6. Выполнить эту команду в терминале на необходимом сервере.

NOTE

Номер, который дал сервер (последние 24 символа команды) совпадает с ID агента.

В течение 30 секунд агент скачивается и устанавливается. Если установка корректна, состояние агента в web-интерфейсе будет отражено зелёным цветом: **Saymon Agent**.

3.1.1.2. Linux

Инструкция применима к операционным системам с менеджерами служб "systemd",

"upstart" и "init.d", например:

- Ubuntu Linux;
- Red Hat Enterprise Linux;
- CentOS Linux;
- Debian 8 "Jessie" (для пользователей Raspberry Pi).

Если вы используете другого менеджера, пишите нам на care@saymon.info.

WARNING

*При установке агента в системе автоматически создаётся пользователь **saymon**, от имени которого будет работать агент.*

Если в системе уже есть пользователь с таким логином, работа агента будет проходить от его имени.

Для установки агента на Linux необходимо:

На хосте без доступа в Интернет

1. [Скачать sh-скрипт установщика агента](#) для Linux 64 bit или Linux 32 bit.
2. Перенести скрипт на нужный хост в домашнюю директорию пользователя.
3. Сделать скрипт исполняемым:

64 bit

```
sudo chmod +x saymon-agent-rl-linux-x64-jre-installer.sh
```

32 bit

```
sudo chmod +x saymon-agent-rl-linux-i586-jre-installer.sh
```

4. Запустить скрипт:

64 bit

```
sudo ./saymon-agent-rl-linux-x64-jre-installer.sh
```

32 bit

```
sudo ./saymon-agent-rl-linux-i586-jre-installer.sh
```

5. Выполнить дальнейшие инструкции в терминале.

На хосте с доступом в Интернет

1. Выполнить однострочник:

64 bit

```
curl https://saymon.info/downloads/saymon-agent-rl-linux-x64-jre-  
installer.sh -o saymon-agent-instal.sh ; chmod +x saymon-agent-instal.sh ;  
sudo ./saymon-agent-instal.sh
```

32 bit

```
curl https://saymon.info/downloads/saymon-agent-rl-linux-i586-jre-  
installer.sh -o saymon-agent-instal.sh ; chmod +x saymon-agent-instal.sh ;  
sudo ./saymon-agent-instal.sh
```

2. Выполнить дальнейшие инструкции в терминале.

Конфигурация агента

При необходимости можно отредактировать конфигурационный файл агента `/opt/saymon-agent/conf/agent.properties`.

3.1.1.3. macOS

Для установки агента на macOS необходимо:

1. Создать системного пользователя **Saymon agent**, из-под которого будет запускаться агент.
2. [Скачать архив](#) Mac OS X 64 bit и распаковать его в папку `/opt`.
3. Отредактировать конфигурационный файл агента `saymon-agent/conf/agent.properties`.
4. Создать папку для хранения log-файлов:

```
$sudo mkdir saymon-agent/log && sudo chown -R saymon:staff saymon-agent
```

5. Сделать файл агента исполняемым:

```
$sudo chmod +x saymon-agent/saymon-agent.sh
```

6. Запустить агента:

```
$cd saymon-agent && sudo -u saymon ./saymon-agent.sh
```

3.1.1.4. Windows

Для установки агента на Windows необходимо:

1. [Скачать инсталлятор](#) для Windows 64bit или Windows 32bit.
2. Запустить скачанный файл от имени администратора.
3. При необходимости изменить каталог установки по умолчанию (для Windows 64-bit – **C:\Program Files (x86)\SAYMON Agent**, для Windows 32-bit – **C:\Program Files\SAYMON Agent**) и нажать кнопку **Install** (Рис. 3):

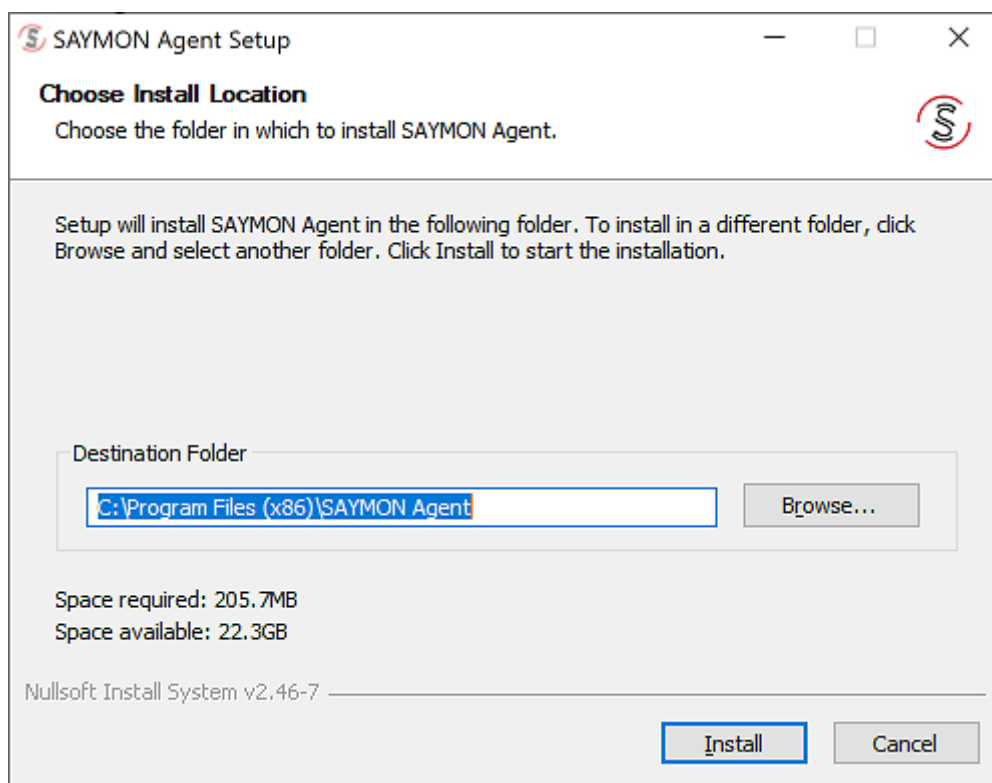


Рис. 3. Выбор каталога установки

1. После завершения установки нажать кнопку **Close**.
2. В открывшемся окне ознакомиться с дальнейшими инструкциями по установке агента и нажать кнопку **OK** (Рис. 4):

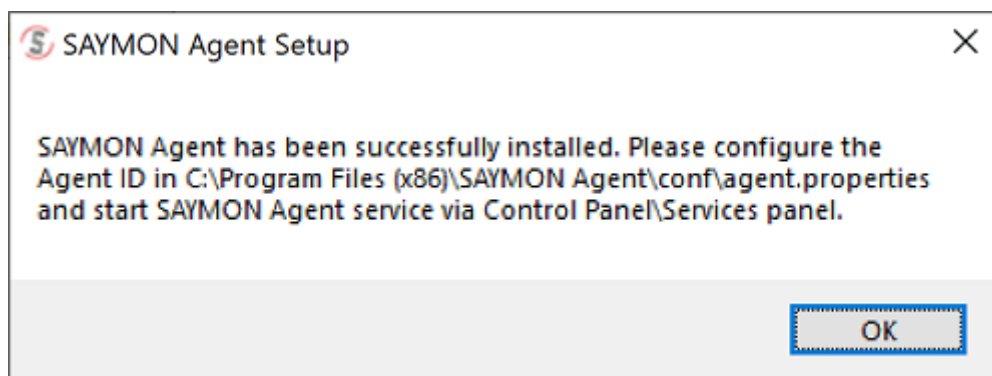


Рис. 4. Первичная настройка агента

1. Перейти в папку **conf** каталога установки агента и открыть конфигурационный файл

агента **agent.properties**.

2. В качестве значения параметра **server.host** указать IP-адрес сервера Центрального Пульта, к которому должен подключиться агент. Закрыть файл, сохранив изменения.
3. Открыть оснастку "Службы" и запустить службу с именем **saymon-agent**.

Агент работает со следующими версиями ОС Windows:

- Windows 7 SP1, 64 bit;
- Windows 7 SP1, 32 bit;
- Windows 10 22H2, 64 bit;
- Windows 10 22H2, 32 bit;
- Windows 10 Enterprise, 64 bit;
- Windows 10 Enterprise, 32 bit;
- Windows 11 Enterprise 22H2, 64 bit;
- Windows Server 2012 (R2 Datacenter), 64 bit;
- Windows Server 2016 (R2 Datacenter), 64 bit;
- Windows Server 2019, 64 bit;
- Windows Server 2022 (Standard Evaluation 21H2), 64 bit.

3.1.1.5. Wiren Board 6

Wiren Board 6 – это универсальный контроллер для автоматизации с открытым ПО на базе Linux. Предназначен для домашней и промышленной автоматизации и мониторинга: опроса датчиков и счетчиков, использования в качестве УСПД, в системах АСКУЭ, для замены ПЛК, а также в системах "умного дома".

Для установки агента на контроллер автоматизации Wiren Board 6 необходимо:

1. [Скачать архив с агентом](#).
2. Распаковать архив в папку `/opt/saymon-agent/`.
3. [Скачать архив с JDK](#) (потребуется регистрация на сайте).
4. Распаковать архив в папку `temp`, оттуда полностью скопировать директорию `jre` в папку `/opt/saymon-agent/`.
5. Выполнить следующие команды:

```
$useradd -M -r -s /bin/false -K MAIL_DIR=/dev/null saymon
$chown -R saymon:saymon /opt/saymon-agent
$cp /opt/saymon-agent/systemd/* /etc/systemd/system
$systemctl enable saymon-agent
$service saymon-agent start
```


6. При необходимости отредактировать файл конфигурации агента **/opt/saymon-agent/conf/agent.properties**.

Благодарим за оказанную помощь пользователя [svdu](#) с [форума компании Wiren Board](#).

3.1.2. Конфигурация агента

Конфигурация агента выполняется в файле **.../saymon-agent/conf/agent.properties**.

Для применения изменений необходимо перезапустить службу **saymon-agent**.

Общие параметры конфигурационного файла агента:

Параметр	Описание
agent.discoveryEnabled	<p>Включает автоматический поиск агентом сетевых устройств;</p> <ul style="list-style-type: none"> • false - автоматический поиск выключен, • true - автоматический поиск включен. <p>По умолчанию - false - автоматический поиск выключен.</p> <p><i>Задать родительский объект для обнаруженных устройств можно с помощью параметра discovery_parent_id в секции Server конфигурационного файла сервера /etc/saymon/saymon-server.conf.</i></p>
agent.id	<p>Уникальный идентификатор объекта класса Saymon Agent в web-интерфейсе системы.</p> <p><i>При указании agent.id=0 на сервере будет создан новый объект класса Saymon Agent. По умолчанию агенты создаются в корневом объекте, задать родительский объект для создаваемых агентов можно в параметре agent_registration_parent_id в секции Server конфигурационного файла сервера /etc/saymon/saymon-server.conf.</i></p> <p><i>ID созданного объекта будет передан агенту. На хосте с агентом будет создан файл *.../saymon-agent/storage/registration.json*, где будет записан данный идентификатор.</i></p> <p><i>Чтобы заново инициировать процедуру получения ID, к примеру, если необходимо подключить агента к другому серверу, достаточно удалить файл registration.json и перезапустить агента.</i></p>

Параметр	Описание
agent.scriptsEnabled	<p>Включает выполнение агентом скриптов с указанным текстом;</p> <ul style="list-style-type: none"> • false - режим оптимизации выключен, • true - режим оптимизации включен. <p>По умолчанию – true – выполнение скриптов включено.</p>
agent.snmpTrapEnabled	<p>Позволяет использовать агента в качестве получателя трапов;</p> <ul style="list-style-type: none"> • false - получение трапов агентом выключено, • true - получение трапов агентом включено. <p>По умолчанию - false - получение трапов выключено.</p>
agent.snmpTrapListenPort	<p>Порт для получения трапов.</p> <p>По умолчанию - 1162.</p>
agent.snmpTrapReceiverThreadPoolSize	<p>Количество одновременных потоков для получения трапов.</p> <p>По умолчанию - 4.</p>
metrics.buffer.in-memory.capacity	<p>Включает буфер для сохранения данных от агента при отсутствии связи между агентом и сервером.</p> <p>Значение параметра - количество наборов данных от агента (результатов проверок), которые будут храниться в буфере до восстановления связи с сервером. После восстановления связи все данные из буфера пересылаются на сервер, и буфер очищается. При переполнении буфера очередные наборы данных заменяют наиболее старые из имеющихся в буфере.</p> <p>По умолчанию буфер выключен.</p> <p><i>Функционал поддерживается агентами версии 4.5.72 и выше.</i></p>
server.host	<p>Адрес сервера Kafka.</p> <p>По умолчанию - 127.0.0.1.</p>
server.port	<p>Порт Kafka, по которому осуществляется подключение агентов к серверу.</p> <p>По умолчанию - 9092.</p>

3.1.2.1. Подключение к Kafka Cluster

При использовании Kafka Cluster необходимо также использовать дополнительные параметры подключения, начинающиеся на **kafka**.

Все возможные параметры представлены в [документации по Kafka](#).

Пример:

```
...
kafka.bootstrap.servers=localhost:9092

kafka.security.protocol=SASL_SSL

kafka.ssl.truststore.location=/opt/kafka/kafka.client.truststore.jks

kafka.ssl.truststore.password=saymon

kafka.sasl.mechanism=PLAIN

kafka.sasl.jaas.config=org.apache.kafka.common.security.plain.PlainLoginModule
required \

username="user" \

password="password"
...
```

kafka.bootstrap.servers - список пар хост:порт, использующихся для установки первоначального соединения с кластером Kafka.

WARNING

При использовании Kafka Cluster поле **server.host** не должно быть пустым, там необходимо указать адрес одного из **kafka.bootstrap.servers**.

Параметры конфигурационного файла агента для авторизации и шифрования:

Параметр	Описание
kafka.sasl.jaas.config	Параметры контекста входа в систему для SASL-соединений в формате, используемом конфигурационными файлами JAAS.
kafka.sasl.mechanism	Механизм SASL, используемый для клиентских подключений.

Параметр	Описание
kafka.security.protocol	<p>Протокол, используемый для взаимодействия с брокерами.</p> <p>Возможные значения:</p> <ul style="list-style-type: none"> • PLAINTEXT, • SSL, • SASL_PLAINTEXT, • SASL_SSL.
kafka.ssl.truststore.location	Расположение файла truststore .
kafka.ssl.truststore.password	Пароль для файла truststore . Если пароль не задан, сконфигурированный файл также будет использоваться, но без проверки целостности.

3.1.3. Конфигурация сервера

Конфигурация сервера системы выполняется в файле `/etc/saymon/saymon-server.conf`.

Для применения изменений необходимо перезапустить службу **saymon-server**:

```
$sudo service saymon-server restart
```

Далее приведены описания разделов и настроек.

3.1.3.1. Agent_stat

В этом разделе задаются настройки хранения истории данных от агента.

```
"agent_stat": {
  "write_to": "mongo",
  "default_ttl": 20,
  "ttl": [
    {
      "key": "payload.cpuInformation.vendor",
      "value": "Intel",
      "expire": 10
    }
  ]
}
```

Agent_stat.ttl

Сроки жизни наборов данных, для которых нужен срок жизни, отличающийся от принятого по умолчанию.

Раздел/Параметр	Описание
agent_stat.ttl. key	Имя метрики.
agent_stat.ttl. value	Значение метрики.
agent_stat.ttl. expire	Срок жизни набора данных.

3.1.3.2. Cache

В этом разделе задаются настройки сервера Redis.

```
"cache": {
  "host": "127.0.0.1",
  "port": 6379
}
```

Cache.cluster

В этом подразделе задаются настройки при использовании Redis Cluster.

```
"cache": {
  "cluster": {
    "nodes": [
      {
        "port": 7000,
        "host": "127.0.0.1"
      },
      {
        "port": 7001,
        "host": "127.0.0.1"
      }
    ],
    "options": {}
  }
}
```

WARNING

Если задействован этот подраздел, система использует указанные здесь настройки подключения, параметры **cache.host** и **cache.port** игнорируются.

3.1.3.3. Custom_scripts

В этом разделе задаются настройки пользовательских скриптов, выполняющихся при создании и удалении объектов.

```
"custom_scripts": {  
  "entity_triggers_path": "/opt/saymon-extensions/entity-triggers-path"  
}
```

3.1.3.4. DB

В этом разделе задаются настройки подключения к базе данных для хранения объектной модели. В качестве базы данных используется MongoDB.

```
"db" : {  
  "mongodb": {  
    "url": "mongodb://localhost:27017/saymon?w=1"  
  }  
}
```

Db.mongodb

Данные для подключения к MongoDB.

3.1.3.5. Elasticsearch

В этом разделе задаются настройки для работы с Elasticsearch.

```
"elasticsearch": {  
  "node": "http://127.0.0.1:9200",  
  "auth": {  
    "username": "elastic",  
    "password": "ypezA1ZoVv46fJh8Abj4"  
  },  
  "index_pattern": "myindex*",  
  "request_timeout": 30000,  
  "hits_per_request": 1000  
}
```

Elasticsearch.auth

Авторизационные данные пользователя elasticsearch.

Раздел/Параметр	Описание
elasticsearch.auth. password	Пароль пользователя elasticsearch.
elasticsearch.auth. username	Логин пользователя elasticsearch.

3.1.3.6. Extensions

В этом разделе задаются параметры пользовательских расширений Центрального Пульта.

```
"extensions": {
  "path": "/opt/saymon-extensions"
}
```

3.1.3.7. Kafka

В этом разделе задаются настройки подключения к брокеру Kafka.

```
"kafka" : {
  "host" : "localhost",
  "port" : 9092,
  "requestTimeout" : 30000,
  "fetch_latest" : false,
  "logLevel" : 2
}
```

Помимо перечисленных здесь параметров, возможно также использовать [дополнительные параметры подключения](#).

Kafka.sasl

[Настройки аутентификации](#). Параметры зависят от механизма аутентификации, используемом в Kafka.

Пример настроек для механизма PLAIN/SCRAM:

```
"kafka" : {
  ...
  "sasl": {
    "mechanism": "plain",
    "username": "user",
    "password": "password"
  },
  ...
}
```

}

Раздел/Параметр	Описание
kafka.sasl.mechanism	Механизм аутентификации.

3.1.3.8. Keycloak

В этом разделе задаются параметры для взаимодействия с сервером Keycloak для авторизации пользователей.

```
"keycloak": {
  "realm_certificate": "<data of keycloak_realm_certificate>",
  "realm_certificate_file": "/etc/saymon/keycloak_realm_certificate",
  "config_file": "/etc/saymon/keycloak_backend.json"
}
```

WARNING

Параметры в примере перечислены в порядке понижения приоритета при одновременном указании. Для работы с Keycloak достаточно указать один из параметров.

3.1.3.9. LDAP

В этом разделе задаются параметры внешнего LDAP-сервера для авторизации пользователей.

```
"ldap" : {
  "url" : "ldaps://192.168.1.1:636",
  "suffix" : "dc=example,dc=com",
  "login" : "cn=admin,dc=example,dc=com",
  "pass" : "root",
  "allow_self_signed" : true,
  "create_user_for_existing_group_only": true,
  "import_non_existing_groups": false,
  "update_existing_groups": false
}
```

3.1.3.10. Monitoring

В этом разделе задаются настройки пользовательских сенсоров мониторинга.

```
"monitoring": {
  "custom_tasks_path": "/opt/saymon-agent/custom_tasks"
}
```


3.1.3.11. MQTT

В этом разделе задаются настройки подключения к MQTT-брокеру.

```
"mqtt" : {
  "broker" : "mqtt://username:password@localhost:1883"
}
```

Раздел/Параметр	Описание
mqtt.broker	<p>Адрес и порт брокера.</p> <p>По умолчанию - "mqtt://localhost:1883".</p> <div style="border: 1px solid orange; padding: 5px; margin-top: 10px;"> <p>WARNING Для аутентификации по имени пользователя и паролю нужно указать пользовательские данные перед адресом сервера.</p> </div>

3.1.3.12. OpenTSDB

В этом разделе задаются параметры доступа к OpenTSDB.

```
"OpenTSDB" : {
  "enabled" : true,
  "host" : "localhost",
  "port" : 4242,
  "timeout" : 15000
}
```

Раздел/Параметр	Описание
openTsdB.enabled	<p>Запись исторических данных в OpenTSDB;</p> <ul style="list-style-type: none"> • false - запись выключена, • true - запись включена. <p>По умолчанию - true - запись включена.</p>
openTsdB.host	<p>Адрес хоста с OpenTSDB.</p> <p>По умолчанию - "localhost".</p>
openTsdB.port	<p>Порт OpenTSDB.</p> <p>По умолчанию - 4242.</p>

Раздел/Параметр	Описание
<code>openTsdB.timeout</code>	Время на ожидание ответа от OpenTSDb (в миллисекундах). По умолчанию - 15000 - 15 секунд.

3.1.3.13. Push_notification

В этом разделе задаются параметры push-уведомлений в мобильном приложении. Для работы с ними используется Firebase Cloud Messaging (FCM).

```
"Push_notification" : {
  "disabled" : true,
  "key_path" : "/etc/saymon/saymon-mobile-firebase-adminsdk.json",
  "on_state_change" : true,
  "timeout" : 0,
  "url" : "https://saymon-mobile.firebaseio.com"
}
```

3.1.3.14. Resource_server

В этом разделе задаются параметры, связанные с хранением файлов, загруженных в Центральный Пульт.

```
"Resource_server" : {
  "debug" : false,
  "file_storage_dir" : "/var/saymon/resources",
  "ip_address" : "127.0.0.1",
  "port" : 8092
}
```

3.1.3.15. Rest_server

В этом разделе задаются параметры REST-сервера.

```
"Rest_server" : {
  "base_url" : "/api",
  "colorize_log" : false,
  "debug" : false,
  "document_download_url" : "http://localhost/node/api/docs",
  "ip_address" : "127.0.0.1",
  "keepAliveTimeout" : 5000,
  "num_workers" : 1,
  "port" : 8090,
  "public_url" : "http://mycompany.com/index.html",
  "session_ttl" : 3600000,
  "snmp_mib_url" : "http://192.168.1.189:5550",
}
```

```
"update_download_url" : "http://localhost/node/api/agents/update"
}
```

Rest_server.limits

В этом подразделе задаются ограничения на число запросов к [API](#).

```
"limits" : [
  {
    "__comment": "администратора не ограничиваем",
    "user": "5048c849d7b6e40593dfee71"
  },
  {
    "__comment": "группу дежурной смены не ограничиваем",
    "group": "60e591928196105ec9c74636"
  },
  {
    "__comment": "запросы на states и classes не ограничиваем",
    "uri": ["/states", "/classes"]
  },
  {
    "__comment": "ограничиваем запросы на объекты",
    "uri": ["/objects"],
    "rate": "100/minute"
  },
  {
    "__comment": "ограничиваем авторизовавшихся пользователей с 127.0.0.1",
    "user": true,
    "ip": '127.0.0.1',
    "rate": "120/minute"
  },
  {
    "__comment": "ограничения для неавторизованных пользователей",
    "user": false,
    "rate": "10/minute"
  }
]
```

По умолчанию ограничения не накладываются.

WARNING

Для каждого запроса поиск правил осуществляется сверху вниз.
Применяется только одно правило.

3.1.3.16. Server

В этом разделе задаются общие параметры сервера.

Server.email

В этом подразделе задаются параметры доступа к почтовому серверу:

```
"email" : {
  "disabled" : false,
  "fields" : {
    "from" : "saymon@saas.saymon.info"
  },
  "max_json_length": 1000,
  "transport" : {
    "auth" {
      "user" : "saymon@saas.saymon.info",
      "pass" : "P@ssw0rd"
    },
    "host" : "smtp.gmail.com",
    "port" : 465,
    "secure" : true
  },
}
```

Server.user

В этом подразделе задаются параметры пользователей:

```
"user" : {
  "auth_enabled": "true",
  "new_user_access": "all",
  "template": {
    "permissions": ["manage-objects", "view-section-stat"],
    "objectPermissions": {
      "include": [],
      "exclude": ["5fb643ddf277b96c8401119b", "5f8dc28407e86603bfe281f8"]
    }
  },
  "usersRoot": "5800d9aaac7bf0f90d3d520e"
}
```

3.1.3.17. SNMP

В этом разделе задаются параметры обработки SNMP-трапов.

```
"snmp" : {
  "filter": "/users/admin/tmp/snmp-filter.js"
}
```

Раздел/Параметр	Описание
snmp.filter	<p>Путь к файлу с функцией фильтрации входящих SNMP Trap.</p> <p>Если функция возвращает значение false, сообщение отбрасывается.</p> <p>Приведённый ниже пример блокирует трапы, которые содержат любую из строк, указанных в массиве <i>filters</i>:</p> <p><i>Пример функции фильтрации</i></p> <pre> module exports === function(message, rawMessage) { rawMessage === String(rawMessage); //console.log(message, rawMessage); let filters === ['HLR Call Processing Message', 'AIR Rejected - No EPS Subscription', 'GPRS SIBB Processing Exception', 'AuC Call Processing Message', 'Received notification from CLIM', 'AuC Event Log Clean Up', 'SOFT 0006']; return !filters some((filter) => rawMessage.includes (filter)); } </pre> <p>В данном примере блокируется трап с <i>enterpriseOid</i> === '1.3.6.1.4.1.169.10.1.1' И <i>genericTrap</i> === 6 И <i>specificTrap</i> === 5:</p> <p><i>Пример функции фильтрации</i></p> <pre> module exports === function(message, rawMessage) { return message payload.enterpriseOid !== '.1.3.6.1.4.1.169.10.1.1' && message payload.genericTrap !== 6 && message.payload.specificTrap !== 5; } </pre>

3.1.3.18. Wetty

В этом разделе задаются параметры логирования истории взаимодействия пользователей с системой через терминал Wetty.

NOTE	<p>История каждой пользовательской сессии сохраняется в отдельный файл.</p> <p>Формат имени файла: YYYY-MM-DD_HH_MM_SS_PROTOCOL_IP_USERNAME.log, например 2023-01-12_11-56-17_ssh_127.0.0.1_admin.log.</p>
-------------	--

Если используется версия Центрального Пульта ниже 3.12.86, при переходе на новую версию необходимо:

1. Внести изменения в файл `/etc/nginx/sites-available/saymon`:

```
location ~ /wetty/(.*)
{
    # take path and query after wetty/ and append it to
    proxy_pass
    proxy_pass http://127.0.0.1:3001/wetty/$1?$query_string;
    proxy_http_version 1.1;
    proxy_set_header Upgrade $http_upgrade;
    proxy_set_header Connection "upgrade";
}
```

WARNING

2. Перезапустить nginx:

```
service nginx reload
```

```
"wetty" : {
  "log_dir" : "/tmp",
  "raw_mode": true
}
```

Раздел/Параметр	Описание
<code>wetty.log_dir</code>	Путь к папке для хранения log-файлов.
<code>wetty.raw_mode</code>	Включает режим записи логов с управляющими ANSI-символами; <ul style="list-style-type: none"> • <code>true</code> - логи записываются с управляющими ANSI-символами; • <code>false</code> - логи записываются без управляющих ANSI-символов. По умолчанию - true - логи записываются с ANSI-символами.

3.1.3.19. Zabbix

В этом разделе задаются параметры для импорта данных из системы Zabbix.

```
"zabbix" : [
  {
    "url" : "http://192.168.1.215/zabbix/api_jsonrpc.php",
    "user" : "saymon",
    "password" : "saymon_user_password",
```

```

"parent_id" : "58b586d5c3a2f96642e25537",
"debug": true,
"polling_period": 30000,
"classes" : {
  "4" : "CPU",
  "619503be0ffb595aeb22222" : "Memory|General"
}
}
]

```

Zabbix.request_options

В этом подразделе можно задать изменения опций для запроса хостов, триггеров и данных.

```

"request_options": {
  "hosts": {
    "filter": {
      ...
    }
  },
  "triggers": {
    "filter": {
      ...
    }
  },
  "data": {
    "filter": {
      ...
    }
  }
}

```

Раздел/Параметр	Описание
zabbix.request_options.hosts	Параметры для запроса хостов.
zabbix.request_options.hosts.filter	Поля фильтров для запроса хостов. Все возможные поля фильтров hosts представлены в документации по Zabbix API.
zabbix.request_options.triggers	Параметры для запроса триггеров.
zabbix.request_options.triggers.filter	Поля фильтров для запроса триггеров. Все возможные поля фильтров triggers представлены в документации по Zabbix API.

Раздел/Параметр	Описание
zabbix.request_options.data	Параметры для запроса данных.
zabbix.request_options.data.filter	Поля фильтров для запроса данных. Все возможные поля фильтров data представлены в документации по Zabbix API.

Управление некоторыми настройками доступно в web-интерфейсе системы:

1. В панели инструментов нажать на имя пользователя и выбрать в меню пункт "Конфигурация".
2. Перейти в раздел "Опции конфигурации".

Изменённые параметры в интерфейсе приоритетнее настроек в конфигурационном файле.

Установленные значения параметров автоматически сохраняются в MongoDB, конфигурационный файл не перезаписывается.

Раздел "Опции конфигурации" окна конфигурации системы состоит из подразделов:

1. "Общие настройки" соответствуют параметрам секции **Server** (Рис. 5):

Общие настройки

Сохранять устаревшие данные от агента

Период обновления истории, мс

Период обновления таблицы состояний, мс

Период проверки устаревания данных от агента, мс

Период обработки пассивных данных от агента, мс

Максимальный размер журнала событий, байты

Срок действия данных от агента по умолчанию, мс

Таймаут погашенной аварии, мс

Рис. 5. Общие настройки

1. "Настройки электронной почты" соответствуют параметрам подсекции **Server.Email** (Рис. 6):

Настройки электронной почты

Выключено

Сервис транспорта

Имя пользователя

Пароль

Адрес отправителя

Адрес получателя

Максимальная длина JSON

Рис. 6. Настройки электронной почты

1. "Настройки пользователей" соответствуют параметрам подсекции **Server.User** (Рис. 7):

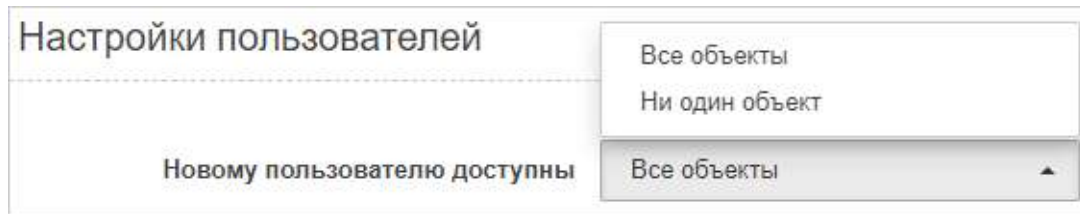


Рис. 7. Настройки пользователей

Просмотреть актуальные настройки сервера возможно при помощи REST API метода:

```
GET /node/api/configuration
```

Пример (bash):

```
login=<...> password=<...> saymon_hostname=<...>
url=https://$saymon_hostname/node/api/configuration
curl -X GET $url -u $login:$password
```

3.1.4. Настройка элементов web-интерфейса

Некоторые элементы web-интерфейса можно настроить в конфигурационном файле клиента `/etc/saymon/saymon-client.yaml`.

WARNING

Настройки конфигурации клиента могут храниться в (в порядке снижения приоритета):

- MongoDB - настройки, измененные через [Rest API](#);
- `/etc/saymon/saymon-client.yaml` - конфигурация клиента в формате yaml;
- `/etc/saymon/saymon-client.yml` - конфигурация клиента в формате yaml (с расширением yml);
- `/etc/saymon/saymon-client.json` - конфигурация клиента в формате json;
- `/etc/saymon/client-config.js` - конфигурация клиента в старом формате;
- `./backend/server/saymon-client.yaml` - конфигурация клиента с настройками по умолчанию.

Примеры настроек приведены для файла `saymon-client.yaml`.

3.1.4.1. Authentication

В этом разделе можно настроить методы аутентификации пользователей.

```
authentication:
  availableMethods:
    - native
    - keycloak
  methodByDefault: keycloak
  services:
    keycloak:
      configUri: keycloak_frontend.json
      legacySupport: false
```

Раздел/Параметр	Описание
<code>authentication.availableMethods</code>	<p>Список доступных для пользователей методов аутентификации:</p> <ul style="list-style-type: none"> • native - вход по логину и паролю непосредственно на сервере Центрального Пульта; • keycloak - вход с использованием внешнего Keycloak-сервера. <p>По умолчанию используется только native.</p>
<code>authentication.methodByDefault</code>	<p>Метод аутентификации, предлагаемый пользователю при отображении формы аутентификации.</p> <p>По умолчанию - native.</p>

Authentication.services

Секция с настройками сервисов аутентификации.

Раздел/Параметр	Описание
<code>Authentication.services.keycloak</code>	Секция с настройками аутентификации через Keycloak.

Раздел/Параметр	Описание
authentication.services.keycloak. configUri	<p>Название файла настроек соединения, сгенерированного Keycloak-сервером для Frontend-клиента.</p> <div style="border: 1px solid orange; padding: 5px; margin-top: 10px;"> <p>WARNING Файл необходимо разместить на сервере Центрального Пульта в каталоге <code>/etc/saymon</code>.</p> </div>
authentication.services.keycloak. legacySupport	<p>Включает поддержку работы с Keycloak-сервером версии ниже 19;</p> <ul style="list-style-type: none"> • <code>false</code> - поддержка работы с Keycloak-сервером версии ниже 19 выключена; • <code>true</code> - поддержка работы с Keycloak-сервером версии ниже 19 включена. <p>По умолчанию - false - работа с Keycloak-сервером версии ниже 19 не поддерживается.</p>

3.1.4.2. Comet

Наряду с REST API, Центральный Пульт использует модель Comet для обмена данными между клиентом и сервером.

```
comet:
  port: 8091
  url:
    - "https://cpult.ru/ws"
    - "http://10.78.5.11:8092"
```

Раздел/Параметр	Описание
comet. port	<p>Номер порта, по которому клиент подключается к Comet-серверу.</p> <p>По умолчанию - 8091.</p>

Раздел/Параметр	Описание
<code>comet.url</code>	<p>Адреса для подключения клиентов к Comet-серверу из разных подсетей.</p> <div style="border: 1px solid orange; padding: 5px; margin-top: 10px;"> <p>WARNING Параметр <code>comet.port</code> игнорируется при использовании параметра <code>comet.url</code>.</p> </div>

3.1.4.3. DefaultClassImage

Позволяет задать стандартное изображение и его размер для объектов определенного класса.

```
defaultClassImage:admin-guide/
  "29":
    img: /images/cloud.svg
    dim:
      - 100
      - 75
```

`DefaultClassImage.<number>`

Идентификатор класса, например, "29".

Раздел/Параметр	Описание
<code>defaultClassImage.<number>.img</code>	Путь до изображения относительно папки, где находится конфигурационный файл.
<code>defaultClassImage.<number>.dim</code>	Размер изображения (в пикселях, [ширина, высота]).

3.1.4.4. DefaultEmbeddableLink

Позволяет встраивать виджеты с информацией об объектах на сторонние ресурсы.

```
defaultEmbeddableLink:
  url: http://saymon.info/
  name: SAYMON
```

Раздел/Параметр	Описание
<code>defaultEmbeddableLink.name</code>	Имя виджета.

Раздел/Параметр	Описание
defaultEmbeddableLink.url	URL, на который ведёт виджет.

3.1.4.5. Documentation

В этом разделе можно настроить ссылки на различную документацию для Центрального Пульта.

```
documentation: {
  main: {
    root: 'https://docs.saymon.tech',
    ru: 'https://docs.cpult.ru
  },
  notificationTemplates: {
    root: 'https://docs.saymon.tech/web-interface/system-configuration-
window/notification-templates.html',
    ru: 'https://docs.cpult.ru/web-interface/system-configuration-
window/notification-templates.html'
  }
},
...
```

Documentation.main

Ссылки на пользовательскую документацию.

Раздел/Параметр	Описание
documentation.main.root	Ссылка на англоязычную документацию.
documentation.main.ru	Ссылка на русскоязычную документацию.

Documentation.notificationTemplates

Ссылки на шаблоны оповещений.

Раздел/Параметр	Описание
documentation.notificationTemplates.root	Ссылка на англоязычные шаблоны оповещений.
documentation.notificationTemplates.ru	Ссылка на русскоязычные шаблоны оповещений.

Documentation.restApi

Ссылки на документацию по REST API.

Раздел/Параметр	Описание
documentation.restApi.root	Ссылка на документацию по REST API.

3.1.4.6. GeoMap

Центральный Пульт позволяет привязывать объекты к их геоположению и просматривать их на карте.

```
geoMap:
  serverName: openstreetmap.org
  initialPosition:
    - 30.341306
    - 59.930089
  initialZoom: 8
```

Раздел/Параметр	Описание
geoMap.initialPosition	Начальная позиция на карте в формате [долгота, широта]. По умолчанию - Россия, Санкт-Петербург.
geoMap.initialZoom	Начальный масштаб карты. По умолчанию - 8 .
geoMap.serverName	Имя сервера, отвечающего за отображения карты. По умолчанию - link:https://www.openstreetmap.org/openstreetmap.org. Также можно использовать локальный OSM-сервер.

3.1.4.7. Grid

В разделе задаются настройки сетки для выравнивания объектов в стандартном виде.

```
grid:
  dim: 20
  color: rgba(128, 128, 128, 0.3)
  border: 4
```

Раздел/Параметр	Описание
grid. border	Максимальное значение отступа границы объекта от границы сетки (в пикселях). По умолчанию - 4 .
grid. color	Цвет сетки в формате RGBA. По умолчанию - rgba(128, 128, 128, 0.3) .
grid. dim	Размер сетки (в пикселях). По умолчанию - 20 .

3.1.4.8. GridView

В разделе задаются параметры отображения объектов в виде сетки.

```
gridView:
  cols: 6
  colWidth: 240
  rowHeight: 90
  gutterSize: 5
```

Раздел/Параметр	Описание
gridView. cols	Количество объектов в строке сетки. При gridView.cols > 0 ширина объектов меняется автоматически. При gridView.cols === 0 число объектов в строке сетки подстраивается под ширину экрана. По умолчанию - 6 .

Раздел/Параметр	Описание
<code>gridView.colWidth</code>	Ширина объектов при <code>gridView.cols === 0</code> (в пикселях). По умолчанию - 240 .
<code>gridView.gutterSize</code>	Величина отступа между объектами (в пикселях). По умолчанию - 5 .
<code>gridView.rowHeight</code>	Высота объектов (в пикселях). По умолчанию - 90 .

3.1.4.9. History

В этом разделе задаются параметры, отвечающие за отображение данных на графиках, а также активируются некоторые дополнительные элементы интерфейса.

```
history:
  slider: false,
  downsampling:
    - period: 1-hours
      dsp: 15s-avg
    - period: 1-days
      dsp: 5m-avg
    - period: 1-weeks
      dsp: 1h-avg
    - dsp: 1d-avg
  modelChangeAnnotations: false
```

Раздел/Параметр	Описание
<code>history.downsampling</code>	Параметры уменьшения размера выборки данных - <code>downsampling</code> .
<code>history.modelChangeAnnotations</code>	Включает смену аннотаций на графиках; <ul style="list-style-type: none"> • <code>false</code> - смена аннотаций выключена, • <code>true</code> - смена аннотаций включена. По умолчанию - <code>false</code> - смена аннотаций выключена.

Раздел/Параметр	Описание
history.slider	<p>Включает ползунок, который используется для задания интервала времени отображения данных;</p> <ul style="list-style-type: none"> • false - ползунок выключен, • true - ползунок включен. <p>По умолчанию - false - ползунок выключен.</p>

3.1.4.10. Incidents

В этом разделе настраиваются дополнительные поля для списков исторических аварий.

```
incidents:
  additionalFields:
    - properties.address,
  additionalFieldNames:
    - IP-address
```

Раздел/Параметр	Описание
incidents.additionalFieldNames	Имена дополнительных полей, отображаемые в интерфейсе.
incidents.additionalFields	Источники данных для дополнительных полей.

3.1.4.11. Languages

Раздел позволяет управлять предустановленными языками web-интерфейса.






```
languages:
  en:
    short: us
    long: English
  ru:
    short: ru
    long: Русский
  it:
    short: it
    long: Italiano
```

Languages.<id>

Идентификатор языка.

Доступны три языка:

- en - английский,
- it - итальянский,
- ru - русский.

Раздел/Параметр	Описание
languages.<id>.long	<p>Полное название языка.</p> <p>Это название отображается в меню текущего пользователя Центрального Пульта при выборе языка интерфейса.</p>
languages.<id>.short	<p>Идентификатор иконки флага.</p> <p>Иконка отображается в меню текущего пользователя Центрального Пульта при выборе языка интерфейса.</p> <p>  - es,  - gb,  - it,  - ru,  - us. </p>

3.1.4.12. Monitoring

В этом разделе настраивается скрывание стандартных мониторинговых сенсоров из интерфейса.

```
monitoring:
  standardTasks:
    - ping
    - snmpGet
```

Раздел/Параметр	Описание
<code>monitoring.standardTasks</code>	Массив идентификаторов отображаемых стандартных сенсоров.

3.1.4.13. Tree

В этом разделе задаются настройки для дерева объектов.

```
tree:
  filterSubmit: true
```

Раздел/Параметр	Описание
<code>tree.filterSubmit</code>	<p>Включает кнопку применения фильтра и отключает его автоматическое применение при вводе символов;</p> <ul style="list-style-type: none"> • <code>false</code> - кнопка выключена, • <code>true</code> - кнопка включена. <p>По умолчанию - false - кнопка выключена.</p>

3.1.4.14. Прочие параметры

Раздел/Параметр	Описание
<code>agentInstallerUrl</code>	Ссылка для скачивания установочного скрипта для агента.
<code>authEnabled</code>	<p>Включает самостоятельную регистрацию для пользователей;</p> <ul style="list-style-type: none"> • <code>false</code> - регистрация выключена, • <code>true</code> - регистрация включена. <p>По умолчанию - false - регистрация выключена.</p>

Раздел/Параметр	Описание
canEditObjectStyle	<p>Включает возможность вручную настраивать CSS-стили для объектов и связей.</p> <p>При включении опции в разделе Параметры появляется подраздел Стили (связей);</p> <ul style="list-style-type: none">• false - настройка стилей выключена,• true - настройка стилей включена. <p>По умолчанию - true - настройка стилей включена.</p>
collapseSections	<p>Задаёт настройки первоначального отображения графиков;</p> <ul style="list-style-type: none">• false - секции закреплены, графики раскрыты,• true - секции откреплены, графики свёрнуты. <p>По умолчанию - true - секции откреплены, графики свёрнуты.</p>
defaultPieChartColors	<p>Цвета секторов по умолчанию для круговой диаграммы.</p> <pre data-bbox="536 1043 1479 1599">defaultPieChartColors : - "#a90329" - "#3276b1" - "#009F3D" - "#9868B8" - "#F2C001" - "#FF742C" - "#90574E" - "#EB77BE" - "#FF9796" - "#AAC6E5" - "#8BDE90" - "#FDB5D0"</pre>

Раздел/Параметр	Описание
disabledSections	<p>Позволяет удалить секции, которые показываются при подробном отображении] объекта в верхней части экрана.</p> <p>Список всех доступных секций:</p> <ul style="list-style-type: none"> • audit-log (История изменений), • config-log (Изменения конфигурации), • documents (Документы), • entity-incident-conditions (Условия генерации аварий), • entity-settings (Параметры [объектов, связей]), • entity-state-conditions (Условия перехода состояний), • history-graph (Графики), • monitoring (Мониторинг), • operations (Операции), • properties (Свойства), • stat (Данные), • stat-rules (Правила формирования данных), • state-history (История состояний), • state-triggers (Действия при смене состояний), • widgets (Виджеты).
enableAnalytics	<p>Включает функционал модуля аналитики;</p> <ul style="list-style-type: none"> • false - функционал выключен, • true - функционал включен. <p>По умолчанию - false - функционал выключен.</p>
enableConditionalIncidents	<p>Включает функционал генерации аварий;</p> <ul style="list-style-type: none"> • false - функционал выключен, • true - функционал включен. <p>По умолчанию - false - функционал выключен.</p>

Раздел/Параметр	Описание
enableSmsTrigger	<p>Включает использование SMS-уведомлений;</p> <ul style="list-style-type: none"> • false - SMS-уведомления выключены, • true - SMS-уведомления включены. <p>По умолчанию - false - SMS-уведомления выключены.</p>
enableSnmpTranslate	<p>Включает трансляцию SNMP OID в текстовый формат;</p> <ul style="list-style-type: none"> • false - трансляция выключена, • true - трансляция включена. <p>По умолчанию - false - трансляция выключена.</p>
enableVoiceCallTrigger	<p>Включает использование голосовых вызовов;</p> <ul style="list-style-type: none"> • false - голосовые вызовы выключены, • true - голосовые вызовы включены. <p>По умолчанию - false - голосовые вызовы выключены.</p>
ForceEmptyStandardView	<p>При выборе объекта, в том числе не содержащего вложенных объектов, Центральный Пульт по умолчанию показывает для него подробную информацию.</p> <p>В параметре задаётся список идентификаторов классов, при выборе пустых объектов которых Центральный Пульт переходит в стандартный вид.</p> <p>По умолчанию:</p> <ul style="list-style-type: none"> • 1 (Root), • 3 (Host), • 13 (Node), • 43 (Dashboard).

Раздел/Параметр	Описание
hideConnectionStatusNotifications	<p>Скрывает оповещения о статусе соединения с Comet-сервером;</p> <ul style="list-style-type: none"> • false - оповещения выводятся в web-интерфейс, • true - оповещения выводятся в консоль. <p>По умолчанию - false - оповещения о статусе соединения с Comet-сервером выводятся в web-интерфейс.</p>
hideResponseErrorNotifications	<p>Скрывает оповещения об ошибках, возникающих при отправке автоматических запросов на сервер;</p> <ul style="list-style-type: none"> • false - оповещения выводятся в web-интерфейс, • true - оповещения скрыты. <p>По умолчанию - false - оповещения выводятся в web-интерфейс.</p> <div data-bbox="536 972 1481 1182" style="border: 1px solid orange; padding: 5px; margin-top: 10px;"> <p>WARNING Ошибки, возникающие при участии пользователя, например сохранение каких-либо данных в формах, будут выводиться в web-интерфейс в любом случае.</p> </div>
millisecondMonitoring	<p>Включает обновление данных каждую миллисекунду;</p> <ul style="list-style-type: none"> • false - обновление выключено, • true - обновление включено. <p>По умолчанию - false - обновление выключено.</p>
pollInterval	<p>Интервал между запросами данных с REST-сервера, если соединение между клиентом и Comet-сервером недоступно (в миллисекундах).</p> <p>По умолчанию - 5000 - 5 секунд.</p>
pollIntervalSocket	<p>Интервал между запросами данных с Comet-сервера, если соединение между клиентом и Comet-сервером доступно (в миллисекундах).</p> <p>По умолчанию - 60000 - 1 минута.</p>

Раздел/Параметр	Описание
title	<p>Заголовок для вкладок, а также имя, отображаемое в левом верхнем углу web-интерфейса.</p> <p>По умолчанию - SAYMON.</p>
themes	<p>Этот параметр отвечает за настройку тем.</p> <p>По умолчанию используется тема Light. Также доступна тема Dark.</p> <p>CSS-файлы с темами находятся в директории <code>\$SAYMON_ROOT/target/client/css</code>.</p> <p><code>\$SAYMON_ROOT</code> - директория, где установлен Центральный Пульт.</p> <pre data-bbox="539 864 1481 1111"> themes: default: name: Light dark: name: Dark </pre>
useNaturalSort	<p>Включает сортировку объектов с помощью библиотеки javascript-natural-sort:</p> <ul style="list-style-type: none"> • <code>false</code> - объекты сортируются с помощью нативных функций браузера, • <code>true</code> - объекты сортируются с помощью библиотеки javascript-natural-sort. <p>По умолчанию - false - объекты сортируются с помощью нативных функций браузера.</p> <div data-bbox="539 1630 1481 1839" style="border: 1px solid #d9534f; padding: 10px; margin-top: 10px;"> <p>CAUTION При установленном <code>useNaturalSort === true</code> возможно замедление работы web-интерфейса при большом количестве объектов в системе (от 100 000).</p> </div>

3.1.5. Служба "saymon-server"

Узнать состояние, запустить, перезапустить и остановить службу сервера можно следующими командами соответственно:

```
sudo service saymon-server status
sudo service saymon-server start
sudo service saymon-server restart
sudo service saymon-server stop
```

3.1.6. Просмотр websocket-нотификаций

Центральный пульт использует гибкий механизм оповещений, который позволяет пользователю оперативно реагировать на возникающие ситуации.

Чтобы просмотреть пришедшие websocket-уведомления, необходимо:

1. Запустить Chrome.
2. Авторизоваться на сервере Центрального Пульта.
3. Перейти по ссылке https://<your_server>/incidents.html?debug=comet
4. Открыть в Chrome Инструменты разработчика.
5. Открыть вкладку Консоль (Console) в Инструментах разработчика.
6. В качестве примера нажать правой кнопкой мыши на строке с аварией и выбрать пункт "Подтвердить".

В консоли отобразится websocket-нотификация.

3.1.7. Увеличение количества обработчиков SNMP-Trap

Для поддержания бесперебойного наблюдения за объектами и оповещения администратора Центральный Пульт использует SNMP-Trap.

Если на сервер поступает большое количество SNMP-Trap, то возможно увеличить количество их обработчиков. Для этого необходимо:

1. Открыть в текстовом редакторе файл настроек акторов:

```
sudo nano /usr/local/saymon/backend/server/actors.json
```

2. Добавить следующую секцию:

```
"snmpTrapMessageHandlerActor": {
  "mode": "forked",
  "clusterSize": 3,
  "onCrash": "respawn"
}
```

3. Перезапустить сервер:

```
sudo service saymon-server restart
```

Для проверки корректности выполненной операции необходимо выполнить команду:

```
ps ax | grep node
```

В выводе должно отображаться число процессов **SnmpTrapMessageHandlerActor**, которое было указано в параметре **clusterSize** выше (в этом примере - 3):

```
5793 ? Rsl 10:23 /usr/bin/nodejs --  
harmony /usr/local/saymon/backend/server/saymon-server.js  
  
5857 ? Sl 3:18 /opt/nodejs/bin/node --  
harmony /usr/local/saymon/backend/server/actors/forked-actor-worker.js  
RestServerActor  
  
5862 ? Sl 0:37 /opt/nodejs/bin/node --  
harmony /usr/local/saymon/backend/server/actors/forked-actor-worker.js  
ResourceServerActor  
  
5867 ? Sl 1:43 /opt/nodejs/bin/node --  
harmony /usr/local/saymon/backend/server/actors/forked-actor-worker.js  
HistoryWriterActor  
  
5872 ? Sl 0:45 /opt/nodejs/bin/node -  
  
harmony /usr/local/saymon/backend/server/actors/forked-actor-worker.js  
SnmpTrapMessageHandlerActor  
  
5877 ? Sl 0:42 /opt/nodejs/bin/node --  
harmony /usr/local/saymon/backend/server/actors/forked-actor-worker.js  
SnmpTrapMessageHandlerActor  
  
5882 ? Sl 0:42 /opt/nodejs/bin/node --  
harmony /usr/local/saymon/backend/server/actors/forked-actor-worker.js  
SnmpTrapMessageHandlerActor
```

3.1.8. Сброс системы к заводским настройкам

Администратор может в любой момент сбросить настройки Центрального Пульта, чтобы восстановить настройки по умолчанию.

Для этого необходимо выполнить следующий скрипт:

```
#!/bin/bash  
#  
#Script deletes instance-specific data from MongoDB, Redis and OpenTSDB.
```

```
# Let's stop SAYMON Server and begin.
service saymon-server stop

mongo saymon --eval 'db.dropDatabase()'

echo flushall | redis-cli

docker stop opentsdb
docker rm opentsdb
docker run -d -p 127.0.0.1:4242:4242 --restart=always --name=opentsdb
rossinno/opentsdb

# Purges logs also.
rm /var/log/saymon/* /var/log/nginx/saymon*

# Forward to new monitoring adventures!
service saymon-server start*
```

3.1.9. Включение функционала генерации аварий

Функционал генерации аварий опционален и выключен по умолчанию. Для его активации необходимо:

1. В секции **Server** конфигурационного файла сервера `/etc/saymon/saymon-server.conf` установить параметру `conditional_incidents_enabled` значение `true`:

```
{
  ...
  "server" : {
    ...
    "conditional_incidents_enabled " : true,
  },
  ...
}
```

NOTE

Подробную информацию о файле `/etc/saymon/saymon-server.conf` см. в подразделе [3.1.3, "Конфигурация сервера"](#).

2. В конфигурационном файле клиента `/etc/saymon/saymon-client.yaml` установить параметру `enableConditionalIncidents` значение `true`:

```
enableConditionalIncidents: true
```

NOTE

Подробную информацию о файле `/etc/saymon/saymon-client.yaml` см. в подразделе [3.1.4, "Настройка элементов web-интерфейса"](#).

3.1.10. Передача аварий в сторонние сервисы

Для передачи аварийных сообщений в системы верхнего уровня необходимо выполнить следующие действия:

1. Включить функционал генерации аварий в секции **Server** конфигурационного файла сервера `/etc/saymon/saymon-server.conf` и там же указать путь к скрипту, который будет отправлять данные:

```
"server": {
  "conditional_incidents_enabled" : true,
  "conditional_incidents_script" :
  "/home/saymon/scripts/send_incident.sh",
  ...
}
```

2. Добавить в MongoDB триггеры по авариям для классов или отдельных объектов:

```
mongo saymon

db.incidentTriggers.insert({ "ownerId" : 1042, "ownerType" : 4, "body" : [
{ "type" : "conditional_incidents_script", "payload" : { }, "states" : [
1,2,4] } ] })
```

где

- `ownerId` - идентификатор сущности, к которой добавляется триггер;
- `ownerType` - тип сущности, для которой добавляется триггер (4 - класс, 1 - объект);
- `states` - уровень критичности аварий, при которых запускается триггер.

В указанном выше примере скрипт триггера будет вызываться при возникновении аварии любой критичности у объектов класса 1042.

В скрипт передается два аргумента:

- `$1` - тело аварии;
- `$2` - свойства объекта, к которому относится авария.

3.1.11. Работа со свойствами аварий

Добавить/изменить или удалить свойства аварий можно с помощью REST API.

При желании можно добавлять в списки активных и исторических аварий дополнительные поля из свойств аварий, значения которых будут отражены в списках. Для этого необходимо указать нужные свойства аварий в разделе **Incidents**

конфигурационного файла клиента `/etc/saymon/saymon-client.yaml`.

```
incidents:
  additionalFields:
    - properties.address,
  additionalFieldNames:
    - IP-address
```

NOTE

Подробную информацию о файле `/etc/saymon/saymon-client.yaml` см. в подразделе 3.1.4, "Настройка элементов веб-интерфейса".

REST API метод для добавления или изменения свойств аварии:

POST `/node/api/incidents/:id/props`

Пример (bash) добавления аварии свойств **foo** со значением **1** и **bar** со значением **second**:

```
login=<...>
password=<...>
saymon_hostname=<...>
incident_id=<...>
url=https://$saymon_hostname/node/api/incidents/$incident_id/props

curl -X POST $url -u $login:$password \
-H "Content-Type: application/json" \
-d @- <<EOF
{
  "foo": 1,
  "bar": "second"
}
EOF
```

REST API метод для удаления свойств аварии:

DELETE `/node/api/incidents/:id/props`

Пример (bash) удаления свойства **foo** аварии:

```
login=<...>
password=<...>
saymon_hostname=<...>
incident_id=<...>
url=https://$saymon_hostname/node/api/incidents/$incident_id/props

curl -X DELETE $url -u $login:$password \
-H "Content-Type: application/json" \
-d @- <<EOF
{
  "foo"
}
```

```
}  
EOF
```

WARNING

Пустой запрос удаляет все свойства аварии.

3.1.12. Перевод SNMP OID в текстовый формат

По умолчанию в результатах проверки SNMP Trap OID отображается в числовом формате.

Для перевода OID в "человекочитаемый" текстовый формат необходимо:

1. В секции **rest_server** конфигурационного файла сервера **/etc/saymon/saymon-server.conf** задать адрес сервера для трансляции OID в текстовый формат в параметре **snmp_mib_url**:

```
{  
  ...  
  "rest_server" : {  
    ...  
    "snmp_mib_url": "http://192.168.1.189:5550",  
  },  
  ...  
}
```

NOTE

Подробную информацию о файле **/etc/saymon/saymon-server.conf** см. в подразделе [3.1.3, "Конфигурация сервера"](#).

2. В конфигурационном файле клиента **/etc/saymon/saymon-client.yaml** установить параметру **enableSnmpTranslate** значение **true**:

```
*enableSnmpTranslate: true*
```

NOTE

Подробную информацию о файле **/etc/saymon/saymon-client.yaml** см. в подразделе [\[настройка_элементов_web-интерфейса\]](#).

3.1.13. Выполнение скриптов при создании и удалении объектов

При создании или удалении объектов того или иного класса система способна выполнять скрипты и передавать им аргументы, настраиваемые через web-интерфейс.

Для применения изменений достаточно обновить страницу в браузере.

Чтобы настроить скрипты, необходимо:

1. Задать каталог для хранения скриптов:

- a. Открыть конфигурационный файл сервера `/etc/saymon/saymon-server.conf` и добавить в него секцию **custom_scripts**:

```
*\{*  
  
*...*  
  
*"custom_scripts " : \{*  
  
*"entity_triggers_path": " /opt/saymon-extensions/entity-triggers-  
path"*  
  
*},*  
  
*...*  
  
*}*
```

NOTE

Подробную информацию о файле `/etc/saymon/saymon-server.conf` см. в подразделе [3.1.3, "Конфигурация сервера"](#).

2. В параметре **entity_triggers_path** указать полный путь до каталога.

WARNING

Поместить исполняемый файл в указанный каталог.

Исполняемые файлы скриптов могут располагаться в любом количестве подпапок. Поиск производится рекурсивно.

3. Задать настройки скрипта в конфигурационном файле.

Параметры скрипта определяются в JSON-файле, именуемом как **имя_исполняемого_файла + .conf**. Располагаться этот файл должен в том же каталоге, что и исполняемый файл.

```
/opt/saymon-extensions/entity-triggers-path/myscript.sh  
/opt/saymon-extensions/entity-triggers-path/myscript.sh.conf
```

NOTE

Возможные поля файла настроек параметров и пример конфигурационного файла представлены в [Приложении Б](#).

3.1.14. Автоматическое обнаружение объектов

Агент сканирует локальную подсеть и отправляет на сервер информацию обо всех IP-адресах, доступных для TCP-соединения либо доступных по PING. Для обнаруженных хостов объекты **класса Host** с соответствующими адресами создаются в Центральном

Пульты и отображаются на карте инфраструктуры. Внутри каждого из этих объектов автоматически создаётся объект **класса Ping**, в котором включается мониторинг с помощью агента, выполнившего автообнаружение, и сенсора **Ping**.

IP-адрес обнаруженного объекта автоматически записывается в свойство **IP**, что можно использовать для перехода по этому адресу в браузере или доступа к объекту по **SSH** или **Telnet** через **контекстное меню**.

Чтобы настроить автообнаружение объектов, необходимо:

1. В конфигурационном файле агента `.../saymon-agent/conf/agent.properties` установить параметру **agent.discoveryEnabled** значение **true**:

```
*agent.discoveryEnabled=true*
```

NOTE

Подробную информацию о файле `.../saymon-agent/conf/agent.properties` см. в подразделе [3.1.1.2.3, "Конфигурация агента"](#).

2. **WARNING:** Указать ID родительского объект для обнаруживаемых объектов в параметре **server.discovery_parent_id** конфигурационного файла сервера `/etc/saymon/saymon-server.conf`:

WARNING

По умолчанию обнаруженные объекты помещаются в корневой объект.

```
...
"server": {
  ...
  "discovery_parent_id": "632aae8b1a687b43378c0657",
  ...
},
...
```

NOTE

Подробную информацию о файле `/etc/saymon/saymon-server.conf` см. в подразделе [3.1.3, "Конфигурация сервера"](#).

3. Перезапустить службы `saymon-agent` и `saymon-server`:

```
sudo service saymon-agent restart
sudo service saymon-server restart
```

3.2. Модуль аналитики

По умолчанию модуль аналитики, используемый для прогнозирования и определения

всплесков, не включен в ISO-образ Центрального Пульта. Чтобы установить модуль аналитики, необходимо:

1. Скачать скрипт [analytics-create-and-run.sh](#) и разместить его в файловой системе на виртуальной машине, где установлен ISO-образ Центрального Пульта.
2. Запустить скрипт командой

sudo ANALYTICS_BRANCH=x.y.z ./analytics-create-and-run.sh

WARNING

x.y.z - версия модуля аналитики, совместимая с вашей версией Центрального Пульта.

Проверить, что контейнер **saymon-analytics** запустился и работает, можно с помощью команды

docker ps -a

3.2.1. Подключение модуля аналитики к Центральному Пульту

Чтобы подключить модуль аналитики, потребуется внести изменения в конфигурацию Центрального Пульта:

1. В секции **Server** конфигурационного файла сервера **/etc/saymon/saymon-server.conf** добавить или изменить следующие параметры:

```
{
  ...
  "server" : {
    ...
    "analytics_enabled": true,
    "analytics_processes": 2 # количество логических ядер для аналитики
  },
  ...
}
```

NOTE

*Подробную информацию о файле **/etc/saymon/saymon-server.conf** см. в подразделе [3.1.3, "Конфигурация сервера"](#).*

2. В конфигурационном файле клиента **/etc/saymon/saymon-client.yaml** установить параметру **enableAnalytics** значение **true**:

enableConditionalIncidents: true,

NOTE

*Подробную информацию о файле **/etc/saymon/saymon-client.yaml** см. в подразделе [3.1.4, "Настройка элементов веб-интерфейса"](#).*

3. Перезапустить сервер Центрального Пульта:

```
sudo service saymon-server restart
```

3.2.2. Мониторинг работы модуля аналитики

Чтобы мониторить работу модуля аналитики можно использовать логи контейнера **saymon-analytics**. Для этого необходимо:

1. Подключиться к контейнеру командой

```
docker exec -ti saymon-analytics bash
```

2. Открыть файл логов модуля командой

```
tail -f /opt/analytics/analytics.log
```

В логах модуля содержится информация о:

- настройках, с которыми запущен модуль (слушающий сокет и количество задействованных в пуле обработчика процессов),
- обрабатываемых метриках,
- результатах обработки метрик,
- ошибках, возникающих в процессе работы модуля.

3.3. Управление сенсорами мониторинга

Системой предусмотрена возможность скрывать в интерфейсе неиспользуемые мониторинговые сенсоры, а также настраивать пользовательские сенсоры и внедрять их в интерфейс.

3.3.1. Скрытие неиспользуемых сенсоров

Для того, чтобы неиспользуемые сенсоры не отображались в пользовательском интерфейсе, необходимо:

1. Открыть конфигурационный файл клиента **/etc/saymon/saymon-client.yaml** и добавить в него секцию **monitoring**:

```
monitoring:  
standardTasks:  
  - ping  
  - snmpGet
```

NOTE

*Подробную информацию о файле **/etc/saymon/saymon-client.yaml** см. в*

2. В переменной **monitoring.standardTasks** указать идентификаторы сенсоров, отображаемых в web-интерфейсе.

WARNING

Идентификаторы сенсоров чувствительны к регистру символов.

Список идентификаторов стандартных сенсоров:

- binaryProtocol (бинарный протокол),
- configFile (конфигурационный файл / директория),
- databaseQuery (запрос в базу данных),
- exec (выполнение программы / скрипта),
- external (внешняя проверка - записанные через API данные),
- ftp (FTP-сенсор),
- httpRequest (HTTP-запрос),
- jmx (jmx-сенсор),
- localNetworkPort (локальный порт),
- mqtt (mqtt-сенсор),
- ping (пинг-сенсор),
- processInfo (процесс по имени),
- remoteNetworkPort (удалённый порт),
- snmpGet (SNMP Get-сенсор),
- snmpTrap (SNMP Trap-сенсор),
- wmi (WMI-сенсор).

3.3.2. Настройка пользовательских сенсоров

В качестве сенсора можно использовать собственноручно написанный скрипт и передавать ему в качестве аргументов значения параметров через удобный web-интерфейс.

Для применения изменений достаточно обновить страницу в браузере.

Чтобы сделать пользовательский скрипт мониторинговым сенсором, необходимо:

1. Задать каталог для хранения пользовательских сенсоров:
 - а. Открыть конфигурационный файл сервера **/etc/saymon/saymon-server.conf** и добавить в него секцию **monitoring**:

```

{
  ...
  "monitoring": {
    "custom_tasks_path": "/opt/saymon-agent/custom_tasks"
  },
  ...
}

```

NOTE

Подробную информацию о файле `/etc/saymon/saymon-server.conf` см. в подразделе [3.1.3, "Конфигурация сервера"](#).

- В параметре **monitoring.custom_tasks_path** указать полный путь до каталога.
- Поместить исполняемый файл в указанный каталог.

WARNING

Исполняемые файлы пользовательских сенсоров могут располагаться в любом количестве подпапок. Поиск производится рекурсивно.

- Проверить идентификатор пользовательского сенсора. Идентификатор сенсора создаётся автоматически и эквивалентен имени исполняемого файла с относительным путём (если есть). Например: **utils/network/ping**.

WARNING

Идентификатор пользовательского сенсора не должен совпадать с идентификатором стандартного, иначе пользовательский сенсор будет проигнорирован.

NOTE

С перечнем идентификаторов стандартных сенсоров можно ознакомиться в подразделе [3.3.1, "Скрытие неиспользуемых сенсоров"](#).

- Задать настройки сенсора в конфигурационном файле.

Параметры сенсора определяются в JSON-файле, именуемом как **имя_исполняемого_файла + .conf**. Располагаться этот файл должен в том же каталоге, что и исполняемый файл.

Пример:

```
/opt/saymon-agent/custom_tasks/mytask.sh
```

```
/opt/saymon-agent/custom_tasks/mytask.sh.conf
```

NOTE

Возможные поля файла настроек параметров и пример конфигурационного файла представлены в [Приложении Б](#).

WARNING

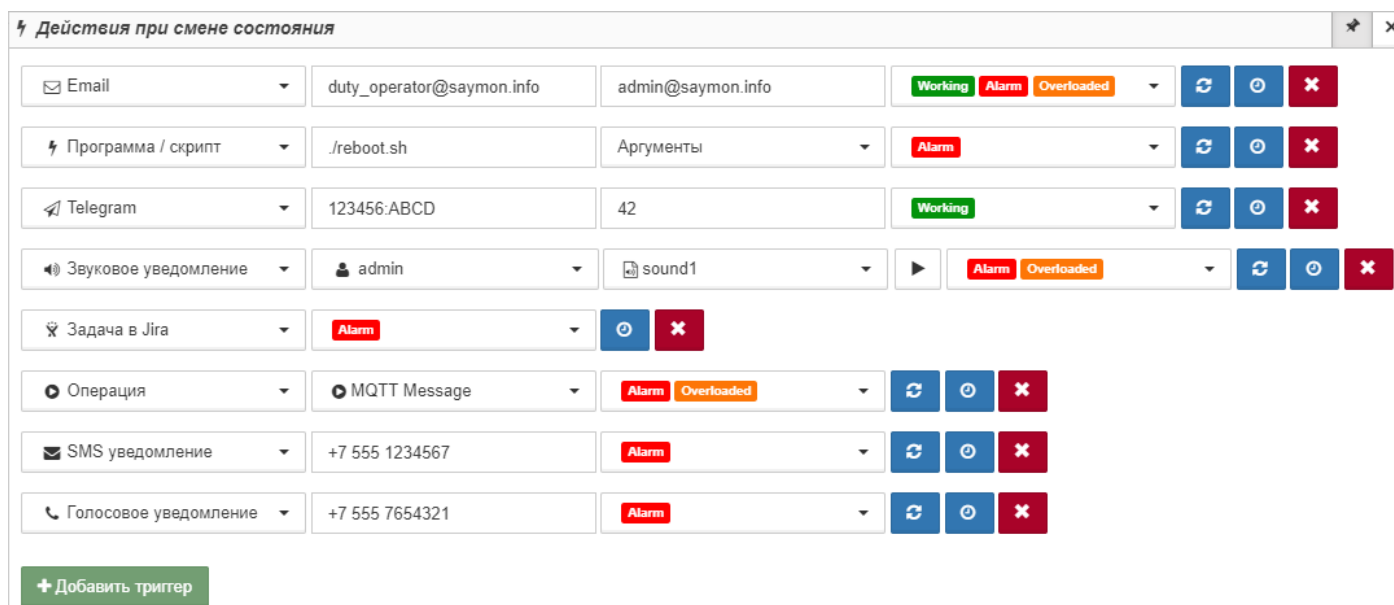
Пользовательские сенсоры выполняются агентами локально, поэтому необходимо скопировать исполняемые файлы в одну и ту же папку на все хосты, на которых предполагается их запускать. Конфигурационные файлы достаточно разместить на хостах с сервером системы.

3.4. Настройка уведомлений

При переходе объектов в определенные состояния система может:

- отправлять email-уведомления,
- автоматически запускать программу или скрипт с параметрами,
- отправлять сообщения в Telegram,
- показывать визуальное уведомление в браузере, сопровождающееся звуком,
- создавать задачи в JIRA,
- запускать операции,
- отправлять SMS,
- совершать голосовые вызовы.

Уведомления настраиваются в виде подробной информации об объекте в секции "Действия при смене состояния" (Рис. 8):



Действие	Параметры	Состояние	Кнопки
Email	duty_operator@saymon.info, admin@saymon.info	Working, Alarm, Overloaded	↻, ⏸, ✖
Программа / скрипт	/reboot.sh, Аргументы	Alarm	↻, ⏸, ✖
Telegram	123456:ABCD, 42	Working	↻, ⏸, ✖
Звуковое уведомление	admin, sound1	Alarm, Overloaded	↻, ⏸, ✖
Задача в Jira	Alarm		⏸, ✖
Операция	MQTT Message	Alarm, Overloaded	↻, ⏸, ✖
SMS уведомление	+7 555 1234567	Alarm	↻, ⏸, ✖
Голосовое уведомление	+7 555 7654321	Alarm	↻, ⏸, ✖

+ Добавить триггер

Рис. 8. Действия при смене состояния

3.4.1. Активация функционала отправки SMS и голосовых вызовов

Для активации функционала необходимо выполнить описанные ниже действия:

1. В файле конфигурации клиента `/etc/saymon/saymon-client.yaml` установить

параметрам `enableSmsTrigger` и `enableVoiceCallTrigger` значение `true`:

```
enableVoiceCallTrigger: true
enableSmsTrigger: true
```

NOTE

Подробную информацию о файле `/etc/saymon/saymon-client.yaml` см. в подразделе 3.1.4, "Настройка элементов web-интерфейса".

- В секции **Server** конфигурационного файла сервера `/etc/saymon/saymon-server.conf` задать путь до скриптов, осуществляющих отправку SMS-уведомлений и голосовые вызовы:

```
{
  ...
  "server" : {
    ...
    "sms_script": "путь до скрипта, отправляющего sms-уведомления",
    "voice_call_script" : "путь до скрипта, осуществляющего голосовые
вызовы",
    ...
  },
  ...
}
```

NOTE

Подробную информацию о файле `/etc/saymon/saymon-server.conf` см. в подразделе 3.1.3, "Конфигурация сервера".

- Для применения изменений перезапустить службу "saymon-server":

```
sudo service saymon-server restart
```

3.4.2. Отправка почтовых уведомлений

Для настройки отправки почтовых уведомлений требуется настроить необходимые параметры доступа к почтовому серверу в подсекции **Email** секции **Server** файла `/etc/saymon/saymon-server.conf`.

NOTE

Подробную информацию о файле `/etc/saymon/saymon-server.conf` см. в подразделе 3.1.3, "Конфигурация сервера".

Пример настройки подсекции **Email**:

```
"email" : {
  "disabled" : false,
  "fields" : {
```

```
    "from" : "saymon@saas.saymon.info"
  },
  "max_json_length" : 1000,
  "transport" : {
    "auth" {
      "user" : "saymon@saas.saymon.info",
      "pass" : "P@ssw0rd"
    },
    "host" : "smtp.gmail.com",
    "port" : 465,
    "secure" : true
  },
}
```

3.4.3. Настройка уведомлений в Telegram

Для настройки отправки уведомлений с помощью Telegram необходимо:

1. Задать бота, от которого будут поступать уведомления:
 - a. Создать нового бота:
 - i. Найти в приложении Telegram контакт **BotFather**.
 - ii. Отправить ему сообщение **/newbot**.
 - iii. Задать боту отображаемое имя (**name**, позже возможно изменить).
 - iv. Задать боту уникальное имя (**username**, изменить будет невозможно).
 - v. Скопировать токен бота вида
210979209 : AAFfT2mt3oW4EK1gYqE_d30jAJSIRLSrAL.
 - vi. Отправить контакту **BotFather** сообщение **/setprivacy**.
 - vii. Выбрать созданного бота по его **username**.
 - viii. Выбрать опцию **Disable**.
 - b. Использовать существующего бота:
 - i. Найти в приложении Telegram контакт **BotFather**.
 - ii. Отправить ему сообщение **/mybots**.
 - iii. Выбрать нужного бота.
 - iv. Выбрать опцию **API Token**.
 - v. Скопировать токен бота вида
210979209 : AAFfT2mt3oW4EK1gYqE_d30jAJSIRLSrAL.
2. Настроить канал, чат или группу:
 - a. Создать приватный канал (рекомендуется):
 - i. В приложении Telegram создать новый канал.

- ii. Открыть настройки канала, добавить бота в список администраторов.
- iii. Отправить сообщение в канал.
- iv. Перейти по ссылке в любом web-браузере, вставив в неё токен своего бота (без пробелов и знаков <>):

```
https://api.telegram.org/bot<токен_бота>/getUpdates
```

- v. Найти текст со словами **chat** и **id**, например

```
... "channel_post": {"message_id": 4, "chat": {"id": -1001156346945  
, "title": "SAYMON" ... } }
```

Здесь **-1001156346945** - искомый ID канала.

- b. Создать чат:

- i. Отправить боту любое сообщение.
- ii. Перейти по ссылке в любом web-браузере, вставив в неё токен своего бота (без пробелов и знаков <>):

```
https://api.telegram.org/bot<токен_бота>/getUpdates
```

- iii. Найти текст со словами **chat** и **id**, например

```
... "K", "chat": {"id": 121399918, "first_ ...
```

Здесь **121399918** - искомый ID чата.

- c. Настроить группу:

- i. Добавить бота в группу.
- ii. Отправить боту в группу любое сообщение, начав его со знака @.
- iii. Перейти по ссылке в любом web-браузере, вставив в неё токен своего бота (без пробелов и знаков <>):

```
https://api.telegram.org/bot<токен_бота>/getUpdates
```

- iv. Найти текст со словами **chat** и **id**, например

```
... "K", "chat": {"id": -209194473, "first_ ...
```

Здесь **-209194473** - искомый ID группы.

3. В настройках Telegram-уведомлений ввести токен (ID) бота и ID канала/чата/группы в соответствующие поля.

3.5. Интеграция со сторонними системами

Система предусматривает возможность импортировать данные с других мониторинговых платформ и экспортировать метрики в платформы для визуализации данных.

3.5.1. Интеграция с Zabbix

Для импорта данных из Zabbix необходимо:

1. Проверить доступ от сервера Центрального Пульта до сервера Zabbix, выполнив команду (в одну строку), заменив <http://192.168.1.215> на адрес файла `api_jsonrpc.php` вашей инсталляции Zabbix:

```
curl -H "Content-Type: application/json" -X POST -d
'\{"jsonrpc": "2.0", "method": "apiinfo.version", "id": 1, "auth": null, "params"
\{\}\}' http://192.168.1.215/zabbix/api_jsonrpc.php
```

При наличии доступа сервер Zabbix ответит похожим сообщением:

```
\{"jsonrpc": "2.0", "result": "3.2.1", "id": 1}
```

2. Добавить в Zabbix пользователя с правами на чтение необходимых хостов или групп хостов.
3. На сервере системы создать объект, в который будут импортироваться данные о хостах из Zabbix, и запомнить его ID.
4. Открыть конфигурационный файл сервера `/etc/saymon/saymon-server.conf` и добавить в него секцию **zabbix**:

```
"zabbix": [
  {
    "url" : "http://192.168.1.215/zabbix/api_jsonrpc.php",
    "user" : "saymon",
    "password" : "saymon_user_password",
    "parent_id" : "58b586d5c3a2f96642e25537"
  },
  ...
]
```

где:

- **url** - адрес файла `api_jsonrpc.php` вашей инсталляции Zabbix, который был использован на шаге 1, например, <http://192.168.1.215/zabbix/>

`api_jsonrpc.php`;

- **user** - имя пользователя, добавленного в Zabbix на шаге 2;
- **password** - пароль пользователя, добавленного в Zabbix на шаге 2;
- **parent_id** - ID объекта, созданного на шаге 3.

NOTE

Подробную информацию о файле `/etc/saymon/saymon-server.conf` см. в подразделе [3.1.3, "Конфигурация сервера"](#).

5. Для каждого дополнительного сервера Zabbix добавить новый JSON-документ с его данными в массив:

```
"zabbix": [  
  {  
    "url": "http://192.168.1.215/zabbix/api_jsonrpc.php",  
    "user": "saymon",  
    "password": "saymon_user_password",  
    "parent_id": "58b586d5c3a2f96642e12345"  
  },  
  {  
    "url": "http://192.168.1.180:8020/zabbix/api_jsonrpc.php",  
    "user": "saymon",  
    "password": "saymon_user_password",  
  }  
]
```

Для применения изменений необходимо перезапустить службу **saymon-server**:

```
sudo service saymon-server restart
```

По умолчанию данные на сервере системы обновляются каждые 2 минуты. Для изменения периода обновления данных необходимо добавить в конфигурационный файл сервера `/etc/saymon/saymon-server.conf` параметр **polling_period** со значением периода обновления данных в миллисекундах:

```
"zabbix": [  
  {  
    "url": "http://192.168.1.215/zabbix/api_jsonrpc.php",  
    "user": "saymon",  
    "password": "saymon_user_password",  
    "parent_id": "58b586d5c3a2f96642e12345" "polling_period": 30000  
  }  
]
```

Возможно изменять опции для запросов

- хостов (hosts),
- триггеров (triggers),
- NOTE: данных (data).

NOTE	<p>Все возможные поля фильтров представлены в документации по Zabbix API:</p> <ul style="list-style-type: none">• <code>hosts</code>,• <code>triggers</code>,• <code>data</code>.
-------------	---

```
"zabbix" : [  
  {  
    ...  
    "request_options": {  
      "hosts": {  
        "filter": {  
          ...  
        }  
      },  
      "triggers": {  
        "filter": {  
          ...  
        }  
      },  
      "data": {  
        "filter": {  
          ...  
        }  
      }  
    }  
  }  
  ...  
]
```

Чтобы убрать фильтр из опции запроса, необходимо указать параметру **filter** значение **null**.

Пример (убрать поле **filter** из опций запроса триггеров Zabbix):

```
"zabbix" : [  
  {  
    ...  
    "triggers": {  
      "filter": null  
    }  
  }  
  ...  
]
```

Обнаруженным объектам по умолчанию присваивается класс **Info**. Чтобы присвоить новым объектам другие классы, необходимо задать массив **zabbix.classes** и указать в нём соответствия идентификаторов классов и имён объектов:

```
"zabbix" : [{  
  ...  
    "classes" : {  
      "4" : "CPU",  
      "619503be0ffb595aebe22222" : "Memory|General"  
    },  
  ...  
}]
```

CAUTION

Порядок применения правил привязки не гарантируется.

Если назначить двум разным классам одинаковые регулярные выражения или использовать выражение "." (любой текст), то нет гарантии, что сначала будет применено правило, указанное выше в списке.*

Для каждого Zabbix-подключения массив настраивается индивидуально. Регулярное выражение по имени объекта нечувствительно к регистру символов.

CAUTION

При необходимости связать класс с несколькими именами необходимо указать все имена в одном регулярном выражении с использованием оператора ИЛИ, в противном случае применится только одно соответствие.

Пример:

```
"619503be0ffb595aebe22222" : "Memory|General"
```

Чтобы выводить параметры и результаты запросов в лог, необходимо задать параметр **zabbix.debug** со значением **true**:

```
"zabbix" : [{  
  ...  
    "debug" : true,  
  ...  
}]
```

3.5.2. Интеграция с Grafana

WARNING

Все действия, связанные с интеграцией, осуществляются в интерфейсе Grafana.

Для экспорта метрик в Grafana необходимо:

1. Авторизоваться в Grafana.
2. С главной страницы рабочего пространства перейти в **Configurations - Data Sources** через боковое меню и добавить data source (источник данных) нажатием на соответствующую кнопку.

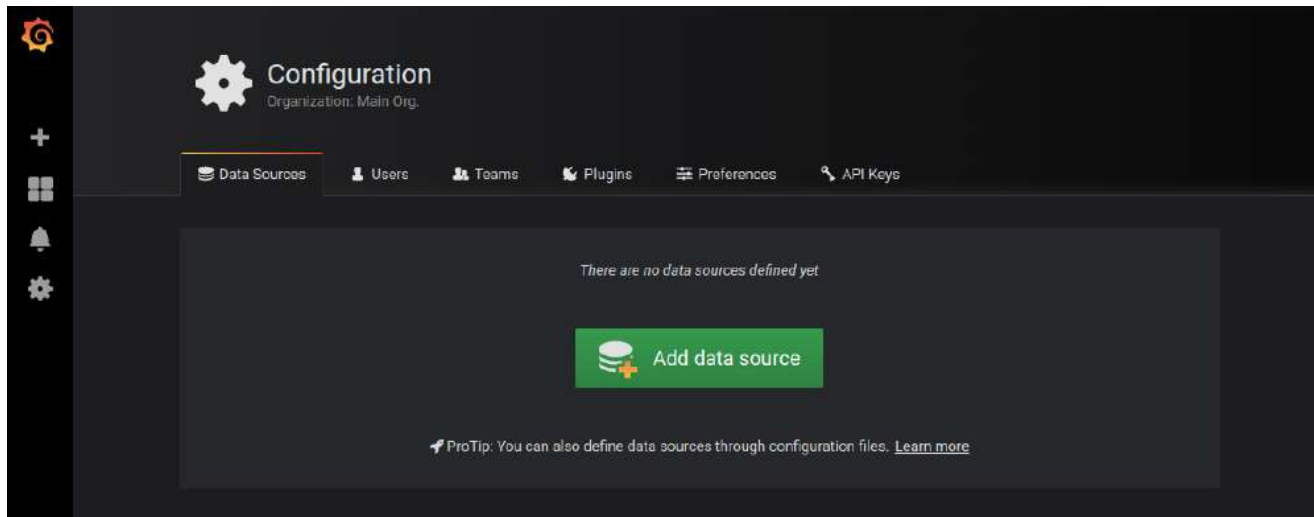


Рис. 9. Добавление источника данных в Grafana

3. В открывшемся окне типов источников данных выбрать SAYMON.

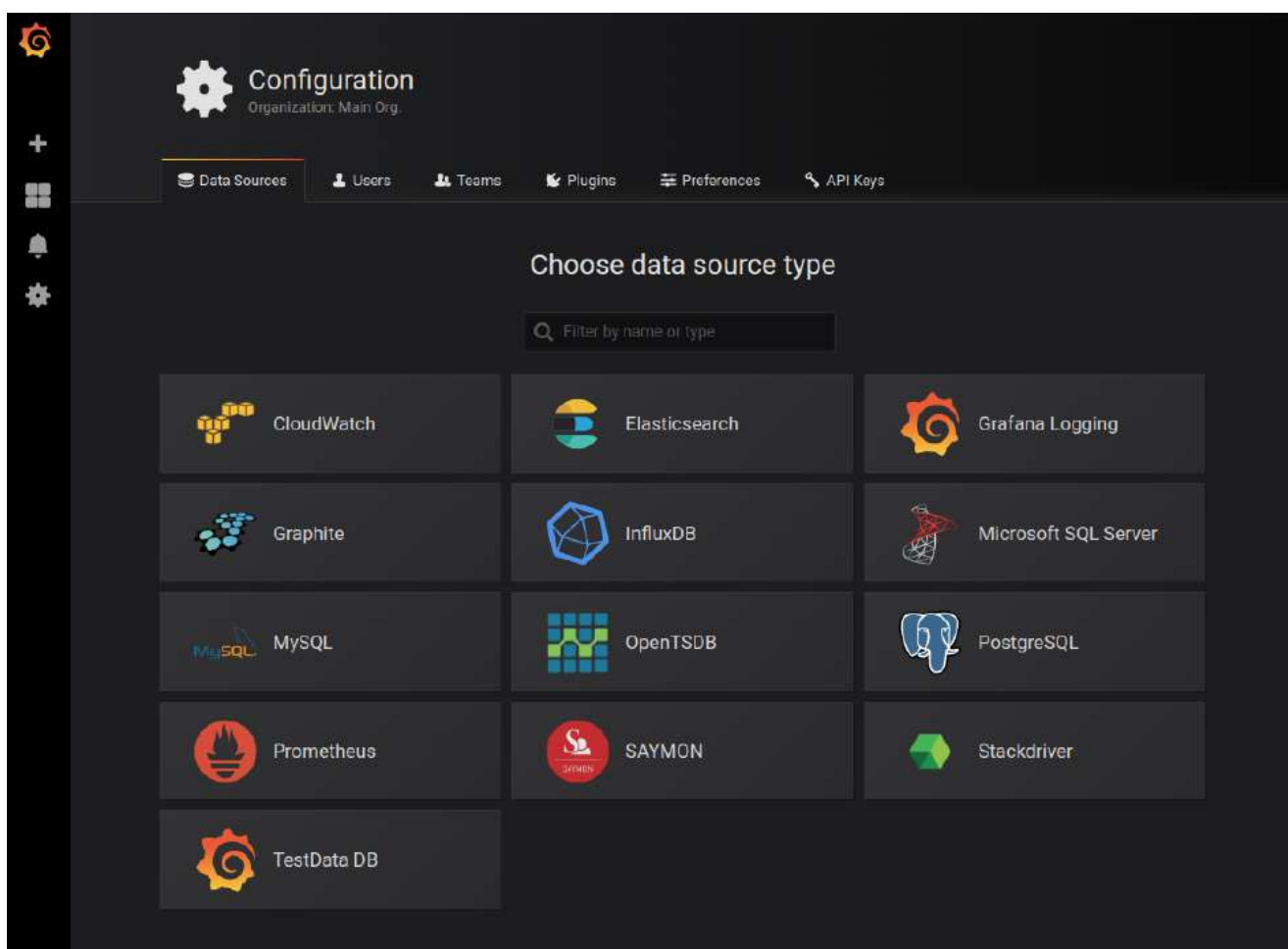


Рис. 10. Выбор источника данных в Grafana

4. В окне настроек необходимо указать:

- имя источника данных;
- URL-адрес, откуда будут собираться данные;
- режим доступа, как будут обрабатываться запросы к источнику данных: **server** или **browser**;
- имена cookies-файлов, которые должны быть перенаправлены в источник данных;
- тип аутентификации;
- логин и пароль, используемые в Центральном Пульте.

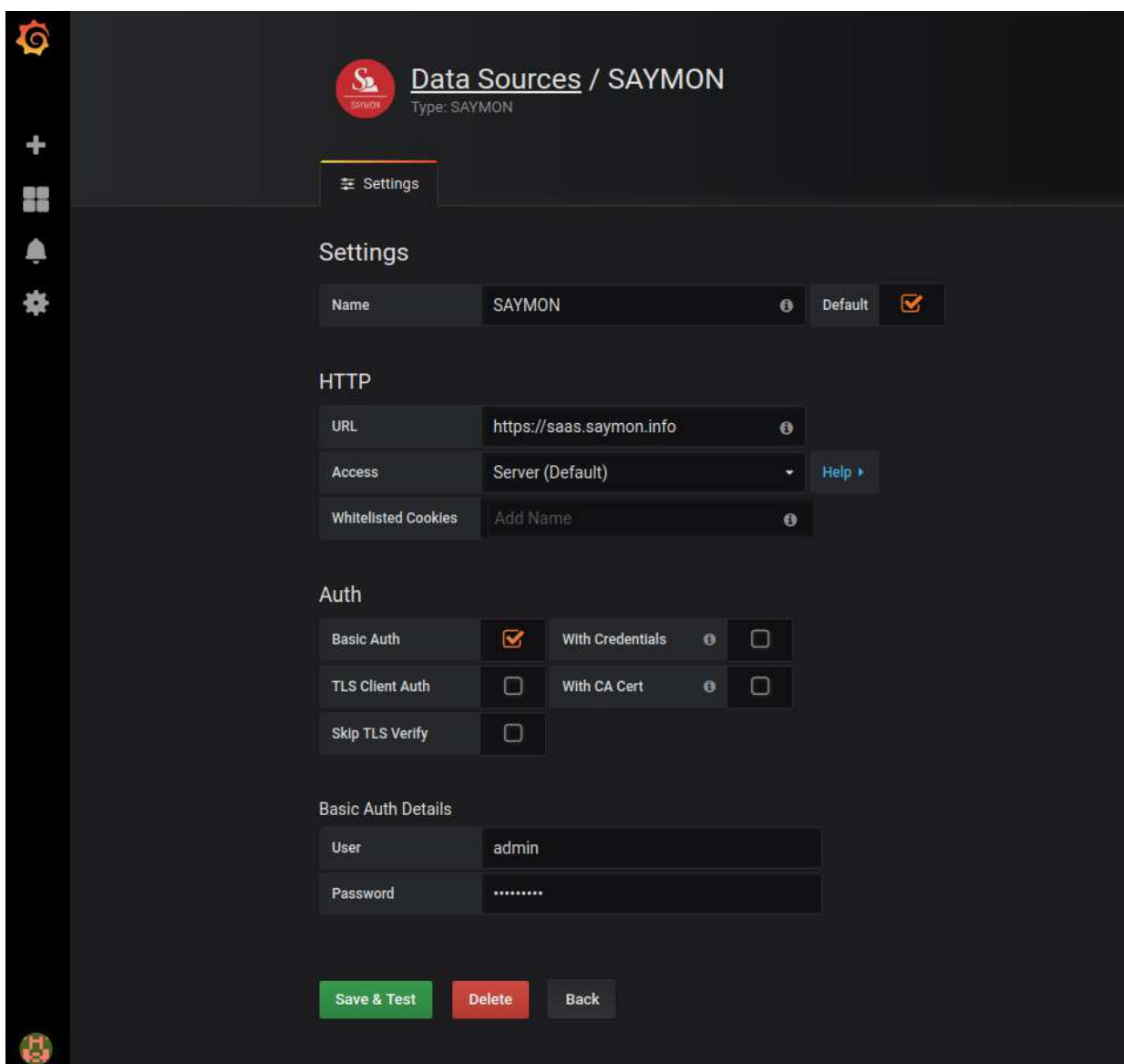


Рис. 11. Настройка источника данных в Grafana

5. Через боковое меню перейти в окно управления дашбордами (**dashboards - manage**) и приступить к созданию нового дашборда нажатием соответствующей кнопки.

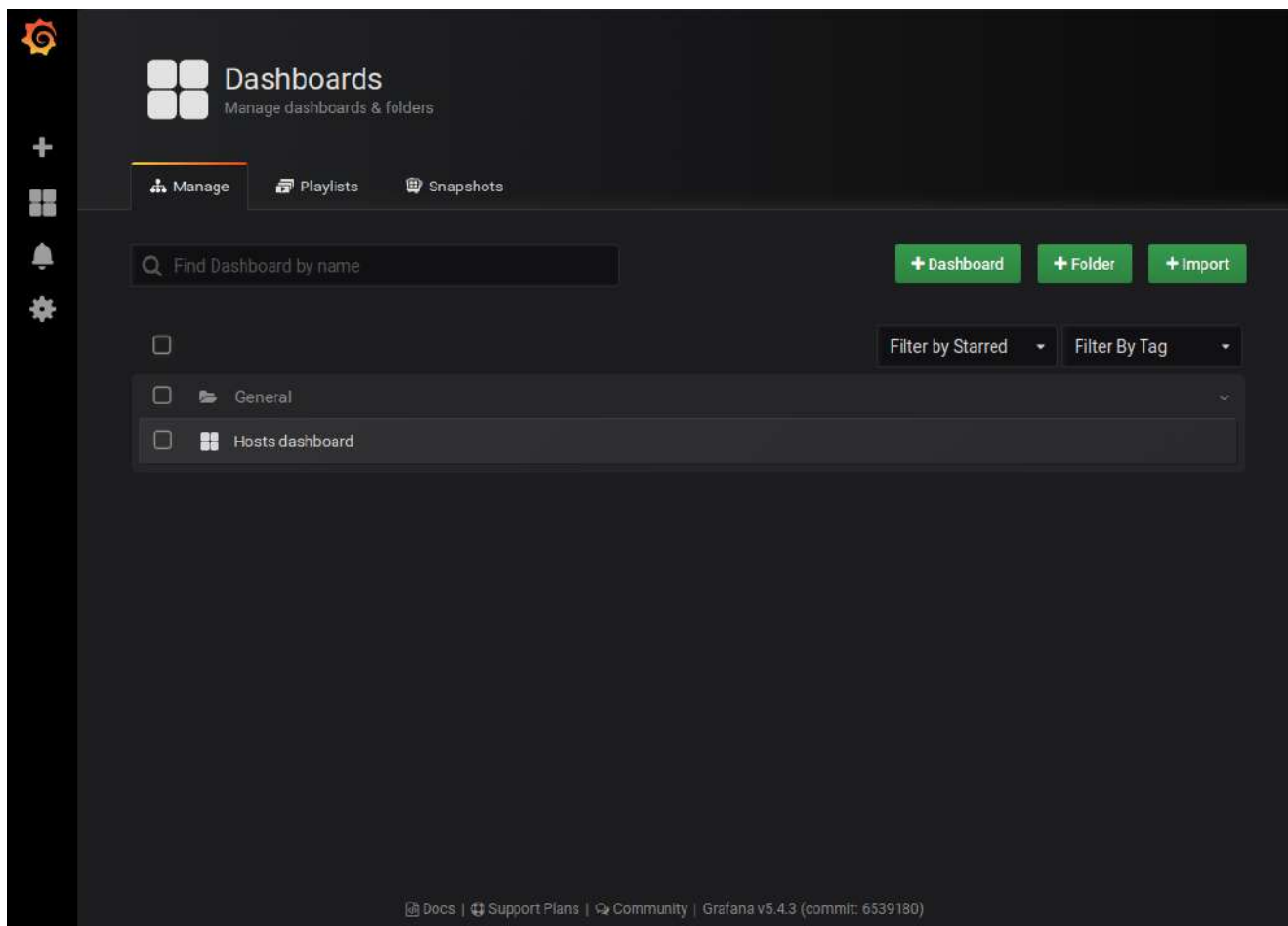


Рис. 12. Создание нового дашборда в Grafana

6. Выбрать тип отображения дашборда. Например, график.

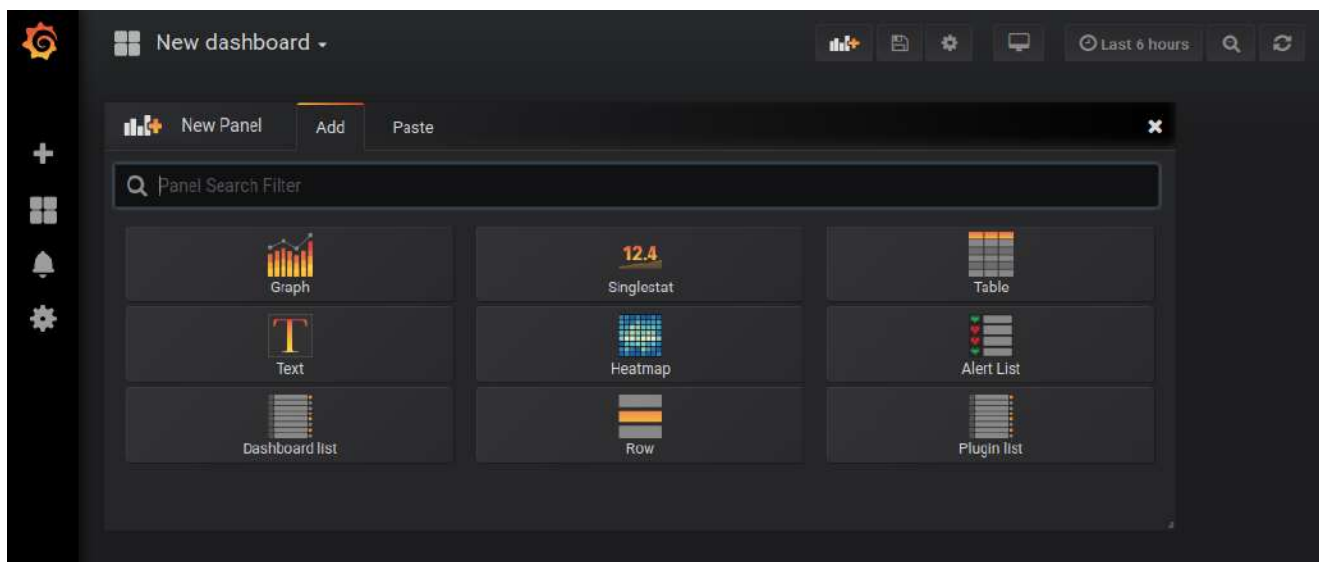


Рис. 13. Выбор типа отображения дашборда в Grafana

7. Открыть контекстное меню нажатием на стрелку возле заголовка панели и перейти в режим редактирования.

8. В окне редактирования перейти во вкладку **Metrics** и добавить объекты, к которым у пользователя есть доступ в Центральном Пульте. Указать:

- имя или ID объекта;

- метрику, которую необходимо отобразить на графике;
- отображать ли путь до объекта.

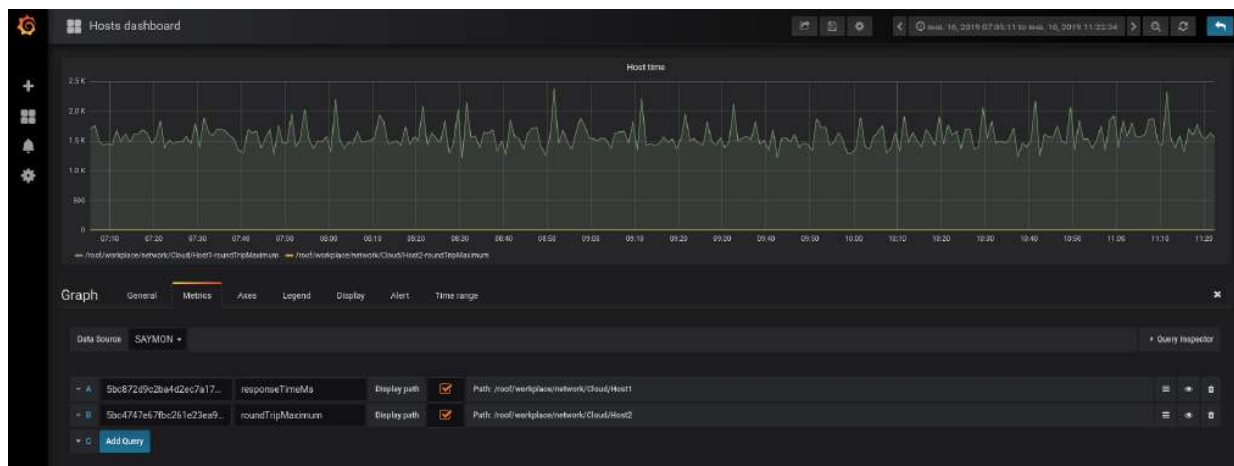


Рис. 14. Отображение дашборда в Grafana

9. Настроить временной период, за который данные отображаются на графике, и интервал их обновления.

3.5.3. Интеграция с Keycloak

Для пользователей Центрального Пульта предусмотрена возможность авторизации посредством сервера Keycloak.

Клиенты в Keycloak — это параметры соединения с Keycloak-сервером для внешних приложений, которые авторизуют пользователей.

Главной особенностью Backend-клиента является наличие собственной аутентификации: такой клиент получает конфиденциальный доступ по секретному ключу.

Главной особенностью Frontend-клиента является отсутствие собственной аутентификации: данный клиент получает публичный доступ.

3.5.3.1. Настройка сервера

Для настройки взаимодействия между сервером Центрального Пульта и внешним сервером Keycloak необходимо получить хотя бы один из параметров:

- Realm Certificate, которым сервер Keycloak подписывает пользовательские токены,
- файл настроек соединения для Backend-клиента.

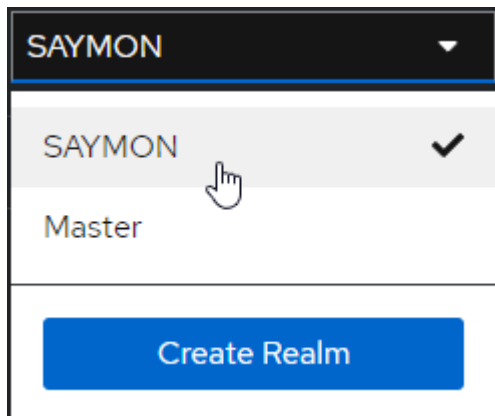
WARNING

Настройка клиента и получение параметров осуществляются в консоли администратора Keycloak.

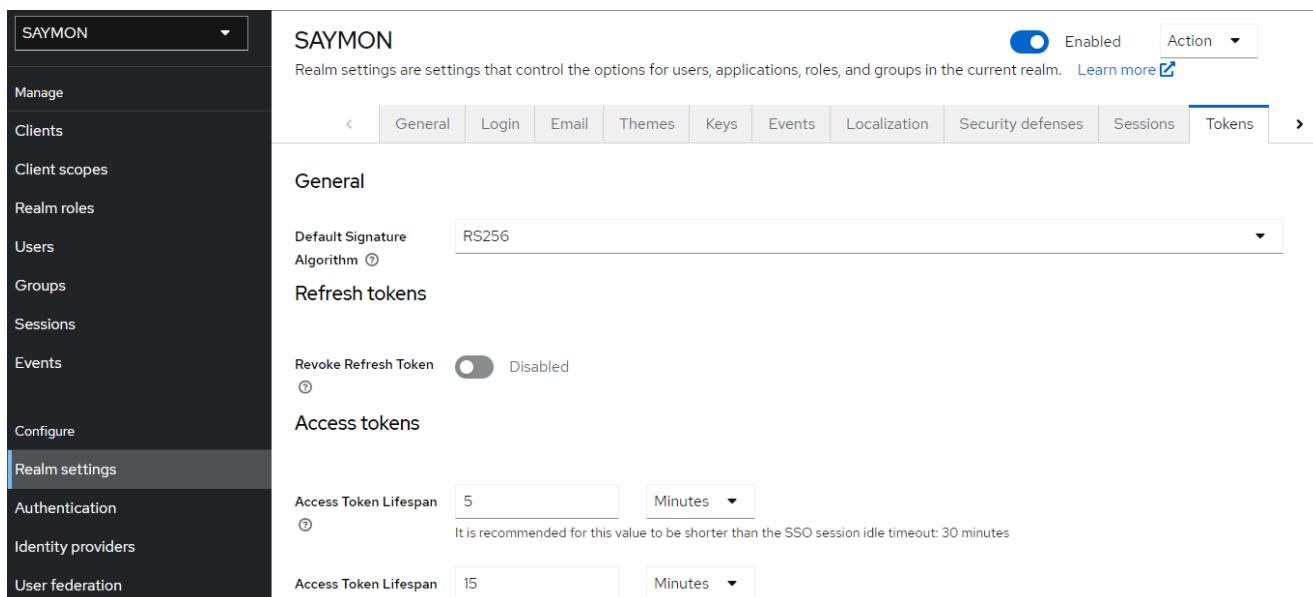
Realm Certificate

Для получения Realm Certificate необходимо:

1. Выбрать требуемую область (Realm) из списка (здесь - **SAYMON**):



2. В секции **Realm settings** на вкладке **Tokens** уточнить алгоритм подписи токенов в поле **Default Signature Algorithm** (здесь - **RS256**):



3. На вкладке **Keys** найти сертификат для данного алгоритма:

SAYMON Enabled Action

Realm settings are settings that control the options for users, applications, roles, and groups in the current realm. [Learn more](#)

General Login Email Themes **Keys** Events Localization Security defenses Sessions Tokens Cli

Keys list Providers

Active keys Search key 1-4

Algorithm	Type	Kid	Provider	Public keys
RSA-OAEP	RSA	jdy8KvSkkKUZMOugtLx-cRhANKfdkw_sh5VZTI-pVNU	rsa-enc-generated	Public key Certificate
HS256	OCT	b93d663a-6612-45f7-806d-915a445b5028	hmac-generated	
RS256	RSA	nJYfREgZ32gezKevCDNeeYNvNeW5_0XJKZ4nBcwfds	rsa-generated	Public key Certificate

4. Кнопка [Certificate](#) в соответствующей строке открывает искомый сертификат:

Certificate X

```
MIIcmzCCAYMCCBgGDnVPsIDANBgkqhkiG9w0BAQsFADARMQ8wDQYDVQQDDAZzYXltb24wHhcNMjIwMDAwOTI2WhcNMjIwMDAwMTAxMTA2WjARMQ8wDQYDVQQDDAZzYXltb24wggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQChBxZpaLIGVro2h5UDApZfi3Q7nGHOo4LSnloQC1Mi+Nb3P4vxY6X/9aluA1/p6eA5gaSBR1DVjH0iLJpEcxwc5kv/aH77IpV+nwHOYkPgjONpjhNMajYLSJ7mH9PBKbfKYe4hAaFOUtIK5kNX9agJn0T1Jgk1/0Dv01QtoWnSu/3E3qelkNjp5D5W2IEU+eoj7M5qEsSykQJs48AcrHcFnYNf3ey3G1OatLufGIrly6tPYAI0tWbXxYV0HepUqV2acMAK+L8EdAT9t/Ar4KQ2y9HU0rfHwgid5r5vjACfAVaX/nzlgWS9ZYDF8ISgfvHYX8nMlpgiiV0w3MQHkJAgMBAAEwDQYJKqZIhvcNAQELBQADggEBAISHUKRIGVKkd6vEozmoy5TdeC2FXkrAgRhxx646Yv4ewXc7a3Er7gJBSR18Vbg1D8NzkOurkGE8rqGVbfkw80CMkjqaFj3v4NVD22eCrCSV5obh3JFbqmarMlj9/nzHiEJ/N4w5WTTenUxb9/3ZyYlXpparrF2ZAbnNijr3ytDdx/LVjc7Gmp0MO7hh1moch8H9tl1mnRLTqs9IWNSMrOagibaD+Em1cbCmMOLINzB2s4PQfixIDX2s6ZLyeJ+229As9ReBXA7qQUp8M9czV+aRQdqemmhYoygVbi2/6FdxZ576Ez47GkPTmN5et2abihbSreY6SBOPsfx2etqku5E=
```

[Close](#) [Cancel](#)

Данные рекомендуется сохранить в файл **keycloak_realm_certificate**.

Backend-клиент

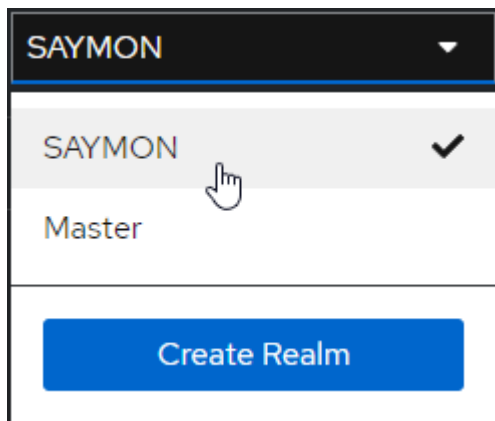
Клиенты - это параметры соединения с Keycloak-сервером для внешних приложений,

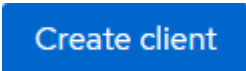
которые авторизуют пользователей.

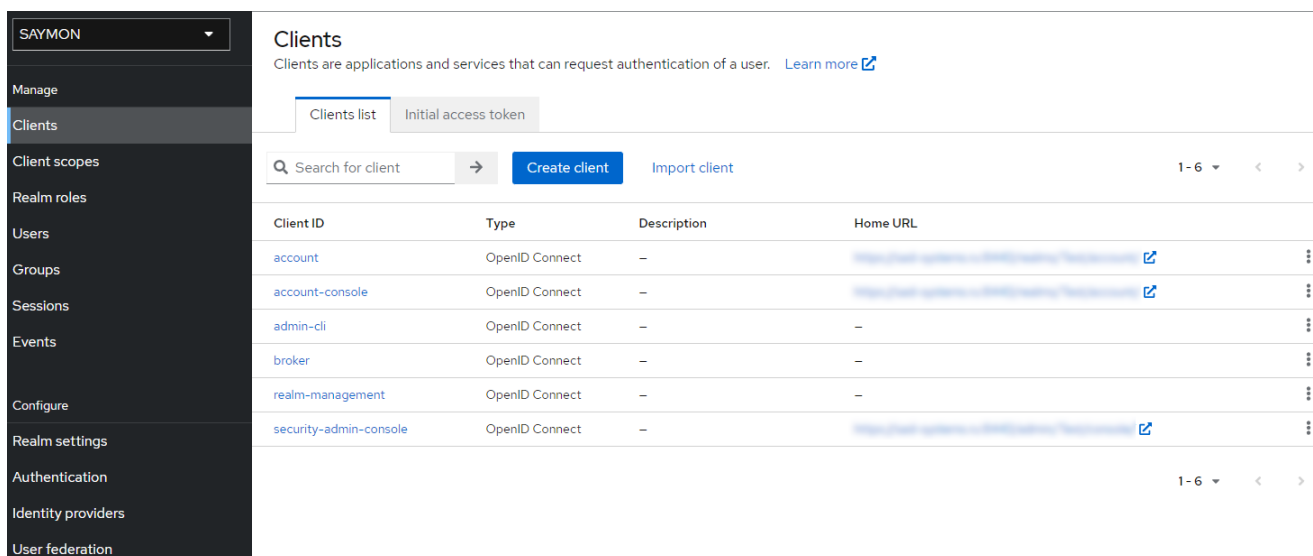
Главной особенностью Backend-клиента является наличие собственной аутентификации: такой клиент получает конфиденциальный доступ по секретному ключу.

Для настройки Backend-клиента необходимо:

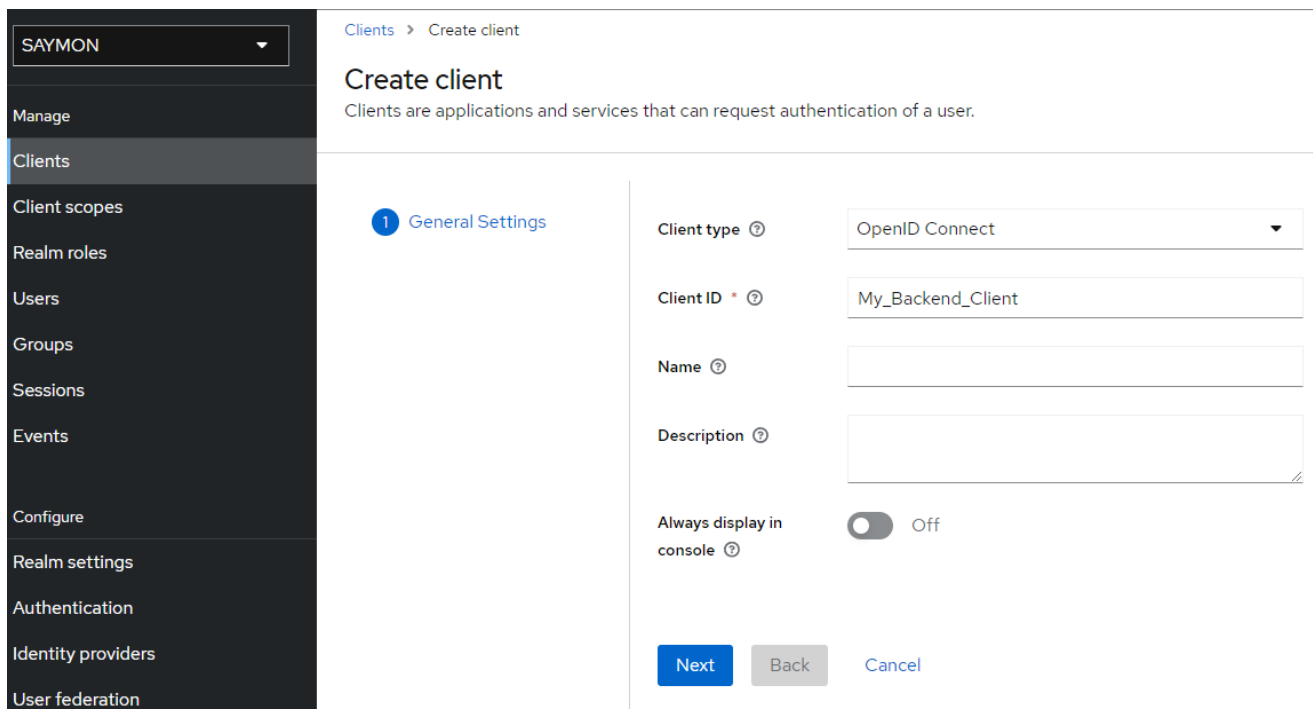
1. Выбрать требуемую область (Realm) из списка (здесь - **SAYMON**):



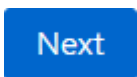
2. В секции **Clients** нажать кнопку :



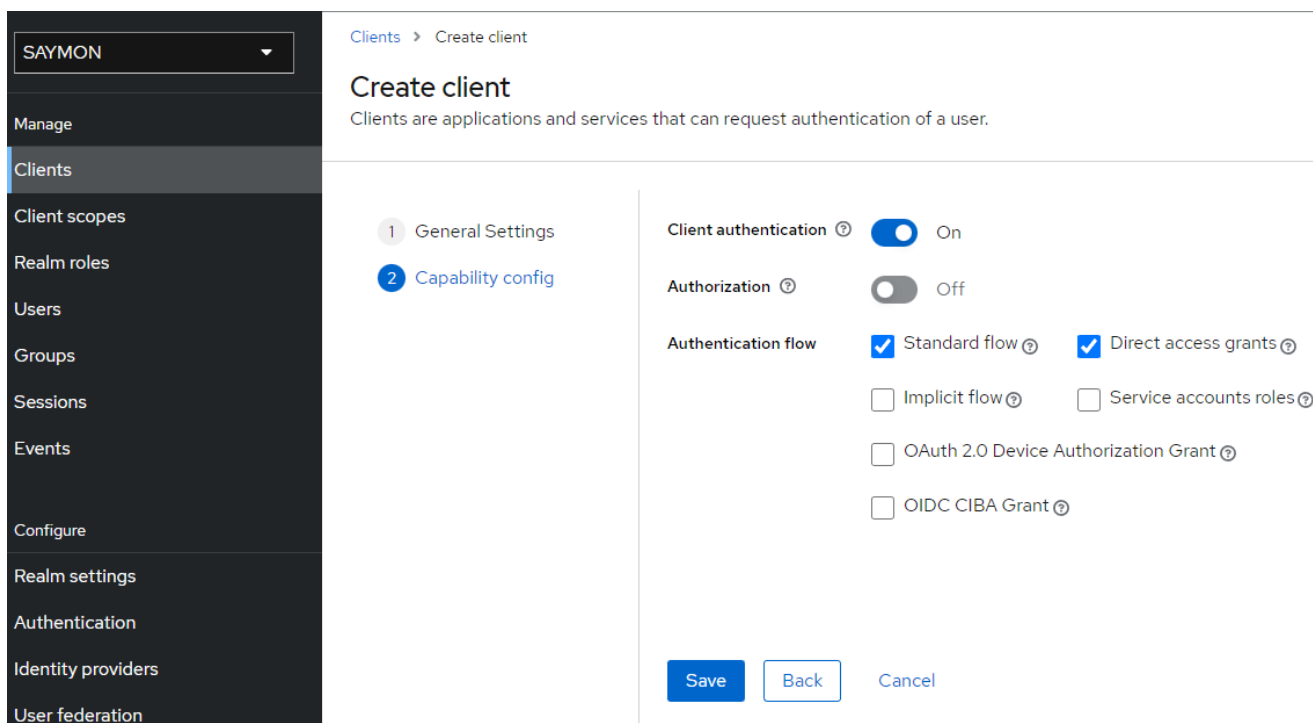
3. Ввести имя Backend-клиента в поле **Client ID**:



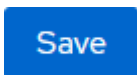
и нажать кнопку



4. Слайдер **Client authentication** установить в положение **On**:



и нажать кнопку



5. В списке **Action** выбрать **Download adapter config**:

The screenshot shows the Keycloak Admin Console interface. On the left is a navigation sidebar with options like 'Manage', 'Clients', 'Client scopes', etc. The main area displays the configuration for 'My_Backend_Client'. At the top right, there is a toggle for 'Enabled' and an 'Action' dropdown menu. The dropdown menu is open, showing options: 'Download adapter config', 'Export', and 'Delete'. The 'Download adapter config' option is highlighted with a mouse cursor. Below the dropdown, there are tabs for 'Settings', 'Keys', 'Credentials', 'Roles', 'Client scopes', 'Sessions', and 'Advanced'. The 'General Settings' section includes fields for 'Client ID', 'Name', and 'Description', and a toggle for 'Always display in console'. At the bottom, there are 'Save' and 'Revert' buttons.

- Нажать кнопку **Download**, чтобы получить файл настроек соединения для Backend-клиента в формате JSON:

The screenshot shows a dialog box titled 'Download adaptor configs'. It contains an information icon and a description: 'keycloak.json file used by the Keycloak OIDC client adapter to configure clients. This must be saved to a keycloak.json file and put in your WEB-INF directory of your WAR file. You may also want to tweak this file after you download it.' Below the description is a 'Format option' dropdown menu set to 'Keycloak OIDC JSON'. Underneath is a 'Details' section with a text area containing the following JSON configuration:

```
{
  "realm": "SAYMON",
  "auth-server-url": "http://localhost:8080/auth",
  "ssl-required": "external",
  "resource": "My_Backend_Client",
  "credentials": {
    "secret": "LLXQ2VnqycLGgQ71S65Il2a1ITbhNCBp"
  },
  "confidential-port": 0
}
```

At the bottom of the dialog, there are 'Download' and 'Cancel' buttons.

Конфигурация сервера Центрального Пульта

Для взаимодействия с сервером Keycloak на сервере Центрального Пульта необходимо:

1. Добавить раздел **Keycloak** в конфигурационный файл сервера `/etc/saymon/saymon-server.conf`:

```
"keycloak": {  
  "realm_certificate": "<данные keycloak_realm_certificate>",  
  "realm_certificate_file": "/etc/saymon/keycloak_realm_certificate",  
  "config_file": "/etc/saymon/keycloak_backend.json"  
}
```

где:

- **realm_certificate** - данные Keycloak Realm Certificate;
- **realm_certificate_file** - путь к файлу с Keycloak Realm Certificate;
- **config_file** - путь к файлу настроек соединения, сгенерированному сервером Keycloak для Backend-клиента.

WARNING

Параметры перечислены в порядке понижения приоритета при одновременном указании. Для работы с Keycloak достаточно указать один из параметров.

2. Перезапустить службу `saymon-server`:

```
$sudo service saymon-server restart
```

3.5.3.2. Настройка web-интерфейса

Для возможности авторизации пользователей посредством Keycloak через web-интерфейс Центрального Пульта необходимо получить файл настроек соединения Frontend-клиента.

WARNING

Настройка клиента и получение файла настроек соединения осуществляются в консоли администратора Keycloak.

Frontend-клиент

Главной особенностью Frontend-клиента является отсутствие собственной аутентификации: данный клиент получает публичный доступ.

Для настройки Frontend-клиента необходимо:

1. Выбрать требуемую область (Realm) из списка (здесь - **SAYMON**):

SAYMON

Manage

Clients

Client scopes

Realm roles

Users

Groups

Sessions

Events

Configure

Realm settings

Authentication

Identity providers

User federation

Clients > Create client

Create client

Clients are applications and services that can request authentication of a user.

1 General Settings

Client type [?] OpenID Connect

Client ID * [?] My_Frontend_Client

Name [?]

Description [?]

Always display in console [?] Off

Next Back Cancel

2. В секции **Clients** нажать кнопку **Next** :

SAYMON

Manage

Clients

Client scopes

Realm roles

Users

Groups

Sessions

Events

Configure

Realm settings

Authentication

Identity providers

User federation

Clients > Create client

Create client

Clients are applications and services that can request authentication of a user.

1 General Settings

2 Capability config

Client authentication [?] Off

Authorization [?] Off

Authentication flow

Standard flow [?] Direct access grants [?]

Implicit flow [?] Service accounts roles [?]

OAuth 2.0 Device Authorization Grant [?]

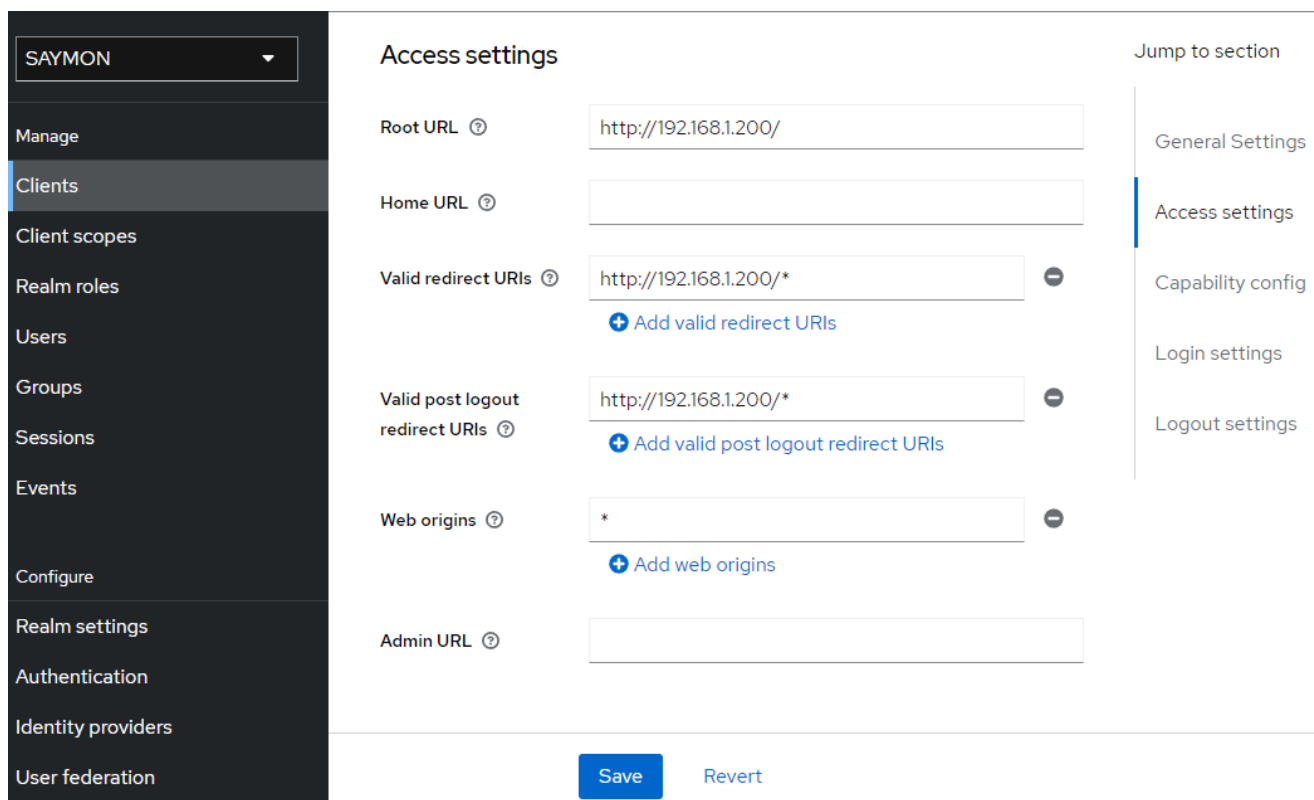
OIDC CIBA Grant [?]

Save Back Cancel

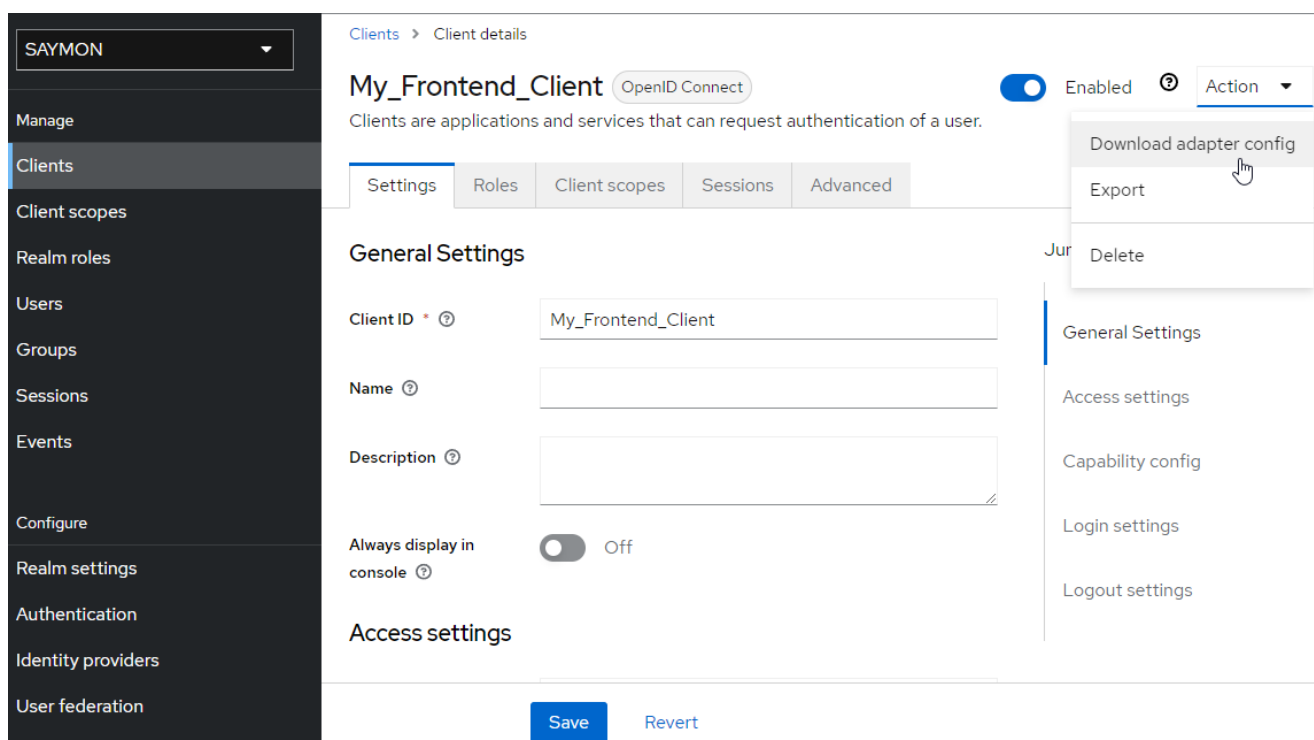
3. Ввести имя Frontend-клиента в поле **Client ID**:

Save

и нажать кнопку



4. Слайдер **Client authentication** установить в положение **Off**:



и нажать кнопку **Download**.

5. На вкладке **Settings** в разделе **Access settings** указать корректные URI-адреса, с которых будет осуществляться доступ к Keycloak-серверу, и на которые будут перенаправляться пользователи после успешной аутентификации и выхода:

Download adaptor configs ✕

i description

keycloak.json file used by the Keycloak OIDC client adapter to configure clients. This must be saved to a keycloak.json file and put in your WEB-INF directory of your WAR file. You may also want to tweak this file after you download it.

Format option ?

Keycloak OIDC JSON

Details ?

```
{
  "realm": "SAYMON",
  "auth-server-url": "https://auth-server.com:8080",
  "ssl-required": "external",
  "resource": "My_Frontend_Client",
  "public-client": true,
  "confidential-port": 0
}
```

Download
Cancel

и нажать кнопку .

6. В списке **Action** выбрать **Download adapter config**:

STAGING Журнал Событий SNMP ▼ ▼ Фильтр ▼ Вкл ▼ Количество сообщений: 100								
Количество ↕	Время ▼	Критичность ↕	Объект на схеме ↕	Адрес отправителя ↕	OID трапа ↕	Текст ↕	Данные ↕	Ответственный ↕
100	03.11.2020, 13:40:53	Major	IT'S A TRAP	127.0.0.1	.1.3.6.1.4.1.5089.2.0.99	0	.1.3.6.1.4.1.5089.2.0.99 "0"	admin
	03.11.2020, 13:39:53	Major	IT'S A TRAP	127.0.0.1	.1.3.6.1.4.1.5089.2.0.99	0	.1.3.6.1.4.1.5089.2.0.99 "0"	
	03.11.2020, 13:37:53	Major	IT'S A TRAP	127.0.0.1	.1.3.6.1.4.1.5089.2.0.99	0	.1.3.6.1.4.1.5089.2.0.99 "0"	

7. Нажать кнопку

Пользователи	Макс. объем хранилища (МБ)
Агенты	<input type="text" value="56"/>
Журнал событий	Максимальное количество записей
Шаблоны уведомлений	<input type="text" value="10020"/>
Состояния	<input type="button" value="Сохранить"/>
Аварии	<div style="background-color: #fff9c4; padding: 5px;">Предупреждение! Операция может занять длительное время.</div>
Классы объектов	
Скрипты	
Опции конфигурации	
Парольные политики	

, чтобы получить файл настроек соединения для Frontend-клиента в формате JSON:



Конфигурация web-интерфейса Центрального Пульта

Для активации формы авторизации через Keycloak в web-интерфейсе системы необходимо:

1. Добавить на сервере Центрального Пульта раздел **Authentication** в конфигурационный файл клиента `/etc/saymon/saymon-client.yaml`:

```
authentication:
  availableMethods:
    - "native"
    - "keycloak"
  methodByDefault: "keycloak"
  services:
    keycloak:
      configUri: keycloak_frontend.json
```

где:

- **availableMethods** - список доступных для пользователей методов аутентификации;
- **methodByDefault** - метод аутентификации, который будет предложен пользователю сразу при отображении аутентификационной формы;
- **configUri** - файл настроек соединения, сгенерированный сервером Keycloak для Frontend-клиента.

WARNING

Файл настроек соединения для Frontend-клиента необходимо поместить в каталог `/etc/saymon`.

2. Обновить страницу Центрального Пульта в web-браузере.

3.6. Управление логированием

Для хранения и дальнейшего анализа изменений в системной работе агента Центральный Пульт использует логирование.

Логи агента хранятся в папках: * Linux, Wiren Board 6 - `/var/log/saymon`; * macOS - `/opt/saymon-agent/log`; * Windows - папка_установки_агента\log.

Также в журналы записывается информация о следующих событиях: * дата и время удачных и неудачных попыток входа пользователей в систему; * причина и время выхода пользователей из системы; * дата, время и инициатор выполнения операций; * факт использования прав администратора; * факт доступа пользователей к основным конфигурационным данным платформы; * запуск и остановка сервисов аудита действий пользователя.

Все записи имеют временной штамп, признак инициатора (ID пользователя или процесса) и признак сессии, если событие было инициировано пользователем.

3.6.1. Конфигурация log-файлов агента

Конфигурация log-файлов агента выполняется в файле:

- `<PROGRAMFILES>/SAYMON Agent/conf/logback-winservice.xml` (Windows).
- `/opt/saymon-agent/conf/logback-daemon.xml` (Linux/Mac OS).

Логирование выполняется на уровне INFO. При необходимости возможно дополнительно настроить debug-режим.

Агент выполняет ротацию логов раз в день, старые логи архивируются.

NOTE

Архивация происходит в момент поступления в лог новой информации от агента. Если новых записей нет, архивация не выполняется.

3.6.1.1. Базовое логирование

Пример секции настроек базового логирования:

```
<appender name="FILE-INFO" class="RollingFileAppender">
  <file>log/saymon-agent.info.log</file>
  <rollingPolicy class="TimeBasedRollingPolicy">
    <!-- Daily rollover -->
```

```

    <fileNamePattern>log/saymon-agent.info.%d{yyyy-MM-dd}.log.zip</fileNamePattern>
    <!-- Keep 10 days' worth of history -->
    <maxHistory>10</maxHistory>
    <!-- Limit total size of all archive logs to 50 MB. -->
    <totalSizeCap>50MB</totalSizeCap>
  </rollingPolicy>
  <encoder class="PatternLayoutEncoder">
    <pattern>%d{dd.MM.yyyy HH:mm:ss.SSS} [%-15thread] %-5level %logger{36}
- %msg %kv%n%n%rootException</pattern>
  </encoder>
</appender>

```

где

- **<file>log/saymon-agent.info.log</file>** – размещение log-файла;
- **<fileNamePattern>... .zip</fileNamePattern>** – указание на архивацию файлов в формат .zip;
- **<maxHistory>10</maxHistory>** – длительность хранения файлов в днях;
- **<totalSizeCap>50MB</totalSizeCap>** – ограничение дискового пространства для хранения логов.

NOTE

Полная информация о настройках доступна в [документации по Logback](#).

3.6.1.2. Активация debug-режима

Для активации debug-режима необходимо:

1. Заменить содержимое файла конфигурации log-файлов агента содержимым файла
 - logback-winservice-debug.xml (Windows);
 - logback-daemon-debug.xml (Linux/Mac OS).

В этом файле находится секция настроек debug-режима:

```

<appender name="FILE-DEBUG" class="RollingFileAppender">
  ...
</appender>

```

NOTE

Файл с секцией настроек debug-режима находится в той же папке, что и файл конфигурации log-файлов агента.

2. Перезапустить агента:
 - служба SaymonAgent (Windows);
 - служба saymon-agent (Linux/Mac OS).

Параметры и структура секции debug-режима аналогичны секции базового логирования.

3.6.2. Конфигурация ротации log-файлов сервера

Конфигурация ротации log-файлов сервера выполняется в файле `/etc/logrotate.d/saymon`.

Параметры файла конфигурации ротации log-файлов:

Параметр	Описание
<code>/var/log/saymon/saymon-rest-server.log</code> <code>/var/log/saymon/saymon-server.log</code>	Размещение log-файлов.
<code>daily</code>	Ежедневная ротация.
<code>missingok</code>	Продолжать ротацию без ошибки, если отсутствует один из файлов.
<code>rotate N</code>	Длительность хранения файлов в днях.
<code>compress</code>	Архивация файлов в формат <code>.gzip</code> .
<code>notifempty</code>	Не производить ротацию лога, если он пуст.
<code>copytruncate</code>	Писать лог в один файл, урезая его после каждого шага ротации.

3.6.3. Просмотр информации о Журнале событий

Информацию о содержании журнала событий возможно просмотреть с помощью REST API метода:

```
GET /node/api/event-log/info
```


Пример (bash):

```
login=<...>
password=<...>
saymon_hostname=<...>
url=https://$saymon_hostname/node/api/event-log/info
curl -X GET $url -u $login:$password
```

3.6.4. Назначение ответственного за событие

При возникновении критической ситуации возможно установить ответственного за неё пользователя двумя способами:

1. Через Web UI:

- a. В панели режимов отображения открыть Журнал событий кнопкой .
- b. Нажать правой кнопкой мыши на требуемое событие, затем выбрать пункт "Назначить ответственного".
- c. Выбрать ответственного за событие из выпадающего списка.

Имя ответственного пользователя отобразится в соответствующем столбце (Рис. 15):

Количество ↕	Время ↕	Критичность ↕	Объект на схеме ↕	Адрес отправителя ↕	OID трапа ↕	Текст ↕	Данные ↕	Ответственный ↕
100	03.11.2020, 13:40:53	Major	IT'S A TRAP	127.0.0.1	.1.3.6.1.4.1.5089.2.0.99	0	.1.3.6.1.4.1.5089.2.0.99 "0"	admin
	03.11.2020, 13:39:53	Major	IT'S A TRAP	127.0.0.1	.1.3.6.1.4.1.5089.2.0.99	0	.1.3.6.1.4.1.5089.2.0.99 "0"	
	03.11.2020, 13:37:53	Major	IT'S A TRAP	127.0.0.1	.1.3.6.1.4.1.5089.2.0.99	0	.1.3.6.1.4.1.5089.2.0.99 "0"	

Рис. 15. Журнал событий

2. REST API методом:

```
PATCH /node/api/event-log/:id/assignee
```

Пример (bash):

```
login=<...>
password=<...>
saymon_hostname=<...>
record_id=<...>
url=https://$saymon_hostname/node/api/event-log/$record_id/assignee*
curl -X PATCH $url -u $login:$password -H "Content-Type: application/json" \
--data '{"userId": "..."}'
```

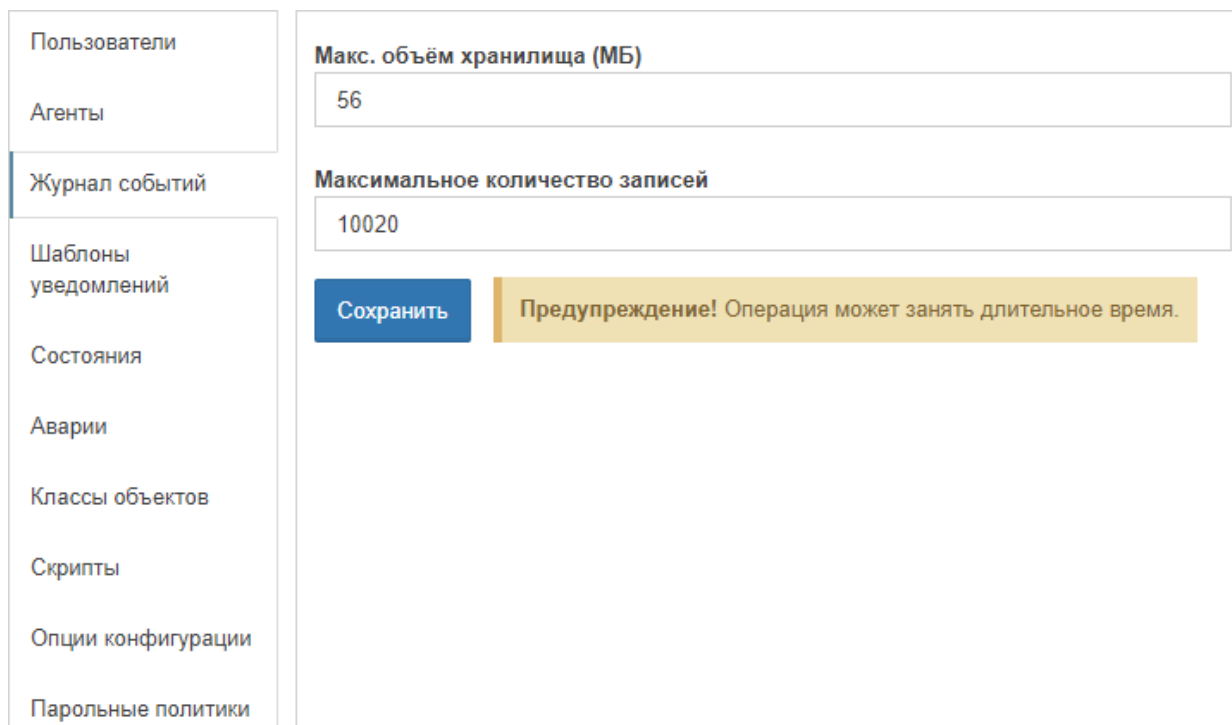
3.6.5. Установка ограничения для логирования

Центральный Пульт позволяет установить максимальный объём хранилища двумя способами:

1. Через Web UI:

- a. В панели инструментов нажать на имя пользователя и выбрать в меню пункт "Конфигурация".

b. Перейти в раздел "Журнал событий" (Рис. 16):



Пользователи

Агенты

Журнал событий

Шаблоны уведомлений

Состояния

Аварии

Классы объектов

Скрипты

Опции конфигурации

Парольные политики

Макс. объем хранилища (МБ)

56

Максимальное количество записей

10020

Сохранить

Предупреждение! Операция может занять длительное время.

Рис. 16. Журнал событий

c. Заполнить поля требуемыми значениями.

d. Сохранить изменения.

2. Через REST API метод:

```
PUT /node/api/event-log/limits
```

Пример (bash):

```
login=<...>
password=<...>
saymon_hostname=<...>
url=https://$saymon_hostname/node/api/event-log/limits

curl -X PUT $url -u $login:$password -H "Content-Type: application/json"
--data '{"maxBytes": 1024, "maxRecords": 100}'
```

3.6.6. Удаление всех SNMP-Trap'ов из Журнала событий

Для очистки Журнала событий необходимо:

1. Выполнить в терминале следующие команды:

```
mongo saymon
>db.eventLog.drop()
```


2. После сброса проверить создание новой коллекции командой:

```
>db.eventLog.stats()
```

3. Если ответ выглядит следующим образом, необходимо создать новую коллекцию:

```
{ "ok" : 0, "errmsg" : "ns not found" }
```

создание новой коллекции с лимитом по объёму в байтах:

```
>db.createCollection("eventLog", {capped:true, size: 100000000})
```

создание новой коллекции без лимита:

```
>db.createCollection("eventLog")
```

4. Повторно проверить создание коллекции командой:

```
>db.eventLog.stats()
```

Ответ должен выглядеть примерно следующим образом:

```
{
  "ns": "saymon.eventLog",
  "count": 0,
  "size": 0,
  "storageSize": 100003840,
  "numExtents": 1,
  "nindexes": 1,
  "lastExtentSize": 100003840,
  "paddingFactor": 1,
  "systemFlags": 1,
  "userFlags": 0,
  "totalIndexSize": 8176,
  "indexSizes": {
    "_id_": 8176
  },
  "capped": true,
  "max": NumberLong("9223372036854775807"),
  "ok": 1
}
```

3.6.7. Удаление логов

Удаление логов осуществляется следующей командой в консоли:

```
sudo rm -rf /var/log/upstart/*
```

Также возможно удалить логи с помощью REST API:

```
DELETE /node/api/event-log/:id
```

Пример (bash):

```
login=<...>
password=<...>
saymon_hostname=<...>
record_id=<...>
url=https://$saymon_hostname/node/api/event-log/$record_id

curl -X DELETE $url -u $login:$password
```

3.7. Управление учётными записями пользователей

Раздел содержит информацию об администрировании пользовательских учётных записей.

Группы пользователей позволяют упростить настройку и редактирование прав пользователей. При необходимости настроить нескольким пользователям идентичные

- права на операции,
- доступы к объектам,
- права на просмотр событий в Журнале событий,

достаточно создать группу пользователей, настроить ей необходимые права и добавить в эту группу нужных пользователей. В дальнейшем возможно редактировать права группы. Пользователь может принадлежать к неограниченному числу групп.


Права пользователя и его групп суммируются. Чтобы пользователь мог выполнять те или иные действия в системе, нужно, чтобы соответствующие права были либо у самого пользователя, либо у какой-то из его групп.

3.7.1. Авторизация пользователей через LDAP

Для авторизации пользователей с помощью сервера службы каталогов (Active Directory, OpenLDAP, и т.д.) необходимо добавить секцию LDAP в конфигурационный файл сервера `/etc/saymon/saymon-server.conf`.

NOTE

Подробную информацию о файле `/etc/saymon/saymon-server.conf` см. в подразделе [3.1.3, "Конфигурация сервера"](#).

При первом входе пользователя, аутентифицированного через LDAP, для него автоматически создаётся учётная запись в системе. У таких пользователей на вкладке "Общие" есть поле "Источник" со значением "LDAP", в списке пользователей они отмечаются иконкой 

WARNING

Если логин LDAP-пользователя совпадает с логином другого пользователя, уже зарегистрированного в системе, LDAP-пользователь авторизоваться не сможет.


Пользователям из LDAP, у которых ещё нет учётной записи в Центральном Пульте, можно ограничить вход в систему, подготовив группы для нужных пользователей. Чтобы учётная запись создавалась только для пользователей, у которых имя группы на LDAP-сервере совпадает с именем существующей (созданной ранее) группы в Центральном Пульте, в конфигурационном файле сервера необходимо указать параметр

ldap.create_user_for_existing_group_only со значением **true**:

```
"ldap" : {  
  ...  
  "create_user_for_existing_group_only": true,  
  ...  
}
```

При использовании протокола LDAP во время авторизации пользователя в системе могут быть автоматически созданы группы с именами, эквивалентными именам групп из LDAP, которые отсутствуют в Центральном Пульте. Авторизовавшийся пользователь автоматически добавляется в импортированные группы. В конфигурационном файле сервера для этого необходимо указать параметр **ldap.import_non_existing_groups** со значением **true**:

```
"ldap" : {  
  ...  
  "import_non_existing_groups": true,  
  ...  
}
```

LDAP-группы - группы, созданные при авторизации LDAP-пользователей - обозначаются в списке иконкой . Для таких групп недоступна смена имени, в качестве источника указан LDAP.

Группы, созданные в Центральном Пульте, имена которых совпадают с именами групп из LDAP, можно обновить до LDAP-групп. В конфигурационном файле сервера для этого необходимо указать параметр **ldap.update_existing_groups** со значением **true**:

```
"ldap" : {  
  ...
```

```
"update_existing_groups": true,
...
}
```

В этом случае при авторизации LDAP-пользователя те группы в Центральном Пульте, для которых есть одноимённые группы среди групп пользователя в LDAP, будут обновлены до LDAP-групп.

WARNING

Если пользователь был исключён из группы в Центральном Пульте - аналога своей группы в LDAP, при следующем входе в систему пользователь автоматически будет возвращён в эту группу.

3.7.2. Авторизация пользователей через Keycloak

Для авторизации пользователей с помощью внешнего сервера Keycloak необходимо:

- выполнить настройку клиентов и получить необходимые параметры в консоли администратора Keycloak;


NOTE

Описание необходимых действий в консоли администратора Keycloak приведено в подразделе [3.5.3, "Интеграция с Keycloak"](#).

- добавить секцию **Keycloak** в конфигурационный файл сервера `/etc/saymon/saymon-server.conf`;

NOTE

Подробную информацию о файле `/etc/saymon/saymon-server.conf` см. в подразделе [3.1.3, "Конфигурация сервера"](#).

При первом входе пользователя, аутентифицированного через Keycloak, для него автоматически создаётся учётная запись в системе. У таких пользователей на вкладке "Общие" есть поле "Источник" со значением " Keycloak", в списке пользователей они отмечаются иконкой 

WARNING

Если логин Keycloak-пользователя совпадает с логином другого пользователя, уже зарегистрированного в системе, Keycloak-пользователь авторизоваться не сможет.

3.7.3. Создание учётной записи

При создании пользователей и групп в именах и паролях допускаются любые символы, кроме символа @.

Создание учётных записей пользователей осуществляется двумя способами:

1. Через Web UI:

- a. В панели инструментов нажать на имя пользователя и выбрать в меню пункт "Конфигурация".
- b. Перейти в раздел "Журнал событий" (Рис. 17):

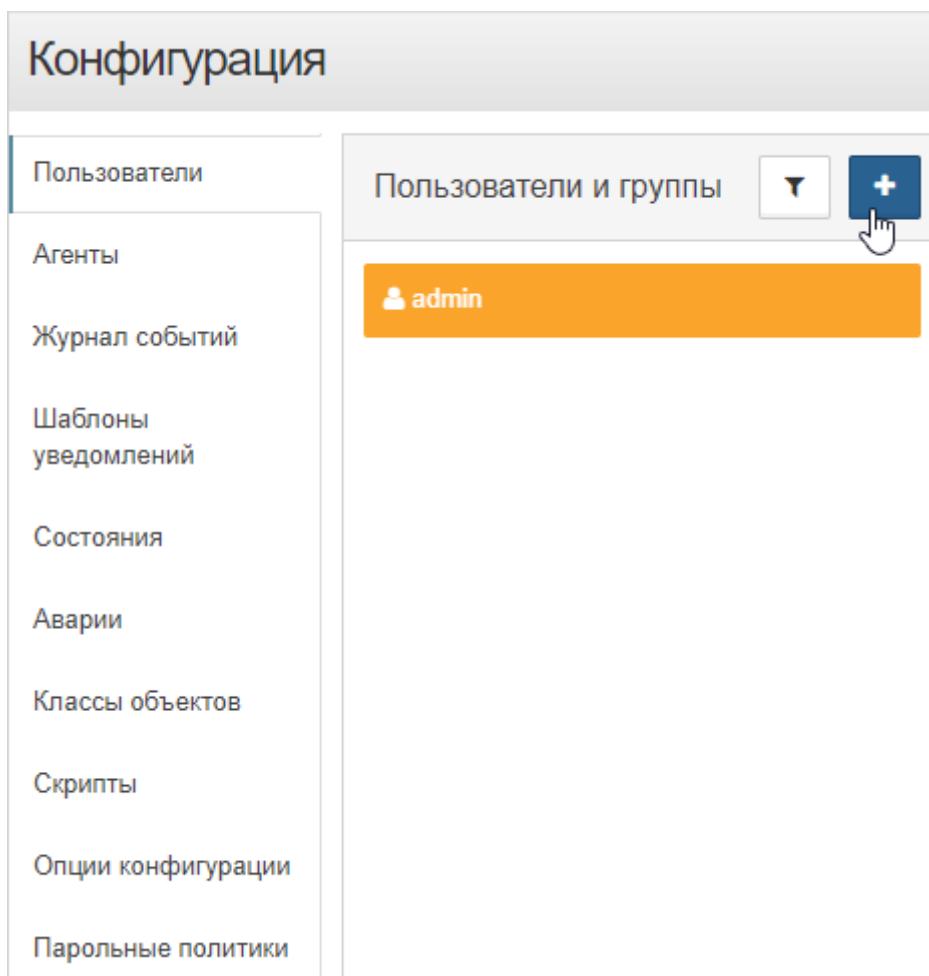




Рис. 17. Журнал событий

- c. Нажать кнопку  и выбрать добавление пользователя.
- d. Ввести логин пользователя, пароль и подтверждение.
- e. Нажать кнопку .

2. Через REST API:

```
POST /node/api/users
```

Пример (bash)

```
login=<...>
password=<...>
saymon_hostname=<...>
url=https://$saymon_hostname/node/api/users

curl -X POST $url -u $login:$password -H "Content-Type: application/json" \
```

```
+  
-d @- <<EOF  
{  
  "login": "Bob",  
  "password": "qwerty",  
  "permissions": [  
    "manage-objects",  
    "manage-links"  
  ]  
}  
EOF
```

3.7.4. Назначение пользователям прав доступа

NOTE

Список операций с подробным описанием содержится в Приложении А на стр. 173.

Настройка прав пользователей осуществляется двумя способами:

1. Через Web UI:
 - a. В панели инструментов нажать на имя пользователя и выбрать в меню пункт "Конфигурация".
 - b. Перейти в раздел "Пользователи".
 - c. Выбрать нужного пользователя или группу из списка.
 - d. Открыть вкладку "Права на операции" (Рис. 18):

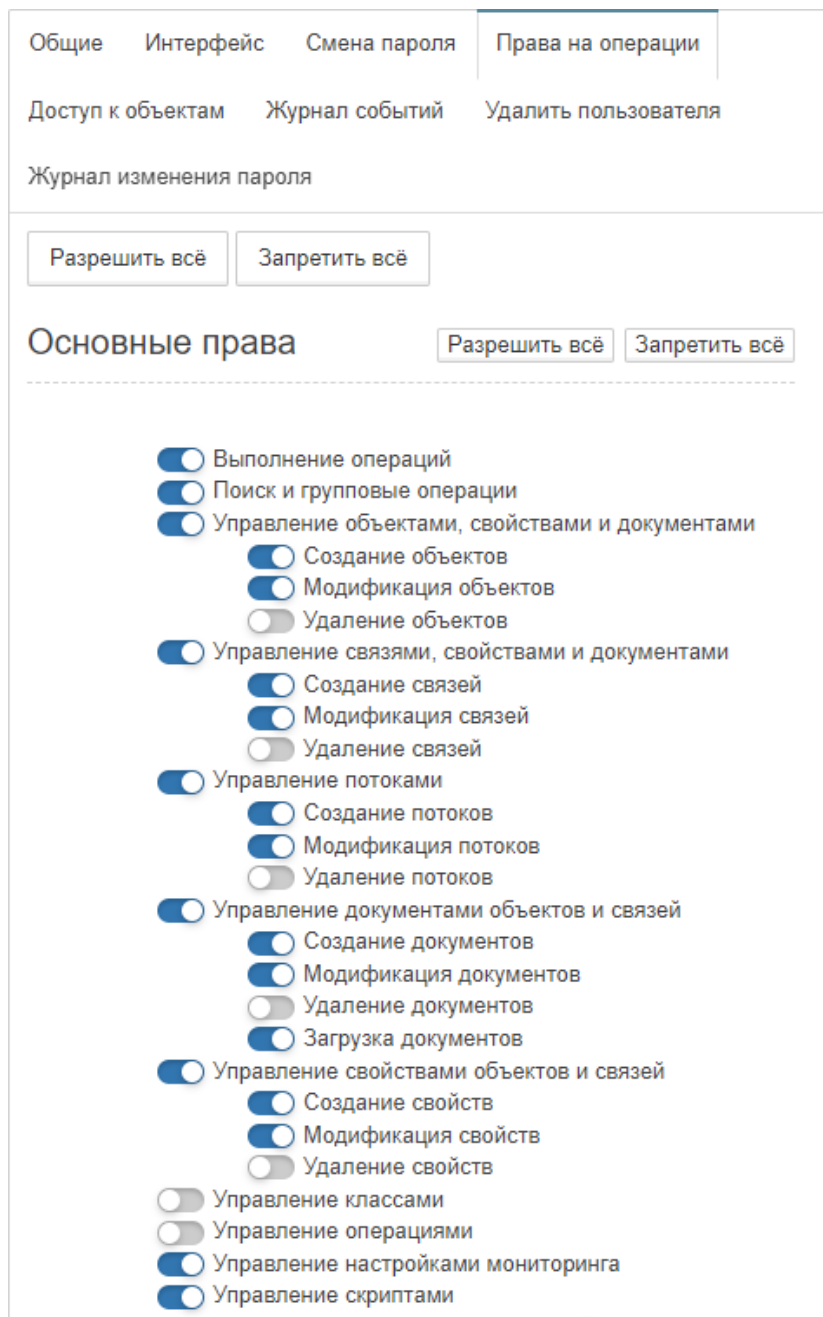


Рис. 18. Права на операции

е. Отредактировать права пользователя или группы.

WARNING	<i>Если пользователь принадлежит к группам, права пользователя и его групп суммируются.</i>
----------------	---

2. Через REST API:

```
PATCH /node/api/users/:id
```

Пример (bash):

```
login=<...>
password=<...>
saymon_hostname=<...>
```

```

user_id=<...>
url=https://$saymon_hostname/node/api/users/$user_id

curl -X PATCH $url -u $login:$password -H "Content-Type: application/json" \
-d @- <<EOF
{
  "login": "Bob",
  "password": "qwerty",
  "permissions": [
    "manage-objects",
    "manage-links"
  ]
}
EOF

```

3.7.5. Назначение парольных политик

Настройка парольных политик осуществляется через Web UI.

1. В панели инструментов нажать на имя пользователя и выбрать в меню пункт "Конфигурация".
2. Перейти в раздел "Парольные политики" (Рис. 19):

Конфигурация

Пользователи

Журнал сессий

Агенты

Журнал событий

Шаблоны уведомлений

Состояния

Аварии

Классы объектов

Скрипты

Опции конфигурации

Парольные политики

Парольные политики

По умолчанию

Редактирование политики

Имя: По умолчанию

Минимальное количество символов: 8

Минимальное количество символов нижнего регистра: 1

Минимальное количество символов в верхнем регистре: 1

Минимальное количество цифр: 1

Минимальное количество специальных символов: 1

Количество уникальных паролей: 1

Пользователь должен сменить пароль при первом входе

Срок действия пароля (дни): 180

Период, за который пользователя необходимо уведомить об окончании срока действия его пароля (дни): 1

Применить

Рис. 19. Парольные политики

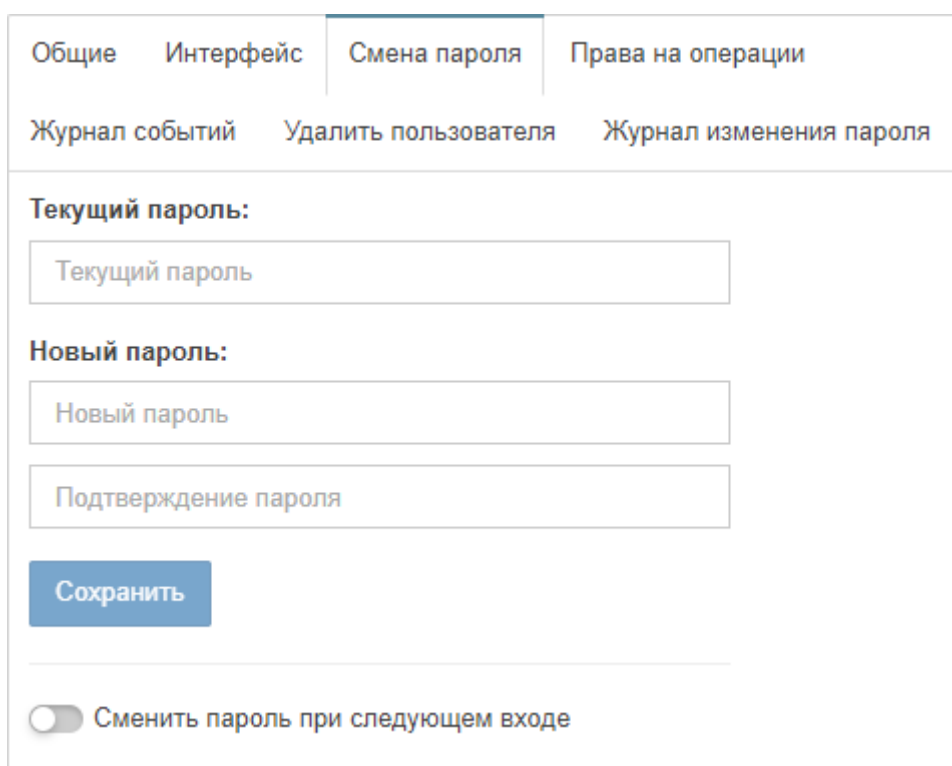
3. Установить необходимые требования к паролям пользователей.

4. Нажать кнопку .

3.7.6. Изменение пароля от учётной записи

Смена пароля пользователей осуществляется двумя способами:

1. Через Web UI:
 - a. В панели инструментов нажать на имя пользователя и выбрать в меню пункт "Конфигурация".
 - b. Перейти в раздел "Пользователи".
 - c. Выбрать нужного пользователя из списка.
 - d. Открыть вкладку "Смена пароля" (Рис. 20):



Общие Интерфейс **Смена пароля** Права на операции

Журнал событий Удалить пользователя Журнал изменения пароля

Текущий пароль:

Новый пароль:

Сохранить

Сменить пароль при следующем входе

Рис. 20. Смена пароля пользователя

- e. Ввести новый пароль пользователя и подтверждение.

WARNING

При смене собственного пароля необходимо также ввести текущий пароль.

- f. Нажать кнопку .

2. Через REST API:

```
PUT /node/api/users/:id/password
```

Пример (bash):

```
login=<...>
password=<...>
saymon_hostname=<...>
user_id=<...>
url=https://$saymon_hostname/node/api/users/$user_id/password

curl -X PUT $url -u $login:$password -H "Content-Type: application/json" \
-d @- <<EOF
{
"currentPassword": "qwerty",
"newPassword": "qwerty_qwerty"
}
EOF
```

3.7.7. Настройка доступа к объектам

Настройка доступа к объектам осуществляется через Web UI.

1. В панели инструментов нажать на имя пользователя и выбрать в меню пункт "Конфигурация".
2. Перейти в раздел "Пользователи".
3. Выбрать нужного пользователя или группу из списка.
4. Открыть вкладку "Доступ к объектам" (Рис. 21):

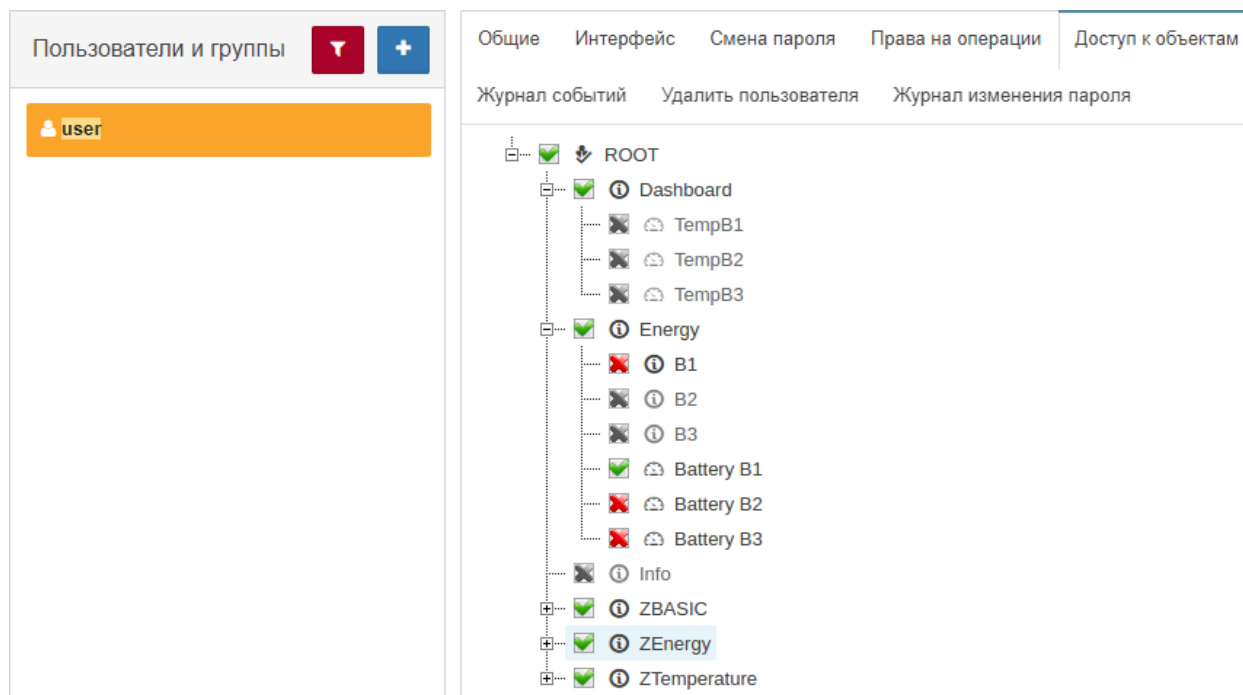


Рис. 21. Доступ к объектам

5. Отметить объекты, к которым необходимо предоставить/заблокировать доступ.

WARNING

Объекты, недоступные для группы, становятся недоступными для всех членов группы.

3.7.8. Блокировка учётной записи

Блокировка учётных записей пользователей осуществляется двумя способами:

1. Через Web UI:

- a. В панели инструментов нажать на имя пользователя и выбрать в меню пункт "Конфигурация".
- b. Перейти в раздел "Пользователи".
- c. Выбрать нужного пользователя из списка.
- d. Открыть вкладку "Общие" (Рис. 22):

Общие | Интерфейс | Смена пароля | Права на операции | Доступ к объектам | Журнал событий

Удалить пользователя | Журнал изменения пароля

ID 58048d843bddd07007607034

Логин demo

Email

Язык English ▾

Статус Активен ▾

Активен до DD.MM.YYY

Группа -- Не выбрано -- ▾

Ссылка для авторизации

Активность	⌚ Время	⚡ Действие	* Тип	Разница
	07.12.2022, 15:58:39	Удалено	Session	+ "id": "09d30aed-583e-4e6b-82e5-f97187627ab9", "userId": "58048d843bddd07007607034", "tokenId": "a6060167-61b1-4e78-ba79-daaa9eeefba0" }

Рис. 22. Общие параметры пользователя

- e. В выпадающем списке "Статус" выбрать "Заблокирован".

В поле **Активен до** можно задать дату и время блокировки учётной записи. При наступлении указанного времени статус учётной записи будет автоматически

изменён на **Заблокирован**.

WARNING

В случае если указанное время меньше текущего, статус учётной записи немедленно изменяется на **Заблокирован**.

2. Через REST API:

```
PATCH /node/api/users/:id
```

Пример (bash):

```
login=<...>
password=<...>
saymon_hostname=<...>
user_id=<...>
url=https://$saymon_hostname/node/api/users/$user_id/password

curl -X PATCH $url -u $login:$password -H "Content-Type: application/json" \
-d @- <<EOF
{
  "status": "3",
}
EOF
```

3.7.9. Удаление учётной записи

Удаление учётных записей осуществляется двумя способами:

1. Через Web UI:

- a. В панели инструментов нажать на имя пользователя и выбрать в меню пункт "Конфигурация".
- b. Перейти в раздел "Пользователи".
- c. Выбрать нужного пользователя из списка.
- d. Открыть вкладку "Удалить пользователя" (Рис. 23):

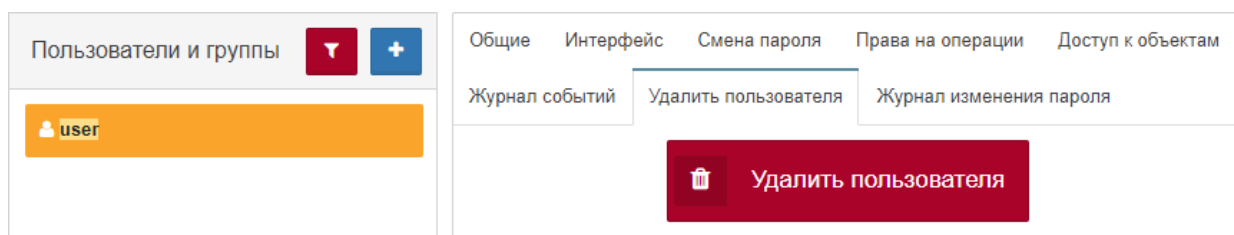


Рис. 23. Удаление пользователя

- e. Нажать кнопку удаления пользователя.
- f. Подтвердить удаление во всплывающем окне.

2. Через REST API:

```
DELETE /node/api/users/:id
```

Пример (bash):

```
login=<...>
password=<...>
saymon_hostname=<...>
user_id=<...>
url=http://$saymon_hostname/node/api/users/$user_id

curl -X DELETE $url -u $login:$password
```

WARNING

После удаления учётной записи пользователя LDAP/Keycloak с сервера системы пользователь по-прежнему сможет войти в систему под своими учётными данными, авторизовавшись через LDAP/Keycloak. Учётная запись при этом будет создана заново с правами по умолчанию; права, полученные через группы, останутся прежними.

*Чтобы заблокировать такому пользователю доступ к данным, достаточно сменить его статус на **Заблокирован**.*

3.7.9.1. Просмотр журнала действий пользователей

Просмотр журнала действий пользователей осуществляется двумя способами:

1. Через Web UI:

- a. В панели инструментов нажать на имя пользователя и выбрать в меню пункт "Конфигурация".
- b. Перейти в раздел "Пользователи".
- c. Выбрать нужного пользователя из списка.
- d. Открыть вкладку "Журнал активности" ([Рис. 24](#)):

Общие	Интерфейс	Смена пароля	Права на операции	Доступ к объектам	Журнал событий	Удалить пользователя
Журнал изменения пароля		Журнал активности				
Время	Действие	Тип	Разница			
23.06.2023, 13:15:09	Удалено	Session	{ "id": "86878afb-ba28-4118-9f06-e278e7790b1f", "userId": "5f5f420005a91e683e101ca7" },			
23.06.2023, 13:07:47	Создано	Session	{ "userId": "5f5f420005a91e683e101ca7" },			
23.06.2023, 12:57:24	Удалено	Session	{ "id": "b71e0671-b5ee-4385-a129-50b5fa3af3f0", "userId": "5f5f420005a91e683e101ca7" },			
23.06.2023, 11:32:20	Создано	Session	{ "userId": "5f5f420005a91e683e101ca7" },			
20.06.2023, 16:54:01	Изменено	Property	{ owner_id: "5f8dc4d207e86603bfe2853a", owner_type: 1, }			
20.06.2023, 16:54:01	Изменено	Property	{ owner_id: "5f8dc4d207e86603bfe2853a", owner_type: 1, }			
20.06.2023, 16:54:01	Изменено	Property	{ owner_id: "5f8dc4d207e86603bfe2853a", owner_type: 1, }			
20.06.2023, 16:54:01	Создано	Property	{ "type_id": 8, "name": "HttpRequestUrl", "value": "http://saymon.info", "id": "6491af79cf93930219af659f" },			

Рис. 24. Журнал активности

е. В поле "Активность" указаны все действия выбранного пользователя в системе.

2. Через REST API:

```
GET /node/api/users/:id/audit-log
```

Пример (bash):

```
login=<...>
password=<...>
saymon_hostname=<...>
user_id=<...>
url=https://$saymon_hostname/node/api/users/$user_id/audit-log

curl -X GET $url -u $login:$password \
-G --data-urlencode "limit=5"
```

3.8. Работа с объектами и связями


Вся управляемая инфраструктура в системе представлена в виде объектов и связей между ними.

Все метрики, характеризующие текущее состояние платформы, связываются с отдельными объектами мониторинга. Контроль загрузки и наличия свободных ресурсов осуществляется стандартными средствами платформы.

3.8.1. Создание объекта

Создать объект можно двумя способами:

1. Через Web UI:

- a. Нажать кнопку  на панели инструментов.
- b. В окне "Новый объект" ввести имя объекта и выбрать класс объекта (Рис. 25):

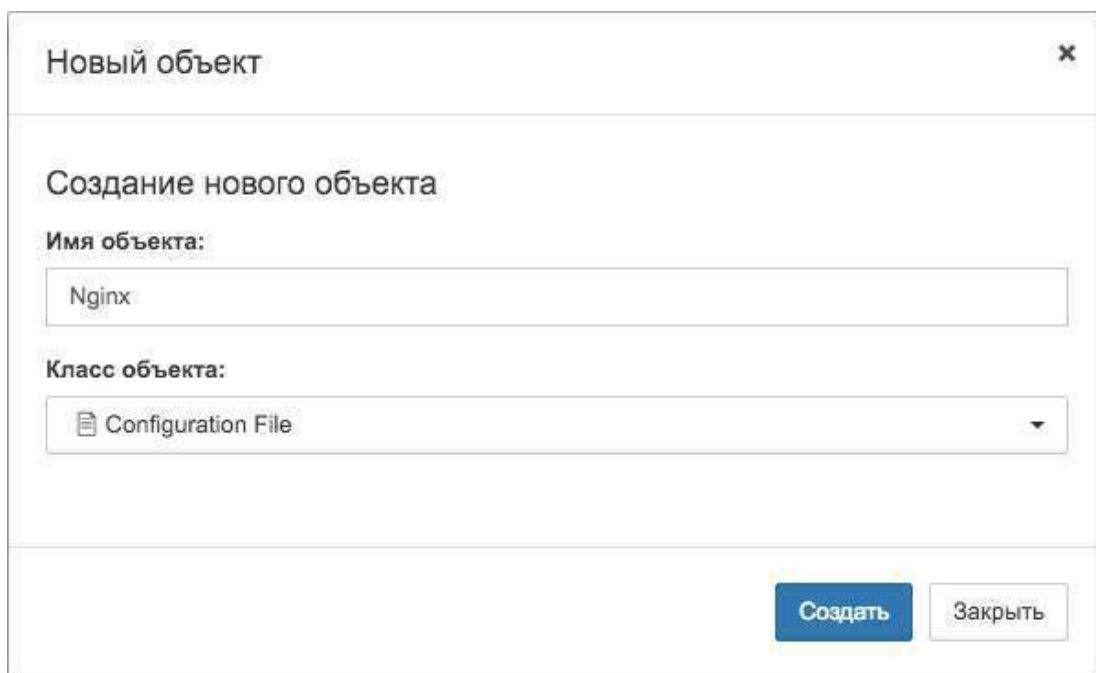


Рис. 25. Окно создания нового объекта

- c. Нажать кнопку .

2. Через REST API:

```
POST /node/api/objects
```

Пример (bash):

```
login=<...>
password=<...>
saymon_hostname=<...>
url=https://$saymon_hostname/node/api/objects

curl -X POST $url -u $login:$password -H "Content-Type: application/json" \
-d @- <<EOF
{
  "name": "New Object",
```

```
"parent_id": "1",  
"class_id": "3"  
}  
EOF
```

WARNING

"parent_id" - ID родительского объекта для создаваемого.

3.8.2. Клонирование объекта

Объект копируется со всеми своими документами, параметрами, свойствами дочерними объектами и связями.

Клонировать объект можно двумя способами:

1. Через Web UI:
 - a. Вызвать контекстное меню копируемого объекта щелчком правой кнопкой мыши.
 - b. Выбрать пункт меню "Клонировать".
2. Через REST API:

```
POST /node/api/objects/:id/clone
```


Пример (bash):

```
login=<...>  
password=<...>  
saymon_hostname=<...>  
object_id=<...>  
  
url=https://$saymon_hostname/node/api/objects/$object_id/clone  
  
curl -X POST $url -u $login:$password
```

3.8.3. Удаление объекта

Удалить объект можно тремя способами:

1. Через Web UI (контекстное меню объекта):
 - a. Вызвать контекстное меню удаляемого объекта щелчком правой кнопкой мыши.
 - b. Выбрать пункт меню "Удалить".
 - c. Подтвердить удаление объекта во всплывающем окне.
2. Через Web UI (режим удаления элементов):

- a. Нажать кнопку  на панели хлебных крошек.
- b. Нажать на такую же иконку на удаляемом объекте.
- c. Подтвердить удаление объекта во всплывающем окне.

3. Через REST API:

```
DELETE /node/api/objects/:id
```

Пример (bash):

```
login=<...>
password=<...>
saymon_hostname=<...>
object_id=<...>
url=https://$saymon_hostname/node/api/objects/$object_id

curl -X DELETE $url -u $login:$password
```


3.8.4. Создание ссылки на объект

Ссылка представляет собой особый тип объекта и служит для отображения уже настроенных в инфраструктуре объектов в других её частях, например, в дашбордах.

С помощью ссылок можно создавать связи и потоки между объектами, которые находятся в разных частях инфраструктуры.

Создать ссылку можно двумя способами:

1. Через Web UI:

- a. Нажать кнопку  на панели инструментов.
- b. В появившемся всплывающем окне "Новая ссылка" ([Рис. 26](#)) выбрать из выпадающего списка объект, на который создаётся ссылка:

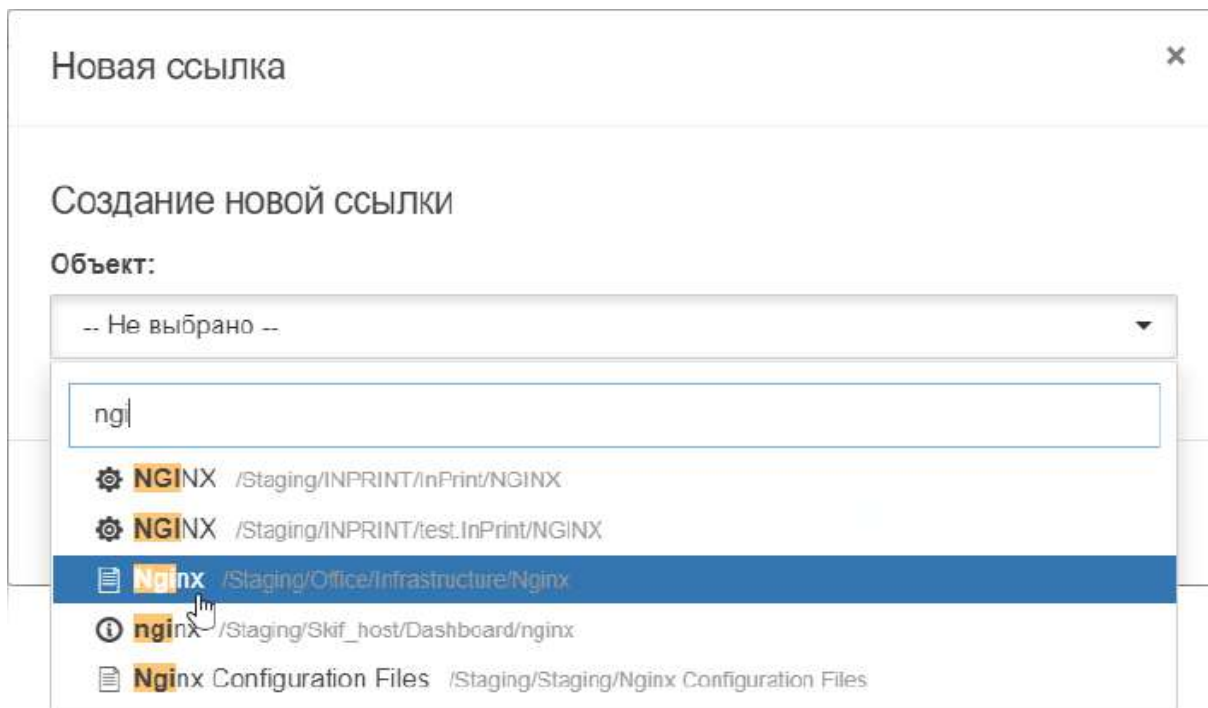


Рис. 26. Окно создания нового объекта

с. Нажать кнопку .

2. Через REST API:

```
POST /node/api/refs
```

Пример (bash):

```
login=<...>
password=<...>
saymon_hostname=<...>
url=https://$saymon_hostname/node/api/refs

curl -X POST $url -u $login:$password -H "Content-Type: application/json" \
--data '{"target": "5e60d9db630502472925fe9f", "owner": "1"}'
```


3.8.5. Создание связи

Связь может отражать как физическое соединение между объектами (например, соединение сервера с маршрутизатором с помощью Ethernet-кабеля), так и логическое отношение (например, поток данных от одного программного компонента к другому).


Связи между объектами также могут являться объектами мониторинга.

Создать связь можно двумя способами:

1. Через Web UI:

а. Перейти в режим создания связей, нажав кнопку  на панели

инструментов.

- b. После того, как на всех объектах появится соответствующий символ , нажать на него на исходном объекте и, удерживая, переместить курсор на целевой объект (Рис. 27):

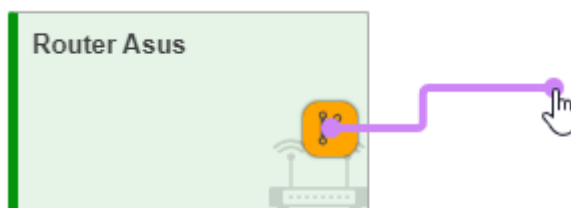


Рис. 27. Создание связи

Созданная связь отобразится между выбранными объектами (Рис. 28):

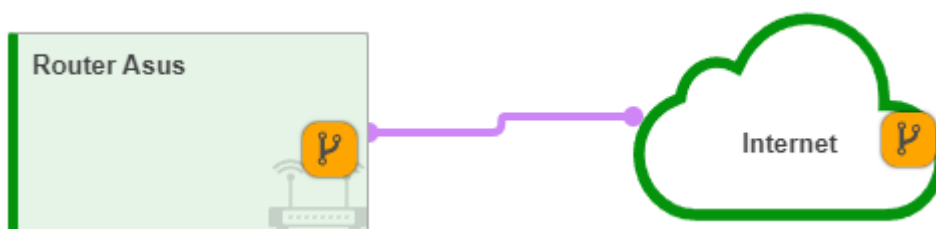



Рис. 28. Связь между объектами

- c. Нажать кнопку  в верхней части главного экрана для выхода из режима создания связей.

2. Через REST API:

```
POST /node/api/links
```

Пример (bash):


```
login=<...>
password=<...>
saymon_hostname=<...>
source_obj_id=<...>
target_obj_id=<...>
url=https://$saymon_hostname/node/api/links

curl -X POST $url -u $login:$password -H "Content-Type: application/json" \
-d @- <<EOF
{
"source": "'$source_obj_id'",
"target": "'$target_obj_id'"
}
EOF
```

3.8.6. Удаление связи

Удалить связь можно двумя способами:

1. Через Web UI:

- a. Нажать кнопку  на панели хлебных крошек.
- b. Нажать на такую же иконку на удаляемой связи.
- c. Подтвердить удаление связи во всплывающем окне.

2. Через REST API:

```
DELETE /node/api/links/:id
```

Пример (bash):

```
login=<...>
password=<...>
saymon_hostname=<...>
link_id=<...>
url=https://$saymon_hostname/node/api/links/$link_id


curl -X DELETE $url -u $login:$password
```

3.9. Настройка интерфейса

В разделе описаны механизмы настройки элементов интерфейса для лучшей визуализации данных.

3.9.1. Выравнивание/расстановка объектов в стандартном виде

Выравнивание объектов в стандартном виде происходит благодаря сетке.

Для отображения/скрытия сетки необходимо нажать кнопку  на панели "хлебных крошек".

Настройки сетки выполняется в файле `/etc/saymon/saymon-client.yaml`

NOTE

Подробную информацию о файле `/etc/saymon/saymon-client.yaml` см. в подразделе [3.1.4, "Настройка элементов web-интерфейса"](#).

В секции **Grid** этого файла можно задать параметры:

- **border** - максимальный отступ границы объекта от границы сетки (в пикселях).
- **color** - цвет сетки в формате RGBA.

- **dim** - размер сетки (в пикселях).

```
grid:  
  dim: 20  
  color: rgba(128, 128, 128, 0.3)  
  border: 4
```

После внесения изменений в файл необходимо обновить страницу в браузере.

3.9.2. Настройка заголовка web-интерфейса

Для изменения заголовка в web-интерфейсе:

1. Открыть файл `/etc/saymon/saymon-client.yaml`.

NOTE

Подробную информацию о файле `/etc/saymon/saymon-client.yaml` см. в подразделе [3.1.4, "Настройка элементов web-интерфейса"](#).

2. В строке "title" ввести желаемое имя:

```
title: <your-new-name>
```

3. Обновить страницу браузера.

3.9.3. Перемещение/отключение фоновой иконки объекта

Все предустановленные классы объектов имеют индивидуальную фоновую иконку. При необходимости переместить иконку, в секции "Параметры" во вкладке "Стили" объекта необходимо добавить:

```
.object-background {  
  background-position: 9px 8px !important;  
  background-size: 15px;  
  opacity: 1;  
}  
  
.object-caption-panel {  
  padding-left: 15px;  
}
```

Для отключения иконки требуется добавить в ту же секцию:

```
.background-component {  
  display: none  
}
```

3.9.4. Вертикальное отображение имени объекта

По умолчанию имя объекта отображается в его левом верхнем углу горизонтально. При необходимости отображать имя вертикально, в секции "Параметры" во вкладке "Стили" объекта необходимо добавить:

```
.js-caption {  
  writing-mode: vertical-lr;  
  text-orientation: upright;  
  text-transform: uppercase;  
  letter-spacing: 1px;  
}
```

3.9.5. Редактирование стилей состояний

В процессе мониторинга в зависимости от данных, получаемых от агента, объект может менять состояние.

Каждое состояние имеет цвет. Цвет и стиль состояний можно изменить двумя способами:

1. Через Web UI:

- a. В панели инструментов нажать на имя пользователя и выбрать в меню пункт "Конфигурация".
- b. Перейти в раздел "Состояния".
- c. В списке состояний выбрать то, которое требуется изменить.
- d. При необходимости изменить имя в соответствующем поле.
- e. Настроить основной цвет, цвет тени, цвет строки таблицы и фон при помощи цветовой палитры или методом ввода номера цвета.

2. Через файл конфигурации:

- a. Создать файл: **/usr/local/saymon/saymon.local/css/saymon.local.css**.
- b. Открыть его в текстовом редакторе и вставить код для изменения цвета состояния или его фона. Например:

```
.state-4  
{  
  background-color: rgba(255, 122, 0, 0.46);  
  box-shadow: 2px 5px 10px rgba(253, 118, 7, 0.5);  
  border-left: 5px solid #FD7607;  
}  
  
.state-bg-4  
{  
  background-color: #FD7607;  
}
```

```
.view-screen-element ul li.state-4
{
  border-left: 5px solid #FD7607;
}

.view-screen-element ul li .badge.state-4
{
  background-color: #FD7607;
}
```

Номер состояния соответствует его ID.

3.10. Настройка мониторинга

Платформа "Центральный Пульт" позволяет осуществлять мониторинг с использованием различных типов проверок. Проверки настраиваются в web-интерфейсе системы.

NOTE

Подробную информацию о каждом типе проверок см. в "Руководстве пользователя" в подразделе 4.2.1 "Настройка процесса мониторинга и типов проверок" на стр. 43.

Далее рассмотрены несколько примеров процесса мониторинга.

3.10.1. Мониторинг основных параметров ПК

Для мониторинга основных параметров работы сервера или ПК: CPU, File System, Memory и Network IO, достаточно выполнить несколько действий:

1. Установить агента на наблюдаемый ПК или сервер.
2. Создать объект, например, класса "Host", в web-интерфейсе.
3. Перейти в созданный объект и добавить внутри него объекты классов:
 - Saymon Agent,
 - CPU,
 - File System,
 - Memory,
 - Network IO.
4. Сконфигурировать и запустить агента.

Через некоторое время информация об основных параметрах работы компьютера начнёт поступать на сервер и отображаться в web-интерфейсе системы.

3.10.2. Мониторинг процесса памяти

Для настройки мониторинга процесса памяти необходимо:

1. Установить, сконфигурировать и запустить агента на наблюдаемом ПК или сервере.
2. Создать объект, например, класса "Process", в web-интерфейсе.
3. Перейти в созданный объект и в его секции "Мониторинг":
 - выбрать агента, установленного ранее на данный компьютер;
 - выбрать тип проверки "Процесс по имени";
 - заполнить необходимые поля.

Через некоторое время информация о проверяемом процессе начнёт поступать на сервер и отображаться в web-интерфейсе системы.

3.10.3. Мониторинг изменения файлов и папок

Для настройки мониторинга изменения файлов и папок сервера или ПК необходимо:

1. Установить, сконфигурировать и запустить агента на наблюдаемом ПК или сервере.
2. Создать объект, например, класса "Configuration File", в web-интерфейсе.
3. Перейти в созданный объект и в его секции "Мониторинг":
 - выбрать агента, установленного ранее на данный компьютер;
 - выбрать тип проверки "Конфигурационный файл/директория";
 - указать путь к проверяемому файлу/директории.

Через некоторое время информация о проверяемом файле/директории начнёт поступать на сервер и отображаться в web-интерфейсе системы.

3.10.4. Проверка доступности web-ресурса

Данный тип мониторинга позволяет убедиться не только в работоспособности web-сайта (статус 200 OK), но и в ограничении доступа к таким ресурсам, как панель администрирования баз данных. В этом случае статус "403 Forbidden" или "404 Not Found" будет говорить о правильности настройки системы, а иной статус - о возможной угрозе безопасности системы.

Для проверки доступности и скорости отклика web-ресурса необходимо:

1. Установить, сконфигурировать и запустить агента на наблюдаемом ПК или сервере.
2. Создать объект, например, класса "Address", в web-интерфейсе.
3. Перейти в созданный объект и в его секции "Мониторинг":

- выбрать агента, который будет выполнять проверку;
- выбрать тип проверки "HTTP-запрос";
- выбрать тип запроса "GET";
- в поле "URL" указать адрес web-сайта.

Через некоторое время информация о доступности и скорости отклика наблюдаемого ресурса начнёт поступать на сервер и отображаться в web-интерфейсе системы.

3.10.5. Безагентный мониторинг web-сервера

Существует ряд случаев, при которых установка агента на сервере невозможна. В таких случаях рекомендуется написать скрипт, который с заданной периодичностью будет выполняться на сервере, собирать необходимые данные и генерировать текстовый файл с результатами в формате JSON по ссылке, доступной извне.

Для мониторинга параметров web-сервера, на который невозможно поставить агента, необходимо:

1. Написать локальный скрипт, выполняющий подготовку данных (например, в папке загрузок: `.../downloads/scripts/webserver_stat.sh`):

```
#!/bin/bash
# Сбор параметров работы web-сервера.

# использование Memory
memUsage=$(free -m | grep Mem | perl -pe 's/Mem:\s+\S+\s+(\S+).*/$1/')

# использование Swap
swapUsage=$(free -m | grep Swap | perl -pe 's/Swap:\s+\S+\s+(\S+).*/$1/')

# загрузка CPU
cpuUsage=$(uptime | awk '{print $10}' | perl -pe 's/,//')

# проверка выполнения какого-либо скрипта, например, webserver_stat.sh
scriptExec=$(ps -ef | grep webserver_stat.sh | grep -v grep | wc -l)

# Write JSON response
echo "{\"memUsageMB\":\"$memUsage\", \"swapUsageMB\":\"$swapUsage\", \"cpuUsage\": \"$cpuUsage\", \"scriptExec\":\"$scriptExec\"}" > webserver_stat.json
```

2. Добавить выполнение скрипта в планировщик заданий cron.
3. Установить, сконфигурировать и запустить хотя бы одного агента в инфраструктуре.
4. Создать объект, например, класса "Info", в web-интерфейсе.
5. Перейти в созданный объект и в его секции "Мониторинг":
 - выбрать агента, который будет выполнять проверку;

- выбрать тип проверки "HTTP-запрос";
- выбрать тип запроса "GET";
- в поле "URL" указать адрес JSON-файла.

Через некоторое время информация о параметрах работы web-сервера начнёт поступать на сервер и отображаться в web-интерфейсе системы.

3.11. Страницы для внедрения в IFrame

Виджет IFrame, предназначенный для внедрения в интерфейс содержимого сторонних ресурсов, можно также использовать и со специально настроенными информационными страницами внутри Центрального Пульта.

3.11.1. Страница с авариями

В адресной строке страницы аварий возможно указать дополнительные URL-параметры, влияющие на представление панели элементов управления и отображение данных в таблице.

Пример для отключения всех элементов управления, кроме кнопки экспорта данных в Excel, и применения ранее сохранённого фильтра **Major** (вместо **127.0.0.1** указать адрес нужного сервера Центрального Пульта):

```
http://127.0.0.1/incidents.html?header=off&exportSection=on&filterName=Major
```


3.11.1.1. Параметры элементов управления

Параметр	Описание
header	<p>Включает/выключает отображение панели, содержащей элементы управления фильтрацией данных в таблицах;</p> <ul style="list-style-type: none"> • on - отображение панели включено, • off - отображение панели выключено. <p>По умолчанию - on - отображение панели включено.</p> <div style="border: 1px solid orange; padding: 10px; margin-top: 10px;"> <p>WARNING При выключенном отображении панели элементов управления выключаются все описанные в таблице элементы. В этом случае, при необходимости включить отдельный элемент, нужно отдельно вписать его в адресную строку со значением on.</p> </div>
cache	<p>Включает/выключает загрузку сделанных пользователем настроек из кэша браузера: будут ли настройки сохраняться при обновлении страницы браузера;</p> <ul style="list-style-type: none"> • on - загрузка настроек включена, • off - загрузка настроек выключена.
entityHref	<p>Включает/выключает отображение ссылок на аварийные элементы в столбце Объект/связь и их родительские объекты в столбце Путь;</p> <ul style="list-style-type: none"> • on - отображение ссылок включено, • off - отображение ссылок выключено.
exportSection	<p>Включает/выключает отображение кнопки экспорта данных в Excel;</p> <ul style="list-style-type: none"> • on - отображение кнопки экспорта данных включено, • off - отображение кнопки экспорта данных выключено.
filterSection	<p>Включает/выключает отображение секции расширенного фильтра;</p> <ul style="list-style-type: none"> • on - отображение секции фильтра включено, • off - отображение секции фильтра выключено.

Параметр	Описание
logo	Включает/выключает отображение логотипа; <ul style="list-style-type: none"> • on - отображение логотипа включено, • off - отображение логотипа выключено.
presetSection	Включает/выключает отображение секции представлений; <ul style="list-style-type: none"> • on - отображение секции представлений включено, • off - отображение секции представлений выключено.
propertyFieldsSection	Включает/выключает отображение секции с выбором полей свойств; <ul style="list-style-type: none"> • on - отображение секции с выбором полей свойств включено, • off - отображение секции с выбором полей свойств выключено.
typeSelect	Включает/выключает отображение селектора выбора типа данных - выбор между активными авариями и историей аварий; <ul style="list-style-type: none"> • on - отображение селектора включено, • off - отображение селектора выключено.

3.11.1.2. Параметры фильтра

Параметр	Описание
filterName	Имя сохранённого фильтра, который сразу будет применён к отображаемым данным.

Параметр	Описание
timeFilter	<p>Включает/выключает отображение дополнительного фильтра по временному диапазону;</p> <ul style="list-style-type: none"> • on - временной фильтр включен, • off - временной фильтр выключен. <p>По умолчанию - off - временной фильтр выключен.</p> 
type	<p>Выбор типа отображаемых данных - список исторических аварий;</p> <ul style="list-style-type: none"> • active - отображать список активных аварий, • history - отображать список исторических аварий. <p>По умолчанию отображается последний выбранный тип.</p>

3.11.2. Страница с графиками

В системе возможно настроить специальную страницу с набором графиков из одного или нескольких объектов ([Рис. 29](#)):

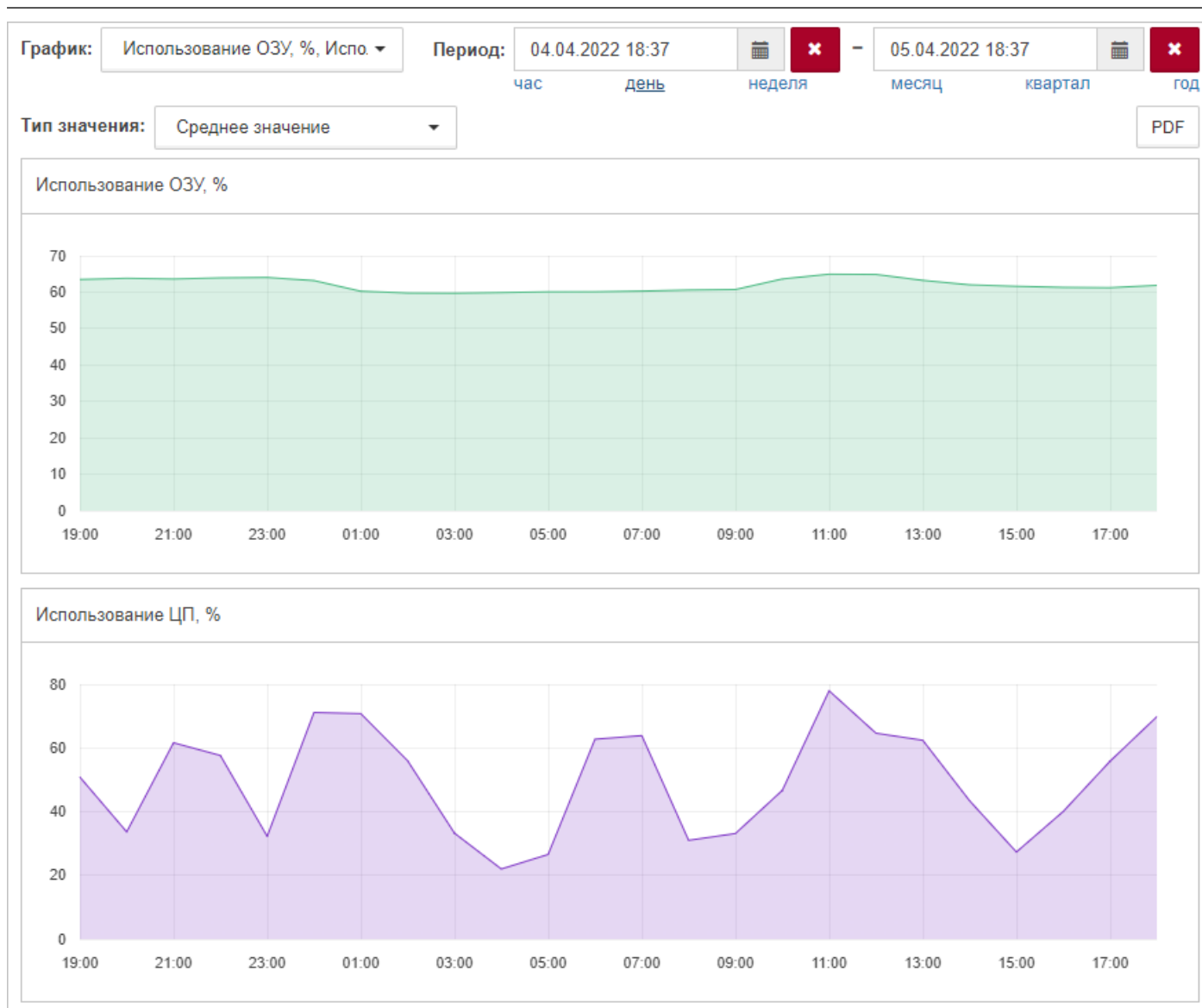


Рис. 29. Пример настроенной страницы графиков

В верхней части страницы представлены элементы управления (Рис. 30):

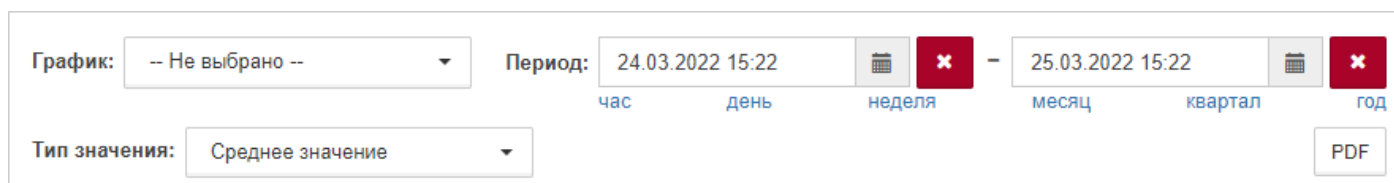


Рис. 30. Элементы управления страницы графиков

Элементы управления страницы графиков:

Элемент	Описание
График	Выпадающий список, в котором можно выбрать отображаемые графики подключенных метрик.

Элемент	Описание
Период	<p>Временной период для отображения графиков. При необходимости можно быстро ввести период за последние</p> <ul style="list-style-type: none"> • час, • день, • неделю, • месяц, • квартал, • год, <p>нажав соответствующую кнопку под полями выбора начала и конца периода.</p>
Тип значения	<p>Если при запросе графика по метрике возвращается большой массив точек, то для оптимального построения графика производится автоматическое объединение точек. Способ вычисления значений точек графиков:</p> <ul style="list-style-type: none"> • среднее значение - значения точек графика рассчитываются как среднее арифметическое значений метрики; • минимальное значение - значения точек графиков соответствуют минимальным значениям метрики; • максимальное значение - значения точек графиков соответствуют максимальным значениям метрики.
PDF	Кнопка сохранения страницы в PDF-формате.

Страница с графиками доступна по ссылке вида

<http://<адрес сервера Центрального Пульта>/iframe-widgets.html?#widget/selectable-chart/<ID основного объекта>/<дополнительные параметры>>

где

- **<адрес сервера Центрального Пульта>** - IP-адрес или FQDN инсталляции Центрального Пульта;
- **<ID основного объекта>** - идентификатор объекта, в свойствах которого хранится список метрик и их отображаемые названия;
- **<дополнительные параметры>** - настройки, которые определяют первоначальный вид страницы при её загрузке.

Пример:

`http://127.0.0.1/iframe-widgets.html?#widget/selectable-chart/5fcc9a4de0a5393f45fb99f3/
chart=61a662bdde9e7970b38b340c:percentageUsage.combined&chart=TOTAL.percentUsed&p
eriod=last-24hours`

В данном примере:

- сервер работает на локальной машине и доступен по адресу **127.0.0.1**;
- основным объектом указан объект с идентификатором **5fcc9a4de0a5393f45fb99f3**;
- при загрузке страницы отображаются график метрики **percentageUsage.combined** объекта с идентификатором **61a662bdde9e7970b38b340c** и график метрики **TOTAL.percentUsed** основного объекта за прошедшие сутки.

Возможные метрики для построения графиков необходимо указать в свойствах основного объекта.

В имени свойства указывается имя метрики в формате:

chart:<имя метрики>

При необходимости добавить метрику из другого объекта нужно добавить идентификатор объекта:

chart:<ID внешнего объекта>:<имя метрики>

В значении свойства задаётся отображаемое имя графика:

<желаемое название отображаемого графика>

Примеры:

Имя свойства: chart:TOTAL.percentUsed

Значение свойства: Использование ОЗУ, %

Описание: График метрики **TOTAL.percentUsed** основного объекта

Имя свойства: chart:61a662bdde9e7970b38b340c:percentageUsage.combined

Значение свойства: Использование ЦП, %

Описание: График метрики *percentageUsage.combined* объекта с ID **61a662bdde9e7970b38b340c**

Дополнительные параметры страницы графиков:

Параметр	Возможные значения	Описание
aggregate	<p>aggregate=avg - выбираются средние значения,</p> <p>aggregate=max - выбираются максимальные значения,</p> <p>aggregate=min - выбираются минимальные значения.</p>	<p>Тип значения, применяемый при загрузке страницы.</p> <p>По умолчанию применяется функция усреднения.</p>
chart	<p>chart:<имя метрики основного объекта> chart:packetsTransmitted</p> <p>chart:<ID нужного объекта>:<имя метрики нужного объекта> chart:61a662bdde9e7970b38b340c:freeMemory</p> <p><i>Используемые в URL метрики должны быть заранее настроены в свойствах основного объекта.</i></p>	<p>Графики, которые будут отображаться при загрузке страницы.</p> <p>Если не указать данный параметр, то по умолчанию откроется страница без графиков.</p>
period	<p>period=last-hour - за последний час,</p> <p>period=last-24hours - за последний день,</p> <p>period=last-7days - за последнюю неделю,</p> <p>period=last-30days - за последний месяц,</p> <p>period=last-90days - за последний квартал,</p> <p>period=last-365days - за последний год.</p>	<p>Период времени для построения графиков, выбираемый при загрузке страницы.</p> <p>По умолчанию отображаются графики за последний час.</p>

Параметр	Возможные значения	Описание
widget-instance-id	<p>Значение параметра - имя представления, уникальное в рамках основного объекта. В качестве значения может быть использован любой набор символов.</p> <p>Примеры:</p> <ul style="list-style-type: none"> • widget-instance-id=1 - сохранение настроек в представлении с именем 1; • widget-instance-id=example - сохранение настроек в представлении с именем example. 	<p>Сохранение настроек страницы в отдельном представлении с уникальным именем.</p> <p>По умолчанию настройки не записываются в представление.</p>

Параметр **widget-instance-id** позволяет сохранять все производимые пользователем настройки в представлении с соответствующим именем.

Имена представлений задаются пользователями произвольно.

TIP

Функционал будет полезен при необходимости одновременного просмотра графиков одних и тех же метрик с различными настройками отображения.

Если встроить одну и ту же страницу с графиками в несколько IFrame-виджетов, то все изменения, вносимые через элементы управления на одном из виджетов, будут применяться и к остальным. Изменения с настройками отображения страницы можно зафиксировать в отдельном представлении с помощью параметра **widget-instance-id**. Имя представления соответствует значению параметра.

При указании **widget-instance-id** в дополнительных параметрах, изменения, вносимые через другие IFrame-виджеты с той же страницей с графиками, не будут влиять на настройки, сохранённые в представлении.

В настройках IFrame-виджета можно менять значение параметра **widget-instance-id**, чтобы создать еще одно представление или восстановить настройки из ранее сохранённого.

Пример ссылки на страницу с графиками с использованием **widget-instance-id**:

<http://127.0.0.1/iframe-widgets.html?#widget/selectable-chart/5fcc9a4de0a5393f45fb99f3/widget-instance-id=default>

В данном примере:

- сервер работает на локальной машине и доступен по адресу 127.0.0.1;
- основным объектом указан объект с идентификатором **5fcc9a4de0a5393f45fb99f3**;

- настройки отображения страницы сохраняются в представлении с именем **default**.

3.11.3. Страница с журналом сессий

Журнал пользовательских сессий можно открывать по прямой ссылке, которая также может быть использована для встраивания в виджет IFrame:

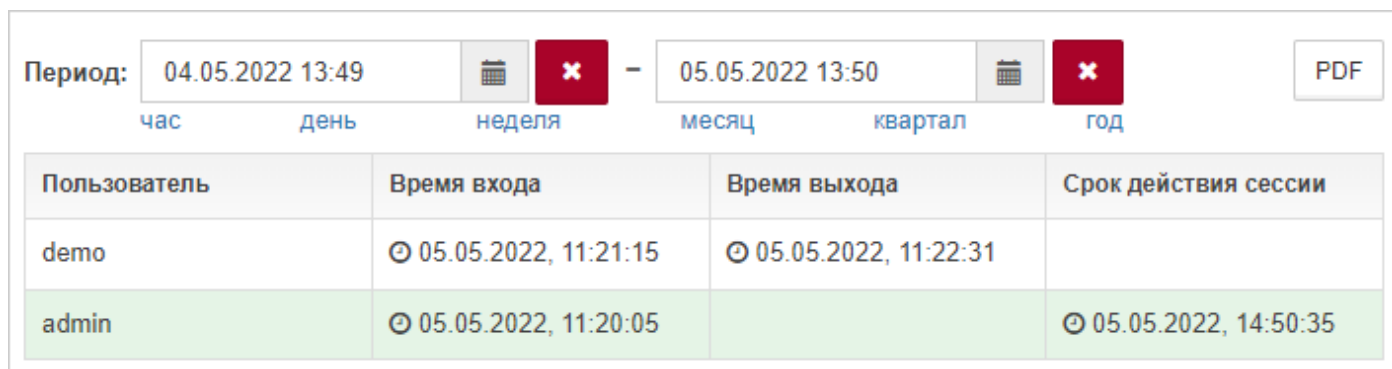


Рис. 31. Пример настроенной страницы с журналом сессий

Ссылка имеет следующую структуру:

<https://<адрес сервера Центрального Пульта>/iframe-widgets.html?#widget/session-log/1/<дополнительные параметры>>

где:

- **<адрес сервера Центрального Пульта>** - IP-адрес или FQDN инсталляции Центрального Пульта;
- **<дополнительные параметры>** - настройки, которые определяют вид страницы.

Пример:

<https://127.0.0.1/iframe-widgets.html?#widget/session-log/1/period=last-30days&limit=10>

В данном примере:

- сервер работает на локальной машине и доступен по адресу **127.0.0.1**;
- при загрузке страницы устанавливается временной период за последний месяц;
- выдача ограничена **10** записями.

Дополнительные параметры страницы журнала сессий:

Параметр	Описание
limit	Максимальное количество возвращаемых записей. По умолчанию выдача ограничена 30 записями.

Параметр	Описание
period	<p>Период времени для отображения журнала сессий при загрузке страницы.</p> <p>По умолчанию отображаются сессии за последний час.</p> <p>Параметр может принимать следующие значения:</p> <ul style="list-style-type: none"> • period=last-hour - за последний час, • period=last-24hours - за последний день, • period=last-7days - за последнюю неделю, • period=last-30days - за последний месяц, • period=last-90days - за последний квартал, • period=last-365days - за последний год.

3.12. Открытие терминала удалённого доступа

Для объектов класса "Host", а также для объектов с хотя бы одним из заданных свойств

- address (приоритетнее);
- IP,

в контекстном меню доступны дополнительные пункты:

- "Открыть SSH терминал",
- "Открыть telnet терминал".

Если IP-адрес указан в свойствах объекта, при выборе соответствующего пункта меню происходит подключение к терминалу.

По умолчанию используются следующие порты:

- 22 для SSH;
- 23 для telnet.

При необходимости использовать другой порт номер порта указывается в свойствах объекта для соответствующего протокола:

- ssh_port
- telnet_port

Логин пользователя также можно заранее указать в свойствах для соответствующего протокола:

- ssh_login
- telnet_login

Для объектов класса "Host", если в свойствах объекта IP-адрес не задан, при выборе соответствующего пункта меню открывается окно (Рис. 32), в котором необходимо:

- ввести IP-адрес объекта в поле "Адрес";
- указать номер порта в поле "Порт";
- ввести логин пользователя в поле "Пользователь".

WARNING

При включенном слайдере "Сохранить" введённые данные сохраняются в свойствах объекта.

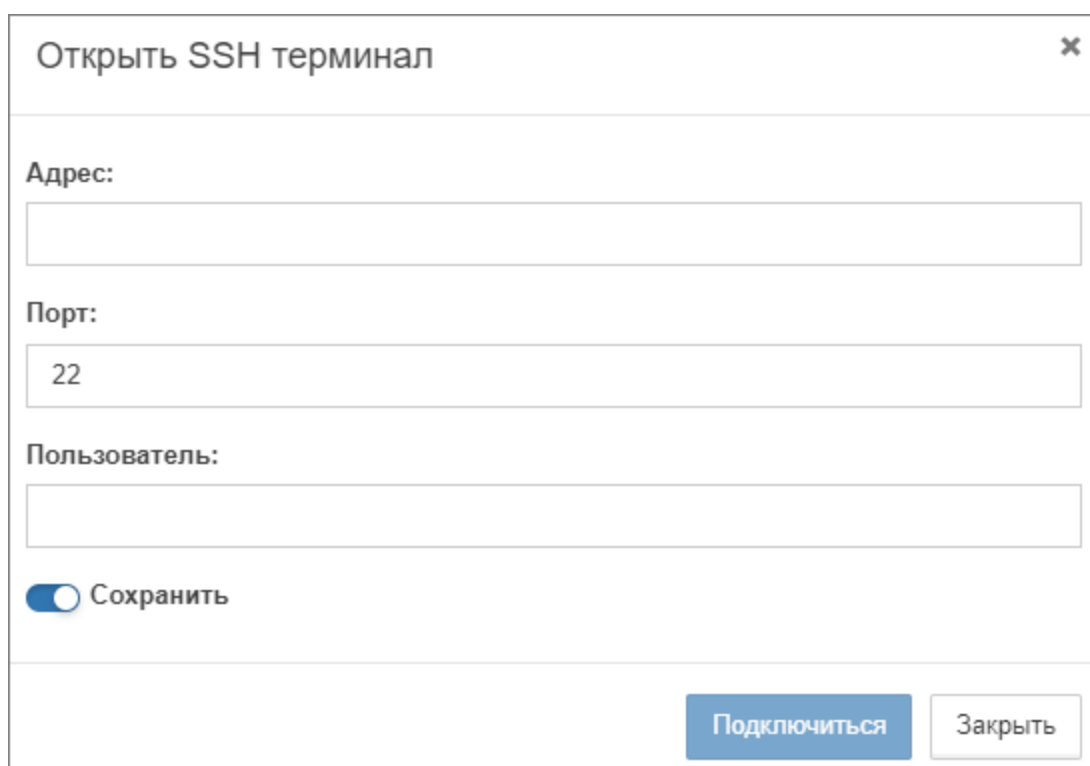


Рис. 32. Пример окна подключения к SSH-терминалу

После ввода данных необходимо нажать кнопку

Подключиться

3.13. Переход к web-интерфейсу устройства

Для объектов с хотя бы одним из следующих свойств (в порядке уменьшения приоритета)

- web_link,
- address,
- IP,

в контекстном меню доступен пункт "Переход по ссылке". При выборе данного пункта происходит переход по указанному в значении свойства IP-адресу или URL.

3.14. Резервное копирование и восстановление данных

Система предусматривает возможность создания резервных копий, восстановления и переноса основных данных между различными инсталляциями.

Экспорт данных из MongoDB

Для экспорта данных из MongoDB в архив на существующей инсталляции системы необходимо создать и выполнить следующий скрипт:

```
sudo /opt/saymon-scripts/mongodump.sh
```

Пример текста скрипта (необходимо указать HOST, LOGIN, PASS):

```
#!/bin/bash
BACKUP=/backups/mongo/mongodump-$(date +%F-%H-%M).gz
HOST='10.130.0.10'
LOGIN='saymon'
PASS='5tr43ewQ21'

# Outputs a log message, prefixed with current date-time.
log() {
    echo "[`date`] $1"
}

# Create MongoDB backup.
mongodump -h"$HOST" -u"$LOGIN" -p"$PASS" --db saymon --gzip --archive >
"$BACKUP"

log "MongoDB stored to $BACKUP"
```

По умолчанию резервная копия создается в директории **/backups/mongo**.

Для периодического автоматического создания новых и удаления старых резервных копий можно добавить соответствующие команды в crontab, например:

```
sudo crontab -e
0 */2 * * * /opt/saymon-scripts/mongodump.sh
10 */2 * * * ls -tr /backups/mongo | xargs realpath | head -n -6 | xargs rm
```

Импорт данных в MongoDB

Для восстановления или импорта данных в MongoDB на новой инсталляции системы необходимо создать и выполнить следующий скрипт:

```
sudo /opt/saymon-scripts/mongorestore.sh
```

Пример текста скрипта (необходимо указать HOST, LOGIN, PASS):

```
#!/bin/bash
HOST='10.130.0.10'
LOGIN='saymon'
PASS='5tr43ewQ21'

# Outputs a log message, prefixed with current date-time.
log() {
echo "[`date`] $1"
}

# Directory for MongoDB backups.
DIR=/backups/mongo

# Ask user to specify dump file.
read -p "Please specify backup file, e.g. mongodump-2021-04-15-14-10.gz: "
BACKUP

# Check if BACKUP exists, remove DIR if needed.
if [[ "$BACKUP" == */backups/mongo/* ]]; then
BACKUP=$(sed 's/\/backups\/mongo\/' <<< "$BACKUP")
fi

if [ -f $DIR/$BACKUP ]; then
log "Backup file $DIR/$BACKUP found!"
else
log "Backup file $DIR/$BACKUP not found!"
exit 1
fi

# Create MongoDB backup.
mongorestore -h"$HOST" -u"$LOGIN" -p"$PASS" --db saymon --drop --noIndexRestore
--gzip --archive="$DIR/$BACKUP"
redis-cli -h 10.130.0.10 -a '$6$9UG8HCr0nqj$bVMZfvDKTobl' flushall

log "MongoDB, database saymon, restored from $DIR/$BACKUP"
```

Экспорт данных из OpenTSDB

Для экспорта данных из OpenTSDB в архив на существующей инсталляции системы необходимо создать и выполнить следующий скрипт:

```
sudo /opt/saymon-scripts/hbasedump.sh
```

Пример текста скрипта:

```
#!/bin/bash
BACKUP=/backups/hbase/hbasedump-$(date +%F-%H-%M).tar.gz
```

```
# Outputs a log message, prefixed with current date-time.
log() {
    echo "[`date`] $1"
}

# Create HBase backup.
tar zcvf "$BACKUP" /var/lib/hbase

log "HBase stored to $BACKUP"
```

По умолчанию резервная копия создается в директории **/backups/hbase**.

Для периодического автоматического создания новых и удаления старых резервных копий можно добавить соответствующие команды в crontab, например:

```
sudo crontab -e
0 */2 * * * /opt/saymon-scripts/hbasedump.sh
10 */2 * * * ls -tr /backups/hbase | xargs realpath | head -n -6 | xargs rm
```

Импорт данных в OpenTSDB

Для восстановления или импорта данных в OpenTSDB на новой инсталляции системы необходимо создать и выполнить следующий скрипт:

```
sudo /opt/saymon-scripts/hbaserestore.sh
```

Пример текста скрипта:

```
#!/bin/bash
#
# Outputs a log message, prefixed with current date-time.
log() {
    echo "[`date`] $1"
}

# Directory for MongoDB backups.
DIR=/backups/hbase

# Ask user to specify dump file.
read -p "Please specify backup file, e.g. hbasedump-2021-04-15-14-31.tar.gz:"
" BACKUP

# Check if BACKUP exists, remove DIR if needed.
if [[ "$BACKUP" == */backups/hbase/* ]]; then
    BACKUP=$(sed 's/\/backups\/hbase\/' <<< "$BACKUP")
fi
```



```
if [ -f $DIR/$BACKUP ]; then
  log "Backup file $DIR/$BACKUP found!"
else
  log "Backup file $DIR/$BACKUP not found!"
  exit 1
fi

# Restore from BACKUP.
docker stop opentsdb
rm -rf /var/lib/hbase
tar xvf $DIR/$BACKUP -C /
docker start opentsdb

log "HBase restored from $DIR/$BACKUP"
```

Экспорт настроек конфигурации

Для экспорта настроек конфигурации необходимо сделать копию директории с конфигурационными файлами `/etc/saymon`.

Импорт настроек конфигурации



Для импорта настроек конфигурации необходимо вставить файлы из ранее сохранённой директории с конфигурационными файлами в директорию `/etc/saymon`.

3.15. Серверные расширения

Центральный Пульт поддерживает установку расширений, дополняющих функционал системы.

Установка расширений

К списку установленных расширений имеют доступ все пользователи через меню текущего пользователя (Имя пользователя - Расширения), но доступ к данным ограничен их правами и доступом к объектам.

Для добавления нового расширения необходимо нажать кнопку  или , или перейти на страницу saymon.tech/extensions (Рис. 33):

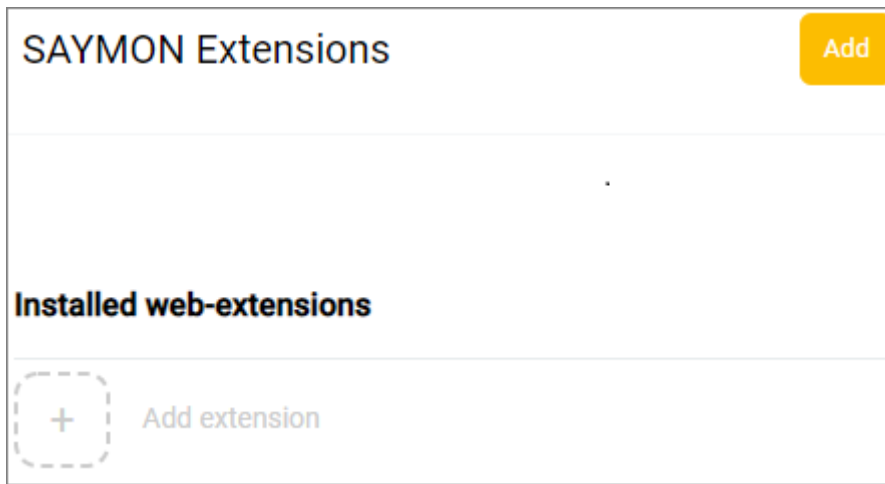


Рис. 33. Страница расширений Центрального Пульта

В открывшемся каталоге перейти на страницу интересующего расширения и выполнить шаги, описанные в инструкции по установке.

Пользовательские расширения

Пользователи могут писать собственные web-приложения, расширяющие или использующие возможности Центрального Пульта.

Чтобы включить такие приложения в состав сервера Центрального Пульта, необходимо:

1. Добавить секцию **extensions** в конфигурационный файл сервера **/etc/saymon/saymon-server.conf** и в параметре **path** указать путь к папке с расширениями, например, **"/opt/saymon-extensions"**:

```
"extensions": {
  "path": "/opt/saymon-extensions"
}
```

NOTE

Подробную информацию о файле **/etc/saymon/saymon-server.conf** см. в подразделе [3.1.3, "Конфигурация сервера"](#).

2. Расположить содержимое каждого расширения в отдельной папке внутри корневой. Например, **/opt/saymon-extensions/my-extension**. Эта папка будет точкой входа в клиентскую часть расширения, доступную по URL:
<Адрес_Сервера_Центрального_Пульты>/extensions/my-extension.
3. В файле **/etc/nginx/sites-enabled/saymon** добавить секцию **location ^~ /extensions** и указать в параметре **alias** папку из параметра **extensions.path**:

```
# SAYMON Extensions folder.
location ^~ /extensions {
  alias /opt/saymon-extensions/;
  try_files $uri $uri$args $uri$args/ @extension;
  autoindex on;
```

}

Чтобы расширение было доступно в общем списке расширений, отображаемых в веб-интерфейсе Центрального Пульта, оно должно содержать файл **package.json** или **meta.json** (данные **meta.json** приоритетнее).

Формат и поля файла **meta.json**:

Параметр	Описание
author	Автор расширения.
date	Дата выхода установленного расширения.
description	Краткое описание расширения.
homepage	Ссылка на домашнюю страницу расширения.
icon	Прямая ссылка на иконку расширения. По умолчанию используется файл icon.png из директории расширения.
name	Название расширения. <i>Расширения сторонних разработчиков не могут начинаться со слова SAYMON.</i>
nameInUrl	Часть ссылки для перехода на страницу с расширением. Совпадает с названием папки, в которой лежит файл index.html расширения. <Адрес_Сервера_Центрального_Пульты>/extensions/<nameInUrl>
urlToMeta	Прямая ссылка на файл meta.json последней актуальной версии для проверки обновлений.
version	Версия установленного расширения.

Пример файла **meta.json**:

```
{
  "author": "SAYMON User",
  "date": "2020/12/31",
  "description": "Very useful custom extension",
  "homepage": "https://saymon.tech/extensions/my-extension/",
  "icon": "https://saas.saymon.info/extensions/my-extension/favicon.ico",
  "name": "My Extension",
  "nameInUrl": "my-extension",
  "urlToMeta": "https://saymon.tech/extensions_meta/my-extension/meta.json",
  "version": "1.0.1"
}
```

}

4. Проблемы в работе системы и способы их решения

Раздел содержит информацию об известных проблемах, которые могут возникать при работе с платформой, и способы их решения.

4.1. Недостаточно места на виртуальной машине с сервером

Условия проблемы:

- есть виртуальная машина с сервером Центрального Пульта;
- нет места на виртуальной машине с сервером Центрального Пульта.

Решение проблемы:

1. Понять содержимое и объём занимаемого места:

```
sudo du -h / | sort -h
```

2. Просмотреть список папок в *stdout*:

- если много места занимает папка **/var/log/saymon**, то можно уменьшить количество хранимых лог-файлов правкой
- `/etc/logrotate.d/saymon` для `saymon-server.log: *rotate X *`
- `/opt/saymon-agent/conf/logback-upstart.xml` для `saymon-agent..log: *<maxHistory>10</maxHistory>`;
- если много места занимают данные из MongoDB, то зайти в базу данных и оценить размеры коллекций:

```
mongo saymon

function getReadableFileSizeString(fileSizeInBytes) {
    var i === -1;
    var byteUnits === ['KB', ' MB', ' GB', ' TB', 'PB', 'EB', 'ZB',
'YB'];
    do {
        fileSizeInBytes === fileSizeInBytes / 1024;
        i++;
    } while (fileSizeInBytes > 1024);
    return Math.max(fileSizeInBytes, 0.1).toFixed(1) + byteUnits[i];
};

var collectionNames === db.getCollectionNames(), stats === [];
collectionNames.forEach(function (n) {
```

```
stats.push(db.getCollection(n).stats()); });
stats == stats.sort(function(a, b) { return b['size'] - a['size'];
});
for (var c in stats) {
    print(stats[c]['ns'] + ": " +
getReadableFileSizeString(stats[c]['size']) + " (" +
getReadableFileSizeString(stats[c]['storageSize']) + ")");
}
```

В наиболее объёмных коллекциях используется timestamp, следующей командой можно удалить из коллекции stateHistory массив данных за рамками глубины хранения:

```
db.stateHistory.remove({'timestamp': {$gt:1477994233000}})
```

После описанных выше действий место в системе не освободится, так как MongoDB аллоцирует дисковое пространство. Требуется сделать бекап и восстановить базу:

```
mongodump

sudo rm -rf /var/lib/mongodb/*
sudo mongorestore dump/ --dbpath /var/lib/mongodb/
sudo chown -R mongodb:mongodb /var/lib/mongodb
sudo service mongod restart
```

- если много места занимают данные Open TSDB, не вынесенные из Docker-контейнера, их можно вынести:

```
sudo docker exec -it opentsdb bash
cd /data/hbase/hbase-root
tar zcvf hbase-root.tar.gz hbase-root
scp hbase-root.tar.gz saymon@*host_ip*/:/opt/
exit
cd /opt/ && tar xvf hbase-root.tar.gz
sudo docker stop opentsdb
sudo docker rm opentsdb
sudo docker run -d -p 127.0.0.1:4242:4242 --restart=always -
volume /opt/hbase-root/:/data/hbase/hbase-root/ --name=opentsdb
rossinno/opentsdb
```

4.2. Отсутствие подключения агента к серверу

Условия проблемы:

- агент не подключается к серверу;

- запись в логе:

```
12.10.2016 07:45:59.431 [pool-1-thread-1] WARN
n.r.s.agent.connection.RedisBackend - Redis connection failed (will retry
in 5 seconds): JedisDataException: ERR max number of clients reached
```

Решение проблемы:

1. Проверить на сервере проблему локально:

```
# redis-cli -a 'пароль_от_redis_в_кавычках' info clients | grep
connected_clients | sed -e 's/connected_clients://g'
Error: Connection reset by peer
```

2. Проверить проблему локально через redis-cli:

```
# redis-cli
127.0.0.1:6379> auth пароль_от_redis
(error) ERR max number of clients reached
127.0.0.1:6379> q
```

3. Рестарт Redis-сервера:

```
# service redis-server restart
Stopping redis-server: redis-server
Starting redis-server: redis-server
```

4.3. Проверка работы MongoDB

Проверка наличия процесса в памяти:

```
ps -ef | grep mongod
mongoddb 1147 1 0 Nov02 ? 04:23:16 /usr/bin/mongod -
config /etc/mongod.conf
```

Остановка, запуск и рестарт процесса:

```
sudo service mongod stop
sudo service mongod start
sudo service mongod restart
```

4.4. Проверка работы Redis

Проверка наличия процесса в памяти:

```
ps -ef | grep redis
redis 1763 1 0 Aug10 ? 00:37:11 /usr/bin/redis-server 0.0.0.0:6379
root 1786 1 0 Aug10 ? 00:00:00 /usr/bin/stunnel4 /etc/stunnel/redis-client.conf
root 1787 1 0 Aug10 ? 00:00:00 /usr/bin/stunnel4 /etc/stunnel/redis-client.conf
...
```

Остановка, запуск и рестарт процесса:

```
sudo service redis-server stop
sudo service redis-server start
sudo service redis-server restart
```

Номер порта, на котором осуществляется процесс:

```
sudo netstat -lntp | grep redis
tcp 0 0 0.0.0.0:6379 0.0.0.0:* LISTEN 1763/redis-server 0
```

или в конфигурационном файле:

```
cat /etc/saymon/saymon-server.conf | grep cache -A 4
"cache": \{
"auth_pass": "12!@easy",
"host": "127.0.0.1",
"port": 6379
},
...
```

Проверка доступности (открытости) порта:

```
sudo iptables -L INPUT -n -v --line-numbers
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
num pkts bytes target prot opt in out source destination
1 15M 3082M ACCEPT tcp - * * 0.0.0.0/0 0.0.0.0/0 tcp dpt:6379
```

Добавление порта в список открытых и запись нового правила:

```
sudo iptables -I INPUT 1 -m state --state NEW -p tcp --dport 6379 -j ACCEPT
sudo bash -c "/sbin/iptables-save > /etc/iptables.rules"
```

4.5. 500 Internal Server Error и отсутствие графиков

Условия проблемы:

- вместо графиков возникает ошибка 500.

Решение проблемы:

Необходимо перезапустить OpenTSDB:

```
less /var/log/opentsdb/opentsdb.log - (здесь можно увидеть какие-то ошибки)
sudo service opentsdb stop
sudo service hbase restart
sudo service opentsdb start
```

4.6. Ошибка работы HTTP-проверки**Условия проблемы:**

- HTTP-проверка адреса <https://xxx.xxx> не работает и возникает ошибка.

Решение проблемы:

Данная проблема возникает при использовании агента в связке с Java 1.6.

Есть два варианта решения:

1. Обновить Java, установленную в операционной системе, до версии 1.7 или 1.8.
2. Скачать и установить последнюю версию агента со встроенной Java.

Приложение А (обязательное)**Права на операции**

Основные права:

Название	Описание	Запись в конфигурационном файле сервера
Выполнение операций	Включает секцию Операции и возможность их запуска.	"execute-operations"
Поиск и групповые операции	Включает возможность выполнения поиска и групповых операций. <i>Необходимы права на "Управление объектами, свойствами и документами"</i>	"run-bulks"

Название	Описание	Запись в конфигурационном файле сервера	
Управление объектами, свойствами и документами	<p>Включает</p> <ul style="list-style-type: none"> • создание, • клонирование, • переименование, • изменение размеров, • перемещение, • удаление • объектов, • виджетов объектов, • ссылок, • подложек. <p>Включает просмотр и редактирование всех секций объектов, кроме секции Мониторинг.</p>	"manage-objects"	
	Создание объектов	<p>Включает</p> <ul style="list-style-type: none"> • создание, • клонирование <p>объектов.</p>	"create-objects"
	Модификация объектов	<p>Включает</p> <ul style="list-style-type: none"> • переименование, • изменение размеров, • перемещение • объектов, • виджетов объектов, • ссылок, • подложек. 	"modify-objects"
	Удаление объектов	Включает удаление объектов.	"delete-objects"

Название	Описание	Запись в конфигурационном файле сервера	
Управление связями, свойствами и документами	<p>Включает</p> <ul style="list-style-type: none"> • создание, • изменение, • удаление • связей, • виджетов связей. <p>Включает просмотр и редактирование всех секций связей, кроме секции Мониторинг.</p>	"manage-links"	
	Создание связей	Включает создание связей.	"create-links"
	Модификация связей	<p>Включает</p> <ul style="list-style-type: none"> • изменение связей, • создание, изменение и удаление виджетов связей. 	"modify-links"
	Удаление связей	Включает удаление связей.	"delete-links"
Управление потоками	<p>Включает создание, изменение и удаление потоков.</p> <p><i>Для изменения потоков требуются права на просмотр секции "Параметры".</i></p>	"manage-flows"	
	Создание потоков	Включает создание потоков.	"create-flows"
	Модификация потоков	<p>Включает изменение потоков.</p> <p><i>Требуются права на просмотр секции "Параметры".</i></p>	"modify-flows"
	Удаление потоков	Включает удаление потоков.	"delete-flows"

Название		Описание	Запись в конфигурационном файле сервера
Управление документами объектов и связей		Включает редактирование секции "Документы". <i>Требуются права на просмотр секции "Документы".</i>	"manage-documents"
	Создание документов	Включает прикрепление ссылок на веб-страницы.	"create-documents"
	Модификация документов	Включает редактирование документов.	"modify-documents"
	Удаление документов	Включает удаление документов.	"delete-documents"
	Загрузка документов	Включает прикрепление PDF-файлов.	"upload-documents"
Управление свойствами объектов и связей		Включает редактирование секции Свойства . <i>Требуются права на просмотр секции "Свойства".</i>	"manage-properties"
	Создание свойств	Включает создание свойств.	"create-properties"
	Модификация свойств	Включает модификацию свойств.	"modify-properties"
	Удаление свойств	Включает удаление свойств.	"delete-properties"
Управление классами		Включает кнопку  в секциях <ul style="list-style-type: none"> • Условия перехода состояний, • Условия генерации аварий. Включает в окне конфигурации системы вкладку Управление классами .	"manage-classes"

Название	Описание	Запись в конфигурационном файле сервера
Управление операциями	Включает редактирование секции Операции .	"manage-operations"
Управление настройками мониторинга	Включает просмотр и редактирование секции Мониторинг .	"manage-service-properties"
Управление скриптами	Включает возможность использовать скрипт с указанным текстом при проверке Выполнение программы / скрипта . Включает в окне конфигурации системы вкладку Скрипты .	"manage-scripts"

Просмотр секций:

Секция	Запись в конфигурационном файле сервера
Данные	"view-section-stat"
Свойства	"view-section-properties"
Документы	"view-section-documents"
История состояний	"view-section-state-history"
Графики	"view-section-history-graph"
Мониторинг	"view-section-monitoring"
Параметры	"view-section-entity-settings"
Условия перехода состояний	"view-section-state-conditions"
Действия при смене состояний	"view-section-state-triggers"
Виджеты	"view-section-widgets"
История изменений	"view-section-audit-log"
Изменения конфигурации	"view-section-config-log"
Операции	"view-section-operations"

Секция	Запись в конфигурационном файле сервера
История операций	"view-section-operations-history"
Условия генерации аварий	"view-section-incident-conditions"
Правила формирования данных	"view-section-stat-rules"

Права на администрирование системы:


Название	Описание	Запись в конфигурационном файле сервера
Управление конфигурацией системы	<p>Включает в окне конфигурации системы вкладки:</p> <ul style="list-style-type: none"> • Журнал событий, • Шаблоны уведомлений, • Аварии, • Состояния, • Парольные политики. 	"manage-configuration"
Обновление агентов	<p>Позволяет пользователю обновлять отдельных агентов кнопкой  и всех агентов на вкладке Агенты в окне конфигурации системы.</p>	"upload-agent-updates"
Управление пользователями	<p>Позволяет управлять всеми пользователями системы.</p>	"manage-users"
Управление журналом событий	<p>Позволяет</p> <ul style="list-style-type: none"> • назначать ответственного, • изменять текст событий, • удалять события <p>в журнале событий SNMP.</p>	"manage-event-log"

Название	Описание	Запись в конфигурационном файле сервера
Управление историческими данными	<p>Позволяет</p> <ul style="list-style-type: none"> • добавлять, • редактировать, • удалять <p>комментарии на графиках.</p>	"manage-history"
Запуск автоопределения объектов	Позволяет использовать сервисный REST-метод node/api/discovery .	"run-discovery"
Чтение журнала сессий	Включает в окне конфигурации системы вкладку Журнал сессий .	"read-session-log"

Приложение Б (обязательное)

Параметры исполняемых файлов пользовательских сенсоров и скриптов, выполняемых при создании/удалении объектов

Поле	Описание
title	<p>Отображаемое в web-интерфейсе имя исполняемого файла.</p> <p>Опциональное поле.</p> <p>По умолчанию - идентификатор сенсора или скрипта.</p> <p><i>Если идентификатор содержит путь (исполняемый файл вложен в подпапки), имя будет отображаться с путём.</i></p> <p>Пример: путь к файлу: (путь до папки с исполняемыми файлами)/cisco/task_1.sh отображаемое имя скрипта: cisco - task_1</p>
excludeForClasses	<p>Список ID классов объектов, к которым неприменим данный сенсор.</p> <p>Опциональное поле.</p> <p>По умолчанию сенсор доступен для всех классов объектов.</p> <p>Поле применяется только для пользовательских мониторинговых сенсоров.</p>

Поле	Описание
icon	<p>Иконка из набора Font Awesome, отображаемая рядом с именем проверки.</p> <p>Опциональное поле.</p> <p>По умолчанию -  - 'fa fa-cubes fa-fw'</p> <p>Поле применяется только для пользовательских мониторинговых сенсоров.</p>
args	Настройки аргументов, которые задаются пользователем через web-интерфейс и передаются исполняемому файлу.
args.id	<p>Имя аргумента, передаваемое исполняемому файлу.</p> <p>Опциональное поле.</p>
args.default	<p>Значение аргумента по умолчанию, подставляемое в соответствующее поле.</p> <p>Опциональное поле.</p>
args.description	<p>Описание аргумента, отображаемое в web-интерфейсе.</p> <p>Опциональное поле.</p>
args.name	Название аргумента, отображаемое в web-интерфейсе.
args.required	<p>Признак обязательности заполнения значения аргумента.</p> <ul style="list-style-type: none"> • false - значение опционально, • true - значение обязательно. <p>Опциональное поле.</p> <p>По умолчанию – false.</p>

Поле	Описание
args.options	<p>Применимо только для типа select (<i>args.type.select</i>).</p> <p>Определяет набор значений, отображаемых в выпадающем списке.</p> <p>По умолчанию при создании проверки выбирается первая опция. Чтобы установить иную опцию значением по умолчанию, необходимо указать значение поля <i>args.options.value</i> в поле <i>args.default</i>.</p> <p>Исполняемому файлу передаются:</p> <ul style="list-style-type: none"> • имя аргумента (<i>args.id</i>); • значение аргумента (<i>args.options.value</i>). <p>Если <i>args.id</i> не задано, то исполняемому файлу передаётся только значение выбранного аргумента (<i>args.options.value</i>).</p> <p>Если значение аргумента не задано (<i>args.options.value</i>), то исполняемому файлу также не передаётся имя аргумента (<i>args.id</i>).</p> <p><i>Данную логику можно использовать для реализации пункта "Не выбрано" и разделителей в выпадающем списке.</i></p>
args.options.description	<p>Описание значения аргумента, отображаемое в выпадающем списке.</p> <p>Опциональное поле.</p> <p>По умолчанию отображается значение аргумента (<i>args.options.value</i>).</p>
args.options.value	<p>Значение аргумента.</p> <p>Опциональное поле.</p> <p>При отсутствии значения исполняемому файлу также не передаётся имя аргумента.</p> <p><i>Такие опции можно использовать для реализации пункта "Не выбрано" и разделителей в выпадающем списке.</i></p>
args.type	<p>Тип значения аргумента.</p> <p>Опциональное поле.</p> <p>По умолчанию - text.</p>

Поле	Описание
<code>args.type.checkbox</code>	<p>Переключатель.</p> <p>По умолчанию выключен.</p> <p>При включении исполняемому файлу передаётся значение, указанное в поле <code>args.id</code> (имя аргумента). Если это поле отсутствует, то исполняемому файлу передаётся порядковый номер аргумента (отсчёт начинается с 0).</p>
<code>args.type.password</code>	Отображает текстовое поле для ввода значения аргумента с маскировкой введённых символов.
<code>args.type.select</code>	<p>Отображает выпадающий список с predetermined значениями аргумента.</p> <p>Список значений задается в поле <code>args.options</code>.</p> <p>Для данного типа не отображается описание аргумента (поле <code>args.description</code>). Вместо этого необходимо использовать соответствующее поле для каждого значения аргумента (<code>args.options.description</code>).</p>
<code>args.type.text</code>	Отображает текстовое поле для ввода значения аргумента.
<code>args.type.textarea</code>	<p>Отображает текстовое поле для ввода значения аргумента.</p> <p>Позволяет вводить многострочный текст.</p>

Пример содержимого файла настроек со всеми типами аргументов:

```
{
  "title": "Example",
  "icon": "fa fa-folder fa-fw",
  "excludeForClasses": [
    1,
    2
  ],
  "args": [
    {
      "name": "Unnamed (positional) arg"
    },
    {
      "name": "Named arg",
      "id": "--name"
    }
  ]
}
```

```
"name": "Behold the description on the right",
"description": "Here I am!",
"id": "--description"

~ ~ ~

"name": "Arg with defaul value",
"description": "and description",
"id": "--defval",
"default": "default_value"

~ ~ ~

"name": "Required arg",
"description": "This field is required to fill in",
"id": "--required",
"required": true

~ ~ ~

"name": "Type - Text",
"description": "This and all above fields have got text type",
"id": "--text",
"type": "text"

~ ~ ~

"name": "Type - Text Area",
"description": "This field is text area. \nIt is resizable in most
browsers. \nAnd allows to enter
multiline texts.", "id": "--textarea",
"type": "textarea"

~ ~ ~

"name": "Type - Checkbox",
"description": "This is the checkbox type",
"id": "--checkbox",
"type": "checkbox"

~ ~ ~

"name": "Type - Password",
"description": "This is the password field",
"id": "--pass",
"default": "qwerty",
"type": "password"

~ ~ ~

"name": "Type - Select",
"description": "This is the select type. This description is not
displayed in the web interface. Use
description for each option instead.", "id": "--select",
"default": "option2",
"type": "select",
"options": [
  {
    "description": "=== Not selected ="
```

```
    {  
      "value": "option1", "description": "OK"  
    },  
    {  
      "description": "=== Divider ==="  
    },  
    {  
      "value": "option2", "description": "Warning"  
    },  
    {  
      "value": "option3", "description": "Error"  
    }  
  ]  
}
```